КОМП'ЮТЕРНІ МЕРЕЖІ

КОМП'ЮТЕРНІ МЕРЕЖІ

Міністерство освіти і науки України
Вінницький національний технічний університет

КОМП’ЮТЕРНІ МЕРЕЖІ

Підручник

Вінниця
ВНТУ
2020

УДК 004.7 (075)
К63

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 5 від 19.12.2019 р.)

Автори:
Азаров О. Д., Захарченко С. М., Кадук О. В., Орлова М. М., Тарасенко В. П.

Підручник для студентів закладів вищої освіти, які навчаються за напрямами підготовки «Комп'ютерна інженерія» та «Програмна інженерія»

Рецензенти:
І. А. Жуков, доктор технічних наук, професор
Р. Н. Квєтний, доктор технічних наук, професор
Л. Б. Ліщинська, доктор технічних наук, професор

Комп'ютерні мережі : підручник / [Азаров О. Д., Захарченко С. М., Кадук О. В. та ін.]. − Вінниця : ВНТУ, 2020. − 378 с.
К63
ISBN 978-966-641-808-4

Підручник складається з дев'яти розділів. Матеріал розташовано в логічній послідовності, тому роботу з підручником доцільно починати з першого розділу. Укінці кожного розділу є питання для самоперевірки, що дозволяють самостійно перевірити ступінь засвоєння навчального матеріалу. Підручник призначений для студентів напряму підготовки 123 − «Комп'ютерна інженерія».

УДК 004.7 (075)

ISBN 978-966-641-808-4

ЗМІСТ

ВСТУП

Потреба у взаємодії з іншими людьми є однією з основних потреб людини. Спілкування так само важливе для людей, як повітря, вода, їжа і дах над головою. Протягом свого існування людство постійно винаходило та вдосконалювало шляхи комунікації, починаючи з тірольських співів та голубиної пошти і завершуючи мережею Інтернет. У сучасному світі за рахунок використання мереж ми пов'язані один з одним, як ніколи раніше. Люди можуть миттєво обмінюватись новинами, ідеями, спільно створювати нові проекти, грати в ігри з іншими людьми, що знаходяться на інших континентах. Комп'ютерні мережі широко застосовують промислові компанії, банки, державні й комерційні установи. Саме тому Білл Гейтс, один із засновників компанії Microsoft, назвав мережу Інтернет нервовою системою сучасної світової економіки.

Цей підручник є результатом творчої співпраці колективів кафедри системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» та кафедри обчислювальної техніки Вінницького національного технічного університету і призначений, в першу чергу, для студентів спеціальності 123 – «Комп'ютерна інженерія» при вивченні дисципліни «Комп'ютерні мережі».

Навчальна дисципліна «Комп'ютерні мережі» є однією з базових для підготовки бакалаврів. Метою викладання дисципліни є формування знань, умінь і навичок для проектування, налаштування, обслуговування та адміністрування сучасних комп'ютерних мереж. Під час вивчення дисципліни студенти отримують інформацію про сучасні принципи побудови комп'ютерних мереж, протоколи їх функціонування, досвід налаштування мережного обладнання. Основне завдання дисципліни «Комп'ютерні мережі» – дати студентам теоретичну та практичну підготовку в галузі проектування та експлуатації сучасних комп'ютерних мереж.

Цей підручник допоможе студентам засвоїти принципи функціонування комп'ютерних мереж та призначення найпоширеніших мережних протоколів; методи та засоби побудови й обслуговування сучасних комп'ютерних мереж різного виду та призначення; тенденції розвитку програмних та апаратних засобів комп'ютерних мереж. Практична частина підручника містить потужний лабораторний практикум і дозволить студентам отримати досвід проектування комп'ютерних мереж і налаштування мережевого обладнання.

Підручник складається з дев'яти розділів. Матеріал розташовано в логічній послідовності, тому роботу з підручником доцільно починати з першого розділу. В кінці кожного розділу є питання для самоперевірки, які дозволяють самостійно перевірити ступінь засвоєння навчального матеріалу.

Перший розділ присвячено основним принципам та архітектурним рішенням побудови комп'ютерних мереж. У цьому розділі можна також ознайомитись з принципами адресації та методами комутації в сучасних мережах. Розглянуто ієрархічні моделі для опису комп'ютерних мереж.

У **другому розділі** розглянуто особливості реалізації фізичного рівня комп'ютерних мереж, зокрема структуру, класифікацію та характеристики каналів передавання даних, різновиди існуючих кабельних систем, проаналізовано їх переваги, недоліки та наведено рекомендації до застосування. Розглянуто сучасні методи передавання цифрових даних на фізичному рівні та способи мультиплексування потоків даних.

У **третьому розділі** описано технології реалізації канального рівня сучасних комп'ютерних мереж, методи доступу до середовища та методи керування логічним каналом. Розглянуто особливості реалізації канального рівня в локальних мережах на прикладі технологій Ethernet, Token Ring та FDDI. Описано основи функціонування комутаторів локальних мереж.

Четвертий розділ присвячено мережному рівню. Розглянуто принципи ієрархічної адресації та алгоритми маршрутизації потоків даних. Описано шляхи реалізації мережного рівня в сучасних мережах на прикладі протоколу IPv4. Розглянуто протоколи динамічної маршрутизації RIP та OSPF. Також описано основи роботи з мережною операційною системою на прикладі Cisco IOS. В кінці розділу розглянуто нову версію протоколу IP – IPv6 та методи його взаємодії з IPv4.

У **п'ятому розділі** розглянуто роботу транспортного рівня та особливості його реалізації на прикладі протоколів TCP та UDP. Описано механізми гарантованої передачі даних, методи керування потоками даних і боротьби з перевантаженнями.

У **шостому розділі** описано поширені протоколи верхнього рівня, зокрема протокол динамічного призначення адрес DHCP, протокол перетворення доменних імен DNS, протоколи віддаленого доступу Telnet та SSH, поштові протоколи POP та SMTP, протоколи передачі файлів FTP та TFTP, протокол гіпертекстових повідомлень HTTP та інші.

Сьомий розділ присвячено сучасним трендам розвитку комп'ютерних мереж, зокрема Інтернету речей, хмарним технологіям та центрам обробки даних, основам віртуалізації, програмно керованим мережам.

В **восьмому розділі** розглянуто методи реалізації сучасних цифрових мереж, зокрема існуючі ієрархії цифрових каналів PDH та SDH. Також описано технології віртуальних каналів Frame Relay та ATM і технологію багатопротокольної комутації по мітках MPLS.

Дев'ятий розділ присвячено опису технологій безпроводового зв'язку від персональних до глобальних.

Автори сподіваються, що підручник допоможе студентам поглибити свої знання в галузі комп'ютерних мереж і бажають успіхів у навчанні.

1 БАЗОВІ ПРИНЦИПИ ПОБУДОВИ КОМП'ЮТЕРНИХ МЕРЕЖ

1.1 Історія розвитку та класифікація комп'ютерних мереж

Комп'ютерні мережі є логічним результатом еволюції двох найважливіших науково-технічних галузей сучасної цивілізації – комп'ютерних і телекомунікаційних технологій. З одного боку, комп'ютерні мережі – це частковий випадок розподілених обчислювальних систем, в яких певна група комп'ютерів виконує набір задач і обмінюється інформацією. А з іншого боку, комп'ютерні мережі можна розглядати як засіб передавання інформації на великі відстані.

Прообразом комп'ютерних мереж стали багатотермінальні системи, що забезпечували доступ до ресурсів так званого мейнфрейму (mainframe) і працювали в режимі з розподілом часу. Історію мейнфреймів прийнято відраховувати з появи в 1964 році універсальної комп'ютерної системи IBM 360. У користувачів цієї системи створювалось враження, що кожний з них працює на окремому комп'ютері, хоча, насправді, один комп'ютер обслуговував всіх по черзі, як гросмейстер під час сеансу одночасної гри. До речі, схожим чином відбувається обслуговування клієнтів сучасних хмарних сервісів.

З появою перших мейнфреймів з'явилась необхідність у віддаленому під'єднанні та об'єднанні мейнфреймів, що знаходилися на великих відстанях. Спочатку було вирішено задачу доступу до комп'ютера через термінал, що знаходиться на значній відстані, з використанням телефонних ліній зв'язку та модемів. Далі з'явилися системи, в яких разом зі з'єднанням типу термінал-комп'ютер було реалізовано з'єднання комп'ютер-комп'ютер. Комп'ютери отримали можливість обмінюватися даними в автоматичному режимі. На основі даного принципу було створено мережі, де реалізовано служби обміну файлами, синхронізації баз даних, електронної пошти тощо.

Таким чином, у хронологічному порядку першими з'явилися глобальні мережі WAN (Wide Area Network). На початкових етапах WAN будувалися на основі телефонних ліній зв'язку. Нововведенням, що прийшло разом з ними, стала відмова від принципу комутації каналів, що використовувався у телефонних мережах. Пульсуючий (інтенсивний обмін чергується зі значними паузами) і нечутливий до затримок комп'ютерний трафік набагато ефективніше передавати у мережах, що працюють за принципом комутації пакетів. Перші глобальні мережі будувалися з використанням аналогових телефонних каналів низької якості та швидкості, що впливало на реалізацію протоколів передачі даних. Типовим прикладом є сімейство протоколів X.25.

У 1969 році міністерство оборони США ініціювало розробку мережі для об'єднання комп'ютерів оборонних і науково-дослідних центрів. Дана

мережа отримала назву ARPANET (Advanced Research Projects Agency Network) і стала початковою точкою для створення глобальної мережі Internet. Мережа ARPANET об'єднувала комп'ютери різних типів, які працювали під керуванням різних операційних систем з додатковими модулями, що реалізовували функції протоколів комунікації. Дані протоколи були спільними для всієї мережі.

Розвиток глобальних комп'ютерних мереж значною мірою визначався розвитком телефонних мереж. Починаючи з 70-х років XX ст. почали з'являтися високошвидкісні цифрові канали зв'язку, що з'єднували автоматичні телефонні станції (АТС) та технології, що дозволяли одночасно передавати багато розмов.

У тих же 70-х роках виникла потреба у створенні локальних комп'ютерних мережах разом із появою міні-ЕОМ, що були побудовані на основі великих інтегральних схем (ВІС). Локальні мережі LAN (Local Area Network) – це об'єднання робочих станцій, що розташовані на невеликій території, зазвичай, у радіусі не більше 1–2 км. Хоча на сьогодні локальна мережа може мати і великі розміри, наприклад, кілька десятків кілометрів.

З появою LAN для з'єднання комп'ютерів використовувалися нестандартизовані мережні технології. Мережна технологія – це погоджений набір програмних і апаратних засобів, а також механізмів передавання даних через лінії зв'язку, що є достатнім для побудови комп'ютерних мереж і передавання даних через них. Застосування нестандартних технологій суттєво ускладнювало процес об'єднання в мережу комп'ютерів різних виробників.

У середині 80-х років XX ст. з'явилися стандартні технології об'єднання комп'ютерів у мережу – Ethernet, Arcnet (Attached Resource Computer Network), Token Ring, Token Bus, FDDI (Fiber Distributed Data Interface). Всі технології базувались на принципі комутації пакетів.

Наступним поштовхом до розвитку LAN стала поява персональних комп'ютерів, що поступово з'являлись у користуванні звичайних людей в їх домівках. Наявність стандартних мережних технологій перетворила процес побудови локальної мережі з вирішення технічної проблеми у рутинну роботу.

Кінець 90-х років минулого століття виявив явного лідера серед технологій локальних мереж – сімейство Ethernet, в яке увійшли: класична технологія Ethernet зі швидкістю передавання 10 Мбіт/с, а також Fast Ethernet зі швидкістю 100 Мбіт/с і Gigabit Ethernet зі швидкістю 1000 Мбіт/с. На теперішній час вже стандартизовано 10 Gigabit, 40 Gigabit і навіть 100 Gigabit Ethernet, які орієнтовані на реалізацію не стільки в LAN, скільки у WAN і MAN (Metropolitan Area Network). Хронологічну послідовність важливих подій на шляху розвитку перших комп'ютерних мереж та мережі Інтернет наведено у табл. 1.1.

Таблиця 1.1 – Важливі події в історії комп'ютерних мереж

Етап	Час
1. Перше повідомлення передається від EOM SDS Sigma 7 Каліфорнійського університету в Лос-Анджелес на EOM SDS 940 Стенфордського дослідницького інституту.	29 жовтня 1969 р.
2. Запрацювала мережа ALOHAnet – перша в світі мережа пакетного передавання даних через радіоканал, розроблена Норманом Абрамсоном, співробітником Гавайського університету.	1970 р.
3. Поява першої програми керування електронною поштою, створеної Ларрі Робертсоном. Рей Томлінсон запропонував використовувати символ @ для позначення адресата.	1972 р.
4. Стандартизація технології X.25.	1974 р.
5. Опубліковано формальний опис протоколів TCP і IP.	1981 р.
6. Поява служби доменних імен (DNS).	1984 р.
7. Поява першої програми для обміну повідомленнями (IRC), розробленої Яркко Ойкариненом.	1988 р.
8. З'явився стандарт Web, запропонований Тімом Бернерсом-Лі і Робертом Кайо.	1991 р.
9. Марк Андреєссен створив перший веб-браузер Mosaic.	1993 р.

Історично головною метою об'єднання комп'ютерів у мережу було розподілення ресурсів. Користувачі комп'ютерів, що були під'єднані до мережі, або програми, що виконувалися на даних комп'ютерах, отримували доступ до різних ресурсів інших комп'ютерів мережі. До основних ресурсів комп'ютера, які можна поділяти з іншими користувачами або програмами традиційно відносять:

- периферійні пристрої: принтери, плотери, сканери, диски;
- дані;
- обчислювальні потужності: процесорний час, оперативна пам'ять тощо.

При проектуванні або розгортанні комп'ютерної мережі важливим є питання, до якої категорії ця мережа належить. Це буде безпосередньо впливати на вибір обладнання, протоколів передачі даних, визначення вимог до кваліфікації персоналу, що буде обслуговувати мережу. Класифікація комп'ютерних мереж може бути виконана за різними ознаками, зокрема за територіальною поширеністю, способом організації взаємодії комп'ютерів, топологією, доступністю тощо.

За **територіальною поширеністю** мережі поділяють на два великих класи: локальні і глобальні. Крім того останнім часом з'явились додаткові класи та підкласи, до яких відносять регіональні, персональні та мережі, що з'єднують сховища даних.

Локальна комп'ютерна мережа LAN (Local Area Network) об'єднує абонентські системи, що розташовані в межах обмеженої території. Характерною рисою такої мережі є її належність тій організації, де вона розташована. В більшості випадків обслуговування та налаштування такої мережі також виконується фахівцями цієї установи. Швидкість передавання даних в таких мережах, як правило, становить від 100 Мбіт/с до 1 Гбіта/с. До LAN відносять мережі підприємств, офісів банків, закладів освіти тощо.

Глобальні комп'ютерні мережі WAN сьогодні застосовуються для об'єднання локальних мереж, що знаходяться на значній відстані. Наприклад, компанія має філіали в різних містах або країнах і має потребу забезпечити обмін даними між ними. Очевидно, що такий сервіс може надати установа, що має розгалужену мережу каналів передавання даних і називається провайдером телекомунікаційних послуг. Споживачі WAN мереж, як правило, мають оплачувати цей сервіс, що висуває специфічні вимоги до протоколів WAN мереж, зокрема підтримка автентифікації користувачів, широкий діапазон пропускних спроможностей, можливість забезпечення якості сервісу QoS тощо.

Регіональні мережі MAN (Metropolitan Area Network) займають проміжне місце між LAN і WAN й об'єднують комп'ютери та мережі, що розташовані в межах певного регіону, міста, адміністративного району. Як правило, будуються на основі швидкісних магістралей, прокладених в межах великих міст.

Персональні мережі PAN (Personal Area Network) об'єднують пристрої, що знаходяться поряд. Типовим прикладом такої мережі є Bluetooth – мережа, що призначена для об'єднання побутових пристроїв, підключення клавіатури, гарнітури тощо.

Мережа зберігання даних SAN (Storage Area Network) являє собою архітектурне рішення для підключення таких зовнішніх пристроїв зберігання даних, як дискові масиви, оптичні приводи до серверів таким чином, щоб операційна система розпізнала підключені ресурси як локальні. Це дає можливість кільком серверам звертатись до спільного дискового простору, що стає особливо актуальним при застосуванні технологій віртуалізації та кластеризації.

За **організацію взаємодії** між комп'ютерами мережі поділяють на однорангові (per to per або P2P) та ієрархічні мережі, які ще називають мережами з виділеним сервером.

Однорангова мережа складається виключно з робочих станцій, кожна з яких має особисто вирішувати питання автентифікації користувачів, керування доступом до власних ресурсів, встановлення та підтримки програмного забезпечення тощо. В одноранговій мережі всі робочі станції рів-

ноправні. Робоча станція може надавати доступ до таких своїх ресурсів, як дисковий простір та периферійне обладнання. Перевагою P2P мереж є простота розгортання і налаштування, однак при великій кількості робочих станцій процес адміністрування такої мережі суттєво ускладняється, а захищеність і надійність знижуються. Місце застосування однорангових мереж – домашні мережі та мережі малих офісів.

Застосування **клієнт-серверної архітектури** передбачає наявність виділеного комп'ютера або комп'ютерів, що виконують специфічні функції в мережі й надають послуги іншим комп'ютерам. Ці пристрої називають серверами (Server – той, хто надає послуги). Однією з найбільш важливих є функція керування доступом до мережі та її ресурсів. Таку задачу вирішують, наприклад, Active Directory сервер від Microsoft, ZENworks від Micro Focus та інші. Крім того в корпоративній мережі актуальними є функції файлового, поштового сервера, сервера друку. Слід звернути увагу, що клієнт-серверна архітектура є основною архітектурою мережі Інтернет, оскільки основні ресурси цієї мережі зберігаються на веб-серверах.

Об'єднуючи у мережу кілька робочих станцій (більше двох), потрібно вирішити, яку вибрати конфігурацію фізичних зв'язків або **топологію**. Під топологією мережі розуміється конфігурація графу, вершини якого відповідають кінцевим вузлам мережі (наприклад, робочі станції), та комутаційного обладнання (наприклад, маршрутизатори), а ребра – фізичні або інформаційні зв'язки між вершинами. Виділяють такі основні топології.

1. **Повнозв'язна топологія** (рис. 1.1, а) відповідає мережі, в якій кожна робоча станція безпосередньо з'єднана з усіма іншими. Даний варіант громіздкий та неефективний, оскільки необхідна велика кількість комутаційний портів, фізичних ліній зв'язку. Повнозв'язна топологія у великих мережах використовується дуже рідко, оскільки для зв'язку n вузлів необхідно $n(n\text{-}1)/2$ фізичних дуплексних ліній зв'язку.

2. **Комірчата топологія**, інша назва – частково зв'язна, отримується з повнозв'язної шляхом вилучення деяких зв'язків (рис. 1.1, б). Застосовується у WAN мережах, вузлами, як правило, є магістральні маршрутизатори, а ребрами – виділені фізичні або віртуальні канали.

3. У мережах з **кільцевою топологією** (рис. 1.1, в) дані передаються по кільцю від одного комп'ютера до іншого. Перевагами даної топології є резервні зв'язки (доступність вузла двома шляхами), можливість організації зворотного зв'язку. Однак в даному випадку потрібно вживати спеціальних заходів, коли один вузол відключено від мережі. Використовувалась в LAN Token Ring та FDDI. Є одним із варіантів реалізації SAN мережі.

4. **Зіркоподібна топологія** (рис. 1.1, г) утворюється у випадку, коли кожний комп'ютер підключається безпосередньо до загального багатовходового центрального пристрою. Даний пристрій виконує перенаправлення потоків інформації комп'ютерам мережі. Є основою побудови сучасних LAN, в яких функцію центрального пристрою виконують комутатори або точки доступу.

5. З’єднання між собою кількох LAN, побудованих за зіркоподібною топологією, породжує топологію **ієрархічної зірки або дерева** (рис. 1.1, д). Ця топологія сьогодні найбільш поширена як в локальних, так і в глобальних мережах.

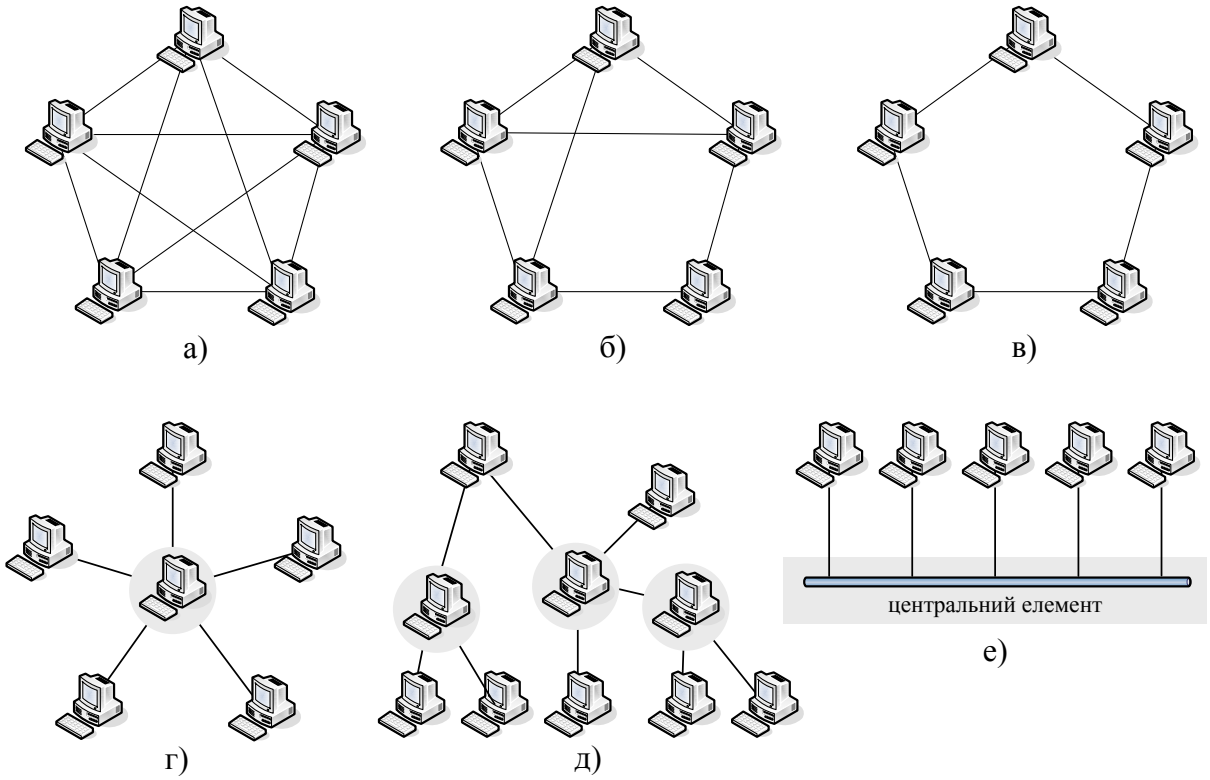


Рисунок 1.1 — Топології комп’ютерних мереж
а) повнозв’язна, б) комірчата, в) кільцева, г) зіркоподібна,
д) деревоподібна, е) загальна шина

6. Частковим випадком зірки є **загальна шина** (рис. 1.1, е). У таких мережах центральним елементом є пасивний кабель або концентратор (hub), до якого підключається кілька комп’ютерів. Таку топологію мають безпроводові мережі, де як загальна шина використовується радіосередовище. Інформація, що передається, доступна відразу всім вузлам мережі, що під’єднані до загальної шини. Основна перевага даної топології – низька вартість і простота під’єднання нових вузлів. Головним недоліком є низька продуктивність, оскільки в певний момент часу передачу може виконувати лише один комп’ютер.

7. Невеликі мережі мають типову топологію – зірка, кільце, загальна шина. Разом з тим, у великих мережах наявні довільні зв’язки між комп’ютерами. У таких мережах можна виділити частини, що мають типову топологію. Тому такі мережі мають **змішану** топологію (рис. 1.2).

За **доступністю** виділяють нижчевказані різновиди мереж.

Загальнодоступна (Internet) – це мережа, що не потребує процесу реєстрації та відкрита для всіх користувачів.

Корпоративна (Intranet) – закрита мережа певної компанії, для доступу до якої користувач обов'язково має мати обліковий запис, в якому фіксуються його права доступу до ресурсів мережі. Процес доступу до корпоративної мережі передбачає проходження процедури автентифікації, як правило за допомогою паролю.

Комбінована (Extranet) – це захищена від несанкціонованого доступу корпоративна мережа, що використовує Internet-технології для внутрішньокорпоративних цілей, а також для надання частини корпоративної інформації та корпоративних ресурсів діловим партнерам компанії.

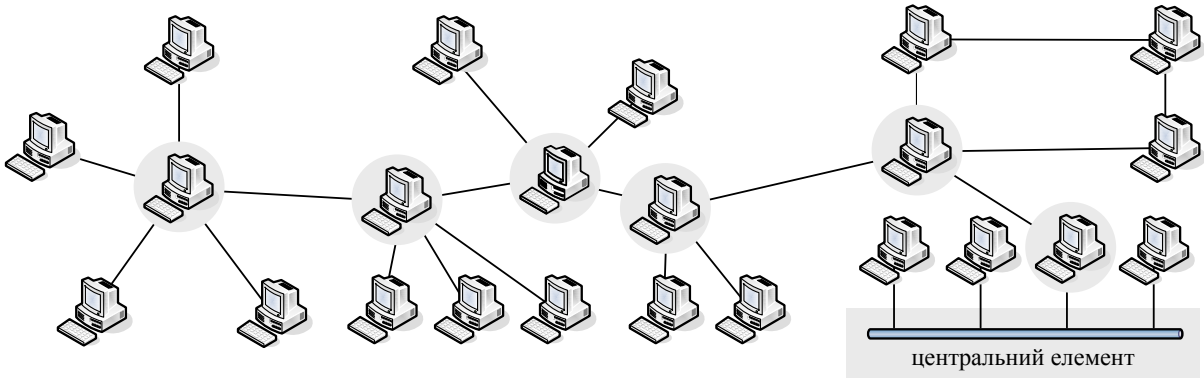


Рисунок 1.2 — Мережа зі змішаною топологією

1.2 Основні компоненти комп'ютерних мереж та їх призначення

Призначення комп'ютерної мережі зводиться до забезпечення комунікації або зв'язку між абонентами. Будь-яка комунікація передбачає наявність трьох обов'язкових компонентів:

- передавача (джерела інформації);
- отримувача;
- середовища, через яке буде проводитися передача даних.

Виділяють апаратні і програмні компоненти комп'ютерних мереж. До **апаратних компонент** відносять пристрої та середовище передавання даних. Всі пристрої, які використовуються в комп'ютерних мережах, поділяють на дві категорії: кінцеві та проміжні (інфраструктурні).

До кінцевих пристроїв відносять «споживачів» мережі, тобто ті пристрої, що є або відправниками, або одержувачами даних. Це робочі станції, сервери, принтери, веб-камери, IP-телефони, смартфони тощо. Протягом тривалого часу основними кінцевими пристроями мережі Інтернет були сервери та клієнтські робочі станції. Однак з переходом у фазу Інтернету речей (Internet of Things) і далі у фазу всеосяжного Інтернету (Internet of Everything) масштаби підключення та різноманіття кінцевих пристроїв значно розширились за рахунок побутових пристроїв, транспортних засобів, найрізноманітніших давачів тощо.

Серед проміжних пристроїв можна виділити:

- **пристрої доступу до мережі**, що призначені для під'єднання кінцевих пристроїв до локальної комп'ютерної мережі: комутатор (switch), концентратор (hub), точка доступу (access point);
- **пристрої для з'єднання між собою локальних мереж**: маршрутизатор (router);
- **комунікаційні сервери і модеми** для обслуговування глобальних мереж;
- **пристрої безпеки** (security appliance), найбільш відомим з яких є брандмауер (firewall).

Основними середовищами передавання даних сучасних мереж є кабельні та безпроводові. Кабельні середовища реалізуються у вигляді оптичних ліній зв'язку і з'єднань на основі мідного дроту. Оптичні канали, як правило, застосовуються для побудови швидкісних магістралей, в той же час мідний кабель є основою побудови локальних мереж. Останнім часом з розповсюдженням мобільних пристроїв значне поширення отримали безпроводові канали, головне призначення яких – підключення до мережі кінцевих користувачів.

До **програмних компонентів** відносять сервіси і процеси, що працюють на мережних пристроях і, залежно від призначення, виконують різні функції. Як правило, програмні компоненти визначаються відповідним протоколом. Таким чином, програмні компоненти можна поділити на:

- мережні операційні системи;
- протокольні стеки;
- мережні застосування (мережні програми).

1.3 Адресація вузлів у мережі

При об'єднані декількох комп'ютерів у мережу виникає потреба у їх ідентифікації, а саме: адресації їх мережних інтерфейсів. Кожен комп'ютер має мати унікальний ідентифікатор в мережі. В сучасних мережах використовуються різні адресні схеми, що мають те чи інше призначення. Загалом всі схеми адресації можна поділити на цифрові та алфавітно-цифрові. У свою чергу, як алфавітно-цифрові, так і цифрові схеми адресації можна поділити на однорівневі та ієрархічні (рис. 1.3). Множину адрес, що є доступною в межах певної схеми адресації, називають адресним простором.

Прикладом **алфавітно-цифрової однорівневої адресації** є NetBIOS (Network Basic Input/Output System) імена, що використовуються для ідентифікації пристроїв локальної мережі в мережному оточенні. Вони призначені для спрощення доступу користувачів LAN до ресурсів інших робочих станцій, зокрема дискових накопичувачів та різноманітного периферійного обладнання. Як правило, NetBIOS імена несуть змістовне навантаження і вказують на розташування пристрою або його призначення.

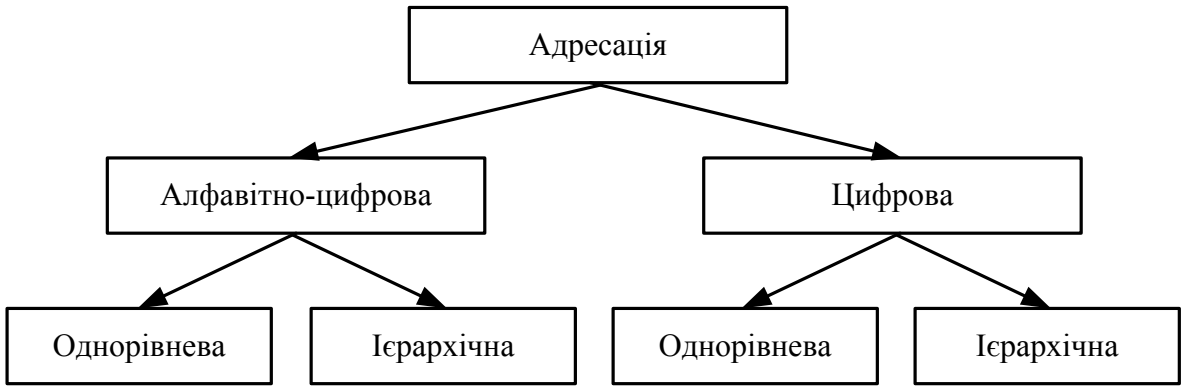


Рисунок 1.3 – Типи адресації вузлів у комп'ютерних мережах

Алфавітно-цифрова ієрархічна адреса – це доменні символьні імена, що мають ієрархічну структуру, наприклад, ftp-serv1.rada.gov.ua. Ця адреса вказує, що комп'ютер виконує функцію ftp-сервера у мережі Верховної Ради (rada). Дана мережа належить до загальнодержавної мережі (gov) України (ua). Така адресація зручна для людей, оскільки має зрозумілу структуру й її нескладно запам'ятати. Наведена у прикладі адреса вузла має чотири рівні: ідентифікатор країни, де знаходиться певний тип установи, далі ідентифікується сама установа і вузол.

Прикладом **цифрової однорівневої адресації** може бути MAC-адреса. Це унікальна адреса мережного адаптера для його ідентифікації в локальних мережах. Така адреса записується у двійковому або шістнадцятковому вигляді, наприклад, 48-5D-60-66-D4-4D. MAC-адреси задаються самими виробниками для своїх пристроїв. Як правило, вони є незмінними, тому коли у комп'ютера змінюється мережний інтерфейс, то, відповідно, змінюється і мережна адреса.

Простір MAC-адрес визначається документами міжнародної асоціації Інституту інженерів електротехніки та електроніки IEEE, а саме: MAC-48, EUI-48, EUI-64. Найбільш поширені адреси MAC-48, які використовуються в технологіях Ethernet, Token Ring, FDDI, WIMAX (Worldwide Interoperability for Microwave Access) тощо. У даному випадку адреса складається із 48 бітів (6 байтів). Таким чином, весь адресний простір нараховує 2^{48} (або 281 474 976 710 656) адрес. MAC-48 і EUI-48 відрізняються лише за призначенням: MAC-48 адреси використовуються для мережних пристроїв, а EUI-48 – для інших типів апаратного та програмного забезпечення. При цьому ідентифікатори EUI-64 складаються з 64 бітів і використовуються у FireWire та для формування хостової частини IP-адреси кінцевого пристрою в IPv6.

Прикладом **цифрової ієрархічної адреси** є IP-адреса, яка складається з двох частин: адреси мережі і адреси комп'ютера. Оскільки кордон між цими частинами не є фіксованим, разом з IP-адресою використовується спеціальний покажчик, який називається маскою підмережі або довжиною префіксу. Наприклад, запис 192.168.1.1/24 говорить про те, що 192.168.1 –

це адреса мережі, а .1 – адреса комп'ютера в цій мережі. На відміну від MAC-адрес IP-адреси не прив'язані жорстко до пристрою і можуть змінюватись при переміщенні комп'ютера з однієї мережи в іншу. Саме тому вони часто називаються логічними адресами.

Користувачам зручно працювати з алфавітно-цифровими адресами, в той же час комп'ютери працюють з цифровими адресами. На практиці, як правило, використовується відразу кілька схем адресації, і один вузол мережі може відразу мати кілька адрес-імен.

Розглянемо приклад, наведений на рис. 1.4. Клієнт зі свого ноутбука хоче дістатись сервера, що має доменне ім'я vntu.edu.ua. Для цього він має спочатку передати свій запит на бездротовий маршрутизатор з адресою 10.0.0.254. Однак оскільки мережний адаптер бездротового маршрутизатора «розуміє» тільки MAC-адреси, а саме: клієнтська станція спочатку дізнається про MAC-адресу маршрутизатора 00-1c-10-9a-4e-0a і тільки після цього починається передавання даних. Наступною задачею є визначення IP-адреси сервера з ім'ям vntu.edu.ua. Впоравшись з цим клієнт може безпосередньо встановити зв'язок з сервером.

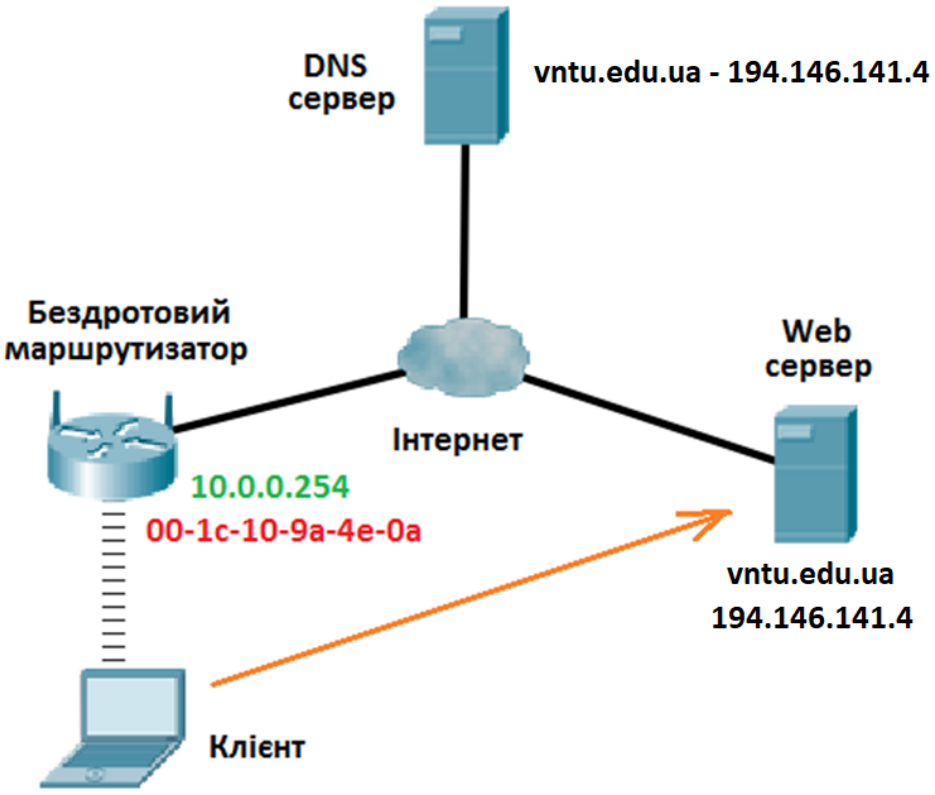


Рисунок 1.4 – Спільне використання різних адресних схем

Для передавання даних через мережу обладнання використовує цифрові адреси, тому виникає задача перетворення алфавітно-цифрової адреси на цифрову. Для перетворення імен з одного типу в інший використовуються спеціальні протоколи, що називаються протоколами перетворення імен.

Проблема встановлення відповідності між адресами різних типів може вирішуватися централізовано або розподілено. При **централізованому підході** у мережі виділяється один або кілька комп'ютерів (сервери імен), що зберігають таблиці відповідності адрес різних типів. Інші комп'ютери мережі звертаються до даного з відповідними записами, щоб перетворити алфавітно-цифрову адресу у цифрову. Коли використовується **розподілений підхід**, то на кожному комп'ютері зберігаються всі його адреси різних типів. Перевагою розподіленого підходу є те, що він дозволяє відмовитися від виділеного сервера, який необхідно адмініструвати. Недоліком даної реалізації є те, що виникає необхідність використовувати широкомовні повідомлення, що є неінформативним надлишковим трафіком у мережі. Тому розподілений підхід використовується у локальних невеликих мережах, а централізований – у великих.

Коли комп'ютеру потрібно встановити відповідність ієрархічно-цифрової адреси і однорівневої цифрової, він відправляє у мережу широкомовний запит. Всі комп'ютери мережі порівнюють адресу у запиті з власною. Той комп'ютер, що знайшов відповідність, надсилає відповідь, де міститься відповідна однорівнева цифрова апаратна адреса. Така схема використана у протоколі визначення адрес ARP (Address Resolution Protocol) стека TCP/IP. Саме таким чином клієнтська робоча станція на рис. 1.4 дізналась MAC-адресу бездротового маршрутизатора.

Для перетворення алфавітно-цифрових ієрархічних адрес (доменних імен) у цифрові ієрархічні (IP-адреси) використовується протокол DNS (Domain Name System), що базується на централізованому підході. На рис. 1.4 саме за допомогою DNS-сервера клієнтська робоча станція дізналась про IP-адресу сервера з доменним іменем vntu.edu.ua.

Перетворення алфавітно-цифрових однорівневих адрес (NetBIOS-імена) у цифрові ієрархічні виконують WinS-сервери (Windows Internet Name Service) з використанням централізованого підходу.

Для адресації одержувачів також можуть бути застосовані різні типи адрес:

- **персональні адреси** (unicast) використовуються для ідентифікації певних вузлів, наприклад MAC-адреса конкретного мережного адаптера або IP-адреса конкретного інтерфейсу;
- **групові адреси** (multicast) використовуються для ідентифікації одразу кількох вузлів. Дані, що спрямовуються на групову адресу, надсилаються відразу на всі вузли, що входять до складу групи. Групова адреса не заміняє персональну і використовується паралельно з нею. Пристрій може мати кілька групових адрес;
- **широкомовні адреси** (broadcast) використовуються для відправлення даних на всі вузли мережі. Частина мережі, в межах якої розповсюджується широкомовний трафік, називається широкомовним доменом;

- **адреса довільного розсилання**, або одного з групи (anycast) задає групу адрес як multicast, але дані, що відправлені на цю адресу, доставляються не всім комп'ютерам групи, а тільки одному з них. Як правило, це комп'ютер, що розташований ближче всіх до відправника.

1.4 Способи комутації

З'єднання кінцевих пристроїв через мережу проміжних вузлів називають комутацією. Послідовність вузлів, що знаходяться на шляху від передавача до приймача, утворюють **маршрут**. Для успішної комунікації необхідно узгодити правила комунікації, зокрема:

- процедуру встановлення з'єднання;
- мову або спосіб кодування даних в процесі комутації;
- порядок передавання та прийому даних у випадку, коли одночасно це робити неможливо.

Методи комутації в комп'ютерних мережах поділяють на **комутацію каналів** та з **проміжним зберіганням**, відповідну класифікацію показано на рис. 1.5. Комутація каналів може бути тимчасовою і тривалою. Комутація з проміжним збереженням може виконуватися як комутація повідомлень або пакетів. У свою чергу, комутація пакетів може виконуватися через комутацію дейтаграм або віртуальних каналів.

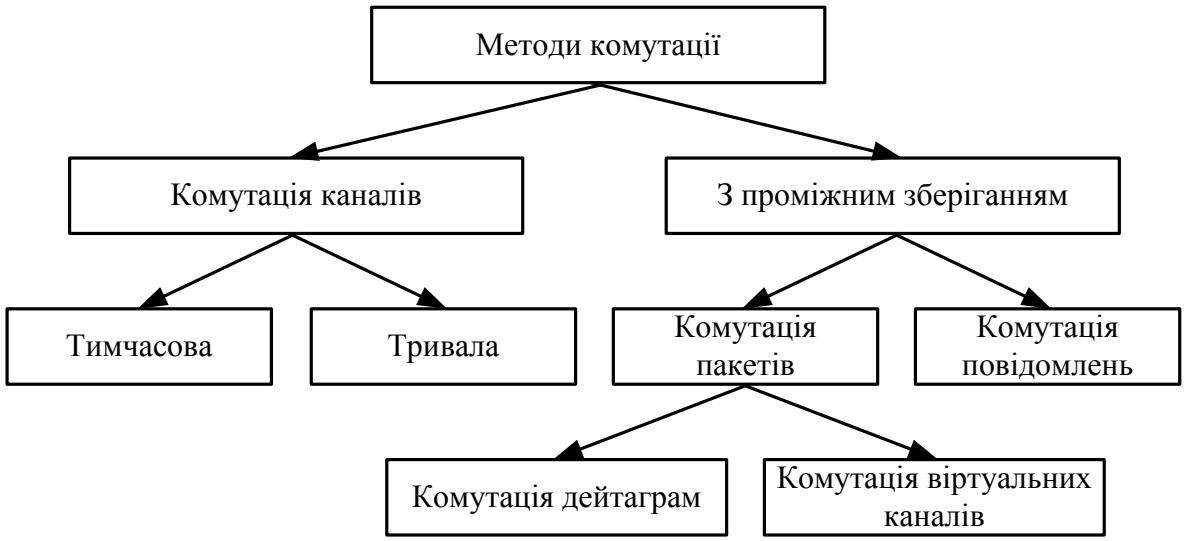


Рисунок 1.5 – Класифікація методів комутації у комп'ютерних мережах

Метод комутації каналів історично з'явився першим з появою телефонних мереж. При застосуванні методу **комутації каналів** перш ніж почати передавання даних потрібно встановити фізичне з'єднання між станцією відправника і станцією отримувача. Процедуру встановлення з'єднання показано на рис. 1.6. Комутація може бути тимчасовою – виклю-

чно на час передавання даних, такі канали називаються комутованими. Якщо канал надається у користування на тривалий термін, то він називається виділеним.

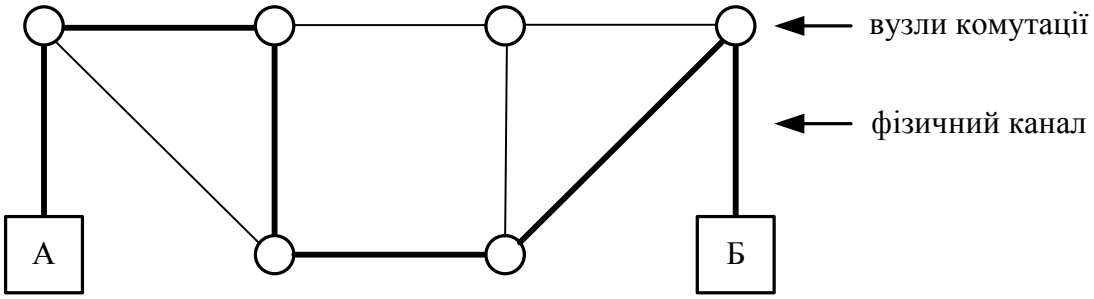


Рисунок 1.6 – Встановлення з’єднання при використанні методу комутації каналів

Крім того, при застосуванні методу комутації каналів виникає задача розподілу на підканали фізичного середовища з великою пропускною спроможністю, тобто, організація передавання кількох незалежних потоків даних через один фізичний канал. Ця задача вирішується методами мультиплексування каналів.

Головний недолік методів комутації каналів полягає в тому, що для встановлення з’єднання та організувати передавання даних обов’язково потрібна наявність вільних підканалів на усіх фізичних ланках від відправника до одержувача. Іншим недоліком, що є специфічним при передаванні комп’ютерного трафіку, є неможливість динамічного розширення смуги пропускання підканалу, навіть якщо в даний момент через інші підканали дані не передаються. Це призводить до неефективного використання фізичного каналу. Головною ж перевагою методу комутації каналів є мінімальна часова затримку при передаванні даних.

Перевагою **методів з проміжним зберіганням** є те, що відпадає необхідність в утворенні фізичного каналу від відправника до отримувача, достатньо наявності вільного каналу тільки до сусіднього вузла комутації. Крім того, при фрагментації повідомлення існує можливість розподілу одного потоку даних по кількох маршрутах, що дозволяє балансувати навантаження. Також ці методи знімають обмеження щодо перерозподілу вільної смуги пропускання каналу.

Методи з проміжним зберіганням поділяються на **комутацію пакетів** і **комутацію повідомлень**.

Якщо повідомлення або файл передаються як єдине ціле без фрагментації – це комутація повідомлення. Перевагою цього методу є мінімізація службової інформації та відсутність необхідності фрагментації і дефрагментації повідомлення. З іншого боку, при передаванні великих повідомлень цей метод характеризується такими недоліками:

- відбувається монопольне захоплення кожної проміжної ланки на час передавання всього повідомлення;
- оскільки кожне повідомлення має повністю завантажитись на проміжний вузол комутації, останній потребує великої буферної пам'яті;
- якщо в процесі передавання відбулась втрата або спотворення частини повідомлення, його потрібно повторно передавати повністю;
- необхідність повної буферизації повідомлення на вузлах комутації призводить до суттєвих затримок передавання.

Внаслідок вищесказаного метод комутації повідомлень застосовується тільки для коротких повідомлень, наприклад, при передаванні службового трафіку.

Основним методом комутації при передаванні комп'ютерного трафіку є **метод комутації пакетів**. У даному випадку відбувається поділ повідомлення на фрагменти, що далі окремо передаються через мережу. До кожного фрагмента додається відповідна службова інформація (адреси, номер фрагмента тощо). Комутацію пакетів може бути реалізовано у вигляді комутації дейтаграм і комутації віртуальних каналів.

Комутація дейтаграм – це метод, за яким в поле службової інформації кожного пакета додаються повні адреси отримувача і відправника, тому пакет (дейтаграма) є незалежною одиницею даних і передається самостійно. Перевагою цього методу є незалежність кожної одиниці даних, наслідком чого є можливість передавання пакетів різними маршрутами та динамічна зміна маршруту при недоступності окремих ланок. Недоліками є завеликий розмір адресної частини та необхідність впорядковування отриманих фрагментів.

При використанні **методу комутації віртуальних каналів** перш ніж починається передавання даних, утворюється так званий віртуальний канал між відправником і отримувачем. Ідентифікатор віртуального каналу в подальшому використовується для адресації. Перевагою є економія пропускної спроможності фізичного каналу за рахунок зменшення службової частини та зменшення розміру таблиці комутації.

Для того, щоб оцінити затримку передавання даних в сучасних комп'ютерних мережах, виконаємо розрахунок затримки передавання даних у мережі з комутацією каналів і пакетів (рис. 1.7 і 1.8 відповідно). Очевидно, що більш ефективною, з точки зору часових затримок, буде мережа з комутацією каналів, де резервується канал на час передавання даних. Дані будуть надходити до адресата без затримок. Однак значну частину часу зарезервований канал буде простоювати під час пауз між передаваннями. Якщо використовувати комутацію пакетів, то швидкість передавання буде меншою, але завантаженість фізичного каналу буде більш рівномірною.

Так, час передавання даних від вузла $B1$ до вузла $B2$ через мережу з комутацією каналів $t_{КК}$ складається з часу розповсюдження сигналу $t_{РС}$ та часу передавання повідомлення $t_{ПП}$

$$t_{КК} = t_{РС} + t_{ПП}.$$

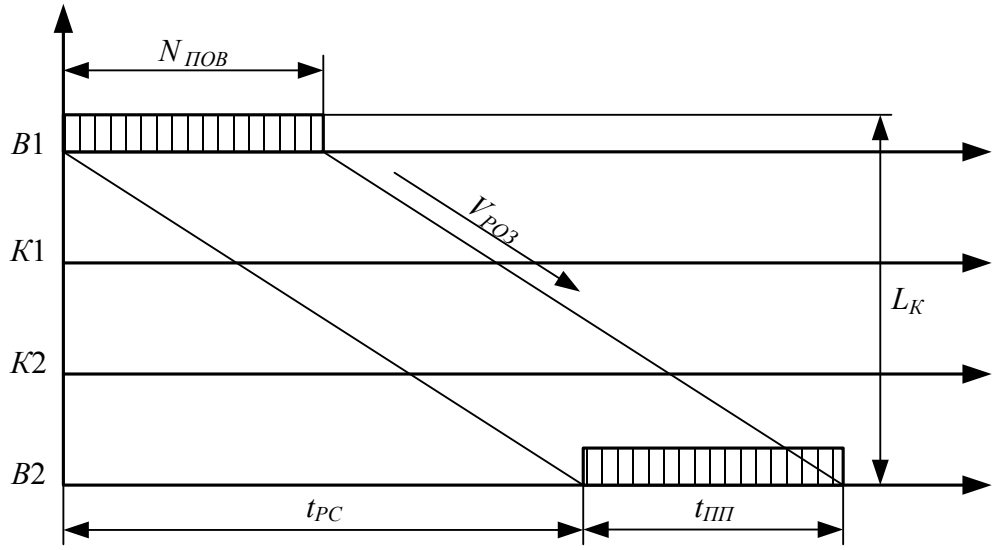


Рисунок 1.7 – Часова діаграма передавання у мережі з комутацією каналів

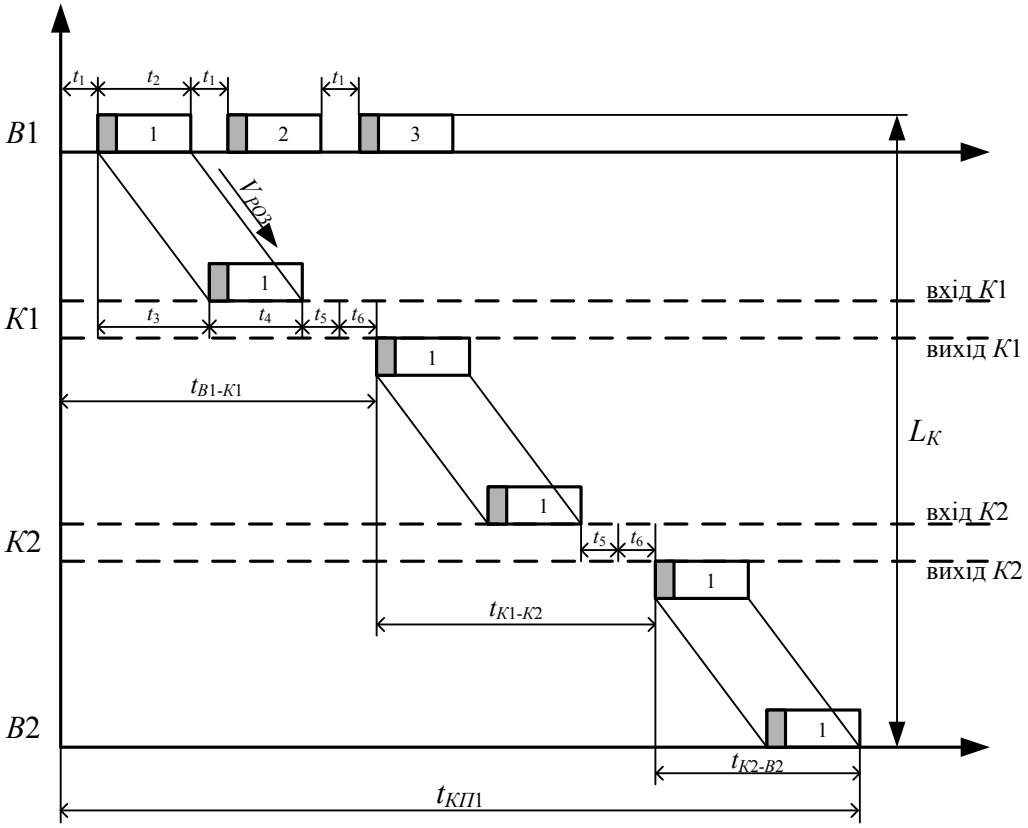


Рисунок 1.8 – Часова діаграма передавання повідомлення, що розділене на пакети, у мережі з комутацією пакетів

Час розповсюдження сигналу визначається як відношення довжини каналу L_K до швидкості розповсюдження сигналу $V_{РОЗ}$

$$t_{РС} = \frac{L_K}{V_{РОЗ}}.$$

Час передавання повідомлення $t_{ПП}$ можна розрахувати за такою формулою

$$t_{ПП} = \frac{N_{ПОВ}}{P_{ЛЗ}},$$

де $N_{ПОВ}$ – розмір повідомлення;

$P_{ЛЗ}$ – пропускна спроможність лінії зв'язку, біт/с.

При передаванні повідомлення через мережу з комутацією пакетів одне повідомлення розміром $N_{ПОВ}$ розділено на пакети, кожен з яких має заголовок. Пакети передаються від передавача до приймача, між якими є комутатори, кожен з яких вносить деяку затримку комутації. Спочатку визначимо час $t_{КП1}$ передавання одного пакета повідомлення від вузла $B1$ до $B2$, між якими є два комутатори $K1$, $K2$. Час передавання пакета з $B1$ до $K1$ складається з таких частин:

1. Час, що витрачається $B1$, містить такі складові:

t_1 – час формування пакета або час пакетування;

t_2 – час передавання в канал пакета (заголовка і поля даних);

2. Час t_3 поширення сигналу через канал зв'язку від $B1$ до $K1$;

3. Додатковий час витрачається на першому комутаторі:

t_4 – час прийняття пакета з його заголовком з каналу у вхідний буфер комутатора, що дорівнює часу передавання пакета t_2;

t_5 – час очікування пакетом своєї черги (завчасно невідомий, залежить від завантаженості мережі);

t_6 – час комутації пакета, тобто передавання пакета на вихідний порт. Цей час є фіксованим, складає кілька мікросекунд, залежить від конкретної моделі комутатора.

Таким чином, загальний час t_{B1-K1} передавання пакета з вузла $B1$ на вихідний інтерфейс комутатора $K1$ складає

$$t_{B1-K1} = t_1 + t_2 + t_3 + t_5 + t_6.$$

Час, що витрачається на двох інших ділянки шляху, позначимо через t_{K1-K2} і t_{K2-B2}. Ці величини містять такі ж складові, що й t_{B1-K1}, але в них

не входить час пакетування t_1, і t_{K2-B2} не містить час комутації t_6. Загальний час передавання одного пакета дорівнює

$$t_{КП1} = t_{B1-K1} + t_{K1-K2} + t_{K2-B2}.$$

Сумарний час передавання всього повідомлення не дорівнюватиме сумі значень часу передавання кожного пакета, оскільки пакети опрацьовуються в кілька етапів і всі пристрої мережі виконують це паралельно. Тому час передавання одного повідомлення буде значно меншим, ніж сума значень часу передавання кожного пакета повідомлення. Визначити загальний час важко, оскільки неможливо визначити значення часу очікування пакетів у черзі комутації. Якщо вважати, що пакети в черзі знаходяться приблизно однаковий час, загальний час передавання повідомлення можна оцінити за формулою

$$t_{КП\Sigma} = t_{КП1} + (n-1)(t_1 + t_2),$$

де – кількість вузлів комутації.

1.5 Моделі опису комп'ютерних мереж

Будь-яка комунікація між людьми, або в межах комп'ютерної мережі є неможливою без узгодження певних правил, які називаються **протоколами**. Оскільки правил або протоколів, що визначають функціонування мережі, потрібно багато, то їх необхідно систематизувати. Для цього використовуються так звані **ієрархічні моделі**, що описують ту чи іншу групу протоколів. Кожне правило або протокол асоціюється з тим чи іншим рівнем ієрархічної моделі. Кожен рівень ієрархічної моделі визначає набір правил, що близько пов'язані з об'єктом і стандартизують процес мережної взаємодії.

Розглянемо приклад можливої ієрархічної моделі правил для спілкування людей (рис. 1.9). Модель складається з трьох рівнів ієрархії. Функція нижнього рівня – узгодження типу середовища, через яке буде відбуватись спілкування, наприклад мобільний телефон, електронна пошта тощо. Наступний рівень визначає правила подання інформації, зокрема вибір мови, механізми підтвердження, сигнали завершення тощо. Верхній рівень використовується для визначення теми розмови.

В ієрархічних моделях правила або протоколи нижніх рівнів надають послуги протоколам або правилам верхніх рівнів. Наприклад, для того, щоб узгодити мову спілкування, попередньо потрібно визначитись з каналом спілкування. Набір протоколів різних рівнів, що разом повністю забезпечують процес комунікації, називають **протокольним стеком**.

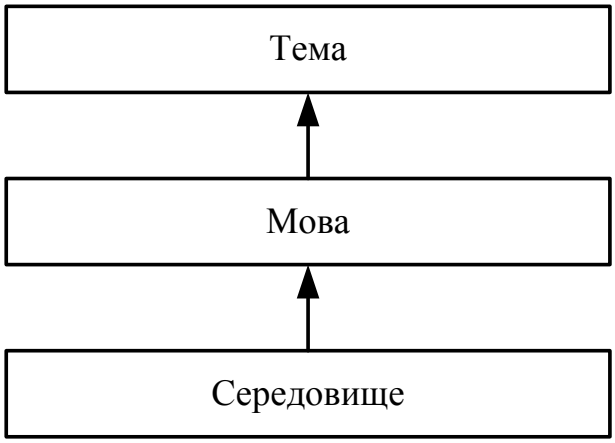


Рисунок 1.9 – Модель опису правил спілкування людей

Виділяють такі моделі опису протоколів:

- Довідкові, наприклад, модель взаємодії відкритих систем OSI (Open Systems Interconnection), що розроблена міжнародною організацією зі стандартизації ISO;
- Протокольні моделі, що близькі до того чи іншого протокольного стека. Прикладом тут є модель TCP/IP, що відповідає протокольному стеку TCP/IP. Додаткову інформацію про міжнародні організації, які займаються стандартизацією комп'ютерних мереж, наведено у додатку А.

1.5.1 Модель OSI. Необхідність у її створенні виникла, коли з'явилась потреба підтримки багатьох програмних й апаратних засобів і забезпечення їх взаємодії. Під відкритою системою розуміють мережне обладнання (сервери, робочі станції, комутаційне обладнання), що реалізовано з використанням вимог щодо елементів відкритих систем.

Відкрита система складається з кількох ієрархічних складових, що виконують специфічні для свого рівня функції. За кожним рівнем моделі OSI закріплені певні функції. Взаємодія компонентів в межах однієї відкритої системи може відбуватися тільки за умови, що ці компоненти розташовані на сусідніх рівнях. На рис. 1.10 показано структуру моделі взаємодії відкритих систем та мережні пристрої, що реалізують функції певних рівнів.

Слід зазначити, що система може містити тільки частину рівнів, обумовлених моделлю взаємодії відкритих систем, наприклад, міст або маршрутизатор. Правило обміну інформацією між сусідніми рівнями однієї системи називається інтерфейсом. Правило взаємодії однакових рівнів різних технічних систем називається протоколом.

Модель OSI містить сім рівнів: фізичний, канальний, мережний, транспортний, сеансовий, представницький та рівень застосувань.

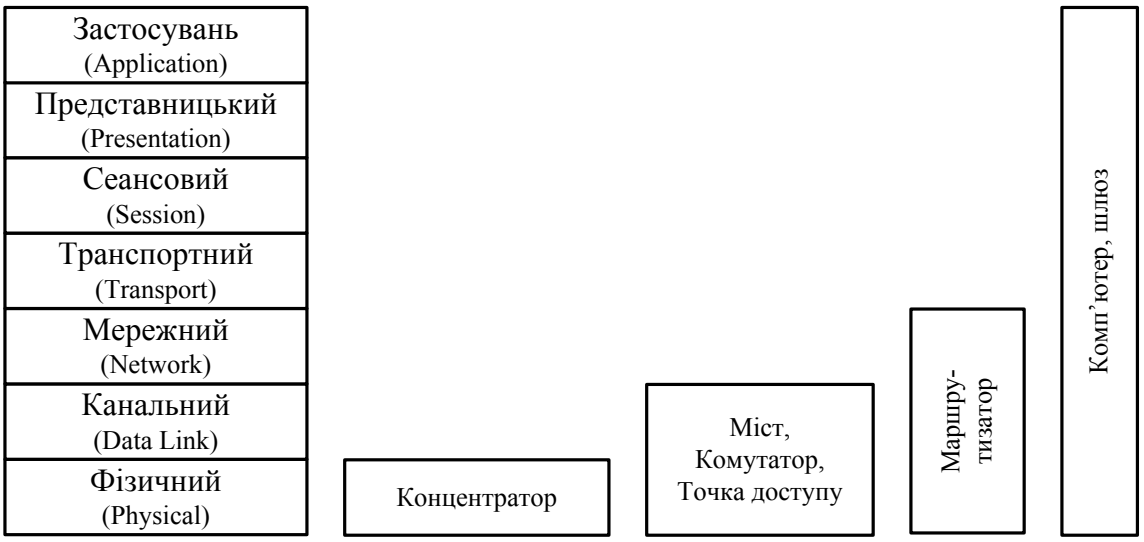


Рисунок 1.10 – Структура моделі взаємодії відкритих систем OSI та популярні мережні пристрої

Фізичний рівень стандартизує:

- тип та параметри середовища передавання даних, головними параметрами якого є смуга пропускання, коефіцієнт згасання тощо;
- спосіб подання інформації (спосіб кодування даних) при передаванні через певне середовище;
- спосіб під'єднання середовища передавання даних до пристрою, стандартизація конекторів.

Канальний рівень поділяється на два підрівні. Верхній – підрівень керування логічним каналом LLC (Logical Link Control) і нижній – підрівень керування доступом до середовища передавання даних MAC (Media Access Control). Канальний рівень має забезпечити передавання між пристроями в межах локальної мережі. На даному рівні формуються блоки даних канального рівня, що називаються кадрами або фреймами і є основними одиницями передавання даних. Кадр – це закінчена змістовна послідовність байтів, яка має початок і кінець.

До функцій LLC підрівня відносять:

- формування логічного каналу з метою забезпечення гарантованого доставляння в межах локальної мережі;
- забезпечення взаємодії з верхнім мережним рівнем шляхом додавання службового поля в якому вказується тип протоколу мережного рівня, як правило, IPv4 або IPv6.

До функцій MAC підрівня відносять:

- перевірку доступності середовища, розташування та вилучення кадру з середовища передавання;
- визначення меж кадру;

- апаратну (цифрову однорівневу) адресацію, яка необхідна у випадку, коли кадр можуть отримати відразу кілька адресатів (у локальних мережах даний тип адресації використовується завжди);
- контроль спотворень при передаванні кадру шляхом розрахунку та перевірки контрольних сум.

Функцій канального і фізичного рівня цілком вистачає для передавання даних в межах локальній мережі. Канальний рівень реалізовано апаратно-програмно за допомогою мережного адаптера та його драйвера.

Мережний рівень використовується для створення єдиної транспортної системи, що об'єднує кілька мереж і визначає спосіб ієрархічної адресації пристроїв в об'єднаній мережі та механізми вибору оптимального маршруту доставляння даних. Пристрої, що забезпечують передавання даних між мережами, називаються маршрутизаторами (router). Блоки даних мережного рівня називаються пакетами, які інкапсулюються в кадри. Мережний рівень у більшості випадків не гарантує надійного доставляння інформації.

Транспортний рівень має прийняти дані від сеансового рівня, розділити їх, за необхідності, на частини, передати мережному рівню і гарантувати, що всі частини у правильному вигляді буде доставлено в мережі. Розмір частин залежить від використовуваних протоколів.

Модель OSI визначає п'ять класів транспортного сервісу: від найнижчого 0 – до найвищого 4. Ці види сервісів відрізняються якістю надаваних послуг: швидкістю, можливістю відновлення втраченого з'єднання, здатністю виявляти і виправляти помилки передачі (спотворення, втрату, дублювання даних).

Сеансовий рівень забезпечує керування діалогом, фіксує, яка сторона є активною; відповідає за встановлення логічного з'єднання через глобальну мережу, активізує або створює порти процесів і підтримує їх в активному стані на період сеансу.

Представницький рівень визначає спосіб подання інформації, узгоджує стандарти кодування різних типів даних, виконує перетворення кодів, шифрування інформації (SSL-протокол).

Рівень застосувань забезпечує взаємодію мережі та користувача. Це набір різних протоколів, за допомогою яких користувачі отримують доступ до ресурсів, організовують свою спільну роботу. Одиницю даних, якою оперує прикладний рівень, зазвичай називають повідомленням.

Інтерфейсом між рівнями, з точки зору блоків даних, є наявність в заголовку нижнього рівня інформації про те, який протокол наступного рівня (вищого) міститься в даному блоці даних. Заголовок даних – це, фактично, та інформація, яку додає певний рівень моделі OSI. Процедура вкладання інформації в блок даних нижнього рівня називається інкапсуляцією, а, відповідно, зворотна процедура – декапсуляцією. Узагальнена назва для фрагментів даних різних рівнів моделі OSI визначається як PDU (Protocol Data Unit).

В мережних пристроях може бути реалізована лише частина рівнів моделі взаємодії відкритих систем (див. рис. 1.10), зокрема:

- концентратор (hub) працює виключно на фізичному рівні;
- міст (bridge), комутатор (switch) та точка доступу (access point) реалізують фізичний та канальний рівні;
- маршрутизатор (router) реалізує рівні з фізичного до мережного;
- шлюз (gateway) – це програмний або програмно-апаратний засіб, який дозволяє з'єднати між собою мережі, що працюють з різними протокольними стеками. Реалізує всі сім рівнів моделі OSI.

1.5.2 Структура моделі TCP/IP. Крім моделі OSI на практиці використовується модель TCP/IP. На рис. 1.11 показано взаємозв'язок моделей OSI та TCP/IP. Дана модель складається з чотирьох рівнів, кожен з яких виконує нижчевказані функції:

- **Рівень доступу до мережі** (Network Access) – це нижній рівень моделі, містить протоколи для фізичного доставляння даних у мережі;
- **Міжмережний рівень** (Internet) містить протоколи, що забезпечують передавання даних між окремими мережами – регламентує функції міжмережної маршрутизації та ієрархічної адресації;
- **Транспортний рівень** відповідає за поділ потоків даних на окремі фрагменти та надання необхідного транспортного сервісу для доставляння даних: надійного, з гарантією доставляння, але повільного; або без гарантій, але швидко;
- **Рівень застосувань** містить протоколи, що виконують опрацювання даних користувачів, керування обміном даними між застосуваннями. На даному рівні також стандартизується форма подання даних.

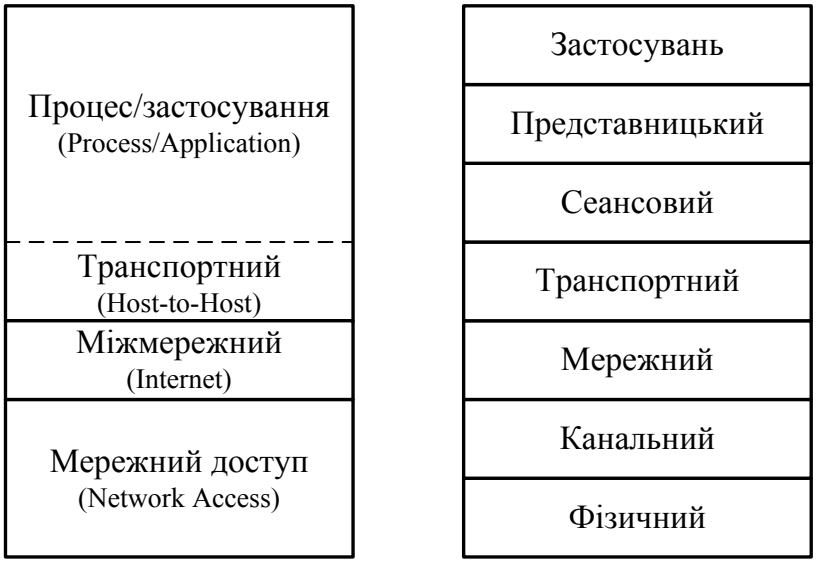


Рисунок 1.11 – Структура моделі TCP/IP, її зв'язок із моделлю OSI

1.6 Питання для самоперевірки

1. Визначте основні етапи розвитку комп'ютерних мереж.
2. Охарактеризуйте різні типи комп'ютерних мереж.
3. Наведіть приклад, коли доцільно використовувати однорангову архітектуру мережі.
4. Які основні топології комп'ютерних мереж ви знаєте? Дайте короткий аналіз кожній з них.
5. Охарактеризуйте повнозв'язну топологію.
6. Які переваги має зіркоподібна топологія комп'ютерних мереж?
7. Охарактеризуйте властивості топології «загальна шина».
8. Назвіть і охарактеризуйте основні компоненти мережі.
9. Назвіть та охарактеризуйте основні типи проміжних пристроїв.
10. Визначте методи адресації комп'ютерів у мережах, їх призначення та взаємозв'язок.
11. Порівняйте алфавітно-цифрові однорівневі та ієрархічні адреси між собою.
12. Проведіть аналіз цифрових однорівневих адрес.
13. Охарактеризуйте цифрові ієрархічні адреси.
14. Вкажіть методи комутації, що використовуються в комп'ютерних мережах, і наведіть їх порівняльну характеристику.
15. Охарактеризуйте основні джерела затримки передавання в мережі з комутацією каналів.
16. Охарактеризуйте основні джерела затримки передавання в мережі з комутацією пакетів.
17. Розрахуйте затримку передавання даних в мережі з комутацією каналів за такими умовами: відстань між відправником і одержувачем 1000 км, швидкість розповсюдження сигналу 200 тис. км/с, обсяг даних, що передаються, 1 Гбайт, пропускна спроможність каналу 100 Мбіт/с.
18. Визначте основні поняття моделі взаємодії відкритих систем (OSI).
19. Визначте загальну структуру моделі взаємодії відкритих систем.
20. Яке призначення фізичного та канального рівнів моделі OSI?
21. Визначте властивості мережного рівня моделі OSI.
22. Яке призначення транспортного рівня OSI?
23. Порівняйте сучасні моделі опису комп'ютерних мереж (моделі OSI та TCP/IP).

2 ФІЗИЧНИЙ РІВЕНЬ СУЧАСНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

2.1 Основні принципи передавання на фізичному рівні

Фізичний рівень призначено безпосередньо для передавання потоку даних і реалізації інтерфейсу між хост-модулем та середовищем передавання. Функції фізичного рівня обов'язково реалізуються апаратно за допомогою мережного адаптера або послідовного порту на всіх модулях, що підключаються до мережі. Тобто, визначаються інтерфейсні схеми підключення до каналу передавання даних, а саме: механічні, електричні, функціональні та процедурні параметри й особливості даного з'єднання.

Основні функції фізичного рівня полягають у: формуванні послідовності сигналів, її передаванні по фізичних каналах, синхронізації та виконанні процедур модуляції або кодування (залежно від типу фізичного каналу). Крім того, фізичний рівень визначає процедури передавання сигналів у канал та отримання їх з каналу, спосіб подання та вид сигналів, що передаються в канал у тому вигляді, який використовується в даному фізичному середовищі (фізичному каналі).

Прикладами мережних інтерфейсів, які реалізують функції фізичного рівня, є інтерфейси RS-232, RS-422A, RS-449, RS-485, V.35, V.90 тощо.

Будь-який канал зв'язку визначається сукупністю взаємопов'язаних характеристик, основними з яких є:

- амплітудно-частотна характеристика,
- ширина смуги пропускання (bandwidth),
- завадостійкість,
- згасання (attenuation),
- пропускна спроможність (throughput),
- перехресні наведення на ближньому кінці лінії NEXT (Near End Cross Talk) тощо.

Ширина смуги пропускання каналу характеризує неперервний діапазон частот, для якого згасання (відношення амплітуди вихідного сигналу до вхідного) не перевищує деякого наперед заданого значення (зазвичай, 0,5). Тобто, смуга пропускання визначає діапазон частот сигналу, в якому сигнал передається в каналі зв'язку без значних спотворень.

Пропускна спроможність лінії зв'язку характеризує максимально можливу швидкість, з якою можуть передаватися дані. Цей параметр, з одного боку, залежить від параметрів фізичного середовища, а з іншого – визначається способом передавання даних і їх подання.

Зв'язок між пропускною спроможністю лінії та її смугою пропускання для каналу без шуму можна визначити за формулою Г. Найквіста

$$C = 2F \log_2 M ,$$

де C – максимальна швидкість передавання даних (біт/с);

F – ширина смуги пропускання лінії (Гц);

M – кількість станів сигналу.

З цієї формули випливає, що якщо передається двійковий сигнал (сигнал з двома різними станами), то пропускна спроможність дорівнює подвоєному значенню ширини смуги пропускання лінії зв'язку. Якщо ж станція-передавач використовує більше двох станів для кодування даних, то пропускна спроможність лінії підвищується.

У цій формулі в явному вигляді не враховується шум у каналі, але опосередковано його вплив відображається при виборі кількості станів інформаційного сигналу.

Клод Шеннон розширив цю формулу на канали з випадковим (термодинамічним) шумом

$$C = F \log_2 \left(1 + \frac{P_c}{P_ш} \right),$$

де C – максимальна пропускна спроможність лінії (біт/с);

F – ширина смуги пропускання (Гц);

P_c, $P_ш$ – потужності сигналу та шуму, відповідно.

Зазвичай на практиці відношення $P_c/P_ш$ не використовується. Замість нього використовують значення $10\lg\frac{S}{N}$, де S – потужність корисного сигналу, а N – потужність шуму. Така одиниця називається децибелом (дБ). Наприклад, якщо відношення сигнал/шум має значення 10, то ця величина дорівнює 10 дБ; якщо це відношення 100, то – 20 дБ тощо.

Наприклад, канал зі смугою пропускання в 3 кГц та відношенням сигнал/шум у 30 дБ (звичайні параметри для аналогової телефонної системи) ніколи не зможе передавати сигнали з пропускною спроможністю, більшою за 30 Кбіт/с незалежно від кількості станів сигналу.

Треба зазначити, що наведені формули визначають верхню, теоретично можливу межу пропускної спроможності інформаційного каналу, що, зазвичай, є недосяжним у реальних системах.

2.2 Класифікація та характеристика каналів передавання даних

Типова організація каналу передавання даних наведена на рис. 2.1. Робоча станція, комп'ютер чи будь-який інший модуль, який підключається до мережі, має узагальнену назву – **обладнання обробки даних** (ООД), згідно з міжнародною термінологією: **DTE** – Data Terminal Equipment. Це термінальне обладнання підключається до каналу зв'язку за допомогою **апаратури передавання (каналу) даних** (АКД), яка, відповідно до міжнародної термінології, має назву **DCE** – Data Communication Equipment. Основною функцією АКД є підключення ООД до лінії або каналу передаван-

ня даних. Апаратура передавання даних підключається до термінального модуля за допомогою інтерфейсів RS-232, RS-422A, RS-449, RS-485 тощо, які є інтерфейсами, що не залежать від середовища передавання. До каналу зв'язку комунікаційне обладнання підключається за допомогою інтерфейсів, що залежать від середовища передавання, наприклад, V.27, V.35, V.90 тощо. Загальна схема організації каналу передавання даних наведена на рис. 2.1, а приклад підключення комунікаційного обладнання з різним набором функцій – на рис. 2.2.

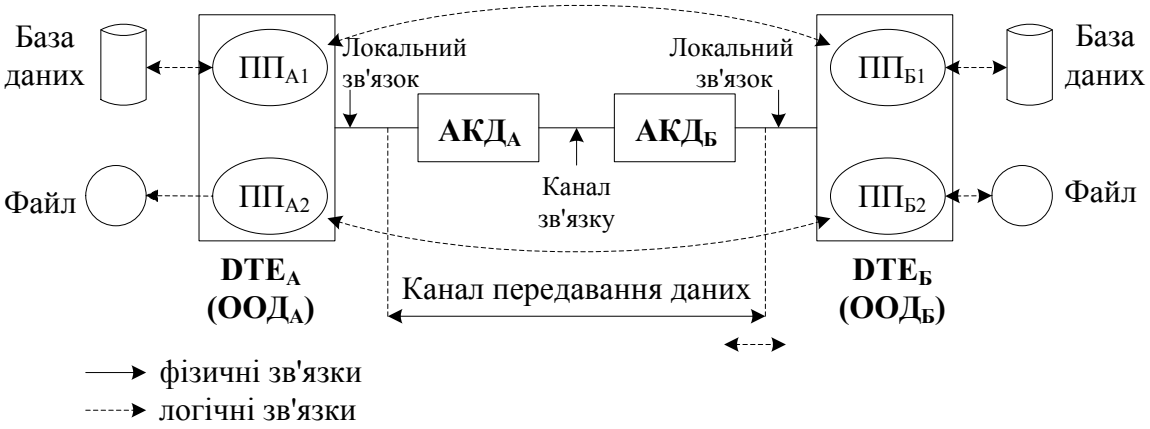


Рисунок 2.1 – Структура каналу передавання даних

Стандарти та рекомендації інтерфейсів DTE-DCE визначають:

- загальні характеристики (швидкість та послідовність передавання);
- функціональні та процедурні характеристики (номенклатура, категорія, виводи інтерфейсу, правила їх взаємодії);
- електронні характеристики (величина напруги, струму та опору);
- механічні характеристики (габарити, розподілення контактів тощо).

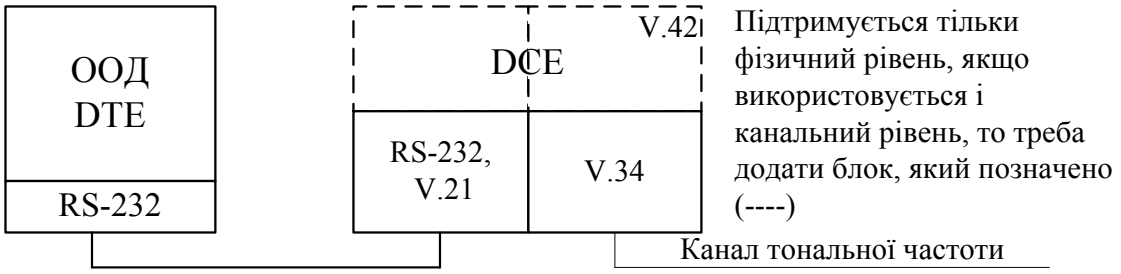


Рисунок 2.2 – Схема підключення DTE та DCE

Єдиної класифікації каналів зв'язку не існує. Але можна виділити деякі ознаки (параметри), за якими і виконується класифікація. До таких характеристик відносять:

- діапазон частот;
- способи передавання даних у каналах зв'язку;

- спосіб організації передавання;
- спосіб створення каналу;
- фізичне середовище каналу зв'язку, яке використовується для передавання;
- способи захисту даних у каналі;
- кількість інформаційних потоків, які можуть одночасно передаватися в каналі (спосіб мультиплексування потоків) тощо.

Класифікація каналів **за діапазоном робочих частот**, що є найбільш суттєвою при використанні електричних сигналів для передавання даних, оскільки визначає пропускну спроможність каналу, наведена в табл. 2.1.

Таблиця 2.1 – Класифікація каналів за діапазоном частот

Найменування хвиль	Діапазон хвиль	Найменування частот	Діапазон частот
Декакілометрові (наддовгі)	10–100 км	наднизькі	3–30 кГц
Кілометрові (довгі)	1–10 км	низькі	30–300 кГц
Гектометрові (середні)	100–1000 м	середні	300–3000 кГц
Декаметрові (короткі)	10–100 м	високі	3–30 МГц
Метрові (ультракороткі)	1–10 м	дуже високі	30–300 МГц
Дециметрові	10–100 см	ультрависокі	300–3000 МГц
Сантиметрові	1–10 см	надвисокі	3–30 ГГц
Міліметрові	1–10 мм	надзвичайно високі	30–300 ГГц
Децимініметрові	0,1–1 мм	гіпервисокі	300–3000 ГГц

У сучасних симетричних кабельних лініях зв'язку використовуються сигнали з частотою не більше декількох сотень кілогерц. Коаксіальні кабелі дозволяють передавати сигнали з частотою до сотень мегагерц. При передачі сигналів у радіоканалах використовують частоти від $3*10^3$ до $3*10^{12}$ Гц. В оптоволоконних каналах використовуються частоти порядку $3*10^{14}$ Гц.

За **направленістю ліній зв'язку** виділяють:

- направлені:
  - — коаксіальні кабелі;
  - — скручені пари;
  - — оптоволоконні;

- ненаправлені (радіолінії):
  — прямого бачення;
  — радіорелейні (ретрансляція в дециметровому і більш короткому діапазонах частот);
  — космічні;
  — іоносферні;
  — тропосферні.

За **способом передавання інформації** в каналі зв'язку розрізняють асинхронні та синхронні канали.

Щоб дані передавача адекватно сприймались приймачем, потрібно на прийомній стороні вміти, по-перше, відділяти біт від біта (навіть якщо передається декілька однакових бітів підряд), й, по-друге, вміти відділити байт від байта, тобто провести межі між байтами чи символами іншої розрядності у неперервному потоці бітів, що надходять у приймач.

Перша проблема легко вирішується через стандартизацію швидкостей передачі. Так, у різних міжнародних стандартах можна знайти такі швидкості передавання у бітах за секунду: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 14400, 16800, 19200, 28800, 38400, 57600, 115200…

Для вирішення другої проблеми використовують один із двох способів передавання інформації:

- асинхронний;
- синхронний.

При асинхронному способі (рис. 2.3) інформація передається в канал посимвольно, починаючи з молодших бітів. У символі може міститися від 5 до 8 інформаційних бітів *D*0-*DN*. Ще один біт *DP* може додаватися, щоб доповнювати інформаційні біти до парної (або непарної) кількості одиниць. Це дозволяє виявляти помилки непарної кратності, що сталися під час передачі.

Якщо передавати нічого, передавач видає сигнал одиничного рівня. Передавання кожного символа починається сигналом нульового рівня *Старт* і закінчується сигналом одиничного рівня *Стоп*.

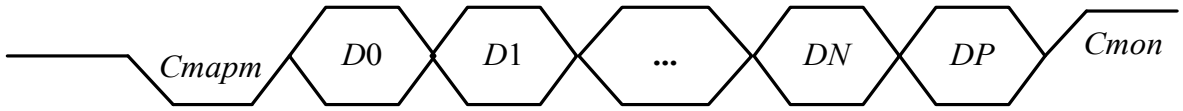


Рисунок 2.3 – Асинхронне передавання

Сигнал *Стоп* може тривати як завгодно довго, але його мінімальна тривалість (залежно від прийнятого стандарту) дорівнює тривалості передавання будь-якого інформаційного біта D_i, помноженій на 1; 1,5 або 2.

Отже, при асинхронному способі передавання будь-який символ «обрамляється» нульовим стартовим і одиничним стоповим бітом. З надхо-

дженням стартового біта розпочинається відлік часу, котрий дозволяє розпізнати решту бітів, не висуваючи великих вимог до однаковості відліку часу в передавачі та приймачі.

Перевагами асинхронного способу є його простота та невелика вартість, а також те, що він забезпечує контроль правильності передавання кожного символа, що дозволяє швидко реагувати на помилки.

Однак передача стартових і стопових бітів займає значний відсоток часу, тому використання асинхронного способу передавання характерне, в основному, для низькошвидкісних каналів зв'язку з достатньо високим рівнем шуму.

При синхронному способі передавання стартові та стопові біти відсутні. Замість них використовуються так звані синхросимволи – неперіодичні послідовності нулів і одиниць довжиною один або два байти.

Як тільки у вхідному потоці бітів приймач виявляє синхросимвол, від цього моменту розпочинається відлік бітів не одного, а цілої послідовності байтів (чи символів іншої розрядності), що утворюють **кадр**, структура якого в загальному вигляді показана на рис. 2.4.

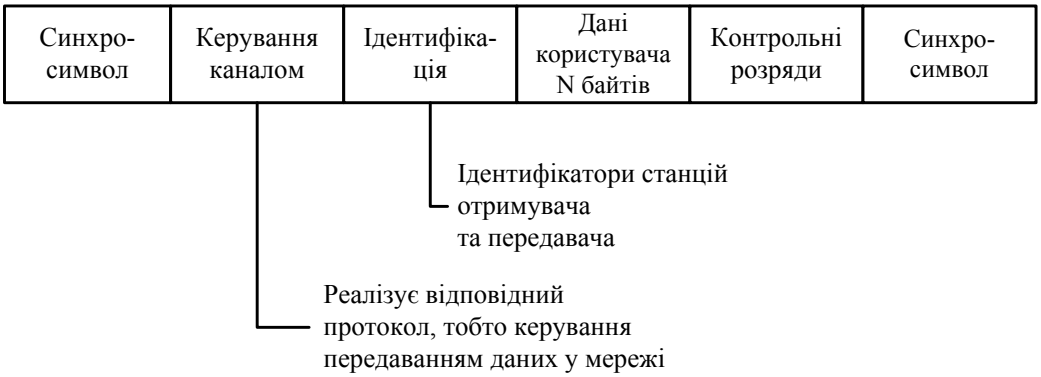


Рисунок 2.4 – Синхронне передавання

Якщо інформація для передавання відсутня, станція-відправник передає синхросимволи, щоб підтримувати синхронну роботу передавача та приймача.

Перевагами синхронного передавання є менша, порівняно з асинхронним, надлишковість, отже, вища швидкість передавання інформації та ефективність використання каналів. До недоліків можна віднести складнішу та дорожчу апаратуру передавання даних, вищу ймовірність виникнення помилок синхронізації, а також довшу реакцію на помилкові кадри, оскільки помилки можуть бути виявлені тільки після отримання всього кадру.

За **типом проміжної апаратури** всі канали зв'язку поділяють на аналогові та цифрові. В аналогових лініях така апаратура виконує підсилення аналогових, тобто неперервних, сигналів. Прикладами аналогових каналів є лінії зв'язку в телефонних мережах загального призначення, канали тональної частоти. У цифрових каналах передаються дискретні, імпульсні

дженням стартового біта розпочинається відлік часу, котрий дозволяє розпізнати решту бітів, не висуваючи великих вимог до однаковості відліку часу в передавачі та приймачі.

Перевагами асинхронного способу є його простота та невелика вартість, а також те, що він забезпечує контроль правильності передавання кожного символа, що дозволяє швидко реагувати на помилки.

Однак передача стартових і стопових бітів займає значний відсоток часу, тому використання асинхронного способу передавання характерне, в основному, для низькошвидкісних каналів зв'язку з достатньо високим рівнем шуму.

При синхронному способі передавання стартові та стопові біти відсутні. Замість них використовуються так звані синхросимволи – неперіодичні послідовності нулів і одиниць довжиною один або два байти.

Як тільки у вхідному потоці бітів приймач виявляє синхросимвол, від цього моменту розпочинається відлік бітів не одного, а цілої послідовності байтів (чи символів іншої розрядності), що утворюють **кадр**, структура якого в загальному вигляді показана на рис. 2.4.

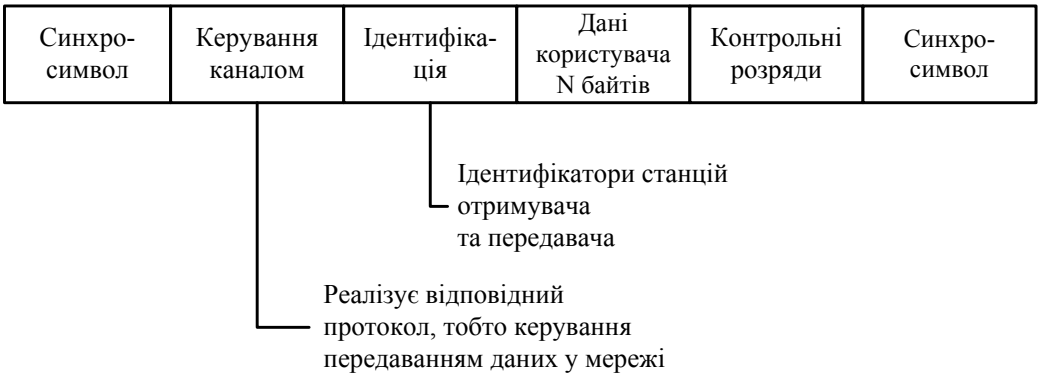


Рисунок 2.4 – Синхронне передавання

Якщо інформація для передавання відсутня, станція-відправник передає синхросимволи, щоб підтримувати синхронну роботу передавача та приймача.

Перевагами синхронного передавання є менша, порівняно з асинхронним, надлишковість, отже, вища швидкість передавання інформації та ефективність використання каналів. До недоліків можна віднести складнішу та дорожчу апаратуру передавання даних, вищу ймовірність виникнення помилок синхронізації, а також довшу реакцію на помилкові кадри, оскільки помилки можуть бути виявлені тільки після отримання всього кадру.

За **типом проміжної апаратури** всі канали зв'язку поділяють на аналогові та цифрові. В аналогових лініях така апаратура виконує підсилення аналогових, тобто неперервних, сигналів. Прикладами аналогових каналів є лінії зв'язку в телефонних мережах загального призначення, канали тональної частоти. У цифрових каналах передаються дискретні, імпульсні

сигнали, які мають два чи більше станів. За допомогою таких сигналів передаються як комп'ютерні дані, так і аудіо- та відеотрафік у оцифрованому вигляді. В цифрових каналах як проміжна апаратура для відновлення форми імпульсів і підсилення дискретних сигналів використовуються регенератори. Прикладами цифрових каналів є канали типу T1/E1, T3/E3, канали мережі з інтегрованим доступом ISDN (Integrated Services Digital Network), які будуть детальніше розглянуті в розділі 8.

За **способом організації передавання** розрізняють симплексні, дуплексні та напівдуплексні канали.

Симплексні канали дозволяють передавати інформацію тільки в одному напрямку, тому в комп'ютерних мережах майже не використовуються, але знайшли застосування в телебаченні та комерційному радіомовленні.

Дуплексні канали забезпечують одночасне передавання в двох напрямках, не викликаючи небажаних зупинок та очікування, які характерні для напівдуплексного каналу. Вони використовується в мережних застосуваннях, які вимагають високої продуктивності обміну та швидкої реакції на передавання інформації.

Напівдуплексні канали передавання також дозволяють передавати інформацію в двох напрямках, але послідовно у часі. Після передавання даних в одному напрямку потрібно виконати переналагоджування інтерфейсів взаємодійних станцій і після цього реалізувати передавання в протилежному напрямку. Такі канали забезпечують меншу швидкість передавання, ніж дуплексні, але мають значно меншу вартість. Напівдуплексні канали широко використовуються в комп'ютерних мережах, для підключення терміналів будь-якого типу, а також у багатьох системах, які функціонують на основі запиту/відповіді. В телефонії напівдуплексні канали реалізуються на основі двопроводової схеми підключення, а дуплексні – на основі чотирипроводової організації. Зазвичай телефонні компанії розглядають двопроводову лінію як комутований канал з автонабором, а чотирипроводову – як орендований некомутований канал.

За **способом створення каналу** розрізняють такі канали зв'язку:

- комутовані;
- некомутовані, які, в свою чергу, також можуть бути:
  - орендованими;
  - виділеними (фізичними).

Комутований канал зв'язку – це канал, що надається комутаційною мережею на тимчасові сеанси та створюється безпосередньо перед передаванням даних з окремих каналів зв'язку (підключення Dial-Up). Після закінчення передавання канал ліквідується. При поновленні сеансу зв'язку між тими ж взаємодійними модулями комутований канал може бути створений вже з інших каналів зв'язку. Швидкість передавання в таких каналах невисока, а довжина та характеристики комутованих каналів можуть часто змінюватися в широкому діапазоні.

Некомутовані (орендовані) канали зв'язку комутуються заздалегідь, існують постійно між взаємодійними станціями і виділяються даним користувачам на тривалий час, протягом якого ці канали не можуть бути зайняті іншими абонентами. Вони мають відносно постійні характеристики й забезпечують більшу пропускну спроможність.

Некомутовані (виділені) канали зв'язку утворені парою проводів і скомутовані постійно (тобто створено фізичний зв'язок між абонентськими станціями). Передавання даних може відбуватися в будь-який момент без необхідності створення чи активізації необхідного каналу. Зазвичай реалізуються за чотирипроводовою схемою.

Переваги та недоліки комутованих і некомутованих каналів наведено в табл. 2.2.

Таблиця 2.2 – Характеристики комутованих і некомутованих каналів

Комутовані канали	Некомутовані канали
Переваги: • гнучкість; • простота реалізації; • невелика вартість обладнання; • невелика вартість підключення.	**Переваги:** • забезпечення високої швидкості передавання даних; • підтримка великих обсягів трафіку; • забезпечення зв'язку вищої якості; • відсутність блокування запиту на з'єднання; • швидка реакція віддаленої станції.
Недоліки: • низька якість; • достатньо великий час очікування відповіді від віддаленої станції; • невисока швидкість передавання даних; • можливість блокування запиту (формування сигналу «зайнято» при значній завантаженості мережі); • велика вартість при значному обсязі трафіку.	**Недоліки:** • неможливість з'єднання та передавання даних при пошкодженнях каналу; • відсутність гнучкості, особливо при пошкодженнях лінії; • велика вартість передавання, особливо при використанні виділених каналів.

За **кількістю інформаційних потоків**, які можуть одночасно передаватися в каналі, виділяють одноканальні та багатоканальні, в яких викори-

стовується будь-який спосіб ущільнення (мультиплексування) інформаційних потоків.

За **способами захисту** канали передавання розрізняють:

- відкриті канали;
- закриті (засекречені) канали, в яких використовується відповідний метод (методи) захисту (шифрування).

За **структурними особливостями організації** каналу розрізняють конфігурації:

- точка-точка (point-to-point);
- точка-багато точок (point-to-multipoint).

При конфігурації **point-to-point** (рис. 2.5, а) у каналі існує тільки дві станції, будь-яка з яких може передавати дані, а інша їх отримувати. При цьому не виникає потреби ідентифікації взаємодійних станцій.

При конфігурації **point-to-multipoint** (рис. 2.5, б) до каналу підключено більше двох станцій, тому при будь-якому передаванні інформації необхідна ідентифікація станцій, які взаємодіють у даному сеансі зв'язку.

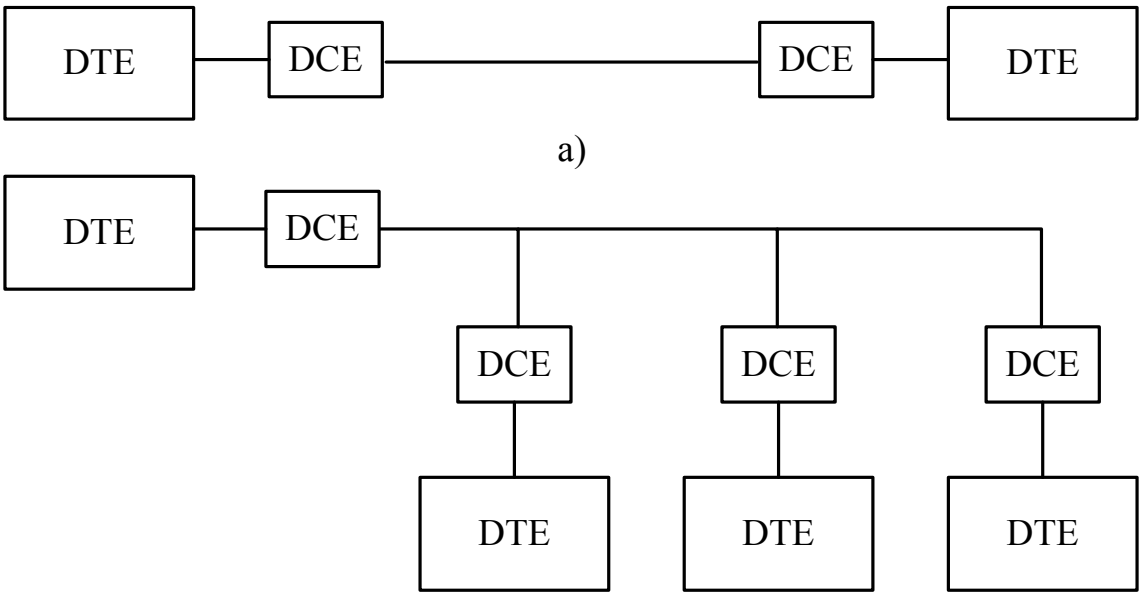


Рисунок 2.5 – Організація каналу типу «point-to-point» (а)
та «point-to-multipoint» (б)

За **значенням бази сигналу** розрізняють канали:

- вузькосмугові;
- широкосмугові.

У теорії зв'язку всі сигнали поділяють на прості та складні. За визначенням, простим сигналом називають сигнал, для якого база $B \approx 1$, де база сигналу – це добуток тривалості сигналу та смуги частот, яку цей сигнал займає. Для складного сигналу $B >> 1$. Вузькосмугові канали передають прості сигнали, а широкосмугові – складні.

2.3 Типи кабельних систем

Лінії зв'язку розрізняються також за типом фізичного середовища, яке використовується для передавання інформації. **Фізичне середовище передавання даних** у комп'ютерних мережах може бути як проводовим, так і безпроводовим. Проводові канали для комп'ютерних мереж зазвичай створюються на основі кабельних ліній зв'язку. В безпроводових каналах широко використовуються радіоканали наземного та супутникового зв'язку. В сучасних комп'ютерних мережах використовуються всі зазначені вище типи зв'язку. В даному розділі розглядаються тільки проводові кабельні системи, а безпроводові канали будуть описані в розділі 10.

Кабельні канали складаються з проводів, які мають у загальному випадку декілька шарів ізоляції: електричної, електромагнітної, механічної та, іноді, кліматичної і, зазвичай, роз'ємів, за допомогою яких спрощується процес підключення до різноманітного мережного обладнання. В комп'ютерних і телекомунікаційних мережах на сьогодні використовуються три основних типи кабелю:

- кабелі на основі скручених (кручених) пар, які бувають таких типів:
  - неекранована скручена пара **UTP** (Unshielded Twisted Pair);
  - екранована скручена пара **STP** (Shielded Twisted Pair);
  - фольгована скручена пара **FTP** (Foiled twisted pair, відома також як **S/UTP** (Shielded UTP);
  - захищена скручена пара **STP** (Shielded twisted pair), у якій для кожної пари проводів використовується свій екран;
  - фольгована екранована скручена пара **S/FTP** (Shielded Foiled twisted pair);
  - захищена екранована скручена пара **S/STP** (Screened Shielded Twisted Pair) – відрізняється наявністю додаткового загального зовнішнього екрана;
- коаксіальні кабелі;
- оптоволоконні кабелі.

Потрібно зазначити, що найчастіше з кабелів на основі скрученої пари використовуються кабелі UTP та STP. Коаксіальні кабелі і скручену пару часто називають мідними кабелями.

На сьогодні існують такі стандарти на кабелі:

- американський: EIA/TIA 568 (Commercial Building Telecommunications Cabling Standard);
- міжнародний: ISO/IEC IS 11801 (Generic cabling for customer premises);
- європейський: CENELEC EN 50173 (Generic cabling systems).

2.3.1 Скручена пара. Один з перших типів кабелю, скручена пара, і до цього часу широко використовується для побудови кабельної системи передавання даних у комп'ютерних мережах. Скручена пара має два ізольо-

вані мідні проводи, зазвичай товщиною 1 мм, скручені у вигляді спіралі, що дозволяє зменшити електромагнітну взаємодію декількох скручених пар, які розташовані поруч (рис. 2.6).

Скручені пари використовуються для передавання як аналогових, так і цифрових сигналів. Смуга пропускання залежить від діаметра і довжини проводу, а максимальна швидкість передавання забезпечується при передаванні на відстань до декількох кілометрів. Безперечною перевагою кабелів на основі скручених пар є достатньо висока пропускна спроможність і невелика вартість.

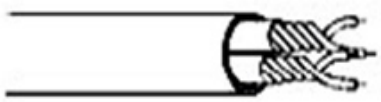


Рисунок 2.6 – Кабель «скручена пара»

Відповідно до стандарту EIA/TIA 568, існують такі категорії кабелів на основі **неекранованої скрученої пари (UTP)**:

- **категорія 1 (UTP1)** – звичайний телефонний кабель зі смугою частот 0,1 МГц, який має одну пару проводів (плоский кабель). Використовувався раніше, а на сьогодні використовується для передавання голосового потоку або даних за допомогою модема;
- **категорія 2 (UTP2)** (смуга частот 1 МГц) – 2 пари проводів, кабель цієї категорії підтримував передавання даних на швидкостях до 4 Мбіт/с, використовувався в мережах Token Ring та ACNet;
- **категорія 3 (UTP3)** – 4 пари проводів, кожна з яких складається з двох скручених проводів, мають єдину пластикову оболонку, смуга частот 16 МГц, використовується в локальних мережах 10BASE-T і Token Ring та підтримує швидкість передавання даних до 10 Мбіт/с або 100 МБіт/с за технологією 100BASE-T4 і відповідає вимогам стандарту IEEE 802.3;
- **категорія 4 (UTP4)** – кабель складається з 4-х скручених пар зі смугою частот 20 МГц, використовується в мережах Token Ring, 10BASE-T, 100BASE-T4, швидкість передавання даних не перевищує 16 Мбіт/с у одній парі, зараз майже не використовується;
- **категорія 5 (UTP5)** – 4-парний кабель зі смугою частот 100 МГц, використовується в локальних мережах 100BASE-TX та підтримує швидкість передавання даних до 100 Мбіт/с при використанні 2-х пар проводів. Зазвичай при розробці нових мереж використовують вдосконалений кабель **категорії 5e**, який дозволяє забезпечити швидкість передавання до 100 Мбіт/с при використанні 2-х пар проводів і до 1000 Мбіт/с при використанні 4-х пар;
- **категорія 6 (UTP6)** – кабель зі смугою частот 250 МГц, застосовується в мережах Fast Ethernet та Gigabit Ethernet, має 4 пари прово-

дів і дозволяє передавати дані на швидкості до 1000 Мбіт/с. Доданий у стандарт у червні 2002 року. Існує також модифікація цього кабелю: **категорія 6а**, яка використовує частоту передаваного сигналу 500 МГц. На сьогодні це найбільш поширений тип кабелю;

- **категорія 7 (UTP7)** – специфікація на даний тип кабелю поки не затверджена, швидкість передавання даних до 100 Гбіт/с, частота сигналу, який передається, до 600–1200 МГц. Кабель цієї категорії екранований і належить, за всіма характеристиками, до кабелів типу S/FTP (Screened Fully shielded Twisted Pair).

Кабелі категорій 6 та 7 використовуються у високошвидкісних магістралях у сегментах великої довжини, однак вартість кабелю категорії 7 на сьогодні приблизно така ж, як і оптоволоконного кабелю, якісні характеристики якого значно вищі.

Кабелі на основі **екранованої скрученої пари STP** добре захищають передавані сигнали від зовнішніх завад, а користувачів мережі – від шкідливого випромінювання. Для кабелів цієї категорії необхідне якісне заземлення екрана, що збільшує їх вартість та ускладнює прокладання.

Основним стандартом, який визначає типи та параметри екранованої скрученої пари, є стандарт IBM, згідно з яким всі кабелі поділяють не на категорії, а на типи: **Type 1–Type 9**.

Кабелі Type 1 – це основний тип кабелю, параметри якого приблизно відповідають параметрам кабелю UTP5. Використовується в локальних мережах Token Ring, 100VG – Any LAN, Fast Ethernet. Кабель STP1 має міжнародний статус завдяки введенню в міжнародну систему стандартів.

Кабелі Type 2 – це кабель STP1 з двома додатковими парами неекранованих проводів, які використовуються для передавання голосових потоків.

Потрібно зазначити, що не всі типи кабелів стандарту IBM відносять до екранованих скручених пар, наприклад, кабель STP3 – це неекранований телефонний кабель, кабель STP5 – це оптоволоконний кабель.

2.3.2 Коаксіальний кабель. Коаксіальні кабелі мають кращу завадостійкість, ніж кабелі на основі скрученої пари, та забезпечують передавання на більші відстані при вищих швидкостях. Розрізняють два типи коаксіального кабелю.

Перший тип (з хвильовим опором 50 Ом) використовується для передавання цифрових даних. Його пропускна спроможність суттєво залежить від довжини кабелю й може досягати 1–2 Гбіт/с при довжині 1 км. На сьогодні коаксіальні кабелі замінюються на оптоволоконні, однак ще використовуються в деяких локальних мережах та кабельному телебаченні.

Другий тип – широкосмуговий коаксіальний кабель – використовується для передавання аналогових даних (наприклад, стандартний телевізійний кабель) і називається широкосмуговим. Термін «широкосмуговий» походить від системи телефонії, де він означав смугу частот ширшу за 4 кГц,

однак у технології комп'ютерних мереж широкосмуговий кабель означає кабель, що використовується для передавання аналогових даних.

Коаксіальний кабель (рис. 2.7) – це електричний кабель, який складається з центрального мідного проводу (1), внутрішньої ізоляції (2), металевого обплетення (3) та зовнішньої ізоляції (4).

Коаксіальні кабелі розділяються за шкалою **Radio Guide**. Найбільш поширені категорії кабелю:

- **RG8 та RG11** – коаксіальний кабель, розроблений для мереж Ethernet 10BASE5 («товстий» Ethernet – Thick Ethernet, yellow Ethernet, Thicknet), має хвильовий опір 50 Ом, зовнішній діаметр 0,5 дюйма (приблизно 12 мм). Характеризується складністю монтажу (для підключення до модуля необхідно додатково використовувати трансиверний кабель та трансивер) і високою вартістю, але має хороші механічні та електричні характеристики;

Рисунок 2.7 – Коаксіальный кабель

- **RG58** – коаксіальний кабель, розроблений для мереж Ethernet 10BASE2 («тонкий» Ethernet – Thin Ethernet, Thinnet, Cheapernet), має хвильовий опір 50 Ом, зовнішній діаметр приблизно 6 мм. Має гірші характеристики, ніж «товстий» кабель, для підключення до модуля використовується роз'єм типу BNC. Існують такі модифікації:
  - **RG58/U** – суцільний центральний провідник;
  - **RG58A/U** – багатожильний центральний провідник;
  - **RG58C/U** – кабель з військовим прийманням;
- **RG59** – телевізійний кабель з хвильовим опором 75 Ом широко використовується в кабельному телебаченні для широкосмугової передачі (Broadband/Cable Television);
- **RG11** – магістральний кабель для передавання на значні відстані;
- **RG62** – коаксіальний кабель з хвильовим опором 93 Ом використовувався в уже застарілих локальних мережах ARCNet.

2.3.3 Оптоволоконні кабелі. Найбільш перспективним середовищем передавання, яке забезпечує швидкість передавання понад 40 Гбіт/с на відстань до 100 км, є оптоволоконний кабель. Оптоволоконні кабелі мають центральний провідник світла (серцевину) – скляне або пластикове волокно, оточене іншим шаром скла (оболонки), у якого показник заломлення менший, ніж у центрального, та захисне покриття (рис. 2.8). Розповсюджуючись по центральному скляному провіднику, світло не виходить за його межі, а відбивається від поверхні оболонки. Промені світла,

які входять в оптоволокно під різними кутами, називають ***модами***. **Залежно від кількості світлових потоків** розрізняють два типи оптоволоконного кабелю:

- в **одномодовому** оптоволоконному кабелі **SMF (**Single Mode Fiber**)** передається тільки один світловий потік; діаметр серцевини 5–15 мікрон; використовується при організації магістральних каналів для передавання на великі відстані;
- **багатомодові** оптоволоконні кабелі **MMF (**Multi Mode Fiber**)** дозволяють паралельно передавати декілька світлових потоків (мод) і мають більший діаметр центрального провідника (40–100 мікрон).

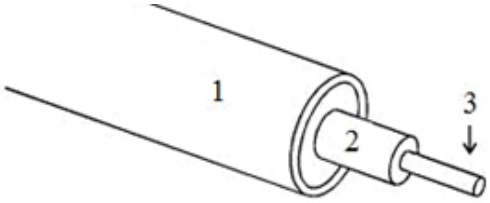


Рисунок 2.8 – Структура оптоволоконного кабелю

Оптичне волокно розрізняють за характером розподілення показника заломлення вздовж діаметра центрального скловолокна. Виділяють:

- одномодове волокно, яке має однаковий показник заломлення по всій серцевині (рис. 2.9, а);
- оптичне волокно зі **ступінчастою** зміною моди, в якому показник заломлення однаковий по всьому діаметру серцевини, а на межі з оболонкою різко змінюється (рис. 2.9, б);
- **градієнтне** оптичне волокно, в якому показник заломлення плавно зменшується від центра серцевини до її периферії (рис. 2.9, в).

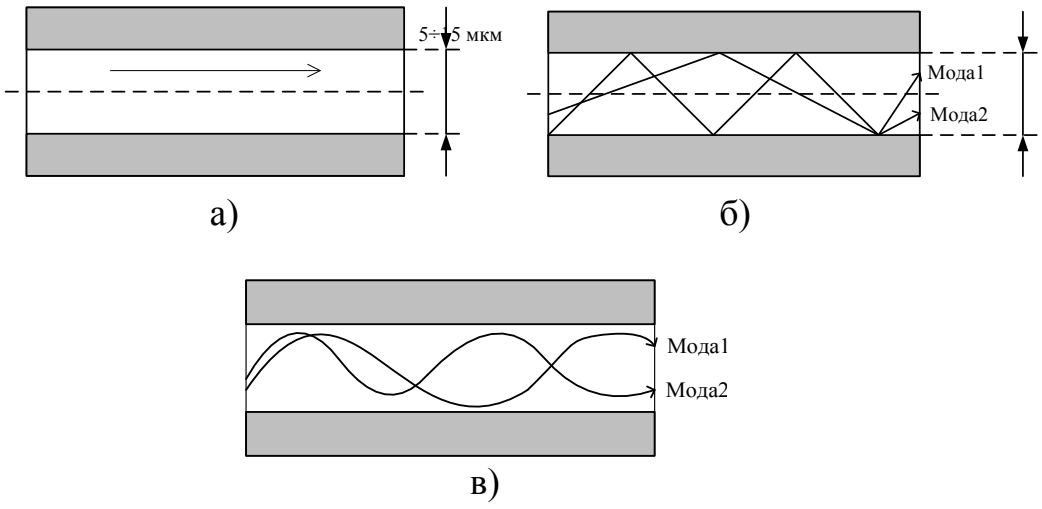


Рисунок 2.9 – Одномодове (а) та багатомодове оптоволокно зі ступінчастою зміною моди (б), градієнтне оптоволокно (в)

Діаметр серцевини одномодового оптоволокна зазвичай складає від 5 до 15 мікрон, градієнтного – від 50 до 62,5 мікрона, а ступінчастого – від 100 до 500 мікрон. В специфікації на оптичне волокно зазвичай вказується два числа (наприклад, 62,5/125), перше з яких відповідає діаметру серцевини, а друге – діаметру зовнішньої оболонки (обидва значення задані в мікронах). Найбільш поширені багатомодові оптичні волокна таких типів: 50/125 (європейський стандарт) та 62,5/125 (південноамериканський та японський стандарти), а також одномодові 9/125.

Затримка передавання сигналу в оптоволоконному кабелі становить приблизно 4–5 нс/м.

2.4 Методи передавання дискретних даних на фізичному рівні

При видачі інформації в канал зв'язку визначається спосіб подання даних сигналами такої форми, яка використовується в даному середовищі передавання. В загальному випадку ця процедура реалізується з використанням логічного та фізичного кодування. **Логічне кодування** виконується до фізичного і виконує заміну бітів вихідної послідовності такою новою послідовністю бітів, яка передає ту ж саму інформацію, але має й додаткові можливості для приймальної станції (зазвичай це можливість виявляти помилки передавання). Логічне кодування дозволяє також забезпечити конфіденційність передавання даних за рахунок їх шифрування, виявити помилки в прийнятій послідовності сигналів, визначити довгі монотонні послідовності нульових і одиничних сигналів, пов'язаних з особливими станами каналу. Зрозуміло, що при логічному кодуванні вихідна послідовність збільшується, що призводить до зменшення пропускної спроможності каналу.

Фізичне кодування передбачає вибір способу подання дискретної інформації у вигляді послідовності сигналів, які передаються в канал. При передаванні даних по каналах зв'язку застосовується два основних типи фізичного кодування:

- на основі синусоїдального несучого сигналу – цей спосіб часто називають **аналоговою модуляцією** чи просто **модуляцією (маніпуляцією)**;
- на основі послідовності прямокутних імпульсів – цей спосіб зазвичай називають **цифровим кодуванням**.

Ці способи відрізняються шириною спектра підсумкового сигналу та складністю апаратури, що використовується для їхньої реалізації.

При використанні прямокутних імпульсів спектр підсумкового сигналу виходить досить широким. Використання синусоїдальних сигналів приводить до спектра меншої ширини з тією ж швидкістю передавання даних. Проте для цього потрібна більш складна апаратура, ніж використовується для реалізації прямокутних імпульсів.

На сьогодні все частіше дані, що мають аналогову природу (голосові, аудіо- та відеопотоки), передаються по цифрових каналах зв'язку в дискретній формі, тобто у вигляді послідовності 0 та 1.

Процес подання аналогової інформації у дискретній формі називається **дискретною модуляцією.**

Потрібно зазначити, що терміни модуляції та кодування часто використовують як синоніми, а замість терміна «модуляція» в літературі використовують поняття «маніпуляція». Однак між цими двома поняттями існують деякі відмінності, а саме: якщо модульований сигнал передає дискретну інформацію, то замість терміна «модуляція» використовують поняття «маніпуляція», в інших випадках – тільки термін «модуляція».

2.4.1 Логічне кодування. Логічне кодування перетворює потік бітів кадру, сформованого на канальному рівні, в послідовність символів для фізичного кодування, після виконання якого вона передається в канал. Логічне кодування дозволяє уникнути довгих послідовностей нульових та одиничних бітів, які порушують синхронізацію модулів, і виявити (а може й виправити) помилки передавання. Крім того, методи логічного кодування дозволяють розпізнати межі кадру та виявити особливі стани в неперервному потоці бітів.

Для логічного кодування розрізняють два методи:

- надлишкові коди;
- скремблювання.

Найбільш розповсюдженими методами надлишкових кодів є коди 4B/5B, 5B/6B, 8B/6T та 8B/10B.

У коді 4B/5B будь-яка комбінація з чотирьох інформаційних бітів замінюється в передавачеві на п'ять бітів коду, в модулі-отримувачі виконується обернене перетворення (з п'яти бітів кодової послідовності отримують 4 біти інформаційної послідовності). При такому підході надлишковість буде дорівнювати двом (з 32 можливих комбінацій для передачі використовується 16). Кодування символів наведено в табл. 2.3. Інші 16 комбінацій використовуються як службові для виконання особливих функцій, наприклад, визначення меж кадру, керування потоком і перевірки коректності. Описаний код використовується в мережі Ethernet стандарту 100BaseFX (оптоволоконний кабель) та в мережі FDDI. Що стосується синхронізації, то слід зазначити, що синхронізація модуля-отримувача виконується один раз на 4 біти.

За таким же принципом побудовано й інші коди цього класу, наприклад, код **5B/6B**, у якому 5 бітів інформаційного потоку, кодується 6 бітами коду (використовується в стандартній мережі 100VG-Any LAN), або код **8B/10B**, який має чотирикратну надлишковість (256 з 1024 комбінацій), що використовується в мережах Gigabit Ethernet.

У коді **8B/6T** застосовується інший підхід: 8 бітів інформаційної послідовності кодуються шістьма трійковими цифрами (0, 1 та 2). Такий підхід використовується в сегменті 100BASE-T4 мережі Fast Ethernet і передбачає

паралельне передавання трьох трирівневих сигналів трьома скрученими парами. Такий підхід дозволяє забезпечити швидкість передавання 100 Мбіт/с у кабелях категорії 3 зі смугою пропускання 6 МГц.

Таблиця 2.3 – Послідовності коду 4B/5B

Інформаційна послідовність	Остаточна послідовність
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Скремблювання передбачає побітове обчислення вихідної кодової послідовності на основі значень вхідної інформаційної послідовності і вже обчислених у попередніх тактах бітів підсумкового коду. Наприклад, скремблер реалізує таке співвідношення:

$$B_i = A_i \oplus B_{i-3} \oplus B_{i-5},$$

де B_i – двійкова цифра підсумкової кодової послідовності;

A_i – двійкова цифра вхідної інформаційної послідовності;

B_{i-3} та B_{i-5} – двійкові цифри підсумкового коду, які отримані на попередніх тактах роботи скремблера, відповідно на 3 та 5 тактів раніше поточного такту;

$\oplus$ – операція додавання за модулем 2.

Після отримання підсумкової послідовності модуль-отримувач передає її в дескремблер, який відновлює інформаційну послідовність на основі оберненого перетворення.

Різні алгоритми скремблювання відрізняються різною кількістю доданків та різним зсувом між доданками. Наприклад, у мережах ISDN використовується два варіанти скремблювання: зі зсувом 5 і 23 та зі зсувом 18 і 23.

Існують і простіші методи боротьби з довгими послідовностями одиниць, які також відносять до скремблювання, але будуть розглянуті нами в наступному розділі (методи HDB3 – High Density Bipolar 3 та B8ZS – Bipolar with 8 Zeros Substitution).

2.4.2 Фізичне кодування. Способи модуляції. Модеми. Для передавання дискретних даних по аналогових каналах, які є каналами з вузькою смугою частот, використовується аналогова модуляція. Прикладом таких каналів є канали тональної частоти, які призначені для передавання голосового потоку. І хоча голос може займати діапазон 100–18000 Гц, для передавання голосового сигналу прийнятної якості використовується канал з вужчою смугою пропускання (смугою частот): для Європи – 300–3400 Гц, для Америки – 300–3300 ГЦ. Типова амплітудно-частотна характеристика каналу тональної частоти наведена на рис. 2.10.

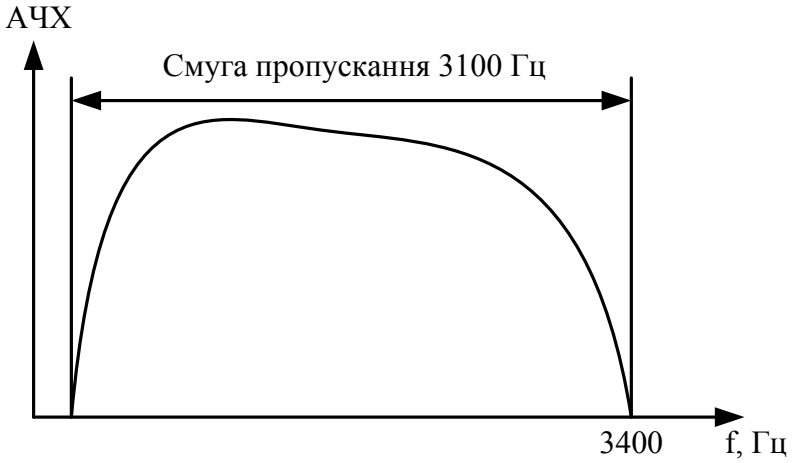


Рисунок 2.10 – АЧХ каналу тональної частоти

Таке строге обмеження смуги пропускання каналу тональної частоти пов’язано з використанням апаратури ущільнення та комутації каналів у телефонних мережах. Перенесення сигналу в канал заданої частоти виконується модемом за допомогою **модуляції**.

Модуляція – процес зміни одного чи декількох параметрів вихідного сигналу за законом зміни вхідного. При цьому вхідний сигнал (який може бути як аналоговим, так і цифровим) називається **модулювальним**, а вихідний (аналоговий) – **модульованим**. Залежно від того, який сигнал (аналоговий чи цифровий) є вхідним, розрізняють, відповідно, аналогову та цифрову модуляції.

Оскільки для аналогових каналів основними характеристиками сигналів є амплітуда, частота та фаза, використовують амплітудну, частотну, фазову модуляції (рис. 2.11), а також їх комбінації. На сьогодні стандартизовано багато різних способів модуляції. Ті з них, які знайшли своє використання в каналах комп’ютерних мереж, наведені в таблиці Б.1.

При **амплітудній модуляції (AM) ASK** (Amplitude Shift Keying) змінною є тільки амплітуда сигналу, при цьому і частота, і фаза мають постійне значення. На практиці в чистому вигляді використовується дуже рідко через низьку завадостійкість. Тому даний підхід використовують, зазвичай, разом з фазовою модуляцією.

При **частотній модуляції (ЧМ) FSK** (Frequency Shift Keying) для передавання нульових та одиничних бітів змінюється тільки частота сигналу, при цьому і його амплітуда, і фаза залишаються незмінним. Розрізняють **двійкову FSK** (BFSK – Binary FSC), при якій при передаванні використовується тільки два значення частоти сигналу; **чотирирівневу FSK** (FFSK – Four-level FSK), при застосуванні якої використовується чотири значення частоти сигналу і, відповідно, за один такт (один період зчитування з каналу) передається чи приймається з каналу одразу два біти інформації, та **багаторівневу FSK** (MFSK – Multi-level FSR). Зазвичай способи частотної модуляції використовуються у низькошвидкісних модемах.

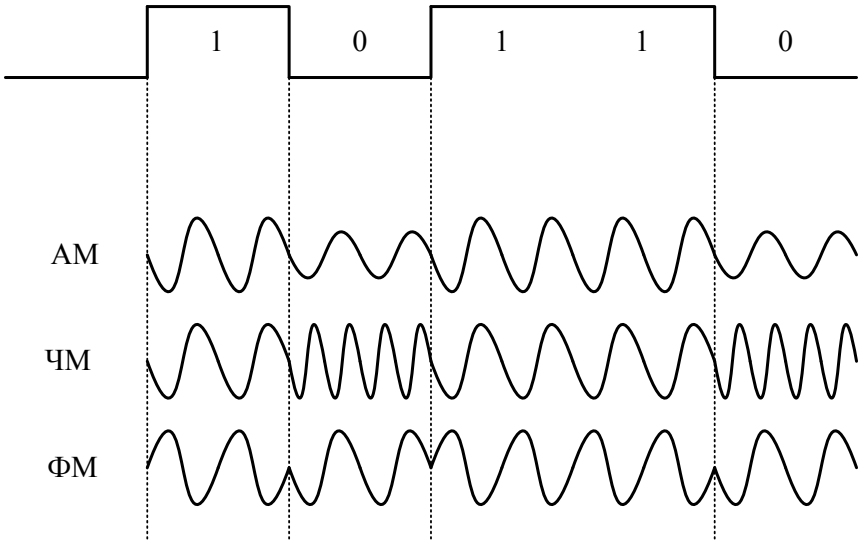


Рисунок 2.11 – Типи модуляції

При **фазовій модуляції (ФМ) PSK** (Phase Shift Keying) нульові та одиничні біти передаються сигналами однакової амплітуди та частоти, але з різною фазою. Якщо використовується два значення фази (0° та 180°), маємо справу з **двійковою PSK (**BPSK – Binary PSK), якщо чотири фази (0°, 90°, 180° та 270°) – то з квадратурною фазовою модуляцією **Quadrature PSK**.

У чистому вигляді такі способи модуляції на сьогоднішній день майже не використовуються. Для підвищення швидкості передавання даних використовують комбіновані способи модуляції, з яких найбільш розповсюджені способи квадратурної амплітудної модуляції **QAM-N** (Quadrature Amplitude Modulation), які є комбінацією амплітудної та фазової модуляцій (зауважимо, що для визначення цього способу модуляції часто використовується абревіатура **КАМ** – квадратурно-амплітудна модуляція). N означає кількість станів несучого сигналу і може приймати значення 4, 8, 16, 32, 64, 128, 256. Кількість бітів, яка передається одним станом, визначається як $log_2 N$, де N означає рівень модуляції. На рис. 2.12 наведено кодування за методом QAM-16 (рис. 2.12, а) в порівнянні з методом фазового коду-

вання PSK-16 (рис. 2.12, б) при однаковій потужності сигналу. Видно, що кодова відстань *d* між сусідніми точками сигнальної сукупності в системі QAM вища за аналогічний показник в системі PSK, що означає і більшу захищеність від помилок.

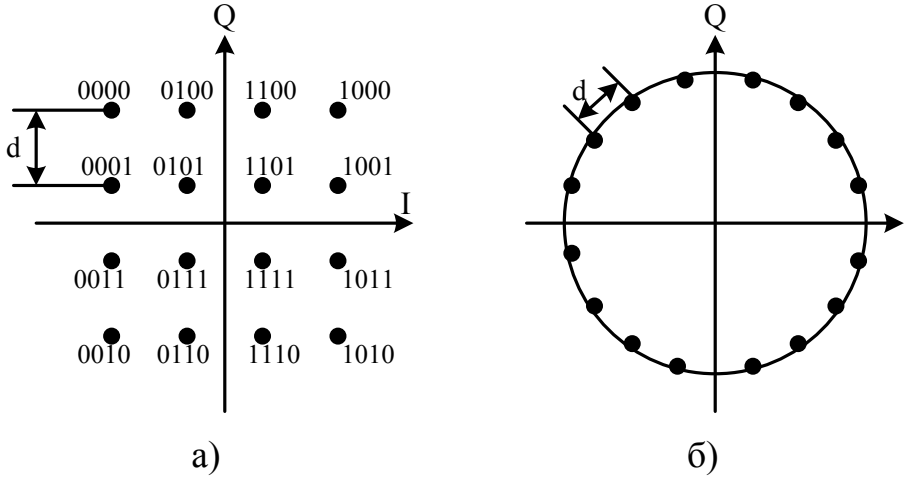


Рисунок 2.12 – Модуляція QAM-16 (а) та PSK-16 (б)

У коаксіальних лініях, які характеризуються високим співвідношенням сигнал/шум, використовується модуляція QAM різного рівня, зазвичай, QAM-64. Для ретрансляції супутникових потоків найчастіше використовується модуляція QAM-32, а в сучасних кабельних мережах найбільш доцільним є використання модуляцій QAM-128 та QAM-256, що веде до більш жорстких вимог до амплітудно-частотних характеристик сигналу і рівня ехо-сигналів в тракті. QAM-модуляція також широко використовується для передачі даних в мережах кабельного телебачення.

Більш сучасний, довершений метод модуляції – це **трелліс-модуляція ТСМ** (Trellis Coded Modulation). Його перевагою є не тільки збільшення кількості бітів, які передаються в одному такті, а й зниження вимог до каналу зв’язку (зниження співвідношення *S/N* на 3–6 дБ), що використовується у високошвидкісних модемах і відповідає рекомендаціям стандарту V.34.

Модеми. Зазвичай модемом називають пристрій передавання даних, який на стороні передавача приймає потік бітів від термінала (DTE) і перетворює його в модульований сигнал відповідної форми, що передається в канал, а при отриманні сигналу на стороні приймача виконуються обернені дії – модульований сигнал перетворюється в цифрові імпульси, які передаються в термінал пункту призначення. Однак на сьогодні модем виконує значно більший набір функцій залежно від особливостей його реалізації та використання. В більш широкому сенсі модемом називають пристрій передавання даних (DCE), який забезпечує інтерфейс між терміналом та каналом зв’язку.

Чіткої класифікації модемів не існує і в принципі не може існувати через велике різноманіття самих модемів, сфер їхнього застосування та режимів роботи. Але можна назвати низку ознак, за якими проводиться умовна класифікація. До таких ознак належать:

- галузь застосування;
- тип каналу, який використовується для передавання;
- метод передавання даних;
- режим роботи;
- конструктивне виконання;
- підтримка протоколів модуляції, ущільнення даних, корекції помилок тощо.

Розглянемо узагальнену класифікацію модемів без деталізації основних характеристик і особливостей функціонування та виконання.

За галуззю застосування виділяють такі групи:

- для комутованих телефонних каналів;
- для виділених (орендованих) телефонних каналів;
- для фізичних ліній:
  - модеми на короткі відстані (short range: SR-модеми, від 1,75 до 10 км), на середні відстані (medium range: MR-модеми, від 8 до 9 км), на великі відстані (extended range: ER-модеми, до 15 км);
  - модеми основної смуги (baseband);
- для цифрових систем передавання (CSU/DSU, каналів EI/T1 або ISDN);
- для сотових систем зв'язку;
- для пакетних радіомереж тощо.

Переважну кількість модемів призначено для роботи з комутованими та орендованими телефонними каналами, забезпечуючи при цьому роботу з автоматичними телефонними станціями.

Модеми для фізичних ліній враховують їхню смугу пропускання, яка не обмежена значенням 3,1 кГц, що справедливо для телефонних каналів. Смуга пропускання фізичної лінії залежить від типу середовища передавання (скручена пара, коаксіальний кабель тощо) та її довжини. При цьому модеми на короткі відстані (низького рівня) передають в канал цифрові сигнали без їх модуляції, а модеми основної смуги використовують відповідні методи модуляції.

Модеми для цифрових ліній забезпечують підключення до стандартних цифрових каналів, наприклад, T1/E1 або ISDN, та підтримують функції відповідних канальних інтерфейсів.

Модеми для сотових систем підтримують спеціальні протоколи модуляції та корекції помилок, що дозволяє ефективно передавати інформацію в сотових каналах з високим рівнем завад.

Модеми для пакетних радіомереж призначені для передавання даних в радіоканалі між мобільними користувачами, використовуючи при цьому

один канал в режимі множинного доступу (зазвичай з контролем несучої) і відповідний метод модуляції.

За типом каналу, що використовується для передавання, розрізняють:

- модеми для 2-проводових мідних ліній (звичайні, професійні, ADSL-модеми, SR-модеми, ER-модеми);
- модеми для 4-проводових мідних ліній (звичайні, професійні, HDSL-модеми, ISDN-модеми, SR-модеми, ER-модеми, MR-модеми);
- модеми для оптоволоконних ліній (FOM, FOM-T1/E1, FOM-T2/E2, FOM-T2/E2);
- модеми для радіоканалів (радіомодем, сотовий модем);
- кабельні модеми (для коаксіального кабелю).

За методом передавання модеми поділяють на асинхронні та синхронні, маючи на увазі спосіб передавання даних в каналі зв'язку між модемами. При цьому модем може працювати з терміналом DTE в асинхронному режимі і одночасно з віддаленим модемом у синхронному режимі та навпаки.

За режимом роботи розрізняють модеми дуплексні, напівдуплексні та симплексні, тобто враховується спосіб організації каналу передавання.

За конструктивним виконанням розрізняють модеми:

- внутрішні;
- зовнішні;
- групові;
- портативні.

І якщо **зовнішні** модеми – це автономні пристрої, які підключаються до робочої станції або іншого термінала за допомогою стандартного інтерфейсу DTE-DCE, то **внутрішній** модем являє собою плату розширення, яка вставляється у відповідний слот комп'ютера. Групові модеми – це сукупність декількох модемів, об'єднаних в єдиний блок, що мають спільні блоки живлення, керування тощо. Кожний з модемів групового модема є платою, яка вставляється в блок, і призначений для підключення одного чи декількох каналів. **Портативні** модеми призначені для використання мобільними користувачами (ноутбуками) і хоча й відрізняються невеликими розмірами, та мають такі ж функціональні можливості, що й інші типи модемів.

За інтелектуальними можливостями виділяють такі типи модемів:

- без системи керування;
- з підтримкою набору АТ-команд;
- з підтримкою команд V.25bis;
- з фірмовою системою команд;
- з підтримкою протоколу керування мережею SMNP (Simple Manager Network Protocol).

Переважна більшість модемів має широкий спектр інтелектуальних можливостей та підтримує набір АТ-команд, які використовуються для керування процесом передавання та параметрами зв'язку. Вперше набір таких команд був запропонований компанією Hayes у 1977 році, але згодом неодноразово доповнювався та розширювався. На сьогодні модеми, які підтримують АТ-команди, називають Hayes-сумісними модемами. Більшість сучасних модемів має широкий спектр інтелектуальних можливостей.

Команди рекомендації V.25bis дозволяють керувати режимами встановлення з'єднання між взаємодійними модулями та процедурою передавання.

Спеціалізовані модеми для промислового використання часто мають фірмову систему команд, яка враховує особливості галузі застосування й відрізняється від набору АТ-команд.

Підтримка протоколу керування мережею SMNP в модемі дозволяє адміністратору керувати модулями мережі з віддаленого термінала.

Модеми класифікують також з урахуванням **протоколів, які в них реалізовані** і які визначають особливості функціонування модемів. Всі протоколи належать до двох великих груп: міжнародні та фірмові. Протоколи міжнародні розробляються і приймаються у вигляді рекомендацій серії V міжнародною організацією ITU-T (International Telecommunication Union – Telecommunication). Фірмові протоколи розробляються окремими компаніями-виробниками модемів і є стандартними протоколами де-факто.

Перелік та характеристики модемів наведено в додатку Б.2.

2.4.3 Методи кодування даних. При цифровому кодуванні дискретної інформації використовують:

- **потенціальні коди**, при використанні яких біти логічного 0 та логічної 1 подані різними потенціалами, тобто різними рівнями напруги сигналу;
- **імпульсні коди**, в яких для подання бітів логічного 0 та логічної 1 використовується або перепад потенціалу відповідного напрямку, або імпульс відповідної полярності.

Для передавання дискретної інформації потрібно вибирати такий метод кодування, який би одночасно відповідав нижченаведеним вимогам:

- при одній бітовій швидкості мав би найменшу ширину спектра підсумкового сигналу, який отримано в результаті кодування;
- забезпечував синхронізацію між модулем-відправником та модулем-отримувачем даних;
- мав би невисокий рівень постійної напруги в лінії зв'язку;
- мав можливість розпізнавати помилки і, за можливості, їх виправляти;
- мінімізував потужність передавача;
- мав підключення, що не потребує врахування полярності проводів у кожній парі;
- мав би низьку вартість передавання.

Спектр сигналу залежить як від способу кодування, так і від тактової частоти передавача. Більш вузький спектр сигналу дозволяє отримати більшу швидкість передавання в каналах з однаковою пропускною спроможністю.

Синхронізація взаємодійних модулів потрібна для того, щоб станція-отримувач точно визначала момент, коли потрібно прийняти з каналу нову порцію даних. Проблема синхронізації в комп'ютерних мережах вирішується значно складніше, ніж між блоками, що розташовані близько один від одного. Для вирішення цієї проблеми в комп'ютерних мережах зазвичай використовуються коди, що самосихронізуються, тобто такі, сигнали яких несуть для станції-отримувача інформацію про те, що потрібно виконати розпізнавання біта (чи декількох бітів у разі використання більшої кількості станів сигналу). Цією ознакою може служити будь-який перепад рівня сигналу (фронт). **Кодами, що самосинхронізуються**, називаються такі коди, які разом з послідовністю бітів передають і строб, що синхронізує їх прийом. При використанні як несучого синусоїдального сигналу проблеми забезпечення синхронізації не виникає, бо зміна амплітуди несучої частоти дає можливість визначити початок нового такту.

Розглянемо способи дискретного кодування даних, які знайшли найбільш широке використання в комп'ютерних мережах (рис. 2.13).

NRZ (Non-Return to Zero – без повернення до нуля) – потенціальний код, при використанні якого значенню логічного 0 відповідає нижній рівень, а логічної 1 – верхній. Варіант коду **NRZI** (Non Return to Zero Inverted) відповідає оберненій полярності. Такий спосіб кодування використовується в інтерфейсі RS-232, який застосовується для підключення через послідовний порт. Недоліками даних кодів є:

- високий рівень постійної напруги при передаванні послідовностей з однаковою кількістю одиничних та нульових бітів;
- широка смуга сигналу;
- при передаванні довгої послідовності одиничних чи нульових бітів ускладнюється синхронізація взаємодійних модулів;
- сигнал є полярним.

Диференціальний NRZ (який також відомий як NRZI – Non Return to Zero Invert to ones) є модифікацією попереднього способу кодування, однак стан біта, який використовується для кодування поточного біта, залежить від стану попереднього біта (диференціальне кодування). При цьому, якщо поточний біт послідовності дорівнює 0, то зберігається (повторюється) значення попереднього біта, якщо ж поточний біт 1, то його стан є інверсією попереднього. Даний код також не має можливості самосинхронізації. Перевагою такого способу є невелика смуга пропускання і неполярність сигналу. Використовується в мережах FDDI, Ethernet стандарту 100Base-FX та в стандарті ATM155.

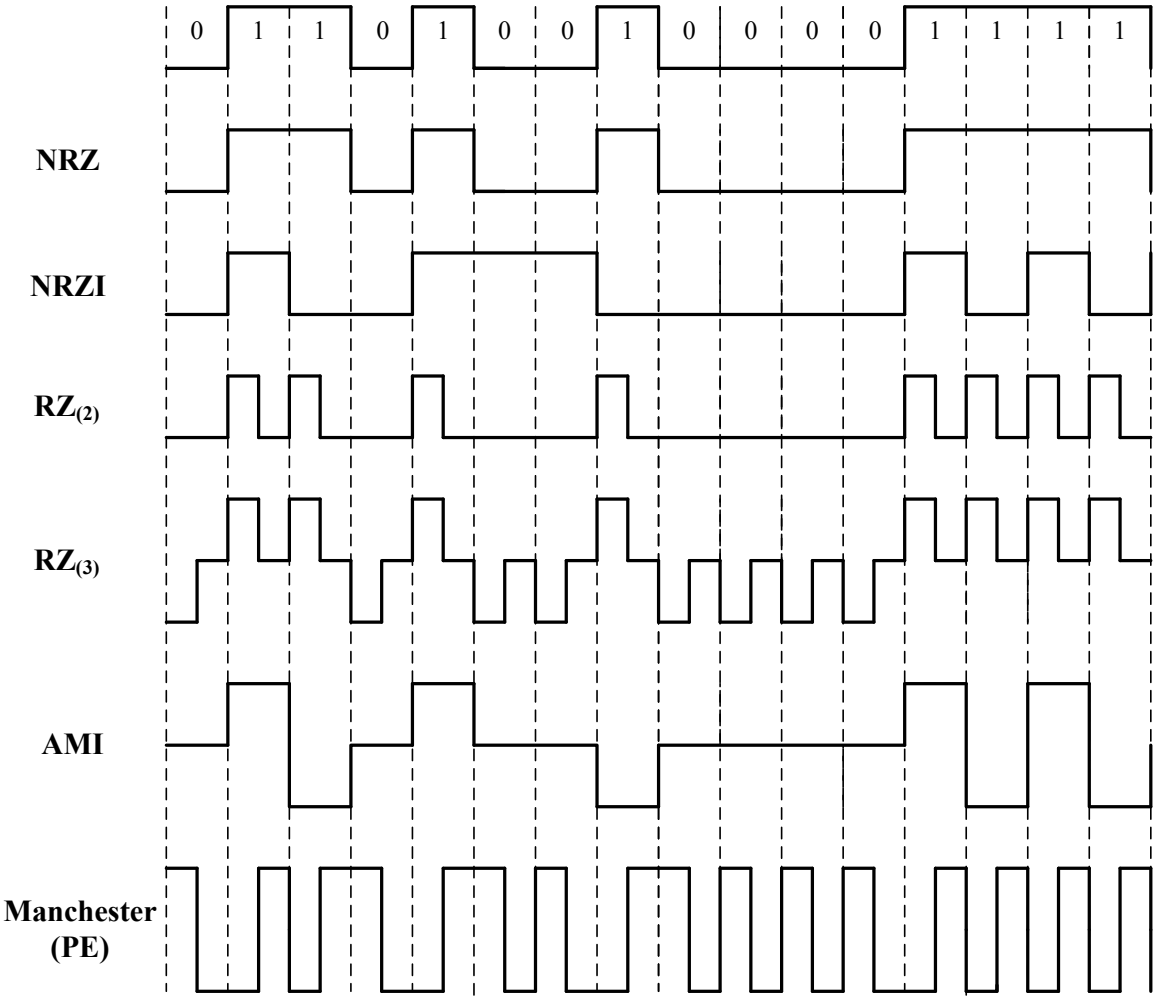


Рисунок 2.13 – Методи цифрового кодування

Код з поверненням до нуля **RZ** (Return to Zero) має дві модифікації: з двома станами біта ($RZ_{(2)}$) і з трьома ($RZ_{(3)}$), причому логічному нулю завжди відповідає додатній імпульс, а логічній одиниці – від'ємний. Інформаційний перехід здійснюється на початку передавання біта, повернення до нульового рівня – у середині біта. Особливістю даного коду є те, що в середині інтервалу передавання біта є додатній або від'ємний перехід, що забезпечує синхронізацію взаємодійних модулів.

Недоліком коду RZ є те, що він забезпечує ту ж швидкість передавання, що і попередні, тобто для забезпечення пропускної спроможності в 10 Мбіт/с необхідна частота несучої 10 МГц. Крім того, для розрізнення трьох рівнів необхідне краще співвідношення сигнал/шум на вході в приймач, ніж для дворівневих кодів.

Найчастіше код RZ використовується в оптоволоконних каналах. При цьому високий рівень біта відповідає наявності світла («сильне світло»), а низький – його відсутності. Використання трирівневого коду дозволяє навіть при відсутності передавання сигналу легко визначити цілісність лінії завдяки використанню третього стану сигналу («середнє світло»).

При біполярному кодуванні з альтернативною інверсією **AMI** (Alternate Mark Inversion) використовуються три стани: нульовий, додатній і від'ємний. Логічний нуль кодується нульовим бітом, а логічна одиниця – бітами, значення яких змінюється по черзі: +V, -V. Перевагою даного коду є забезпечення нульового рівня напруги в каналі, а також можливість підтримки синхронізації модулів у каналі при передаванні довгих послідовностей одиниць. Даний код є модифікацією коду NRZ, але має менший спектр сигналу, отже, забезпечує більшу пропускну спроможність. Крім того, дане кодування забезпечує розпізнавання деяких помилкових сигналів, що використовується в мережах ISDN і каналах типу DS-n.

Біполярне кодування з високою щільністю **HDB3** (High Density Bipolar 3) є модифікацією попереднього підходу, але дозволяє забезпечити синхронізацію і при передаванні довгих нульових послідовностей. Для цього передавання чотирьох послідовних нульових бітів замінюється послідовністю 000V, де значення біта V буде таким же, що і попереднього одиничного імпульсу (рис. 2.14). Таке кодування відповідає всім вимогам до цифрового кодування і використовується для кодування потоків у цифрових каналах E1 та E2. В Америці використовують модифікацію даного підходу – схему кодування **B8ZS** (Bipolar with 8 Zeros Substitution), у якій таким же чином замінюється послідовність з 8 нульових бітів.

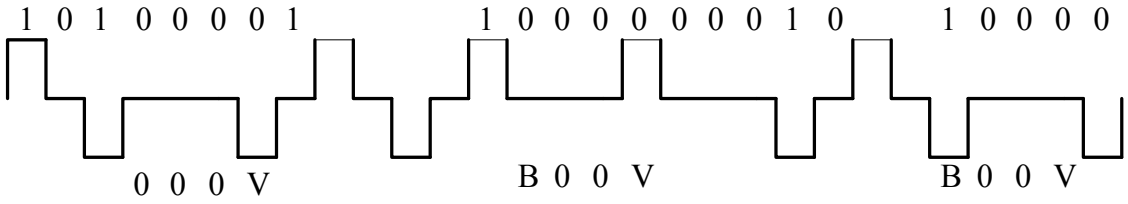


Рисунок 2.14 – Кодування з високою щільністю HDB3

У локальних мережах Ethernet та Token Ring найбільше використання знаходить **манчестерський код** (Manchester Encoding чи PE – Phase Encode). Це двофазний полярний самосинхронізувальний код, в якому логічна 1 кодується перепадом потенціалу від низького рівня до високого, логічний 0 – оберненим перепадом. Даний код задовольняє всі вимоги, але має широку смугу пропускання та є полярним.

Модифікацією цього методу є **диференціальне манчестерське кодування** (Differential Manchester Encoding), при використанні якого логічний 0 кодується наявністю на початку перепаду потенціалу, а логічна 1 – відсутністю перепаду, але при цьому для забезпечення синхронізації перепад у середині кожного такту є. В мережах Token Ring застосовується модифікація цього методу, крім «0» і «1», що використовує два службові біти, які не мають перепаду в середині такту.

Комбінацією методів NRZI та Manchester є кодування **CDP** (Conditional Diphase), яке розроблене компанією Cisco System. Використовується таке подання бітів цифрового потоку: біти 0 кодуються переходом у тому ж напрямку, що і попередній біт (від +V до -V або навпаки від -V до +V), а одиничні біти – переходом у напрямку, протилежному попередньому біту. При цьому забезпечується неполярний сигнал, який займає достатньо широку смугу пропускання.

Код **MLT-3** (Multi Level Transmission-3) має багато спільного з кодом NRZ, але використовує для передавання три рівні сигналу. Одиничному біту відповідає перехід з одного рівня сигналу на інший, нульовий біт кодується як збігання зі станом попереднього біта. Недолік даного коду – відсутність синхронізації, але він широко використовується в мережах FDDI (CDDI), Fast Ethernet стандарту 100Base-FX, а також у комунікаційних модулях компанії Cisco System.

Для підвищення пропускної спроможності каналу використовуються коди з більшою кількістю станів сигналу, що передається в каналі. Прикладом такого коду є лінійний код **2B1Q** (2 Binary 1 Quarternary), який використовується в мережах ISDN при реалізації базового інтерфейсу BRI (Base Rate Interface) і являє собою модифікацію коду NRZ, але з використанням 4-х станів сигналу. Біти інформаційного потоку кодуються у такій відповідності: бітам «00» відповідає рівень сигналу «-2,5 В», бітам «01» – рівень сигналу «-0,833 В», бітам «10» – рівень сигналу «+2,5 В», бітам «11» – рівень сигналу «+0,833 В» (рис. 2.15). Системи, які використовують даний код, забезпечують пропускну спроможність від 64 Кбіт/с до 2 Мбіт/с.

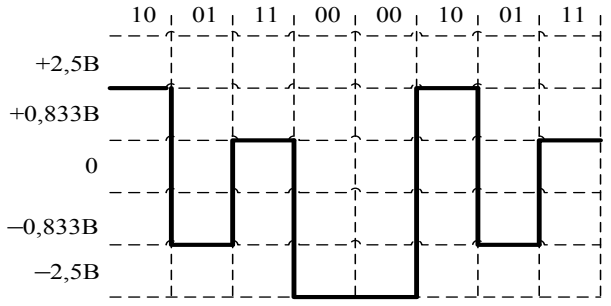


Рисунок 2.15 – Код 2B1Q

Модифікацією даного кодування є кодування **PAM-5** (Pulse-Amplitude Modulation 5), в якому для створення надмірності коду до чотирьох станів додається п’ятий рівень сигналу, що дає можливість виявляти та виправляти помилки. Використовується у мережі 1000Base-T Gigabit Ethernet, де забезпечує передавання даних зі швидкістю 1 Гбіт/с при ширині спектра сигналу 125 МГц. На практиці знаходять використання також коди PAM-8, PAM-16, PAM-32, які використовують іншу кількість станів сигналу.

2.5 Структуровані кабельні системи

Структурована кабельна система (СКС, Structured Cabling System – SCS) – це сукупність комутаційних елементів (кабелів, роз'ємів, конекторів тощо), які є основою інформаційної інфраструктури будь-якого підприємства. Ідея такого підходу була сформульована в першій версії стандарту TIA-568 (Telecommunication Industry Association – Асоціація телекомунікаційної галузі), яка була опублікована в 1991 році. Структуризація кабельної системи створює універсальне середовище для передавання комп'ютерного трафіку в локальних мережах, організації локальної телефонної мережі та передавання різноманітної інформації від датчиків різних типів.

У загальному випадку СКС охоплює такі три підсистеми:

- **горизонтальна підсистема** (горизонтальна СКС) відповідає поверхам будинку, в якому розташована дана компанія, і містить кабельну систему, розетки користувачів, комутаційне обладнання, до якого підключаються кабелі користувачів, кросові шафи поверху тощо;
- **вертикальна підсистема** (вертикальна СКС) містить магістральні кабелі між кросовими шафами поверхів і описує з'єднання окремих поверхів будинку, а також кожного поверху з центральним апаратним модулем будинку;
- **підсистема кампуса** (підсистема зовнішніх магістралей) з'єднує в єдину мережу окремі будинки, а також їх підключення до головного комунікаційного модуля кампуса. Зазвичай ця частина кабельної системи називається магістраллю. Підсистема кампуса відсутня, якщо СКС створюється тільки в одному будинку.

Такий підхід до створення СКС забезпечує достатньо простий доступ до комунікаційних ресурсів мережі, що забезпечує можливість динамічної зміни конфігурації кабельної системи. При створенні СКС закладається надлишковість (використання додаткової кількості кабелів, роз'ємів, кроспанелей тощо), завдяки якій можна підключати додаткові робочі станції, змінювати конфігурацію підключення модулів мережі.

На сьогодні існує три основних стандарти в сфері структурованих кабельних систем:

- **міжнародний стандарт**: ISO/IEC IS 11801-2002 Information Technology. Generic cabling for customer premises;
- **європейський стандарт**: CENELEC EN 50173 Information Technology. Generic cabling systems;
- **американський стандарт**: EIA/TIA-568C Commercial Building Telecommunications Wiring Standard.

Для стандартів ISO 11801-2002 и EN 50173 визначені такі класи кабельних ліній:

- C, до 16 МГц;
- D, до 100 МГц;

- E, до 250 МГц;
- E (A), до 500 МГц;
- F(A), до 600 МГц.

У стандарт EIA/TIA-568C для кабелів та компонентів входять такі категорії:

- категорія 3, яка дозволяє передавати сигнал у смузі до 16 МГц;
- категорія 5e, що має смугу частот до 100 МГц;
- категорія 6A, зі смугою частот до 500 МГц.

Мережі, створені на основі структурованих кабельних систем, мають такі характеристики:

- забезпечення централізованого або децентралізованого управління мережею залежно від поточних завдань;
- зміна фізичної та логічної топологій мережі;
- сегментація мережі за рахунок створення робочих груп користувачів або організації віртуальних локальних мереж (VLAN – Virtual Local Area Network);
- реконфігурація топології існуючої мережі та заміна старих низькошвидкісних каналів на нові високошвидкісні;
- швидке виявлення та ліквідація пошкоджень у кабельній системі мережі;
- можливість підключення нових користувачів.

2.6 Методи мультиплексування інформаційних потоків

Існує декілька способів підвищення пропускної спроможності систем передавання даних, більшість з яких використовує один із методів мультиплексування інформаційних потоків у один груповий, який і передається в каналі зв'язку. В результаті в лінії одночасно передається деяка кількість потоків, яка належить різним застосуванням різних користувачів. Мультиплексування (чи ущільнення каналу – multiplexing*)* для телекомунікаційних систем – це передавання даних по декількох логічних каналах у одному фізичному каналі. Для інформаційних технологій мультиплексування – це об'єднання декількох потоків даних (віртуальних каналів) у один.

У сучасних комп'ютерних мережах широке застосування знаходять такі методи мультиплексування:

- частотне ущільнення **FDM** (Frequency Division Multiplexing);
- часове ущільнення **TDM** (Time Division Multiplexing);
- мультиплексування за довжиною хвилі **WDM** (Wavelength Division);
- кодове ущільнення **CDM** (Code Division Multiplexing);
- мультиплексування за поляризацією **PDM** (Polarization Division Multiplexing);
- модове ущільнення **MDM** (Mode Division Multiplexing).

Частотне ущільнення використовується в аналогових каналах зв'язку і полягає в тому, що кожному інформаційному потоку для передавання по фізичному каналу виділяється канал з шириною смуги частот 4 кГц, з яких 3 кГц з центральною несучою частотою $f_{нi}$ використовуються для передавання потоку, 1 кГц – це смуга розфільтрування (захисні межі) між ними (рис. 2.16).

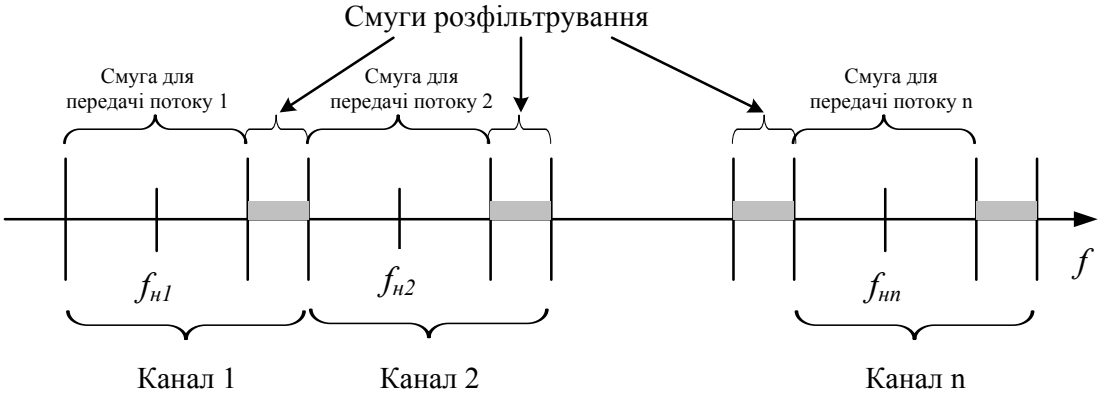


Рисунок 2.16 – Частотне ущільнення каналів

Міжнародний консультативний комітет з телефонії і телеграфії встановив таку ієрархію каналів із частотним ущільненням:

- первинна група шириною 48 кГц містить 12 каналів у діапазоні 60–108 кГц;
- вторинна група зі смугою частот 240 кГц містить 5 первинних груп (60 каналів) у діапазоні 312–552 кГц;
- третинна група зі смугою частот 1234 кГц містить 5 вторинних груп (300 каналів) у діапазоні 564–1798 кГц;
- четвертинна група зі смугою частот 3872 кГц містить 3 третинних групи (900 каналів).

Частотне ущільнення широко використовується в телефонних каналах і характеризується високою кратністю мультиплексування, але має невисоку завадостійкість, що пояснюється взаємним впливом потоків, які передаються в сусідніх каналах.

Часове ущільнення використовується в цифрових каналах і на сьогодні знайшло найбільше застосування серед усіх методів мультиплексування. При його використанні весь період роботи каналу розбивається на декілька часових інтервалів (тайм-слотів або тайм-доменів), кількість яких дорівнює кількості користувачів у групі та, відповідно, кількості каналів, що мультиплексуються в даному каналі. За допомогою TDM-комутатора канал кожного користувача по черзі підключається до каналу зв'язку, в результаті чого формується контейнер (кадр), який містить дані від усіх користувачів групи і передається як єдине ціле в каналі (рис. 2.17).

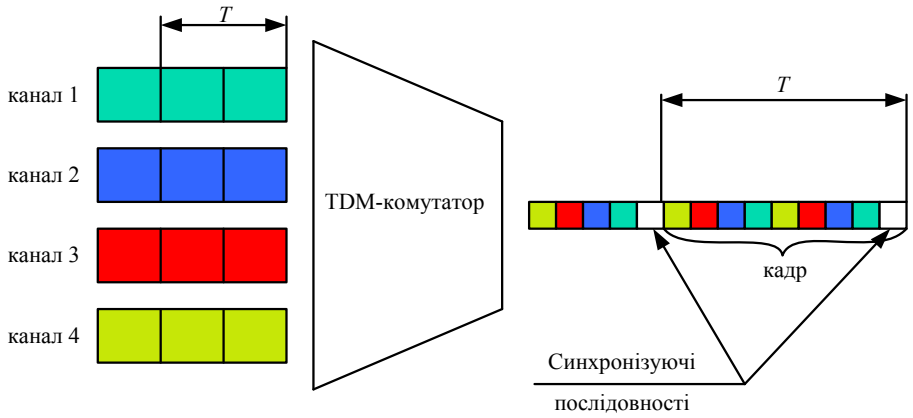


Рисунок 2.17 – Часове ущільнення каналів

Часове ущільнення може бути двох типів: асинхронним або плезіохронним (мережі ATM та технологія PDH) та синхронним (технологія SDH). Сучасні технології з часовим ущільненням дозволяють передавати потоки даних зі швидкістю 40 Гбіт/с (канал STM-256) і більше. Детальніше технології PDH та SDH будуть розглянуті в розділі 9.

Мультиплексування за **довжиною хвилі** WDM (іноді називають хвильове мультиплексування) широко використовується в оптоволоконних каналах. Суть даного підходу полягає в об'єднанні декількох оптичних несучих λ_i за допомогою спеціального комутатора на стороні передавача і передачі отриманого мультиплексованого потоку $\Sigma\lambda_i$ по одному оптоволоконному каналу. В станції-отримувачі за допомогою фільтрації окремі несучі виділяються з отриманого потоку.

Розрізняють три типи WDM мультиплексування:

- звичайні WDM-системи (часто називають грубими WDM-системами: CWDM – Coarse WDM), які дозволяють мультиплексувати не більше 16-ти каналів;
- щільні DWDM-системи (Dense WDM), які дозволяють мультиплексувати не більше 64-х каналів;
- надщільні HWDM-системи (High Dense WDM), які дозволяють ущільнити більше 64-х каналів.

Кодове ущільнення **CDM** передбачає передавання сигналів з використанням різних кодів, тобто кожному абоненту надається унікальний код, і всі дані кодуються з його допомогою. Як правило, послідовність одиничних сигналів у потоці даних замінюється на деякий CDM-код, а для заміни послідовності нульових сигналів – той же код, але інвертований. Модуль-отримувач знає CDM-код передавача, сигнали якого має отримувати. Сигнали від інших передавачів з іншими CDM-кодами отримувач сприймає як адитивний шум і не звертає на нього уваги. Даний метод часто називають ще методом розширення спектра прямої послідовності (**DSSS** – Direct Sequence Spread Spectrum), що буде розглянуто в розділі 9.

Перевагою даного способу є підвищена захищеність передавання даних, а також можливість присвоєння кожному передавачу свого індивідуального коду. Суттєвим недоліком кодового ущільнення є складність технічної реалізації та необхідність забезпечення точної синхронізації модулів передавача та отримувача для гарантованої передачі.

Кодове ущільнення використовується в технології зв'язку, яка реалізує множинний доступ з кодовим ущільненням **(CDMA** – CDM Access). Прикладом таких систем є стандарт стільникового телефонного зв'язку IS-95a, а також низка стандартів третього покоління стільникових систем зв'язку CDMA2000, WCDMA та інших.

Мультиплексування за **поляризацією PDM** – це ущільнення інформаційних потоків за допомогою оптичних несучих, які мають лінійну поляризацію. При цьому площина поляризації кожної несучої має бути розташована під своїм кутом. Таке мультиплексування виконується за допомогою спеціальних оптичних призм і можливо тільки при відсутності неоднорідностей в оптоволоконному кабелі. Використовується в оптичних ізоляторах та підсилювачах.

Модове ущільнення **MDM** використовується в каналах передавання на основі багатомодового оптоволокна. Кожний інформаційний потік передається за допомогою оптичного променя, який вводиться в канал під своїм кутом. Модове ущільнення може використовуватись при відсутності перемішування та взаємного перетворення мод оптичних променів. Це потребує підвищених вимог до оптоволоконних каналів щодо відсутності неоднорідностей та згинання кабелю.

2.7 Питання для самоперевірки

1. Охарактеризуйте базові функції протоколів фізичного рівня.
2. Поясніть залежність між пропускною спроможністю каналу та шириною його смуги пропускання.
3. Проаналізуйте формули Г. Найквіста та Клода Шеннона.
4. Визначте максимальну пропускну спроможність безшумних каналів зв'язку для кожного з напрямків дуплексного каналу, якщо відомо, що його смуга пропускання дорівнює 20 кГц, а метод кодування використовує передачу цифрових сигналів з двома бітами на кожне зчитування (сканування).
5. Визначте максимальну пропускну спроможність безшумних каналів зв'язку для напівдуплексного каналу, якщо відомо, що його смуга пропускання дорівнює 10 кГц, а метод кодування використовує передачу цифрових сигналів з трьома бітами на кожне зчитування (сканування).
6. Визначте, чи може канал зі смугою пропускання 4 кГц і співвідношенням сигнал/шум 20 дБ передавати цифрові сигнали зі швидкістю 50 Кбіт/с.

7. Охарактеризуйте параметри, за якими проводиться класифікація каналів передачі даних.

8. Поясніть характеристики та особливості вузькосмугових та широкосмугових каналів.

9. Порівняйте характеристики асинхронних та синхронних каналів передачі даних.

10. Проаналізуйте характеристики комутованих і некомутованих каналів передачі.

11. Охарактеризуйте типи кабельних кабелів (типів фізичного середовища, які використовуються для передачі інформації).

12. Охарактеризуйте типи оптоволоконних кабелів та їх основні характеристики.

13. Проаналізуйте характеристики категорій і типів неекранованої та екранованої скручених пар.

14. Охарактеризуйте методи передачі дискретних даних на фізичному рівні.

15. Поясніть різницю між методами модуляції та маніпуляції.

16. Поясніть поняття модуляції. Охарактеризуйте методи модуляції та маніпуляції.

17. Охарактеризуйте характеристики та особливості потенціальних і імпульсних кодів.

18. Що спільного і в чому різниця між логічним та фізичним кодуванням?

19. Поясніть специфіку використання способів логічного кодування 4B/5B, 5B/6B, 8B/6T та 8B/10B.

20. Поясніть, якою буде послідовність 0110011011011110 при передачі в каналі з використанням коду 4B/5B.

21. Поясніть, якою буде послідовність 0100111011010111 при передачі в каналі з використанням амплітудно-фазової модуляції з 4-ма станами сигналів.

22. Поясніть, якою буде послідовність 0111001000110101 при передачі в каналі з використанням амплітудно-частотної модуляції з 4-ма станами сигналів.

23. Поясніть процес скремблювання та принципи функціонування.

24. Проаналізуйте основні способи дискретного кодування даних.

25. Покажіть, який вигляд буде мати сигнал в каналі при кодуванні методами цифрового кодування NRZ, RZ, 2B1Q вхідної послідовності 0111101100010011.

26. Покажіть, який вигляд буде мати сигнал в каналі при кодуванні методами цифрового кодування 2B1Q, AMI, PE вхідної послідовності 0111001000110101.

27. Охарактеризуйте поняття мультиплексування (ущільнення) інформаційних потоків. Які методи мультиплексування існують?

28. Поясніть принципи та особливості використання часового ущільнення каналів.

29. Поясніть принципи та особливості використання частотного ущільнення каналів.

30. Поясніть принципи та особливості використання ущільнення за довжиною хвилі.

31. Класифікація модемів: основні параметри та типи модемів.

32. Проаналізуйте кабельну систему комп'ютерних мереж.

33. Охарактеризуйте особливості структурованих кабельних систем та їх підсистем.

3 КАНАЛЬНИЙ РІВЕНЬ СУЧАСНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

3.1 Основні функції протоколів канального рівня

Як було розглянуто в попередньому розділі, фізичний рівень відповідає за передавання бітового потоку даних відповідно до фізичних особливостей каналу зв'язку. Біти на фізичному рівні передаються як єдиний нерозділений потік незалежно від того, є помилки при передаванні даних чи вона відбувається коректно.

Канальний рівень працює з даними, тому його часто називають рівнем передавання даних. **Основною задачею** другого рівня еталонної моделі є забезпечення надійного передавання інформації між будь-якими двома (або декількома у разі групового чи широкомовного передавання) вузлами мережі через інформаційний канал. Для цього потік даних, який передається з вищого (мережного) рівня, розбивається на окремі кадри (фрейми), структура і оформлення яких залежать від способу передавання в каналі та типу протоколу. Крім того, в кадрі обов'язково використовуються коди корекції, за допомогою яких виявляються помилки, що виникли при передаванні.

Треба зазначити, що функції та реалізація канального рівня суттєво залежать від того, в яких мережах він використовується: глобальних чи локальних. Оскільки в локальних мережах середовище передавання даних використовується всіма модулями мережі, реалізація канального рівня LAN (Local Area Network) має певні особливості, які будуть розглянуті в підрозділі 3.3 даного розділу.

На рис. 3.1 показана взаємодія двох комунікаційних вузлів мережі, в кожному з яких для кожного каналу зв'язку існують модулі фізичного (1) і канального (2) рівнів та єдиний модуль мережного рівня (3), призначений для визначення напрямку подальшого передавання інформації (процес маршрутизації).

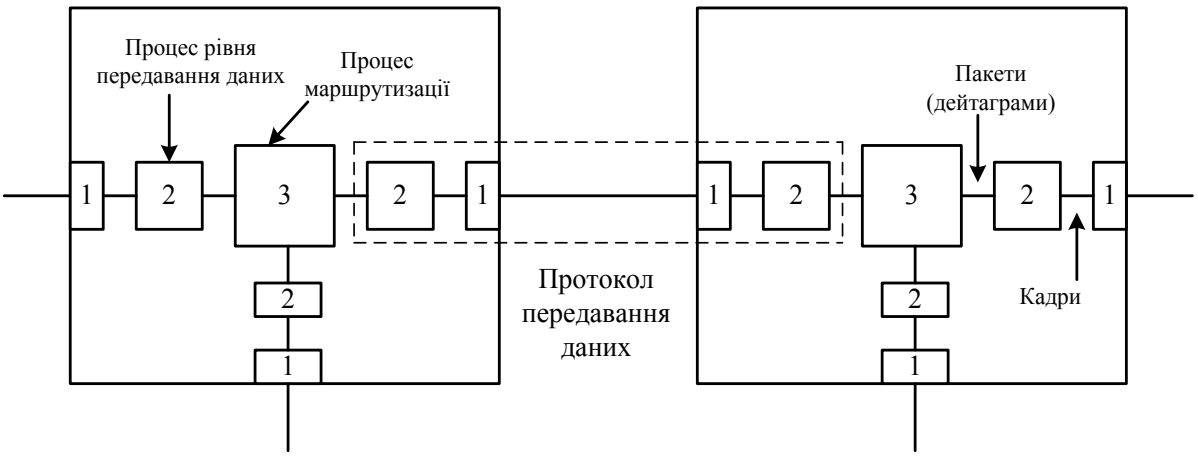


Рисунок 3.1 – Протокол канального рівня (рівня передавання даних)

Пакети або дейтаграми (залежно від способу передавання даних між вузлами мережі) надходять в модуль канального рівня, де інкапсулюються в кадри заданої структури та розміру. Модуль фізичного рівня перетворює отримані кадри в послідовність сигналів такої форми, яка використовується в даному фізичному каналі та забезпечує синхронізацію передавання.

Протокол передавання даних визначає правила передавання та прийому інформації через канал, який з'єднує взаємодійні вузли. Крім того він управляє потоком кадрів, регулює швидкість передавання, а у разі виникнення помилок при передаванні ініціює відповідний алгоритм повторного передавання.

При цьому рівень передавання даних виконує і низку **специфічних функцій**:

- ініціалізація та ідентифікація – обмін між взаємодійними вузлами службовими повідомленнями, які підтверджують відповідно готовність до передавання даних та коректність з'єднання;
- синхронізація – виділення в послідовності бітів, що передаються в каналі, меж символів;
- формування кадру – створення кадру з блока даних, що надходить з мережного рівня, та виділення кадру з послідовності бітів, яка надходить з каналу;
- прийом кадру (або послідовності) з каналу мережі та його передавання в канал;
- прозорість – надання модулям вищих рівнів можливості передавання довільної послідовності бітів та символів;
- керування потоками даних й інтенсивністю передавання кадрів – узгодження швидкості передавання та швидкості прийому потоку кадрів;
- контроль помилок і управління послідовністю передавання кадрів – виявлення помилок, які виникають при передаванні в фізичному каналі, та (за можливості) їх корекція, а також запит на повторне передавання втрачених або пошкоджених кадрів;
- керування каналом – контроль за станом каналу та обробка колізій в мережі, виявлення відмов, відновлення роботи каналу тощо, збір статистики про роботу каналу;
- адресація взаємодійних станцій з використанням фізичних адрес;
- контроль за доступом до фізичного каналу.

Протоколи канального рівня враховують максимальне значення одиниці передавання даних **MTU** (Maximum Transfer Unit), яка визначає максимальний розмір поля даних кадру, що передається в інформаційному каналі. Значення MTU залежить від характеристик каналу та комунікаційного середовища і тому задається в момент встановлення з'єднання або визначається відповідним стандартом мережі. Від значення MTU залежить пропускна спроможність мережі (при збільшенні MTU пропускна спромо-

жність підвищується за рахунок зменшення обсягу надлишкової службової інформації). Значення MTU для найбільш поширених стандартів і протоколів наведені в таблиці 3.1.

Таблиця 3.1 – Значення MTU для основних мережних стандартів

Інтерфейс мережі	Значення MTU, байтів
Максимальне значення MTU	65535
IBM Token Ring (16 Мбайт/с)	17914
IEEE 802.4	8166
FDDI	4500
IEEE 802.5 Token Ring (4 Мбайт/с)	4464
Frame Relay, LAP-F	4096
IEEE 802.11	2272
Ethernet	1500
Point-to-Point Protocol	1500
IEEE 802.3/802.2	1492
Internet (IPv6)	1280
Internet (IPv4)	576
X.25	576
NetBIOS	512
Point-to-Point Protocol (при невеликій затримці)	296

Робота протоколів канального рівня (незалежно від типу і особливостей реалізації) здійснюється в три етапи (фази):

- встановлення з'єднання;
- передавання даних прикладного процесу;
- роз'єднання (анулювання встановленого з'єднання).

Для реалізації цих процедур та доступу до послуг канального рівня використовується сукупність службових повідомлень, які часто називають примітивами.

На етапі **встановлення з'єднання** активізується канал і задаються режими його функціонування та нумерації кадрів. Процедура з'єднання реалізується з використанням таких примітивів:

- «Запит з'єднання», який використовується мережним рівнем для вимоги на організацію логічного каналу і підтверджується канальним рівнем примітивом «Підтримка з'єднання»; при отриманні запиту на з'єднання від віддаленого абонента цей примітив передається мережному рівню у вигляді примітиву «Індикація запиту з'єднання», у відповідь на який видається примітив «Відповідь на запит з'єднання»;

- «Запит активізації», який переводить канальний рівень в активний стан і підтверджується примітивом «Підтвердження активізації»;
- «Запит ідентифікації» кінцевих точок з'єднання, які будуть взаємодіяти при обміні даними;
- «Узгодження параметрів», який визначає якість послуг, що надаються при передаванні;
- «Запит вибору конкретного фізичного з'єднання», який підтверджується примітивом «Підтвердження вибору».

Передавання даних виконується в форматі повідомлення або масиву даних з використанням відповідного алгоритму управління інформаційним каналом. На цьому етапі обов'язково перевіряється коректність переданих даних (відсутність або наявність помилок). Використовуються такі шість примітивів:

- «Запит передавання блока даних» до віддаленого мережного модуля, передавання підтверджується примітивом канального рівня «Підтвердження передавання даних»; блок даних, який надійшов від віддаленого абонента, передається мережному рівню з використанням примітиву «Індикація отримання даних», який підтверджується примітивом «Відповідь на отримання даних»;
- «Запит поточного стану канального рівня»;
- «Запит на термінове передавання блока даних», який використовується для невідкладного (пришвидшеного) передавання блока даних (кадру);
- «Запит переривання» процесу передавання блока даних;
- «Запит управління потоком», що передається від мережного рівня канальному, а при отриманні аналогічного запиту від віддаленого абонента канальним рівнем використовується примітив «Індикація запиту управління потоком»;
- «Запит на перехід до початкового стану», за допомогою якого виконується скидання всіх блоків даних в модулі канального рівня і встановлюється початкова нумерація блоків даних.

Закриття з'єднання може бути коректним (роз'єднання) або аварійним (розрив). Для завершення роботи з'єднання використовуються два примітиви:

- «Запит роз'єднання» створеного каналу, який передається від модуля мережного рівня в канальний;
- «Запит деактивізації» компонентів канального рівня.

Всі фази передавання обов'язково реалізуються в **режимі підтвердження** (режимі видачі квитанції або квоти), тобто і передавання кадрів даних, і запити встановлення різних режимів роботи мають підтверджуватись віддаленою станцією. Загальна процедура взаємодії мережних і канальних рівнів двох станцій наведена на рис. 3.2.

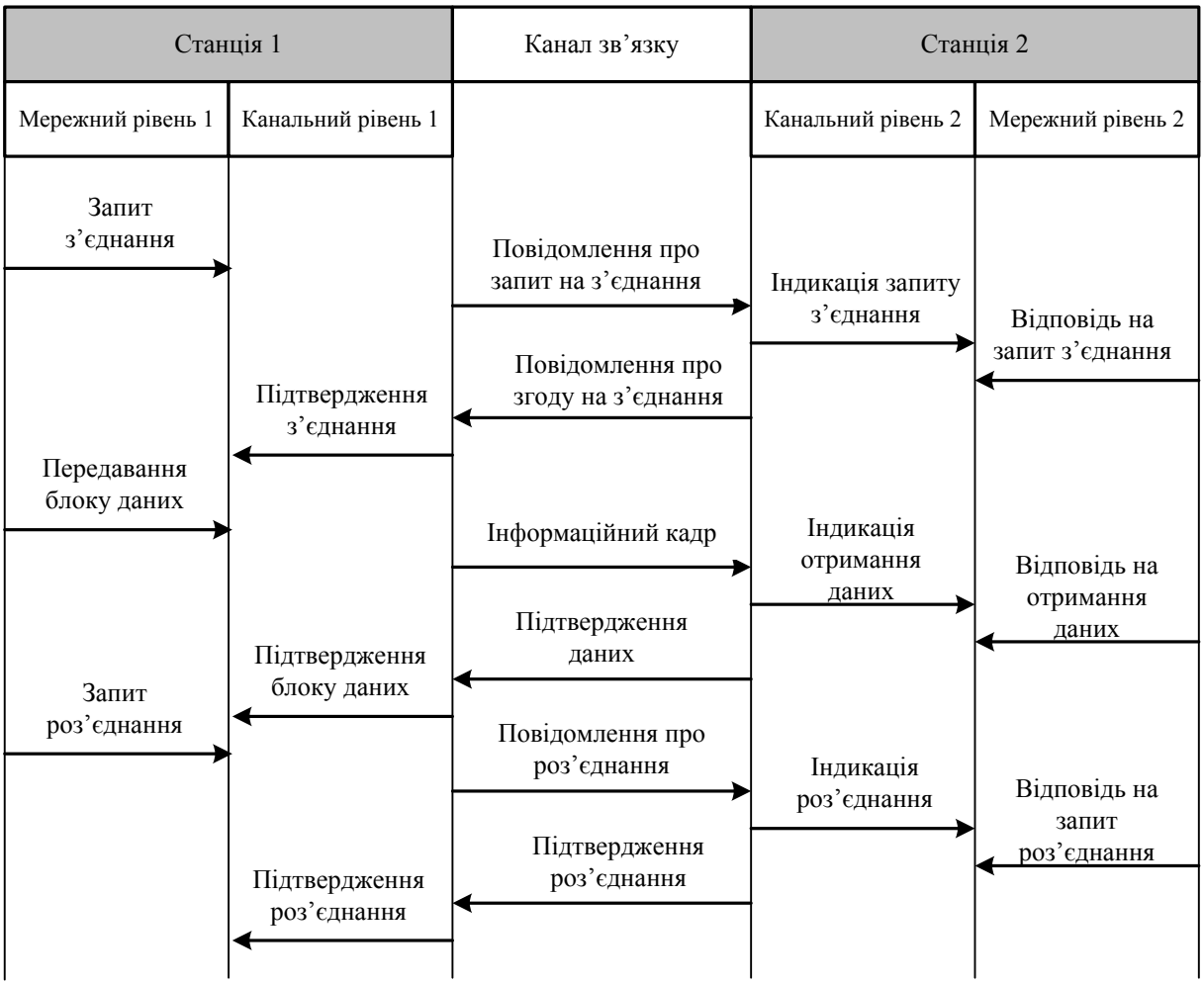


Рисунок 3.2 — Послідовність взаємодії станцій на канальному рівні

В станції 1, в якій сформовано повідомлення для передавання в станцію 2, мережний модуль відправляє канальному рівню запит на встановлення з'єднання. На основі цього запиту канальний рівень станції 1 формує службове повідомлення про встановлення з'єднання, формат якого визначається відповідним протоколом канального рівня, та передає його в віддалену станцію. При отриманні цього запиту канальний рівень віддаленої станції 2 передає мережному рівню примітив «Індикація запиту з'єднання». У відповідь на це аналізується можливість і доцільність встановлення з'єднання з такими параметрами і рішення про це підтверджується примітивом «Відповідь на запит з'єднання», яка може бути як позитивною, так і негативною. У разі позитивної відповіді канальний рівень формує службове повідомлення і передає його через фізичний рівень і середовище передачі в станцію 1. Ця згода надходить на мережний рівень станції 1 в формі примітиву «Підтвердження з'єднання», після чого починається передавання інформаційного потоку, який обов'язково підтверджується відповідними примітивами. Фаза роз'єднання виконується аналогічно фазі з'єднання, але з видачею інших примітивів.

3.2 Класифікація протоколів канального рівня

Розрізняють **різнорангові (несиметричні)** протоколи (типу первинний/вторинний або головний/підлеглий) та **однорангові (симетричні)** протоколи (однорівневі або рівнорангові).

В **однорангових протоколах** передбачається однаковий статус та рівні права всіх станцій в каналі. При такому підході для станцій забезпечуються однакові можливості використання ресурсів каналу, однак управління каналом забезпечується децентралізовано. Такі протоколи використовуються в локальних мережах будь-якої топології і будуть розглянуті в підрозділі 3.3.

У глобальних та регіональних мережах зазвичай використовуються **різнорангові протоколи**, які передбачають наявність в каналі станцій з різним статусом і різними правами. Виділяють станції трьох типів:

- головні (первинні);
- вторинні (підлеглі);
- комбіновані.

Головна станція, яка може бути лише одна в каналі, забезпечує керування каналом передавання, формуючи команди, що є обов'язковими для всіх вторинних модулів каналу. При цьому саме ця станція відповідає за організацію передавання потоку кадрів, їх відновлення після пошкодження й забезпечує керування та коректне функціонування всього каналу передавання. Первинна станція відповідає також за організацію сеансу зв'язку з кожною станцією, яка підключена до каналу.

Вторинна станція є підлеглою щодо первинної і реагує на команди від неї формуванням відповідей. Така станція не відповідає за керування каналом і підтримує лише один сеанс зв'язку з головною.

Комбіновані станції одночасно об'єднують функції як первинної, так і вторинної станцій. Формує і передає в канал команди та відповіді, а також отримує команди й відповіді від іншої комбінованої станції, з якою підтримує сеанс.

Залежно від способу передавання розрізняють **асинхронні** та **синхронні** протоколи передавання.

Асинхронні протоколи забезпечують посимвольне передавання даних у старт-стопному режимі, причому символи видаються в канал в довільний момент часу залежно від їх надходження від станції (ці процедури описані в підрозділі 2.2 (див. рис. 2.3 та 2.4) даного підручника).

Синхронні протоколи канального рівня забезпечують передавання послідовності символів або кадрів чіткої структури і можуть бути двох типів:

- байт-орієнтованими;
- біт-орієнтованими.

Байт-орієнтований протокол (кодозалежний, знакозалежний) забезпечує передавання даних в інформаційному каналі у вигляді послідовності байтів. Крім інформаційних байтів передаються також керівні та службові

байти, які можуть зустрітися в будь-якому місці послідовності. Такий тип протоколу більше підходить для станцій, тому що він орієнтований на обробку даних, поданих у вигляді двійкових байтів. Для комунікаційного середовища використання байт-орієнтованого протоколу призводить до необхідності передачі додаткових (службових) байтів, що знижує загальну пропускну спроможність каналу зв'язку. Прикладами байт-орієнтованих протоколів, які знайшли найбільше використання, є:

- BSC (Binary Synchronous Communication);
- SLC (Synchronous Link Control);
- DDCMP (Digital Data Communication Message Protocol).

Біт-орієнтований протокол (кодонезалежний, знаконезалежний) забезпечує передавання даних в інформаційному каналі у вигляді послідовності бітів. Інформація передається у вигляді кадрів чіткої структури, кожний біт (або їх сукупність) якої призначений для передавання строго визначених даних. Тому керівні біти можуть зустрітися у чітко визначених місцях кадру. Такі протоколи передачі потенційно забезпечують більшу швидкість (порівняно з байт-орієнтованими) і більше підходять для комунікаційного середовища, що обумовлює їх широке використання в сучасних комп'ютерних мережах. Прикладами біт-орієнтованих протоколів є:

- SDLC (Synchronous Data Link Control Protocol);
- HDLC (High-Level Data Link Control Protocol);
- LAP (Link Access Procedures);
- LAPB (Balanced Link Access Procedures);
- ADCCP (Advanced Data Communication Control Procedures);
- BDLC (Burroughs Data Link Control);
- UDCL (Univac Data Link Control).

Прозорість каналу. При передаванні потоку даних через інформаційний канал виникає необхідність розділення символів на інформаційні та керівні (службові), тобто виділення з загального потоку тих символів, які використовуються або для передавання даних, або для керування процесом обробки даних. Суть проблеми полягає в тому, що одні й ті ж самі символи можуть використовуватись як для передавання даних повідомлення, так і в командах, які керують процесом передавання та визначають функціонування інформаційного каналу. Тому при передаванні послідовності даних потрібно вживати спеціальних заходів, що реалізовані в канальних протоколах, для ідентифікації керівних та інформаційних бітів. Для цього, зазвичай, перед керівними символами передаються певні байти (або біти), які і повідомляють модулю-отримувачу, що ці символи використовуються для керування і не належать до прикладних сервісів. Така процедура передавання називається процедурою передавання **прозорих даних**, а сам канал, який забезпечує передавання таких даних, – **прозорим каналом.**

Байт-орієнтовані протоколи. Одним з найбільш відомих і поширених байт-орієнтованих протоколів є протокол двійкового синхронного зв'язку

BSC (Binary Synchronous Communication або скорочено Bisync протокол), який розроблено фірмою IBM.

Протокол BSC є напівдуплексним протоколом і забезпечує передавання комутованими та некомутованими каналами з організацією point-to-point та point-to-multipoint. Структура послідовності, що передається в канал, і розташування в ній керівних символів не фіксуються, тобто є довільними.

Оскільки даний протокол є кодозалежним, формат керівних символів для різних типів кодувань є різним, наприклад, в коді ASCII символ SYN дорівнює 0010110, а в коді EBCDIC – 00110010. Крім того, немає чіткої структури послідовності символів, яка передається в канал, тобто керівні (службові) символи можуть зустрітися в будь-якому місці передаваної послідовності, а прикладний процес може сформувати такі символи даних для передавання, байти яких збігаються за кодуванням з керівними символами. Якщо не вжити відповідних заходів, це призведе до некоректного розпізнавання послідовності символів приймальною станцією. Тому при використанні протоколів даного класу необхідно робити відповідні дії для ідентифікації керівних та інформаційних символів. В протоколі BSC з цією метою використовується символ DLE, який вставляється при передаванні перед кожним службовим символом. Ця процедура називається **стаффінгом символів** (stuffing або байт-стаффінгом).

На сьогоднішній день протоколи цієї групи використовуються все менше і поступово замінюються біт-орієнтованими протоколами. Байт-орієнтовані протоколи використовуються при передаванні довгих блоків даних, наприклад, при передаваннітекстових файлів для додатків, які не вимагають високої швидкості, а також при посимвольному передаванні, наприклад, для зв’язку комп’ютера з принтером або терміналом.

Тому зупинимось детальніше лише на функціонуванні біт-орієнтованих протоколів.

Біт-орієнтовані протоколи. Першим синхронним біт-орієнтованим протоколом був протокол керування синхронним каналом передавання даних **SDLC** (Synchronous Data-Link Control), розроблений IBM для мереж SNA (Systems Network Architecture). Цей протокол став основою для протоколу високорівневого управління каналом **HDLC** (High-level Data Link Control), який розроблено Міжнародною організацією зі стандартизації ISO (International Organization for Standardization). І хоча HDLC не реалізує деякі характеристики протоколу SDLC, саме протокол HDLC є базовим для розробки великої кількості модифікацій для різних типів мереж.

Протокол високорівневого керування HDLC одержав найбільш широке використання. HDLC має декілька необов’язкових можливостей, які об’єднують характеристики рівнорангових і різнорангових методів, що веде до ліквідації команд вибору та зменшення команд опитування.

Протокол підтримує напівдуплексне та дуплексне передавання по комутованих і некомутованих каналах з організацією point-to-point та point-

to-multipoint. При цьому розрізняють три типи статусу станцій: головна, вторинна та комбінована.

Взаємодіяти станції можуть в одному з **трьох режимів**:

- нормальної відповіді (відгуку) **NRM** (Normal Response Mode);
- асинхронної відповіді (відгуку) **ARM** (Asynchronous Response Mode);
- асинхронний збалансований **ABM** (Asynchronous Balance Mode).

Режим **NRM** передбачає, що вторинна станція може розпочати передавання тільки після одержання явного підтвердження від первинної. Після одержання команди (дозволу) від головної станції вторинна починає передавання відповіді, яка може містити дані. В одному циклі вторинна станція може передавати один або декілька кадрів. Після передачі останнього кадру вторинна станція має знову очікувати на явний дозвіл перш ніж розпочати передавання наступних кадрів. Як правило, цей режим використовується вторинними станціями в конфігураціях каналу передавання даних point-to-multipoint.

У режимі асинхронної відповіді **ARM** вторинній станції дозволяється у разі вільного каналу (зазвичай, у стані спокою) ініціювати передавання без отримання явного дозволу від первинної станції. Цей режим дає велику гнучкість роботі вторинної станції. Можуть передаватися один або декілька кадрів даних або керівна інформація, яка може змінювати статус вторинної станції. Як правило, такий режим використовується для керування станціями, що з'єднані у кільце, або ж у багатоточкових з'єднаннях з послідовним опитуванням. В обох випадках вторинна станція може отримати дозвіл від іншої вторинної станції і у відповідь на нього розпочати передавання. Такий режим зменшує накладні витрати та час передавання, але ускладнює управління інформаційним каналом.

Режим **ABM** використовують комбіновані станції, кожна з яких може ініціювати передавання без отримання попереднього дозволу від іншої комбінованої станції. Цей режим найбільш часто використовується на практиці, оскільки забезпечує двосторонній обмін потоками даних між станціями.

В процесі взаємодії станції можуть знаходитись в одному з трьох логічних станів:

- ініціалізації **IS** (Initialization State);
- передавання інформації **ITS** (Information Transfer State);
- логічного роз'єднання **LDS** (Logical Disconnect State).

Ініціалізація (з'єднання) використовується для передавання команд керування на віддалену вторинну або комбіновану станцію, а також для обміну параметрами між віддаленими станціями в каналі передавання.

При **передаванні інформації** станції будь-якого статусу головна, вторинні (або комбіновані) виконують передавання та прийом інформації

користувача. При цьому передавання здійснюється в режимах NRM, ARM і ABM.

Реалізація **логічного роз'єднання** полягає у видачі запиту на роз'єднання і отриманні підтвердження цього роз'єднання.

У протоколі HDLC передбачено **три способи конфігурування** каналу при його використанні первинною, вторинною або комбінованою станціями:

- незбалансована конфігурація **UN** (Unbalanced Normal);
- симетрична конфігурація **UA** (Unbalanced Asynchronous);
- збалансована конфігурація **BA** (Balanced Asynchronous).

Незбалансована конфігурація забезпечує роботу однієї головної станції та однієї або більшої кількості вторинних станцій в конфігурації point-to-point та point-to-multipoint, з напівдуплексним і дуплексним передаваннями по комутованих і некомутованих каналах зв'язку. Ця конфігурація називається незбалансованою, оскільки головна станція відповідає за керування кожною вторинною станцією і за виконання команд встановлення режиму.

Симетрична конфігурація передбачає функціонування двох незалежних незбалансованих конфігурацій каналу point-to-point. Кожна станція має одразу два статуси: головної та вторинної, і тому логічно розглядається як дві станції. Незважаючи на те, що станція може працювати як у статусі головної, так і вторинної станцій, що є самостійними логічними об'єктами, реальні команди і відповіді мультиплексуються в один фізичний канал. Така конфігурація застосовувалась в перших версіях протоколу.

Збалансована конфігурація складається з двох комбінованих станцій в каналі point-to-point, при цьому реалізуються дуплексне і напівдуплексне передавання по комутованих і некомутованих каналах зв'язку. Оскільки комбіновані станції мають однаковий статус у каналі й можуть несанкціоновано формувати і передавати свій трафік, кожна з них несе однакову відповідальність за керування каналом.

HDLC забезпечує передавання між модулями кадру, структура якого строго фіксована. **Кадром** (фреймом, frame) називається незалежний об'єкт даних, який передається як єдине ціле через інформаційний канал. Структура кадру протоколу HDLC встановлена стандартом ISO 3309 і наведена на рис. 3.3. При цьому використовуються два способи нумерації кадрів: **нормальний** (з нумерацією кадрів за модулем 8) і **розширений** (з нумерацією кадрів за модулем 128). Структура цих кадрів наведена, відповідно, на рис. 3.3 та 3.4. Режим розширеної нумерації використовується в довгих каналах зв'язку, наприклад, супутникових, які забезпечують передавання не безпосередньо між станціями, а через третій модуль.

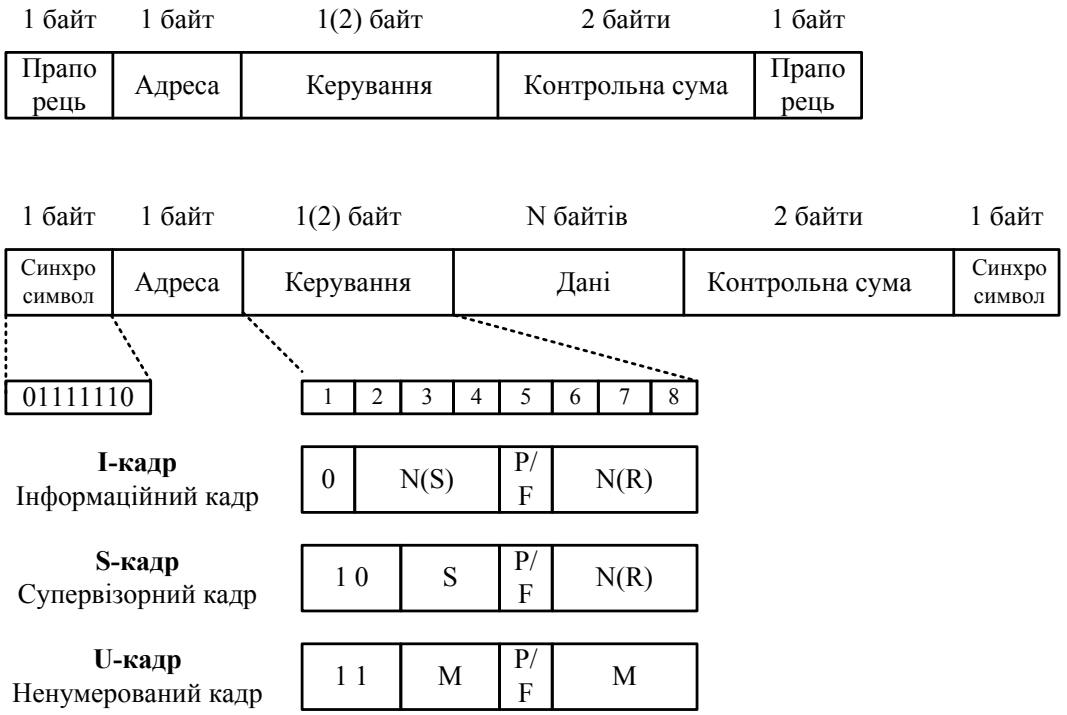


Рисунок 3.3 – Формат кадру протоколу HDLC (нормальна нумерація)

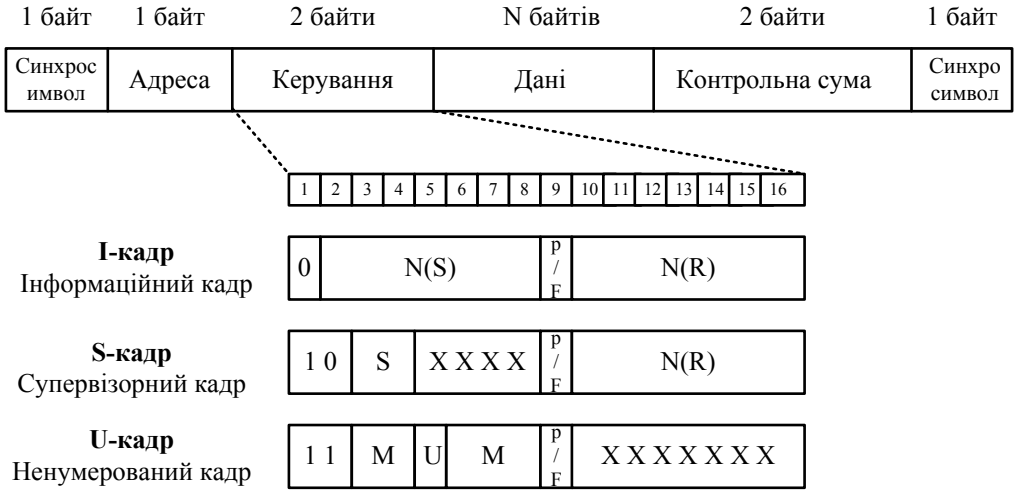


Рисунок 3.4 – Структура кадру HDLC при розширеній нумерації

Кожний кадр містить відкривальний та закривальний **синхросимволи**, які мають значення 7E та дозволяють станції, що отримує кадр, визначити початок і закінчення фрейму. Іноді закривальний синхросимвол кадру може об'єднуватись з відкривальним синхросимволом наступного кадру, визначаючи, таким чином, його початок.

Поле **Адреса** ідентифікує головну або вторинну станцію, яка бере участь в обміні. При незбалансованій конфігурації каналу в цьому полі записується адреса вторинної станції, а при збалансованій – кадр команди

містить адресу станції-отримувача, а кадр відповіді – адресу станції, яка передає інформацію.

Керування каналом передавання даних визначає типи кадру, команди або відповіді, а також порядкові номери кадрів для керування їх передаванням.

Поле **Дані** містить дані користувача, які передаються між станціями. Переважна кількість керівних кадрів не має цього поля.

Контрольні біти призначені для контролю коректності передавання. Зазвичай використовується циклічний код, генерувальний поліном якого визначається рекомендаціями ITU-T V.41.

Виділяють три типи кадрів (табл. 3.2), які використовуються протоколом HDLC: інформаційний I-кадр (I-frame), супервізорний S-кадр (S-frame) та ненумерований U-кадр (U-frame).

Таблиця 3.2 – Типи кадрів протоколу HDLC

Найменування	Мнемоніка	Код поля (S або M)	Функція
Інформаційний кадр	I		К/В
Супервізорні кадри			
Готовність до прийому	RR	00	К/В
Неготовність до прийому	RNR	10	К/В
Відмова від прийому	REJ	01	К/В
Вибіркова відмова від прийому	SREJ	11	К/В
Ненумерована інформація	UI	00000	К/В
Установлення нормального відгуку (Set Normal Regime Mode)	SNRM	00001	К
Розрив з'єднання, роз'єднання (Disconnect)	DISC	00010	К
Запит роз'єднання (Request Disconnect)	RD	00011	В
Ненумерований запит передавання, опитування (Unnumbered Poll)	UP	00100	К
Ненумероване підтвердження (Unnumbered Acknowledgement)	UA	00110	В
Тестування системи передавання даних	TEST	00111	В
Установлення режиму ініціалізації (Set Initialization Mode)	SIM	10000	К
Запит режиму ініціалізації (Request Initialization Mode)	RIM	100100	В
Відмова від кадру (Frame Reject)	FRMR	10001	К/В
Установлення режиму асинхронного відгуку (Set Asynchronous Acknowledgement Mode)	SARM	11000	К
Перезапуск	RSET	11001	К
Установлення SARM з розширеною нумерацією	SARME	11010	К
Установлення SNRM з розширеною нумерацією	SNRME	11011	К
Установлення асинхронного збалансованого режиму	SAMB	11100	К
Обмін ідентифікаторами (Exchange Identifier)	XID	11101	К/В
Встановлення SABM з розширеною нумерацією	SAMBE	11110	К
Ініціалізація режиму логічного роз'єднання (Disconnect Mode)	DM	11000	В

Інформаційні кадри використовуються для передавання блоків даних користувача, які записані в полі **Дані**.

Супервізорні кадри призначені для керування процесом передавання інформаційних кадрів, керування потоком, а також для формування запитів на повторну передавання кадрів, які пошкоджені і містять помилки. Розрізняють супервізорні кадри, типи яких визначаються бітами S:

- готовність до прийому RR (Receive Ready);
- неготовність до прийому RNR (Receive Not Ready);
- відмова від прийому REJ (Reject);
- вибіркова відмова від прийому SREJ (Selective Reject).

Ненумеровані кадри використовуються для встановлення режимів обміну інформацією між взаємодійними станціями та способу нумерації кадрів, коректного завершення встановлених режимів, обміну параметрами тощо. Всього в базовому стандарті протоколу HDLC визначено 18 типів U-кадрів, які визначаються бітами M. Ненумеровані кадри можна розділити на групи відповідно до функцій, які вони виконують:

- команди встановлення режиму: SNRM, SARM, SABM, (SNRME, SARME, SABME – для розширеної нумерації кадрів), SIM, RIM, DISC;
- команди передавання інформації: UI, UP;
- команда перезапуску (відновлення роботи): RSET;
- інші команди: XID, TEST, DM, UA, FRMR, RD.

Біт опитування/закінчення P/F (poll/final) обробляється тільки у випадку, коли він встановлений в 1. Коли біт P/F використовується головною станцією, він називається бітом P, коли вторинною – бітом F. Головна станція використовує біт P у режимі NRM для опитування, а вторинна – біт F в останньому I-кадрі відповіді. В режимах ARM и ABM біти P/F використовуються для отримання термінової відповіді на команду. У випадку коли станція отримала команду з встановленим бітом опитування P=1, вона обов’язково має сформувати відповідь з бітом закінчення F=1. В іншому випадку буде зроблено висновок про некоректну обробку команди (тобто помилку).

Синхронізація станцій та кодонезалежність HDLC. Для синхронізації взаємодії станцій каналу визначено декілька службових сигналів: аварійного завершення, спокою, міжкадрового часового заповнення.

Сигнал аварійного завершення (AЗ) складається з послідовності одиниць, кількість яких від семи до чотирнадцяти, і записується в кінці кадру. Станція передає цей сигнал при виникненні нештатної ситуації, яка вимагає відновлення. За сигналом AЗ можуть передаватися синхросигнали для підтримки каналу в активному стані.

Сигнал спокою (CC) означає, що канал перебуває в стані спокою, являє собою послідовність п’ятнадцяти або більшої кількості одиниць і також записується в кінці кадру. Найчастіше використовується в напівдупле-

ксному каналі для зміни напрямку передавання на протилежний. Сигнал спокою вимагає від віддаленої станції переналаштування вихідного інтерфейсу і видачі відповідної реакції на прийнятий кадр.

Для підтримки каналу в активному стані між кадрами передається неперервна послідовність байтів синхросимволів, яка називається сигналом **міжкадрового часового заповнення.** Ця послідовність передавання може бути як з побайтовим передаванням синхросимволів, так і з об'єднанням останнього нульового біта попереднього синхросимволу з першим нульовим бітом поточного синхросимволу. Тобто, в каналі може передаватися як послідовність 011111100111111001111110, так і 01111110111111101111110.

Оскільки протокол HDLC є кодонезалежним, структура його кадру є строго фіксованою і не залежить від конкретного коду. Тому немає необхідності розрізняти інформаційні та керівні символи. Єдиним варіантом внесення некоректності в послідовність передавання є випадок, коли прикладний процес формує дані для передавання, які збігаються з синхросимволом, тобто 01111110. Для нейтралізації такої ситуації виконується процедура **біт-стаффінгу** (вставляння біта, bit staffing), яка передбачає, що при передаванні кадру в усі поля між відкривальним і закривальним синхросимволами після п'яти послідовних одиниць, незалежно від значення наступного біта, вставляється нульовий біт (рис. 3.5), а при прийомі кадру на віддаленій станції цей вставлений нульовий біт відкидається. Ця процедура аналогічна процедурі байт-стаффінгу, але значно більш економна і зовсім несуттєво зменшує швидкість передавання.

Рисунок 3.5 – Процедура біт-стаффінгу в протоколі HDLC

3.3 Процедури передавання даних в інформаційному каналі за допомогою протоколу HDLC

Протокол HDLC керує інформаційним каналом шляхом використання спеціальних керівних кадрів, за допомогою яких передаються команди. Інформаційні кадри послідовно циклічно нумеруються (за модулем 8 або модулем 128). Детальніше процедура передавання виглядає так.

Процедура з'єднання. Головна станція задає режим роботи каналу за допомогою формування відповідного ненумерованого кадру (SNRM, SARM, SABM – для нормальної нумерації кадрів або SNRME, SARME, SABME – для розширеної нумерації). Після відправлення в канал цього кадру запускається процедура тайм-ауту, до закінчення якої необхідно отримати відповідь від вторинної станції. Якщо цього не сталося, головна станція повторно відправляє команду в канал. Кількість цих повторів задається і, зазвичай, дорівнює трьом. Якщо і після цього немає відповіді від вторинної станції, головна станція переходить в режим тестування каналу. Вторинна станція в разі неготовності до взаємодії відповідає ненумерованим кадром «Запит роз'єднання RD», а в разі готовності – кадром «Ненумероване підтвердження UA». Прийом цього кадру завершує процедуру встановлення режиму та ініціалізації каналу.

Процедура передавання даних. Після встановлення з'єднання станція-відправник передає дані за допомогою інформаційних I-кадрів, які вимагають перевірки коректності їх передавання та підтвердження прийому від віддаленої станції. За необхідності передавання інформаційного потоку іноді використовують кадри «Ненумерована інформація UI», які містять інформацію користувача, але, на відміну від прийому I-кадрів, не можуть бути повторені при їх пошкодженні або втраті.

При передаванні інформаційного кадру вказується його порядковий номер N(S) і номер кадру N(R), який станція готова відправити в наступному циклі. **Копії відправлених кадрів обов'язково мають зберігатися у вихідному буфері станції-відправника до отримання позитивного підтвердження від станції-отримувача,** яка перевіряє отриманий кадр на наявність помилок за допомогою циклічного коду. Кадр, прийнятий без помилок, передається для подальшої обробки на третій, мережний рівень. Якщо помилки, які виникли при передаванні, не можуть бути виправлені наявними ресурсами, ініціюється його повторне передавання. Існує декілька алгоритмів керування передаванням на канальному рівні:

- зупинки та очікування SAW (Stop And Wait);
- з поверненням на N кадрів GBN(Go-Back-N);
- вибіркового повторення SR (Selective Repeat).

Тобто, ці способи корекції пошкоджених кадрів в комп'ютерних мережах базуються на повторному передаванні інформаційного кадру у випадку, коли цей кадр втрачений і не отриманий адресатом, або станція-отримувач виявила в ньому наявність помилок. При реалізації алгоритмів керування інформаційним каналом реалізовано механізм автоматичного запиту на повторне передавання **ARQ** (Automatic Repeat Quest).

Процедура з зупинками та очікуванням SAW. Такий алгоритм керування (рис. 3.6) передаванням в інформаційному каналі передбачає, що після відправлення інформаційного кадру станція-відправник (станція А) переходить в режим очікування до отримання кадру підтвердження від від-

даленої станції (станції В). У разі коректного передавання станція В формує і видає в канал супервізорний кадр *Si* позитивного підтвердження «Готовність до прийому RR» (аналог символу АСК в протоколі BSC), який містить номер кадру N(R), на який станція В очікує в наступному циклі передачі. При виникненні помилки в інформаційному кадрі формується кадр негативної квитанції «Неготовність до прийому RNR» (аналог символу NAK в протоколі BSC), а поле N(R) містить номер кадру з помилкою, який необхідно повторно передати.

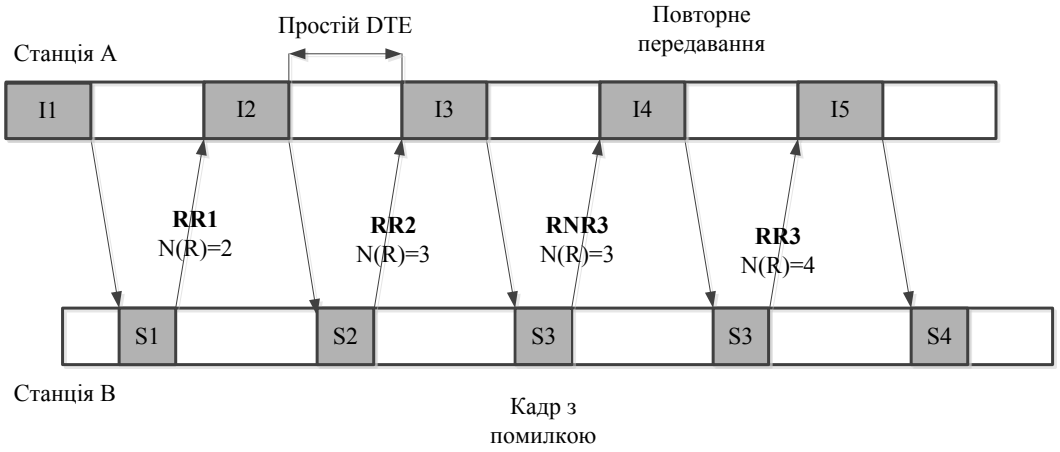


Рисунок 3.6 – Процедура передавання SAW

Головним недоліком даного протоколу є його неефективність: невелика завантаженість каналу і значний час простою станцій. Тому, зазвичай, використовується блокове передавання декількох інформаційних кадрів з видачею одного кадру підтвердження для всього блоку. Наприклад, після передачі блоку з семи інформаційних кадрів (I1-I7) формується позитивна квитанція RR з номером N(R) = 8, яка свідчить, що всі ці кадри прийнято без помилок. Блокове передавання забезпечує більш високу пропускну спроможність каналу.

Процедура з поверненням на N кадрів GBN. Алгоритм GBN (як і алгоритм вибіркового повторення) використовується в дуплексних каналах. При цьому (рис. 3.7) інформаційні кадри передаються один за одним без очікування підтвердження на попередні кадри (тобто реалізується потокове передавання).

Прийом кожного I-кадру підтверджується супервізорним кадром S типу «Відмова від прийому REJ», у полі N(R) якого міститься номер кадру з помилкою. При одержанні такого кадру (або при закінченні тайм-ауту встановленого терміну) непідтверджений кадр і всі наступні, які вже були передані, передаються в канал знову. Така процедура значно збільшує пропускну спроможність каналу порівняно з попереднім. Але необхідність повторного передавання всіх кадрів, починаючи з пошкодженого, призводить до зниження ефективності, особливо в довгих каналах, час отримання

підтвердження досить значний. Але така процедура забезпечує передавання кадрів без порушення їх послідовності, тому адресат не повинен виконувати додаткових дій з сортування прийнятих кадрів.

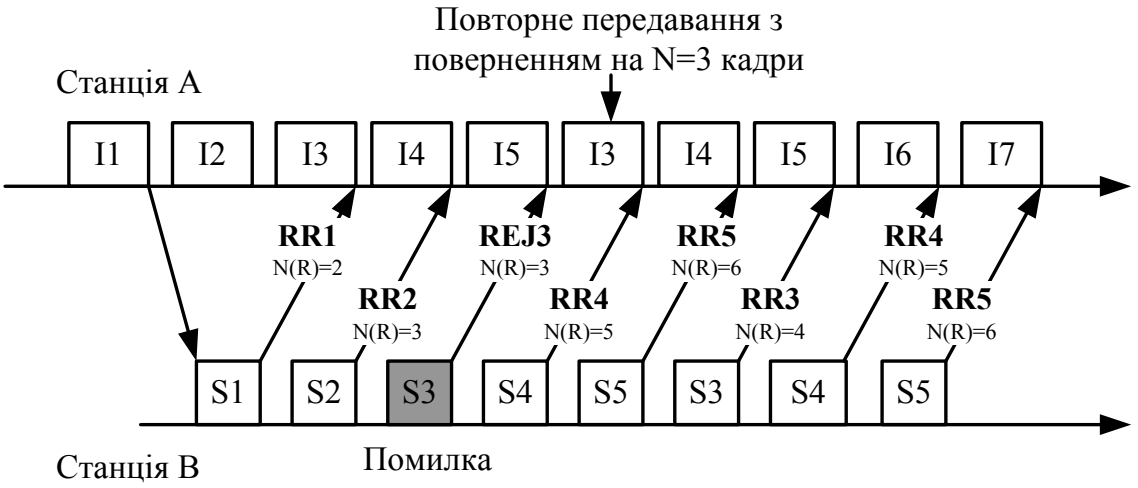


Рисунок 3.7 – Процедура передавання з поверненням на N кадрів

Алгоритм вибіркового повторення SR. При реалізації такої процедури також реалізується потокове передавання (рис. 3.8). При отриманні негативної квитанції (супервізорний кадр «Вибіркова відмова від прийому SREJ») повторно передається тільки той кадр, який був прийнятий з помилкою. Така процедура призводить до порушення послідовності передавання кадрів, в результаті адресат отримує потік кадрів з їх переплутаним порядком. Тому в модулі-отримувачі необхідно виконати додаткове сортування кадрів для створення цілісного повідомлення, що призводить до збільшення обсягу пам’яті вхідного буфера.

Перевагою даного алгоритму є відсутність дублювання при повторному передаванні правильно переданих кадрів, що веде до збільшення пропускної спроможності інформаційного каналу.

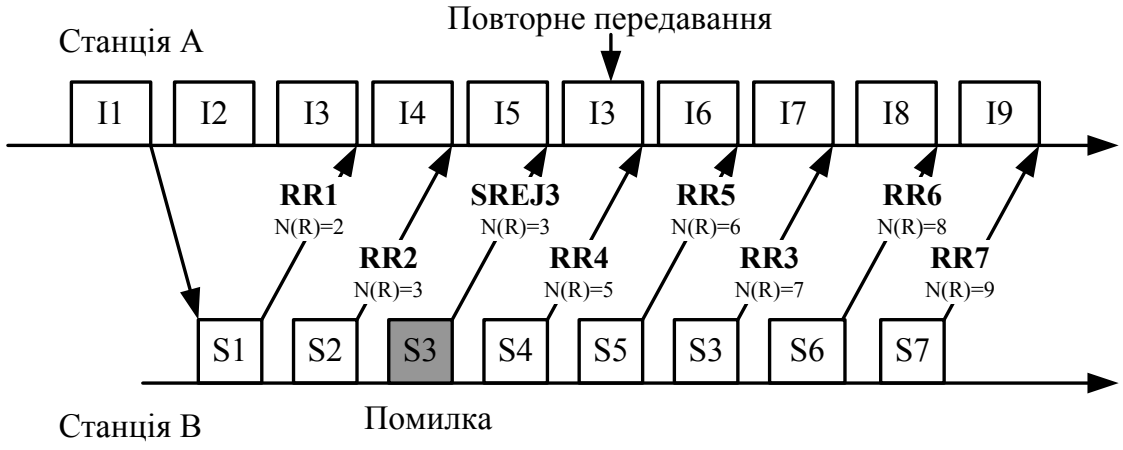


Рисунок 3.8 – Процедура передавання вибіркового повторення SR

Окремим випадком алгоритмів з видачею квитанцій є **алгоритм ковзного вікна (sliding window)** чи **алгоритм вікна зі змінними розмірами (sizeable window)**. При реалізації цього алгоритму в процесі встановлення з'єднання станції домовляються про кількість кадрів, які можуть бути передані в канал без отримання квитанції, тобто про розмір вікна (рис. 3.9). В процесі передавання розмір вікна постійно змінюється, в зв'язку з чим алгоритм і отримав таку назву. У випадку, коли розмір вікна стане дорівнювати нулю, передавання припиняється і очікується на прийом квитанції від віддаленої станції.

Розмір вікна встановлюється з урахуванням обсягу пам'яті вхідного і вихідного буферів взаємодійних станцій, а також способу нумерації кадрів (для нормальної нумерації – максимум 8 кадрів, для розширеної – максимум 128 кадрів).

Треба враховувати і те, що всі кадри потоку, для яких ще не отримано підтвердження, потрібно зберігати у вихідному буфері, тому розмір вікна не слід встановлювати занадто великим.

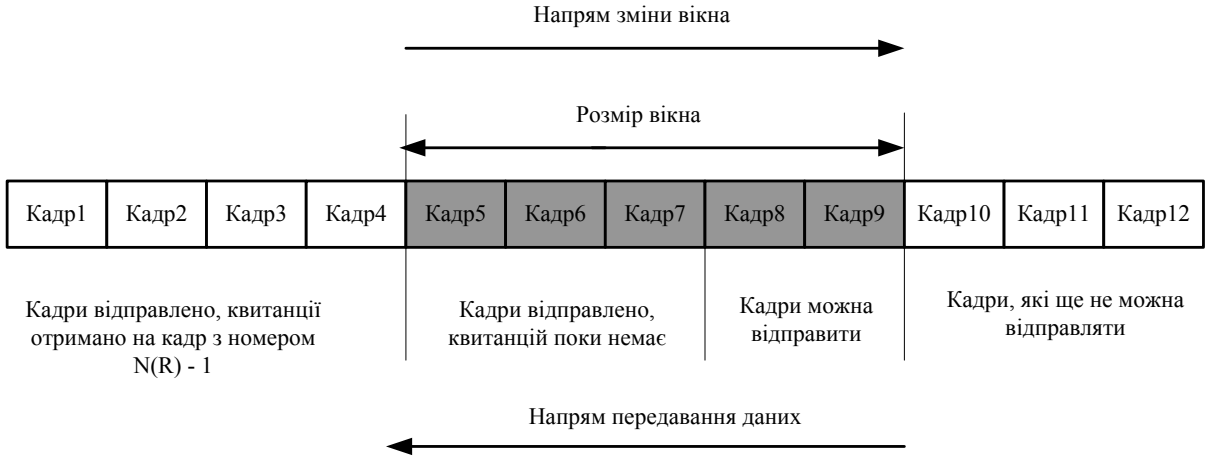


Рисунок 3.9 – Реалізація алгоритму ковзного вікна

При використанні алгоритму ковзного вікна ефективність передавання даних між взаємодійними станціями суттєво залежить як від розміру вікна, так і від значення тайм-ауту очікування квитанції від віддаленої станції. Тому при використанні в мережі надійних каналів зв'язку для підвищення пропускної спроможності мережі розмір вікна необхідно збільшувати, що приведе до зменшення пауз між відправленими кадрами. Якщо в мережі використовуються переважно ненадійні канали зв'язку, розмір вікна необхідно зменшувати, оскільки часті пошкодження та втрати кадрів призводять до різкого збільшення кількості кадрів, які необхідно повторно передавати в мережу, що, в свою чергу, призведе до зменшення загальної корисної пропускної спроможності мережі.

Тобто, вибір тайм-ауту залежить не стільки від надійності каналів зв'язку і модулів мережі, скільки від затримок передавання кадрів в мере-

жі, які залежать від багатьох причин і постійно змінюються. Тому при використанні алгоритму ковзного вікна значення тайм-ауту і розміру вікна обираються адаптивно, залежно від поточного стану мережі.

Аналогічний алгоритм управління використовується і протоколом TCP з єдиною відмінністю: розмір вікна вираховується не в кількості кадрів, а в кількості байтів, які можуть бути передані без підтвердження.

Процедура роз'єднання. Для роз'єднання каналу зв'язку, створеного між взаємодійними станціями, головна станція формує і видає в канал команду «Роз'єднання DISC», яка підтверджується відповіддю «Ненумероване підтвердження UA». Запит на роз'єднання може формуватися і з боку вторинної станції видачею кадру «Запит роз'єднання RD», який також підтверджується кадром UA.

Сімейство протоколів HDLC. Протокол HDLC є базовим протоколом для розробки великої кількості модифікацій і версій, які враховують особливості різних типів мереж і вимог виробників мережного обладнання та користувачів. До сімейства протоколів HDLC належать такі протоколи:

- LLC (Logical Link Control) – протоколи керування логічним каналом для локальних мереж;
- PPP (Point-to-Point Protocol) – сукупність протоколів, яка охоплює протокол керування зв'язком LCP (Link Control Protocol), протокол керування мережею NCP (Network Control Protocol), протоколи аутентифікації PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), багатоканальний протокол MLPPP (Multilink PPP);
- прикладами та модифікаціями протоколу є PPPoE (Point-to-Point Protocol over Ethernet), який використовується при підключенні до мережі Ethernet, PPPoA (Point-to-Point Protocol over ATM), який використовується для підключення до мереж з асинхронним доступом ATM тощо;
- LAP (Link Access Procedure) – процедура доступу до каналу є однією з перших і використовувалась в каналах з асинхронною відповіддю та збалансованою конфігурацією;
- LAPB (Link Access Procedure Balance) – застосовується в мережах X.25;
- LAPD (Link Access Procedure D-channel) – застосовується в цифрових мережах з інтегральним доступом ISDN (Integrated Services Digital Network);
- V.120 – застосовується в цифрових мережах з інтегральним доступом ISDN;
- LAPM (Link Access Procedure for Modems) – протокол синхронного передавання в комутованих телефонних мережах загального використання PSTN (Public Switched Telephone Network);

- LAPX (Link Access Procedure eXtention) – напівдуплексний варіант HDLC використовується в термінальних системах і стандарті TELETEX;
- SDLC (Synchronous Data-Link Control), розроблений IBM для мереж SNA (Systems Network Architecture);
- LAPF – протокол канального рівня для мереж Frame Relay.

Використання циклічних кодів для захисту від помилок. Забезпечення надійного передавання – основна функція протоколів канального рівня. Але процедури виявлення та корекції помилок залежать від параметрів і типу фізичного каналу. Для можливості корекції виявлених помилок до n інформаційних розрядів кадру додається k контрольних, сформованих відповідним чином. Такий підхід використовується в таких каналах з великою кількістю помилок, як безпроводові, супутникові канали, лінії доступу абонентських мереж. В деяких сучасних комп'ютерних мережах і протоколах канального рівня кадри з виявленими помилками не коректуються, а знищуються (відкидаються), і корекція помилок виконується протоколами вищих рівнів. Такий спосіб використовується при передаванні високоякісними фізичними каналами трафіку (голосового, мультимедійного тощо), для якого може бути пошкоджено встановлений відсоток кадрів без суттєвого зниження якості обслуговування.

Найбільш поширеним способом контролю помилок, які виникають при передаванні в каналах зв'язку, є використання циклічних надлишкових кодів **CRC** (Cyclic Redundancy Check), які відносять до поліноміальних кодів. В основі будь-якого поліноміального коду лежить подання послідовності бітів у вигляді многочленів з коефіцієнтами 0 та 1, тобто кадр з n бітів подається сукупністю коефіцієнтів многочлена степеня $(n\text{-}1)$, який складається з n членів від x^{n-1} до x^0. Наприклад, число 10110001 подається поліномом 7 степеня: $x^7 + x^5 + x^4 + 1$. Зауважимо, що з даними многочленами можна виконувати арифметичні дії за *mod*2.

При використанні циклічного коду для контролю коректності передавання кадр розглядається як одне двійкове число відповідної розрядності, для якого за допомогою генерувального многочлена $G(x)$ формується циклічний надлишковий код $(n + k)$. В генерувальному многочлені обов'язково старший і молодший біти мають дорівнювати 1, а сам поліном, зазвичай, відповідає простому числу.

Алгоритм формування циклічного надлишкового коду (поля CRC кадру) такий.

1. До послідовності $M(x)$, яка являє собою кадр без поля контрольних розрядів, додаються k бітів, які дорівнюють степеню генерувального многочлена $G(x)$. Таким чином отримана послідовність містить $(n+k)$ бітів і відповідає послідовності (многочлену) $x^k M(x)$.
2. Отримана послідовність $x^k M(x)$ ділиться за модулем 2 на генерувальний многочлен $G(x)$.

3. Залишок від ділення $R(x)$, який має не більше k бітів і являє собою значення контрольних розрядів CRC, які передаються разом з кадром $M(x)$, утворюючи многочлен $T(x)$, що ділиться (за $mod2$) без залишку на генерувальний поліном $G(x)$.

При передаванні кадру без помилок модулем-отримувачем буде прийнято послідовність $T(x)$, а у випадку наявності помилок – послідовність $T(x)+E(x)$, де многочлен $E(x)$ вказує на наявність помилок. Для перевірки коректності прийнятого кадру в модулі-отримувачі прийнятий многочлен ділиться на генерувальний поліном $G(x)$. За відсутності помилки в отриманій послідовності залишок від ділення $T(x)/G(x)$ дорівнює 0, а за їх наявності залишок від ділення $(T(x)+E(x))/G(x)$ ненульовий. Таким чином можна виявити всі помилки, крім тих, що кратні генерувальному многочлену $G(x)$. Циклічні коди дозволяють виявити всі однократні помилки, дві ізольовані однократні і, що особливо важливо при передаванні в каналах зв'язку, всі пакети помилок довжиною не більше k. Взагалі для циклічного коду довжиною $(n+k)$ бітів ймовірність появи невиявлених помилок дорівнює $2^{-(n+k)}$ при незначній надлишковості (для кадру розміром 1500 байтів і 16-бітовому генерувальному поліномі додаткова службова інформація складає лише 0,13%, а при 32-бітовій контрольній послідовності – 0,26%.

Зазвичай використовуються генерувальні многочлени, залишок від ділення на які дорівнює 16 або 32 бітам. Найбільше поширення знайшли нижченаведені многочлени, деякі з яких є міжнародними стандартами:

- CRC-16 $x^{16}+x^{15}+x^2+1$, який запропоновано фірмою IBM і використовується в багатьох протоколах (наприклад, BSC);
- CRC-16 (CRC-CCITT) $x^{16}+x^{12}+x^5+1$, що розроблений комітетом CCITT (ITU) і використовується протоколом HDLC в мережах X.25, Bluetooth тощо;
- CRC-32 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$, який набув в останні роки широкого використання і запропонований IEEE для використання в мережах Ethernet та FDDI.

3.4 Особливості реалізації канального рівня в локальних мережах

Різниця між протоколами передачі даних для локальних і глобальних мереж пояснюються, в основному, особливостями використовуваних в цих мережах каналів. Канали локальних мереж мають високу якість і невелику довжину, а глобальних – значно нижчу якість і велику довжину. Такі особливості дають можливість спільного використання всіма вузлами мережі каналу передачі в режимі розподілу часу. Крім того, висока якість кабелів, що використовуються в локальних мережах, дозволяє відмовитися від складних алгоритмів повторного передавання пошкоджених або втрачених кадрів, оскільки ймовірність помилок при передаванні значно нижча, ніж у глобальних мережах.

В інституті інженерів з електротехніки та електроніки **IEEE** (Institute of Electrical and Electronics Engineers) створено комітет 802, який займається стандартизацією локальних мереж. Цим комітетом розроблено сімейство стандартів IEEE 802.x, які охоплюють тільки два нижніх рівні еталонної моделі OSI, а саме: фізичний та канальний, оскільки саме ці рівні відображають специфіку локальних мереж. Стандарти IEEE 802.x стали основою для розробки міжнародних стандартів ISO 8802.x. Перелік робочих груп IEEE 802.x та основна сфера їх розробок наведено в додатку В.1. Потрібно зазначити, що кількість робочих груп та їх призначення постійно змінюються, деякі з них ліквідуються або вносяться в інші.

На фізичному рівні моделі IEEE 802.x визначаються та стандартизуються різні типи середовища передавання, що, в принципі, не належить до фізичного рівня моделі OSI.

Канальний рівень моделі IEEE враховує специфіку локальних мереж, яка привела до його розділення на два підрівні, які часто називають рівнями. Рівень передавання даних для локальних мереж подають як сукупність двох підрівнів (рис. 3.10):

- **LLC** (Logical Link Control) – керування логічним каналом;
- **MAC** (Media Access Control або Medium Access Control) – керування доступом до середовища передавання.

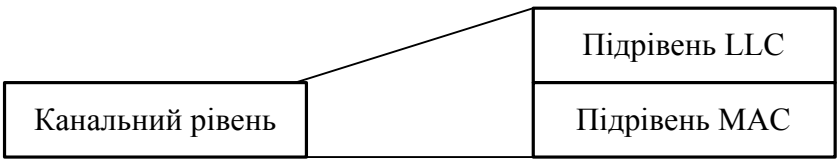


Рисунок 3.10 – Реалізація канального рівня в локальних мережах

Підрівень LLC виконує передавання кадрів між вузлами мережі з різним ступенем надійності, а також реалізує інтерфейс зв’язку з мережним рівнем, який через рівень LLC запитує необхідну йому транспортну послугу відповідної якості. Протоколи LLC підтримують декілька режимів роботи, які не залежать від конкретної технології локальної мережі. Треба зазначити, що протоколи підрівня LLC використовують й інші технології, які не розроблено комітетом 802, наприклад, протокол FDDI, який стандартизовано інститутом ANSI. Основними функціями підрівня LLC є:

- керування передаванням кадрів між станцією-відправником і станцією-отримувачем (аналогічно протоколу HDLC);
- забезпечення єдиного, незалежного від методу доступу до середовища передавання, інтерфейсу зв’язку з мережним рівнем.

Підрівень MAC враховує існування в локальних мережах середовища передавання даних, що поділяється між усіма вузлами та забезпечує його коректне використання відповідно до алгоритму доступу до каналу. Після того, як отримано доступ до каналу (середовища) передавання, керування

передається рівню LLC, який забезпечує передавання логічних одиниць даних з різною якістю обслуговування. Основні функції підрівня MAC:

- реалізація відповідного протоколу доступу до середовища передавання;
- формування кадрів відповідної структури;
- розпізнавання кадрів, призначених конкретній станції з використанням MAC-адрес (фізичних адрес мережних плат);
- виявлення та корекція помилок.

У сучасних локальних мережах найбільше використання знайшли декілька протоколів підрівня MAC, які реалізують різні алгоритми доступу до середовища передавання та визначають специфіку таких технологій локальних мереж, як Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, ARCNet, FDDI, 100VG-AnyLan та інші.

Потрібно зауважити, що протоколи підрівнів LLC та MAC взаємно незалежні: будь-який протокол підрівня LLC може використовуватись з будь-яким протоколом MAC і навпаки.

Стандарти IEEE 802.x мають чітку структуру, яка наведена на рис. 3.11. Така структура визначає загальні функції та підходи в різних технологіях локальних мереж. Особливості кожної технології LAN визначаються специфікою реалізації підрівня MAC і середовища передавання (тобто, фізичним рівнем). В кожній технології одному протоколу MAC відповідає декілька варіантів реалізації фізичного каналу передавання, тобто будь-яка локальна мережа може бути побудована на основі різних типів фізичного середовища.

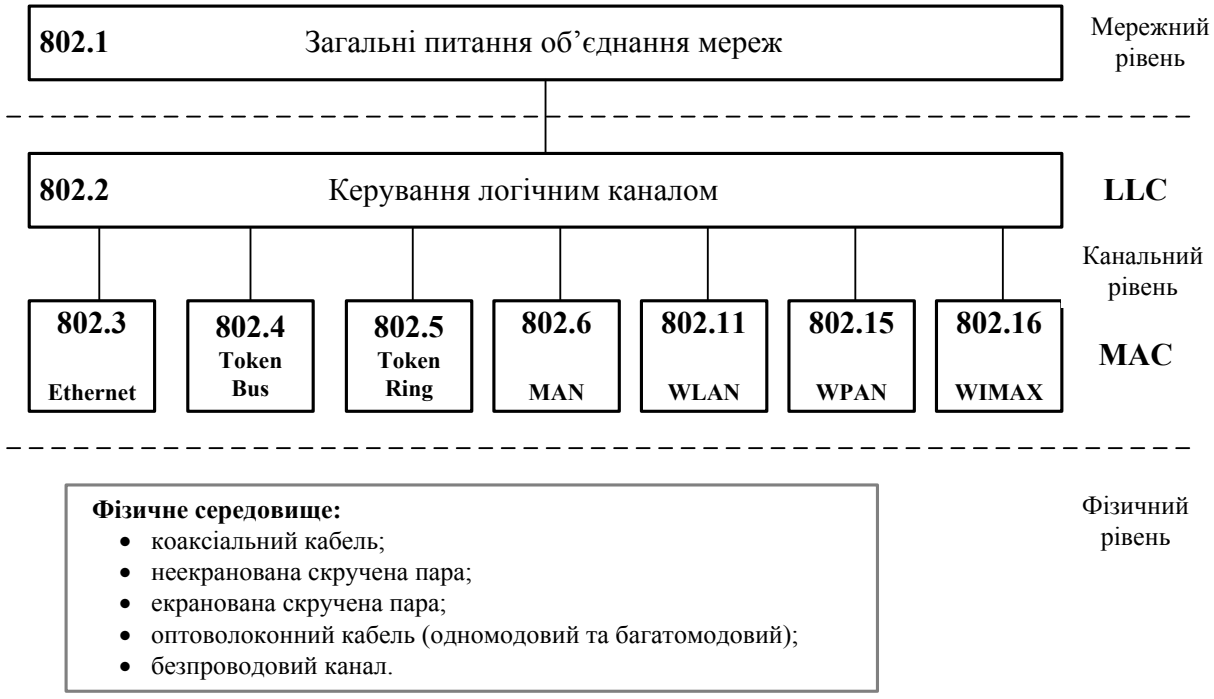


Рисунок 3.11 – Структура стандартів IEEE 802.x

Стандарти, які розроблені підкомітетом 802.1, є загальними для всіх технологій і містять визначення та характеристики локальних мереж, зв'язок моделі IEEE 802 з моделлю OSI. Але найбільш важливими, з практичної точки зору, є стандарти, які розглядають взаємодію мереж різних топологій, а також розробку складних мереж на основі базових технологій.

Поточний перелік стандартів IEEE 802.1 наведено в додатку В.2. Він містить такі важливі стандарти:

- IEEE 802.1D, в якому розглянута логіка роботи моста/комутатора;
- IEEE 802.1H, що визначає функціонування транслювального моста для об'єднання мереж різних топологій, наприклад, Ethernet і Token Ring, Ethernet та FDDI тощо (єдиним обмеженням при цьому є використання в сегментах, що об'єднуються, значення MTU одного розміру);
- IEEE 802.1Q, в якому визначено спосіб побудови віртуальних мереж VLAN на основі комутаторів.

3.5 Підрівень керування логічним каналом

Протокол LLC базується на протоколі HDLC і забезпечує передавання кадрів між модулями локальної мережі з необхідною якістю послуг транспортної служби (дейтаграмним способом, сервіс з встановленням з'єднання та визначенням і корекцією помилок), вимоги до якої передаються з мережного рівня. Протоколи мережного рівня передають для протоколу LLC блок даних (пакет), який крім адресної інформації містить вимоги до якості транспортних послуг, які має забезпечити протокол LLC. Цей пакет, разом з необхідними службовими опціями протоколу LLC, передається відповідному протоколу підрівня MAC.

Відповідно до стандарту IEEE 802.2 підрівень керування логічним каналом LLC надає верхнім рівням три типи процедур:

- LLC1 – процедури без встановлення з'єднання та без підтвердження;
- LLC2 – процедури з встановленням з'єднання та з підтвердженням;
- LLC3 — процедури без встановлення з'єднання, але з підтвердженням.

Сервіс без встановлення з'єднання та без підтвердження не гарантує доставку кадрів і тому найчастіше використовується в застосуваннях з потоковим передаванням даних або протоколами вищих рівнів, які самі забезпечують виявлення та корекцію помилок.

Сервіс із встановленням з'єднання та з підтвердженням забезпечує надійний обмін кадрами між модулями локальної мережі. Перед початком передавання будь-якого блоку кадрів встановлюється логічне з'єднання між відправником і адресатом, а в процесі передавання, що реалізується в режимі ковзного вікна, контролюється коректність переданих кадрів. Зазвичай

використовується в каналах з високим рівнем шуму. Цей сервіс аналогічний протоколам сімейства HDLC (LAP-B, LAP-D, LAP-M), які використовуються в глобальних мережах для забезпечення надійного передавання.

Сервіс без встановлення з'єднання з підтвердженням доставки у випадку, коли часові затрати на встановлення з'єднання неприйнятні, а підтвердження коректності прийому відправлених даних необхідне.

Вибір однієї з цих процедур залежить від стратегії розробки конкретного стека протоколів. В локальних мережах найчастіше використовується сервіс LLC1, оскільки їх канали зв'язку якісні, з низьким рівнем похибок. В стеках TCP/IP та IPX/SPX завжди використовується сервіс LLC1, а в стекові Microsoft/IBM використовуються сервіси LLC1 та LLC2, залежно від особливостей роботи базового протоколу NetBIOS/NetBEUI. Якщо він забезпечує передачу в дейтаграмному режимі, то використовується LLC1, якщо в режимі віртуального з'єднання – то LLC2. Режим LLC2 використовується також стеком протоколів SNA при використанні локальної мережі Token Ring. LLC2 використовується також компанією Hewlett-Packard при підключенні принтерів до мережі Ethernet. Сервіс LLC3 зазвичай використовується в системах автоматизованого управління.

Найчастіше в локальних мережах на сьогодні використовують протоколи LLC1.

Загальна структура кадру LLC, який, згідно зі стандартом IEEE 802.2, називають протокольним блоком даних PDU (Protocol Data Unit), наведена на рис. 3.12 (розмір опцій кадру наведено в байтах). Вони можуть бути трьох типів: інформаційні, керівні та ненумеровані.

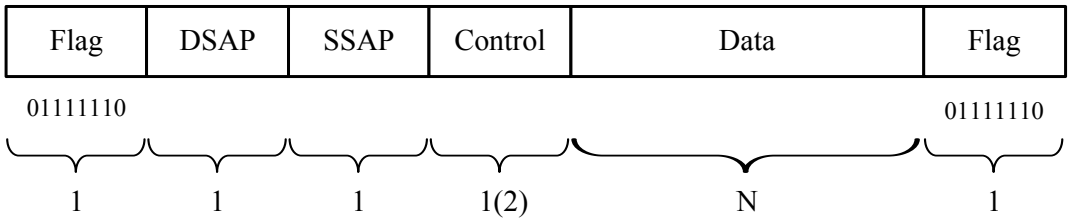


Рисунок 3.12 – Формат кадру LLC

Кадри LLC починаються і закінчуються синхросимволами **Flag**, які визначають межі блоку передачі. Поля **SSAP** (Source Service Access Point) та **DSAP** (Destination Service Access Point) дозволяють, відповідно, визначити сервіс верхнього рівня, який передає дані за допомогою цього кадру, та модулю якого протоколу на віддаленій станції необхідно передати цей кадр для подальшої обробки. Зазначимо, що всі протоколи відповідно до стандарту IEEE 802.2 мають свій код, наприклад, протокол IP - 0x6, NetBIOS - 0xF0. Поле **Control** визначає тип кадру та порядкові номери кадрів і повністю збігається з відповідним полем протоколу HDLC (LAPB). Поле даних кадру LLC **Data** використовується для передавання пакетів протоколів верхніх рівнів IP, IPX, AppleTalk, DECnet тощо.

Взаємодія протоколів підрівнів LLC та MAC наведена на рис. 3.13. Пакет, який надходить з мережного рівня, обробляється згідно з вибраною процедурою, створюється кадр LLC, який передається на підрівень MAC. Протокол MAC, який реалізує відповідний алгоритм доступу до середовища передавання, формує кадр, що і передається в канал. При цьому використовуються фізичні адреси (6-байтові MAC-адреси) вузлів.

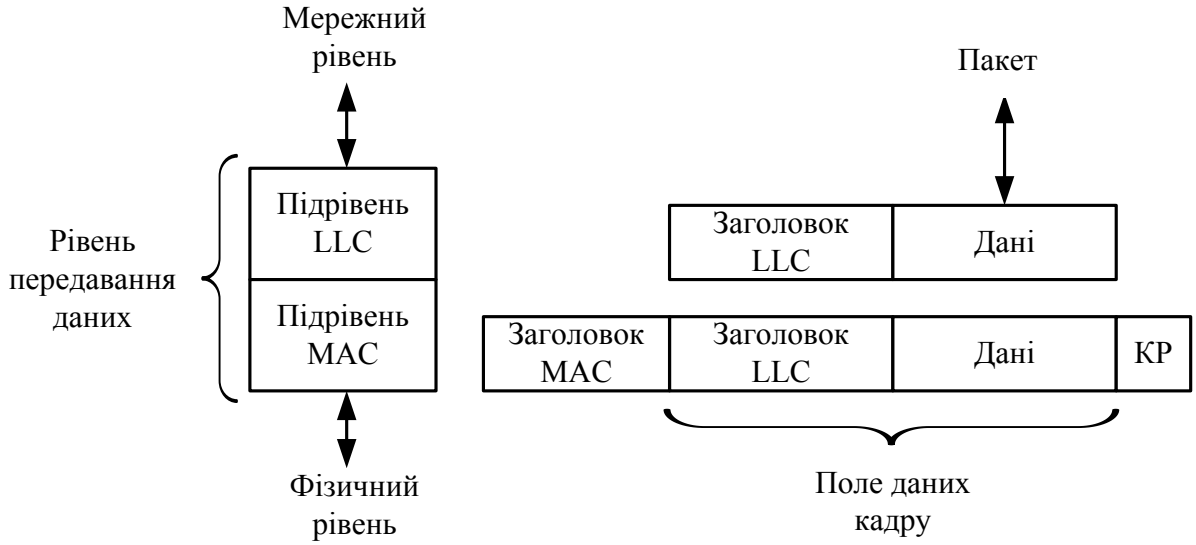


Рисунок 3.13 – Взаємодія протоколів підрівнів LLC та MAC

3.6 Підрівень керування доступом до середовища передавання даних

Основною проблемою при розробці протоколів підрівня MAC є вибір процедур та правил доступу модулів мережі до середовища передавання, тобто алгоритмів розподілу спільного мережного каналу між всіма вузлами LAN. Це пояснюється тим, що від алгоритму доступу до спільного середовища передавання залежить ефективність взаємодії станцій локальних мереж. **Методом доступу до середовища передавання** називається правило, згідно з яким організовано доступ модулів локальної мережі до фізичного каналу. Критерієм ефективності методу доступу найчастіше служить **час доступу до середовища передавання** – проміжок часу між появою запиту на передавання та безпосередньо початком передачі інформації. Значення цього параметра суттєво залежить від топології мережі, методу доступу, що використовується в даній LAN, способу керування мережею. Оскільки існують різні типи локальних мереж і вимог до них, неможливо розробити універсальний метод доступу, який би був ефективним у всіх LAN. Кожний з існуючих методів доступу має свої переваги та недоліки.

Складність проблеми полягає в тому, що окремі станції або модулі мережі мають виконувати передавання таким чином, щоб не заважати один одному, оскільки середовище передавання поділяється між усіма вузлами, а при одночасному передаванні інформації від двох або більшої кількості

станцій відбувається інтерференція та взаємне спотворення сигналів, тобто виникає **конфлікт** (або **колізія**) в каналі. При цьому на сьогодні локальні мережі будуються переважно таким чином, щоб в мережі не було єдиного вузла (диспетчера), який би координував роботу інших модулів, а всі станції були б з однаковим статусом і могли б функціонувати автономно.

Тому в LAN на сьогодні використовуються **розподілені методи доступу** до середовища передавання, які є більш надійними, оскільки відмова центрального вузла мережі з централізованим методом призводить до відмови всієї мережі.

Для доступу до середовища передавання в локальних мережах використовуються два типи методів доступу: детерміновані та недетерміновані (випадкові або ймовірнісні). **Детерміновані методи доступу** передбачають наявність певного алгоритму (опитування, передачі права доступу тощо), відповідно до якого модулю мережі надається доступ до середовища передавання. Такі методи вимагають передавання додаткової спеціальної інформації керування і дозволяють врахувати особливості фізичної та логічної топологій локальної мережі й характер інформації, що передається в мережі. Такі методи вимагають використання більш складної апаратури та реалізації, але дозволяють забезпечити найбільш ефективне використання середовища передавання мережі.

Методи **недетермінованого доступу** прості в реалізації, завдяки чому знайшли більш широке використання в локальних мережах (завдяки поширенню мережі Ethernet). Такі методи більш ефективні в разі передавання коротких кадрів (до 1500 байтів) і низької інтенсивності запитів на передавання (завантаженості каналу передавання). Методи ж детермінованого доступу забезпечують стабільну роботу мережі при передаванні кадрів значного розміру (зазвичай, від 4 Кбайтів до 16 Кбайтів) і підвищеному рівні завантаженості каналу, оскільки передбачають чіткий алгоритм доступу модулів мережі до середовища передавання. Крім того, в разі необхідності детерміновані методи доступу дозволяють достатньо просто реалізувати пріоритетне передавання як кадрів, так і всього повідомлення.

Серед **детермінованих** методів доступу розрізняють:

- метод опитування;
- метод кільцевих слотів;
- маркерні методи;
- інтервальні методи;
- інтервально-маркерні методи.

Метод опитування зазвичай використовується в мережах з фізичною топологією «зірка» і передбачає наявність центрального вузла, який послідовно опитує кожну зі станцій на наявність в ній кадру для передачі. Якщо станція, яку опитує центральний вузол, має інформацію для передавання, вона сповіщає про це центральний модуль передаванням запиту на доступ до середовища передавання. У відповідь на це центральний вузол надає станції монопольне право використання середовища передавання. Після

закінчення передавання інформації центральний вузол продовжує опитування наступних станцій. Таким чином всі станції мережі можуть здійснювати передавання інформації тільки після того, як отримають дозвіл на це від центрального вузла. Оцінюючи час очікування доступу станції до середовища передавання можна сказати, що він суттєво залежить від тривалості опитування абонентських станцій та інтенсивності запитів. Мінімальний час очікування залежить від кількості станцій в мережі та тривалості їх опитування, а максимальний час визначається за умови наявності інформації для передавання в кожній абонентський станції та є фіксованим, оскільки кожний наступний запит на передавання кадру в конкретній станції буде оброблятися тільки після обслуговування поточного запиту.

Метод кільцевих слотів використовується лише в локальних мережах з топологією «кільце» і передбачає тактований доступ до середовища передавання. В мережах з таким доступом одночасно передається деяка фіксована кількість пакетів (сегментів), які можуть мати один з двох станів: «вільно» та «зайнято». Станція, яка має інформацію для передавання, очікує на появу в каналі чергового сегмента і перевіряє його стан. При виявленні порожнього сегмента (який не містить даних для передавання) станція записує в нього дані для передавання та позначає його як зайнятий сегмент. Кожна станція кільця перевіряє отриманий сегмент і при появі пакета з даними для неї записує отриману інформацію в свій буфер, а сегмент позначає як вільний.

Очевидно, що для оптимального завантаження мережі з таким методом доступу потрібно, щоб кількість станцій або дорівнювала кількості сегментів кільця, або була більшою від неї. Основною перевагою методу кільцевих слотів є малий час відповіді і стабільна робота мережі, однак досягається це через неефективне використання каналу передавання даних. Тому найчастіше мережі з таким методом доступу використовуються в системах оперативного контролю та керування технологічними процесами. Прикладом такої мережі є Cambridge Ring.

Метод передавання права передбачає постійне передавання в мережі спеціального кадру невеликого розміру (зазвичай, 3 байти), який називається маркером (token). Маркер, який має два стани: «вільно» і «зайнято», послідовно передається від однієї станції до іншої і фактично є ознакою дозволу на передавання даних. Станція, яка має дані для передачі, очікує на вільний маркер, при отриманні якого додає до нього свій кадр, встановлює маркеру ознаку зайнятості і передає їх в мережу. Маркер разом з під’єднаним кадром послідовно передається наступним станціям, кожна з яких перевіряє адресу отримувача, і якщо дані призначені не їй, а іншій станції, маркер передається далі в канал. Якщо ж кадр призначено цій станції, інформація записується в її буфер, а в маркері встановлюється спеціальний біт підтвердження і передається станції, що відправила кадр. Після отримання підтвердження станція-відправник змінює стан маркера на «вільно» і передає його в канал. Маркерні методи доступу мають такі перева-

ги: гарантований максимальний час затримки передавання кадру, що приводить до можливості їх використання в реальному часі; неможливість виникнення колізій в каналі і завдяки цьому забезпечення стабільної роботи мережі; досить ефективне використання ресурсів каналу і мережі в цілому достатньо просто дозволяє реалізувати пріоритетний доступ окремих станцій до ресурсів мережі. Однак складність процедур ініціалізації кільця та відновлення коректного функціонування мережі після відмов або втрати маркера призводять до ускладнення модулів мережі й значної їх вартості.

Прикладами локальних мереж, які функціонують на основі маркерного доступу, є мережі кільцевої топології Token Ring (TR, TRN), шинної топології Token Bus та ARCnet-Bus, а також топології «зірка» ARCnet-Star.

В деяких мережах з кільцевою топологією використовується **метод вставлення регістра**, який дозволяє під'єднувати черговий кадр станції до потоку кадрів, що передаються в каналі. Такий підхід дозволяє одночасно передавати в кільцевому каналі за допомогою одного маркера декілька кадрів від різних станцій, що приведе до підвищення ефективності функціонування всієї локальної мережі.

Інтервальні методи доступу або методи синхронного розподілу часу передбачають, що весь цикл обміну зі станціями каналу розбивається на декілька часових інтервалів (тайм-доменів, тайм-слотів), кількість яких дорівнює кількості модулів в каналі N (рис. 3.14). Кожній станції надається свій часовий інтервал, протягом якого вона отримує доступ до всіх ресурсів мережі і може передавати дані Д. В разі відсутності в ній даних для передавання (В) часовий інтервал не використовується іншими модулями. Можна сказати, що при великій інтенсивності запитів на передавання ефективність використання каналу наближається до максимальної. При зниженні інтенсивності збільшуються періоди очікування (простою), що призводить до зниження ефективності.

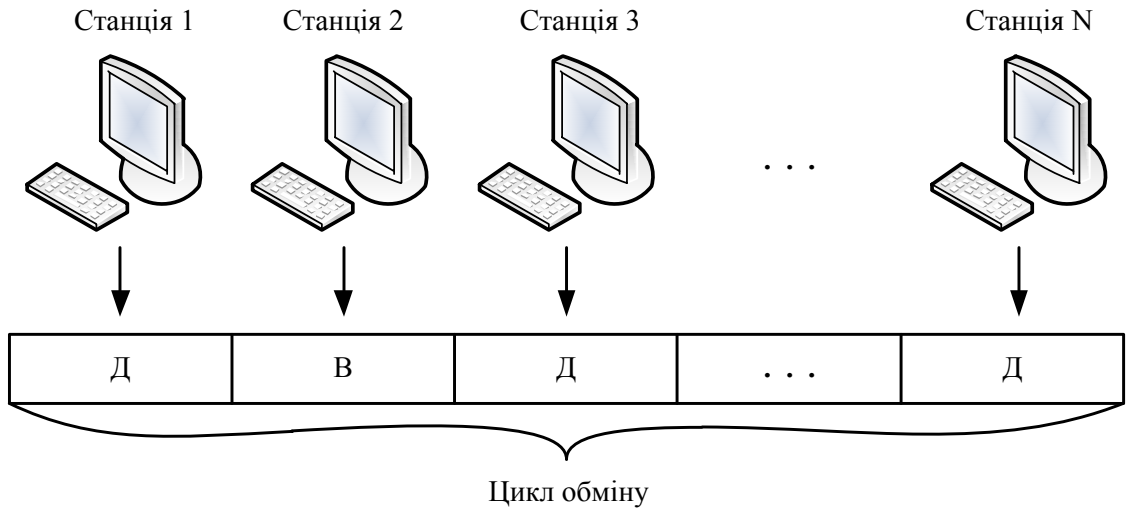


Рисунок 3.14 – Реалізація інтервального доступу до каналу:
Д – передавання даних, В – відсутність даних для передавання

Розрізняють мережі з упорядкованим і невпорядкованим розташуванням станцій, в яких, відповідно, послідовність передавання права доступу до середовища передавання збігається з послідовністю підключення модулів до каналу або не збігається. Останній варіант дозволяє враховувати особливості доступу станцій до середовища передавання, але призводить до ускладнення процедури передавання прав доступу.

Інтервально-маркерні методи доступу забезпечують право на доступ до каналу як на основі синхронного розподілу часу, так і передавання маркера з урахуванням завантаженості мережі. Якщо мережа достатньо завантажена, то передавання кадрів відбувається згідно з часовим розподілом, при зменшенні завантаженості нижче встановленого значення в мережі генерується маркер, який передається в каналі і визначає право станцій передавати дані. Такий адаптивний алгоритм передавання приведе до зменшення часу очікування та підвищення ефективності використання ресурсів мережі.

При реалізації **методів недетермінованого (випадкового, імовірнісного) доступу** кожна станція мережі незалежно від інших станцій може намагатися передавати дані в канал. Оскільки при цьому можлива одночасна спроба декількома станціями отримати доступ до спільного середовища передавання, такі методи часто називають **методами множинного доступу**. Загальною проблемою для всіх типів цих методів є проблема змагань, яка виникає при спробі одночасного передавання даних декількома станціями, що призводить до **колізій** в каналі, тобто зіткнень кадрів від різних відправників. Імовірність виникнення колізій залежить від кількості станцій в каналі та від інтенсивності їх звертань до середовища передавання.

Першою мережею, в якій використовувався випадковий доступ, була мережа ALOHA, розроблена в Гавайському університеті. В таких мережах одразу після формування кадр передається в канал, і при виникненні колізії передавання припиняється, а через інтервал часу, який необхідний для передавання одного кадру, станція з імовірністю p повторно передає кадр. Ефективність такого доступу суттєво залежить як від кількості станцій, так і від інтенсивності запитів на захоплення каналу для передавання й, зазвичай, становить не вище 0,2.

У мережі «синхронна (тактована) ALOHA» весь час роботи поділено на інтервали часу (слоти), протягом кожного з яких передається один кадр. Станції синхронізовано таким чином, що кожна з них знає, коли починається новий слот. Передавання кадрів можливе тільки в момент початку чергового слоту, і при виникненні колізії всі станції припиняють передавання та, з імовірністю p, повторно намагаються передати кадр на початку нового часового слоту. Ефективність такого підходу вища, але не перевищує 0,37.

Одним зі способів зниження кількості колізій є прослуховування каналу перед передаванням кадру й передавання його тільки за наявності вільного каналу. Такий режим передавання називається **множинним доступом**

з контролем несучої (МДКН) CSMA (Carrier Sense Multiple Access). Проте через затримку розповсюдження сигналу станція, що розпочала передавання, не може визначити, що інша станція також розпочала передавання. В результаті цього відбувається інтерференція сигналів і спотворення інформації в середовищі передавання.

В мережах з множинним доступом і контролем несучої передбачено декілька методів захоплення каналу (табл. 3.3):

Таблиця 3.3 – Характеристика методів захоплення каналу

Умова	Ненаполегливий	p-наполегливий	1-наполегливий
Канал вільний	Передати невідкладно	Передати з ймовірністю p, відкласти передавання з ймовірністю 1 - p	Передати невідкладно
Канал зайнято	Випадковий час очікування і контроль	Станція очікує випадковий період часу і знову намагається захопити канал. Після звільнення каналу передає з ймовірністю p, або відкладає передачу з ймовірністю 1–p	Безперервно контролює несучу
Колізія	Повторно передає кадр після випадкового часу очікування	Повторно передає кадр після випадкового часу очікування	Повторно передає кадр після випадкового часу очікування

- ненаполегливий контроль несучої, коли станція прослуховує канал не постійно, а через випадковий інтервал часу;
- p-наполегливий контроль несучої;
- 1-наполегливий контроль, при реалізації якого станція передає кадр одразу, як тільки звільнився канал.

Принципова відмінність алгоритму ALOHA від CSMA полягає у процедурі виявлення колізії: в першому випадку колізія виявляється на вході станції-отримувача, а в другому – на виході станції-відправника.

Розрізняють такі модифікації методу CSMA:

- метод множинного доступу з контролем несучої і виявленням колізії CSMA/CD (Carrier Sense Multiple Access with Collision Detection);
- метод множинного доступу з контролем несучої з уникненням колізії CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

Метод множинного доступу з контролем несучої і виявленням колізії CSMA/CD використовується в усіх існуючих мережах Ethernet, Fast Ethernet та Gigabit Ethernet, робота яких відповідає специфікації IEEE 802.3.

Середовище, що розподіляється між усіма модулями мережі, в будь-який момент часу може знаходитись в одному з трьох станів: простою (канал вільний), передавання (канал зайнятий), конкуренції (колізія), що показано на рис. 3.15. Кожна станція одночасно може передавати дані і «прослуховувати» канал, причому прослуховування відбувається постійно. Така процедура дозволяє визначити, що відбувається в каналі, тобто в якому стані він знаходиться.

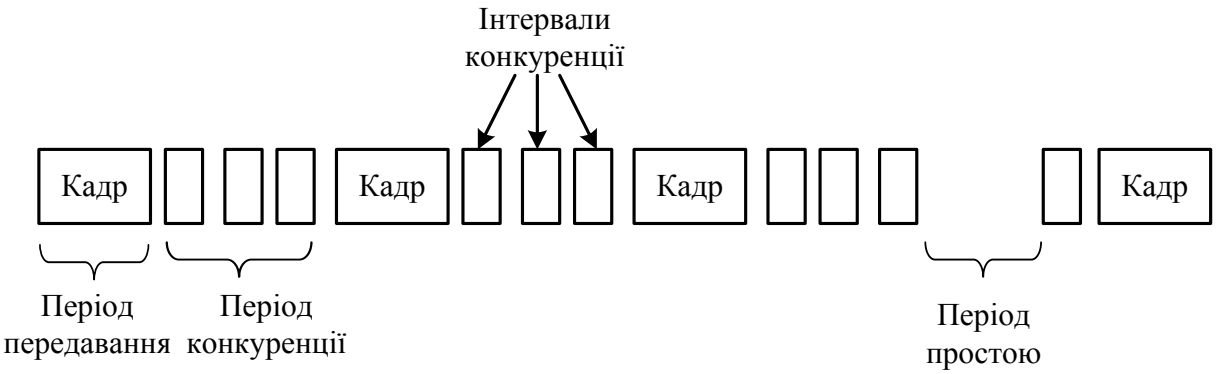


Рисунок 3.15 – Протокол CSMA/CD

Якщо канал вільний і у MAC-рівня сформовано кадр для передавання, то кадр передається в мережний канал через фізичний рівень, який одночасно контролює стан середовища передавання. У випадку, коли дані, які передаються в канал і приймаються з нього, не збігаються, це буде означати, що ще якась станція виконує передавання (чи намагається це зробити).

Схема виявлення колізії в каналі достатньо проста і наведена на рис. 3.16. У випадку конфлікту (колізії) в каналі передача припиняється, а відправник передає в канал спеціальну 32-бітову послідовність *jam*, яка інформує всі інші станції, що вже виконується передавання кадру і їм забороняється виконувати передавання своїх даних. Після випадкового інтервалу часу виконується повторна спроба передавання кадру, причому, для кожної наступної спроби передавання час очікування довільно збільшується. Після 16 послідовних невдалих спроб передати дані видається сигнал помилки *link error*, який передається протоколу верхнього рівня і свідчить про те, що канал недоступний. Передавання даного кадру відкладається і розпочинається передавання наступного кадру, який передано з підрівня LLC.

Завдяки такій процедурі метод CSMA/CD, порівняно з CSMA, дозволяє швидше передавати дані і більш ефективно використовувати мережний канал.

Метод множинного доступу з контролем несучої з уникненням колізії CSMA/CA зазвичай використовується в безпроводових мережах, робота яких описана в специфікаціях IEEE 802.11, що будуть розглянуті в

розділі 9. На відміну від методу доступу CSMA/CD, при реалізації якого *jam-послідовність* передається тільки при виявленні колізії, при використанні методу CSMA/CA спочатку відправляється сигнал *jam,* який інформує станції каналу, що деяка станція має дані для передавання та бажає це реалізувати. Після передавання *jam-послідовності* станція ще деякий час перевіряє канал на наявність в ньому аналогічних сигналів *jam* від інших станцій. Якщо такий сигнал виявлено, тобто, деякий інший вузол виконує передавання, дана станція очікує довільний проміжок часу і знов намагається передати дані; при відсутності інших передавань станція починає передавання своїх даних. При такому підході можлива лише колізія коротких *jam-кадрів,* а не колізія інформаційних кадрів.

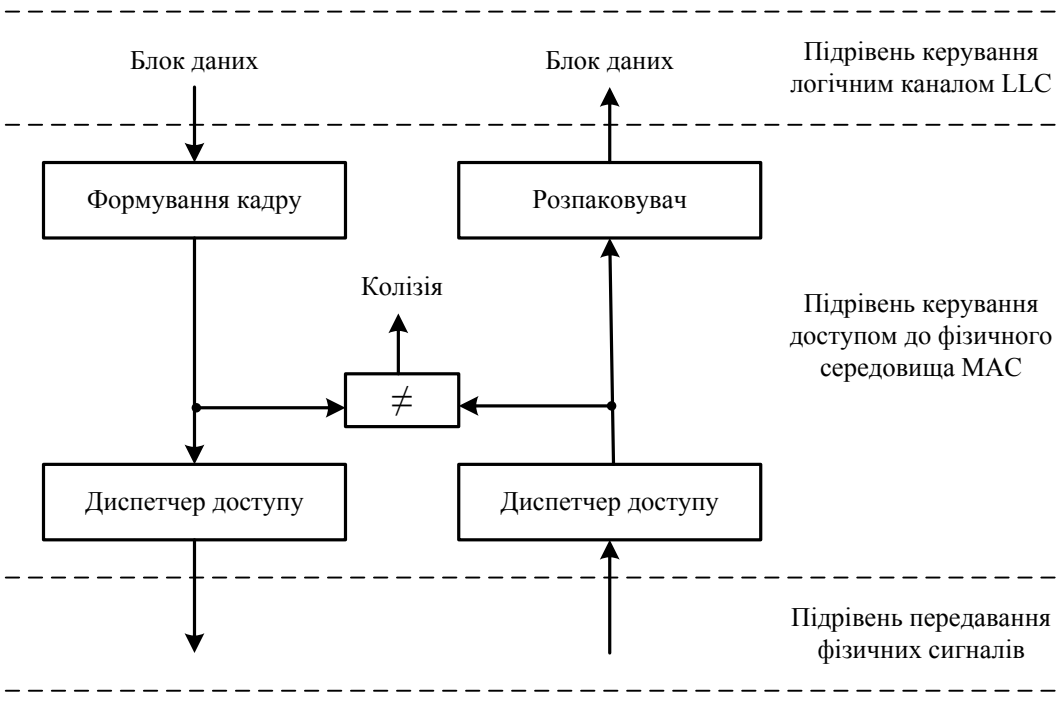


Рисунок 3.16 – Схема виявлення колізій

Перевагою випадкових недетермінованих методів доступу є простота їх реалізації, невелика вартість, а також незначний час очікування при низькій інтенсивності запитів. Збільшення кількості запитів на захоплення каналу призводить до нестабільної роботи мережі, оскільки імовірність успішного передавання зменшується, а час затримки та імовірність колізій збільшуються, що є суттєвим недоліком такого підходу. Крім того, не можна визначити максимальний гарантований час затримки до успішного передавання кадру, що призводить до неможливості використання цих методів в мережах, які мають передавати дані в реальному часі.

Протоколи без колізій. Хоча в протоколі CSMA/CD конфлікти не відбуваються після того, як станція захопила канал передавання, вони можуть відбуватися в період конкуренції, що призводить до зниження загальної пропускної спроможності (продуктивності) мережі, особливо при значній

довжині кабелю і невеликих кадрах. Тому розроблено декілька способів доступу до середовища передавання, які дозволяють станціям визначити право передавання даних в каналі без колізій. Розглянемо один з таких методів, який називається **методом бітової карти (bit map)**. При реалізації цього методу (рис. 3.17) після передавання кадру вводиться період конкуренції, який складається з такої кількості часових інтервалів, яка дорівнює числу станцій в LAN (на рисунку – 8 інтервалів конкуренції). Якщо станція має кадр для передавання, вона передає одиничний біт заявки у свій часовий інтервал. Таким чином, після завершення періоду конкуренції всім станціям відомо, які з них мають інформацію для передавання. Станції починають передавання відповідно до своїх порядкових номерів, тобто, відповідно до черги, яка визначена в період конкуренції. Реалізуючи такий алгоритм можна повністю уникнути конфлікту в каналі, оскільки всі станції прослуховують лінію, відслідковуючи її стан.

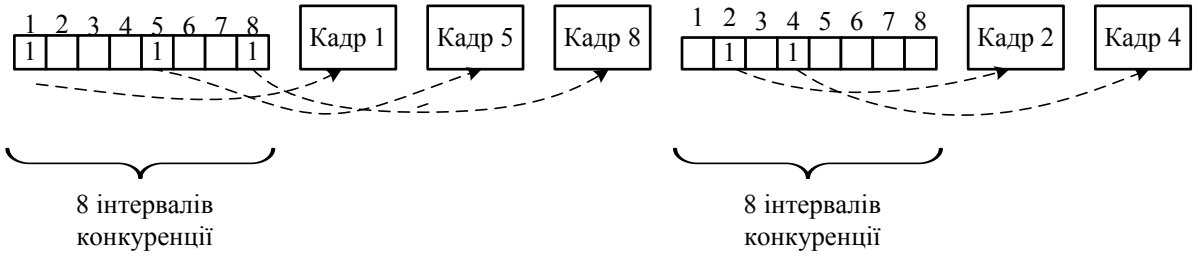


Рисунок 3.17 – Протокол бітової карти

Оцінюючи ефективність даного алгоритму можна сказати, що в незавантаженому каналі біт-карта буде просто постійно повторюватись.

При великій завантаженості каналу, коли всі станції мають інформацію для передавання, після періоду подачі заявок з N бітів передаються N кадрів. При цьому накладні витрати на передавання одного кадру складають всього один біт.

3.7 Технологія Ethernet

На сьогодні технологія Ethernet є найбільш поширеною технологією локальних мереж і описується стандартами IEEE 802.3, поточний список яких наведено в додатку В.3, а також (з невеликими відмінностями) в стандартах міжнародних ISO 8802.3 та європейських ECMA 82. Перша версія мережі Ethernet розроблена фірмою Xerox у 1975 році. Безперечною перевагою даного типу локальних мереж є її простота, невелика вартість, можливість масштабування.

Топологія, параметри та особливості реалізації мережі Ethernet залежать від типу середовища передавання та маркуються як **XBaseY**, де X визначає швидкість передавання (пропускну спроможність), зазвичай, в Мбіт/с, а Y – максимальну довжину сегмента (в сотнях метрів), що справе-

дливо для перших версій мережі, або тип середовища передавання (Base вказує, що використовуються вузькосмугові канали, тобто передача відбувається на одній базовій частоті, на відміну від широкосмугових мереж Broad, які використовують декілька несучих частот).

Для доступу до середовища передавання в усіх модифікаціях мережі Ethernet використовується метод множинного доступу з контролем несучої та виявленням колізії CSMA/CD (більш детально даний алгоритм розглянуто в попередньому підрозділі). Всі станції підключені до одного спільного середовища передавання, тому передавання реалізується тільки між двома будь-якими вузлами мережі. Мережний адаптер кожної станції мережі приймає та аналізує всі кадри, які передаються в каналі. Якщо прийнятий кадр має MAC-адресу отримувача, що збігається з MAC-адресою станції, фрейм записується в буфер і перевіряється на наявність помилок за допомогою циклічного коду CRC. У разі відсутності помилок кадр передається на мережний рівень для подальшої обробки, за наявності помилок кадр знищується, і ніяких повідомлень про помилку отримувач не відправляє. Повторне передавання втрачених або пошкоджених кадрів можуть ініціювати протоколи вищих рівнів. Тому мережа Ethernet є мережею, яка функціонує в режимі негарантованої доставки, але використання якісних, надійних каналів зв'язку компенсує відсутність механізмів корекції помилок. Значення основних параметрів процедури передавання кадрів наведені в таблиці 3.4.

Таблиця 3.4 – Основні параметри процедури передавання кадру

Параметр	Значення
Міжкадровий інтервал	96 бітових інтервалів
Інтервал відтермінування (slot time)	512 бітових інтервали
Довжина jam-послідовності	32 біти
Максимальна кількість спроб передавання	16
Максимальна довжина кадру (без преамбули)	1518 байтів
Мінімальна довжина кадру (без преамбули)	64 байти (512 бітів)
Довжина преамбули	8 байтів

Slot time – це час, протягом якого станція гарантовано може визначити, що в каналі немає колізії. Цей параметр пов'язаний з іншим параметром – «вікном колізії» (collision window), який дорівнює часу двократного проходження сигналу між найбільш віддаленими вузлами мережі, тобто найбільшої затримки, при якій станція ще може виявити виникнення колізії. Якщо станція, яка передає кадр, не розпізнає колізію, фрейм буде втрачено, що призведене до необхідності повторного передавання за допомогою протоколу вищого рівня, яка, в свою чергу, суттєво знизить корисну пропускну спроможність мережі.

У мережах Ethernet використовуються кадри нижчевказаних чотирьох форматів (рис. 3.18), які, зазвичай, підтримують реалізацію найбільш поширених протоколів мережного рівня (розміри опцій вказані в байтах):

- Ethernet II (протоколи IPX, IP, AppleTalk Phase I);
- Ethernet 802.2 (протоколи IPX, FTAM);
- Ethernet 802.3 (протокол IPX);
- Ethernet SNAP (протоколи IPX, IP, AppleTalk Phase II).

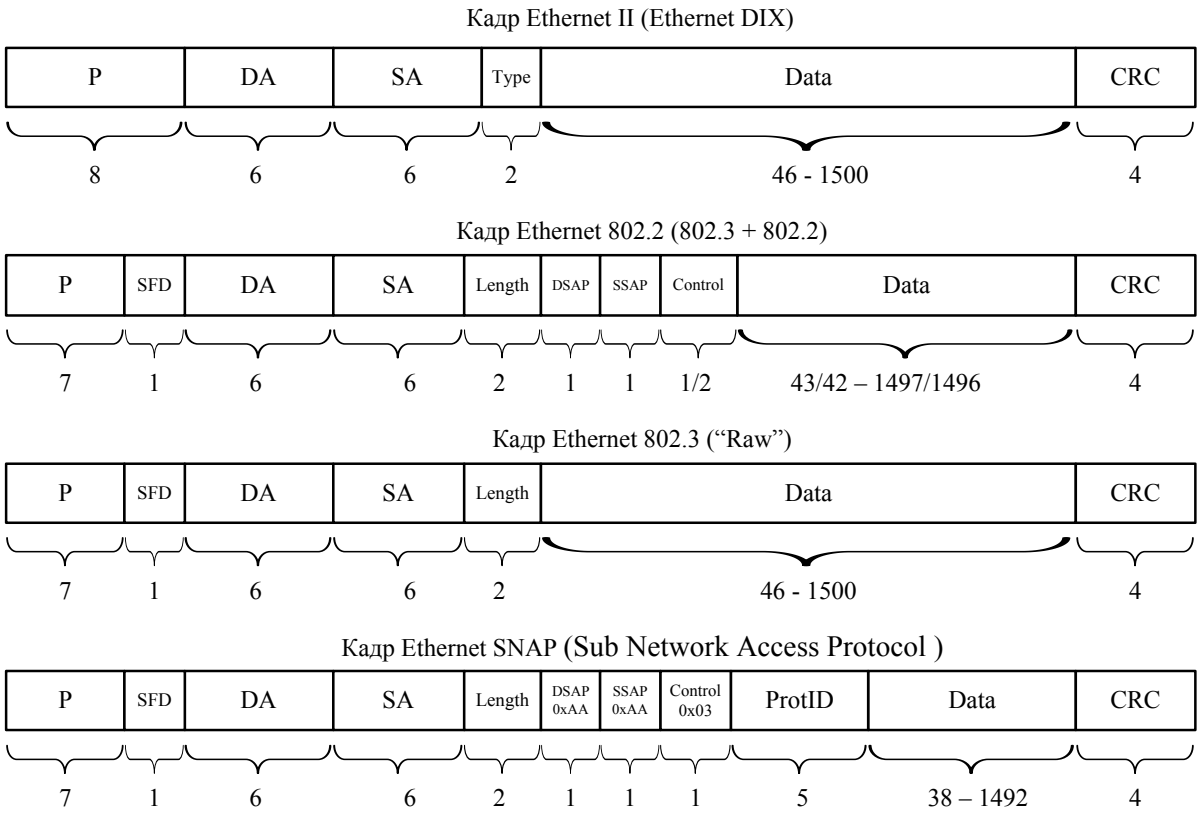


Рисунок 3.18 – Формати кадрів Ethernet

Зауважимо, що на сьогодні найчастіше використовується кадр Ethernet II.

P (Preamble) – преамбула, використовується для синхронізації та складається з семи (або восьми) байтів, які мають значення 10101010;

SFD (Start of Frame Delimiter) – роздільник початку кадру (значення 10101011) вказує на те, що наступний байт є байтом заголовка кадру;

DA (Destination Address) – адреса призначення (6-байтова MAC-адреса отримувача), яка може бути типів:

- **unicast** – індивідуальна адреса, яка визначає одного адресата для отримання даного кадру (в цьому випадку перший біт старшого байта дорівнює 0, а унікальність адреси забезпечується виробником мережного обладнання: старша половина MAC-адреси визначає ви-

робника, а молодша – ідентифікатор даного мережного обладнання у виробника);

- **broadcast** – широкомовна адреса, яка вказує на те, що даний кадр отримують всі вузли мережі, та має значення 0xFFFFFFFFFFFF;
- **multicast** – групова адреса, яка визначає, що отримувачами кадру є всі вузли вказаної групи (при цьому перший біт старшого байта дорівнює 1, а інші містять ідентифікатор групи вузлів мережі);

SA (Source Address) – MAC-адреса відправника (unicast-адреса);

Type – поле визначає який протокол верхнього рівня передає свої дані в кадрі (виконує функції полів DSAP і SSAP з заголовка LLC);

Length – містить розмір поля даних Data в байтах;

Data – містить дані, що передаються протоколом верхнього рівня;

CRC (Cyclic Redundancy Check) – контрольна послідовність кадру містить контрольну суму кадру, яка обчислена за алгоритмом CRC-32;

DSAP, SSAP та **Control** фактично належать до заголовка LLC-кадру;

ProtID – дентифікатор протоколу, який дозволяє використовувати кадри Ethernet для передачі даних більш широкої сукупності протоколів верхнього рівня й складається з двох полів: трибайтового OUI (Organizationally Unique Identifier), яке визначає організацію, що контролює коди протоколів в другому двобайтовому полі Type, яке за змістом повністю ідентичне розглянутому одноіменному полю.

Відповідно до стандарту IEEE 802.3 розрізняють декілька типів мереж Ethernet з різними пропускною спроможністю та типом середовища передачі. Розглянемо основні модифікації стандартів мережі Ethernet та їх основні характеристики, які відносять і до фізичного рівня.

Мережі Ethernet з пропускною спроможністю 10 Мбіт/с. Мережі даної групи мають декілька типів, які наведені в таблиці 3.5, і мають різну топологію: як «шина» (10Base5, 10Base2, 10Broad36), так і «зірка» (всі інші). Для передачі сигналів використовується манчестерське кодування, а для синхронізації – надлишковий код 4B/5B (див. пункт 2.4.1), який дозволяє уникнути передачі довгих нульових і одиничних послідовностей. Мережі з топологією «зірка», які найбільш поширені на сьогодні, використовують концентратори (hub) для об’єднання станцій (рис. 3.19). Використання такого підходу дає можливість створювати мережі і з багаторівневою деревоподібною топологією, причому до кожного концентратора можуть підключатися як станції, так і інші концентратори.

Мережі такої топології характеризуються значно більшою живучістю, оскільки відключення сегмента з будь-яких причин не позначається на роботі інших сегментів. Крім того, така конфігурація мережі дозволяє достатньо просто організувати контроль за станом мережі і знизити, таким чином, кількість конфліктів та достатньо легко виявити як несправний сегмент, так і станцію, яка є джерелом перевантаження мережі.

Таблиця 3.5 – Параметри мереж Ethernet

Найменування стандарту	Номер стандарту	Тип середовища передачі	Максимальна відстань між модулями, м	
			напів-дуплекс	дуплекс
10 Мбіт/с				
10Base 5	DIX-802.3	коаксіальний кабель діаметром 0,5 дюйма (1,27 см), «товстий Ethernet», 50 Ом	500	
10Base2	IEEE 802.3a	коаксіальний кабель діаметром 0,25 дюйма (0,125 см), «тонкий Ethernet», 50 Ом	185	
10Broad 36	IEEE 802.3b	коаксіальний кабель, 75 Ом	1800	
FOIRL	IEEE 802.3d	два оптоволоконних кабелі	1000	1000
10Base-T	IEEE 802.3i	дві скручені пари категорії 3	100	100
10Base-FL	IEEE 802.3j	два оптоволоконних кабелі	2000	2000
10Base-FB	IEEE 802.3j	два оптоволоконних кабелі	2000	
10Base-FP	IEEE 802.3j	два оптоволоконних кабелі	1000	
100 Мбіт/с (Fast Ethernet)				
100Base-TX	IEEE 802.3u	дві скручені пари категорії 5	100	100
100Base-FX	IEEE 802.3u	два оптоволоконних кабелі	412	2000
100Base-T4	IEEE 802.3u	чотири скручені пари категорії 3	100	
100Base-T2	IEEE 802.3y	чотири скручені пари категорії 3	100	100
1 Гбіт/с (Gigabit Ethernet)				
1000Base-LX	IEEE 802.3z	багатомодовий оптоволоконний кабель діаметром 62,5 (або 50) мкм, діапазон хвиль 1270–1355 нм	316 (або 550)	316 (або 550)
		одномодовий оптоволоконний кабель діаметром 10 мкм	316 (або 550)	5 км
1000Base-SX	IEEE 802.3z	багатомодовий оптоволоконний кабель діаметром 62,5 (або 50) мкм, діапазон хвиль 770–860 нм	275 (або 550)	275 (або 550)
1000Base-CX	IEEE 802.3z	екранована скручена пара	25	25
1000Base-T	IEEE 802.3ab	чотири неекрановані скручені пари категорії 5 (або 6)	100	100
10 Гбіт/с (10G Ethernet)				
10GBase-T	IEEE 802.3an-2006	скручена пара категорії 5e, 6 та 6a (або 7)		24, 55 та 100 відповідно
10GBase-CX4	IEEE 802.3ae	мідний кабель		15

Продовження таблиці 3.5

10GBase-SR	IEEE 802.3ae	багатомодовий оптоволоконний кабель різного діаметра		26 (82)
10GBase-LR	IEEE 802.3ae	багатомодовий оптоволоконний кабель різного діаметра		10 км
10GBase-ER	IEEE 802.3ae	багатомодовий оптоволоконний кабель різного діаметра		40 км
10GBase-LX4	IEEE 802.3ae	багатомодовий оптоволоконний кабель		240 (300)
		одномодовий оптоволоконний кабель		10 км
10GBase-SW	IEEE 802.3ae	багатомодовий оптоволоконний кабель, довжина хвилі 850 нм		300
10GBase-LW	IEEE 802.3ae	одномодовий оптоволоконний кабель, довжина хвилі 1310 нм		25 км
10GBase-EW	IEEE 802.3ae	одномодовий оптоволоконний кабель, довжина хвилі 1550 нм		40 км
10GBase-KR	IEEE 802.3ap	оптоволоконний кабель		
40G Ethernet (40GbE)				
40GBase- KR4	IEEE 802.3ba	для об'єднувальної плати		1 м
40GBase- CR4	IEEE 802.3ba	мідний біаксіальний кабель		7 м
40GBase-T	IEEE 802.3ba	UTP категорії 8		30 м
40GBase-SR4	IEEE 802.3ba	оптоволоконний кабель		100 м
40GBase-LR4	IEEE 802.3ba	оптоволоконний кабель		10 км
40GBase-FR4	IEEE 802.3bg	оптоволоконний кабель		2 км
100G Ethernet (100GbE)				
100GBase- KR10	IEEE 802.3ba	для покращеної об'єднувальної плати		1 м
100GBase- CR10	IEEE 802.3ba	мідний біаксіальний кабель		7 м
100GBase-SR10	IEEE 802.3ba	оптоволоконний кабель		125 м
100GBase-LR10	IEEE 802.3ba	оптоволоконний кабель		10 км
100GBase-ER10	IEEE 802.3ba	оптоволоконний кабель		40 км

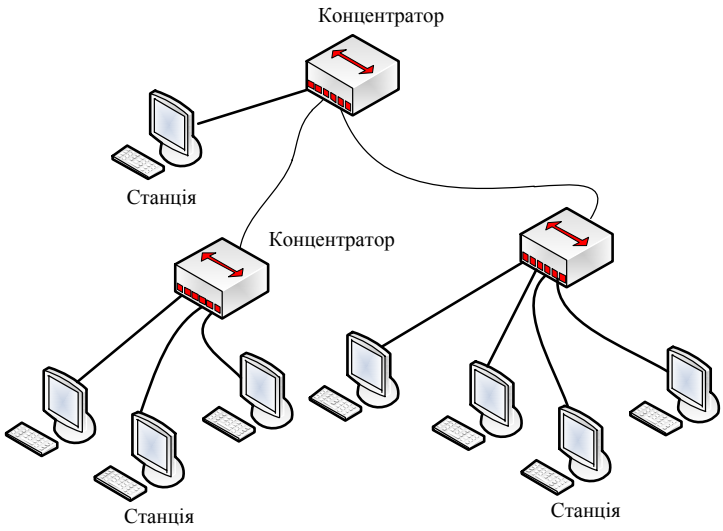


Рисунок 3.19 – Структура мережі з дворівневою деревоподібною топологією

Мережі Ethernet з пропускною спроможністю 100 Мбіт/с (Fast Ethernet). Стандарт Fast Ethernet було прийнятий у 1995 році і визначає як модифікації на основі скрученої пари, так і оптоволоконного кабелю (див. таблицю 3.5). У мережі специфікації 100Base-TX використовується кодування 4B/5B, а для 100Base-T4 – 8B/6T, останнє з яких дозволяє забезпечити не лише кращі електричні параметри передавання сигналів, але й їх стійку синхронізацію. Як фізичне кодування використовуються коди NRZ та MLT-3.

У Fast Ethernet, порівняно з мережами Ethernet, використовується той же метод доступу до середовища передавання і такий же формат кадру, але суттєвою відмінністю є повна її прозорість для протоколів TCP/IP та IPX, тобто для забезпечення роботи цих протоколів в мережах Fast Ethernet не потрібне додаткове програмне забезпечення, нові рівні трансляції тощо.

Стандарт Fast Ethernet містить і режим **автопереговорів**, який дозволяє підтримувати найбільш вигідний режим роботи двом з'єднаним станціям, що підтримують декілька стандартів фізичного рівня і розрізняються швидкістю передавання та кількістю скручених пар. Визначають таку пріоритетність режимів (в порядку зменшення пріоритетів):

- дуплексний режим 100Base-TX або 100Base-FX;
- 100Base-T4;
- 100Base-TX;
- дуплексний режим 10Base-T;
- 10Base-T.

Треба відзначити, що традиційно мережі Ethernet є мережами з напівдуплексним передаванням. Для забезпечення роботи в дуплексному режимі потрібно використовувати більш складні мережні адаптери, які мають працювати в такому режимі, а концентратор має бути комутувальним.

Топологія мережі Fast Ethernet аналогічна топології 10-мегабітових мереж, але підтримує спільну роботу локальних мереж з пропускною спроможністю і 10 Мбіт/с, і 100 Мбіт/с.

Мережі Gigabit Ethernet. Мережа Gigabit Ethernet, стандарт якої було прийнято в 1996 році, використовує таку ж технологію, що і Fast Ethernet, і є сумісною з мережами стандартів 10Base-T та 100Base-T, що значно полегшує перехід на більш високошвидкісні мережі. В стандарті Gigabit Ethernet визначено 4 варіанти, які використовують скручену пару та оптоволоконний кабель (див. табл. 3.5). Типова структура мережі, яка має декілька робочих груп, наведена на рис. 3.20. Комутатор локальної мережі при цьому забезпечує взаємодію центральних серверів і високошвидкісних комутаторів робочих груп, які підтримують різні швидкості передачі: від 100 Мбіт/с до декількох гігабітів.

Для підтримки потрібної швидкості передавання в кабелі категорії 5 (і вище) використовується кодування PAM5 (пункт 2.4.3), а також логічне кодування 8B/10B. І так, як і мережі Fast Ethernet, Gigabit Ethernet використовує процедуру автопереговорів.

Мережі 10 Gigabit Ethernet. Стандарт 10 Gigabit Ethernet на сьогодні – це найбільш швидкісний варіант технології Ethernet, який орієнтовано тільки на дуплексний режим передачі. Це перший стандарт Ethernet, який не поділяє середовище передавання і, крім того, містить специфікації фізичного рівня, які сумісні зі стандартами глобальних мереж, а саме: мереж SONET/SDH, які розглянуті в підрозділі 9.3.

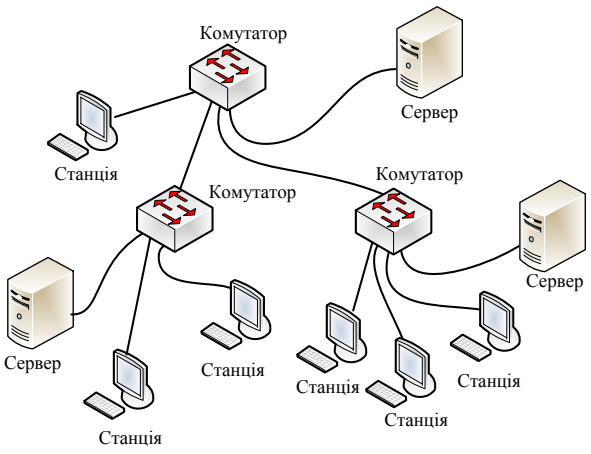


Рисунок 3.20 – Типова структура мережі Gigabit Ethernet

Перша версія стандарту 10 Gigabit Ethernet, орієнтована на використання оптоволоконного кабелю, прийнята у 2002 році, а в 2006 році – стандарт, в якому описується функціонування мережі на скрученій парі.

Існує декілька модифікацій стандарту 10G Ethernet (див. табл. 3.5), одна з яких (10GBase-T) передбачає використання скрученої пари категорії 6 (або 6а), всі інші – оптоволоконного кабелю.

У мережах стандарту 10GBase-LX4 використовується кодування 8B/10B, в усіх інших – 64B/66B, що дозволяє забезпечити стійку синхронізацію передавання та уникнути передавання довгих нульових та одиничних послідовностей.

На сьогодні основним призначенням мереж 10Gigabit Ethernet є не побудова локальних мереж, а організація високошвидкісних магістралей для регіональних і глобальних мереж, які об'єднують географічно віддалені локальні мережі. Такий підхід значно простіший (і дешевший), оскільки не вимагає складного і дорого в реалізації перетворення форматів даних і процедур передавання, але забезпечує необхідну якість обслуговування.

Мережі 40Gigabit Ethernet та **100Gigabit Ethernet**

За останні десять років безперервно збільшуються обсяги даних, які передаються в мережах, крім того, вимоги до швидкості передавання також безперервно ростуть. З'являється багато задач, які вимагають передавання все з більшою швидкістю. Саме тому перед розробниками була поставлена задача розробки технологій, які б дозволили передавати великі обсяги даних зі швидкостями в декілька сотень гігабітів і навіть терабітів.

Мережі 40GbE та 100GbE, розроблені групою IEEE 802.3ba Ethernet Task Force, є наступним етапом розвитку стандартів Ethernet. Вони забезпечують швидкість передавання даних відповідно 40 та 100 Гбіт/с, можуть використовувати оптоволоконний та біаксіальний кабелі, а також об'єднувальну плату.

Мережі 200Gigabit Ethernet

Такі мережі відповідають стандарту IEEE 802.3bs, пропонуючи декілька варіантів специфікацій і передбачають дуплексне передавання на різні відстані. Основні характеристики специфікацій:

- 200GBase-DR4 використовує 4 смуги одномодового оптоволокна та 4 рівні амплітудної модуляції, передавання на відстань 500 м;
- 200GBase-FR4 забезпечує передавання на 2 км і використовує таку ж систему і рівні модуляції, дані передаються по 4-х лініях WDM одномодового оптоволокна;
- 200GBASE-R використовує ті ж характеристик, що і 200GBase-FR4, передавання на відстань 10 км.

Мережі 400Gigabit Ethernet

Такі мережі також відповідають стандарту IEEE 802.3bs і забезпечують дуплексне передавання. Основні характеристики специфікацій:

- 400GBASE-SR16 забезпечує передавання на відстань 100 м з використанням кодування на 16 лініях багатомодового оптоволокна;
- 400GBASE-DR4 забезпечує передавання на відстань 500 м з використанням 4-рівневої модуляції на 4-х каналах одномодового оптоволокна;

- 400GBASE-FR8 –передавання на відстань 2 км з використанням 4-рівневої модуляції на 8-ми каналах CWDM-лініях одномодового оптоволокна;
- 400GBASE-LR8 забезпечує передавання на відстань 10 км з використанням 4-рівневої модуляції на 8-ми каналах WDM одномодового оптоволокна.

3.8 Технологія Token Ring

Мережа Token Ring (мережа з маркерним доступом) була розроблена компанією IBM і призначалась для об'єднання всіх робочих станцій, які випускаються цією компанією. Стандартами IEEE 802.5 та ISO 8802.5 (хоча між ними і є деякі несуттєві відмінності) регламентовані версії мережі з пропускною спроможністю 4 Мбіт/с та 16 Мбіт/с (TR4 і TR16 та зі значеннями MTU 4464 та 17914 байтів, відповідно), в стандарті IEEE 802.5t (1998 р.) описана мережа з пропускною спроможністю 100 Мбіт/с. І хоча на сьогодні мережа Ethernet є найбільш поширеною локальною мережею, яка витісняє всі інші типи, розробляється стандарт для TR, який регламентує передавання зі швидкістю 1 Гбіт/с. Компанією IBM спеціально для мережі Token Ring була розроблена концепція емулювальної програми NetBIOS, що дозволило більш гнучко реагувати на особливості апаратури і підтримувати сумісність із програмами більш високого рівня.

Мережа Token Ring значно складніша за Ethernet, що веде і до більшої вартості, і має логічну топологію типу «кільце» та фізичну топологію типу «зірка». Це пов'язано з тим, що окремі станції (і всі моделі) мережі підключаються до каналу не безпосередньо, а через спеціальні концентратори (або пристрої множинного доступу) MAU (Media Attachment Unit) або MSAU (Multi-Station Access Unit). На рис. 3.21 наведена структура мережі, причому використовуються MAU, які, зазвичай, мають 8 портів (хоча існують 16- та 24-портові концентратори) для підключення робочих станцій та інших модулів мережі і два порти (RI та RO) для підключення інших MAU. Всі порти мають нормально замкнутий стан, а при підключенні до будь-якого модуля цілісність кільця буде забезпечуватись через мережний адаптер підключеного вузла. Для передавання даних на фізичному рівні використовується манчестерське кодування.

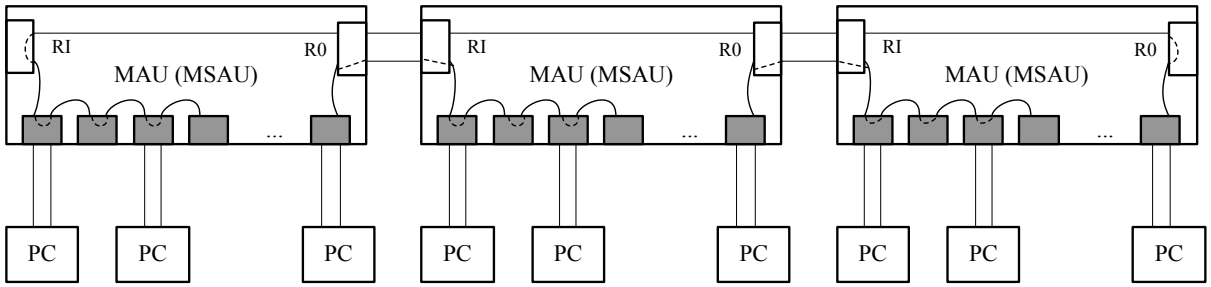


Рисунок 3.21 – Структура мережі Token Ring

У перших версіях мережі Token Ring функції MAU виконували пасивні концентратори IBM 8228 або активні IBM 8229. Для збільшення відстані між цими концентраторами використовуються, відповідно, пасивні IBM 8218 та активні IBM 8219 повторювачі. На сьогодні часто використовується пристрій керованого доступу CAU (Controlled Access Unit) IBM 8230, який являє собою високоінтелектуальний активний концентратор, до якого можна підключати до 4-х локальних мереж.

Для реалізації мереж стандарту IEEE 802.5 розроблено набір інтегральних схем TMS 380, який призначено для реалізації фізичного та канального рівнів. Цей набір забезпечує передавання даних, відео, мультимедіа застосувань з різними пріоритетами і містить схеми:

- TMS 38051 та TMS 38052 виконують функції приймання та передавання кадру, інтерфейсу з кабелем;
- TMS 38010 – спеціалізований мережний 16-розрядний контролер;
- TMS 38020 реалізує функції протоколу IEEE 802.5;
- TMS 38016 – 32-розрядний мережний контролер.

У мережах Token Ring можуть використовуватись такі кабелі: екрановані типу 1, 2, 6, 9; неекрановані категорій 3, 5; одномодові та багатомодові. Від характеристики кабельної системи, кількості і різновидів портів MAU, а також пропускної спроможності каналу залежать усі характеристики мережі, які наведені в табл. 3.6.

Таблиця 3.6 – Характеристики мережі Token Ring

Параметр	Середовище передавання							
	STP type 1, 2/6, 9		UTP category 3		UTP category 5		SM	MM
Швидкість передавання, Мбіт/с	4	16	4	16	4	16	4/16	4/16
Кількість станцій	250	250	150	150	150	150	250	250
Відстань RI/RO, м	770	350	200	100	250	120	10 км	2 км
Довжина радіального кабелю при пасивному MAU, м	200	100	100	60	130	85	10 км	2 км
Довжина радіального кабелю при активному MAU, м	300	150	200	100	250	120	10 км	2 км

З усіх станцій, які підключені до мережі, виділяють одну, що має функції **активного монітора** (Active Monitor), який виконує головну керівну функцію в кільці і підтримує виконання таких функцій:

- забезпечення роботи головного тактового генератора;
- контроль передавання маркера в мережі і, в разі його пошкодження або втрати, генерація нового маркера;

- передавання керівної інформації, необхідної для нормальної роботи всіх станцій кільця;
- забезпечення необхідних часових затримок в кільці;
- видалення з кільця кадрів для попередження зациклювання.

Активний монітор встановлюється під час ініціалізації кільця і зазвичай має найбільшу MAC-адресу, всі інші станції вважаються резервними моніторами (Standby monitor), кожний з яких при відмові активного монітора може стати активним (при виконанні відповідних процедур). Активний монітор періодично (через кожні 7 с) передає службовий кадр, який показує функціонування цього модуля в кільці, якщо ж за 16 с не було такого кадру, вважається, що активний монітор несправний і з резервних моніторів вибирається новий активний.

У мережах Token Ring використовуються три типи кадрів: маркера, даних та переривання, формати яких наведено на рис. 3.22 (розмір опцій наведено в байтах). **Кадр маркера** визначає права доступу до середовища передавання станції, яка його отримала. **Кадр даних** забезпечує передавання як даних для керування кільцем (MAC-кадри), так і даних користувача (LLC-кадри). Стандарт Token Ring визначає 25 типів керівних кадрів MAC-рівня, 6 з яких є основними. **Кадр переривання** – спеціальний керівний кадр, який може бути відправлений в будь-який момент і в будь-якому місці потоку даних, свідчить про те, що поточне передавання кадру або маркера відміняється.

ST (Starting Delimiter) – роздільник початку кадру, значення якого JK0JK000, де J та K – біти, значення яких не відповідає диференціальному кодуванню Manchester (пункт 2.4.3 даного посібника) і протягом перших двох бітових інтервалів JK утримується один рівень сигналу, протягом других JK – протилежний рівень, тому такий збій синхронізації легко виявляється модулем-отримувачем;

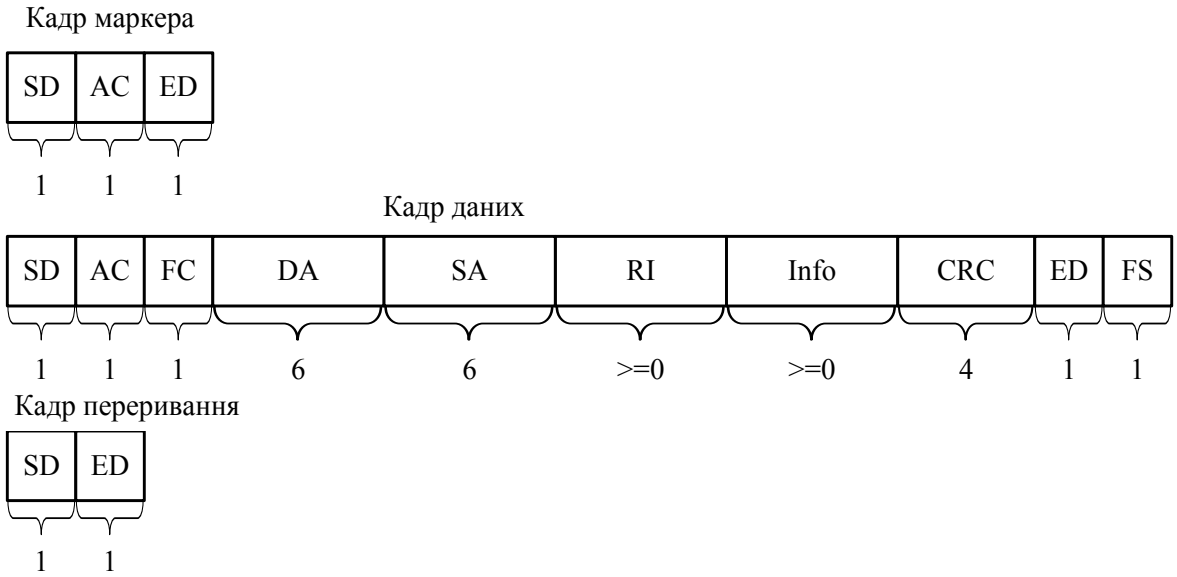


Рисунок 3.22 – Формати кадрів Token Ring

AC (Access Control) – керування доступом, значення якого PPPTMRRR визначає який з наступних вузлів отримає право на передавання, при цьому: біти PPP означають поточний пріоритет кадру (000 – найменший, 111 – найвищий); T – біт маркера (T = 0 для маркера, T = 1 для маркера з кадром); M – біт монітора (якщо даний кадр передано активним монітором, M = 1); RRR – біти резервованого пріоритету (дозволяють абоненту резервувати своє право подальшого захоплення каналу);

ED (Ending Delimiter) – роздільник кінця кадру, значення якого JK1JK1IE, де JK мають такий же стан, що і в ST, біт I (Intermediate) визначає, є даний кадр проміжним в послідовності кадрів (I = 1) або кінцевим/єдиним в повідомленні (I = 0), а біт E (Error) вказує на виявлену помилку (E = 1);

FC (Frame Control) – керування кадром, формат якого FFZZZZZZ, біти FF визначають тип кадру: 00 – MAC-кадр, тобто кадр зі службовою інформацією, 01 – LLC-кадр, тобто кадр даних користувача, 10 та 11 – зарезервовані; біти ZZZZZZ для LLC-кадрів зберігають рівень його пріоритету, а для MAC-кадрів визначають один з його 25 типів;

DA (Destination Address) – адреса призначення (6-байтова MAC-адреса отримувача, в якій старший біт адреси визначає отримувача: 0 – індивідуальна, 1 – групова, а другий біт вказує на спосіб призначення адреси: 0 – глобальний (універсальний, який записано в адаптері), 1 – локальний), яка може бути:

- індивідуальною;
- груповою;
- функціональною, яка призначається модулям, що виконують спільні функції і які або заздалегідь визначають архітектуру TR, або призначаються користувачем залежно від його потреб (наприклад, адреси, які зарезервовано для відповідних службових цілей:
  FF FF FF FF FF FF – широкомовний кадр;
  C0 00 FF FF FF FF – широкомовний MAC-кадр;
  C0 00 00 00 00 01 – активний монітор;
  C0 00 00 00 00 02 – сервер параметрів кільця;
  C0 00 00 00 00 08 – монітор помилок кільця;
  C0 00 00 00 00 10 – сервер звітів про конфігурацію;
  C0 00 00 00 01 00 – сервер мостового з’єднання;
  C0 00 00 00 20 00 – системний адміністратор;
  а також адреси серверів, які призначаються користувачами:
  C0 00 00 80 00 00 та C0 00 04 00 00 00);

SA (Source Address) – MAC-адреса відправника;

RI (Routing Information) – маршрутна інформація, що використовується тільки в мережах, які містять декілька сегментів TR, призначена для ідентифікації кілець і мостів (що використовуються для об’єднання маркерних кілець), через які необхідно передати кадр адресату; містить послідовність 2-байтових адрес сегментів на маршруті до адресата, які керують роботою

мостів в режимі маршрутизації від джерела (при цьому старший біт в адресі відправника дорівнює 1);

Info (Information) – інформаційне поле, що містить дані верхніх рівнів (кадр LLC), в яких наведені вказівники доступу до сервісів відправника та отримувача (див. рис. 3.12), або службові дані (кадр MAC), причому розмір цього поля залежить від версії TR і складає, зазвичай, 4 Кбайти для TR4 і 16 Кбайтів для інших.

CRC (Cyclic Redundancy Check) – байти контролю за циклічним кодом CRC-32;

FS (Frame Status) – статус кадру, формат якого ACrrACrr (однобайтове поле, яке для надійності містить дві однакові тетради), де: r – зарезервовані біти, A (Address Recognized) – біт розпізнавання адреси, який вказує на те, що отримувач є в сегменті, C (Frame Copied) – біт копіювання кадру, який визначає, що адресат скопіював даний кадр в свій буфер; це поле дозволяє відправнику отримати інформацію, що переданий кадр було отримано.

У мережі Token Ring на підрівнях LLC та MAC використовуються процедури без встановлення з`єднання, але з підтвердженням отримання переданих кадрів. Для доступу до середовища передавання використовується класичний маркерний доступ, тобто в кільці мережі постійно передається маркер, який визначає право станції, що його отримала, передати свої кадри даних. У мережах TR використовується модифікація цього методу, а саме: маркерний доступ з пріоритетами, при реалізації якого визначаються пріоритети кадрів маркера та маркера з даними, які передаються в каналі, а також поточних кадрів, які очікують на передавання в станціях. При цьому можуть виникнути нижчеописані ситуації.

Станція, яка отримала маркер, не має інформації для передавання. В такому випадку станція передає маркер наступній станції без зміни значення пріоритету.

Станція, яка отримала маркер, має інформацію для передавання. При цьому станція порівнює пріоритет свого кадру, підготовленого до передавання, з пріоритетом P отриманого кадру. Якщо він нижчий за P, то станція виконує друге порівняння з бітами резервованого пріоритету R. В разі якщо він менший і цього значення, то отриманий кадр буде передано наступній в кільці станції без зміни значення R. Якщо ж пріоритет кадру для передачі вищий за значення R, то станція встановлює його нове значення, що означає заявку на можливість наступного передавання свого кадру. Якщо ж пріоритет кадру, підготовленого до передавання, вищий за пріоритет отриманого кадру, станція записує в свій буфер значення P та R з отриманого кадру і передає кадр з пріоритетом P, а біти R встановлює в 0, що дозволяє іншим станціям конкурувати за право отримати доступ до кільця. Після повернення по кільцю переданого кадру станція, яка відправила раніше кадр, відновлює записані значення P та R і передає його в канал. Реалізація такого алгоритму обробки пріоритетів кадрів дозволяє зрівняти шанси всіх станцій на доступ до середовища передавання.

При виявленні некоректної роботи станції, її найближчої сусідньої станції або кабелю з'єднання станція формує і видає в канал сигнальний кадр попередження (MAC-кадр), який адресується всім станціям кільця. Використання такої процедури дозволяє локалізувати несправність до рівня, який називається доменом несправності і складається з робочої станції, її активної станції-сусіда та кабелю між ними.

Таким чином, важливою перевагою мереж Token Ring є відсутність нестабільної роботи, оскільки конфліктних ситуацій не виникає, що веде до значно більшої ефективності використання ресурсів каналу. До недоліків таких мереж треба віднести складність алгоритмів керування і мережних адаптерів, що призводить до значно більш високої вартості, ніж Ethernet, необхідність контролю наявності маркера в мережі, а також залежність функціонування мережі від кожного з модулів мережі.

3.9 Мережі FDDI

Технологія **FDDI** (Fiber Distributed Data Interface) – перша технологія локальних мереж, яка використовувала оптоволоконний кабель як середовище передавання, розроблена на основі принципів доступу до каналу мережі Token Ring. Мережа FDDI створюється на основі двох кілець: первинного (primary) та вторинного (secondary), причому обов'язково інформація в них передається в протилежних напрямках (рис. 3.23).

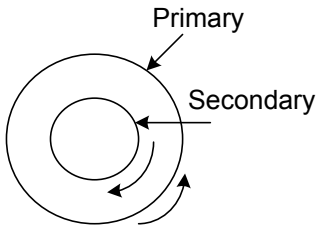


Рисунок 3.23 – Топологія мережі FDDI

Такий підхід дозволяє при відмові будь-якого модуля мережі або при фізичному розриві мережного кабелю виконати динамічну реконфігурацію топології, відновлюючи таким чином коректне функціонування мережі. При реконфігурації (згортанні кільця) первинне кільце об'єднується з вторинним за допомогою концентраторів та (або) мережних адаптерів FDDI. Технологія має такі особливості:

- передавання даних зі швидкістю 100 Мбіт/с (для двох кілець 200 Мбіт/с) при довжині кільця 100 км;
- можливість підключення до 500 станцій в одному кільці;
- ефективне використання пропускної спроможності мережі як для асинхронного, так і для синхронного трафіків;
- забезпечення відмовостійкості мережі за рахунок стандартних процедур реконфігурації топології після відмов різних типів.

І первинне, і вторинне кільця являють собою спільне розподілене середовище передавання даних, для якого визначено метод маркерного доступу, що розроблений на базі методу доступу мережі Token Ring. Відмінності в цих методах доступу полягають у:

- відсутності активного монітора, передбачається, що всі станції рівноправні і постійно контролюють інтервал між маркерами (токенами), та наявності фізичного з'єднання між сусідніми портами;
- відсутності процедур обробки пріоритетів кадрів (замість 8-рівневого пріоритету кадрів весь трафік мережі розділяють на 2 класи), що значно спрощує алгоритми обробки потоків;
- тому, що для асинхронного режиму, для якого несуттєві затримки, час утримання маркера в станції залежить від завантаженості кільця (при незначній завантаженості кільця час утримання маркера збільшується, при збільшенні навантаження – зменшується), для синхронного режиму час утримання маркера залишається постійним (як і в мережах Token Ring) і тому такий трафік обслуговується навіть при перевантаженні кільця;
- використанні алгоритму раннього вивільнення маркера, який передбачає, що новий маркер формується станцією одразу після передавання нею кадру (в мережі Token Ring станція звільнює маркер після того, як він буде переданий через весь сегмент).

Враховуючи топологію мережі FDDI розрізняють такі компоненти мережі:

- SAS (Single Access Station) – станції одиничного підключення, які підключаються тільки до первинного кільця (станції класу B);
- DAS (Dual Access Station) – станції подвійного підключення, які одночасно можуть підключатися до двох магістральних кілець: і первинного, і вторинного (станції класу A);
- SAC (Single Access Concentrator) – концентратори одиничного підключення;
- DAC (Dual Access Concentrator) – концентратори подвійного підключення.

Ці станції та концентратори мають 4 різних типи портів (табл. 3.7), які відрізняються за своїм призначенням та особливостями підключення.

Особливості реалізації цих модулів наведені на рис. 3.24. Кожний вузол мережі має мати мінімум один модуль доступу до середовища MAC, який має свою унікальну MAC-адресу. Станції одиничного підключення SAS, які зазвичай підключаються до кільця за допомогою концентратора, мають один порт S, який працює як на прийом, так і на передавання (рис. 3.24, а). Станції подвійного підключення DAS мають два порти для підключення до двох кілець і дозволяють одночасно передавати та приймати дані по двох напрямках (рис. 3.24, б). Концентратори одиничного підключення мають тільки порти M для підключення станцій SAS, а концентрато-

ри DAC (рис. 3.24, в) мають три типи портів, які дозволяють станціям SAS та DAS підключатися до двох кілець мережі.

Таблиця 3.7 – Типи портів станцій і концентраторів FDDI

Тип порту	Підключення	Призначення
A	PI/SO (Primary In/Secondary Out) Вхід первинного кільця і вихід вторинного кільця	Забезпечує з'єднання модулів подвійного підключення з магістральними кільцями
B	PO/SI (Primary Out/Secondary In) Вихід первинного кільця і вхід вторинного кільця	Забезпечує з'єднання модулів подвійного підключення з магістральними кільцями
M	Master PI/PO Вхід первинного кільця/вихід первинного кільця	Порт концентратора для зв'язку з пристроями одиничного підключення
S	Slave PI/PO Вхід первинного кільця/вихід первинного кільця	Порт модуля одиничного підключення для зв'язку з концентратором

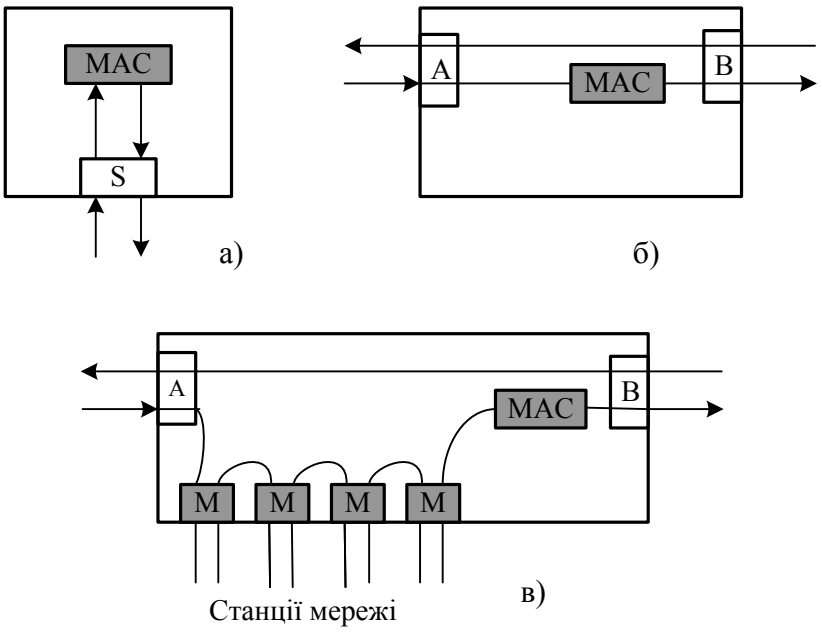


Рисунок 3.24 – Компоненти мережі FDDI: станції одиничного підключення (а), подвійного підключення (б) та концентратори (в)

Для того, щоб станція могла передавати свої дані в канал, а не тільки виконувати ретрансляцію кадрів інших станцій, вона має мати хоча б один модуль MAC. Зазвичай цій умові відповідають всі станції мережі FDDI і концентратори, які використовують модуль MAC для генерації таких службових кадрів, як кадри ініціалізації кільця, кадри діагностики (пошуку несправностей), кадри захоплення каналу тощо.

На сьогодні лише технологія FDDI дає можливість реконфігурації топології при несправностях на фізичному рівні. Концентратори продов-

жують функціонування при відключенні станції від портів або при обриві кабелю.

Компоненти фізичного і канального рівнів мереж FDDI показані на рис. 3.25.

Фізичний підрівень MIC визначає тип та порт з'єднання з оптичним середовищем передавання, а підрівень **PMD**, який залежить від середовища передавання, визначає електричні та оптичні компоненти, тип зв'язку з середовищем, вимоги до багатомодового кабелю тощо. Підрівень **PHY** виконує кодування відповідно до коду NRZ, а підрівень MAC забезпечує керування доступом станції до середовища передавання, виконує логічне кодування 4B/5B тощо. Засоби керування станцією **SMT** виконують моніторинг та контроль роботи всього кільця, виявлення помилок тощо.

Таким чином, технологія FDDI забезпечує високошвидкісне передавання в каналі на великі відстані, можливість реконфігурації топології, завдяки чому забезпечується її відмовостійкість, але призводить до значного ускладнення структури модулів мережі та процедур їх функціонування. Тому мережі FDDI зазвичай використовують не для побудови локальних мереж, а для створення високошвидкісних магістралей, до яких підключаються окремі станції (зазвичай через концентратори) та локальні мережі.

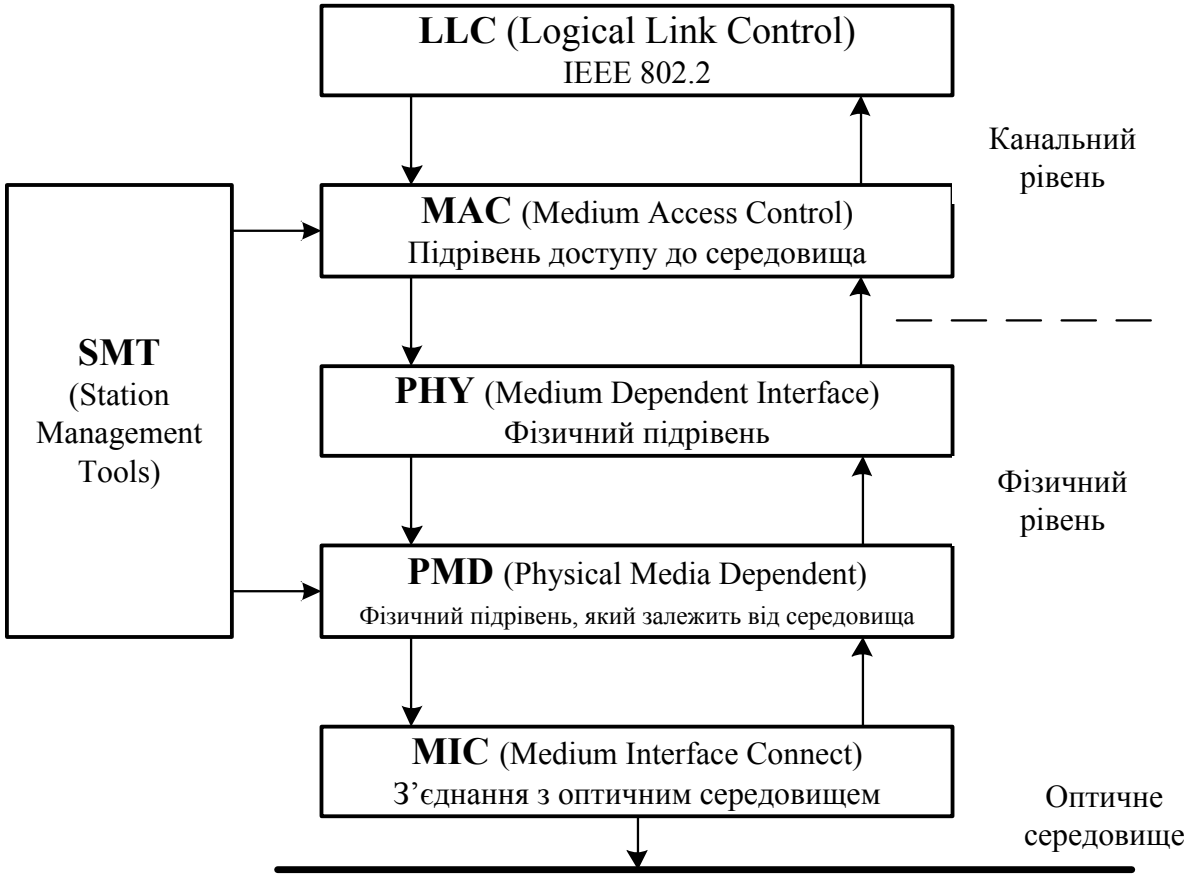


Рисунок 3.25 – Компоненти рівнів моделі OSI в мережі FDDI

3.10 Основи функціонування комутаторів локальних мереж

Всі станції, зв'язані одним загальним середовищем передавання, в мережах Ethernet складають домени колізій. Мережі, побудовані виключно на концентраторах (хабах), складають один домен колізій. Практично в невеликих комп'ютерних мережах (до 20 вузлів) немає умов для виникнення шторму колізій – існуючий механізм доступу до середовища передавання дозволяє запобігти зіткненням кадрів. Проте, якщо до мережі підключається більша кількість станцій, то можлива ситуація, коли протягом короткого інтервалу часу мережа припиняє функціонувати, і після цього робота мережі відновлюється без якого-небудь втручання. Найбільш ймовірна причина даної ситуації – шторм колізій. Для неможливості виникнення шторму колізій необхідно розділити домен колізій на домени менших розмірів, зв'язаних за допомогою комутатора. Оскільки кожен домен містить меншу кількість вузлів в кожному домені колізій, зменшується ймовірність виникнення шторму колізій.

Таким чином, використання комутаторів, які працюють як вузол комутації, приймаючи та передаючи кадри між різними парами абонентів одночасно, дозволяє зменшити ймовірність виникнення колізій і за рахунок цього підвищити ефективність мереж Ethernet. Локальні мережі, побудовані на основі комутаторів, називаються Switch Ethernet.

Залежно від функцій, які виконують комутатори, та їх відповідності рівням еталонної моделі OSI, розрізняють комутатори 2-го та 3-го рівнів. **Комутатори 2-го рівня** функціонують на канальному рівні і передають отриманий кадр тільки на один вихідний порт, який визначається з урахуванням MAC-адреси одержувача. В цей же час інші порти комутатора можуть використовуватись для передавання між іншою парою станцій. Таким чином, реальна пропускна спроможність мережі може перевищувати максимальну швидкість передавання даних з кожної станції.

Для передавання кадрів комутатор використовує **таблицю просування (або адресну таблицю)**, яка містить два поля: MAC-адресу станції-одержувача та ідентифікатор порту комутатора, на який необхідно передати кадр з такою адресою призначення. В початковому стані ці поля таблиці порожні. Комутатор заповнює свою таблицю автоматично, на основі пасивного спостереження на трафіком, який циркулює в сегментах, що підключені до його портів.

Якщо комутатор не може знайти адресу одержувача в своїй адресній таблиці, він направляє кадр на всі порти за виключенням того, з якого отримано кадр. Максимальний розмір таблиці просування є ознакою конкретного комутатора і може бути від кількох десятків адрес (комутатори рівня малого офісу) до сотен тисяч (магістральні комутатори). Крім того, деякі типи комутаторів обмежують кількість адрес, які можуть бути зареєстровані для кожного порту.

Комутатор може працювати в двох основних режимах:

- прозорого з'єднання (Transparent Bridging);
- термінової комутації (Express Switching).

При **прозорому з'єднанні** розмір адресної таблиці має максимальне значення, оскільки містить MAC-адреси всіх станцій локального сегменту, в тому числі і тих, що безпосередньо не приєднані до комутатора. Якщо розмір адресної таблиці менший за кількість пристроїв в локальному сегменті, то трафік, спрямований на MAC-адреси, що не вмістились в адресну таблицю, буде передаватись на всі порти комутатора.

Особливостями комутатора, що працює на основі способу **термінової комутації**, є такі:

- один з портів визначають як магістральний;
- комутатор керує його адресною таблицею таким же чином, як і при прозорому з'єднанні, проте він не запам'ятовує адреси відправників тих кадрів, що надійшли через магістральний порт;
- комутатор передає кадри з відомими адресами одержувача (зареєстрованими в адресній таблиці);
- кадри з невідомою адресою одержувача замість того, щоб передавати на всі порти, він передає на магістральний порт;

Кожна станція, підключена (безпосередньо або ні) до комутатора, що працює в режимі термінової комутації, має спочатку передати кадр для того, щоб комутатор запам'ятав і записав її адресу.

В обох описаних вище способах комутатор ніколи не передає кадр на порт, з якого він був отриманий.

Існує декілька методів комутації, які реалізовано в комутаторах 2-го рівня:

- наскрізна (cut-through) або комутація «на льоту»;
- безфрагментна (fragment-free processing);
- з буферизацією кадрів (store-end-forward processing);
- інтелектуальна комутація.

При використанні **наскрізної комутації** комутатор починає відправляти кадр на відповідний вихідний порт каналу одразу, як тільки отримана інформація, необхідна для визначення адресата, тобто після отримання і аналізу перших 6 байтів кадру, в яких міститься MAC-адреса одержувача. Це веде до значного скорочення затримки передавання, але не дозволяє виявити кадри з помилками, які можуть бути виявлені тільки після отримання всього кадру. Крім того, після отримання тільки 6 байтів комутатор не може виявити колізію, що призводить до збільшення їх кількості між доменами колізій.

Єдина відмінність **безфрагментної комутації** від наскрізної полягає в тому, комутатор на розпочинає подальше передавання кадру поки не отримає перші 64 байти, що є (згідно зі специфікацією Ethernet) гарантією того, що колізії в сегменті не буде. Таким чином, даний метод комутації не

збільшує число колізій, проте, порівняно з попереднім методом, призводить до більшої затримки передавання.

При передаванні **з буферизацією кадрів** комутатор отримує весь кадр, перевіряє його на наявність помилок і тільки потім передає на відповідний порт на основі записів в адресній таблиці просування. Даний метод, порівняно з двома попередніми, запобігає збільшенню кількості колізій і передаванню пошкоджених кадрів, проте затримка, яку вносить такий комутатор, може бути значною і досягати значення часу, необхідного для прийому та передавання всього кадру.

Інтелектуальна комутація є комбінацією комутації «на льоту» і комутації з буферизацією кадрів. Спочатку комутатор використовує комутацію «на льоту», і якщо кількість відмов при передачі кадрів перевищує деяке встановлене значення, комутатор автоматично починає відправляти отримані кадри на основі методу з буферизацією. Якщо число відмов падає нижче даного значення, комутатор починає виконувати комутацію «на льоту».

Таким чином, комутатори 2-го рівня дозволяють передавати потоки даних значного розміру, проте, зі збільшенням кількості станцій, які підключені до мережі, вони вже не можуть виконувати свої функції в повному обсязі. У мережі з великою кількістю станцій користувачів це може призвести до значного перевантаження або до виникнення широкомовного шторму (broadcast storm).

Комутатори 3-го рівня дозволяють частково розв’язати цю проблему. Розрізняють пакетні та потокові комутатори 3-го рівня. **Пакетні комутатори (packet-by-packet switch)** функціонують аналогічно маршрутизатору, однак, оскільки всі його функції реалізовано апаратно, продуктивність мережі значно збільшується. При використанні **потокового комутатора (flow-based switch)** продуктивність збільшується за рахунок ідентифікації потоків IP-пакетів з однаковими адресами відправника і одержувача. Це реалізується шляхом спостереження за трафіком, який передається через комутатор, або за допомогою спеціального ідентифікатора потоку в заголовку пакета. Після ідентифікації конкретного потоку для прискорення доставки пакета зазвичай обирається маршрут, який вже був раніше визначено, що веде до збільшення продуктивності мережі.

3.11 Питання для самоперевірки

1. Охарактеризуйте призначення канального рівня та проаналізуйте основні функції протоколів цього рівня.
2. Поясніть значення параметра MTU та його вплив на пропускну спроможність каналу.
3. Проаналізуйте етапи функціонування протоколів канального рівня та типи примітивів.
4. Поясніть особливості класифікації протоколів канального рівня.

5. Проаналізуйте характеристики байт-орієнтованих та біт-орієнтованих протоколів канального рівня.

6. Поясніть особливості функціонування байт-орієнтованих та біт-орієнтованих протоколів канального рівня.

7. Проаналізуйте особливості реалізації режимів роботи біт-орієнтованого протоколу HDLC.

8. Проаналізуйте особливості конфігурування каналу в протоколі HDLC.

9. Охарактеризуйте формати кадрів протоколу HDLC та особливості їх використання.

10. Охарактеризуйте способи нумерації кадрів протоколу HDLC і особливості та сфери їх використання.

11. Поясніть способи забезпечення прозорості каналу в протоколах BSC та HDLC.

12. Покажіть структуру 11-го кадру при передаванні за допомогою протоколу HDLC на станцію з адресою 7F в напівдуплексному режимі у прозорому каналі такої послідовності даних: 11111111 10110101 00000011 0111110.

13. Проаналізуйте особливості та сфери використання процедур керування передаванням в каналі (процедури SAW, GBN, SR).

14. Покажіть процедуру передавання даних в напівдуплексному каналі в режимі нормальної відповіді за допомогою протоколу HDLC. Передавання виконується блоками з 12 кадрів. При передаванні п’ятого та восьмого кадрів відбулася помилка.

15. Покажіть процедуру передавання даних в дуплексному каналі в режимі асинхронної відповіді за допомогою протоколу HDLC. Для керування потоком кадрів з розширеною нумерацією використовується алгоритм вибіркового повторення. Кадри з номерами 2 та 6 отримано з помилкою. Затримка одержання кадру підтвердження становить 3 кадри.

16. Охарактеризуйте принцип роботи «ковзного вікна». Поясніть від яких параметрів залежить розмір вікна та яким чином він впливає на ефективність передавання.

17. Поясніть принципи кодування з використанням циклічних кодів. Генерувальний поліном та його види.

18. Охарактеризуйте особливості реалізації канального рівня в локальних мережах.

19. Дайте загальну характеристику структури стандартів IEEE 802.x та її компонентів.

20. Проаналізуйте особливості реалізації підрівня управління логічним каналом та процедур його функціонування.

21. Поясніть принципи взаємодії протоколів підрівнів LLC та MAC. Які опції обов’язково мають бути в заголовках LLC та MAC?

22. Дайте загальну характеристику методів доступу до середовища передавання даних.

23. Проаналізуйте особливості використання та типи детермінованих методів доступу до середовища передавання.

24. Проаналізуйте особливості використання та типи недетермінованих (випадкових) методів доступу до середовища передавання.

25. Поясніть особливості функціонування та реалізації методів CSMA/CD та CSMA/CA.

26. Охарактеризуйте метод бітової карти та особливості його функціонування.

27. Охарактеризуйте метод доступу до середовища передавання, який використовується в мережі Ethernet.

28. Чим визначається мінімальний та максимальний розміри кадру, який використовується в мережах Ethernet?

29. Охарактеризуйте методи логічного та фізичного кодування, які використовуються в мережах Ethernet.

30. Проаналізуйте типи кадрів мережі Ethernet, їх можливості та особливості використання.

31. Які типи мережі Ethernet вам відомі? Проаналізуйте їх характеристики та особливості структури.

32. Які типи мережі Fast Ethernet вам відомі? Проаналізуйте їх характеристики та особливості структури.

33. Проаналізуйте характеристики та особливості структури Gigabit Ethernet.

34. Які типи мереж 10 Gigabit Ethernet вам відомі? Проаналізуйте їх характеристики та особливості структури.

35. Охарактеризуйте перспективи розвитку технології Ethernet.

36. Поясніть особливості структурної організації мереж Token Ring. Різниця між фізичною та логічною топологіями мережі.

37. Поясніть функції активного монітора та призначення й структуру кадрів, які використовуються в мережах Token Ring.

38. Проаналізуйте особливості структурної організації мереж FDDI та можливості динамічної реконфігурації.

39. Поясніть особливості підключення компонентів мережі до первинного та вторинного кілець.

40. Проаналізуйте функції компонентів канального і фізичного рівнів в мережі FDDI.

41. Проаналізуйте особливості функціонування комутаторів 2-го та 3-го рівнів.

42. Поясніть особливості методів комутації, які використовуються в комутаторах 2-го рівня.

43. Порівняйте особливості функціонування пакетних і потокових комутаторів. До якого класу комутаторів їх відносять?

44. Поясніть особливості наскрізної та безфрагментної комутацій та їх вплив на характеристики мережі.

4 МЕРЕЖНИЙ РІВЕНЬ

4.1 Адресація комп'ютерів на мережному рівні на прикладі IP-адресації

У кожного хоста та маршрутизатора в Internet є IP-адреса, що складається з двох логічних частин: ідентифікаторів мережі і хоста. У заголовку IP-пакета для зберігання IP-адрес відводяться 2 поля довжиною по 4 байти (32 біти).

Найчастіше IP-адреси записують у вигляді чотирьох десяткових чисел, що є значеннями кожного байта, розділених крапками, наприклад, 192.168.0.1. Дану адресу можна подати у двійковому форматі 11000000 10101000 00000000 00000001, або шістнадцятковому C0.A8.00.01.

Треба відзначити, що запис адреси не передбачає спеціального розмежувального знака між ідентифікаторами мережі і хоста. Але при маршрутизації виникає необхідність розділити адресу на ці дві частини. Для цього використовуються такі підходи:

- використання фіксованої межі. При цьому 32-бітове поле адреси завчасно ділиться на дві частини фіксованої довжини, в одній з яких завжди розміщується ідентифікатор мережі, в іншій – вузла. Але в даному випадку, оскільки поле, що відводиться для зберігання номера вузла, має фіксовану довжину, всі мережі будуть мати однакове максимальне число вузлів. Такий підхід не дозволяє диференційно використовувати адресний простір, тому сьогодні майже не використовується;
- використання маски, що дозволяє максимально гнучко встановлювати межі між ідентифікаторами мережі й вузла. Тут адресний простір можна використовувати для створення багатьох мереж різного розміру. Маска – це 32-бітове число, що використовується разом з IP-адресою, причому двійковий запис маски містить неперервну послідовність одиниць в тих розрядах, що мають в IP-адресі інтерпретуватися як номер мережі. Межа між послідовностями одиниць і нулів у масці відповідає межі між ідентифікаторами мережі і вузла в IP-адресі. Кількість одиниць в масці називають префіксом. Наприклад, масці 255.0.0.0 відповідає префікс 8;
- використання класів адрес (повнокласова адресація). Вводиться п'ять класів адрес: A, B, C, D, E. Три з них – A, B, C – використовуються для адресації мереж, а два – D, E – мають спеціальне призначення. Для кожного класу мережних адрес визначено положення межі між номером мережі і номером вузла.

4.1.1 Класи IP-адрес. Ознакою, на основі якої IP-адреси відносять до того чи іншого класу, є значення кількох перших бітів адреси. У табл. 4.1 і на рис. 4.1 наведено структуру IP-адрес різних класів.

Таблиця 4.1 – Класи IP-адрес

Клас	Перші біти	Найменший – найбільший номери мережі	Маска	Кількість вузлів у мережі
A	0	1.0.0.0–126.0.0.0 (0 – не використовується, 127 – зарезервовано)	255.0.0.0	2^{24}–2, поле 3 байти
B	10	128.0.0.0–91.255.0.0	255.255.0.0	2^{16}–2, поле 2 байти
C	110	192.0.0.0–23.255.255.0	255.255.255.0	2^{8}–2, поле 1 байт
D	1110	224.0.0.0–239.255.255.255		Групові адреси
E	11110	240.0.0.0–247.255.255.255		Зарезервовано

Рисунок 4.1 – Структури IP-адрес різних класів

До **класу A** відносять адреси, в яких старший біт має значення 0. В адресах класу A під ідентифікатор мережі відводиться 1 байт, а інші 3 байти інтерпретуються як номер вузла у мережі. Мережі, всі IP-адреси яких мають значення першого байта в діапазоні від 1 (00000001) до 126 (01111110), відносять до мереж класу A. Значення 0 (00000000) першого байта не використовується, а значення 127 (01111111) зарезервовано як адреса зворотної петлі (loopback). IP-адреси, перший октет яких дорівнює 127, використовуються для тестування програм, організації роботи клієнтської та серверної частин програмного забезпечення, що встановлене на одному комп'ютері.

До **класу B** відносять всі адреси, старші два біти яких мають значення 10. У даних адресах під ідентифікатор мережі та вузла відводиться по 2 байти.

До **класу C** відносять всі адреси, старші 3 біти яких мають значення 110. В адресах класу C під ідентифікатор мережі відводиться 3 байти, а під ідентифікатор вузла – 1 байт.

Якщо адреса мережі починається з послідовності 1110, то вона належить **класу D** і визначає особливу групову адресу (multicast address). Групова адреса ідентифікує групу мережних інтерфейсів, які, в загальному випадку, можуть належати різним мережам. Інтерфейс, що входить у групу, разом зі звичайною індивідуальною IP-адресою отримує ще одну, групову. Якщо при відправленні пакета адресою отримувача вказана адреса класу D, то такий пакет має бути доставлений всім вузлам, які входять в групу. Один хост може входити у кілька груп. В загальному випадку члени групи можуть знаходитися у різних мережах, що розташовані на великих відстанях. Групова адреса не ділиться на ідентифікатор мережі та вузла і опрацьовується маршрутизатором особливим чином. Основне призначення групової адреси – поширення інформації за схемою «один до багатьох».

Якщо адреса починається з послідовності 11110, то дана адреса належить **класу E**. Адреси цього класу зарезервовано для майбутнього використання.

Щоб отримати з IP-адреси ідентифікатор мережі і вузла, потрібно розділити адресу на дві відповідні частини і доповнити кожну з них нулями до повних 4 байтів. Наприклад, для адреси класу A 10.120.200.1 перший байт ідентифікує мережу, а останні три – вузол. Таким чином, ідентифікатором мережі є адреса 10.0.0.0, а вузла – адреса 0.120.200.1.

При призначенні IP-адрес ідентифікатор мережі і вузла не можуть складатися лише з одних двійкових нулів й одиниць. Тому у табл. 4.1 максимальна кількість вузлів зменшена на 2, адреси 0 і 255 заборонено використовувати для адресації мережних інтерфейсів.

Тут також потрібно пам’ятати, якщо у полі «номер мережі» стоять лише нулі, то за замовчуванням вважається, що вузол призначення належить тій же мережі, що і вузол-відправник. Така адреса може бути використана лише як адреса відправника. Якщо всі двійкові розряди IP-адреси дорівнюють 1, то пакет з такою адресою призначення має розсилатися всім вузлам, що знаходяться у тій же мережі, що і відправник пакета. Така адреса називається обмеженою широкомовною (limited broadcast). Обмеженість означає, що пакет не вийде за межі даної мережі. Якщо ж поле адреси отримувача у розрядах, що відповідають номеру хоста, містить лише 1, то пакет, що має таку адресу, розсилається всім вузлам мережі, номер якої вказано в адресі отримувача. Наприклад, пакет з адресою 192.168.1.255 буде відправлено всім вузлах мережі 192.168.1.0. Такий тип адреси називається широкомовним (broadcast). При цьому треба розуміти, що у протоколі IP нема розуміння широкомовлення у тому сенсі, як воно використовується в протоколах канального рівня, коли дані мають бути доставлені всім вузлах мережі. Як обмежене, так і звичайне широкомовне

розсилання мають межі поширення у підмережі: вони обмежуються або мережею, до якої належить відправник пакета, або мережею, ідентифікатор якої вказано як отримувача. Тому поділ мережі на частини за допомогою маршрутизаторів локалізує широкомовний шторм межами однієї з підмереж лише тому, що немає способу адресувати пакет одночасно всім вузлам всіх підмереж мережі.

4.1.2 Використання масок при IP-адресації. Якщо разом з IP-адресою використовувати маску, то це дозволить виконувати адресацію більш гнучко й відмовитися від класової адресації. При цьому частину мережі називають підмережею. Для виділення підмережі маршрутизатору необхідно «накласти» маску підмережі на IP-адресу, що дозволить виділити в адресі ідентифікатор мережі, підмережу і хост.

Наприклад, нехай для IP-адреси 172.16.155.5 задано маску 255.255.192.0. Двійковий вигляд відповідно такий:

- IP-адреса – 10101100.00010000.10011011.00000101;
- маска – 11111111.11111111.11000000.00000000.

Якщо ігнорувати маску й інтерпретувати адресу 172.16.155.5 на основі класів, то ідентифікатором мережі є 172.16.0.0, а вузла – 0.0.155.5 (оскільки адреса належить до класу B).

Якщо використовувати маску, то 18 послідовних одиниць в масці 255.255.192.0 після «накладання» на IP-адресу 172.16.155.5 ділять її на дві частини:

- ідентифікатор мережі – 10101100.00010000.10;
- ідентифікатор вузла – 011011.00000101.

У десятковій формі запис номера мережі і вузла після доповнення до 32 бітів мають, відповідно, такий вигляд: 172.16.128.0 і 0.0.27.5.

Виділення за допомогою маски адреси мережі та хоста можна інтерпретувати як виконання логічної операції І (AND).

Для запису масок використовуються також такі формати запису:

- шістнадцяткові коди: FF.00.00.00 – маска для адрес класу A;
- використання префіксу, наприклад, 172.16.155.5/18. Тут 18 – це префікс, що вказує на кількість одиниць у масці.

4.2 Алгоритми маршрутизації потоків даних

Важливою функцією мережного рівня є вибір маршруту для передавання пакетів від початкового до кінцевого вузла. У більшості мереж пакетам потрібно проходити через кілька маршрутизаторів. Алгоритми вибору маршруту і структури даних, що їх (алгоритми) використовують, є головною метою при проектуванні мережного рівня.

Алгоритм маршрутизації реалізується тією частиною програмного забезпечення мережного рівня, яка відповідає за вибір вихідної лінії для відправлення вхідного пакета. Також потрібно розуміти, що маршрутизація та

пересилання – це різні процеси. Пересилання полягає у опрацюванні вхідних пакетів і виборі для них, відповідно до таблиці маршрутизації, вихідної лінії. Маршрутизація відповідає за заповнення та оновлення таблиць маршрутизації. При цьому використовуються алгоритми маршрутизації.

Алгоритм вибору маршруту має бути коректним, простотим, надійним (вірно реагувати на зміни топології й трафіку), стійким, справедливим та оптимальним.

Алгоритми вибору маршруту можна поділити на два основних класи:

- адаптивні;
- неадаптивні.

Неадаптивні алгоритми не враховують при виборі маршруту топологію, поточний стан мережі і не змінюють трафік по лініях. Вибір маршруту для пари хостів виконується завчасно, в автономному режимі, і список маршрутів завантажується у маршрутизатори під час завантаження мережі.

Адаптивні алгоритми приймають рішення про вибір маршрутів при зміні топології, а також залежно від завантаженості ліній. Адаптивні алгоритми відрізняються джерелами отримання інформації, моментами зміни маршрутів (наприклад, через певні рівні інтервали часу, при зміні навантаження на канал або при зміні топології) і даними, що використовуються для оптимізації (відстань, кількість транзитних частин або очікуваний час пересилання).

При виборі шляху від одного вузла мережі до іншого має використовуватися оптимальний маршрут. Тут потрібно сказати, що в одній мережі може бути кілька оптимальних маршрутів. Множину оптимальних маршрутів від всіх відправників до отримувачів називають вхідним деревом. Завданням всіх алгоритмів вибору маршруту є обчислення і використання вхідного дерева для всіх маршрутизаторів. Вхідне дерево не містить петель, тому кожен пакет доставляється отримувачу за обмежене число пересилань. Однак у реальних умовах лінії зв’язку та маршрутизатори можуть виходити з ладу під час виконання певних операцій. Тому різні маршрутизатори можуть мати різне уявлення про поточну топологію мережі.

4.2.1 Вибір найкоротшого шляху використовується у різних формах завдяки своїй простоті. Ідея полягає у побудові графу підмережі, у якого кожен вузол буде відповідати маршрутизатору, а кожна дуга – лінії зв’язку. Для вибору маршруту між двома маршрутизаторами алгоритм знаходить найкоротший шлях між ними на графі. Найкоротшим шляхом може бути або кількість транзитних ділянок, або фізична довжина лінії, або середня довжина черги і час затримки пересилання, або пропускна спроможність.

Відомо кілька алгоритмів знаходження найкоротшого шляху між двома вузлами графу. Один з них було запропоновано Едсгером Вібе Дейкстрою (Edsger Wybe Dijkstra) у 1959 р. Кожен вузол «помічається» відстанню до нього від вузла відправника по найменшому відомому шляху. Спочатку шляхи невідомі, тому всі вузли вважаються недосяжними. Після того як

починає працювати алгоритм і знаходяться відстані, помітки вузлів змінюються й вказують на оптимальний шлях. Після того як підтверджується, що помітка відповідає найкоротшому шляху, вона стає постійною і надалі не змінюється.

Для прикладу розглянемо зважений ненаправлений граф, що наведений на рис. 4.2, а. Вагові коефіцієнти для ребер відповідають відстані між маршрутизаторами. Потрібно знайти найкоротший шлях від *А* до *Г*. Починаємо з вузла *А*, який позначимо чорним кругом. Далі досліджуємо всі зв'язані з ним вузли і вказуємо біля них відстань до вузла *А*. Якщо знайдеться більш короткий шлях до якого-небудь вузла, то разом з відстанню у помітці біля вузла зміниться назва вузла, через який буде проходити найкоротший шлях. Таким чином можна буде відновити весь шлях. Після розгляду всіх сусідів *А* помічаємо постійним той вузол, що є найближчим. Це буде вузол *Б*, він стає новим робочим вузлом. Відповідні позначки показано на рис. 4.2, б.

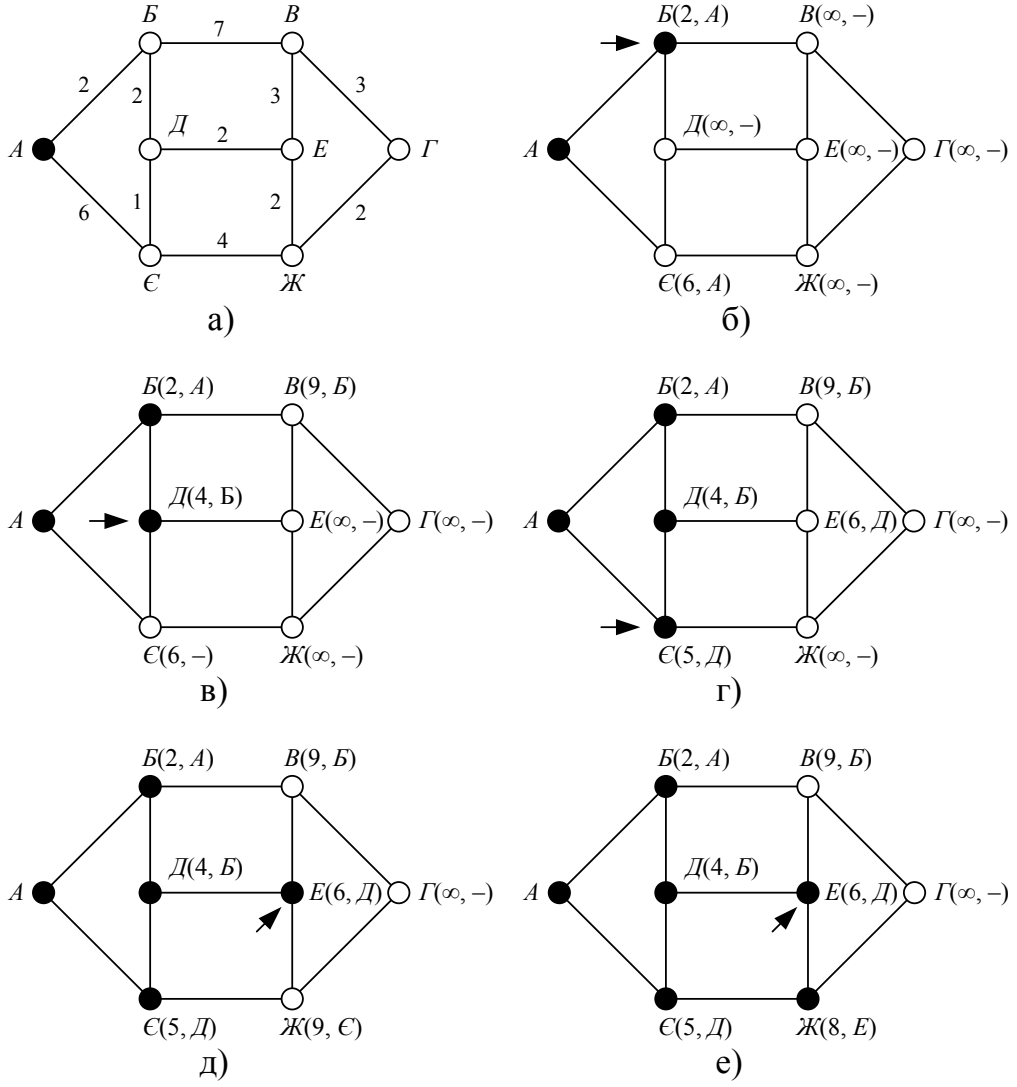


Рисунок 4.2 – Приклад обчислення найкоротшого шляху від *А* до *Г* з використанням алгоритму Дейкстри

Далі повторюємо таку ж процедуру з вузлом *Б* – досліджуємо всіх його сусідів. Якщо сума відстані від вузла *Б* і значення помітки у вузлі *Б* (відстань від *А* до *Б*) виявляється меншою, ніж значення помітки у вузлі, що досліджується (відстань до *А*, яку знайдено іншим шляхом), то це означає, що знайдено коротший шлях. Помітка вузла буде змінюватися.

Після дослідження всіх сусідніх з робочим вузлів і зміні тимчасових поміток, по всьому графу шукається вузол з найменшою поміткою. Цей вузол помічається як постійний і стає поточним робочим вузлом. На рис. 4.2 показано перші 5 етапів роботи алгоритму.

Для прикладу розглянемо рис. 4.2, в. На даному етапі вузол *Д* було помічено постійним. Нехай, наприклад, існує коротший шлях ніж *АБД*, наприклад, через уявні вершини *Х*, *У*, тобто *АХУД*. У даному випадку можливі два випадки: або вузол *У* вже є постійним, або ще ні. Якщо це так, то вузол *Д* вже перевірявся, тоді вузол *У* було призначено постійним і робочим вузлом. Тобто, шлях *АХУД* вже досліджувався.

Якщо вузол *У* помічено як тимчасовий, то помітка вузла *У* або більша, або дорівнює помітці вузла *Д*, чи менша неї. У першому випадку шлях *АХУД* не може бути коротшим, ніж шлях *АБД*. Якщо ж помітка вузла *У* менша помітки вузла *Д*, тоді вузол *У* має був стати постійним раніше вузла *Д*, і вузол *Д* перевірявся б з вузла *У*.

4.2.2 Метод заливання є статичним алгоритмом, при якому кожний пакет, що приходить, відправляється на всі вихідні лінії, крім тієї, з якої прийшов даний пакет. Даний алгоритм породжує велику кількість дублів пакетів, якщо не використовувати ніяких додаткових засобів. Один з них полягає у розміщенні в заголовку пакета лічильника, що буде рахувати кількість пройдених ділянок і зменшуватися на кожному маршрутизаторі. Коли значення цього лічильника стає нульовим, пакет знищується. В ідеальному випадку даний лічильник має встановлюватися таким, що дорівнює довжині шляху від відправника до отримувача. Якщо відправник не знає відстані до отримувача, він може встановити значення лічильника таким, що дорівнює довжині максимального шляху (діаметру) у даній підмережі.

Альтернативний спосіб обмеження кількості дублікатів полягає у підрахунку пакетів, що проходять через маршрутизатор. Це дозволяє не відправляти їх повторно. Наприклад, кожен маршрутизатор записує у кожний отриманий від своїх хостів пакет порядковий номер. Всі маршрутизатори мають список маршрутизаторів-джерел, у якому зберігаються всі порядкові номери пакетів, які їм зустрічаються. Якщо пакет від даного джерела з таким порядковим номером вже є у списку, то він далі не поширюється і знищується.

Щоб не було збільшення списку, можна забезпечити для всіх списків лічильник *k*, що буде показувати, що всі порядкові номери, включно з *k,* вже зустрічалися. І коли приходить пакет, можна легко перевірити, чи він не є дублікатом.

На практиці зустрічається варіант даного алгоритму під назвою «вибіркове заливання». У даному випадку маршрутизатори посилають пакети не по всіх лініях, а лише по тих, які йдуть приблизно у потрібному напрямку.

У більшості випадків даний алгоритм не використовується. Однак він застосовується там, де потрібна висока надійність, яка забезпечується надлишковим пересиланням пакетів.

4.2.3 Маршрутизація за вектором відстаней. Сучасні комп'ютерні мережі зазвичай використовують не статичні, а динамічні методи маршрутизації, оскільки статичні не враховують поточне завантаження мережі. Найбільш популярні два методи:

- маршрутизація за вектором відстаней;
- маршрутизація з урахуванням стану каналу.

Перший метод працює на основі таблиць (векторів), що підтримуються всіма маршрутизаторами й містять найкращі відомі шляхи для кожного з можливих адресатів. Для оновлення даних у таблицях виконується обмін інформацією між сусідніми маршрутизаторами.

Алгоритм маршрутизації за вектором відстаней ще називають за іменем його авторів Беллмана-Форда і Форда-Фулкерсона.

У даному випадку таблиці, з якими працюють і які оновлюють маршрутизатори, містять записи про кожен маршрутизатор підмережі. Кожен запис складається з двох частин: найпріоритетніший номер лінії для даного отримувача і відстань або час проходження пакета до даного отримувача. Одиницею вимірювання може слугувати число транзитних ділянок, мілісекунди, кількість пакетів, які чекають у черзі у даному напрямку, тощо.

Вважається, що маршрутизаторам відома відстань до кожного із сусідів. Якщо за одиницю вимірювання використовується кількість транзитних ділянок, то відстань дорівнює одній транзитній ділянці. Якщо відстань вимірюється в одиницях часу затримки, то маршрутизатор може її визначати за допомогою спеціального ECHO (exo) пакета, в який отримувач поміщає час отримання і який відправляється назад якомога швидше.

Нехай, наприклад, за одиницю вимірювання використовується час затримки, і для даного маршрутизатора цей параметр відомий для всіх сусідів. Через кожні T мілісекунд всі маршрутизатори відправляють своїм сусідам список із затримками для кожного отримувача. Кожен з них отримує такий список від усіх своїх сусідів. Наприклад, одна з таких таблиць прийшла від сусіда X, і в ній вказано, що час поширення від маршрутизатора X до маршрутизатора i дорівнює X_i. Якщо маршрутизатор знає, що час пересилання до маршрутизатора X дорівнює m, то затримка при передачі пакета маршрутизатору i через X дорівнює X_i+m. Після виконання таких обчислень для всіх своїх сусідів маршрутизатор може вибрати найкращий шлях і розмістити відповідний запис у нову таблицю. При цьому стара таблиця в обрахунках не використовується.

Процес оновлення таблиці показано на рис. 4.3. Зліва показано підмережу. Перші чотири стовпці показують вектори затримок, що їх отримує

маршрутизатор *Л* від свої сусідів. Маршрутизатор *А* вважає, що час передавання від нього до *Б* дорівнює 12 мс, 25 мс – до маршрутизатора *В* і т. д. Нехай маршрутизатор *Л* оцінив затримки до своїх сусідів *А*, *К*, *І* і *М* відповідно як 8, 10, 12 і 6 мс.

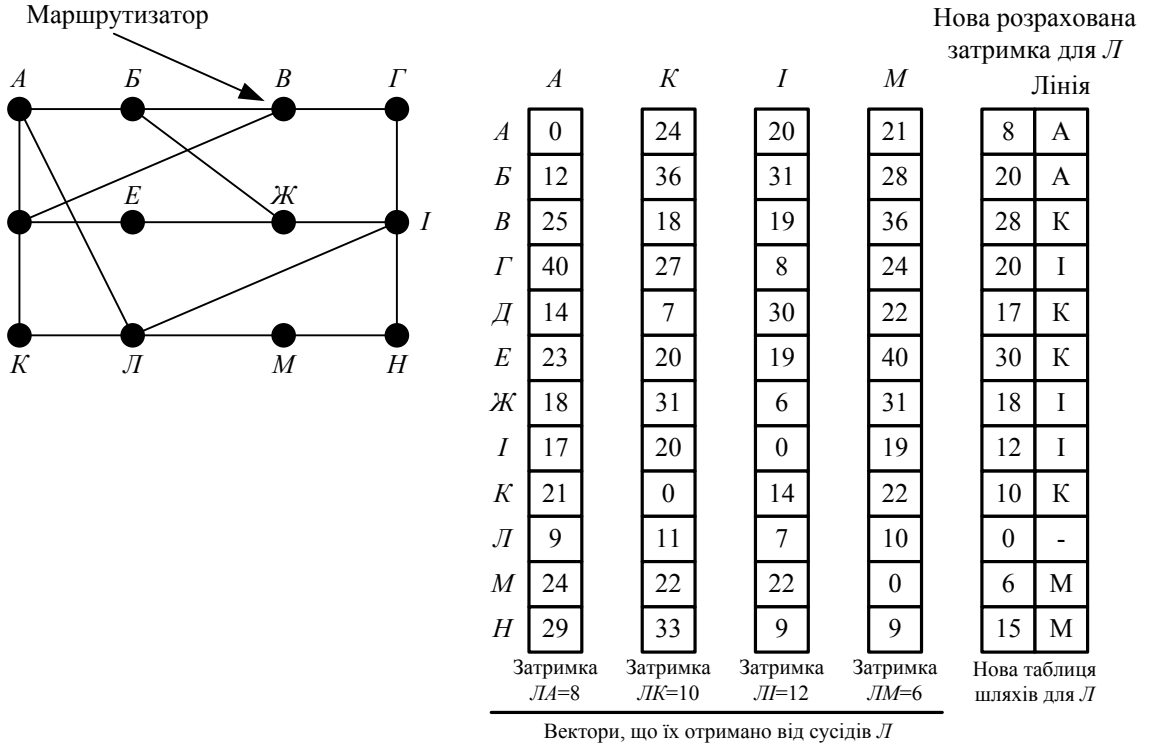


Рисунок 4.3 – Приклад виконання алгоритму Беллмана-Форда

Тепер розглянемо, як вузол *Л* розраховує новий маршрут до маршрутизатора *Ж*. Він знає, що затримка до *А* складає 8 мс, *А* вважає, що від нього до *Ж* дані буде доставлено за 18 мс. Таким чином, *Л* знає, що коли він буде відправляти пакети для *Ж* через *А*, то затримка складе 26 мс. Аналогічно він обчислює значення затримок для маршрутів від нього до *Ж*, що проходять через інших сусідів (*К*, *І* і *М*), й отримує відповідно 41 (31+10), 18 (6+12) і 37 (31+6) мс. Найкраще значення – це 18 мс, тому саме воно поміщається у рядок таблиці для отримувача *Ж*. Разом з числом записується значення лінії, через яку проходить найкоротший шлях до *Ж*, тобто це *І*. Даний метод повторюється для всіх інших адресатів, і при цьому отримується нова таблиця, що показана як правий стовпець на рисунку.

Алгоритм маршрутизації за вектором відстаней працює теоретично, але має серйозний недолік на практиці: хоча правильна відповідь нарешті знаходиться, процес її пошуку може зайняти дуже багато часу. Він швидко перераховує таблиці, якщо отримує «гарні» новини (наприклад, затримка стала меншою), але повільно після отримання «поганих» новин (наприклад, затримка стала більшою). Погані новини повільно поширюються – жоден маршрутизатор не може встановити значення відстані таким, що

більше ніж на одиницю перевищує мінімальне значення цієї відстані, що зберігається у сусідів. Таким чином, всі маршрутизатори будуть до нескінченності збільшувати значення відстані до маршрутизатора, що «зник» у підмережі.

4.2.4 Маршрутизація з урахуванням стану лінії. Відмовитися від алгоритму на основі векторів відстаней довелося у 1979 р. з двох причин: не враховувалася пропускна спроможність лінії, алгоритм буде довго приходити у стійкий стан (проблема рахунку до нескінченності). Основою нового алгоритму була проста ідея, що ґрунтується на основі п'яти вимог до маршрутизатора, а саме: кожен маршрутизатор має:

- знаходити своїх сусідів і дізнаватися їх мережні адреси;
- вимірювати затримку або вартість зв'язку з кожним зі своїх сусідів;
- створювати пакет, що містить всю зібрану інформацію;
- відправляти цей пакет всім маршрутизаторам;
- обчислювати найкоротший шлях до всіх маршрутизаторів.

У результаті кожному маршрутизатору відправляються: повна топологія й усі виміряні значення затримок. Після цього для знаходження найкоротшого шляху до кожного маршрутизатора може використовуватися алгоритм Дейкстри.

4.2.5 Ієрархічна маршрутизація. Розмір таблиць маршрутизації, що підтримуються маршрутизаторами, збільшується пропорційно збільшенню розміру мережі. При цьому потрібен значний обсяг пам'яті для її зберігання, більше часу для її опрацювання, збільшується кількість службових пакетів. У певний момент мережа може збільшитися до таких розмірів, що зберігати на маршрутизаторах всю необхідну інформацію буде неможливо. Тому у великих мережах маршрутизація має виконуватися ієрархічно.

При використанні ієрархічної маршрутизації маршрутизатори розбиваються на окремі регіони. Кожен маршрутизатор знає всі деталі вибору маршруту в межах свого регіону, але нічого не знає про структуру інших регіонів.

У великих мережах може використовуватися не лише дворівнева ієрархія. Може бути необхідне групування регіонів у кластери, кластерів у зони, зон у групи і т. д. Тут було доведено, що оптимальна кількість рівнів ієрархії для підмереж, що складаються з N маршрутизаторів, дорівнює $\ln N$. При цьому потрібно $e \ln N$ записів для кожного маршрутизатора. Доведено, що збільшення довжини ефективного середнього шляху, що є наслідком використання ієрархічної маршрутизації, досить мале і зазвичай є прийнятним.

4.2.6 Широкомовна маршрутизація. У деяких застосуваннях потрібно відправляти повідомлення на багато хостів або відразу на всі. Найбільш ефективно поширювати відповідні дані широкомовним способом, що дає можливість всім зацікавленим хостам отримувати їх. Широкомовним називається розсилання пакетів всім пунктам призначення. Для його реалізації використовуються різні методи.

Один із методів виконує розсилання окремих пакетів у всіх напрямках. Але це призводить до завантаження каналу. Крім того, відправнику потрібно мати повний список всіх хостів-отримувачів. Тому використовується алгоритм багатоадресної маршрутизації. При використанні даного методу у кожному пакеті міститься або список адресатів, або бітова карта, що вказує на хости-отримувачі. Маршрутизатор, що отримує такий пакет, перевіряє список, що міститься в пакеті, і визначає набір вихідних ліній. Далі створює копії пакета для кожної вихідної лінії. Весь список розсилання розподіляється між вихідними лініями. Після певної кількості пересилань кожен пакет буде містити лише одну адресу призначення.

Інший алгоритм широкомовної маршрутизації використовує кореневе дерево або інше зв'язне дерево. Зв'язне дерево являє собою підмножину підмережі, що містить у собі всі маршрутизатори, але не містить замкнутих шляхів. Якщо кожен маршрутизатор знає, які з його ліній належать зв'язному дереву, він може відправити вхідний пакет на всі лінії зв'язного дерева, крім тієї, з якої його отримав. Такий метод оптимально використовує пропускну спроможність мережі, утворює мінімальну кількість пакетів. Єдиною умовою є наявність у всіх маршрутизаторів інформації про зв'язне дерево.

Існує подібний широкомовний алгоритм, в якому може не використовуватися зв'язне дерево. У його основу покладено ідею просування зустрічним шляхом. Коли приходить широкомовний пакет, маршрутизатор перевіряє, чи використовується для передавання джерелу широкомовлення лінія, якою прийшов пакет. Якщо відповідь позитивна, то велика ймовірність того, що широкомовний пакет прийшов найкращим маршрутом і є першою копією, яку отримав маршрутизатор. Тоді маршрутизатор розсилає цей пакет всім лініям крім тієї, якою він його отримав. Але якщо пакет отримано від того ж джерела іншою лінією, він відкидається як потенційна копія.

4.3 Принципи реалізації протоколів мережного рівня на прикладі протоколу IPv4

Протокол IP (Internet Protocol – міжмережний протокол) описано у документі RFC 751. У кожній наступній мережі, що знаходиться на шляху переміщення пакета, протокол IP звертається до засобів транспортування даної мережі, щоб з їх допомогою передати пакет на маршрутизатор, який веде до наступної мережі, або безпосередньо до вузла-отримувача. Однією з головних функцій протоколу IP є підтримання інтерфейсу з технологіями мереж, що знаходяться нижче за ієрархією і утворюють загальну мережу. Також протокол IP підтримує інтерфейс з протоколами транспортного рівня.

Протокол IP відносять до протоколів без встановлення з'єднання, він підтримує опрацювання кожного IP-пакета як незалежної одиниці обміну, що не пов'язана з іншими пакетами. Він не забезпечує достовірну доставку даних.

Отримати інформацію про основні функції протоколу допоможе розгляд структури заголовка протоколу. IP-пакет складається з поля заголовка і даних. Структуру заголовка показано на рис. 4.4.

Рисунок 4.4 – Структура заголовка IP-пакета

Поле **номер версії** займає 4 біти та ідентифікує версію протоколу IP. Сьогодні в основному використовується версія 4 (IPv4), хоча все частіше зустрічається нова версія IPv6.

Значення **довжини заголовка** IP-пакета також займає 4 біти та вимірюється у 32-бітових словах. Зазвичай заголовок має довжину 20 байт (п’ять 32-бітових слів), але при додаванні деякої службової інформації це значення може бути збільшено за рахунок додаткових байтів у полі параметрів. Найбільша довжина заголовка складає 60 байтів.

Поле **тип сервісу** (Type of Service, ToS) використовується для зберігання ознак, що відображають вимоги до якості обслуговування пакета. Призначення бітів таке:

- перші 3 біти містять значення пріоритету пакета: від найнижчого 0 до найвищого 7. Пріоритет пакета може братися до уваги, і більш важливі пакети опрацьовуються у першу чергу;
- наступні 3 біти визначають критерій вибору маршруту. Якщо біт D (Delay – затримка) встановлено в 1, то маршрут має бути таким, щоб затримка доставляння була мінімальною. Встановлення біта T (Throughput – пропускна спроможність) в 1 означає максимізацію пропускної спроможності, а біт R (Reliability – надійність) використовується для максимізації надійності доставлення;
- 2 біти, що залишилися, мають нульове значення. Але наприкінці 90-х років XX ст. було прийнято стандарти диференційного диференційованого обслуговування, і поле «тип сервісу» почали назива-

ти байтом диференційованого обслуговування (DS-байт). При цьому підтримується диференційоване обслуговування класів трафіку. Класом трафіку називається сукупність пакетів, що надходять на опрацювання і мають загальну ознаку, наприклад, всі пакети голосових застосувань або всі пакети з MTU у певних межах.

Поле **загальної довжини** займає 2 байти і характеризує загальну довжину пакета з урахування заголовка і поля даних. Максимальна довжина пакета обмежується розрядністю поля, що визначає дану величину, і складає 65535 байтів. Але у більшості комп'ютерів і мереж такі великі пакети не використовуються. Довжина пакета вибирається з урахуванням максимальної довжини пакета протоколу нижнього рівня, що буде містити IP-пакет. Якщо це кадри Ethernet, то вибирають пакет з максимальною довжиною 1500 байтів.

Ідентифікатор пакета займає 2 байти і використовується для розпізнання пакетів, що утворюються у результаті поділу на частини (фрагментації) початкового пакета. Всі частини (фрагменти) одного пакета мають однакове значення цього поля.

Прапорці займають 3 біти і містять ознаки, що пов'язані з фрагментацією. Встановлений у 1 біт DF (Do not Fragment – не фрагментувати) забороняє маршрутизатору фрагментувати даний пакет. Встановлений в 1 біт MF (More Fragments – більше фрагментів) говорить про те, що даний пакет є проміжним (не останнім) фрагментом. Останній біт зарезервовано.

Поле **зсув фрагмента** займає 13 бітів і задає зміщення поля даних цього фрагмента відносно початку поля даних вихідного (нефрагментованого) пакета. Використовується при збиранні/розбиранні фрагментів пакетів. Зміщення має бути кратне 8 байтам.

Поле **час життя** (Time to Live, TTL) займає 1 байт і використовується для задання граничного терміну, протягом якого пакет може передаватися через мережу. Час життя пакета вимірюється або в секундах і джерелом-відправником, або кількістю проміжних вузлів. Після кожної секунди перебування на кожному маршрутизаторі, через який проходить пакет по мережі, з поточного часу життя віднімається одиниця. Ця одиниця віднімається і тоді, коли час перебування був менше секунди. Оскільки сучасні маршрутизатори рідко опрацьовують пакет більше секунди, то час життя можна інтерпретувати як максимальну кількість транзитних вузлів, які дозволено пройти пакету. Якщо значення поля стає нульовим до того, як пакет потрапляє отримувачу, пакет знищується.

Поле **протоколу верхнього рівня** займає один байт і містить ідентифікатор, що вказує, якому протоколу верхнього рівня належить інформація, що розміщена у полі даних пакета. Значення ідентифікаторів для різних протоколів наведено у документі RFC 1700, наприклад, 6 означає, що в пакеті знаходиться повідомлення протоколу TCP, 17 – протоколу UDP, 1 – протоколу ICMP.

Контрольна сума заголовка займає 2 байти і розраховується лише для заголовка. Оскільки деякі поля заголовка змінюють своє значення у процесі передавання пакета мережею (наприклад, час життя), контрольна сума перевіряється і повторно розраховується на кожному маршрутизаторі і кінцевому вузлі як доповнення до суми всіх 16-бітових слів заголовка. Під час обчислення контрольної суми значення самого поля контрольної суми встановлюється у нуль. Якщо контрольна сума не збігається, то пакет відкидається.

Поле **IP-адрес відправника і отримувача** мають однакову довжину – 32 біти.

Поле **параметрів** є необов'язковим і використовується, зазвичай, лише для налагоджування мережі. Це поле складається з кількох опцій одного з восьми визначених типів. У цих опціях можна вказувати точний маршрут, реєструвати маршрут, яким проходить пакет, записувати дані системи безпеки і часові відліки.

Оскільки кількість підполів у полі параметрів може бути довільною, то в кінці заголовка має бути додано кілька нульових байтів для вирівнювання заголовка пакета за 32-бітовою межею.

4.4 Класифікація протоколів динамічної маршрутизації

Як було зазначено у попередніх підрозділах, маршрутизація являє собою вибір напрямків (маршрутів) передавання даних від однієї мережі до іншої. Маршрути можуть задаватись безпосередньо адміністратором (статичні маршрути) або створюватись динамічно на основі інформації, отриманої від інших маршрутизаторів (динамічні маршрути). Інформація про маршрути зберігається в **таблиці маршрутизації**. Основною перевагою динамічної маршрутизації є можливість оперативного реагування на стан мережі: появу нових сегментів, відмову каналів зв'язку і навіть вихід із ладу окремих маршрутизаторів. Якщо при налаштуванні невеликої мережі, що містить два, три маршрутизатори, статична маршрутизація є цілком прийнятним рішенням, то в мережах з десятками та сотнями маршрутизаторів альтернатива динамічній маршрутизації відсутня.

Так, на рис. 4.5 показано мережу, що містить три маршрутизатори. Нехай адміністратор налаштував обладнання таким чином, що маршрут з мережі А в мережу Б проходить через перший та другий маршрутизатори. Якщо канал між роутерами Rt1 та Rt2 відмовить, то проходження даних за цим маршрутом стане неможливим, хоча, як видно з рисунка, існує альтернативний маршрут через Rt3. Однак для того, щоб цей маршрут став доступним, адміністратор має внести зміни в маршрутні таблиці на Rt1 та Rt2. При використанні динамічної маршрутизації зміни будуть зроблені автоматично протягом кількох хвилин, або навіть секунд.

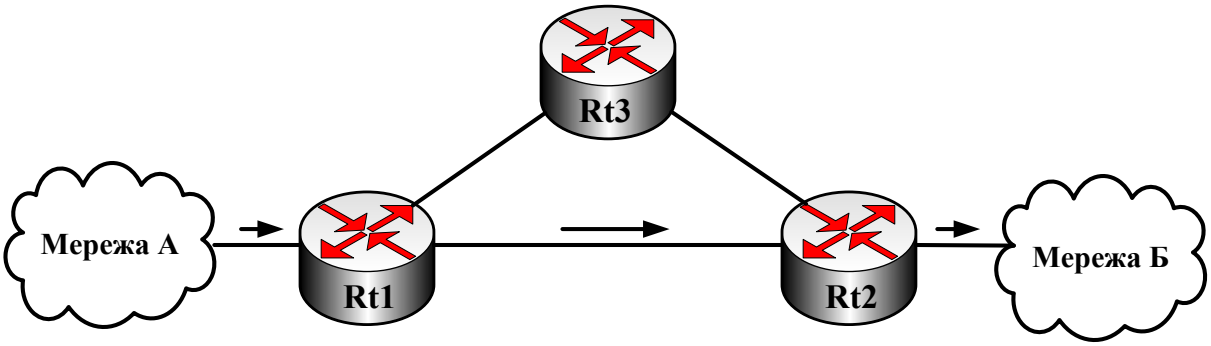


Рисунок 4.5 – Статичний маршрут

Успішне функціонування динамічної маршрутизації залежить від виконання маршрутизатором двох основних функцій:

- підтримки таблиці маршрутизації в актуальному стані;
- своєчасного розповсюдження інформації про стан того чи іншого фрагмента мережі.

Для розповсюдження інформації про мережу використовується один або кілька протоколів маршрутизації. Протокол маршрутизації визначає набір правил, що використовуються маршрутизатором при здійсненні зв'язку з сусідніми маршрутизаторами, зокрема він визначає:

- яким чином розсилаються оновлення маршрутів;
- яка інформація міститься в оновленнях;
- з якою періодичністю відсилаються оновлення;
- яким чином визначаються адресати оновлень.

При формуванні таблиць маршрутизації та їх оновленні постає задача вибору найкращого маршруту. Кожний алгоритм маршрутизації використовує свій власний спосіб вибору найкращого маршруту. Для цього з кожним маршрутом асоціюється певне значення, яке називається **метрикою**. Як правило, менше значення метрики вказує на кращий маршрут.

Одні протоколи використовують прості метрики, що розраховуються на основі якогось одного параметра, наприклад, кількості проміжних маршрутизаторів. Існують також протоколи, що використовують так звані композитні метрики, які розраховуються на основі двох або більше параметрів маршруту. Для розрахунку метрик найчастіше використовують нижченаведені параметри.

Смуга пропускання (Bandwidth) – характеризує пропускну спроможність каналу, чим більша пропускна спроможність, тим менше значення метрики. Цей параметр є статичним і не змінюється в процесі роботи мережі.

Затримка (Delay) – час, який потрібен пакету для проходження каналом зв'язку; також статичний параметр і залежить тільки від параметрів інтерфейсів, що утворюють канал.

Рівень завантаження (Load) – ступінь використання ресурсів каналу або маршрутизатора, динамічний параметр.

Надійність (Reliability) – характеризує рівень похибок, що виникають в каналі.

Кількість переходів (Hop count) – кількість маршрутизаторів, через які має пройти пакет для досягнення адресата.

Вартість (Cost) – довільне значення, що розраховується на основі ширини смуги пропускання або встановлюється безпосередньо адміністратором.

Коли мова йде про протоколи маршрутизації слід чітко відрізняти протоколи маршрутизації від протоколів, що маршрутизуються.

Протокол маршрутизації – це засіб комунікації між маршрутизаторами, який дозволяє пристроям спільно використовувати інформацію про мережі і визначати відстань до окремих вузлів та мереж. Інформація, яку один маршрутизатор отримує від іншого за допомогою протоколу маршрутизації, використовується для створення або оновлення таблиці маршрутизації. Найпоширенішими протоколами маршрутизації є:

- RIP (Routing Information Protocol);
- OSPF (Open Shortest Path First);
- EIGRP (Enhanced Interior Gateway Protocol);
- IS-IS (Intermediate System To Intermediate System).

Базовими характеристиками, за якими адміністратор вибирає той чи інший протокол маршрутизації, є:

- час реакції на зміни в мережі (час конвергенції);
- тип метрики;
- складність налаштування та підтримки;
- протокол є відкритим чи проприєтарним (належить фірмі-розробнику).

Протоколи, **що маршрутизуються**, використовуються для доставляння даних користувачів. Їх часто називають протоколами передавання даних. Слід звернути увагу, що протоколи маршрутизації для доставляння своїх службових повідомлень використовують протоколи, що маршрутизуються. На сьогоднішній день найпоширенішим протоколом, що маршрутизується, є протокол IP.

Як було зазначено вище, протокол динамічної маршрутизації має підтримувати в актуальному стані таблиці маршрутизації. Для цього йому необхідно оперативно реагувати на зміни в мережі. Очевидно, що швидкість реакції буде безпосередньо залежати від розміру мережі, яку він обслуговує. Саме тому досить часто адміністратори великих корпоративних мереж поділяють мережу на кілька зон маршрутизації. Більшість сучасних протоколів динамічної маршрутизації підтримує багатозонну маршрутизацію.

У той же час виникає інша задача – об’єднання кількох мереж, що належать різним власникам і, як правило, використовують різні протоколи динамічної маршрутизації, в одну глобальну мережу з підтримкою марш-

рутизації між ними. Для вирішення цієї задачі введено поняття автономної системи (АС) та розроблено протоколи маршрутизації між АС.

Автономна система – це набір мереж, які знаходяться під єдиним адміністративним керуванням і в яких використовується єдина стратегія та правила маршрутизації. Як правило, питаннями маршрутизації між автономними системами займаються Internet-провайдери. Надання номерів автономним системам здійснюється централізовано організацією IANA аналогічно IP-адресації. На середину 2011 р. було зареєстровано більше 37 тисяч автономних систем.

Загальну класифікацію протоколів динамічної маршрутизації наведено на рис. 4.6.

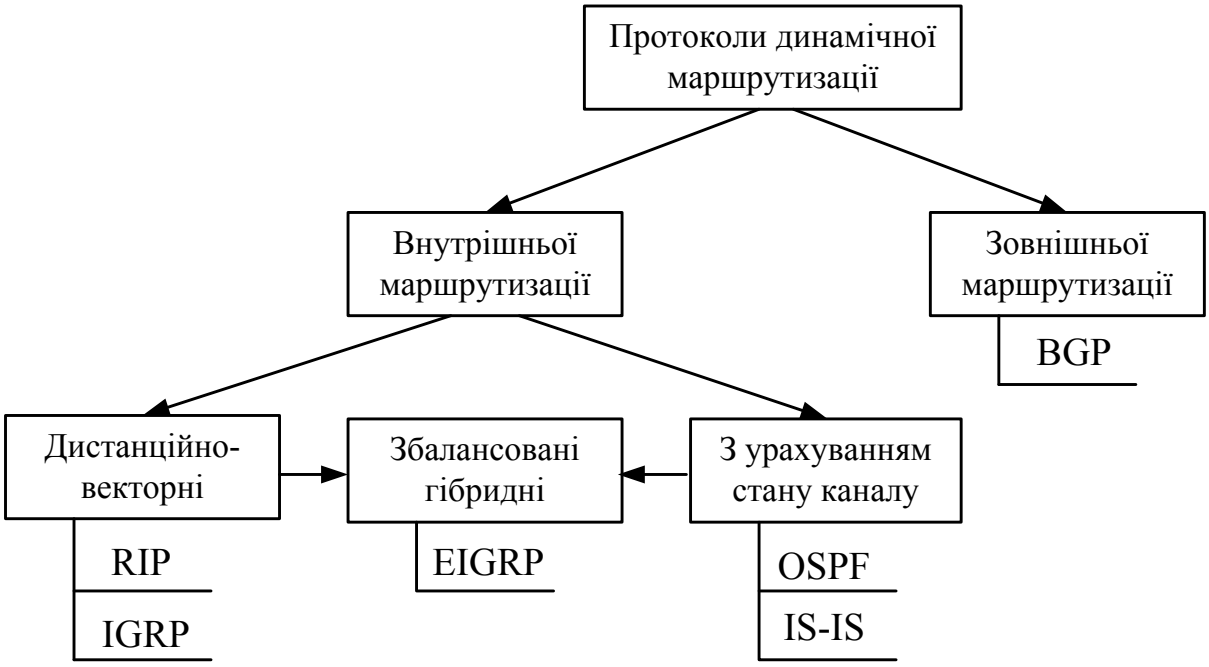


Рисунок 4.6 – Класифікація протоколів динамічної маршрутизації

За місцем застосування протоколи динамічної маршрутизації поділяються на два класи: **внутрішньої маршрутизації** IGP (Interior Gateway Protocols) та **зовнішньої маршрутизації** EGP (Exterior Gateway Protocols). Протоколи класу IGP використовуються всередині автономних систем і призначені для розповсюдження інформації про наявні мережі. Інформація про зміни (поява нових і недосяжність існуючих мереж) передається іншим маршрутизаторам автономної системи також за допомогою протоколів IGP.

Для того, щоб маршрутизаторам однієї АС стало відомо про мережі, що розташовані в інших АС, використовуються протоколи класу EGP. Маршрутизатори, що забезпечують зовнішню маршрутизацію, розташовані на межі автономних систем і передають іншим примежовим маршрутизаторам інформацію про наявні в межах АС мережі (рис. 4.7).

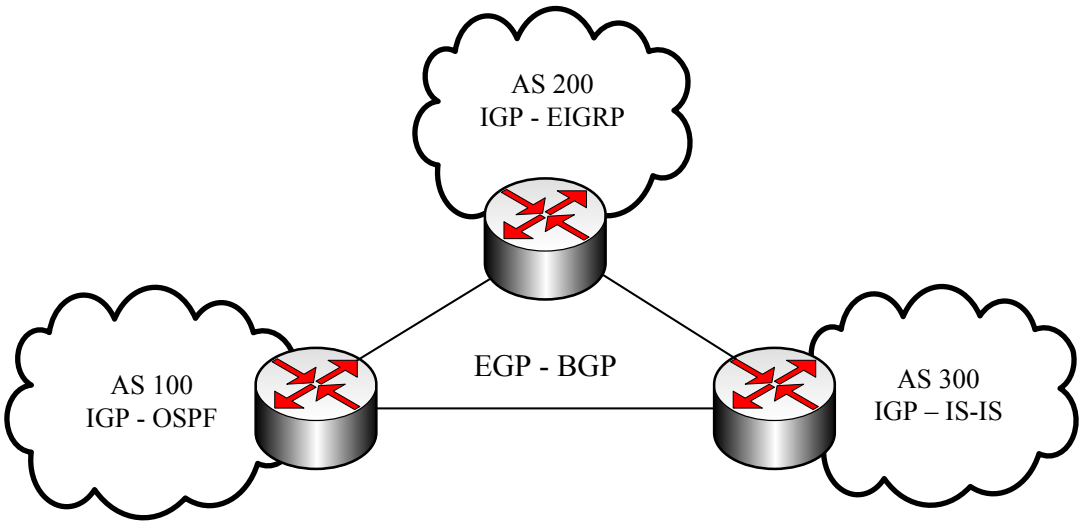


Рисунок 4.7 – Зовнішня та внутрішня маршрутизації

На сьогоднішній день функцію зовнішньої маршрутизації виконує єдиний протокол – BGP (Border Gateway Protocol).

За принципом функціонування протоколи класу IGP можна поділити на 2 базові типи:

- дистанційно-векторні протоколи;
- протоколи з урахуванням стану каналу.

Дистанційно-векторний протокол (distance vector routing protocol) визначає напрямок (вектор) та відстань до кожної мережі. **Протоколи з урахуванням стану каналу** (link-state routing protocol – LSR) відтворюють топологію всієї мережі і на основі алгоритму SPF (shortest path first) розраховують оптимальний маршрут до кожної мережі. Існують також комбіновані протоколи, які називаються **збалансованими гібридними протоколами,** які поєднують певні риси базових класів.

Найвідомішим і, певно, найстарішим дистанційно-векторним протоколом є **протокол RIP** (Routing Information Protocol). Цей протокол є відкритим і підтримується мережним обладнанням всіх виробників, досить простий в налаштуванні. Як метрику використовує інформацію про кількість проміжних маршрутизаторів, що іноді призводить до неоптимальної маршрутизації. Існують дві версії RIP: RIPv1 та RIPv2, головна відмінність між якими полягає в тому, що RIPv1 в оновленнях не передає інформацію про маску.

Протокол IGRP (Interior Gateway Routing Protocol) теж належить до групи дистанційно-векторних протоколів, був розроблений в середині 80-х років минулого століття фірмою Cisco. Головна відмінність від RIP – використання композитної метрики, яка розраховується на основі кількох параметрів маршруту. На сьогоднішній день вважається застарілим.

Протокол OSPF (Open Shortest Path First), як і RIP, належить до відкритих протоколів, однак базується на принципово інших підходах щодо визначення маршрутів. Кожний маршрутизатор на основі інформації про

топологію всієї мережі розраховує оптимальні маршрути, використовуючи алгоритм Дейкстри. Метрики розраховуються на основі пропускної спроможності каналів. Широко застосовується в гетерогенних мережах.

Протокол IS-IS (Intermediate System To Intermediate System) також є стандартним протоколом класу IGP і за принципом роботи дуже схожий на OSPF. Відмінності полягають в реалізації багатозонної маршрутизації. Крім того IS-IS підтримує більшу кількість маршрутів, що робить його привабливішим з точки зору масштабованості.

Протокол EIGRP (Enhanced Interior Gateway Routing Protocol) розроблений фірмою Cisco як альтернатива IGRP. Він уособлює позитивні риси як дистанційно-векторних протоколів, так і протоколів з урахуванням стану каналу.

4.5 Дистанційно-векторні протоколи маршрутизації

При використанні дистанційно-векторних протоколів маршрутизатори періодично обмінюються один з одним копіями таблиць маршрутизації (оновлення). Таким чином вони регулярно повідомляють своїх сусідів про зміни топології мережі. Дистанційно-векторні протоколи базуються на алгоритмі Беллмана-Форда.

Проілюструємо роботу дистанційно-векторного протоколу на прикладі протоколу RIP. На рис. 4.8 наведено схему мережі, в якій для забезпечення маршрутизації використовується протокол RIP.

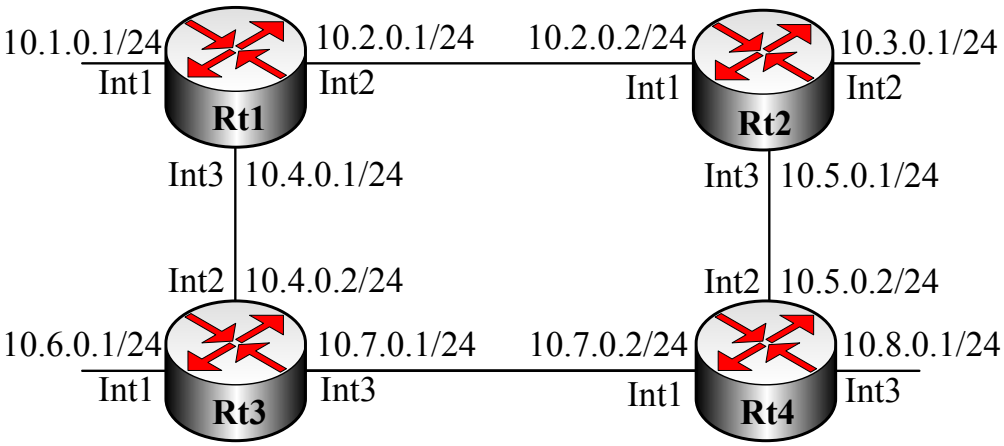


Рисунок 4.8 – Мережа для налаштування RIP

Формування таблиць маршрутизації при використанні протоколу RIP здійснюється в кілька етапів. Розглянемо цей процес на прикладі маршрутизатора Rt1.

На початковому етапі маршрутизатор заносить в таблицю маршрутизації інформацію про безпосередньо приєднані мережі. Це відбувається

автоматично після налаштування інтерфейсів без залучення засобів статичної або динамічної маршрутизації. Таким чином, після завершення цього етапу таблиця маршрутизації Rt1 матиме вигляд, який показано в табл. 4.1.

Таблиця 4.1 – Таблиця маршрутизації Rt1 на початковому етапі

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	0
10.2.0.0	255.255.255.0	Int2	-	0
10.4.0.0	255.255.255.0	Int3	-	0

Зауважимо, що поле Next Router (наступний маршрутизатор) порожнє, а поле метрики дорівнює 0, оскільки мережі безпосередньо приєднані. Нагадаємо, що протокол RIP за метрику використовує кількість проміжних маршрутизаторів.

На наступному етапі кожний маршрутизатор відправляє свої таблиці маршрутизації сусідам і отримує аналогічні таблиці від них. В оновленнях передається інформація про адреси мереж, маску та метрики, причому значення метрик збільшується на одиницю. Отримавши оновлення, маршрутизатор додає нові записи в таблицю маршрутизації. Звертаємо увагу, що в поле Next Router заноситься адреса відправника оновлення, а в поле Interface – ім’я інтерфейсу, через який оновлення отримано. Так, для схеми на рис. 4.8 після отримання оновлень від Rt3 та Rt2 таблиця маршрутизації Rt1 набуде вигляду, наведеного в табл. 4.2.

Таблиця 4.2 – Таблиця маршрутизації Rt1 з першими оновленнями

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	0
10.2.0.0	255.255.255.0	Int2	-	0
10.4.0.0	255.255.255.0	Int3	-	0
~~10.4.0.0~~	~~255.255.255.0~~	~~Int3~~	~~10.4.0.2~~	~~1~~
10.6.0.0	255.255.255.0	Int3	10.4.0.2	1
10.7.0.0	255.255.255.0	Int3	10.4.0.2	1
~~10.2.0.0~~	~~255.255.255.0~~	~~Int2~~	~~10.2.0.2~~	~~1~~
10.3.0.0	255.255.255.0	Int2	10.2.0.2	1
10.5.0.0	255.255.255.0	Int2	10.2.0.2	1

У тому випадку, коли на маршрутизатор надходить інформація про мережу, для якої вже є запис в таблиці маршрутизації, перевага надається запису з меншою метрикою. Так, в наведеному прикладі оновлення про мережі 10.4.0.0 та 10.2.0.0, що надійшли відповідно від Rt3 та Rt2, будуть проігноровані.

Аналогічно відбувається формування маршрутних таблиць іншими маршрутизаторами. Так, після отримання перших оновлень в таблицях мар-

шрутизації Rt3 та Rt2 з'явиться інформація про мережу 10.8.0.0 з метрикою 1, яка на наступному етапі розсилання оновлень потрапить на Rt1, і таблиця маршрутизації набуде вигляду, наведеного в табл. 4.3.

Таблиця 4.3 – Таблиця маршрутизації Rt1 з другими оновленнями

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	0
10.2.0.0	255.255.255.0	Int2	-	0
10.4.0.0	255.255.255.0	Int3	-	0
10.6.0.0	255.255.255.0	Int3	10.4.0.2	1
10.7.0.0	255.255.255.0	Int3	10.4.0.2	1
10.3.0.0	255.255.255.0	Int2	10.2.0.2	1
10.5.0.0	255.255.255.0	Int2	10.2.0.2	1
10.8.0.0	255.255.255.0	Int3	10.4.0.2	2
10.8.0.0	255.255.255.0	Int2	10.2.0.2	2

Таким чином, для мережі, наведеної на рис. 4.8, процес формування таблиці маршрутизації завершиться після отримання двох оновлень. Звертаємо увагу, що мережа 10.8.0.0 подана в таблиці двома записами, тобто до неї ведуть два маршрути з однаковими метриками. У такому випадку маршрутизатор буде одну частину трафіку пересилати одним маршрутом, а іншу – іншим. Така технологія називається балансуванням навантаження (load balancing).

Одна з основних функцій протоколу маршрутизації – оперативне реагування на зміни в топології мережі, які можуть бути додатними (поява нових мереж) та від'ємними (втрата доступу або відключення існуючих мереж). Реакція дистанційно-векторного протоколу на додатні зміни є прогнозованою – інформація про нові мережі потрапляє в чергові оновлення й розповсюджується на інші маршрутизатори. Що стосується від'ємних змін, то вони можуть суттєво ускладнити роботу протоколу і навіть привести мережу в непрацездатний стан.

У випадку, коли мережа 10.3.0.0 стала недоступною, очевидно, що Rt2 вилучить запис про неї зі своєї таблиці маршрутизації. Після цього надходить оновлення від Rt1, в якому є маршрут до мережі 10.3.0.0 (рис. 4.9).

У даному випадку протокол RIP створив так звану маршрутну петлю, а саме: пакети, що прямують в мережу 10.3.0.0, будуть передаватись між Rt1 та Rt2 доки поле TTL не дорівнюватиме 0. Оскільки записи в таблиці маршрутизації також обмежені в часі, настане момент, коли інформацію про мережу 10.3.0.0 буде вилучено з Rt1. Однак після надходження чергового оновлення з Rt2 цей запис знову з'явиться, тільки вже з метрикою 3 і т. д. Якщо не вжити якихось заходів, петля буди існувати нескінченно.

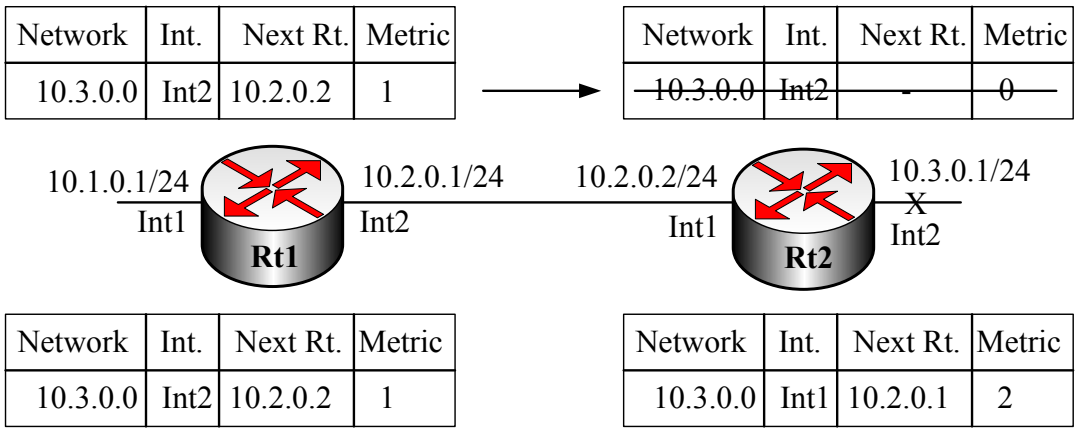


Рисунок 4.9 – Утворення маршрутних петель протоколом RIP

Для обмеження часу існування петель та запобігання їх утворенню розроблено низку методів, зокрема: обмеження максимального значення метрики, розщеплення горизонту, миттєве оновлення тощо. Максимальне значення метрики, а, відповідно, й максимальна кількість послідовно розташованих маршрутизаторів в протоколі RIP обмежені числом 15. Метрика 16 використовується для інформування сусідів про недосяжність мережі.

Головна причина утворення маршрутних петель полягає в поширенні застарілої інформації. Так, в наведеному прикладі Rt2 сприйняв як актуальну інформацію від Rt1 про мережу 10.3.0.0, яку, до речі, він раніше сам і надіслав до Rt1. Тобто, Rt1 не був вчасно проінформований, що мережа 10.3.0.0 стала недосяжною, і в своїх оновленнях розповсюджував інформацію, що втратила свою актуальність.

Суть методу **розщеплення горизонту** (split horizon) полягає в тому, що маршрутна інформація про певну мережу ніколи не передається на той маршрутизатор, звідки її було отримано. Так, маршрутизатор Rt1 (див. рис. 4.8) в оновленнях, що будуть відсилатись через Int2, передаватиме тільки маршрути до мереж 10.1.0.0, 10.4.0.0, 10.6.0.0 та 10.7.0.0, а через Int3 – 10.1.0.0, 10.2.0.0, 10.3.0.0 та 10.5.0.0. Метод розщеплення горизонту вирішує проблему петель між двома сусідніми маршрутизаторами (рис. 4.9), однак петля може утворюватись трьома і більше маршрутизаторами. Так, в наведеному прикладі застаріла інформація про мережу 10.3.0.0 може надійти через Rt3 та Rt4, і тоді петлю утворять всі чотири маршрутизатори.

Метод **миттєвих оновлень** (triggered update) полягає в тому, що, отримавши інформацію про змінення метрики до певної мережі, маршрутизатор не чекає моменту розсилання періодичних оновлень, а відправляє оновлення терміново. Насправді передавання здійснюється не миттєво, а з затримкою у кілька секунд, тому певна ймовірність утворення петлі залишається.

Метод **тимчасового утримання від змін** (holddown) змушує маршрутизатори, яким стало відомо про недосяжність тієї чи іншої мережі, утриматись на певний час від прийняття оновлень про цю мережу від сусідніх

маршрутизаторів. Припускається, що протягом відповідного періоду маршрутизатори вилучать зі своїх таблиць інформацію про недосяжну мережу, оскільки про неї не будуть надходити оновлення.

Не зважаючи на значний арсенал методів боротьби з маршрутними петлями стовідсоткової гарантії уникнення цього явища дистанційно-векторні протоколи не дають. Окрім маршрутних петель існують й інші недоліки дистанційно-векторних протоколів, зокрема обмежений діаметр мережі (15 послідовних маршрутизаторів для RIP), періодичне відсилання повної таблиці маршрутизації на широкомовну або групову адресу, що завантажує службовим трафіком канали зв'язку та ресурси отримувачів, використання як метрики кількості проміжних маршрутизаторів (протокол RIP). Саме тому протокол RIP здебільшого використовується в невеликих локальних мережах, де зазначені недоліки не є критичними.

4.6 Протоколи маршрутизації з урахуванням стану каналу

Протоколи маршрутизації з урахуванням стану каналу принципово відрізняються від дистанційно-векторних. Головна відмінність – наявність в кожному маршрутизаторі інформації про **топологію всієї мережі**, на основі якої здійснюється розрахунок оптимальних маршрутів. Для визначення маршруту, як правило, використовується алгоритм Дейкстри, який ще називають алгоритмом вибору найкоротшого шляху (Shortest Path First – SPF). Для реалізації даного типу маршрутизації використовуються нижченаведені компоненти.

Анонси стану каналу LSA (Link-State Advertisement) – невеликі пакети, що містять інформацію про параметри та стан безпосередньо приєднаних до маршрутизатора каналів.

Топологічна база даних, яка створюється безпосередньо кожним маршрутизатором на основі отриманих LSA.

Алгоритм вибору найкоротшого шляху використовується для знаходження оптимальних маршрутів до всіх мереж.

Таблиця маршрутизації містить інформацію про всі відомі мережі, інтерфейси, через які проходять найкращі маршрути та метрики цих маршрутів.

Процес створення таблиці маршрутизації засобами LSR-протоколу містить такі етапи.

1. Кожний маршрутизатор вивчає інформацію про всі безпосередньо приєднані мережі.
2. Кожний маршрутизатор відправляє в усі безпосередньо приєднані мережі пакет Hello для встановлення так званих сусідських відносин, в результаті чого формується база даних «сусідів».
3. Кожний маршрутизатор формує LSA і відправляє їх всім своїм сусідам.

4. Отримавши LSA, кожний маршрутизатор заносить цю інформацію до власної топологічної бази і відправляє анонс іншим сусідам.

5. Отримавши LSA від усіх маршрутизаторів, кожний маршрутизатор будує граф мережі, вершинами якого є маршрутизатори та локальні мережі, а ребрами – канали зв'язку між ними.

6. Застосовуючи алгоритм Дейкстри, кожний маршрутизатор будує логічну топологію у вигляді дерева, коренем якого є він сам, а гілками – мінімальні маршрути до інших вершин графу.

7. Маршрутизатор заносить до таблиці маршрутизації найкращі маршрути та порти, через які вони доступні.

Одним із найпоширеніших LSR-протоколів є OSPF. Розглянемо процес формування таблиці маршрутизації на маршрутизаторі Rt1 (рис. 4.10).

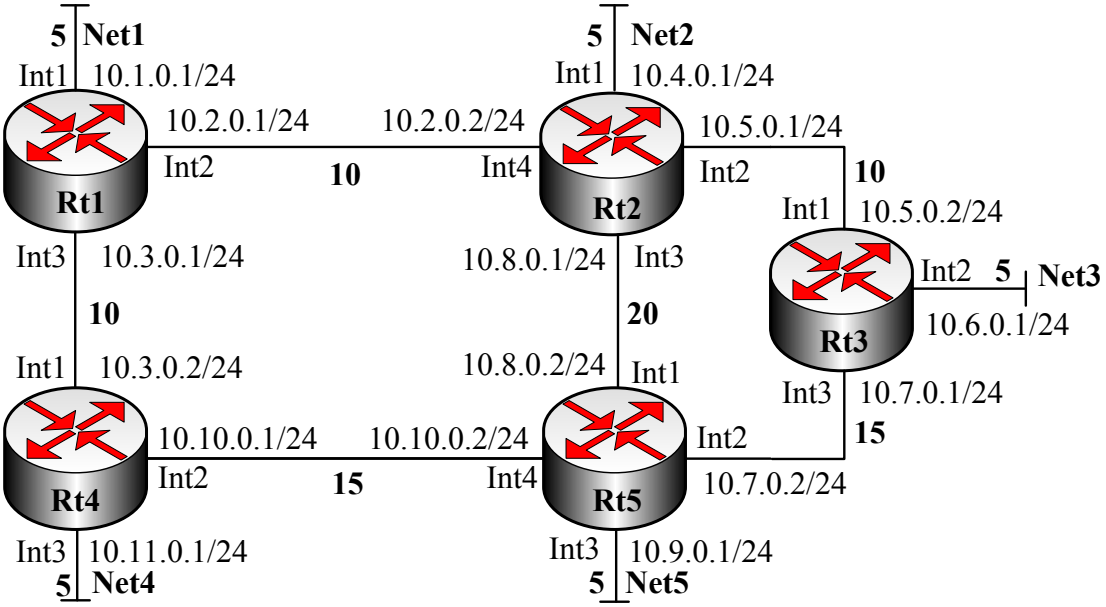


Рисунок 4.10 – Мережа для налаштування OSPF

Після завершення першого і другого етапів на Rt1 буде сформовано таблицю сусідів (табл. 4.4).

Таблиця 4.4 – Таблиця сусідів Rt1

Router	Interface	Network	Next Router	Metric
Rt2	Int2	10.2.0.0/24	10.2.0.2	10
Rt4	Int3	10.3.0.0/24	10.3.0.2	10
-	Int1	10.1.0.0/24	-	5

Аналогічні таблиці створюють і інші маршрутизатори. На основі отриманих таблиць формуються LSA і відправляються сусіднім маршрутизаторам (етап 3).

У процесі обміну LSA-повідомленнями маршрутизатори формують топологічні бази (етап 4), які на завершальному етапі для всіх маршрутизаторів будуть однаковими і матимуть вигляд, наведений в табл. 4.5.

Таблиця 4.5 – Топологічна база протоколу OSPF

	Rt1	Rt2	Rt3	Rt4	Rt5	Net1	Net2	Net3	Net4	Net5
Rt1		10		10		5				
Rt2	10		10		20		5			
Rt3		10			15			5		
Rt4	10				15				5	
Rt5		20	15	15						5

Етап 5 завершується формуванням графу мережі, який наведено на рис. 4.11.

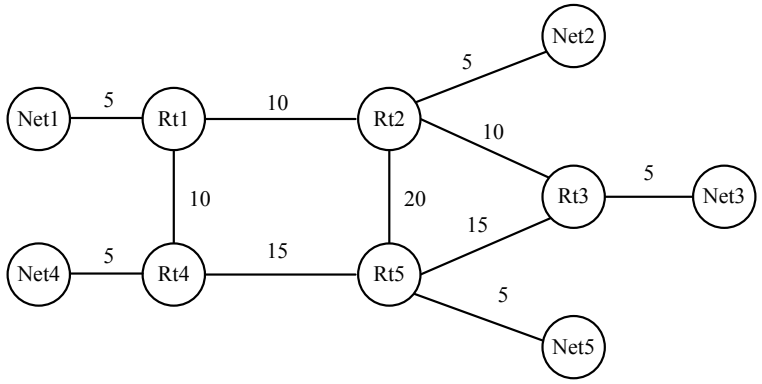


Рисунок 4.11 – Граф, що відповідає мережі, наведеній на рис. 4.10

Застосовуючи алгоритм Дейкстри, кожен маршрутизатор визначає оптимальні маршрути до кожної з мереж (етап 6). Результати виконання цього етапу для маршрутизатора Rt1 наведені на рис. 4.12.

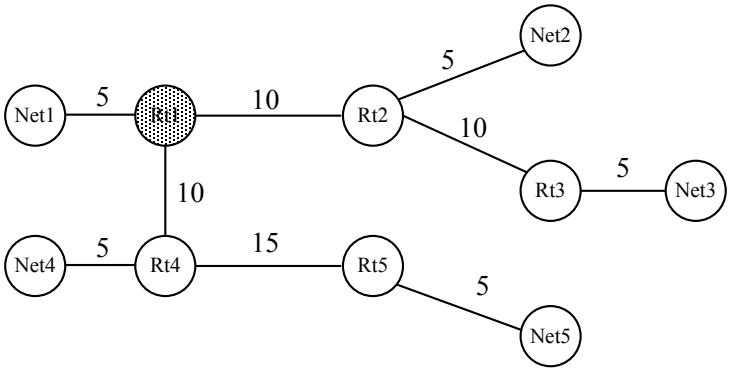


Рисунок 4.12 – Логічна топологія мережі, побудована Rt1

З логічної топології генеруються записи для таблиці маршрутизації Rt1 (табл. 4.6).

Таблиця 4.6 – Таблиця маршрутизації Rt1

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	5
10.2.0.0	255.255.255.0	Int2	-	10
10.3.0.0	255.255.255.0	Int3	-	10
10.4.0.0	255.255.255.0	Int2	10.2.0.2	15
10.5.0.0	255.255.255.0	Int2	10.2.0.2	20
10.6.0.0	255.255.255.0	Int2	10.2.0.2	25
10.7.0.0	255.255.255.0	Int2	10.2.0.2	35
10.8.0.0	255.255.255.0	Int2	10.2.0.2	30
10.9.0.0	255.255.255.0	Int3	10.3.0.2	30
10.10.0.0	255.255.255.0	Int3	10.3.0.2	25
10.11.0.0	255.255.255.0	Int3	10.3.0.2	15

Після завершення формування таблиці маршрутизації кожен маршрутизатор періодично (для протоколу OSPF кожні 10 с) відправляє своїм сусідам пакети Hello, які фактично підтверджують працездатність маршрутизатора та відповідних каналів зв'язку. Це єдиний службовий трафік, що генерується за відсутності змін в топології.

Якщо маршрутизатору стає відомо про зміни топології, як додатні (поява нових мереж), так і від'ємні (втрата зв'язку з існуючими мережами), генеруються відповідні LSA та відсилаються сусіднім маршрутизаторам, які, в свою чергу, розповсюджують цю інформацію далі. Отримавши інформацію про зміни в топології, кожен маршрутизатор відповідно модифікує свою топологічну базу та здійснює перерахунок маршрутів.

Метрика в протоколі OSPF називається **вартістю** (cost). З кожним інтерфейсом маршрутизатора асоціюється певне значення вартості, яке може бути примусово встановлено адміністратором, або розраховано за замовчуванням на основі смуги пропускання. Більшому значенню смуги пропускання відповідає менше значення метрики. Сумарна метрика маршруту визначається шляхом звичайного додавання метрик його окремих ланок.

Особливості роботи LSR-протоколів висувають додаткові вимоги до обладнання, зокрема маршрутизатори мають мати:

- достатню кількість оперативної пам'яті для зберігання топологічної бази, таблиці сусідів, таблиці маршрутизації тощо;
- потужний процесор, який має оперативно розраховувати оптимальні маршрути.

При роботі з LSR-протоколами слід враховувати, що на етапах 3 та 4 відбувається лавиноподібне зростання кількості LSA в мережі, що може тимчасово знизити пропускну спроможність каналів зв'язку. Однак після завершення етапу формування топологічних баз обсяг службового трафіку зводиться до мінімуму.

Оскільки будь-які зміни в топології мережі призводять до перерахунку оптимальних маршрутів усіма маршрутизаторами, використання великої кількості маршрутизаторів та каналів зв'язку може ввести мережу в стан постійної конвергенції. Крім того, збільшення кількості маршрутизаторів збільшує розміри топологічної бази і, таким чином, ускладнює задачу пошуку оптимальних маршрутів. Виходом з такої ситуації є застосування багатозонної маршрутизації, коли одна велика мережа поділяється на кілька окремих доменів маршрутизації.

4.7 Основи функціонування та конфігурування маршрутизаторів

Будь-який маршрутизатор фактично є спеціалізованим комп'ютером. Він містить ті ж самі компоненти, що і звичайний персональний комп'ютер, зокрема, центральний процесор, оперативну та постійну пам'ять, набір різноманітних інтерфейсів тощо. Однак специфіка роботи цього пристрою передбачає і наявність спеціалізованих компонентів. Як і у випадку зі звичайним комп'ютером, для його функціонування необхідна операційна система.

4.7.1 Основні компоненти сучасного маршрутизатора. Центральний процесор (ЦП) виконує такі інструкції операційної системи, як ініціалізація системи, визначає найкращий шлях для передавання пакета, керує процесом передавання пакета з вхідного інтерфейсу на вихідний тощо.

Оперативна пам'ять (ОП) використовується для:

- зберігання файлу операційної системи, який копіюється в ОП в процесі завантаження маршрутизатора;
- зберігання робочого конфігураційного файлу (running-config), який містить набір актуальних команд для операційної системи;
- зберігання таблиці маршрутизації та інших маршрутних таблиць;
- тимчасового зберігання пакетів, що надходять на інтерфейси.

Постійна пам'ять (ПП) використовується для зберігання початкового завантажувача, діагностичного програмного забезпечення (тестує апаратні компоненти після включення пристрою), урізаної версії операційної системи.

Флеш-пам'ять використовується для постійного зберігання файлу операційної системи.

Енергонезалежна оперативна пам'ять (ЕНОП) використовується для зберігання резервного конфігураційного файлу (startup-config). Після завантаження операційної системи startup-config завантажується в ОП і на його основі автоматично утворюється running-config.

Інтерфейси використовуються для під'єднання до маршрутизатора, поділяються на інтерфейси: керування (консольний порт), локальних та розподілених мереж.

Операційна система використовується для інтерпретації конфігураційних файлів, які містять параметри та інструкції керування потоками да-

них, що передаються через маршрутизатор. Найбільш розповсюдженою є операційна система Cisco IOS (Cisco Internetwork Operating System) від корпорації Cisco Systems. Для кожної моделі або групи моделей обладнання розробляється окрема версія IOS, яка має власний набір функцій, що призначені для роботи з відповідним пристроєм (пристроями). Це пояснює існування великого різноманіття версій IOS. Для спрощення ідентифікації Cisco IOS назва файлу операційної системи має фіксовану структуру і складається з таких основних частин:

- назва апаратної платформи, для якої призначено IOS;
- ознака спеціальних можливостей, що підтримуються IOS;
- вказівник форми збереження файлу IOS (ущільнений чи неущільнений);
- власне версія IOS.

Існують два основних способи під'єднання до маршрутизатора з метою його налаштування – локальне та віддалене. При локальному з'єднанні консольний порт маршрутизатора з'єднується з послідовним портом персонального комп'ютера за допомогою спеціального консольного кабелю. На комп'ютері запускається програма, що емулює режим термінала VT100. Консольний доступ є основним способом адміністративного доступу до маршрутизатора, оскільки може бути здійснений за будь-яких обставин. Консольний доступ є єдиним можливим способом доступу при початковому налаштуванні маршрутизатора у випадку виникнення проблем з завантаженням операційної системи або конфігураційного файлу.

Віддалений доступ є можливим за умови коректного налаштування хоча б одного LAN- або WAN-інтерфейсу. Крім того він передбачає наявність фізичного з'єднання з мережею та налаштування на маршрутизаторі сервісів віддаленого з'єднання Telnet або SSH.

4.7.2 Основи роботи з операційною системою Cisco IOS. IOS являє собою специфічне програмне забезпечення, що керує функціями маршрутизатора. Традиційним інтерактивним середовищем Cisco IOS є інтерфейс командного рядка (CLI – Command Line Interface). Інтерфейс командного рядка Cisco має ієрархічну структуру. Для виконання різних задач ця структура потребує переходу в різні режими. Так, для налаштування інтерфейсів маршрутизатора необхідно перейти в режим конфігурування інтерфейсів, в якому можна виконувати тільки дії, пов'язані з певним інтерфейсом, наприклад, призначення IP-адреси. В різних режимах CLI має різні позначки запрошення, що дозволяє адміністратору використовувати лише ті команди, що притаманні певному режиму.

Операційна система Cisco IOS забезпечує роботу інтерпретатора команд (EXEC). Інтерпретатор перевіряє і виконує всі команди, що були введені з консолі. З метою безпеки Cisco IOS EXEC-режими розділені на два базових рівні доступу – користувацький (user EXEC mode) і привілейований (privileged EXEC mode).

У користувацькому режимі доступний обмежений набір основних команд, що дозволяє тільки відслідковувати роботу маршрутизатора. Користувацький режим не дозволяє вносити ніяких змін в конфігурацію маршрутизатора. Адміністратор використовує цей режим тільки для моніторингу системи. Для ідентифікації користувацького режиму в запрошенні командного рядка використовується позначка «>», якій передує ім'я маршрутизатора (за замовчуванням ім'я маршрутизатора Router).

Привілейований режим доступу дає можливість використовувати повний набір команд маршрутизатора. Як правило, доступ до цього режиму захищають паролем. Для виконання будь-яких команд налаштування і керування маршрутизатором необхідно перейти в привілейований режим. Інтерфейс командного рядка ідентифікує цей режим позначкою «#». Для переходу з користувацького режиму в привілейований використовується команда *enable*.

Для виконання конкретних конфігураційних дій потрібно перейти в режим глобальної конфігурації:

Router# configure terminal
Router(config)#

У цьому режимі можна виконувати конфігураційні задачі, що стосуються пристрою в цілому, наприклад, змінити ім'я маршрутизатора, призначити пароль на перехід в привілейований режим, створити статичний маршрутний запис тощо.

Для конфігурування окремих компонентів маршрутизатора (інтерфейсів, віртуальних ліній, протоколів динамічної маршрутизації) потрібно з режиму глобальної конфігурації перейти у відповідний режим. Наприклад, для переходу в режим конфігурування інтерфейсу FastEthernet0/0 потрібно ввести команду:

Router(config)#interface FastEthernet0/0
Router(config-if)#

Перехід з одного рівня ієрархії на попередній здійснюється командою *exit*, а перехід з будь-якого рівня відразу на найвищий – командою *end*. Спрощену структуру ієрархічних рівнів CLI Cisco IOS наведено на рис. 4.13.

Для спрощення роботи адміністратора Cisco IOS забезпечує різноманітні функції допомоги. Функція контекстної допомоги дозволяє переглянути перелік команд, які підтримуються в поточному режимі, або набір ключових слів, що їх використовує та чи інша команда. Виклик цієї функції здійснюється за допомогою символа «?», що вводиться у відповідному контексті або після певної команди. Функція перевірки синтаксису спрощує пошук помилок в імені команди або ключових словах. Крім того існує набір різноманітних функцій, що дозволяє пришвидшити введення команд, наприклад, можливість виклику та редагування попередніх команд, використання тільки початкових літер команди або ключового слова, використання так званих «гарячих клавіш».

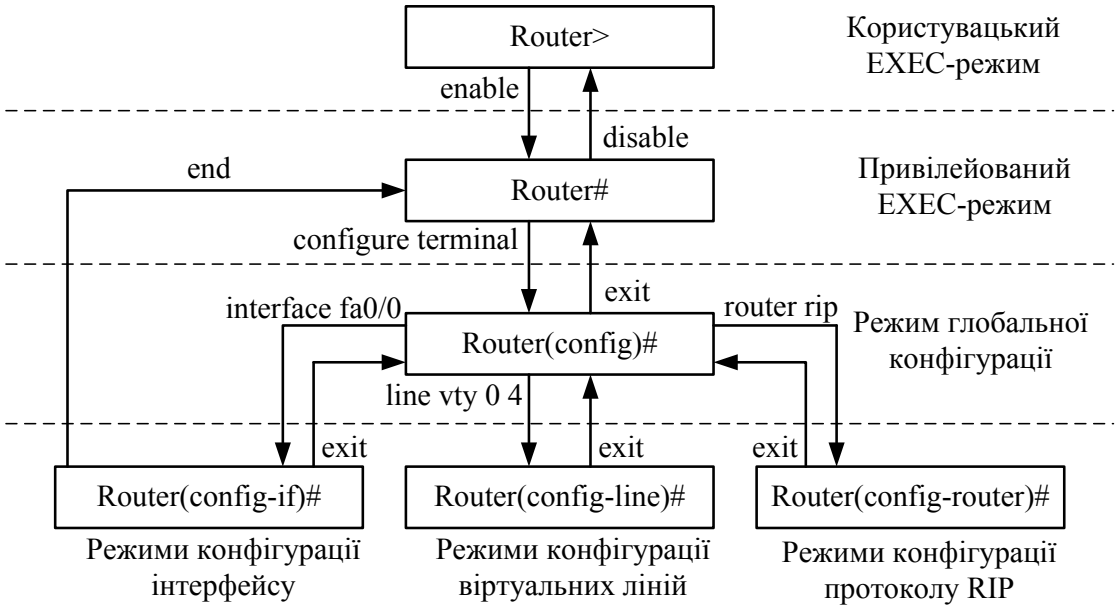


Рисунок 4.13 – Структура ієрархічних рівнів CLI Cisco IOS

4.7.3 Початкове конфігурування маршрутизатора Cisco. Незалежно від типу маршрутизатора та версії IOS існує базовий набір конфігураційних дій, який, як правило, передує налаштуванню власне маршрутизації. Сюди можна віднести:

- конфігурування імені маршрутизатора;
- конфігурування паролю на привілейований режим;
- конфігурування віддаленого доступу;
- конфігурування інтерфейсів;
- перевірка здійснених налаштувань;
- збереження конфігураційного файлу.

Конфігурування імені маршрутизатора здійснюється в режимі глобальної конфігурації. Для призначення маршрутизатору імені *Rt1* необхідно ввести команду *Router(config)#hostname Rt1*. Після чого рядок запрошення набуде вигляду *Rt1(config)#.*

Конфігурування паролю на привілейований режим здійснюється з метою запобігання несанкціонованого доступу до нього. Для цього може бути використано одну з двох команд:

Rt1(config)#enable password pass1
Rt1(config)#enable secret pass2

У першому випадку пароль *pass1* буде зберігатись в конфігураційному файлі у відкритому вигляді, в другому випадку пароль *pass2* буде зашифровано, що підвищує захищеність маршрутизатора. У випадку одночасного застосування обох команд актуальним паролем буде *pass2.*

Іншою важливою задачею початкового конфігурування є забезпечення можливості віддаленого доступу до маршрутизатора. За замовчуванням цей доступ здійснюється за допомогою протоколу Telnet. Для налаштуван-

ня цієї можливості необхідно перейти в режим конфігурування віртуальних ліній:

Rt1(config)#line vty 0 4
Rt1(config-line)#password pass3
Rt1(config-line)#login

У наведеному прикладі відкрито доступ до п'яти віртуальних ліній (0÷4) з однаковим паролем *pass3.* Команда *login* в даному випадку вимагає запитувати пароль під час віддаленого доступу. Для підвищення захищеності віддаленого доступу в сучасних маршрутизаторах замість протоколу Telnet застосовують SSH.

Конфігурування інтерфейсів маршрутизатора в найпростішому випадку передбачає призначення IP-адреси і маски підмережі та включення інтерфейсу:

Rt1(config)#interface FastEthernet 0/0
Rt1(config-if)#ip address 10.1.0.1 255.255.255.0
Rt1(config-if)#no shutdown

При конфігуруванні послідовних інтерфейсів в деяких випадках може виникнути необхідність додаткового призначення смуги пропускання та частоти синхронізації.

Перевірку коректності введених команд і параметрів може бути виконано шляхом перегляду робочого конфігураційного файлу за допомогою команди *Rt1#show running-config.*

Для збереження зроблених налаштувань в енергонезалежній пам'яті слід використати команду *Rt1#copy running-config startup-config,* або її скорочений варіант *Rt1#copy run start.*

4.7.4 Конфігурування статичної та динамічної маршрутизацій. Базова функція маршрутизатора визначається його назвою. Для забезпечення функції маршрутизації обов'язковим є виконання як мінімум двох умов. По-перше, на маршрутизаторі мають бути включені і правильно налаштовані як мінімум два інтерфейси. По-друге, на маршрутизаторі має бути коректна таблиця маршрутизації.

Всі записи в таблиці маршрутизації, з точки зору їх походження, можна поділити на три категорії:

- автоматичні, які містять інформацію про безпосередньо приєднані мережі, з'являються після завершення конфігурування інтерфейсів;
- статичні, які створюються власноручно адміністратором;
- динамічні, які генеруються протоколами динамічної маршрутизації.

Статична маршрутизація, як правило, використовується в невеликих мережах. Нехай на маршрутизаторі Rt1 (див. рис. 4.8) необхідно додати статичний запис для мережі 10.3.0.0. Створення статичних маршрутів здійснюється в режимі глобальної конфігурації за допомогою команди.

Rt1(config)#ip route 10.3.0.0 255.255.255.0 10.2.0.2.

Для перегляду маршрутної інформації використовується команда

Rt1#show ip route .

Результат виконання команди наведений на рис. 4.14.

```
     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.0.0 is directly connected, FastEthernet1/0
C       10.2.0.0 is directly connected, FastEthernet2/0
S       10.3.0.0 [1/0] via 10.2.0.2
C       10.4.0.0 is directly connected, FastEthernet3/0
```

Рисунок 4.14 – Маршрутна інформація на Rt1 після створення статичного маршруту

На рис. 4.14 можна побачити записи про три безпосередньо приєднані мережі: 10.1.0.0, 10.2.0.0, 10.4.0.0 та один статичний запис про мережу 10.3.0.0. Для коректної роботи маршрутизатора Rt1 необхідно додати статичні записи для інших мереж (10.5.0.0÷10.8.0.0).

Якщо виникає потреба видалити статичний запис (було допущено помилку при його створенні або мережа стала недосяжною), використовується та ж сама команда, якій передує «no». Наприклад, для видалення створеного нами статичного запису видається команда

Rt1(config)#no ip route 10.3.0.0 255.255.255.0 10.2.0.2.

Процедура конфігурування динамічної маршрутизації, незалежно від типу протоколу, містить два основних кроки:

- активізація протоколу динамічної маршрутизації;
- визначення переліку мереж, про які будуть відсилатись анонси.

Налаштування протоколу RIP на Rt1 здійснюється таким чином:

Rt1(config)# router rip
Rt1(config-router)# network 10.0.0.0.

Слід звернути увагу, що за допомогою команди «*network*» вказуються всі безпосередньо приєднані мережі, про які маршрутизатор буде інформувати сусідів. Для Rt1 такими мережами є 10.1.0.0, 10.2.0.0, 10.4.0.0. Однак оскільки вони всі належать до однієї мережі класу А, а саме 10.0.0.0, то достатньо одного запису замість трьох.

Після завершення конфігурування RIP на всіх маршрутизаторах команда «*show ip route*» на Rt1 видасть інформацію, наведену на рис. 4.15.

Літера «R», яка передує записам в таблиці маршрутизації, свідчить що вони створені саме протоколом RIP. Два числа в квадратних дужках визначають відповідно адміністративну відстань та метрику. Адміністративна відстань – це число, яке характеризує ступінь пріоритетності цього запису і використовується в тому випадку, коли для певної мережі в таблиці маршрутизації є два однакових записи, отриманих з різних джерел (наприклад, один був створений статично, а інший отримано динамічно). В такому випадку перевага надається запису, що має меншу адміністративну відстань. 120 – адміністративна відстань протоколу RIP за замовчуванням. На рис. 4.15 також можна побачити, скільки часу пройшло з моменту отримання останнього оновлення про ту чи іншу мережу.

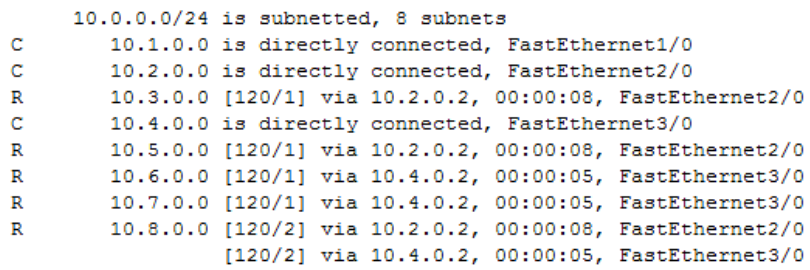


Рисунок 4.15 – Маршрутна інформація на Rt1 після налаштування RIP

Для конфігурування OSPF на маршрутизаторі Rt1 (див. рис. 4.10) потрібно ввести такі команди:

Rt1(config)# router ospf 1
Rt1(config-router)# network 10.1.0.0 0.0.0.255 area 0
Rt1(config-router)# network 10.2.0.0 0.0.0.255 area 0
Rt1(config-router)# network 10.3.0.0 0.0.0.255 area 0.

«*1*» в першому рядку визначає ідентифікатор OSPF-процесу. Це число має локальне значення і не обов'язково має повторюватись на інших маршрутизаторах.

«*area 0*» задає номер зони маршрутизації, який в даному випадку має бути однаковим для всіх мереж і всіх маршрутизаторів.

«*0.0.0.255*» – інвертована маска (wild card mask), яка в даному випадку виконує ту ж саму функцію, що і звичайна маска.

Для змінення метрики інтерфейсу використовується команда

Rt1(config-if)# ip ospf cost 5,

де «*5*» – нове значення ospf-метрики.

Після здійснення відповідних налаштувань на всіх маршрутизаторах маршрутна інформація на Rt1 набуває вигляду, наведеного на рис. 4.16.

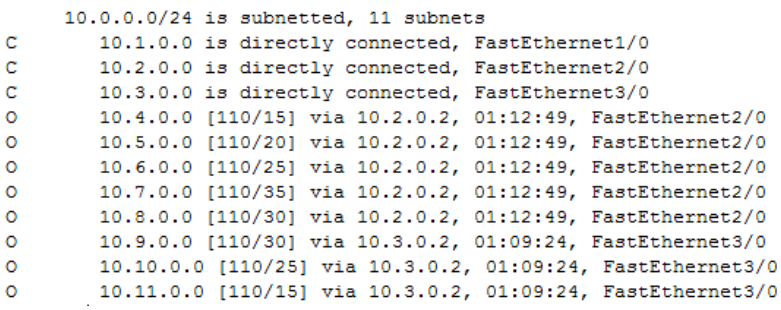


Рисунок 4.16 – Маршрутна інформація на Rt1 після налаштування OSPF

4.8 Особливості протоколу IPv6

Проблема нестачі IP-адрес існує вже декілька років. В червні 1992 року співтовариство Internet для вирішення цієї проблеми розробило три пропозиції щодо протоколу IP нової версії: «TCP and UDP with Biggest Addresses (TUBA)», «Common Architeature for the Internet (CathIP)» та «Simple Internet Protocol Plus (SIPP)». Після аналізу цих документів в січні 1995 року була

опублікована рекомендація щодо протоколу IP наступного покоління «The Recommendation for the IP Next Generation Protocol», яка описана в документі RFC 1752.

На сьогодні протокол IPv6 вже використовується в декількох тисячах мереж в усьому світі. IPv6 – це нова версія протоколу IP, що розроблена для заміни старої версії IPv4. Відмінності протоколу IPv6 від IPv4 зводяться до наступного.

Зміна та розширення адресного простору. В протоколі IPv6 довжина адреси збільшена до 128 бітів (замість 32 розрядів в IPv4). Отже, адресний простір збільшується в 296 разів. Крім того, в IPv6 передбачена можливість створення адресної ієрархії зі значно більшою кількістю рівнів, ніж передбачена протоколом IPv4.

Зміна формату заголовка пакета. Деякі поля заголовка IPv4 вилучені (у силу непотрібності чи неефективності використання) або стали необов'язковими для використання. Введено також декілька таких нових функцій, як поле мітки для ідентифікації потоку пакетів, що вимагають спеціальної обробки; розширення заголовка для спрощення операцій шифрування та ідентифікації, а також заголовок маршрутизації.

Збільшення продуктивності маршрутизаторів. При використанні протоколу IPv6 швидкість обробки пакетів при звичайній завантаженості мережі буде вища, ніж у пакетів протоколу IPv4. Це пояснюється тим, що, на відміну від протоколу IPv4, в якому маршрути від джерела подані в додатковій опції заголовка і кожний маршрутизатор має перевіряти та аналізувати значення цього параметра, в пакеті протоколу IPv6 значення заголовка маршрутизації аналізується тільки у випадку, якщо маршрутизатор виявить одну зі своїх адрес у полі Destination address основного заголовка пакета IPv6. Крім того, фрагментація пакетів можлива тільки на станції, яка відправляє пакети, що приводить до збільшення продуктивності роботи маршрутизаторів. Станція-відправник при цьому має визначити максимальний розмір блоку передавання даних MTU на шляху до станції-отримувача і фрагментувати пакет відповідно до значення MTU. Для більш ефективної роботи мережі протокол IPv6 вимагає, щоб в усіх проміжних ланках передавання даних розмір MTU був би не нижче 1280 байтів (замість 576 байтів для протоколу IPv4).

Поява можливості маркування потоку даних. З'являється можливість ідентифікувати потоки пакетів, які належать конкретному додатку відправника, і який необхідно обробляти особливим чином (наприклад, обробка даних у реальному часі, нестандартний тип сервісу ToS тощо).

Додавання полів для аутентифікації пакетів. В IPv6 введено опції аутентифікації для забезпечення захисту конфіденційної інформації, що передається в мережі, та ідентифікації мережних модулів.

Формат заголовка протоколу IPv6 описано в документах RFC 1883 та RFC 2460 (рис. 4.17).

Основний заголовок протоколу IPv6 становить 40 байтів (з них 32 байти використовуються для передавання IP-адрес відправника та отримувача).

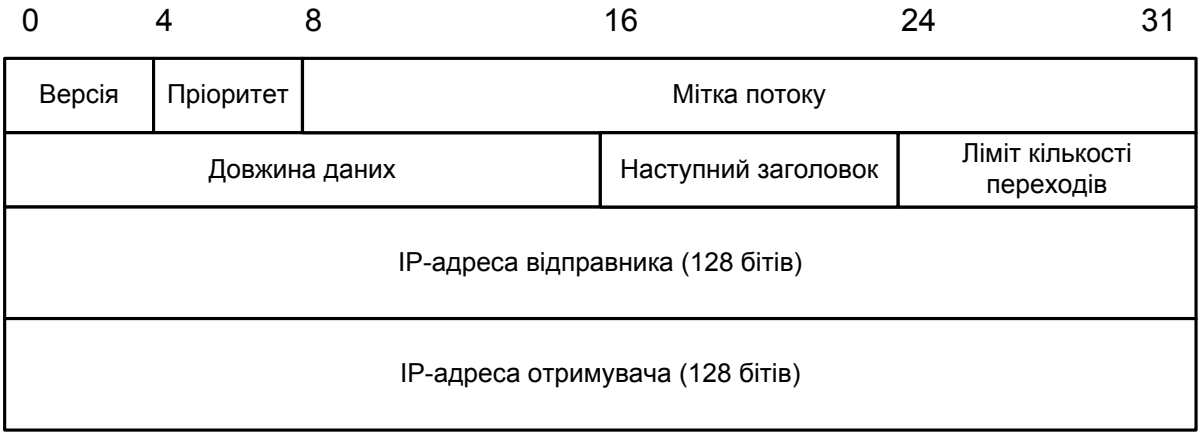


Рисунок 4.17 – Формат основного заголовка пакета IPv6

Версія – значення поля дорівнює номеру версії протоколу, тобто 6.

Пріоритет (згідно з RFC 1883) – поле дозволяє відправнику призначати пакету певний рівень пріоритету. Можливі 16 значень пріоритету розділені на дві групи:

- пріоритети 0–7 використовуються для пакетів, які не можуть бути передані при занадто переповненій лінії. Сюди відносять: TCP-трафік, зокрема передавання E-mail, трафік сервісів FTP, NFS, TELNET, X-interactive;
- пріоритети 8–15 призначаються пакетам, які мають бути відправлені при будь-якому стані лінії (крім обриву). Наприклад, пріоритет рівня 8 користувач може призначити пакетам, які можна відправити в останню чергу при перевантаженій лінії, а пріоритет 15 – в першу (це пакети реального часу з відео-, аудіо- та аналогічними даними, які мають передаватися з постійною швидкістю).

В останніх модифікаціях протоколу відповідно до документа RFC 2460 поле **Пріоритет** замінено на **Клас трафіку**, яке розширене на 1 тетраду за рахунок поля **Мітка потоку** і має розмір 1 байт. При цьому перші три біти визначають клас трафіку, а інші – пріоритет. Рекомендовані класи трафіку для різних категорій прикладних застосувань наведено в таблиці 4.7.

Мітка потоку. Це поле може використовуватися відправником для того, щоб відмічати пакети, які вимагають спеціальної обробки в комунікаційних модулях мережі (на шлюзах, маршрутизаторах тощо). Хости та шлюзи, які не підтримують цієї опції, мають встановити це поле в 0 та ігнорувати при обробці пакета. Потік – це послідовність пакетів, що їх відправляють певному одержувачеві (або групі одержувачів), на шляху до яких пакети мають пройти спеціальну обробку (наприклад, інформація про проходження певного потоку буде реєструватися.). Міткою потоку служить

псевдовипадкове число в діапазоні 1–(F)FFFFF. Якщо значення поля дорівнює нулю, то вважається, що пакет не належить жодному потоку.

Таблиця 4.7 – Характеристика класів трафіку

Клас трафіку	Призначення
0	Нехарактеризований трафік
1	Заповнювальний трафік (мережні новини)
2	Несуттєвий інформаційний трафік (електронна пошта)
3	Зарезервовано
4	Суттєвий трафік (трафік сервісів FTP, HTTP, NFS)
5	Зарезервовано
6	Інтерактивний трафік (трафік сервісів Telnet, X-terminal, SSH)
7	Трафік керування (маршрутна інформація, повідомлення SNMP)

Довжина даних. Поле визначає довжину даних пакета (в байтах), які розташовані одразу за заголовком. Якщо значення поля дорівнює 0, то довжина даних пакета більше 65535 байтів, і її значення зберігається у відповідному полі додаткового заголовка Hop-by-Hop. Дане поле подібно полю **Довжина заголовка** (IHL – Internet Header Length) в пакеті протоколу IPv4, але якщо дане поле містить довжину даних після заголовка, то поле **IHL** визначає довжину самого заголовка пакета.

Поле наступного заголовка містить інформацію про тип заголовка, який розташований за основним заголовком IPv6 (як і поле **Протокол верхнього рівня** протоколу визначає, дані якого транспортного протоколу знаходяться за IP-заголовком). В IPv6 дане поле дозволяє вставляти додаткові заголовки між даними протоколів IP та TCP або UDP. Використовуються ті ж значення кодів протоколів, що і протоколом IPv4, та описані в документі RFC 1700 (наприклад, значення 1 визначає протокол ICMP, 6 – протокол TCP, 17 – протокол UDP).

Ліміт кількості переходів. Це поле аналогічно полю **Час життя** (TTL – Time to Live) протоколу IPv4 і визначає максимальну кількість проміжних комунікаційних вузлів, через які може бути передано пакет. Величина зменшується на 1 при проходженні пакета через кожний шлюз або маршрутизатор. Якщо поле стає нульовим, то пакет вилучається, а його відправнику засобами протоколу ICMP передається повідомлення про видалення пакета.

Адреса відправника. IP-адреса відправника пакета (128 бітів), структура якої описана в документі RFC 1884.

Адреса отримувача – IP-адреса отримувача пакета (128 бітів). Якщо заголовок пакета містить додатковий заголовок маршрутизації, це поле може визначати не адресу призначення кінцевої станції, а адресу проміжного маршрутизатора.

Існує два формати пакетів IPv6: без додаткових заголовків (рис. 4.18) та з деякою кількістю додаткових заголовків (рис. 4.19), які записуються між основним заголовком IPv6 пакета та заголовком протоколу верхнього рівня (зазвичай, це протокол TCP чи UDP).

Рисунок 4. 18 – Структура пакета IPv6 без додаткових заголовків

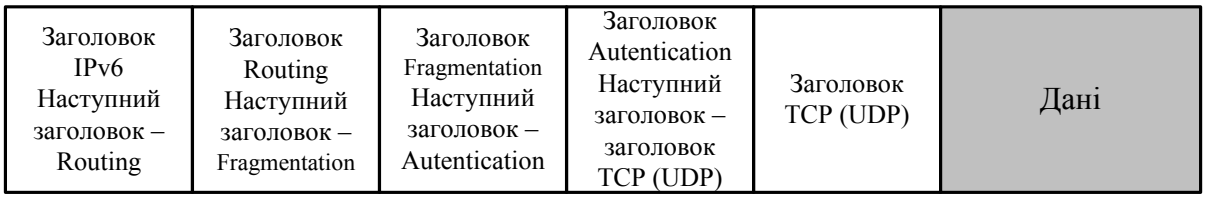


Рисунок 4.19 – Структура пакета IPv6 з декількома додатковими заголовками

На сьогодні стандартизовано декілька додаткових заголовків, кожний з яких задається своїм значенням коду поля **Наступний заголовок**. Стандартизовано такі додаткові заголовки: фрагментації, маршрутизації, інкапсуляції, аутентифікації, опцій Hop-by-Hop, місця призначення та відсутності наступного заголовка. Ці додаткові заголовки розширення зазвичай не аналізуються та не обробляються комунікаційними вузлами на маршруті передавання. Семантика та вміст кожного заголовка визначають необхідність обробки наступного заголовка в пакеті. Тобто, додаткові заголовки потрібно обробляти строго в порядку їх розміщення в пакеті. Єдиний виняток з цього правила стосується опцій заголовка Hop-by-Hop, який призначений для передавання інформації, яку необхідно обробляти кожним комунікаційним вузлом на маршруті передавання, охоплюючи відправника і отримувача. Якщо цей заголовок є в пакеті, то він обов’язково має бути записаний одразу після основного заголовка пакета IPv6. У табл. 4.8 наведені значення полів для визначення типу додаткових заголовків.

У разі використання в одному пакеті більше одного додаткового заголовка стандарт рекомендує розташовувати їх у такому порядку:

- заголовок пакета (основний);
- заголовок опцій Hop-by-Hop;
- заголовок опцій місця призначення Destination Options;
- заголовок маршрутизації Routing;
- заголовок фрагментації Fragment;
- заголовок аутентифікації Authentication;

- заголовок безпечних вкладень Encapsulating Security Payload;
- заголовок опцій місця призначення Destination Options;
- заголовок протоколу верхнього рівня (TCP, UDP тощо).

Таблиця 4.8 – Значення поля **Наступний заголовок**

Додатковий заголовок	Значення (десяткове)	Призначення	Розмір	RFC
Hop-by-Hop	0	Містить інформацію для комунікаційних вузлів на маршруті передавання пакета.	-	2460
Routing	43	Дозволяє відправнику визначити перелік вузлів, через які пакет обов'язково має бути переданий.	-	2460 3775 5095
Fragmentation	44	Міститься інформація про фрагментацію пакета.	64	2460
Encapsulating Security Payload (ESP)	50	Забезпечення конфіденційності даних (входить в протокол IPSec).	-	4303
Autentication Header (AH)	51	Служить для ідентифікації кінцевих вузлів та забезпечення цілісності пакетів (входить в протокол IPSec).	-	4302
No Next Header	59	Відсутність наступного заголовка. Дані, які знаходяться за цим заголовком, мають ігноруватися і передаватися без змін.	-	2460
Destination Option	60	Опції, які мають оброблятися тільки станцією-отримувачем.	-	2460

Відмінність заголовків опцій місця призначення Destination Options Header першого та другого типів полягає в тому, що опції заголовка першого типу мають оброблятися не тільки в станції призначення, адреса якої зазначена в полі пакета **Адреса отримувача**, а й в усіх комунікаційних вузлах, адреси яких наведено в заголовку маршрутизації. Опції заголовка другого типу мають оброблятися тільки в кінцевому вузлі.

Кожний додатковий заголовок розширення може міститись в пакеті тільки один раз, за винятком описаного вище випадку. Якщо за заголовком верхнього рівня є ще один заголовок IPv6 (у випадку тунелювання чи інкапсуляції в пакет IPv6), то за ним можуть міститися його додаткові заголовки.

4.9 Адресація в IPv6

Існують три стандартні форми подання адрес IPv6.

1. **Основна форма**, при використанні якої 128-бітова адреса подається сукупністю з восьми блоків, кожний з яких містить шістнадцяткові 16-бітові числа. Наприклад:

0125:A7C8:3542:F7DB:89E5:91A4:FFEE:5425
8210:DC07:40:7:0:0:4521:3
(при цьому можна не наводити початкові нулі в кожному з блоків).

2. **Стисла (скорочена) форма**. Особливості адреси IPv6 призводять до того, що вона часто містить довгі послідовності нульових бітів. Для того, щоб зробити запис більш компактним і зручним в користуванні, розроблено спеціальний синтаксис для видалення послідовності нульових бітів. Наприклад:

Призначення	**Основна форма**	**Стисла форма**
unicast-адреса	2835:0:0:0:57:700:100D:7792	2835:: 57:700:100D:7792
multicast-адреса	FF01:0:0:0:0:0:0:43	FF01::43
адреса loopback	0:0:0:0:0:0:0:1	::1
	0:0:0:0:0:0:0:0	::

3. **Альтернативна форма**. Цей запис дуже зручний при роботі з адресами IPv4 та IPv6, в якому в молодших 32 розрядах подається стандартна адреса IPv4 в десятковій формі. Наприклад:

   0:0:0:0:0:0:71.18.33.10 чи :: 71.18.33.10
   0:0:FFFF:0:0:0:201.54.32.7 чи 0:0: FFFF:: 201.54.32.7.

Протокол IPv6 передбачає використання 3-х типів адрес.

Персональні (Unicast) – ідентифікатор індивідуального (одного) інтерфейсу. Адреса визначає окремий модуль мережі або порт маршрутизатора і, в свою чергу, може бути:

Групові (Multicast) – ідентифікатор сукупності інтерфейсів (адреса набору вузлів). У протоколі IPv6 відсутнє поняття широкомовної адреси. Широкомовна адресація замінена підтримкою групового передавання даних. Такий механізм необхідний протоколу IPv6 для керування пропускною спроможністю мережі при передаванні мультимедийного трафіка. Пакет, який відправляється за такою адресою, передається всім інтерфейсам модулів, які задаються такою адресою.

Адреса довільного розсилання (Anycast) – ідентифікатор набору вузлів. Цей тип адрес використовується для забезпечення проходження трафіка через маршрутизатори окремих провайдерів. На відміну від групових адрес, такий пакет має бути доставлений будь-якому члену групи (зазвичай, передається в найближчий вузол).

При призначенні адреси кожному порту маршрутизатора разом з персональною адресою присвоюється ще одна адреса, загальна для всіх портів

всіх маршрутизаторів в мережі даного провайдера (ця адреса і є anycast-адресою).

Виділяють такі різновиди персональних адрес:

- **Глобальні персональні (Global unicast)** – основний тип адрес в Internet, є аналогічними публічним IPv4 адресам, глобально унікальні, маршрутизуються в мережі Інтернет, можуть призначатись статично або динамічно;
- **Локальні адреси лінії (Link-local)** – використовуються тільки в локальному каналі, автоматично конфігуруються на всіх інтерфейсах, префікс, що використовують Link-Local адреси – FE80::X/10, маршрутизатори не передають пакети з Link-local адресами відправника або одержувача, здебільше використовуються в службових цілях і є обов'язковими;
- **Самотестування (Loopback)** – позначається 0:0:0:0:0:0:0:1 або ::1, є аналогом IPv4 адреси 127.0.0.1, використовується для відправлення пакета самому собі;
- **Невизначена (Unspecified)** – позначається 0:0:0:0:0:0:0:0 або **::,** не може бути призначена інтерфейсу, використовується як адреса відправника за відсутності IP-адреси;
- **Локальні унікальні (Unique local)** – визначені в RFC 4193, є подібними до приватних адрес IPv4, маршрутизуються в межах окремої мережі, префікс FC00::/7;
- **IPv4 вбудовані (IPv4 embedded)** – використовуються при переході з IPv4 на IPv6 та поділяються на адреси для трансляції IPv4 в IPv6 **(**::FFFF:0:0:0/96) та адреси для відображення IPv6 на IPv4 **(**::FFFF:0:0/96).

Схема адресації протоколу IPv6 суттєво відрізняється від протоколу IPv4. 128-бітова довжина адреси простору дозволяє зняти дефіцит адрес у мережі Internet. В схемі глобальної персональної адресації IPv6 закладений ієрархічний розподіл адресного простору на окремі рівні. І замість двох або трьох рівнів в адресі IPv4 в протоколі IPv6 використовується 5 рівнів, охоплюючи: 2 рівні ідентифікації провайдерів та 3 рівні ідентифікації абонентів у мережі. Загальна структура адреси протоколу IPv6 наведена на рис. 4.20.

Префікс	Ідентифікатор провайдера	Ідентифікатор абонента	Ідентифікатор підмережі	Ідентифікатор вузла

Рисунок 4.20 – Загальна структура адреси протоколу IPv6

В адресному просторі IPv6 відмінено поділ адрес на класи. В основі розподілу адресного простору лежить технологія безкласової міждоменної

маршрутизації CIDR. Тип IP-адреси визначається префіксом формату (FR – Format Prefix) змінної довжини. Розподіл адресного простору і мету використання адреси, залежно від значення префікса, наведено в RFC 1884. Всі поля адреси є змінними, а їх довжина та структура можуть змінюватися залежно від значення префікса. Такий розподіл адрес підтримує пряме виділення адрес провайдера, адрес локального використання та групових адрес.

Для прикладу на рис. 4.21 та 4.22 наведено, відповідно, структуру адреси IPv6 з префіксом 010, який використовується для адрес з ідентифікацією провайдера в межах глобально унікальних адрес, та локальні адреси лінії.

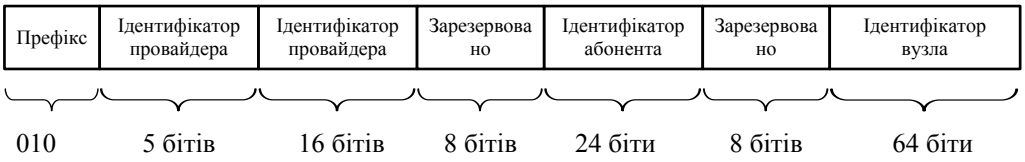


Рисунок 4.21 – Формат адреси з ідентифікацією провайдера

Префікс (1111 1110 1000)	000 . . . 000	Унікальна адреса лінії
	. . .	
Префікс (1111 1110 1011)	000 . . . 000	Унікальна адреса лінії

Рисунок 4.22 – Локальні адреси лінії

Для призначення IPv6 адрес використовуються статичні і динамічні методи. При **статичному** методі адреса хосту призначається вручну адміністратором. Як правило, цей підхід використовують для призначення адрес, які мають бути відомими і не змінюватись, наприклад: адреса інтерфейсу маршрутизатора, адреса мережного принтера, адреса сервера. Існують два різновиди статичного методу: звичайний і з використанням **інтерфейсу EUI-64**, останній передбачає призначення тільки мережної частини адреси, хостова частина генерується автоматично на основі MAC-адреси за правилом, що буде описано нижче.

Динамічний спосіб призначення IPv6 адрес також має свої різновиди. Серед них повнофункціональний **DHCP для IPv6** (**DHCPv6**), при якому, як і для IPv4, хосту, призначається повний набір адресної інформації: повна IP-адреса, довжина префікса, адреса шлюзу, адреса DNS сервера тощо. Цей спосіб орієнтований на робочі станції кінцевих користувачів. Другий спосіб – **автоконфігурування (Stateless Address Autoconfiguration, SLAAC)** не передбачає наявності DHCPv6, натомість достатньо звичайного маршрутизатора, на якому включено IPv6-маршрутизацію. Клієнтська

станція за запитом отримує у маршрутизатора адресу мережі, довжину префікса і адресу шлюзу, хостова частина адреси генерується випадковим чином або з використанням опції EUI-64. Цей спосіб призначення адрес орієнтований на побутові пристрої, пристрої Інтернету речей тощо. Третій різновид призначення адрес **(Stateless DHCP)** є комбінацією першого та другого і передбачає доповнення адресної інформації, отриманої за допомогою автоконфігурування, додатковими даними, отриманими з DHCPv6 сервера, наприклад про адресу DNS-сервера.

Розглянемо процедуру перетворення 48-бітової MAC-адреси в 64-бітовий ідентифікатор вузла, який використовується протоколом IPv6.

Відомо, що більшість мережних протоколів канального рівня використовує один з трьох просторів MAC-адрес, які керуються IEEE: MAC-48, EUI-48 та EUI-64.

MAC-адреса (MAC-48) використовується в багатьох протоколах Ethernet (стандарти IEEE 802.2 та IEEE 802.3), Token Ring, FDDI, Wi-Fi (стандарт IEEE 802.11) та ін.

Унікальний ідентифікатор **OUI** (Organizationally Unique Identifier) – це 24-бітовий номер, який надається реєстраційним підрозділом IEEE (Registration Authority Committee) і формує старші три байти MAC-адреси.

Розширений ідентифікатор **EUI-48** (Extended Unique Identifier) використовується для інших типів апаратного та програмного забезпечення (наприклад, мережних протоколів).

Інститут IEEE вважає термін MAC-48 застарілим і розглядається його як окремий випадок використання ідентифікатора EUI-48 для стандартів IEEE 802.x та ін. В подальшому виробники та інші організації мають використовувати визначення EUI-48. Ідентифікатори MAC-48 і EUI-48 ідентичні при самостійному використанні, але є деякі особливості при їх інкапсуляції в EUI-64.

Розширений ідентифікатор **EUI-64** використовується в мережах IPv6 для формування молодших 64 бітів в мережній адресі вузла, а також в мережах FireWire. Ідентифікатор інтерфейсу в форматі EUI-64 складається з трьох частин:

- 24-бітовий OUI на основі MAC-адреси клієнта, в якому сьомий біт є оберненим, тобто, якщо 7-й біт має значення 0, він стає 1, і навпаки;
- всередину вставляється 16-бітове значення FFFE (в шістнадцятковій системі числення) для OUI-48, або FFFF для MAC-48;
- 24-бітовий ідентифікатор пристрою на основі MAC-адреси клієнта.

Процедура перетворення EUI-48 (MAC-48) в EUI-64 (процес EUI-64) наведено на рис. 4.23.

В таблиці 4.9 наведено порівняння основних параметрів протоколів 4-ї та 6-ї версій.

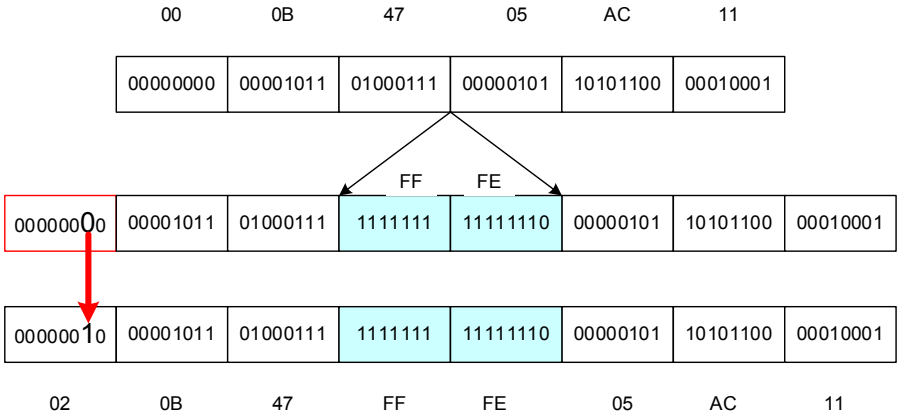


Рисунок 4.23 – Процес визначення розширеного унікального ідентифікатора

Таблиця 4.9 – Порівняння протоколів IPv4 та IPv6

Параметр	**IPv4**	**IPv6**
Розрядність	32 біти	128 бітів
Приклад адреси	154.37.28.201	1025:A7C8:3542:F7DB:89E5:91A4:FFEE:5425 8210:dc07:40:7::421:3
Рік публікації стандарту	1981	1998
Адресний простір:	2^{32} (4 294 967 296) адрес	2^{128} адрес
на людину	≈ 1	$\approx 4{,}7 \cdot 10^{28}$
на 1 кв. км поверхні	≈ 8,5	$\approx 6{,}7 \cdot 10^{26}$
MTU (мінімальний, байт)	576, фрагментація	1280, без фрагментації
Фрагментація	Маршрутизатори і хости	Тільки хости
DNS запису	A PRT in-addr.arpa	AAAA PRT ip6.arpa
Налаштування адреси	Ручне або DHCP	Ручне, StateLess Address AutoConfiguration (SLAAC) і/або DHCPv6
Визначення MAC	Broadcast ARP	Multicast Neighbor Solicitation
Broadcast	Так	Ні
Multicast/Anycast	Так/Так	Так/Так
IPSec заголовки	Опціонально	Обов’язково, але насправді не використовується
Маска під мережі	Так	Ні

4.10 Протоколи ICMPv4 та ICMPv6

Протокол передавання команд і повідомлень про помилки **ICMP (Internet Control Message Protocol)** (RFC 792, 1256) є допоміжним в стеці TCP/IP, але розглядається як невід’ємна частина протоколу IP, тобто використовується всіма хост-вузлами, що функціонують на основі TCP/IP.

Протокол ICMP дозволяє маршрутизатору або іншому комунікаційному вузлу повідомляти станцію-відправника даних про помилки або нештатні ситуації, що виникли при передаванні IP-пакета, виконувати діагностичні функції (наприклад, утиліта ping), передавати значення MTU, здійснювати пошук сусідів тощо.

Необхідно зауважити, що протокол призначено тільки для повідомлення про помилки, що виникли при передаванні, а не для виправлення цих помилок. Відправник має самостійно визначити, де і чому виникла помилка і вжити заходів для її усунення та недопущення в майбутньому. При цьому даний протокол не може використовуватись для передавання повідомлень про помилки будь-яким проміжним вузлам, оскільки IP-пакет містить тільки IP-адреси відправника і отримувача.

ICMP виконує в мережах IP чітко визначені функції, які полягають в нижчевикладеному:

- контроль доступності IP-адреси (echo request, echo respond);
- контроль часу життя пакета в мережі;
- переадресація пакета;
- видача повідомлень про недосяжність адресата або про некоректність параметрів;
- формування та пересилання часових міток;
- видача запитів та відгуків для адресних масок та іншої інформації.

Протокол ICMP використовується для розсилання інформаційних повідомлень та повідомлень керування таких типів:

- **Керування потоком (Flow control)** – якщо хост-отримувач (шлюз або реальний отримувач інформації) не встигає обробляти інформацію, то дане повідомлення інформує про необхідність призупинення відправлення пакетів;
- **Визначення недосяжності отримувача (Detecting unreachable destination)** – якщо пакет не може бути доставлений (з якихось причин) до пункту-призначення, то шлюз, який не може доставити пакет, повідомляє про це відправнику пакета. Інформувати про неможливість доставити повідомлення також може пристрій, IP-адреса якого вказана в пакеті;
- **Зміна маршруту (Redirect routing)** – повідомлення відправляється у випадку, якщо шлюз не може доставити пакет або маршрут через нього не є оптимальним, але в цього шлюзу є інші пропозиції з доставляння, а саме: адреса іншого шлюзу;
- **Перевірка доступності віддаленого хоста (Checking remote host)** – якщо необхідно перевірити наявність стека протоколів TCP/IP на віддаленій станції, використовується повідомлення ICMP Echo Message, отримавши яке віддалена система одразу відповідає, підтверджуючи його отримання.

Протокол ICMP інкапсулює свої повідомлення безпосередньо в IP-пакет. При цьому пакети, що містять ICMP-повідомлення, обробляються

аналогічно пакетам клієнтських сервісних додатків. Це призводить до того, що повідомлення ICMP можуть бути втрачені в процесі передавання, як і будь-який інший пакет. Крім того, для передавання ICMP-повідомлень досить часто обмежується смуга пропускання, що призводить до того, що частина ICMP-повідомлень вилучається проміжними маршрутизаторами. Оскільки ICMP-протокол можуть також використовувати зловмисники для проведення атак розвідницького типу або DoS-атак, певні різновиди ICMP-повідомлень можуть також блокуватись пристроями безпеки.

Структура ICMP-повідомлень визначається типом повідомлення, але обов'язково містить поля **Тип** та **Код** (рис. 4.24, а) і передбачає два етапи інкапсуляції на мережному рівні (рис. 4.24, б).

Рисунок 4.24 – Структура ICMP-повідомлення (а)
та процедура його інкапсуляції в IP-пакет (б)

В таблиці 4.10 наведені типи ICMP-повідомлень, визначені стандартом для використання в тандемі з протоколом IPv4, та їх коди (за потреби), що забезпечує додатковий рівень деталізації повідомлень. Зауважимо, що якщо не вказується версія протоколу (як IP, так і ICMP), мають на увазі саме протоколи 4-ї версії.

Структура кожного з ICMP-повідомлень залежить від його типу і коду, але зазвичай для більш точної ідентифікації в поле **Дані** заноситься IP-заголовок пакета, для якого сформовано це повідомлення, та перші 64 біти області даних IP-пакета.

Треба зазначити, що в тандемі з протоколом IPv6 функціонує інша версія протоколу передавання керівних повідомлень ICMPv6, який має обов'язково підтримуватись кожним вузлом мережі, що працює на основі протоколу IPv6.

Протокол ICMPv6 (RFC 2463, RFC 4443) використовується вузлами з IPv6 для формування повідомлень про помилки, що виникли при обробці IPv6 пакета, діагностиці та передаванні повідомлень про участь вузлів у multicast групах.

Таблиця 4.10 – Типи і коди ICMP-повідомлень

ICMP-повідомлення		**Опис повідомлення**
Тип	**Код**	
0		Echo-відповідь (ping-відклик)
3	0 — 15	Адресат недосяжний
4	0	Відключення відправника при переповненні черги
5	0 — 3	Переадресувати (змінити маршрут)
8	0	Echo-запит (ping-запит)
9	0	Оголошення маршрутизатора
10	0	Запит маршрутизатора
11	0, 1	Для пакета час життя вийшов (TTL=0):
12	0, 1	Проблема з параметрами пакета
13		Запит часової мітки
14		Часова мітка-відповідь
15		Запит інформації (застаріло)
16		Інформаційна відповідь (застаріло)
17		Запит адресної маски
18		Відповідь на запит адресної маски

Інформація ICMPv6-повідомлень може використовуватись протоколами більш високого рівня (транспортного чи рівня додатків) для ліквідації проблем при передаванні. Ця ж інформація може бути використана мережними адміністраторами для виявлення проблем в мережі.

Протокол ICMPv6 використовує два типи повідомлень:

- про помилки (типи 1–4, 100,101, 127), які визначаються документом RFC 4443;
- інформаційні повідомлення (типи 128–153, 200, 201, 255), що описано у документах RFC (2710, 3122, 3775, 3810, 3971, 4065, 4286 та 4443).

Загальна структура повідомлення протоколів ICMPv6 (код 58 в полі **Наступний заголовок** в пакеті IPv6) та ICMPv4 (код 1 в полі **Протокол верхнього рівня** в пакеті IPv4) однакова. Відмінності стосуються тільки кодів типів повідомлень і особливостей структури IPv6 пакета та принципів його обробки.

4.11 Взаємодія протоколів IPv6 та IPv4

На сьогодні переважна кількість хостів мережі Internet функціонує на базі протоколу IPv4, але вже існує велика кількість мереж і хостів в них, які підтримують IP-протокол нового покоління. Тому виникає необхідність забезпечення коректного функціонування та взаємодії хостів з різними версіями протоколу IP. Тому з самого початку розробки протоколу IPv6 розроблялись і механізми його взаємодії з протоколом IPv4. Всі шляхи можна

розділити на дві групи: забезпечення взаємодії хостів з IPv6 між собою через існуючу мережу Internet, яка функціонує на основі протоколу IPv4, та методи, які забезпечують взаємодію між хостами з протоколами IPv6 та IPv4 та поступовий перехід від протоколу версії 4 на версію 6 (RFC 1933). Виділяють такі методи взаємодії:

- підтримка двох стеків протоколів (системи з подвійним стеком);
- організація тунелів для передачі трафіку IPv6 через мережі з протоколом IPv4;
- використання шлюзу прикладного рівня;
- трансляція адрес.

Подвійний стек – найбільш простий спосіб забезпечення взаємодії. В цьому випадку на кожному хості з IPv6, який має взаємодіяти з хостами, що функціонують на базі протоколу IPv4, встановлюється ще і стек протоколу IPv4 і йому присвоюється адреса IPv4 (рис. 4.25). Після цієї процедури даний хост може взаємодіяти як з IPv4 хостами, так і з IPv6 хостами.

Застосування подвійного стека має певні недоліки, зокрема необхідно встановити додаткове програмне забезпечення і виконати його конфігурування на кожному хості мережі, що призведе до збільшення навантаження на хости й підвищення вимог до наявних в них ресурсів. Крім того не тільки хости, а й усі маршутизатори та шлюзи мережі мають мати ресурси для обробки як пакетів IPv4, так і пакетів IPv6, що вимагає модернізації всього прикладного програмного забезпечення не тільки кінцевих станцій, а й комунікаційних вузлів.

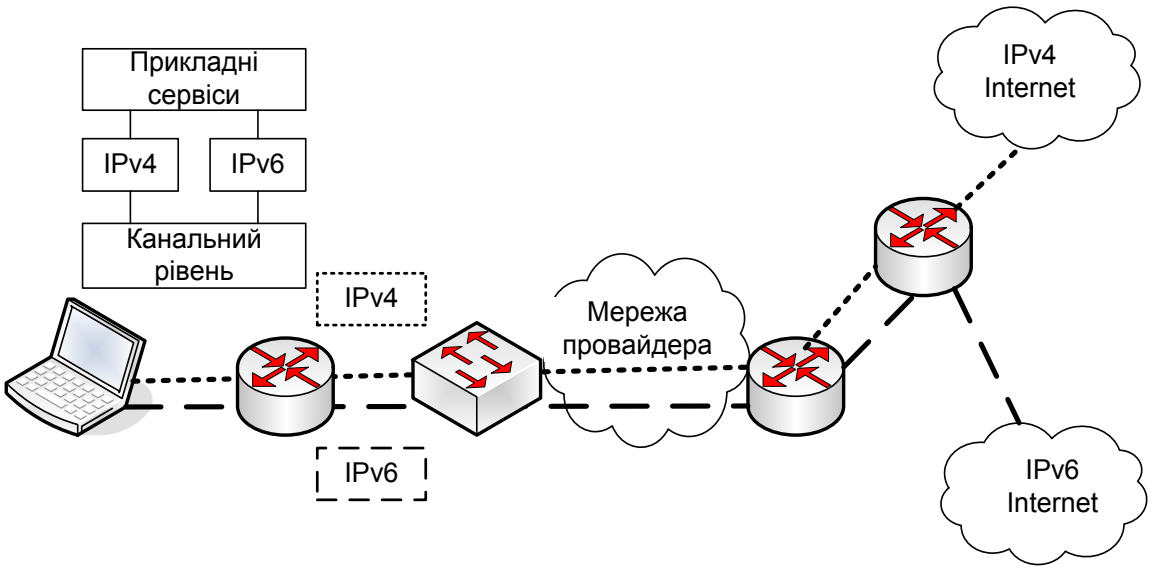


Рисунок 4.25 – Організація подвійного стека

Тунелювання. Даний метод призначений для створення IPv6 тунелів через існуючі мережі, які підтримують протокол IPv4, але не підтримують протокол IPv6. Такі тунелі створюються автоматично або вручну різними

способами та об'єднують окремі мережі з протоколом IPv6 між собою. Пакети IPv6 при вході в такий тунель інкапсулюються в пакет IPv4 і пересилаються через IPv4 мережу на інший кінець тунелю (рис. 4.26). Там вони деінкапсулюються і обробляються як звичайні IPv6 пакети (рис. 4.27). На основі таких тунелів функціонує експериментальна глобальна IPv6 мережа **6bone**. Дане рішення проблеми сумісності є частковим, оскільки не забезпечує взаємодії IPv4 хостів з IPv6 хостами. Але на даний момент саме цей механізм є найбільш поширеним. Існують такі типи тунелів:

- хост-хост (host-to-host). Два хости з подвійним стеком протоколів, які мають доступ тільки до інфраструктури IPv4, створюють тунель «з кінця в кінець»;
- маршрутизатор-хост (router-to-host) – тунель «з середини в кінець»;
- хост-маршрутизатор (host-to-router) – тунель «з початку в кінець»;
- маршрутизатор-маршрутизатор (router-to-router) – тунель з'єднує дві проміжні точки на маршруті.

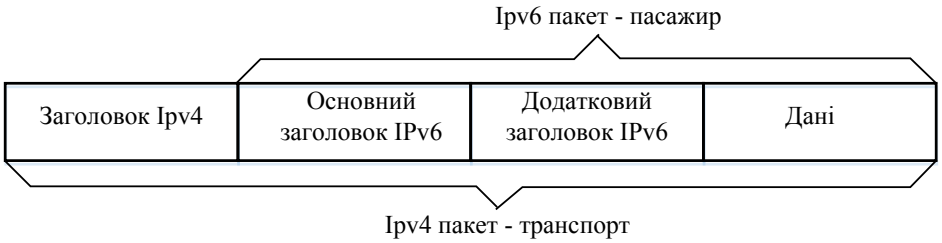


Рисунок 4.26 – Інкапсуляція IPv6 пакета в пакет IPv4

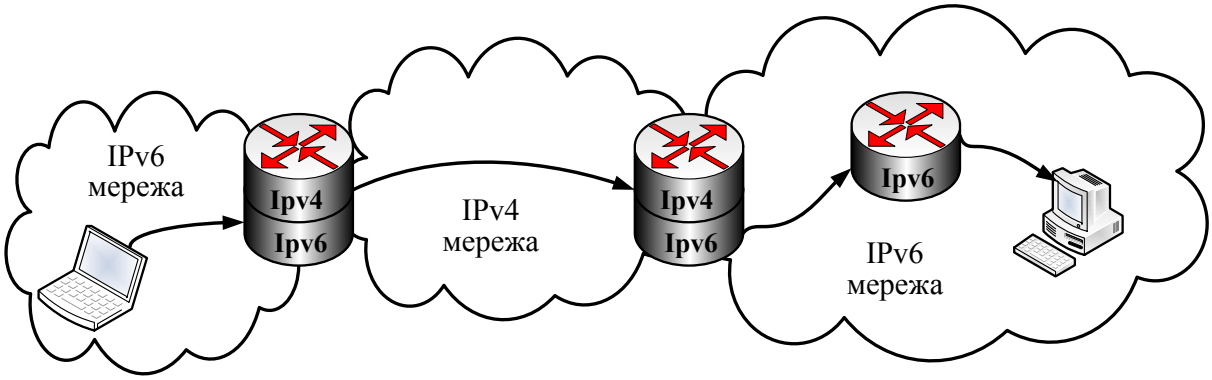


Рисунок 4.27 – Процедура тунелювання

В перших двох випадках кінцевою точкою тунелю є хост, тобто кінцева точка маршруту передавання пакета IPv6. В такому разі адреса кінця тунелю обчислюється автоматично, використовуючи для цього IPv4-сумісні адреси («IPv4-compatible IPv6 address»), які створюються за рахунок додавання до 32-бітових адрес 96 нульових бітів (рис. 4.28).

Якщо кінцева адреса не визначається за цільовою адресою отримувача, необхідно використовувати сконфігурований тунель, для якого параметри

тунелю задаються маршрутною таблицею у вузлі, на якому здійснюється інкапсуляція. Це необхідно у випадку, коли цільова адреса отримувача не є IPv4-сумісною, і тоді відправник має знати IPv4-адресу маршрутизатора з подвійним стеком, який і забезпечує доставку пакета IPv6. Зауважимо, що обидва кінця будь-якого тунелю мають мати IPv4-сумісні адреси.

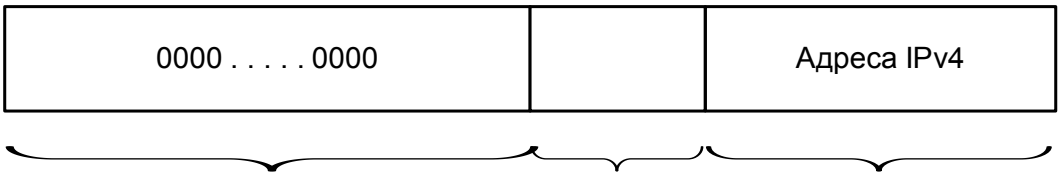


Рисунок 4.28 – IPv4-сумісна адреса

Шлюз прикладного рівня (**ALG** – Application Level Gateway) припускає, що для кожного мережного додатка, який функціонує на кінцевих станціях, створюється спеціальне прикладне програмне забезпечення, яке призначене для перетворення трафіку цього мережного додатка з трафіка IPv4 у трафік IPv6 і навпаки. Недоліки цього методу пов'язані з необхідністю створення відповідних ALG-шлюзів для кожного мережного додатка кожного хоста.

Безконтекстний IP/ICMP транслятор – даний механізм передбачає розташування на кордоні IPv6/IPv4 мереж спеціального агента (транслятора), який виконує трансляцію протоколів. При цьому IPv6 хостам призначаються спеціальні типи адрес: ::FFFF:0:0:0/96 – адреса для трансляції IPv4 в IP v6 та ::FFFF:0:0/96 – адреса для відображення IPv6 на IPv4. В обох типах адрес молодші 4 байти заміняються на відповідну IPv4 адресу.

Пакети IPv4, що надходять в таку систему, перенаправляються цьому агенту, де виконується перетворення формату IPv4 на формат IPv6, і пересилаються далі до станції одержувача. Пакети, що відправляються у відповідь, від хостів з протоколом IPv6 до хостів з IPv4, також мають пройти через IP/ICMP транслятор, але необов'язково через той же самий, тому що сам транслятор є безконтекстним, тобто не потребує додаткової інформації для трансляції. Пройшовши через транслятор, пакет протоколу IPv6 перетворюється на пакет IPv4 і передається відповідній станції за призначенням. Зручністю цього способу є прозорість для взаємодійних хостів і повна безконтекстність, що істотно полегшує її реалізацію та використання.

Перевагами такого підходу є простота використання, встановлення та налаштування транслятора. При цьому необхідно всім хостам IPv6 мережі надати IPv4-сумісні адреси, а прикладне програмне забезпечення залишається незмінним. Недоліком даного методу є те, що він може використовуватись тільки для зв'язку мереж з протоколом IPv6 через простір протоколу IPv4, а не навпаки. Тому даний підхід доцільно використовувати на етапі переходу мережі Internet на нову версію протоколу. В подальшому, при

збільшенні кількості хостів, маршрутизаторів, шлюзів, які будуть функціонувати на основі протоколу IPv6, необхідно буде забезпечити зв'язок мереж з протоколом IPv4 через простір IPv6.

4.12 Питання для самоперевірки

1. Визначте структуру IP-адреси.
2. Укажіть класи IP-адрес та охарактеризуйте їх.
3. Визначте особливості використання масок при IP-адресації.
4. Адреса хоста становить 10.156.211.47. Визначіть адреси мережі у випадках, коли довжини префікса становлять 26, 20 і 13.
5. Адреса хоста становить 10.201.87.145. Визначить адресу мережі, широкомовну адресу, першу і останню доступні адреси за умови, що довжина префікса становить 22.
6. Охарактеризуйте основні алгоритми маршрутизації потоків даних у комп'ютерних мережах.
7. Як визначається найкоротший шлях з використанням алгоритму Дейкстри?
8. Як визначається найкоротший шлях з використанням алгоритму Беллмана-Форда?
9. Визначте структуру заголовка і призначення полів пакета, що формується протоколом IPv4.
10. Які параметри мережі або каналів зв'язку використовуються для формування метрики?
11. Які основні характеристики мають протоколи маршрутизації?
12. За якими параметрами здійснюється класифікація протоколів маршрутизації?
13. Охарактеризуйте принципи роботи дистанційно-векторних протоколів.
14. Поясніть причини утворення маршрутних петель та опишіть основні механізми боротьби з цим явищем.
15. Охарактеризуйте принципи роботи протоколів з урахуванням стану каналу.
16. Дайте порівняльну характеристику дистанційно-векторних протоколів та протоколів з урахуванням стану каналу.
17. Які таблиці та для чого застосовують в своїй роботі дистанційно-векторні протоколи та протоколи з урахуванням стану каналу?
18. Назвіть основні компоненти сучасного маршрутизатора та охарактеризуйте їх призначення.
19. Які механізми обмеження доступу передбачені в операційній системі Cisco IOS?
20. В чому полягає ієрархічність CLI Cisco IOS?
21. Які основні задачі початкового конфігурування маршрутизатора?

22. Яким чином здійснюється конфігурування статичної та динамічної маршрутизацій?

23. Яку інформацію можна отримати з таблиці маршрутизації?

24. Охарактеризуйте основні відмінності протоколу IPv6 від протоколу IPv4.

25. Поясність особливості введення додаткових заголовків в структуру пакетів IPv6.

26. Наведіть структуру пакета IPv6, який передає TCP сегмент і містить заголовки маршрутизації, аутентифікації, фрагментації та Hop-by-Hop.

27. Охарактеризуйте структуру адресації IPv6 та принципи розподілу адресного простору.

28. Протокол ICMP. Призначення та особливості функціонування.

29. Принципи роботи протоколу ICMP.

30. Типи ICMP-повідомлень та їх призначення і функціонування.

31. Охарактеризуйте процедуру інкапсуляції ICMP-повідомлень та її необхідність.

32. Поясніть особливості протоколу ICMPv6.

33. Проаналізуйте основні способи взаємодії протоколів IPv6 та IPv4.

34. Охарактеризуйте особливості типів адрес, які введені в протоколі IPv6.

35. Охарактеризуйте методи призначення адрес, що реалізовані в IPv6.

36. Наведіть класифікацію персональних IPv6-адрес та охарактеризуйте їх призначення.

37. Поясніть особливості організації та функціонування систем з подвійним стеком.

38. Типи тунелів і їх особливості.

39. Призначення та необхідність процедура перетворення EUI-48 (MAC-48) в EUI-64.

5 ТРАНСПОРТНИЙ РІВЕНЬ

5.1 Базові принципи реалізації транспортного рівня

Як було зазначено в попередньому розділі, мережний рівень не контролює процес доставляння пакетів і не надає ніяких гарантій. Однак на шляху від відправника до одержувача пакети можуть бути спотворені або втрачені. Деякі програми мають власні засоби обробки помилок, пов'язаних з втратою та спотворенням пакетів, однак існують і такі, що вимагають наявності надійного з'єднання. **Саме транспортний рівень (ТР) забезпечує передавання даних між кінцевими системами з необхідним їм ступенем надійності**, що визначає його ключову роль в моделі OSI. Для надання відповідного сервісу верхнім рівням моделі OSI протоколи транспортного рівня мають реалізовувати певний набір функцій.

5.1.1 Базові функції протоколів транспортного рівня. До основних функцій протоколів транспортного рівня належать:

- підтримка різних типів комунікації;
- сегментація та відновлення;
- інкапсуляція та декапсуляція;
- впорядкування даних на боці одержувача;
- мультиплексування й демультиплексування;
- ідентифікація додатків;
- керування потоками даних;
- керування з'єднанням;
- виправлення помилок;
- забезпечення якості сервісу (Quality of Service - QoS);
- передавання даних;
- інтерфейс користувача;
- термінове доставляння.

Транспортний рівень підтримує два базових **типи комунікацій**: орієнтований на встановлення з'єднання та без встановлення з'єднання (дейтаграмний). Тип комунікації з встановленням з'єднання передбачає процедуру встановлення, підтримки та розриву з'єднання між об'єктами верхніх рівнів. Цей тип комунікації використовує багато протоколів верхніх рівнів і його досить часто називають надійним. Переваги цього типу сервісу є очевидними – це можливість виправлення помилок, керування потоками даних, впорядкування блоків даних ТР. Однак в деяких випадках більш привабливим є сервіс передавання даних без встановлення з'єднання. Це, в першу чергу, стосується програм, які вимагають мінімізації витрат, пов'язаних з передаванням даних. Прикладами таких споживачів транспортного сервісу є системи збирання даних, системи розповсюдження інфор-

мації, системи, що функціонують в режимі «запит-відповідь», системи реального часу тощо.

Передавання повідомлення через мережу передбачає його розбиття на фрагменти. Розміри фрагментів визначаються максимальним розміром блоку даних канального рівня (MTU). Ця процедура називається **сегментацією** (рис. 5.1). Саме тому блоки даних транспортного рівня називають **сегментами**. Насправді термін «сегмент» застосовують до одиниць передавання даних в межах надійного (гарантованого) транспортного сервісу, наприклад, TCP-сегменти. Блоки даних негарантованого транспортного сервісу називають **дейтаграмами**, наприклад, UDP-дейтаграма. На боці одержувача виникає обернена задача – **відновлення** повідомлення з отриманих фрагментів.

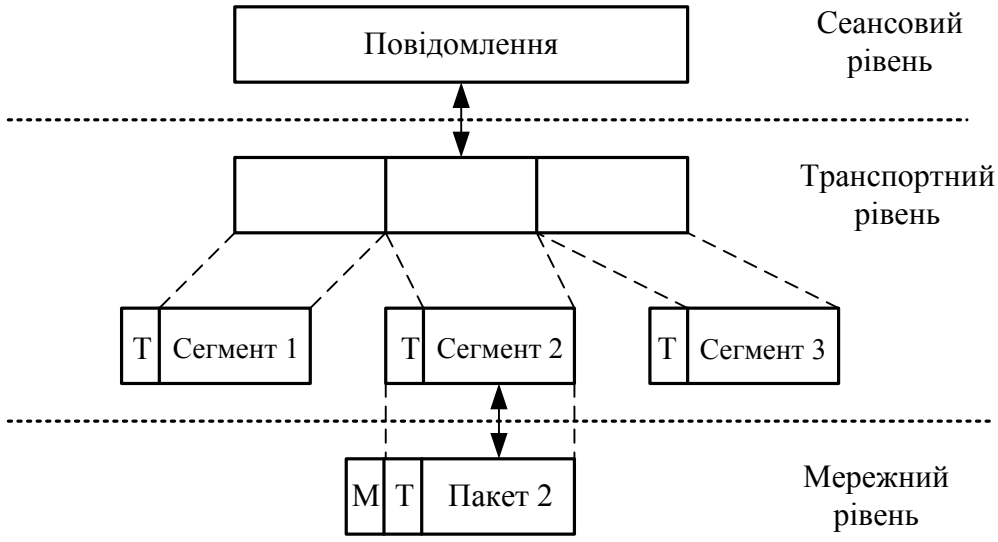


Рисунок 5.1 – Схема проходження повідомлення через транспортний рівень

Після здійснення сегментації до кожного фрагмента додаються певні службові дані, зокрема номер фрагмента, адресна інформація, контрольна сума тощо. Таким чином, фрагмент повідомлення перетворюється на сегмент, а цей процес називається **інкапсуляцією**. На рис. 5.1 інкапсуляція на транспортному рівні показана додатковим заголовком з літерою Т. Слід звернути увагу, що процедура інкапсуляції здійснюється й на інших рівнях, зокрема, на мережному і канальному. На рис. 5.1 службова інформація мережного рівня показана літерою М. Процедура відкидання службового заголовка на боці одержувача називається **декапсуляцією**.

Після отримання сегментів для відновлення повідомлення необхідно здійснити їх **впорядкування**, оскільки в загальному випадку вони можуть надійти до одержувача не в тому порядку, в якому були відправлені.

Задача протоколів транспортного рівня полягає в передаванні даних між прикладними процесами, що виконуються на комп’ютерах в мережі.

На кожному комп’ютері може виконуватись декілька процесів (рис. 5.2), що є одержувачами та відправниками потоків даних. Тому доставляння даних на мережний інтерфейс комп’ютера-одержувача – це ще не кінець шляху, оскільки дані потрібно спрямувати конкретному процесу-одержувачу, наприклад, процесу А комп’ютера І. Процедура розподілу протоколами транспортного рівня пакетів, що надходять від мережного рівня, між прикладними процесами називається **демультиплексуванням**.

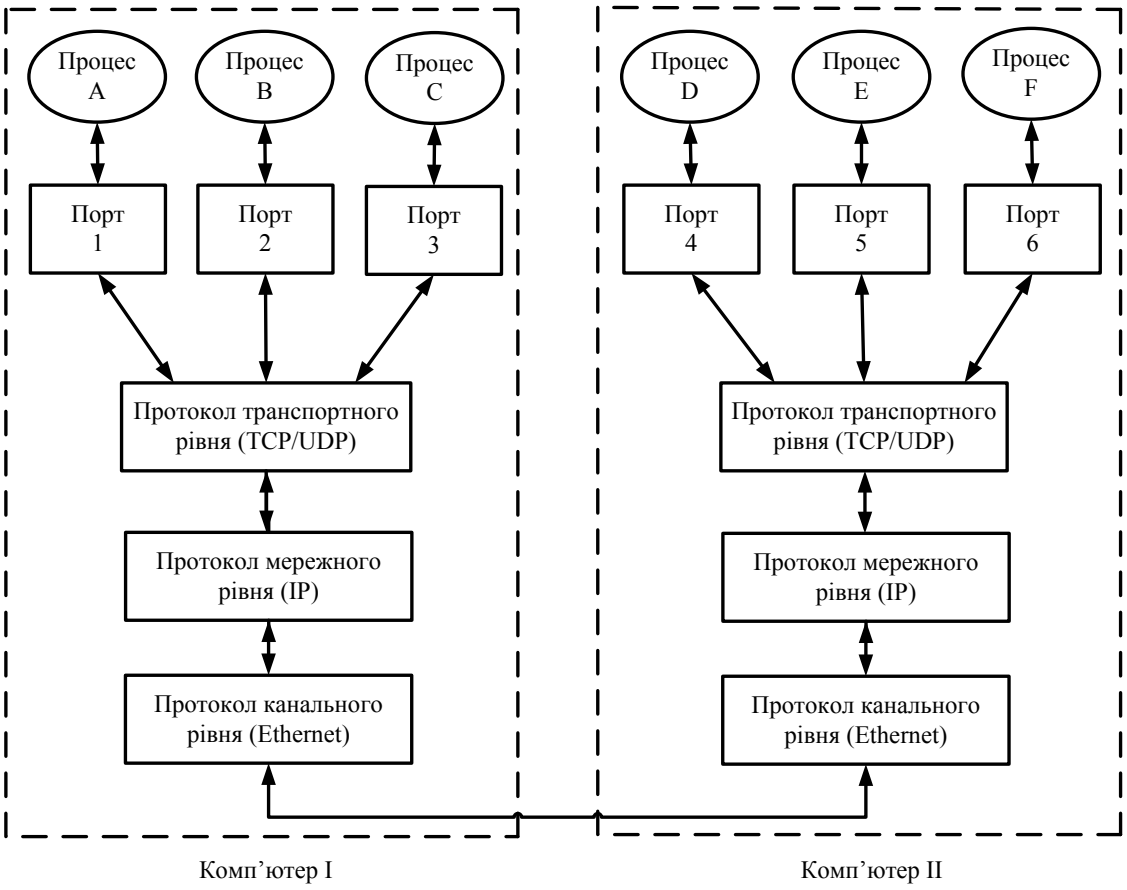


Рисунок 5.2 – Схема взаємодії прикладних процесів через мережу

Існує обернена задача: дані, що генеруються різними програмами (процеси D, E, F), які працюють на одному кінцевому вузлі (комп’ютер II), мають бути передані загальному для всіх них протокольному модулю мережного рівня для подальшого відправлення в мережу. Цю роботу, що має назву **мультиплексування**, також виконують протоколи транспортного рівня.

Процедура мультиплексування (демультиплексування) здійснюється за допомогою так званих **протокольних портів**. Протокольний порт унікальним чином ідентифікує прикладний процес на комп’ютері, тобто фактично виконує функцію **адресації**. Наприклад, на комп’ютері І порт 1 закріплено за процесом А.

Оскільки швидкість передавання даних в межах встановленого з'єднання може бути більшою за пропускну спроможність на боці одержувача, виникає потреба в механізмі **керування потоками даних**. Це дає змогу одержувачу впливати на інтенсивність відправлення даних відправником. Інше застосування цієї функції – реакція на перевантаження мережі, коли необхідно терміново примусити всіх відправників суттєво зменшити інтенсивність трафіку.

Сервіс **керування з'єднанням** передбачає процедури встановлення і розриву з'єднання. Процедура встановлення з'єднання може бути симетричною, коли обидва учасники можуть бути ініціаторами встановлення з'єднання, або асиметричною, яка використовується при утворенні симплексного логічного каналу. Існують також два варіанти розриву з'єднання – непередбачуваний та поступовий. У першому випадку можлива втрата даних, в другому – закриття з'єднання відбувається після завершення передавання.

Ще однією важливою задачею ТР є ідентифікація та **виправлення помилок**, що виникають під час передавання даних. Найпоширеніший метод знаходження спотворених сегментів базується на використанні контрольних послідовностей. Для відновлення спотворених та загублених сегментів найчастіше використовують повторне передавання.

Надання користувачам ТР певної **якості сервісу** передбачає забезпечення в межах з'єднання певних параметрів процесу передачі даних, зокрема допустимого рівня помилок і втрат, фіксованої максимальної затримки передавання даних, фіксованої мінімальної пропускної спроможності та підтримку різних рівнів пріоритету. Можливості мережного рівня забезпечити параметри QoS є обмеженими, тому виникає потреба їх реалізації на ТР. Наприклад, при передаванні файлів може виникнути потреба в максимізації пропускної спроможності логічного з'єднання, протоколи обслуговування транзакцій вимагають мінімізації затримки, поштові протоколи «зацікавлені» в такому параметрі QoS, як пріоритетність.

Функція **передавання даних** передбачає створення різних типів логічних каналів передавання даних з точки зору дуплексності, крім того в деяких випадках виникає потреба відокремлення каналів передавання даних та службової інформації.

Наявність стандартизованого **інтерфейсу** між ТР та сусідніми рівнями є також вкрай важливою функцією ТР. Зокрема це дає змогу керувати потоками даних між об'єктами сусідніх рівнів, впливати на механізм відправлення підтверджень, здійснювати прямий доступ до пам'яті тощо.

Сервіс **термінового доставлення** надає інструментарій передавання важливих даних з мінімальною затримкою. Після надходження таких даних об'єкт ТР миттєво інформує програму одержувача про таку подію і передає дані адресату, тоді як у звичайному режимі відбувається накопи-

чення даних у вхідному буфері. Цей сервіс може бути використаний для передавання символа «Break» під час термінального з'єднання, попереджень про небезпечні ситуації тощо.

Відповідно до моделі OSI ступінь надійності, що забезпечується транспортним рівнем, може варіюватись в широкому діапазоні, починаючи від найпростішого сервісу доставлення без процедури логічного з'єднання та підтвердження прийому до гарантованого доставлення всіх блоків даних з правильною послідовністю з можливістю керування потоками даних. Стандартом визначено **п'ять класів сервісу** TP0-TP4. Набір сервісів для кожного класу наведений в табл. 5.1.

Таблиця 5.1 – Класи сервісу транспортного рівня

Сервіс	TP0	TP1	TP2	TP3	TP4
Мережний сервіс із встановленням з'єднання	+	+	+	+	+
Мережний сервіс без встановлення з'єднання	-	-	-	-	+
Сегментація та відновлення	+	+	+	+	+
Відновлення з'єднання	-	+	-	+	-
Виправлення помилок	-	+	-	+	+
Мультиплексування та демультиплексування	-	-	+	+	+
Керування потоками даних	-	-	+	+	+
Повторна передача по тайм-ауту	-	-	-	-	+
Надійний транспортний сервіс	-	+	-	+	+

Як видно з таблиці, класи сервісу відрізняються якістю надаваних послуг, а саме: терміновістю, можливістю відновлення перерваного з'єднання, наявністю засобів мультиплексування декількох з'єднань між різними прикладними протоколами через загальний транспортний протокол, а головне – здатністю до виявлення і виправлення таких помилок передавання, як спотворення, втрата і дублювання пакетів.

Вибір класу сервісу транспортного рівня визначається, з одного боку, тим, якою мірою задача забезпечення надійності вирішується самими програмами і протоколами вищих (розташованих над транспортним) рівнів. З іншого боку, цей вибір залежить від того, наскільки надійною є система транспортування даних в мережі, яка забезпечується рівнями, що розташовані нижче транспортного: мережним, канальним і фізичним. У випадку, коли якість каналів передавання даних є дуже високою, а ймовірність виявлення всіх помилок протоколами нижніх рівнів досить велика, доцільно буде використати один із полегшених сервісів транспортного рівня, який не передбачає багаторазових перевірок, підтверджень та інших механізмів підвищення надійності. В той же час за умови ненадійності транспортних засобів нижніх рівнів доцільно застосовувати більш розвинені сервіси транспортного рівня, які використовують різноманітні засоби виявлення та усунення помилок, зокрема, попереднє встановлення логічного з'єднання,

контроль доставлення блоків даних із застосуванням циклічної нумерації, встановлення тайм-аутів доставлення тощо.

Всі протоколи, починаючи з транспортного рівня і вище, реалізуються програмними засобами кінцевих вузлів мережі – компонентами їх мережних операційних систем. Прикладами транспортних протоколів є протоколи **TCP і UDP стека TCP/IP** та **протокол SPX стека IPX/SPX**. Оскільки протокольний стек TCP/IP на сьогоднішній день є стандартом де факто, є сенс детально зупинитись на реалізації TP саме в TCP/IP.

5.1.2 Протокольні порти стека TCP/IP. До транспортного рівня стека TCP/IP відносять:

- протокол керування передаванням (Transmission Control Protocol, TCP), описаний в RFC793;
- протокол користувацьких дейтаграм (User Datagram Protocol, UDP), описаний в RFC768.

Протоколи TCP і UDP ведуть для кожної програми дві системні черги: черга даних, що надходять в програму з мережі, і черга даних, які дана програма відправляє в мережу. Такі системні черги називаються відповідно **TCP-** та **UDP-портами**, причому вхідна і вихідна черги однієї програми розглядаються як один порт. Для ідентифікації портів їм присвоюють номери.

Якщо процеси являють собою такі популярні системні служби, як DNS, SSH, FTP, telnet, HTTP, TFTP тощо, то за ними закріпляються **стандартні призначені** номери, які ще називаються **загальновідомими** (well-known) номерами портів. Ці номери централізовано призначені тією ж самою організацією (IANA), що контролює адресний простір в мережі Інтернет, і опубліковані у відповідних документах (RFC 1700, RFC 3232). Так, номер 80 закріплений за серверною частиною служби доступу до гіпертекстових документів HTTP, а 23 – за серверною частиною служби віддаленого керування telnet. Призначені номери в діапазоні від **0** до **1023** є унікальними в межах Інтернет.

Номери з діапазону **1024–49151** також призначаються і контролюються IANA, однак це відбувається за ініціативи розробників програмного забезпечення. Ці порти називаються **зареєстрованими** і призначені для тих програм, що відносно недавно розроблені або не стали глобально поширеними, наприклад, 1027, 1029 tcp-порти закріплені за популярною програмою миттєвих повідомлень ICQ, 8080, 8008 – альтернативні порти HTTP-сервісу.

Інші номери (від **49152 до 65535)** називаються динамічними (приватними) і генеруються операційною системою у відповідь на надходження запиту від програми. На кожному комп'ютері операційна система веде список зайнятих і вільних номерів портів. Під час надходження запиту від програми, що виконується на даному комп'ютері, операційна система виділяє її перший вільний номер. У подальшому всі мережні програми «спілкуються» з цією програмою через виділений їй динамічний номер порту.

Після того, як програма завершить роботу, її номер повертається до списку вільних і може бути призначений іншій програмі. Динамічні номери є унікальними в межах одного комп'ютера, але при цьому звичайною є ситуація збігання номерів портів програм, що виконуються на різних комп'ютерах. Як правило, клієнтські частини відомих програм (HTTP, SMTP, DNS, FTP тощо) отримують динамічні номери портів від операційної системи.

Прикладний процес унікальним чином визначається в межах мережі парою: IP-адреса хоста та номер порту. Така комбінація називається **сокетом** (socket). Сокет, визначений IP-адресою і номером UDP-порту, називається **UDP-сокетом**, а IP-адресою і номером TCP-порту – **TCP-сокетом**.

5.2 Протокол UDP

Протокол UDP виконує мінімум дій, необхідних для протоколу транспортного рівня. Фактично функції UDP зводяться до операцій мультиплексування та демультиплексування, а також нескладної перевірки наявності помилок в даних. Таким чином, під час використання UDP програма майже напряму взаємодіє з протоколом мережного рівня IP. UDP отримує повідомлення з рівня додатків, додає до них поля номерів портів відправника і одержувача, а також два інших спеціальних поля і через нижні рівні передає створену дейтаграму одержувачу.

Якщо останній успішно отримує дейтаграму, протокол UDP за допомогою поля номера порту одержувача направляє дані потрібному процесу. Слід звернути увагу, що протокол UDP не передбачає процедури встановлення логічного з'єднання перед початком передавання дейтаграм. Тому UDP відносять до протоколів, що здійснюють передавання даних без встановлення з'єднання. Структуру UDP-дейтаграми наведено на рис. 5.3.

Рисунок 5.3 – Структура UDP-дейтаграми

Дані програми розташовуються в полі даних дейтаграми, наприклад, в полі даних може бути DNS-повідомлення (запит або відповідь) або семпл потокового аудіо. Заголовок UDP-дейтаграми складається з чотирьох двобайтових полів. **Номери портів відправника і одержувача** дозволяють

хосту призначення направити дані дейтаграми необхідному сокету (іншими словами, здійснювати процедуру демультиплексування). **Контрольна сума** призначена для перевірки помилок в отриманих даних. **Довжина дейтаграми** дозволяє визначити межу даних.

Не дивлячись на відсутність гарантій доставлення даних мінімізація накладних витрат надає протоколу UDP низку переваг порівняно з надійним TCP.

Відсутність процедури встановлення логічного з'єднання дозволяє уникнути додаткових витрат часу. Це є головною причиною використання UDP прикладним протоколом DNS.

Відсутність інформації про стан з'єднання. Протокол TCP для реалізації гарантованого доставлення потребує використання буферної пам'яті, в якій зберігаються непідтверджені сегменти та інша службова інформація. Таким чином один і той же комп'ютер може одночасно підтримувати значно більше UDP-сеансів ніж TCP.

Невеликий розмір заголовка. UDP-заголовок складає 8 байтів, а TCP – 20. Це дозволяє ефективніше використовувати смугу пропускання.

Покращений механізм керування передаванням даних. Передавання даних в TCP-сокет може призупинитись внаслідок багатьох різних причин: наявності ознак перевантаження каналу, зменшення вікна передавання, відсутність підтвердження на переданий сегмент тощо. Це призводить до вкрай негативних наслідків при передаванні трафіку реального часу. Саме з цієї причини програми, що працюють в реальному часі, надають перевагу протоколу UDP.

5.3 Протокол TCP

Протокол TCP – це основний транспортний протокол зі стека протоколів TCP/IP. Він забезпечує надійне передавання потоку даних, використовуючи при цьому ненадійний сервіс транспортування пакетів, що надається протоколом IP. В мережах IP протокол TCP використовується для обробки запитів на вхід в мережу, доступу до файлових ресурсів в локальних і глобальних мережах, реплікації інформації між контролерами доменів Active Directory, передавання списків ресурсів тощо. На протокол TCP покладається задача керування потоками даних і забезпечення механізмів виходу з перевантаження. Він відповідає за узгодження швидкості передавання даних із технічними можливостями робочої станції-одержувача та проміжних пристроїв в мережі.

5.3.1 Структура TCP-заголовка. Даний протокол використовує тільки один тип протокольного блоку даних (PDU – Protocol Data Unit), який називається **TCP-сегментом**. Сегмент складається з заголовка і поля даних (корисного навантаження). На рис. 5.4 показаний формат TCP-сегмента.

Мінімальна довжина TCP-заголовка складає **20 байтів**. Такий великий розмір викликаний тим, що один і той самий заголовок використовується

протоколом для різних цілей. Для визначення функцій більшості полів призначені **контрольні біти**, формат і значення яких наведено в табл. 5.2.

Рисунок 5.4 – Формат TCP-сегмента

Таблиця 5.2 – Формат і значення контрольних бітів

№ біта	Скорочення	Призначення
1	URG	Задіяне поле «вказівник терміновості»
2	ACK	Задіяне поле «номер підтвердження»
3	PSH	Ввімкнена функція «проштовхування»
4	RST	Перевантаження даного з’єднання
5	SYN	Синхронізація номерів у черзі
6	FIN	Даних для передавання немає

Номер порту відправника і **номер порту одержувача,** як і в протоколі UDP, ідентифікують процеси, що є відправниками і одержувачами даних.

Номер в послідовності (Sequence Number) визначає номер першого байта у черзі (послідовності) байтів в поточному сегменті. Винятком є випадки, коли встановлено біт синхронізації SYN. Тоді дане поле позначає **початковий номер в послідовності** (Initial Sequnce Number, ISN), і перший байт даних має номер у черзі ISN+1. Використовується для нумерації сегментів для забезпечення гарантій доставлення.

Номер підтвердження (Acknowledgment Number) містить наступний номер байта, який очікує відправник у відповідь на надісланий сегмент. Іншими словами, на надісланий сегмент з даними відправник очікує сегмент з підтвердженням його успішного прийому. Підтвердження передається у вигляді запиту наступного сегмента. При цьому має бути встановлений **контрольний біт підтвердження ACK**. Підтвердження (або ACK) відправляються постійно після встановлення з’єднання.

Поля номера в послідовності і номера підтвердження вирівнюються за числом байтів в полі даних, а не за довжиною всього сегмента. Наприклад, якщо в полі номера в послідовності вказано значення 2000 і сегмент містить в полі даних 700 байтів, то сегмент підтвердження в полі номера підтвердження передасть 2701.

Зсув даних (Data Offset) визначає кількість 32-бітових слів в TCP-заголовку. Таким чином вказується початок поля даних. TCP-заголовок завжди закінчується на 32-бітовій межі, навіть якщо в ньому наявні необов'язкові поля.

Поле **резерв** (Reserved) заповнюється нулями і призначене для майбутнього розширення протоколу.

Вікно (Window) вказує кількість байтів, які може передати одержувач сегмента, не очікуючи підтвердження. Це поле використовується для керування потоками даних.

Контрольна сума (Checksum) розраховується на основі вмісту сегмента з урахуванням псевдозаголовка. 96-бітовий псевдозаголовок містить частину IP-заголовка, зокрема IP-адреси відправника і одержувача, протокол та довжину сегмента. Завдяки додаванню псевдозаголовка протокол TCP захищає себе від спотвореного доставлення протоколом IP.

Вказівник терміновості (Urgent Pointer) повідомляє поточне значення вказівника терміновості. Ця величина визначає зсув відносно номера в черзі даного сегмента. Даний вказівник повідомляє номер байта, що є наступним за терміновими даними, тобто, починаючи з цього байта, дані мають звичайний рівень терміновості. Поле використовується спільно з **контрольним бітом URG**.

Поле **опції** (Options) має змінну довжину і може взагалі бути відсутнім. Опції (необов'язкові параметри) розташовані в кінці TCP-заголовка, а їх довжина є кратною **8 бітам**. Протокол TCP має бути готовий обробляти всі види опцій. Опції використовуються для вирішення допоміжних задач, наприклад, для вибору максимального розміру сегмента.

Вирівнювання (Padding) може мати змінний розмір і являє собою фіктивне поле, що використовується для доповнення розміру заголовка до цілого числа 32-бітових слів.

Контрольні біти **PSH** (Push) і **URG** (Urgent) реалізують дві служби протоколу TCP: просування (проштовхування) потоку даних і сигналізацію про термінові дані.

Зазвичай протокол TCP здійснює передавання даних, коли кількість байтів, що мають бути передані, дорівнює максимальному значенню довжини поля даних сегмента. Однак прикладний процес може вимагати, щоб протокол передав всі дані, що залишились, і помітив їх позначкою PSH. На приймальному боці модуль протоколу TCP доставить ці дані програмі-одержувачу відразу (тобто дані не будуть очікувати в буфері). Механізм проштовхування, як правило, застосовується у випадку досягнення логічного кінця даних, наприклад, при передаванні останнього сегмента файлу.

Контрольний біт URG дозволяє інформувати програми на приймальному боці про те, що вхідний потік містить важливі дані, а поле вказівника терміновості визначає межу між важливими і звичайними даними. Як і у попередньому випадку важливі дані відразу після надходження передаються програмі-одержувачу.

5.3.2 Основні фази роботи TCP-протоколу. В роботі TCP-протоколу можна виділити такі три основні фази: **відкриття TCP-з'єднання**, **передавання даних** і **завершення з'єднання**.

Спрощено процес відкриття з'єднання передбачає виконання такої послідовності дій:

1. Програма-ініціатор, що працює на хості відправника і потребує передавання даних за допомогою TCP-протоколу, відправляє запит до протоколу TCP на відкриття порту для передавання;
2. Після відкриття порту протокол TCP, що працює на хості відправника, відправляє запит програмі-відповідачу, що знаходиться на хості одержувача і з якою необхідно встановити з'єднання;
3. Протокол TCP на хості одержувача відкриває порт для прийому даних і відправляє квитанцію, яка підтверджує прийом запиту;
4. TCP-протокол на хості одержувача відкриває порт для передавання і також передає запит програмі-ініціатору;
5. Програма-ініціатор відкриває порт для прийому і повертає квитанцію.

TCP-з'єднання фактично є дуплексним, тому передбачає встановлення двох з'єднань – від відправника до одержувача і в зворотному напрямку.

Під час передавання даних через з'єднання кожен байт інформації нумерується. Нумерація ведеться і в черзі відправлення, і в черзі прийому. Перш ніж почнеться передавання модуля(ів) протоколу TCP відправник і одержувач мають синхронізувати один з одним початкові номери байтів в чергах. Синхронізація виконується шляхом обміну сегментами, які використовуються при встановленні з'єднання.

Сегменти синхронізації містять встановлений біт синхронізації SYN і початковий номер у черзі відправлення в полі номера в послідовності. Синхронізація вимагає, щоб кожна сторона надіслала свій власний початковий номер в черзі і отримала підтвердження про отримання даного номера. Нумеруються і самі сегменти: номером сегмента вважається номер першого байта в полі корисного навантаження цього сегмента. Приклад встановлення TCP-з'єднання та порядок синхронізації показано на рис. 5.5, а.

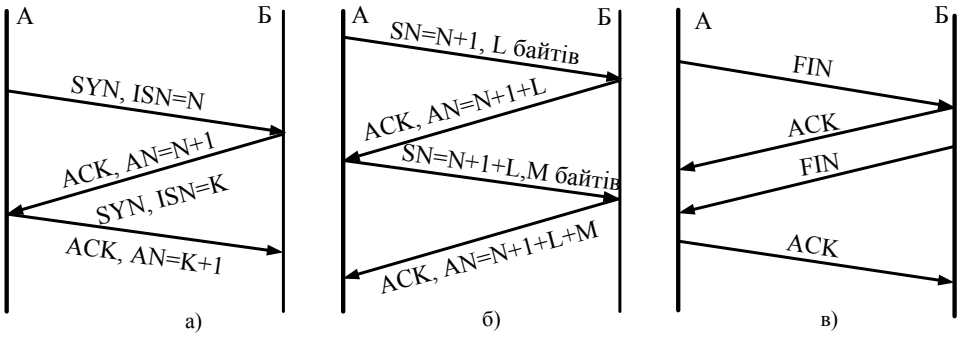


Рисунок 5.5 – Робота протоколу TCP
а) встановлення з'єднання, б) передавання даних, в) завершення з'єднання

Процес синхронізації передбачає виконання таких дій:

1. Станція відправник (А) відправляє сегмент з відміткою SYN і своїм номером в черзі ISN=N станції одержувачу (Б);

2. Одержувач передає відправнику підтвердження, в якому вказує, що очікує на сегмент з номером AN=N+1;

3. Одержувач також відправляє сегмент з відміткою SYN і власним номером в черзі ISN=K;

4. Відправник передає підтвердження, в якому вказує, що очікує на сегмент з номером AN=K+1.

Кроки 2 і 3 можна об'єднати, тому такий обмін називається відкриттям з'єднання з підтвердженням трьох повідомлень (three-way handshake). Слід звернути увагу, що у випадку взаємодії клієнта з сервером ініціатором TCP-з'єднання завжди є клієнт.

Фаза передавання даних (див. рис. 5.5, б) передбачає використання механізмів нумерації відправлених сегментів і підтверджень про їх отримання. Оскільки TCP-з'єднання є дуплексним, то використовуються дві нумерації сегментів – від А до Б та від Б до А. Як було показано вище, початкові номери для нумерації були визначені на етапі встановлення TCP-з'єднання. Так, перший сегмент даних, що буде спрямовано від А до Б, матиме номер SN=N+1. Відповідно, перший сегмент, спрямований в зворотному порядку, буде під номером SN=K+1.

Номером сегмента вважається номер першого байта в сегменті. Так, наприклад, якщо номер першого сегмента N+1 і він містить L байтів даних, то номер другого сегмента відповідно буде N+1+L.

Для підтвердження отриманих даних TCP-протокол використовує так звані очікувальні підтвердження – отримавши сегмент станція відправляє запит на наступний. На рис. 5.5, б показано процес передавання та підтвердження даних в напрямку від А до Б. На практиці передавання та підтвердження даних здійснюються одночасно в обох напрямках. Приклад встановлення з'єднання та дуплексного передавання даних наведено на рис. 5.6.

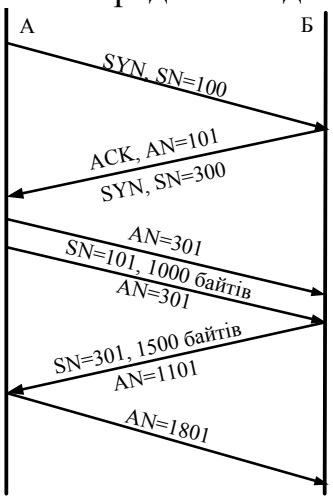


Рисунок 5.6 – Приклад встановлення та передавання даних в межах TCP-з'єднання

Процедура завершення TCP-з'єднання є аналогічною до встановлення (рис. 5.5, в). Відмінність полягає в тому, що ініціатором завершення може бути як клієнт, так і сервер. Крім того, біти підтвердження отримання запиту на завершення з'єднання та власний запит на завершення з'єднання не обов'язково передаються в одному сегменті.

5.3.3 Керування процесом передавання даних в межах TCP-з'єднання. Для керування потоком даних протокол TCP використовує механізм **ковзного вікна зі змінними розмірами**. Розглянемо схему роботи даного механізму на прикладі двох станцій: А – відправника даних і Б – одержувача.

Станція Б виділяє буферний простір для прийому W байтів, таким чином станція Б може прийняти W байтів, а станція А може відправити ті самі W байтів без необхідності очікування підтвердження про їх отримання. Як вище було зазначено, підтвердження містить номер в послідовності наступного сегмента, який очікується. Воно сповіщає станцію А про те, що станція Б отримала всі попередні сегменти і непрямо інформує про готовність отримання наступних сегментів, що містять чергові W байтів.

Така схема роботи підходить для підтвердження отримання множини сегментів і продемонстрована на рис. 5.7, а для випадку W=3000 байтів. Станція Б отримує сегменти з номерами 101, 1101 і 2101, але утримується від відправлення підтвердження на перші два сегменти до отримання сегмента з номером 2101. Відправляючи підтвердження, в якому вказується на очікування сегмента з номером 3101, станція Б одночасно підтверджує отримання сегментів з номерами 101, 1101 і 2101.

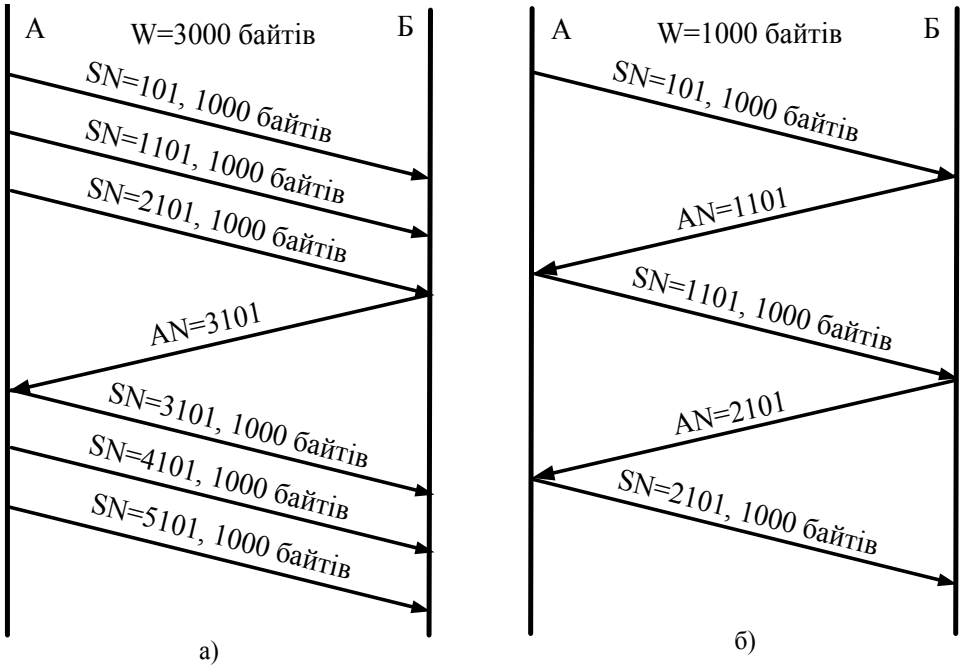


Рисунок 5.7 – Приклад передавання з застосуванням механізму ковзного вікна при а) W=3000 байтів та б) W=1000 байтів

Таким чином, можна сказати, що станція А веде список номерів в послідовності сегментів, які їй дозволено відправляти, а станція Б підтримує список номерів в послідовності сегментів, які вона готова прийняти. Ці списки називаються **вікнами сегментів**, а таку схему передавання даних називають керуванням потоком з використанням **ковзного вікна**.

Порівнявши рис. 5.7, а та 5.7, б неважко побачити, що розмір вікна безпосередньо впливає на пропускну спроможність TCP-з'єднання. Так, в наведеному прикладі збільшення вікна в три рази дозволило за однаковий період часу передати вдвічі більше даних. В загальному випадку максимальна швидкість передавання даних через TCP-з'єднання залежить від розміру вікна передавання та затримки за умови, якщо вона не перебільшує пропускної спроможності каналу.

Нехай розмір вікна передавання становить W байтів, затримка передавання – D с, швидкість передавання – R біт/с. Припустимо, що відправник починає передавати послідовність байтів через встановлене з'єднання. Для того, щоб перший байт досяг одержувача, необхідно D секунд. Такий самий час (D секунд) необхідний для отримання підтвердження. Протягом цього часу відправник може передати $2RD$ бітів, або $RD/4$ байтів. Однак відправник обмежений розміром вікна у W байтів і не може зсувати вікно, доки не отримає підтвердження. Таким чином максимальна швидкість передавання даних в межах TCP-з'єднання визначатиметься виразом $R = 4W/D$. В ідеальному випадку це значення збігається з пропускною спроможністю TCP-з'єднання.

Природно, що розмір вікна визначає одержувач. Крім того від одержувача не вимагається негайного підтвердження сегментів, що надійшли. Він може очікувати деякий час, а потім сформувати підтвердження одразу на декілька сегментів. Одержувач має проводити деяку політику, яка б регулювала кількість даних, що їх він дозволяє передавати відправнику. Можна виділити дві політики одержувача: консервативну та оптимістичну.

Консервативна схема визначення розміру вікна базується на тому, що ліміт виділяється відповідно до наявного доступного буферного простору. Консервативна схема може суттєво обмежити пропускну спроможність TCP-з'єднання в ситуації, коли затримка D є значною.

Одержувач може ефективніше використовувати пропускну спроможність каналу за допомогою **оптимістичної** політики, встановлюючи розмір вікна більший за вільний буферний простір. В даному випадку одержувач припускає, що до моменту надходження даних обсяг вільного буферного простору збільшиться. Така схема здатна підвищити пропускну спроможність, але якщо відправник працює швидше, ніж одержувач, то деякі сегменти будуть відкидатися, оскільки буфер буде зайнятий. Це, в свою чергу, призведе до повторного передавання відкинутих сегментів і, навпаки, зменшить ефективність використання пропускної спроможності каналу.

Слід звернути увагу, що, окрім розміру буфера на боці одержувача, на розмір вікна впливає факт наявності перевантажень в мережі. Особливості реакції TCP на перевантаження будуть розглянуті нижче.

5.3.4. Політики відправлення та прийому сегментів. Якщо відсутні дані, помічені відміткою PSH, протокол TCP **на боці відправника** може самостійно вирішувати, коли слід здійснювати передавання. Під час передавання даних TCP-модулю від програми користувача вони записуються в буфер передавання. Протокол TCP може створювати сегмент для кожної групи даних, що надходять від програми, або він може очікувати накопичення певної кількості даних і тільки після цього формувати і відправляти сегмент. Очевидно, що в першому випадку час знаходження даних у вихідному буфері буде мінімальним, другий варіант забезпечить формування сегментів максимального розміру і дозволить збільшити корисну пропускну спроможність.

Вибір на користь тієї чи іншої політики відправлення повністю залежить від вимог до швидкодії. Якщо передавання сегментів відбуваються рідко, але при цьому передаються великі обсяги даних, то сегменти можна формувати відразу після надходження даних – накладні витрати тут будуть невеликими. З іншого боку – навіть якщо обсяг даних, що передаються, невеликий, іноді є сенс відправляти їх відразу після отримання – при цьому накладні витрати будуть великими, але така система забезпечує максимальну швидкість передавання.

Якщо відсутні дані, помічені відміткою PSH, протокол TCP **на боці одержувача** також може самостійно вирішувати, коли слід доставляти дані програмі користувача. Він може доставляти дані після отримання кожного сегмента, або здійснювати буферизацію даних. Так само, як і у випадку з відправленням, вибір політики доставки залежить від вимог до швидкодії. Якщо дані надходять рідко і мають великий обсяг, то є сенс передавати їх «нагору» відразу. З іншого боку, якщо дані надходять рідко маленькими частинами, то їх негайне передавання програмі-одержувачу призведе до неефективного використання ресурсів програми і протоколу TCP.

Якщо сегменти надходять у порядку їх відправлення, протокол TCP поміщає їхні дані в буфер прийому для доставки програмі користувача. Але досить вірогідною є ситуація, за якої порядок надходження відрізнятиметься від порядку відправлення. Причиною цього може бути проходження сегментів різними маршрутами. В такому випадку протокол TCP на боці одержувача може або приймати тільки ті сегменти, які надходять в порядку відправлення (інші сегменти просто відкидаються), або приймати всі сегменти, номери яких зафіксовані у вікні прийому (незалежно від порядку їх надходження).

Протокол TCP підтримує чергу сегментів, які були відправлені, але ще не були підтверджені. Специфікація протоколу визначає, що він буде повторно передавати сегмент, якщо на нього не було отримано підтверджен-

ня протягом деякого проміжку часу. Реалізація протоколу TCP може підтримувати три **режими повторного передавання**.

- **Тільки перший** (рис. 5.8, а). Підтримується один таймер повторного передавання для всієї черги. Якщо було отримано підтвердження, з черги повторного передавання видаляється перший сегмент і таймер скидається. Якщо таймер спрацьовує (підтвердження не надійшло протягом вказаного проміжку часу), сегмент з початку черги передається повторно, а таймер скидається.
- **Пакетний** (рис. 5.8, б). Підтримується один таймер повторного передавання для всієї черги. Якщо було отримано підтвердження, з черги повторного передавання видаляються всі підтверджені сегменти і таймер скидається. Якщо таймер спрацьовує (за вказаний проміжок часу підтвердження не було отримано), всі сегменти з черги передаються повторно, а таймер скидається.
- **Індивідуальний** (рис. 5.8, в). Для кожного сегмента в черзі підтримується окремий таймер. Якщо на сегмент було отримано підтвердження, він видаляється з черги повторного передавання, а його таймер обнуляється. Якщо якийсь із таймерів спрацьовує, то відповідний сегмент передається повторно, а його таймер скидається.

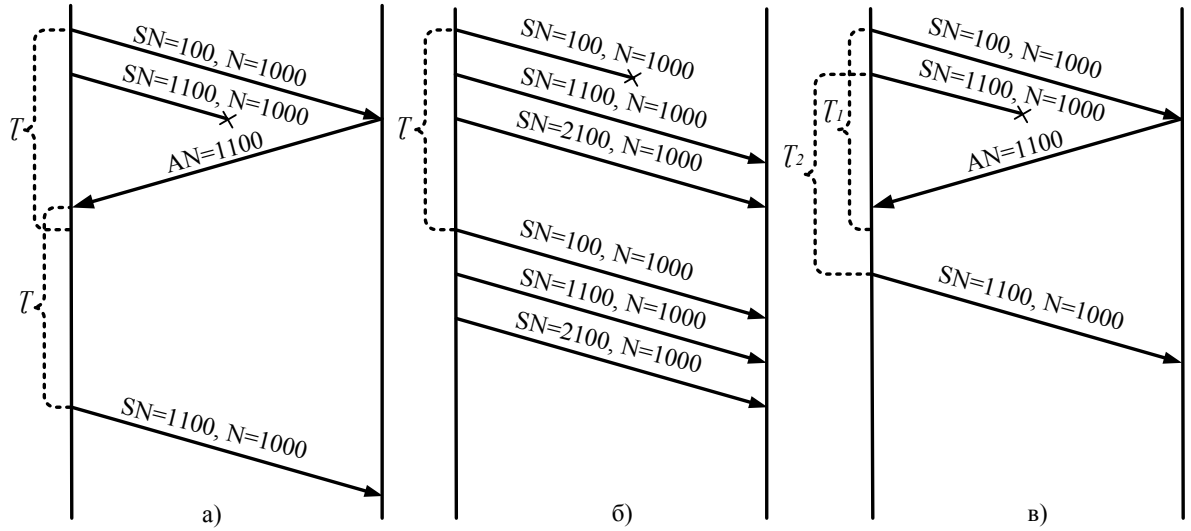


Рисунок 5.8 – Режим повторного передавання за такими схемами:
а) «тільки перший», б) пакетний, в) індивідуальний

Ефективність тієї чи іншої політики повторного передавання залежить від політики прийому сегментів на боці одержувача. Якщо одержувач використовує політику прийому, відповідно до якої приймаються лише сегменти, що надходять у порядку відправлення, то він буде відкидати сегменти, отримані після втраченого сегмента. Така схема роботи добре підходить для пакетної політики повторного передавання. Якщо одержувач

приймає всі сегменти незалежно від порядку їх прибуття, то оптимальними є режими «Тільки перший» або «Індивідуальний».

Після надходження сегмента протокол TCP на боці одержувача має два варіанти дій для того, щоб відправити підтвердження.

- **Негайно.** Відразу після отримання даних передається порожній (без даних) сегмент, який містить відповідний номер підтвердження.
- **З накопиченням.** Позначка ACK встановлюється в сегменті з даними, які одержувач надсилає відправнику у відповідь. Щоб запобігти довгим затримкам, встановлюється таймер вікна. Якщо таймер збігає до моменту відправлення чергового сегмента з підтвердженням, то відсилається порожній сегмент, що містить відповідний номер підтвердження.

Ефективність передавання даних в межах TCP-з'єднання також суттєво залежить від затримки повторного передавання, що визначається **таймером повторного передавання**.

Протокол TCP не формує явних негативних підтверджень, тобто підтверджень, що явно вказують на порушення, які сталися. Замість цього протокол покладається виключно на позитивні підтвердження і на повторне передавання, яке має відбуватися, якщо підтвердження не надходить протягом визначеного проміжку часу. Дві події приводять до повторного передавання сегмента: сегмент був спотворений під час передавання та сегмент згублено.

Для прийняття рішення про повторне передавання використовується спеціальний таймер, що запускається після відправлення сегмента. Якщо час таймера збігає до моменту надходження підтвердження про доставку сегмента, відправник має здійснити повторне передавання. Ефективність роботи TCP суттєво залежить від вибору значення таймера повторного передавання. Базовим елементом для розрахунку затримки повторного передавання є значення так званого **часу подвійного обернення** (Round Trip Time). RTT – це подвоєний час проходження сегмента від відправника до одержувача. Затримка повторного передавання має бути дещо більшою за RTT.

Існує два способи визначення затримки повторного передавання.

- **Фіксований таймер**. Використовується фіксоване значення, яке визначається за статистичними даними, що характерні для «нормальної» поведінки розподіленої мережі. Інакше кажучи, визначається середнє значення RTT, і таймер встановлюється дещо більшим. Оскільки така політика базується на стійкому режимі роботи мережі, то вона не здатна адекватно та гнучко реагувати на різкі зміни умов в мережі.
- **Адаптивний таймер**. Значення таймера постійно оновлюється. Формула для його розрахунку базується на актуальному значенні RTT, його відхиленні від середнього значення та деяких інших параметрах.

5.3.5 Боротьба з перевантаженнями. Контроль за перевантаженнями в мережах IP досить важко реалізувати з огляду на цілу низку причин, зокрема:

- протокол IP не орієнтований на встановлення з'єднання. Він не забезпечує виявлення перевантаження і внаслідок цього не може використовуватись для контролю за перевантаженнями;
- протокол TCP здійснює контроль передавання даних в межах з'єднання і може лише за непрямими ознаками виявити перевантаження в мережі. Однак, оскільки затримки в розподілених мережах постійно змінюються, інформація, отримана на основі непрямих ознак (наприклад, розмір вікна), не завжди є достовірною;
- не існує розподіленого алгоритму, який би забезпечив взаємодію TCP-модулів різних хостів, що передають дані через спільний фізичний канал. Тобто, протоколи на різних комп'ютерах не можуть взаємодіяти один з одним для підтримки певного рівня загального потоку, внаслідок чого кожен з них робить спробу «захопити» всю смугу пропускання.

Протокол TCP може впливати на ступінь завантаження мережі, керуючи потоком даних за допомогою ковзного вікна, застосовуючи різні методи передавання та прийому даних і відправлення підтверджень, слідкуючи за рівнем помилок і використовуючи різні методи повторного передавання.

Механізми боротьби з перевантаженнями, що застосовуються в TCP, полягають в обмеженні швидкості передавання даних відправником у випадку наявності факту перевантаження. Якщо перевантаження в мережі не спостерігаються, то відправник може збільшувати швидкість передавання, в іншому випадку швидкість передавання примусово знижується.

При реалізації такого підходу необхідно визначитись з відповідями на нижченаведені запитання.

- Яким чином відправник обмежуватиме швидкість передавання даних?
- Як відправник може визначити наявність перевантаження на шляху між ним і одержувачем?
- В який спосіб буде змінюватись швидкість передавання при виявленні перевантаження?

Залежно від набору відповідей можна отримати різні алгоритми боротьби з перевантаженнями. Найбільш відомими є алгоритми Tahoe, Reno, Vegas, Hybla тощо. Розглянемо механізми боротьби з перевантаженнями на прикладі алгоритму Reno, припускаючи, що відправник здійснює передавання великого файлу даних.

Як зазначалося вище, головним механізмом обмеження швидкості передавання даних є вікно сегментів, або вікно передавання, а його розмір залежить від обсягу вільного місця в буфері прийому. Механізм контролю перевантаження використовує додатковий параметр, що має назву **вікно**

перевантаження. У випадку використання останнього розмір вікна передавання визначається як мінімальне з двох чисел – обсягу вільного місця та розміру вікна перевантаження.

Інше важливе питання – яким чином TCP визначає факт наявності перевантаження. Ознакою перевантаження є подія **втрата пакета**. Під час значних перевантажень в мережі буфери одного або декількох маршрутизаторів переповнюються, внаслідок чого частина пакетів відкидається. Втрата пакета фіксується відправником після закінчення інтервалу очікування або під час отримання трьох дублювальних квитанцій (рис. 5.9).

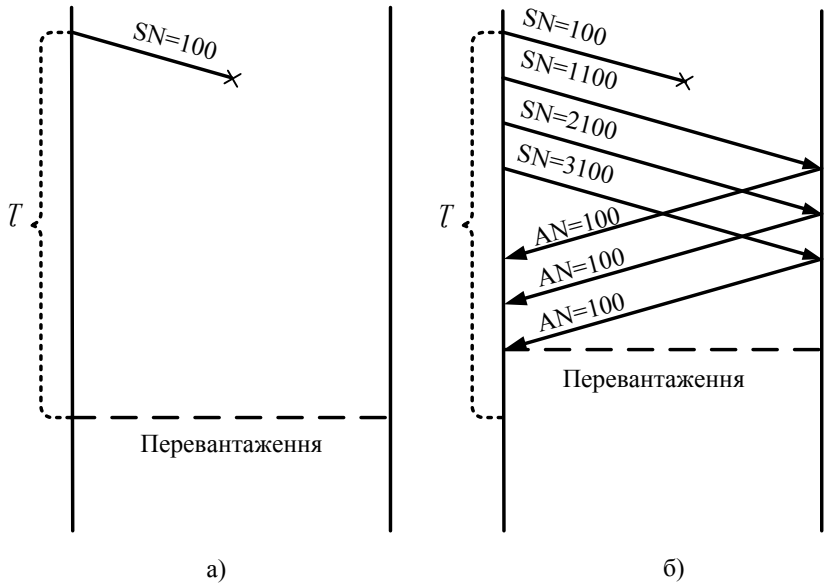


Рисунок 5.9 – Визначення факту наявності перевантаження:
а) за таймаутом, б) отримання трьох дублювальних квитанцій

Після встановлення факту перевантаження потрібно вжити певних заходів, спрямованих на вихід з цього стану. До переліку дій, спрямованих на вихід з перевантаження, входять: **адитивне збільшення** разом із **мультиплікативним зменшенням**, **повільний старт**, **реакція на перебільшення тайм-ауту.** Певну послідовність зазначених дій називають **алгоритмом контролю перевантаження протоколу TCP**.

Адитивне збільшення і мультиплікативне зменшення. Основною ідеєю механізму боротьби з перевантаженням засобами протоколу TCP є зниження швидкості передавання шляхом зменшення розміру вікна перевантажень після втрати пакетів. Цілком ймовірно, що у всіх TCP-з'єднаннях, які обслуговуються даним маршрутизатором, спостерігаються втрати пакетів, що призводить до одночасного зменшення вікон перевантажень всіма цими з'єднаннями. Кінцевий ефект полягає у зниженні трафіку, що проходить через перевантажений маршрутизатор і, як наслідок, в послабленні перевантаження. Проте актуальним залишається питання про

те, наскільки суттєвим має бути зниження швидкості передавання під час втрати пакета.

У протоколі TCP використовується так зване **мультиплікативне зменшення**, що значить зменшення вікна перевантаження у два рази після втрати пакета. Якщо втрата пакета відбулася при розмірі вікна перевантажень у 20 Кбайтів, то останнє буде зменшено до 10 Кбайтів. У випадку втрати ще одного пакета розмір вікна дорівнюватиме 5 Кбайтам. Зменшення розміру вікна перевантаження може відбуватися багаторазово, однак його значення не може бути меншим за максимальний розмір сегмента (MSS).

Відсутність перевантажень, або втрат пакетів, вказує на ймовірність присутності на лінії зв'язку ресурсів, що не використовуються, і є спонукальним мотивом для збільшення швидкості передавання даних. Збільшення швидкості відбувається повільно та плавно. Протокол TCP ніби «прощупує» вільні ресурси на шляху з'єднання. Кожен раз під час отримання квитанції за умови відсутності втрат пакетів значення вікна трохи збільшується (як правило, на величину MSS).

Таким чином, TCP здійснює адитивне збільшення швидкості передавання при відсутності перевантажень в мережі і мультиплікативне зменшення швидкості передавання за наявності перевантажень. Алгоритм контролю перевантажень протоколу TCP часто називають алгоритмом AIMD (Additive-Increase, Multiplicative-Decrease – адитивне збільшення і мультиплікативне зменшення). Фаза лінійного зростання потоку даних в протоколі контролю перевантаження називається виходом з перевантаження. Величина вікна перевантаження циклічно проходить через стадії лінійного зростання і різкого спаду до половини поточного значення під час втрати пакета. Графік змінення розміру вікна для TCP-з'єднань з тривалим часом життя нагадує зубці пилки і наведений на рис. 5.10.

Повільний старт. Під час встановлення TCP-з'єднання початковим значенням вікна перевантаження є величина MSS; отже, початкова швидкість передавання даних складає MSS/RTT. Наприклад, якщо MSS=1500 байтам, а RTT=30 мс, то початкова швидкість передавання з'єднання приблизно дорівнює 400 Кбіт/с. Оскільки максимально можлива швидкість передавання значно перевищує величину MSS/RTT, лінійне збільшення початкової швидкості є нераціональним, тому що цей процес тягнеться занадто довго.

Для прискорення збільшення розміру вікна на початковому етапі замість лінійного збільшення використовується експоненціальне збільшення, тобто, розмір вікна збільшується вдвічі після кожного підтвердження про отримання. Експоненціальне зростання продовжується до першої втрати пакета, після чого значення розміру вікна перевантаження зменшується вдвічі і в подальшому збільшується адитивно.

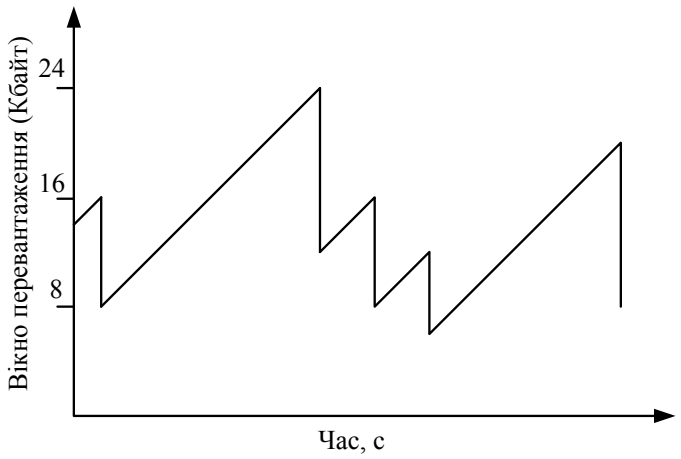


Рисунок 5.10 – Змінення розміру вікна перевантаження при застосуванні алгоритму мультиплікативного зменшення та адитивного збільшення

Отже, в першій фазі передавання даних через TCP-з'єднання, що має назву повільний старт, відправник починає передавання з повільною швидкістю, яка збільшується за експоненціальним законом. Вікно перевантаження спочатку дорівнює 1MSS, після успішного отримання даних одержувач відправляє підтвердження і збільшує вікно до 2MSS. Отримавши без втрат наступну порцію даних одержувач збільшує вікно до 4MSS і т. д. Ця процедура продовжується до першої втрати сегмента. Фазу повільного старту продемонстровано на рис. 5.11.

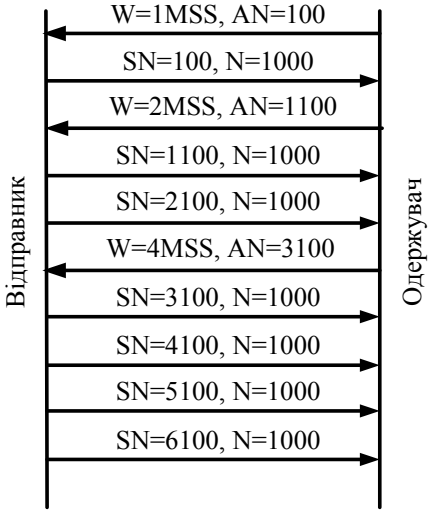


Рисунок 5.11 – Фаза повільного старту в роботі TCP-протоколу

Реакція на перебільшення тайм-ауту. Застосування механізмів повільного старту, мультиплікативного зменшення та адитивного збільшення дозволяє реалізувати всі необхідні для контролю перевантажень дії. Так, розмір вікна перевантажень протоколу TCP після встановлення з'єднання дорівнює величині MSS, далі експоненціально зростає до першої втрати

пакета, після чого спрацьовує алгоритм адитивного збільшення і мультиплікативного зменшення.

Для полегшення виходу з перевантаження протокол TCP неоднаково реагує на різні ознаки втрати пакета. Нагадаємо, що ознаками втрати пакета є спрацювання таймера повторного передавання та отримання трьох дублювальних квитанцій (див. рис. 5.9). У першому випадку спостерігається «жорстке» перевантаження – підтвердження не надходить вчасно не тільки на перший сегмент, а й на наступні. Другий варіант називають «м'яким» перевантаженням, оскільки втрачено тільки один сегмент, а інші надходять вчасно.

У випадку «м'якого» перевантаження поведінка TCP відповідає наведеному раніше алгоритму: розмір вікна перевантаження зменшується вдвічі, а потім починає лінійно збільшуватися. Якщо наявний факт «жорсткого» перевантаження, відправник переходить в стан повільного старту, зменшуючи розмір вікна перевантаження до величини MSS. Далі відбувається збільшення вікна за експоненціальним законом, доки його розмір не досягне величини, що дорівнює половині розміру вікна на момент спрацювання таймера перевантаження. Після цього зростання значення вікна перевантажень знову стає лінійним.

Для реалізації покращеного механізму керування перевантаженнями TCP використовує спеціальний параметр – порогове значення вікна, що визначає розмір вікна перевантаження, при якому завершується фаза повільного старту і починається фаза адитивного збільшення.

Зазвичай за замовчуванням порогове значення вікна є досить великим (найчастіше 65 Кбайтів) для мінімізації впливу цього параметра на початку передавання. Під час втрати пакета порогове значення вікна дорівнює половині поточного значення вікна перевантаження. Наприклад, якщо перед втратою пакета розмір вікна перевантаження становив 30 Кбайтів, то порогове значення вікна дорівнюватиме 15 Кбайтам і збереже таке значення до наступної втрати пакета.

Таким чином, покращений алгоритм керування перевантаженнями базуватиметься на нижченаведених правилах.

- Якщо розмір вікна перевантаження не перевищує порогового значення вікна, відправник знаходиться у фазі повільного старту, і розмір вікна перевантаження зростає за експоненціальним законом.
- Якщо розмір вікна перевантаження перевищує порогове значення вікна, відправник переходить у фазу виходу з перевантаження, і розмір вікна перевантаження збільшується лінійно.
- Після отримання трьох дублювальних квитанцій порогова величина встановлюється такою, що дорівнює половині поточного значення розміру вікна перевантаження, а розмір вікна перевантаження стає таким, що дорівнює пороговому значенню.
- Після спрацювання таймера повторного передавання значення порогової величини дорівнюватиме половині поточного значення ро-

зміру вікна перевантаження, а розмір вікна перевантаження зменшується до MSS.

Приклад роботи алгоритму керування перевантаженням показано на рис. 5.12.

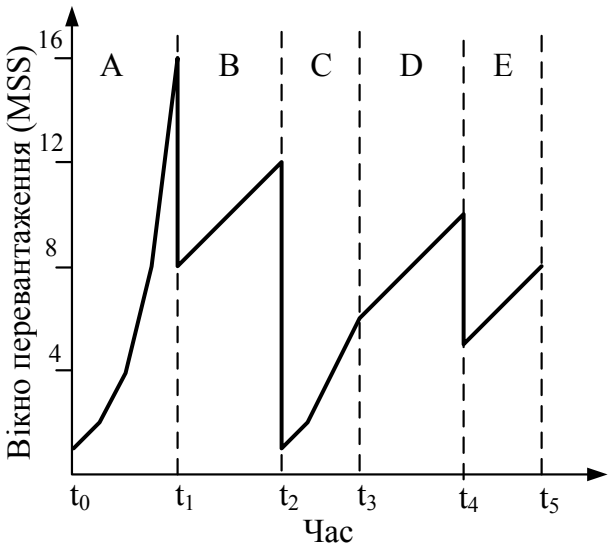


Рисунок 5.12 – Приклад роботи алгоритму керування перевантаженням TCP-протоколу

У момент часу t_0 відкривається TCP-з'єднання, встановлюється мінімальний розмір вікна перевантаження і починається фаза повільного старту (зона A).

У момент часу t_1 відбувається втрата пакета, про яку сповіщає потрійна квитанція. Це призводить до зменшення вікна в два рази і перехід до фази адитивного збільшення (зона B).

У момент часу t_2 знову відбувається втрата пакета, однак на цей раз спостерігається «жорстке» перевантаження. Розмір вікна перевантаження встановлюється 1MSS, а порогове значення фіксується на рівні половини поточного вікна перевантаження – 6MSS. Починається фаза повільного старту (зона C).

У момент часу t_3 розмір вікна перевантаження починає дорівнювати пороговому значенню, що ініціює перехід до фази адитивного збільшення (зона D).

У момент часу t_4 знову відбувається втрата пакета, але на цей раз спостерігається «м'яке» перевантаження, тому вікно зменшується вдвічі і відбувається адитивне збільшення (зона E).

5.4. Модифікації протоколу TCP

Протокол TCP є основним протоколом роботи в мережі Інтернет, який виконує доставку даних у вигляді потоків даних з встановленням з'єднання, і використовується у випадках, коли необхідна гарантована дос-

тавка повідомлень. Протокол TCP функціонує нормально при виконанні декількох принципових умов, а саме:

- ймовірність помилки доставки невелика і втрата пакетів відбувається внаслідок переповнення буфера;
- мережа має фіксовану полосу пропускання без суттєвих коливань;
- час доставки повідомлень на кінцеву станцію стабільний, пакети повідомлення передаються чітко послідовно;
- довжина TCP-сесії в декілька разів більша значення RTT (Round Trip Time), тобто інтервалу часу між відправленням пакета і закінченням його обробки у адресата;
- взаємодія з іншими TCP-сесіями не має призводити до різкого зниження ефективності створеного віртуального каналу;
- достатній розмір мережних буферів взаємодійних станцій, інакше пропускна спроможність буде знижуватись і визначатися не смугою пропускання каналу, а розміром буфера;
- достатньо значний розмір поля даних пакета з контролем на початку сесії значення максимального розміру даних, які передаються в каналі MTU.

Такі умови виконуються далеко не завжди, особливо при передаванні через супутникові канали та при використанні мобільного зв'язку, що призводить до зниження ефективності передавання.

Крім того, на сьогодні існує необхідність передавання великих обсягів даних (терабайти інформації), що, в першу чергу, актуально в таких сферах, як обмін даними між датацентрами, багатьма серверами та окремими абонентськими станціями. Протокол же TCP було розроблено для передавання набагато менших обсягів даних, ніж ті, з якими доводиться працювати сучасним додаткам. Тому в TCP можливі значні часові затримки, що одразу впливає на загальну пропускну спроможність мережі. Крім того, даний протокол не забезпечує необхідні в сучасних умовах надійність і масштабованість. Тому розглянемо основні модифікації протоколу TCP та основні їх характеристики.

В останні роки запропоновано декілька нових модифікацій протоколу TCP:

- TCP-Tahoe;
- TCP-Reno та його більш популярна версія NewReno (RFC-3782).;
- TCP-Vegas та більш сучасна версія Fast TCP;
- TCP-Westwood.

Алгоритм TCP Tahoe є найбільш давнім і достатньо широко використовується. При його реалізації контролюється заповнення буфера і при повному його заповненні і втраті якоїсь кількості переданих сегментів втрачений сегмент і всі, що були надіслані після нього, передаються адресату повторно. При значній ймовірності втрати пакетів такий підхід суттєво знижує пропускну спроможність і додатково збільшує завантаження каналу.

Протокол Fast TCP, який базується на TCP-Vegas, контролює розмір вікна передавання за рахунок аналізу затримки в черзі та втрати пакетів для визначення факту виникнення перевантаження в каналі. При встановленні факту, що мережа наближається до перевантаження, розмір вікна зменшується. Якщо перевантаження буде подолано, розмір вікна збільшується. Таким чином, розмір вікна передавання буде наближатися до необхідного, оптимального значення, що дозволяє зменшити втрати значної кількості пакетів.

Модифікація TCP-Westwood дозволяє досягнути більшої ефективності використання каналу за рахунок розробленого алгоритму керування вікном перевантаження, який базується на аналізі потоку даних, рівня втрат пакетів і поточного значення смуги пропускання. Такий підхід за певних умов дозволяє отримати кращі результати, ніж при використанні інших алгоритмів.

Версія протоколу TCP-Reno NewReno має декілька модифікацій, а саме: BIC-TCP (Binary Increase Control TCP), CUBIC TCP, P-TCP (Parallel TCP Reno), S-TCP (Scalable TCP), HS-TCP (HighSpeed TCP), HSTCP-LP (HighSpeed TCP Low Priority), H-TCP (Hamilton TCP).

Модифікації BIC-TCP та CUBIC TCP забезпечують керування перевантаженням для усунення проблеми повільної реакції протоколу TCP у високошвидкісних мережах великої довжини (fast long-distance networks), яка призводить до того, що значна частина пропускної спроможності залишається невикористаною.

HS-TCP використовує механізм керування перевантаженням в TCP для покращення функціонування протоколу TCP у високошвидкісних мережах зі значною (великою) затримкою.

Метою модифікації HSTCP-LP є використання тільки надлишкової пропускної спроможності мережі, при цьому невикористана пропускна спроможність може бути використана іншими TCP-потоками.

В модифікації S-TCP для зменшення втрати продуктивності після виникнення перевантаження змінено традиційний алгоритм керування перевантаженням TCP за рахунок керування розміром вікна.

Як було показано раніше, протокол TCP не задовольняє вимоги багатопоточних мереж з множинною адресацією. Наприклад, коли мобільний Wi-Fi інтерфейс телефона втрачає сигнал, всі TCP з'єднання, що прив'язані до IP-адреси інтерфейсу Wi-Fi, буде перервано. TCP також не може бути ефективно використано для встановлення з'єднання з задіянням декількох шляхів.

Тому будо розроблено **протокол MPTCP (Multipath TCP)** – багатомаршрутний TCP, який є набором розширень регулярного протоколу TCP для реалізації сервісів Multipath TCP, що дає можливість для одного з'єднання передавати дані через декілька маршрутів одночасно (RFC 6824). Даний протокол дозволяє організувати функціонування TCP-з'єднання з доставкою пакетів одночасно декількома маршрутами через різні мережні інтерфейси, які з'єднано з різними IP-адресами (рис. 5.13).

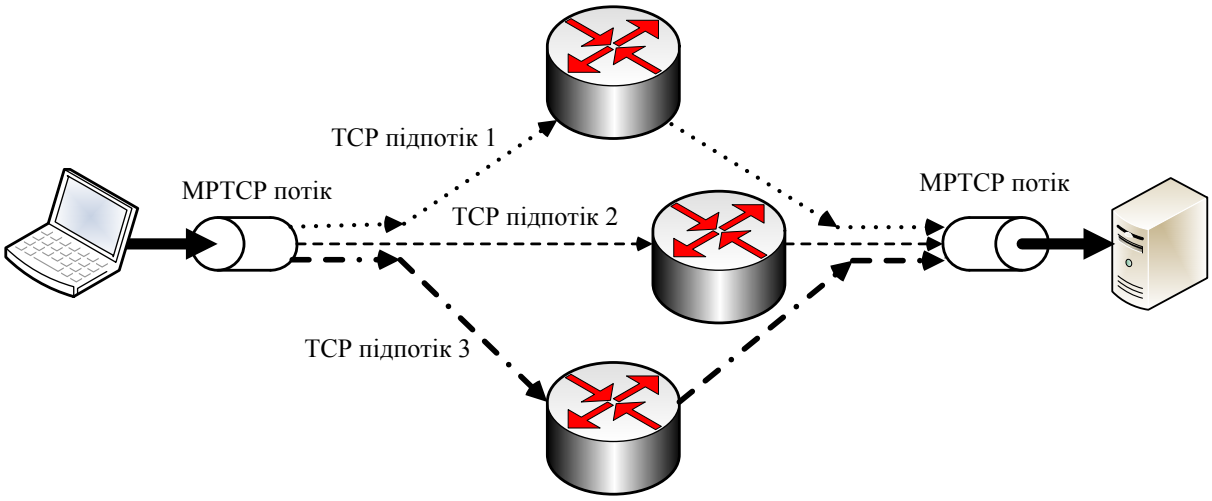


Рисунок 5.13 – Принцип функціонування протоколу MPTCP

Для мережних додатків таке об'єднання розглядається як звичайне TCP-з'єднання, а все керування потоками реалізується ресурсами MPTCP. Існують такі ключові обмеження реалізації Multipath TCP:

- реалізація має бути сумісною з уже існуючими версіями стандартного протоколу TCP;
- один або декілька вузлів мають декілька входів/виходів та декілька IP-адрес.

Протокол Multipath TCP може використовуватись не тільки для збільшення пропускної спроможності, а й для підвищення надійності. MPTCP функціонує на транспортному рівні та має бути прозорим для більш високих і більш низьких рівнів. Він фактично є набором додаткових можливостей поверх стандартного протоколу TCP (рис. 5.14).

Рівень додатків	
MPTCP	
Підпотік TCP	Підпотік TCP
IP	IP

Рисунок 5.14 – MPTCP у стеку TCP-IP

Додаток ініціює з'єднання шляхом відкриття TCP-сокета звичайним способом. З'єднання MPTCP ініціюється так само, як звичайне з'єднання TCP. Це показано на рисунку 5.15, де встановлено з'єднання MPTCP між адресами A1 і B1, між вузлами A і B. Якщо додаткові шляхи доступні, додаткові TCP сесії (MPTCP «підпотоки») ініціюються на цих шляхах і об'єднуються з існуючим підпотоком, який продовжує виступати єдиним

підключення для додатків на прикладному рівні. Додаткові TCP підпотоки подано між адресою A2 вузла A і B1 адресою вузла B. Створення нових підпотоків супроводжується передаванням SYN і ACK-пакетів, що містять опцію MP_JOIN. Клієнт ініціює новий підпотік, використовуючи пару IP-адрес (клієнт + сервер). А для ідентифікації з'єднання, яке встановлюється, використовується маркер, що генерується з використанням ключа.

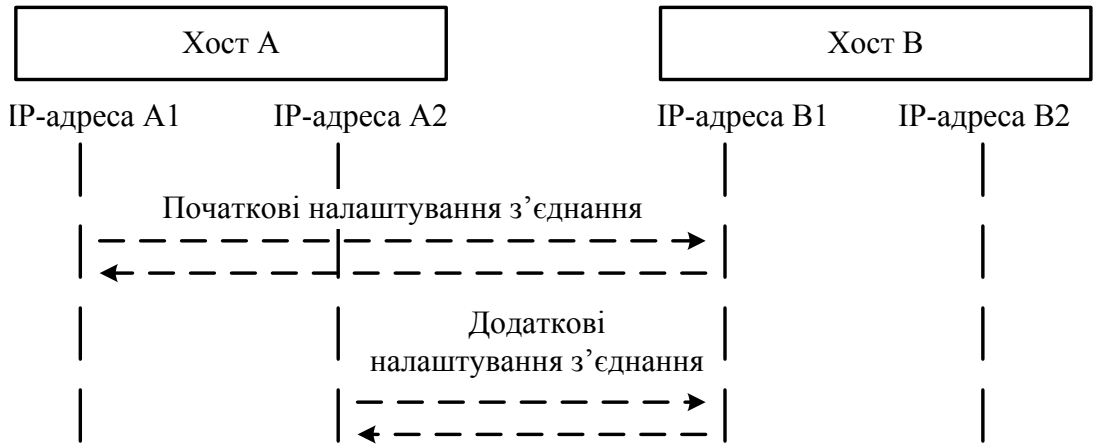


Рисунок 5.15 – MPTCP з'єднання з декількома адресами

MPTCP ідентифікує кілька шляхів наявності множинних адрес. Кількість комбінацій цих адрес може бути еквівалентна кількості додаткових шляхів. Наприклад, інші потенційні шляхи, що можуть бути створені: адреса A1 — адреса B2 і адреса A2 — адреса B2 тощо. Хоча показано, що додаткова сесія ініціюється з адреси A2, вона може бути ініційована і з адреси B1. Встановлення додаткових підпотоків досягається за рахунок способу керування маршрутом. У MPTCP-з'єднанні набір IP-адрес кінцевого вузла може бути змінено. MPTCP може додати нові адреси або видалити їх. Коли NAT блокує встановлення з'єднання в одному напрямку, сервер не може встановити додаткове з'єднання (новий підпотік). У цьому випадку сервер може анонсувати доступну адресу для клієнта без встановлення з'єднання через систему NAT. Наприклад, сервер інформує клієнта про альтернативні IP-адреси. Після чого клієнт може відправити опцію MP_JOIN до цієї нової адреси. Проміжне мережне обладнання може змінити IP-адресу, саме тому цей параметр використовує ідентифікатор адреси для ідентифікації адреси на хості.

Коли декілька підпотоків використовуються разом, дані, що приймаються з різними затримками, обробляються MPTCP завдяки використанню порядкових номерів на рівні з'єднання. MPTCP підпотоки можуть бути додані або видалені в будь-який час і забезпечують надійне з'єднання, де для доставки даних MPTCP використовує номер послідовності даних (DSN).

Кожний підпотік MPTCP ідентифікується 32-бітовою послідовністю, що виділена для порядкового номера підпотоку SSN (Session Sequence

Number). Директива MPTCP дозволяє відображати SSNs на DSNs (Data Sequence Numbers). Тому дані можуть бути повторно передані на різних підпотоках, у разі, коли стався збій. Ця директива здійснює відображення, а також може передавати підтвердження на рівні з'єднання DANs (Data Acknowledgement Numbers) для прийнятого DSN (рисунок 5.16).

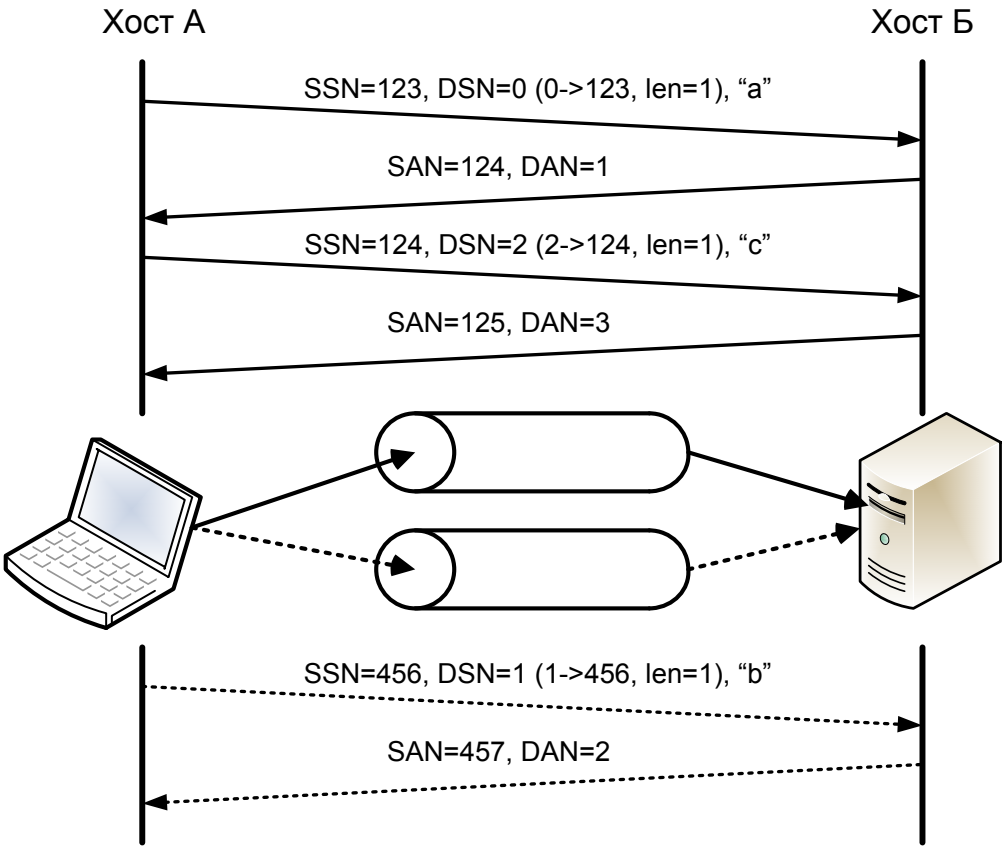


Рисунок 5.16 – Підтвердження пакетів MPTCP

Усі підпотоки мають спільний буфер ддя вхідних повідомлень і використовують той же розмір вікна прийому. Є два рівні підтверджень в MPTCP. Підтвердження на рівні підпотоків у MPTCP використовуються для підтвердження прийому сегментів, що відправляються підпотоками незалежно від їх джерела даних. Інший тип – підтвердження прийому на рівні MPTCP з'єднання для DSS. Ці підтвердження відстежують еволюцію в потоці і ковзних вікнах прийому.

Для завершення з'єднань через підпотоки клієнт передає «Data FIN» в рамках опції Data Sequence. Процедура є аналогічною TCP FIN, але виконується на рівні з'єднання. Після того, як всі дані отримані на рівні з'єднання, відправляється повідомлення «Data ACK».

Ці операції в MPTCP здійснюються завдяки обміну керівною інформацією (даними керування) MTPCP між хостом А (кліент) і хостом Б (сервер), що наведені на рисунку 5.16. Керівна інформація MPTCP представле-

на опціями в пакетах, тобто кожна операція MPTCP передається за допомогою одного числового типу з субваріантами – TCP опції.

SCTP (Stream Control Transmission Protocol) – протокол передавання з керуванням потоком (RFC 3257, 3286, 4960) – порівняно з протоколами TCP та UDP, виконує такі додаткові функції, як багатопоточність, захист від DDos атак, синхронне з'єднання між двома хостами двома або більшою кількістю незалежних фізичних каналів (multi-homing). SCTP забезпечує надійне передавання повідомлень між кінцевими хостами, використовуючи множинну адресацію (рис. 5.17), що, в свою чергу, забезпечує відмовостійкість, використовуючи надлишкові мережні шляхи. SCTP – протокол, орієнтований на постійне з'єднання, де кінцевий вузол може бути адресований за допомогою декількох IP-адрес в комбінації з SCTP-портами.

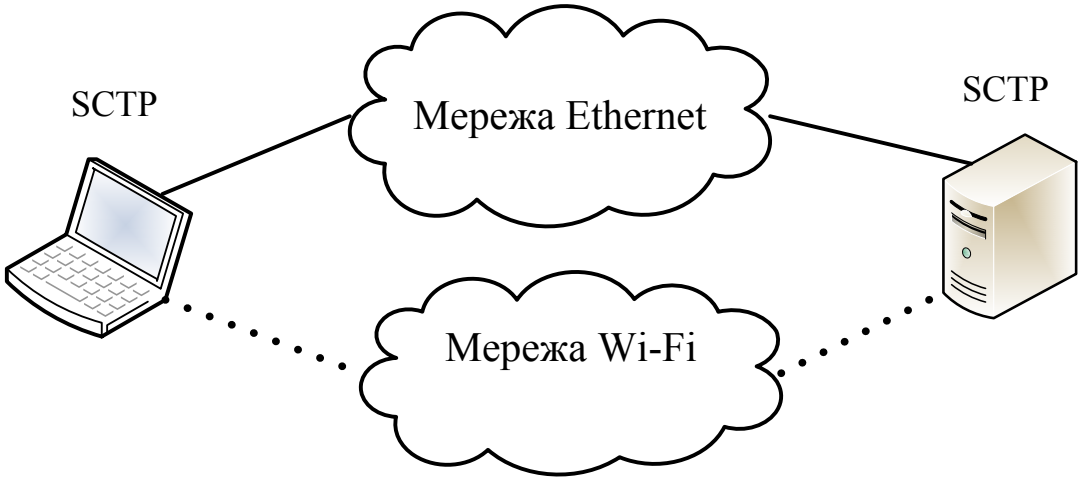


Рисунок 5.17 – Застосування SCTP

Цей протокол доступний в Linux, починаючи з версії ядра 2.6.

DCCP (Datagram Congestion Control Protocol) – протокол надає механізми для відслідковування перевантаження в мережі. Цей протокол не гарантує доставку інформації в необхідній послідовності, але є ефективним для додатків, в яких дані, що не доставлені в потрібний час, стають некорисними. Таким чином втрачається сенс обробки таких даних, оскільки вони вже не є актуальним. Наприклад, потокове медіа-мовлення, онлайн-ігри, інтернет-телефонія тощо. На сьогодні більшість таких додатків самостійно відслідковує перевантаження та використовує протоколи TCP та UDP. Протокол підтримується з версії 2.6.14 ядра Linux.

XTP (Xpress Transport Protocol) – протокол нового покоління, який долає такі недоліки протоколу TCP, як повільний старт, неефективна робота при втраті пакетів. В XTP розділені функції забезпечення надійного передавання даних та керування ним, а також допускає керування пропускною спроможністю каналу та вирішує проблему перевантаження каналу.

Принциповою особливістю XTP є незалежність від числа учасників інформаційного обміну (звичайний режим є еквівалентним груповому передаванню, з цієї причини XTP може використовуватися і як протокол групового передавання) і можливість роботи з MAC, IP або AAL5. Протокол призначений в перспективі замінити TCP та UDP. Групове передавання в XTP, на відміну від інших протоколів, може гарантувати доставку інформації, що може виявитися важливим при багатоточковому керуванні або в деяких розподілених базах даних.

При 5% втрати пакетів XTP забезпечує в 6 разів більшу пропускну спроможність, ніж TCP. У таблиці 5.3 наведено порівняльні результати вимірювання пропускної спроможності каналу ATM (155 Мбіт/с) при використанні протоколів TCP, UDP і XTP (використовувалися пакети довжиною 8190 байтів).

Таблиця 5.3 – Порівняльна характеристика TCP, UDP та XTP

Назва протоколу	Пропускна (Мбіт/с) спроможність
TCP	89–93
UDP	93–94
XTP	112–115

5.5 Питання для самоперевірки

1. Охарактеризуйте базові функції протоколів транспортного рівня.
2. Поясніть, яким чином відбувається мультиплексування та демультиплексування потоків даних на транспортному рівні. Чим обумовлена необхідність цієї процедури саме на транспортному рівні?
3. Порівняйте класи сервісів транспортного рівня та визначіть, для яких задач найкраще підходить кожен із них.
4. Поясніть призначення протокольних портів на транспортному рівні. Наведіть аналогії з інших галузей людської діяльності.
5. Охарактеризуйте протокол UDP. Наведіть приклади сервісів, що використовують протокол UDP.
6. Поясніть призначення основних полів TCP-заголовка.
7. Вкажіть, використання яких полів TCP-заголовка дозволяє пришвидшити доставку даних отримувачу.
8. Охарактеризуйте призначення контрольних бітів в заголовку TCP-сегмента.
9. Дайте порівняльну характеристику TCP- та UDP-протоколу, вказавши переваги і недоліки кожного з них.
10. Охарактеризуйте основні фази роботи TCP-протоколу.

11. Опишіть механізм підтверджень, що їх використовує протокол TCP.

12. Станція A відправила на станцію B три сегменти, всі вони були успішно доставлені. Номер в послідовності першого з них дорівнює 2300, в полі даних сегментів відповідно було передано 1000, 1200 та 1400 байтів. Вкажіть, що вказано в полі номера підтвердження в квітанції про отримання кожного з сегментів. Що зміниться, якщо перший сегмент було втрачено?

13. Що таке вікно сегментів і як його розмір впливає на швидкість передавання даних через TCP-з'єднання.

14. Нехай затримка передавання даних через TCP-з'єднання становить 200 мс; який треба встановити розмір вікна, що б швидкість передавання даних становила 2 Мбіт/с при пропускній спроможності каналу 10 Мбіт/с.

15. Порівняйте між собою режими повторного передавання даних, що використовуються TCP-протоколом. Вкажіть переваги та недоліки кожного з них.

16. Поясніть, які причини виникнення перевантажень в IP-мережі і які основні ознаки цього явища.

17. Які основні заходи вживає TCP-протокол для виходу зі стану перевантаження?

18. Опишіть, яким чином реагує протокол TCP на перебільшення тайм-ауту і отримання трьох дублювальних квитанцій, поясніть різницю.

19. Вкажіть основні переваги MPTCP порівняно з TCP.

20. Охарактеризуйте особливості протоколу MPTCP.

6 ПРОТОКОЛИ ВЕРХНІХ РІВНІВ

6.1 Протокол DHCP

Протокол DHCP (Dynamic Host Configuration Protocol) – це мережний протокол, що дозволяє вузлам мережі автоматично отримувати IP-адреси та інші параметри, що необхідні для роботи у мережі TCP/IP. Стандарт протоколу прийнято у 1993 р., поточну версію протоколу описано у RFC 2131. У 2003 р. з'явилася нова версія DHCP, що призначена для використання з IPv6, називається DHCPv6 і описана у RFC 3315. Далі буде розглядатися протокол, що оперує з IPv4-адресами.

Протокол працює відповідно до клієнт-серверної схеми. Клієнт – це комп'ютер мережі, що хоче отримати в оренду IP-адресу, а DHCP-сервери виконують функції диспетчерів, що надають адреси, контролюють їх використання і повідомляють клієнтам необхідні конфігураційні параметри. Сервер підтримує пул вільних адрес і веде власну реєстраційну базу даних. Також має забезпечувати взаємодію з ретрансляційними агентами протоколу BOOTP і обслуговувати BOOTP-клієнтів. DHCP має забезпечувати унікальність мережних адрес, що використовуються різними комп'ютерами мережі в даний момент, збереження попередніх конфігурацій клієнтських станцій після перезавантаження клієнта або сервера.

Протокол DHCP надає три способу розподілу IP-адрес:

- ручний розподіл. При цьому способі адміністратор встановлює відповідність апаратної адреси (для Ethernet мереж це MAC-адреса) кожного клієнтського комп'ютера і певної IP-адреси. Фактично, даний спосіб розподілу адрес відрізняється від ручного налагоджування кожного комп'ютера лише тим, що відомості про адреси зберігаються централізовано;
- автоматичний розподіл. При даному способі кожному комп'ютеру на постійне використання виділяється деяка вільна IP-адреса з певного діапазону, який задає адміністратор;
- динамічний розподіл. Цей спосіб аналогічний автоматичному розподілу за винятком того, що адреса видається комп'ютеру не на постійне користування, а на певний термін. Це називається орендою адреси. Після закінчення терміну оренди IP-адреса знову вважається вільною. Крім того, клієнт сам може відмовитися від отриманої адреси.

Деякі реалізації служби DHCP здатні автоматично оновлювати відповідні записи DNS при виділенні клієнтам нових адрес.

Крім IP-адреси DHCP також може повідомляти клієнтові додаткові параметри, необхідні для нормальної роботи в мережі. Ці параметри називаються опціями DHCP. Список стандартних опцій перераховано у RFC 2132. З них найбільш часто використовуються:

- IP-адреса маршрутизатора за замовчуванням;
- маска підмережі;
- адреси серверів DNS;
- ім’я домену DNS.

Взаємодія між DHCP-серверами і клієнтами виконується шляхом обміну повідомленнями. Формат повідомлення DHCP показано на рис. 6.1. У табл. 6.1 наведено типи полів повідомлення та їх призначення.

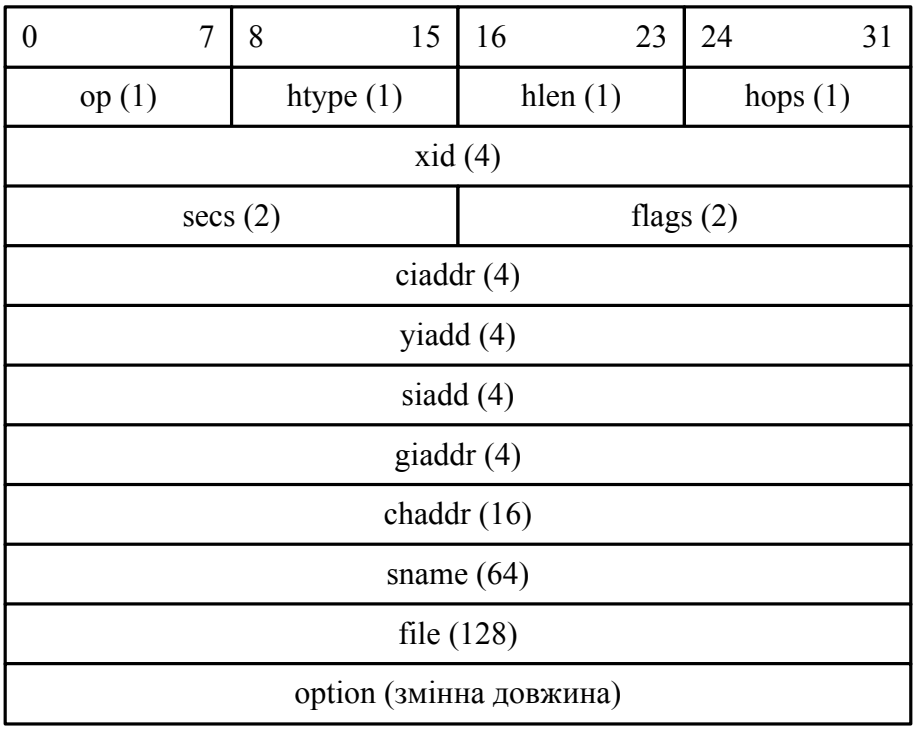


Рисунок 6.1 – Формат DHCP-повідомлення

Таблиця 6.1 – Значення полів DHCP-повідомлення

Поле	Байт	Призначення поля
op	1	Код операції повідомлення / тип повідомлення.
	1	1=BOOTREQUEST, 2=BOOTREPLY
htype	1	Тип апаратної адреси, наприклад, 1 для Ethernet.
hlen	1	Довжина апаратної адреси, наприклад, 6 для Ethernet.
hops	1	Клієнт встановлює це поле в нуль, поле може використовуватися агентами трасування, коли взаємодія виконується через посередника.
xid	4	ID транзакції, випадкове число, що вибирається клієнтом і використовується як клієнтом, так і сервером для встановлення відповідності між запитами і відповідями.
secs	2	Заповнюється клієнтом, кількість секунд з моменту початку запиту адреси або рестарту процесу.
flags	2	Прапорці.
ciaddr	4	IP-адреса клієнта, що заповнюється лише в тому випадку, якщо клієнт знаходиться в стані BOUND, RENEW або REBINDING і може реагувати на запити ARP.

Продовження табл. 6.1

yiadd	4	IP-адреса, що пропонується клієнту.
siadd	4	IP-адреса сервера.
giaddr	4	IP-адреса агента ретрансляції, використовується, якщо взаємодія здійснюється через посередника.
chaddr	16	Апаратна адреса клієнта.
sname	64	Ім'я сервера, рядок закінчується нулем.
file	128	Ім'я файлу завантаження (Boot-файлу), стрічка закінчується нулем; ім'я «generic» або нуль у DHCPDISCOVER, повний опис шляху у DHCPOFFER.
option	var	Поле необов'язкових параметрів.

Як транспортний протокол для обміну DHCP-повідомленнями використовується UDP. Коли відправляється повідомлення від клієнта на сервер, то використовується 67-й порт DHCP-сервера, для передавання повідомлення у зворотному напрямку – 68-й. Конкретні процедури взаємодії між клієнтами і DHCP-серверами регламентує RFC 1542.

Надання адреси в оренду виконується у відповідь за запит клієнта. DHCP сервер (або група серверів) гарантують, що надана адреса до закінчення терміну оренди не буде надаватися іншому клієнту. Коли клієнт повторно звертається до сервера, то останній намагається надати йому адресу, яку той використовував раніше. У свою чергу клієнт може надіслати запит на продовження оренди або довчасно відмовитися від неї. Протокол також може надати IP-адресу в необмежене використання.

6.1.1 Взаємодія клієнта і сервера при наданні мережної адреси. Послідовність подій при наданні адреси наведено на рис. 6.2, а опис повідомлень – у табл. 6.2.

Таблиця 6.2 – Типи і призначення повідомлень при взаємодії клієнта і сервера

Повідомлення	Призначення повідомлення
DHCPDISCOVER	Клієнт відправляє широкомовне повідомлення, щоб знайти доступний сервер.
DHCPOFFER	Відправляється сервером клієнту у відповідь на повідомлення DHCPDISCOVER і містить пропозицію з конфігураційними параметрами.
DHCPREQUEST	Повідомлення клієнта серверу: — запит параметрів від одного сервера і неявно відкидає пропозиції інших серверів; — підтвердження коректності раніше призначеної адреси, наприклад, після перезавантаження системи; — запит розширення часу життя конкретної мережної адреси.
DHCPACK	Відправляється сервером клієнту і містить конфігураційні параметри, в яких є також мережна адреса.
DHCPNAK	Відправляється сервером клієнту повідомлення про те, що мережна адреса некоректна, наприклад, клієнт переміщений в іншу підмережу, або час використання адреси закінчився.

Продовження табл. 6.2

DHCPDECLINE	Клієнт і сервер виявили, що мережна адреса вже використовується.
DHCPRELEASE	Відправляється клієнтом серверу з метою відмови від мережної адреси та анулює час дії адреси, що залишився.
DHCPINFORM	Відправляється клієнтом серверу з проханням надати локальні конфігураційні параметри; клієнт вже має отриману мережну адресу.

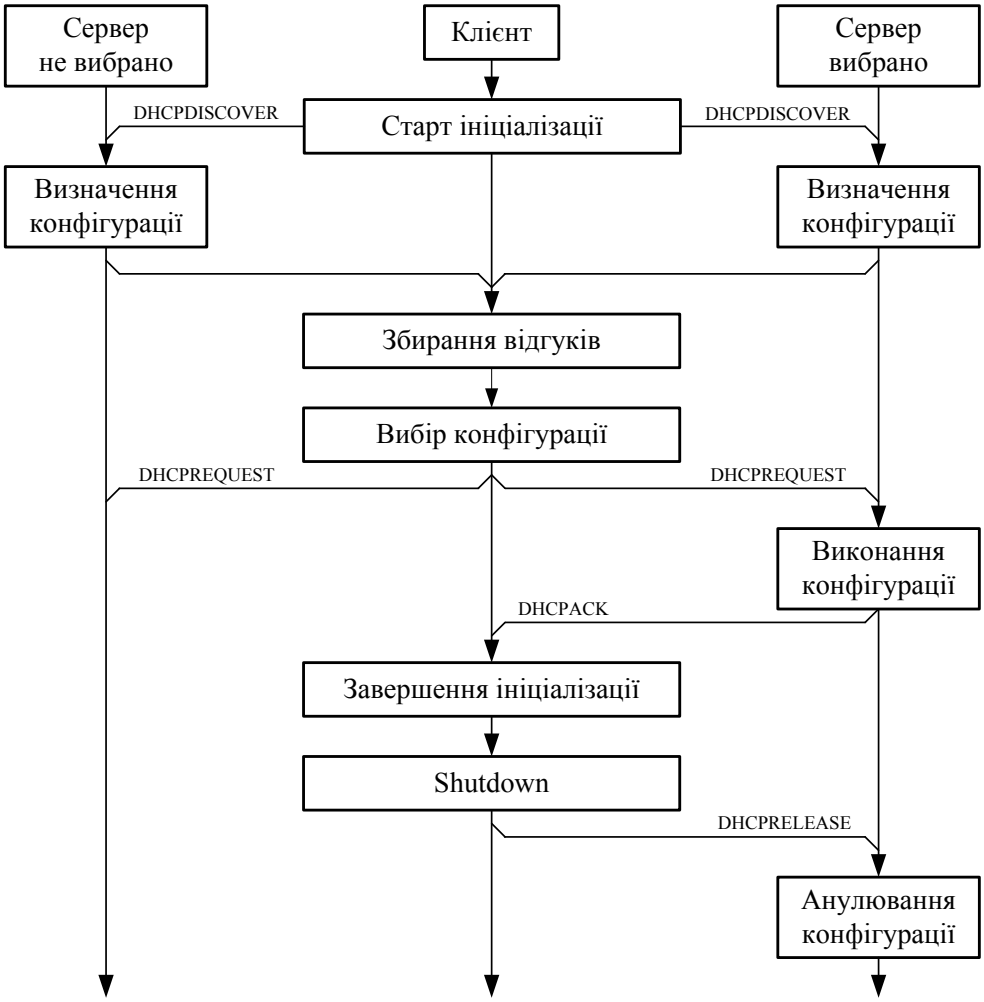


Рисунок 6.2 – Порядок взаємодії клієнта і сервера при видачі IP-адреси

1. Клієнт відправляє у власну фізичну підмережу широкомовне повідомлення DHCPDISCOVER, в якому може вказуватися адреса, що влаштовує клієнта, і час її оренди. Якщо в даній підмережі DHCP-сервера немає, повідомлення буде передано в інші підмережі ретранляційними агентами протоколу BOOTP (вони ж повернуть клієнту відповіді сервера).

2. Будь-який із серверів може відповісти на повідомлення DHCPDISCOVER повідомленням DHCPOFFER, додавши до нього доступну IP-адресу (yiaddr) і, якщо потрібно, параметри конфігурації клієнта. На даному етапі сервер не зобов’язаний резервувати вказану адресу. Він може

запропонувати іншому клієнту дану адресу. Разом з тим сервер може виконати перевірку з допомогою ICMP-запиту, чи не використовується дана IP-адреса в мережі.

3. Клієнт не зобов'язаний реагувати на першу ж пропозицію, що надійшла. Допускається, щоб він дочекався відгуків від декількох серверів і, зупинившись на одній із пропозицій, відправив у мережу широкомовне повідомлення DHCPREQUEST. У ньому містяться ідентифікатор обраного сервера і, можливо, бажані значення параметрів конфігурації. Якщо клієнта не влаштовує жодна з пропозицій, то він замість DHCPREQUEST знову надсилає в мережу запит DHCPDISCOVER. Якщо в процесі очікування відгуків від сервера на DHCPDISCOVER досягнуто тайм-аут, клієнт відправляє дане повідомлення повторно.

4. Ідентифікатор у повідомленні DHCPREQUEST дозволяє відповідному DHCP-серверу переконатися в тому, що клієнт прийняв саме його пропозицію. У відповідь сервер відправляє підтвердження DHCPACK, що містить значення необхідних параметрів конфігурації, і виконує відповідний запис у базу даних.

Якщо до моменту надходження повідомлення DHCPREQUEST запропонована адреса вже присвоєна іншому клієнту (наприклад, перша станція занадто довго реагувала), сервер відповідає повідомленням DHCPNACK.

5. Отримавши повідомлення DHCPACK, клієнт зобов'язаний переконатися в унікальності IP-адреси (засобами протоколу ARP) і зафіксувати сумарний термін її оренди. Останній розраховується як час, що минув між відправленням повідомлення DHCPREQUEST і прийомом відповідного повідомлення DHCPACK, плюс термін оренди, зазначений у DHCPACK.

Якщо адреса вже використовується іншою станцією, клієнт зобов'язаний надіслати серверу повідомлення DHCPDECLINE і не раніше ніж через 10 с почати всю процедуру знову. Процес конфігурування також поновлюється, коли клієнт отримав від сервера повідомлення DHCPNACK.

6. Для дострокового припинення оренди адреси клієнт відправляє серверу повідомлення DHCPRELEASE.

Наведена послідовність дій помітно спрощується, якщо клієнт бажає повторно працювати з IP-адресою, що була йому виділена раніше. У цьому випадку першим відправляється повідомленням DHCPREQUEST, в якому клієнт вказує адресу, що використовувалася раніше. У відповідь він може отримати повідомлення DHCPACK або DHCPNACK (якщо адреса зайнята або клієнтський запит є некоректним). У даному випадку унікальність адреси має перевіряти також клієнт.

Якщо на момент отримання запиту DHCPDISCOVER сервер не має вільних IP-адрес, він може надіслати повідомлення про проблему адміністратору. Разом з тим, зазвичай, клієнту виділяється адреса, що зафіксована за ним на даний момент. Якщо це неможливо, сервер запропонує адресу, якою користувався клієнт до закінчення терміну останньої оренди (за умови, що дана адреса вільна), або адресу, яку вказує сам клієнт у своєму за-

питі (знову ж таки, якщо адреса не зайнята). У тому випадку, коли всі попередні варіанти не підходять, нова адреса вибирається з пулу доступних адрес з урахуванням підмережі, з якої надійшов клієнтський запит.

6.1.2 Інтерпретація часу оренди, закінчення оренди. Клієнт отримує мережну адресу на певний період часу (який може бути нескінченним). Протокол DHCP час вимірює у секундах. Значення часу 0xffffffff зарезервовано для позначення нескінченності.

Оскільки клієнт і сервер можуть не мати синхронізованих годин, значення часу в DHCP-повідомленнях є відносним й інтерпретується з урахуванням показів локальних годин клієнта. Час вимірюється в секундах і подається у вигляді 32-бітових кодів без знака. Це дозволяє описувати відносні інтервали часу від 0 до приблизно 100 років, що цілком прийнятно для цілей протоколу DHCP.

Відповідно до алгоритму інтерпретації часу дії конфігураційних параметрів вважається, що годинник клієнта і сервера стабільні один відносно одного. Якщо годинник клієнта працює некоректно, то може виникнути ситуація, коли сервер вважає час дії наданої конфігурації вичерпаним, а клієнт – ні. Щоб уникнути такої ситуації, сервер може відправити клієнту значення часу дії коротше того, яке він записує у своїй базі даних.

Якщо термін оренди IP-адреси закінчується, то клієнт може завершити роботу з даною адресою, відправивши на DHCP-сервер повідомлення DHCPRELEASE, або завчасно запросити продовження терміну оренди. У першому випадку для отримання нової адреси необхідне виконання всієї процедури ініціалізації заново. У другому – клієнт продовжить функціонувати у мережі прозоро для користувача.

Для продовження оренди клієнт проходить два стани: поновлення адреси (RENEWING) і поновлення конфігурації (REBINDING). Перший настає приблизно на половині терміну оренди адреси (так званий момент T1), другий – після закінчення приблизно 7/8 повного часу оренди (момент T2); для розсинхронізації процесів реконфігурування різних клієнтів значення цих часових міток рандомізуються за допомогою випадкового часового доважка.

У момент T1 клієнт відправляє DHCP-серверу, який видав йому адресу, повідомлення DHCPREQUEST з проханням продовжити термін оренди. Отримавши позитивну відповідь (DHCPACK), клієнт перераховує термін оренди і продовжує роботу в звичайному режимі. Клієнт чекає приходу відповіді від сервера протягом (T2−t)/2 с (за умови, що це значення не менше 60 с), де t – час відсилання останнього повідомлення DHCPREQUEST, після чого дане повідомлення відправляє повторно.

Якщо відповідь від сервера не надійшла до моменту T2, клієнт переходить у стан REBINDING і передає вже широкомовне повідомлення DHCPREQUEST зі своєю поточною мережною адресою. У цьому випадку моменти повторних видач запитів DHCPREQUEST розраховуються анало-

гічно попередньому випадку, тільки замість T2 фігурує час закінчення терміну оренди.

Не виключено однак, що відповідь DHCPACK не прийде до закінчення терміну оренди. Тоді клієнт зобов'язаний негайно припинити виконання будь-яких мережних операцій і заново почати процес ініціалізації. Якщо відповідь DHCPACK все-таки надійде із запізненням, то клієнтові рекомендується відразу ж відновити роботу з використанням колишньої адреси.

6.2 Протокол DNS

Протокол DNS (Domain Name Service) використовується для отримання відповідності між іменами вузлів та їх IP-адресами. Система доменних імен являє собою розподілену базу даних, що використовується застосуваннями TCP/IP для встановлення даної відповідності. Також DNS використовується для маршрутизації електронної пошти. Термін розподілена база даних означає, що вона зберігається не на одному мережному вузлі. Кожен вузол підтримує власну інформаційну базу даних, а також запускає застосування, що відправляє запити до інших серверів. Протокол DNS дозволяє клієнтам і серверам спілкуватися між собою.

Доступ до DNS виконує застосування-визначник (resolver). Визначник – це підпрограма, що використовується для створення, відправлення й інтерпретації пакетів, що використовуються серверами імен у мережі. Для мережної програми потрібно виконати перетворення імені вузла у IP-адресу перед тим, як вона почне відкривати TCP-з'єднання або відправляти дейтаграму з використанням UDP. Концепції DNS описано у RFC 1034, а деталі розробки та специфікації DNS викладено у RFC 1035.

6.2.1 Основи DNS. Простір DNS імен має деревоподібну ієрархічну структуру. На рис. 6.3 показано організацію DNS.

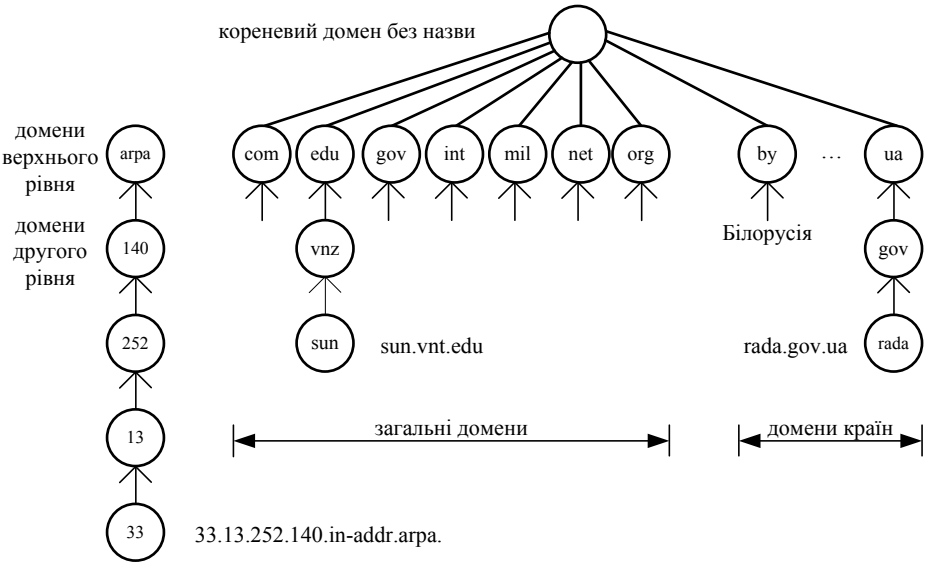


Рисунок 6.3 – Ієрархічна організація DNS

Кожен вузол на рис. 6.3 має мітку довжиною до 63 символів. Корінь дерева – це спеціальний вузол без мітки. Мітки можуть містити літери верхнього або нижнього регістрів. Ім'я домену для будь-якого вузла в дереві – це послідовність міток, що починається з кореневого вузла. При цьому мітки розділяють крапками. Кожен вузол дерева має мати унікальне ім'я домену, але однакові мітки можуть використовуватися в різних точках дерева.

Ім'я домену, що закінчується крапкою, називається абсолютним доменним іменем (absolute domain name) або повним іменем домену (FQDN – fully qualified domain name), наприклад, «static.rada.gov.ua.».

Домени верхнього рівня поділено на три зони:

1) .arpa – це спеціальний домен, що використовується для зіставлення адрес та імен;

2) сім трисимвольних доменів називаються загальними (general) або організаційними доменами (organizational). Їх класифікацію наведено у табл. 6.3. Імена .edu, .gov, .mil зарезервовані за організаціями США, інші можуть використовуватися за межами США. У 2000 р. було сім нових доменів верхнього рівня;

Таблиця 6.3 – Загальні домени

Домен	Опис	Домен (уведено з 2000 р.)	Опис
.com	Комерційні організації.	.aero	Галузі, що пов'язані з повітряним транспортом.
.edu	Освітні організації.	.biz	Організації, що пов'язані з бізнесом.
.gov	Урядові організації США.	.coop	Некомерційні організації.
.int	Міжнародні організації.	.info	Для необмеженого використання.
.net	Мережі.	.museum	Музеї.
.org	Інші організації.	Name	Окремі особи.
.mil	Військові організації США.	.pro	Бухгалтери, юристи, лікарі.

3) всі двосимвольні домени відповідають кодам країн. Їх перелік можна знайти у ISO 3166. Вони називаються доменами країн (country) або географічними (geographical) доменами.

Важлива характеристики DNS – це передавання відповідальності всередині DNS. Не існує організації, що керувала б і обслуговувала все дерево цілком. Замість цього одна організація NIC обслуговує лише частину дерева (домени верхнього рівня), а відповідальність за окремі зони передається іншим організаціям. Зона – це окрема керована частина дерева DNS. Відповідальна за керування зоною організація виконує налаштування серверів DNS (name servers) для цієї зони. Коли в зоні з'являється новий вузол (відповідно, нове ім'я), адміністратор зони поміщає ім'я і IP-адресу вузла в базу даних сервера DNS.

Сервер DNS може обслуговувати одну або кілька зон. Для цієї зони створюється основний сервер DNS (primary name server) і один або кілька вторинних серверів (secondary name server). Первинний і вторинний сервери мають бути незалежними і надлишковими таким чином, щоб система DNS не вийшла з ладу при відмові одного із серверів.

Якщо сервер DNS не має необхідної інформації, він встановлює контакт з іншим DNS-сервером. Кожен сервер має знати, як встановити контакт з кореневими серверами DNS (root name servers). У свою чергу, кореневі сервери знають імена та IP-адреси кожного офіційного сервера DNS для всіх доменів другого рівня.

Фундаментальна характеристика DNS – це кешування. Коли DNS-сервер отримує інформацію про відповідність імені та адреси, він кешує цю інформацію таким чином, щоб у випадку наступного запиту могла бути використана інформація з кешу. При цьому додатковий запит до інших серверів не виконується.

6.2.2 Формат повідомлення DNS. Як відзначалося, протокол DNS керує взаємодією між DNS-клієнтом та DNS-сервером. DNS-клієнт відправляє запит, а DNS-сервер повертає відповідь, що містить необхідну для клієнта інформацію. Локальний DNS-сервер надсилає відповіді клієнтам і видає запити іншим серверам. Кореневі сервери надають лише відповіді. Наприклад, програма хоче встановити IP-адресу для www.test.site.com. Клієнт зв’язується з локальним DNS-сервером, який звертається до кореневого DNS-сервера з метою дізнатися IP-адресу DNS-сервера .com. Далі локальний DNS-сервер відправляє запит DNS-серверу .com, щоб дізнатися IP-адресу DNS-сервера site.com. Після цього локальний DNS-сервер надсилає запит DNS-серверу зони site.com. Якщо зона має підзони, то може бути виданий додатковий запит домену test.site.com, який надасть у відповідь IP-адресу для www.test.site.com.

DNS-запит може бути рекурсивним або ітеративним. **Рекурсивний запит** вимагає, щоб DNS-сервер, який приймає запит, сам виконував перетворення. Наприклад, перетворювач видає рекурсивний запит локальному серверу імен на перетворення доменного імені у IP-адресу. На рис. 6.4 показано, що на етапі 1 перетворювач активізується через системний виклик. Далі перетворювач надсилає DNS-запит локальному серверу (етап 2) і чекає відповіді (етап 9). Локальний DNS-сервер здійснює дії з опрацювання запиту перетворювача. **Ітеративний запит** вимагає, щоб даний сервер у відповіді клієнту надав IP-адресу наступного в ієрархії DNS-сервера. Кореневі сервери обслуговують лише ітеративні запити. Локальний DNS-сервер надсилає запит кореневому DNS-серверу (етап 3), щоб дізнатися ім’я та IP-адресу DNS-сервера для зони на наступному рівні ієрархії (етап 4). Це допомагає розвантажити кореневі сервери. Локальний DNS-сервер може відправити запит наступному серверу в ієрархії (етапи 5, 6, 7, 8). Нарешті локальний сервер відповідає перетворювачу (етап 9), а перетворювач надає IP-адресу застосуванню (етап 10).

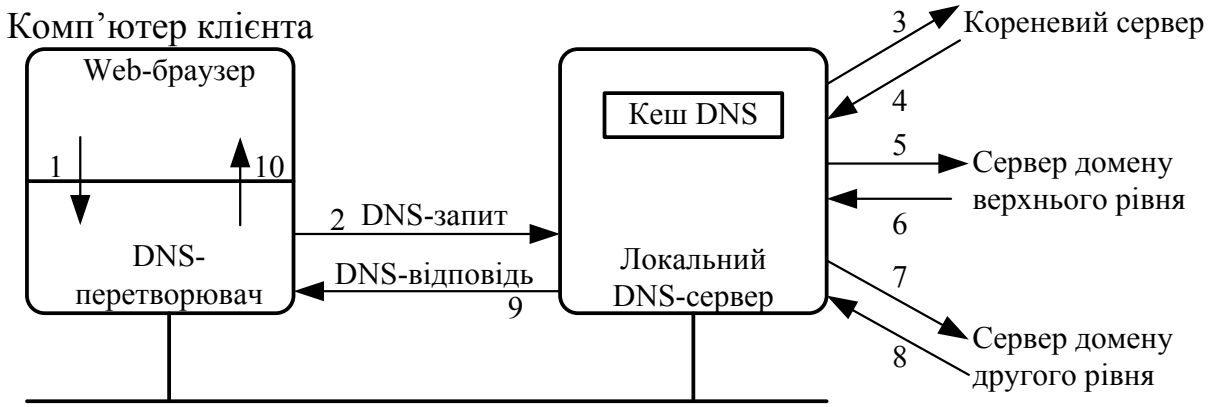


Рисунок 6.4 – Опрацювання DNS-запиту

Для DNS-запитів і DNS-відповідей використовується однаковий формат. На рис. 6.5 показано загальний формат DNS-повідомлення.

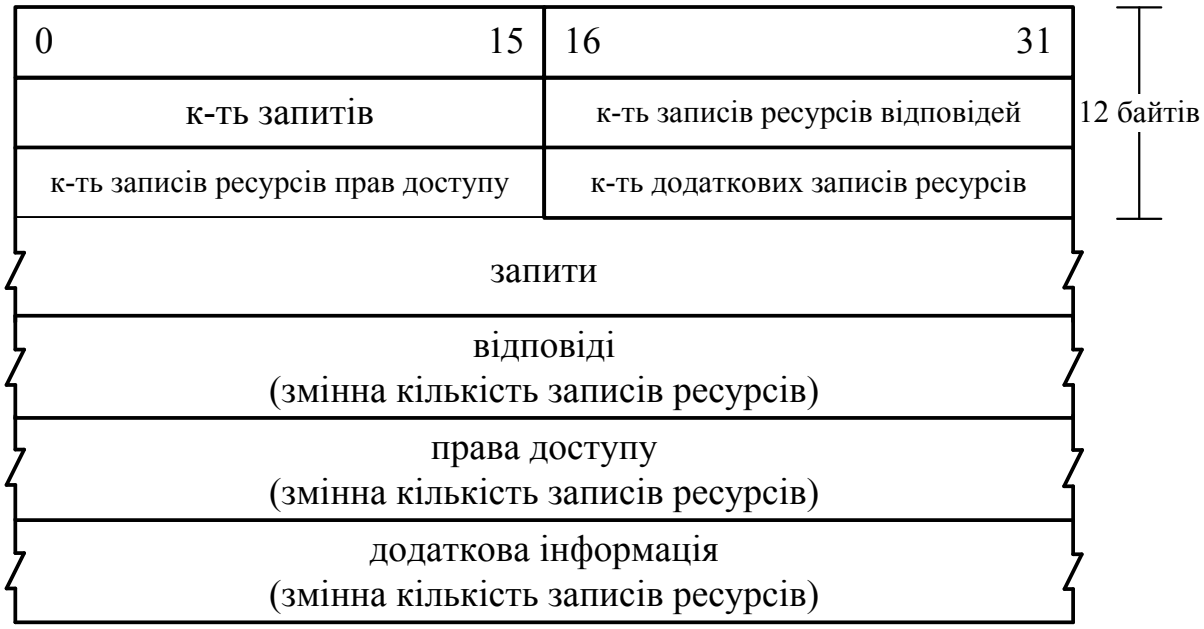


Рисунок 6.5 – Формат DNS повідомлення

Повідомлення містить фіксований 12-байтовий заголовок, за яким йде чотири поля змінної довжини. Значення полів:

1) поле ідентифікації (identification) встановлюється клієнтом і повертається сервером. Це поле дозволяє клієнту визначити, на який запит прийшла відповідь;

2) 16-бітове поле прапорців (flags) поділено на кілька частин, як показано на рис. 6.6.

1	4	1	1	1	1	3	4
QR	opcode	AA	TC	RD	RA	zero	rcode

Рисунок 6.6 – Поле прапорців заголовка DNS-повідомлення

Тут:

- QR (тип повідомлення), 1-бітове поле: 0 означає запит, 1 – відповідь;
- opcode (код операції), 4-бітове поле. Як правило, містить значення 0 (стандартний запит). Інші значення – це 1 (інверсний запит) і 2 (запит статусу сервера);
- AA – 1-бітовий прапорець, що означає «авторитетна відповідь» (authoritative answer). Сервер DNS має повноваження для цього домену у розділі запитів;
- TC – 1-бітове поле, що означає «обрізано» (truncated). Для UDP це означає, що повний розмір відповіді перевищує 512 байтів, але було повернуто лише перші 512 байтів відповіді;
- RD – 1-бітове поле, що означає «необхідна рекурсія» (recursion desired). Біт може бути встановлений у запиті, а потім повернутий у відповіді. Цей прапорець вимагає від DNS-сервера опрацювати даний запит як рекурсивний (recursive query), тобто сервер має сам визначити необхідну IP-адресу, а не повертати адресу іншого DNS-сервера. Якщо даний біт не встановлено і DNS-сервер, що отримав запит, не має авторитетної відповіді, він повертає у відповіді список інших DNS-серверів, до яких необхідно звернутися, щоб отримати відповідь. Це називається запитом, що повторюється (iterative query);
- RA – 1-бітове поле, що означає «рекурсія можлива» (recursion available). Цей біт встановлюється в 1 у відповіді, якщо сервер підтримує рекурсію. Більшість серверів DNS підтримує рекурсію, за винятком кількох кореневих серверів, які є надто завантаженими;
- zero – 3-бітове поле, що має дорівнювати 0;
- rcode – це 4-бітове поле коду відповіді. Звичайні значення: 0 (нема помилок) і 3 (помилка імені). Помилка імені повертається тільки від авторитетного DNS-сервера і означає, що імені домену, яке вказано у запиті, не існує;

3) наступні чотири 16-бітових поля вказують на кількість пунктів у чотирьох полях змінної довжини, що закінчують повідомлення. У запиті кількість запитань зазвичай дорівнює 1, а інші три лічильники дорівнюють 0. У запиті, що повертається, кількість відповідей дорівнює, як мінімум, 1, а інші можуть бути як нульовими, так і ненульовими. Формат кожного DNS-запиту у полі запитання показано на рис. 6.7 (зазвичай є лише одне запитання).

Рисунок 6.7 – Формат поля запитання DNS-повідомлення

Тут:

- ім’я запиту – це ім’я, що шукається. Воно виглядає як послідовність з однієї або кількох міток. Кожна мітка починається з 1-байтового лічильника, який містить кількість байтів, що йдуть за ним. Ім’я закінчується байтом, що дорівнює 0. Він є міткою з нульовою довжиною, а також міткою кореня. Кожний лічильник байтів має бути в діапазоні від 0 до 63, оскільки довжина мітки обмежується 63 байтами. Це поле може закінчуватися на обмежувачі, що не дорівнює 32 бітам, заповнення не використовується. На рис. 6.8 показано, як зберігається ім’я домену test.domain.edu.ua.

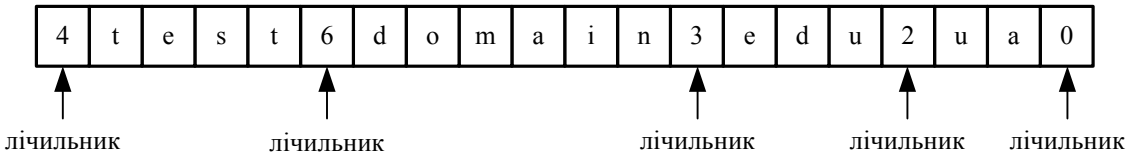


Рисунок 6.8 – Зберігання імені домену test.domain.edu.ua
у частині ім’я поля запитання DNS-запиту

- у кожного запитання є тип запиту (query type), а також відповідь має тип (type). Існує біля 20 різних значень, частина з яких уже застаріла. У табл. 6.4 наведено деякі з цих значень. Тип запиту – це надмножина (множина, підмножиною якого є дана множина) типів: два з наведених значень можуть використовуватися лише у запитаннях. Найбільш поширений тип запису – тип A, що означає, що необхідна IP-адреса для вказаного імені (query name). PTR-запит вимагає імені, що відповідає IP-адресі.

Таблиця 6.4 – Значення типу запиту поля запитання DNS-повідомлення

Ім’я	Цифрове значення	Опис	Тип (type)?	Тип запиту (query type)?
A	1	IP-адреса	Так	Так
NS	2	Сервер DNS	Так	Так
CNAME	5	Канонічне ім’я	Так	Так
PTR	12	Запис вказівника	Так	Так
HINFO	13	Інформація про вузол	Так	Так
MX	15	Запис про обмін поштою	Так	Так
AXFR	252	Запит про передавання зони	Ні	Так
* або ANY	255	Запит всіх записів	Ні	Так

- клас запиту (query class) звичайно дорівнює 1, що вказує на адреси Internet;

4) останні три поля DNS-повідомлення – це відповіді (answers), права (authority) і додаткова інформація (additional information). Загальний формат називається записом ресурсу (RR – resource record). На рис. 6.9 показано загальний формат запису ресурсу.

Рисунок 6.9 – Загальний формат запису ресурсу

На рис. 6.9 відповідні поля означають таке:

- ім’я домену (domain name) – це ім’я, якому відповідають наступні дані ресурсу. Формат імені такий же, як показано на рис. 6.7 для поля запитання;
- тип (type) вказує на один із типів кодів RR. Це те саме, що і значення типу запиту. Для даних мережі Internet поле клас (class) звичайно встановлено в 1;
- поле час життя (TTL – time-to-live) – це кількість секунд, протягом яких RR може бути кешований клієнтом. Звичайно TTL RR дорівнює 2 дням;
- довжина запису ресурсу (resource data length) вказує на кількість даних ресурсу (resource data). Формат цих даних залежить від типу (type). Для типу, що дорівнює 1 (запис A), дані ресурсу – це 4-байтова IP-адреса.

6.3 Протоколи Telnet та SSH

6.3.1 Протокол Telnet (TErminaL NETwork). Одна з найстаріших інформаційних технологій мережі Internet. Основним призначенням протоколу є реалізація мережного термінала для доступу до ресурсів віддаленого комп’ютера. Протокол Telnet забезпечує двонаправлений 8-бітовий канал передавання даних. Його головним завданням є створення стандартного методу взаємодії термінальних пристроїв і термінал-орієнтованих процесів через мережу. Специфікацію протоколу наведено у RFC 854.

Telnet використовує TCP-з’єднання для передавання даних, що суміщається з керівною інформацію протоколу. Протокол може використову-

ватися для з'єднання з будь-яким сервісом. Він знаходиться на сеансовому рівні моделі OSI. Для забезпечення віддаленого доступу до термінала сервера стандартом резервується порт 23.

Основу протоколу складають три базові концепції:

- концепція мережного віртуального термінала (NVT – network virtual terminal);
- принцип узгодження параметрів з'єднання;
- забезпечення симетрії терміналів і процесів, що дозволяє протоколу відправляти команди керування будь-якій стороні з'єднання. Поділу на «клієнт» і «сервер» немає.

NVT – це уявний пристрій, що знаходиться на обох кінцях з'єднання у клієнта і сервера, за допомогою якого встановлюється відповідність між їхніми реальними терміналами. Таким чином, операційна система клієнта має визначати відповідність між типом термінала, за яким працює користувач, і NVT. У свою чергу, сервер має встановлювати відповідність між NVT і тими типами терміналів, які він (сервер) підтримує.

NVT слугує символьним пристроєм з клавіатурою і принтером. Дані, введені користувачем з клавіатури, відправляються серверу, а дані, отримані від сервера, надходять на принтер. За замовчуванням клієнт відображає ехом на принтер все, що ввів користувач. Відповідно до принципу узгодження параметрів NVT це мінімально необхідний набір параметрів, що дозволяє працювати навіть «найпримітивнішим» пристроям. Сучасніші пристрої мають більші можливості подання інформації. Принцип узгодження параметрів дозволяє використовувати ці можливості. Наприклад, NVT є терміналом, який не може використовувати функції керування курсором, а реальний термінал, з якого здійснюється робота, можливо, вміє це робити. Використовуючи узгодження параметрів, термінальна програма пропонує обслуговувальному процесу використовувати керівні послідовності для керування виведенням інформації. Отримавши таку команду процес починає вставляти керівні послідовності в дані, призначені для відображення.

Протокол пропонує структуру додаткових параметрів «DO, DONT, WILL, WONT», що дозволяє користувачеві і серверу більш точно домовитися про особливості з'єднання. Команди мають таке значення:

- WILL – відправник хоче встановити дану опцію для себе;
- DO – відправник хоче, щоб отримувач встановив дану опцію;
- WONT – відправник хоче вимкнути цю опцію для себе;
- DONT – відправник хоче, щоб отримувач вимкнув опцію.

Наприклад, команда WILL XXX вказує на пропозицію стороні-відправника використовувати параметр XXX. DO XXX і DON'T XXX є, відповідно, позитивною і негативною відповіддю. Сценарії обговорення опцій Telnet наведено у табл. 6.5.

Команди Telnet передаються у вигляді 8-бітових послідовностей (байтів). Так, команді DO відповідає байт 253, WONT – 252 і т. д.

Таблиця 6.5 – Шість сценаріїв обговорення опцій Telnet

Команда відправника	Команда отримувача	Опис
WILL	DO	Відправник хоче встановити опцію, отримувач відповідає ТАК
WILL	DONT	Відправник хоче встановити опцію, отримувач говорить НІ
DO	WILL	відправник хоче, щоб отримувач встановив опцію, отримувач говорить ТАК
DO	WONT	Відправник хоче, щоб одержувач встановив опцію, отримувач говорить НІ
WONT	DONT	Відправник хоче вимкнути опцію, отримувач має сказати ТАК
DONT	WONT	відправник хоче, щоб отримувач вимкнув опцію, отримувач має сказати ТАК

Установлення певного параметра відбувається так: один з учасників з'єднання посилає іншому запит, пропонуючи використовувати під час роботи певний параметр. Якщо інша сторона погоджується, параметр негайно задіюється. Якщо приходить відмова, використовується те значення параметра, яке визначається для NVT. Звичайно, параметри узгоджуються спочатку при встановленні з'єднання, хоча при роботі одна зі сторін може знову змінити якісь параметри. Команди клієнта Telnet описано у табл. 6.6.

Таблиця 6.6 – Команди Telnet-клієнта

Параметр	Дія
open або o	Встановлення Telnet-підключення до комп'ютера або віддаленого сервера. Можна використовувати повну команду open або скорочену o. Наприклад, командою o remcomp 44 буде встановлено підключення до комп'ютера з іменем remcomp через порт 44.
close або c	Закриття з'єднання Telnet. До даної команди можна додати ім'я віддаленого вузла і номер порту. Наприклад, командою c remcomp 44 підключення до віддаленого сервера remcomp через порт 44 буде закрито.
display	Перегляд поточних параметрів клієнта Telnet. Наберіть дану команду, щоб відобразити список поточних робочих параметрів. Для зміни параметрів під час встановленого Telnet-сеансу (підключення до Telnet-серверу) натисніть комбінацію клавіш CTRL+] для виходу з Telnet-сеансу. Для повернення в сеанс натисніть клавішу ENTER. Доступні такі робочі параметри: WILL AUTH (перевірка дійсності NTLM); WONT AUTH; WILL TERM TYPE; WONT TERM TYPE; LOCALECHO off; LOCALECHO on.
quit або q	Вихід із програми Telnet.

Продовження табл. 6.6

Set	Завдання типу telnet-термінала для підключення, включення виведення локального еха, встановлення значення перевірки автентичності NTLM, задання символа перемикання режиму та встановлення ведення журналу. SET NTLM вмикає NTLM. Якщо використовується перевірка автентичності NTLM, при підключенні з віддаленого комп'ютера не буде видаватися запит введення імені та пароля. SET LOCALECHO включає режим локального відображення команд. SET TERM {ANSI | VT100 | VT52 | VTNT} задає зазначений тип термінала. При роботі зі звичайними програмами командного рядка слід використовувати тип термінала VT100. При роботі з розширеними програмами командного рядка слід використовувати тип термінала VTNT. ESCAPE + символ задає послідовність клавіш для перемикання з режиму сеансу в режим команд. Наприклад, щоб задати символ перемикання режиму CTRL + P, уведіть SET ESCAPE, натисніть клавіші CTRL + P і натисніть клавішу ENTER. LOGFILE ім'я файлу задає файл журналу активності Telnet. Файл журналу має знаходитися на локальному комп'ютері. Запис подій у журнал при встановленні цього параметра починається автоматично. LOGGING вмикає ведення журналу. Якщо не заданий файл журналу, видається повідомлення про помилку.
unset	Вимикає виведення локального еха або встановлення перевірки автентичності за запитом імені чи пароля. UNSET NLM відключає NLM. UNSET LOCALECHO відключає режим локального відображення команд.
status	Визначає, чи встановлено підключення клієнта Telnet.
CTRL+]	Перемикання у режим командного рядка Telnet із сеансу підключення.
enter	Перемикання в сеанс, якщо встановлено підключення.
? або help	Відображення довідкової інформації.

6.3.2 Протокол SSH (Secure SHell). Коли почали широко використовуватися алгоритми шифрування при передаванні даних у мережі, одне з перших завдань було організувати безпечне середовище. До цього існувала система RSH, що дозволяла певним користувачам з певних комп'ютерів (між ними мають були бути довірчі відносини) працювати на сервері з його оболонкою. Це практично те ж саме, що і Telnet-доступ. Але з розвитком мереж було виявлено «дірки» RSH.

Тому було розроблено новий протокол SSH. Всі дані, що передаються через SSH, шифруються. Існує кілька версій протоколу SSH, що розрізняються алгоритмами шифрування та загальними схемами роботи. У даний час використовується протокол SSH версії 2.

Загалом, зараз SSH є комерційним продуктом, що суперечить вимогам безпеки. Всім має бути відомий початковий код системи захисту інформації, щоб переконатися у відсутності будь-яких backdoors.

SSH надає 3 способи аутентифікації клієнта:

- за IP-адресою клієнта (небезпечно);
- через публічний ключ клієнта;
- стандартний парольний метод.

При запиті клієнта сервер повідомляє йому, які методи аутентифікації він підтримує і клієнт по черзі намагається перевірити їх. За замовчуванням клієнт спочатку намагається аутентифікуватися своєю адресою, потім публічним ключем і, якщо нічого не спрацювало, передає пароль, введений з клавіатури (при цьому пароль шифрується асиметричним шифруванням). Після проходження аутентифікації одним з методів з наявних у клієнта і сервера пар ключів генерується ключ симетричного шифрування, він генерується на підставі свого секретного та віддаленого публічного ключів. Після чого всі наступні дані, що передаються через SSH, шифруються даним ключем (зазвичай використовується алгоритм AES з довжиною ключа 128 бітів).

Протокол SSH версії 1 мав деякі «баги» в шифруванні даних, що передаються, і був, насправді, методом безпечної аутентифікації. Протокол версії 2 підтримує більш сучасні методи шифрування даних, також разом з даними надсилаються контрольні суми формату SHA або MD5, що унеможливлює підміну або іншу модифікацію даних.

6.4 Протоколи електронної пошти

Електронна пошта або мережна поштова служба – це розподілене застосування, головною функцією якого є надання користувачам мережі можливості обмінюватися електронними повідомленнями. Електронна пошта побудована відповідно до клієнт-серверної архітектури. Поштовий клієнт розміщується на комп'ютері користувача, а поштовий сервер, як правило, – на виділеному комп'ютері.

Поштовий клієнт – це програма, що використовується для надання інтерфейсу користувачеві, а також для засобів роботи з електронними повідомленнями. Поштовий сервер виконує прийняття повідомлень від клієнтів і перенаправлення їх іншим клієнтам відповідно до заданих правил.

Обмін поштою з використанням TCP виконується через агентів передавання повідомлень (MTA – message transfer agent). Наприклад, найбільш поширений MTA для Unix-систем – це Sendmail. Користувачі звичайно не спілкуються з MTA. Задача системного адміністратора встановити локальний MTA.

Електронні повідомлення мають стандартний формат. У спрощеному варіанті його можна подати у вигляді двох частин: заголовка, що містить службову інформацію, та тіла повідомлення, що містить сам «лист» користувача. Головна частина заголовка – це адреса відправника і отримувача у форматі user@domain.com, де user – ідентифікатор користувача поштової служби, domain.com – ім'я домену, до якого належить даний користувач. Також в заголовок поміщаються дата і тема листа, відмітки про шифру-

вання, терміновість доставлення, необхідність підтвердження факту прочитування листа отримувачем, використовуваний вид кодування. Також до листа можуть додавати медіа-файли.

Загалом електронна пошта складається з трьох частин:

1. Конверт, що використовується MTA для доставлення. RFC 821 визначає вміст та інтерпретацію конверта, а також протокол, який використовується для обміну поштою TCP-з'єднанням;
2. Заголовки, що використовуються користувацькими агентами;
3. Тіло – це вміст повідомлення від відправники до отримувача. RFC 822 визначає тіло повідомлення.

Як засіб передавання повідомлень поштова служба використовує **протокол SMTP** (Simple Mail Transfer Protocol). SMTP реалізується через несиметричні взаємодійні частини: SMTP-клієнта та SMTP-сервера. Цей протокол орієнтований на передавання даних від клієнта до сервера, тобто SMTP-клієнт працює з боку відправника, ініціює з'єднання і відсилає запит на обслуговування, а SMTP-сервер з боку отримувача відповідає на запити клієнта.

Сьогодні, як правило, використовується схема поштової служби з виділеним сервером, що постійно передає повідомлення від багатьох відправників до багатьох отримувачів. Для кожного домену імен DNS створюються записи типу MX, в яких зберігаються DNS-імена поштових серверів, що обслуговують користувачів даного домену.

Користувач за допомогою поштового клієнта створює повідомлення, вказує адресу отримувача і відправляє лист. Далі відбувається звертання до системи DNS, щоб визначити DNS-ім'я поштового сервера. Після його отримання відбувається повторне звернення до DNS для отримання IP-адреси поштового сервера.

SMTP-клієнт відправляє за даною IP-адресою запит на встановлення TCP-з'єднання через порт 25. З цього моменту починається діалог між клієнтом і сервером за протоколом SMTP. У табл. 6.7 наведено команди протоколу SMTP. Тут потрібно відзначити, що кожна цифра у коді відповіді має певне значення. Перша цифра означає, чи було виконання команди успішним (2), неуспішним (5) або ще не закінчилося (3). Друга і третя цифри коду відповіді пояснюють значення першої.

Таблиця 6.7 – Команди протоколу передавання пошти SMTP

Команда/Код	Опис
HELO	Ініціює передавання.
MAIL	Починає поштову транзакцію, яка закінчується передаванням даних в один або кілька поштових ящиків.
RCPT	Ідентифікує отримувача поштового повідомлення.
DATA	Дані, що йдуть після цієї команди, розглядаються отримувачем як дані поштового повідомлення. Поштове повідомлення закінчується комбінацією символів CRLF-крапка-CRLF.
RSET	Перериває поточну поштову транзакцію.

Продовження табл. 6.7

NOOP	Вимагає від отримувача не виконувати ніяких дій, а лише видавати відповідь OK. Використовується у більшості випадків для тестування.
QUIT	Вимагає видавати відповідь OK і закривати поточне з'єднання.
VRFY	Вимагає від отримувача підтвердити, що його аргумент є дійсним іменем користувача.
SEND	Починає поштову транзакцію, що доставляє дані на один або кілька терміналів (а не у поштові ящики).
SOML	Починає транзакцію MAIL або SEND, що доставляють дані на один або кілька терміналів і у поштові ящики.
SAML	Починає транзакцію MAIL і SEND, що доставляють дані на один або кілька терміналів і у поштові ящики.
EXPN	Команда SMTP-отримувачу підтвердити, чи дійсно аргумент є адресою поштового розсилання і якщо так, то повернути адресу отримувачу повідомлення.
HELP	Команда SMTP-отримувачу повернути повідомлення-довідку про його команди.
TURN	Команда SMTP-отримувачу або сказати OK і змінити ролі, тобто стати SMTP-передавачем, або відправити повідомлення-відмову і залишатися у ролі SMTP-отримувача.
211	Відповідь про стан системи або допомогу.
214	Повідомлення-підказка (допомога).
220	<ім'я_домену> служба готова до роботи.
221	<ім'я_домену> служба закриває канал зв'язку.
250	Дія поштової транзакції, запит на яку було виконано, успішно закінчилася.
251	Даний адресат на є місцевим; повідомлення буде передано по маршруту <forward-path>.
354	Починай передавання повідомлень.
421	<ім'я_домену> служба недоступна; з'єднання закривається.
450	Команда поштової служби не виконана, оскільки поштовий ящик недоступний.
451	Команда не виконана; відбулась локальна помилка при опрацюванні повідомлення.
452	Команда не виконана; системі не вистачило ресурсів.
500	Синтаксична помилка в тексті команди; команда не ідентифікована.
501	Синтаксична помилка в аргументах або параметрах команди.
502	Дана команда не реалізована.
503	Неправильна послідовність команд.
504	У даної команди не має бути аргументів.
550	Команда не виконана, оскільки поштовий ящик недоступний.
551	Даний адресат не є місцевим; спробуйте передати повідомлення по маршруту <forward-path>.
552	Команда транзакції перервана; переповнено дисковий простір.
553	Команда не виконана; вказано недоступне ім'я поштового ящика.
554	Транзакція не виконана.

Лист зберігається у буфері сервера, а потім переноситься в індивідуальний буфер отримувача. Такий буфер називається поштовим ящиком. Таким чином, поштовий сервер має вирішувати різні задачі щодо організації багатокористувацького доступу, зокрема: керування розподіленими ресурсами і забезпечення безпечного доступу.

Далі у певний момент отримувач запускає свого поштового агента і виконує команду перевірки пошти. Після цього запускається протокол доступу до поштового сервера (POP3 або IMAP) для отримання від нього даних, тобто відбувається доставляння отримувачу його листа.

Протоколи POP3 (Post Office Protocol v.3) і IMAP (Internet Mail Access Protocol) забезпечують доступ користувачам до кореспонденції, що зберігається на поштовому сервері. Обидва протоколи підтримують аутентифікацію користувачів на основі ідентифікаторів і паролей. Отримуючи доступ до сервера за протоколом POP3, поштовий агент завантажує адресовані йому повідомлення в пам'ять свого комп'ютера, при цьому на сервері не залишається «сліду» від пошти, що прийнята. Якщо доступ реалізовано за протоколом IMAP, то у пам'ять локального комп'ютера передаються лише копії повідомлень, що зберігаються на поштовому сервері. Також протокол IMAP дозволяє зчитувати лише заголовок повідомлення, після чого користувач приймає рішення про те, чи потрібно отримувати сам лист з сервера.

6.5 Протоколи FTP та TFTP

6.5.1 Протокол FTP (File Transfer Protocol). Використовується для передавання файлів. Передавання файлів полягає у копіюванні цілого файлу з одного вузла мережі на інший. Щоб використовувати FTP потрібно мати обліковий запис на сервері або можна використати анонімний FPT (anonymous FTP).

Як і Telnet, протокол FTP був створений для того, щоб можна було працювати між вузлами, які використовують різні операційні системи, різну структуру файлів. FTP згладжує різницю між системами за рахунок підтримання обмеженої кількості різних типів файлів (ASCII, двійкові та ін.) і структури файлів. Офіційну специфікація FTP наведено у RFC 959.

FTP відрізняється від інших застосувань тим, що він використовує два TCP-з'єднання для передавання файлу:

1. Керівне з'єднання встановлюється як звичайне з'єднання клієнт-сервер. Сервер здійснює пасивне відкриття заздалегідь відомого порту FTP (21) і очікує запит на з'єднання від клієнта. Клієнт здійснює активне відкриття на TCP-порт 21, щоб встановити керівне з'єднання. Керівне з'єднання триває весь час, поки клієнт спілкується з сервером. Це з'єднання використовується для передавання команд від клієнта до сервера і для передавання відповідей від сервера. Тип IP-сервісу для керівного з'єднання встановлюється для отримання «мінімальної затримки», оскільки команди, зазвичай, вводяться користувачем;

2. З’єднання даних відкривається кожного разу, коли здійснюється передавання файлу між клієнтом і сервером. Тип сервісу IP для поєднання даних має мати «максимальну пропускну спроможність», оскільки це з’єднання використовується для передавання файлів.

На рис. 6.10 показано підтримання сесії між клієнтом і сервером по двох з’єднаннях. Із рисунка видно, що клієнт не бачить команди і відповіді, що передаються через керівне з’єднання. Ці деталі реалізують два інтерпретатори протоколу.

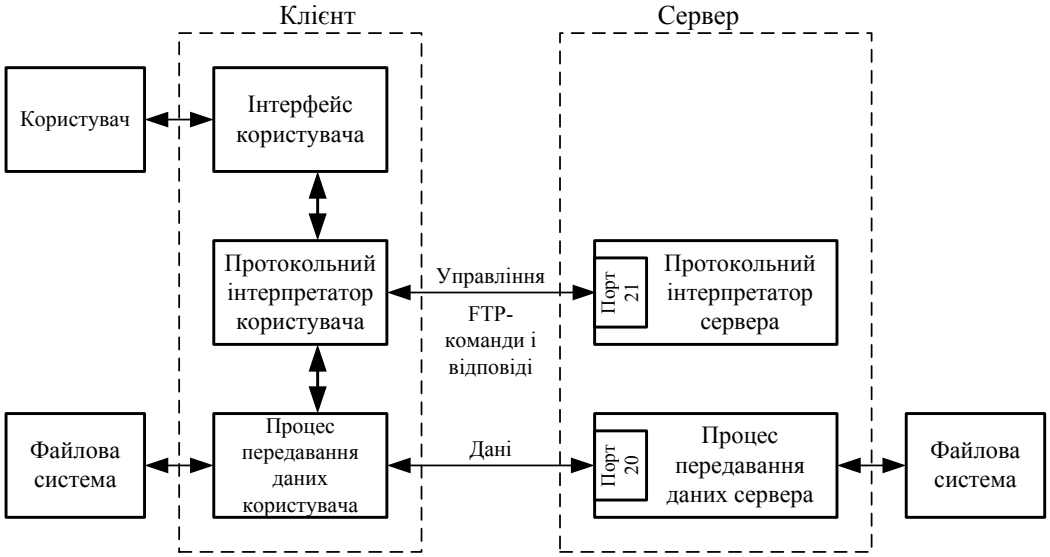


Рисунок 6.10 – Встановлене FTP-з’єднання

Алгоритм роботи протоколу FTP полягає у нижченаведеному.

1. Сервер FTP використовує як керівне з’єднання на TCP порт 21, який завжди знаходиться в стані очікування з’єднання з боку користувача FTP.

2. Після того, як встановлюється керівне з’єднання модуля «Інтерпретатор протоколу користувача» з модулем сервера «Інтерпретатор протоколу сервера», користувач (клієнт) може відправляти на сервер команди. FTP-команди визначають параметри з’єднання: ролі учасників з’єднання (активні чи пасивні), порт з’єднання, тип передавання, тип передаваних даних, структуру даних і керівні директиви, що позначають дії, які користувач хоче зробити (наприклад, зберегти, зчитати, додати або видалити та ін.).

3. Після того, як узгоджені всі параметри каналу передавання даних, один із учасників з’єднання, який є пасивним, переходить в режим очікування відкриття з’єднання на заданий для передавання даних порт. Після цього активний модуль відкриває з’єднання і починає передавання даних.

4. Після закінчення передавання даних з’єднання між «Програмою передавання даних сервера» і «Програмою передавання даних користувача» закривається, але керівне з’єднання «Інтерпретатора протоколу сер-

вера» і «Інтерпретатора протоколу користувача» залишається відкритим. Користувач, не закриваючи сесії FTP, може ще раз відкрити канал передавання даних.

Можлива ситуація, коли дані можуть передаватися на третій комп'ютер. У цьому випадку користувач організовує канал керування з двома серверами і прямий канал даних між ними. Команди керування йдуть через користувача, а дані – напряму між серверами. Канал керування має бути відкритим при передаванні даних між серверами. Інакше, в разі його закриття, передавання даних припиняється.

6.5.2 Подання даних у FTP. Протокол надає різні способи керування передаванням та зберіганням файлів. Потрібно зробити вибір за чотирма пунктами:

1. Тип файлу:

- ASCII-файли (за замовчуванням). Текстовий файл передається як NVT ASCII. При цьому потрібно, щоб відправник конвертував локальний текстовий файл в NVT ASCII, а отримувач конвертував NVT ASCII в текстовий файл. Кінець кожного рядка передається у вигляді NVT ASCII-символа повернення каретки, після чого йде переклад рядка. Це означає, що отримувач має переглядати кожен байт в пошуках пари символів CR, LF;
- EBCDIC-файли. Альтернативний спосіб передавання текстових файлів, коли на обох кінцях системи EBCDIC;
- двійкові або бінарні файли. Дані передаються як неперервний потік бітів;
- локальний тип файлів. Спосіб передавання бінарних файлів між хостами, що мають різний розмір байта. Кількість бітів у байті визначається відправником. Для систем, що використовують 8-бітові байти, локальний тип файлу з розміром байта, що дорівнює 8, еквівалентний бінарним типам файлу.

2. Керування форматом. Застосовується тільки для ASCII- і EBCDIC-файлів:

- Nonprint (за замовчуванням). Файл не містить інформацію вертикального формату;
- Telnet format control. Файл містить керівні символи вертикального формату Telnet, що інтерпретуються принтером;
- Fortran carriage control. Перший символ кожного рядка – це Fortran-символ керування форматом.

3. Структура:

- структура файлу (за замовчуванням). Файл сприймається у вигляді неперервного потоку байтів. Файл не має внутрішньої структури;
- т структура запису. Ця структура використовується тільки для текстових файлів (ASCII або EBCDIC);
- структура сторінки. Кожна сторінка передається зі своїм номером, що дозволяє отримувачеві зберігати сторінки у випадковому порядку.

4. Режим передавання. Вказує на те, як файл передається через з'єднання:

- режим потоку (за замовчуванням). Файл передається як потік байтів. Для файлової структури кінець файлу вказує на те, що відправник закриває з'єднання даних. Для структури запису спеціальна 2-байтова послідовність позначає кінець запису і кінець файлу;
- режим блоків. Файл передається як послідовність блоків, перед кожним з них стоїть один або кілька байтів заголовків;
- стиснутий режим. Просте кодування байтів, що неодноразово зустрічаються і повторюються. У текстових файлах зазвичай стискаються порожні рядки або рядки з пропусками, а в бінарних – рядки з нульових байтів.

6.5.3 Команди та відповіді FTP. Команди та відповіді передаються через керівне з'єднання між клієнтом та сервером у форматі NVT ASCII. У кінці кожного рядка команди або відгуку присутня пара CR, LF. Команди складаються з 3 або 4 байтів, а саме: з ASCII-символів верхнього регістра, деякі з них мають необов'язкові аргументи. Клієнт може відправити серверу більше 30 різних FTP-команд. У табл. 6.8 наведено деякі найбільш вживані команди. Тут <SP> – пропуск; всі команди закінчуються натисканням кнопки введення, в квадратних дужках вказано опційні аргументи, виконання будь-якої команди можна перервати з допомогою комбінації клавіш Ctrl+C.

Таблиця 6.8 – Команди FTP та їх призначення

Команда	Опис
ABOR	Перервати попередню команду FTP і будь-яке передавання даних
LIST [<SP> <прохід>]	Список файлів або директорій.
PASS <SP> <пароль>	Пароль на сервері
PORT <SP> n1, n2, n3, n4, n5, n6	IP-адреса клієнта (n1.n2.n3.n4) і 16-бітовий номер порту, що формується з двох цифр і визначається як n5*256+n6
QUIT	Закрити сесію на сервері
RETR <SP> <ім'я файлу>	Отримати (get) файл
STOR <SP> <ім'я файлу>	Зберегти (put) файл
SYST	Сервер повертає тип системи
TYPE<SP> <тип>	Вказати тип файлу: A для ASCII, I для двійкового
USER<SP> <ім'я користувача>	Вказати ім'я користувача на сервері

FTP-відповіді складаються з 3-цифрових значень у форматі ASCII і необов'язкових повідомлень, що йдуть за символами. Цифрові коди використовують програми, а текстові пояснення – користувачі. Кожна з трьох цифр в коді відповіді має власний зміст. У табл. 6.9 наведено деякі значен-

ня першої та другої цифр в коді відгуку. Третя цифра дає додаткове пояснення повідомленням про помилку.

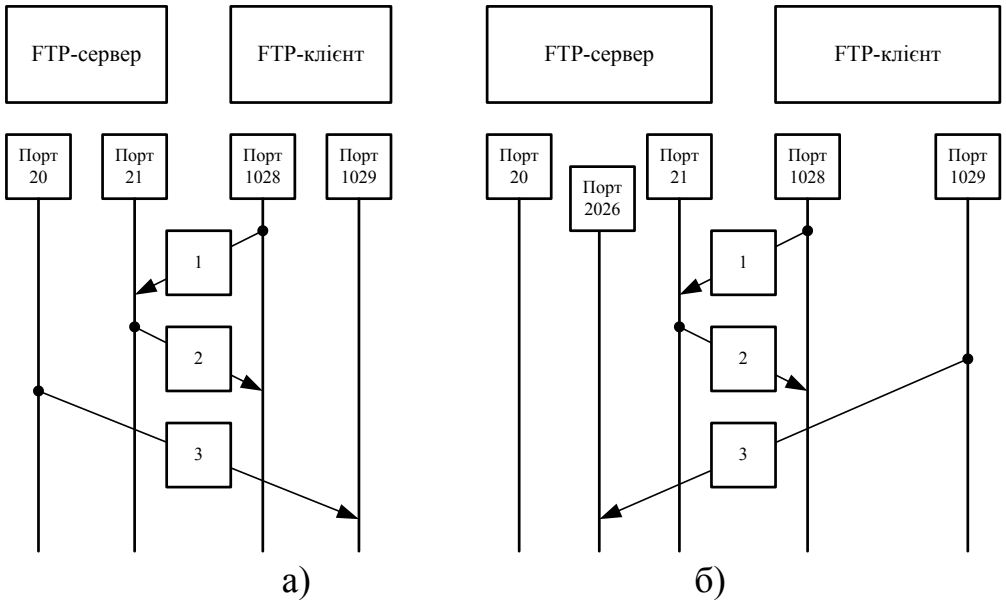
Таблиця 6.9 – Коди FTP відповідей FTP та їх опис

Відповідь	Опис
1yz	Позитивна попередня відповідь. Дія почалася, однак потрібно дочекатися ще однієї відповіді перед відправленням наступної команди.
2yz	Позитивна відповідь про завершення. Може бути відправлена нова команда.
3yz	Позитивна проміжна відповідь. Команда прийнята, однак необхідно відправити ще одну команду.
4yz	Тимчасова негативна відповідь про завершення. Потрібна дія не відбулася, проте помилка тимчасова, команду потрібно повторити пізніше.
5yz	Постійна негативна відповідь про завершення. Команда не була сприйнята і повторювати її не варто.
x0z	Синтаксична помилка.
x1z	Інформація.
x2z	З'єднання. Відповіді стосуються процесу з'єднання або керування.
x3z	Аутентифікація. Відповідь стосується реєстрації або команди, що пов'язана з обліковим записом.

6.5.4 Режими роботи FTP-сервера:

1. Активний режим (показано на рис. 6.10, а):

- клієнт встановлює з'єднання на 21 порт сервера з порту N (N>1024);
- сервер відправляє відповідь на порт N клієнта;
- сервер встановлює з'єднання для передавання даних з порту 20 на порт клієнта N+1;



а) б)

Рисунок 6.10 – Режими робота FTP-сервера:
а) активний режим; б) пасивний режим

2. Пасивний режим (показано на рис. 6.10, б):

- клієнт встановлює з'єднання на 21 порт сервера з порту N (N>1024) і просить сервер перейти в пасивний режим;
- сервер відправляє відповідь і повідомляє номер порту для каналу даних P (P>1024) на порт N клієнта;
- клієнт встановлює з'єднання для передавання даних з порту N+1 на порт сервера P.

Робота FTP на користувацькому рівні містить кілька етапів.

1. Ідентифікація (введення імені-ідентифікатора і пароля).
2. Вибір каталогу.
3. Визначення режиму обміну (поблоково, потоковий, ASCII або двійковий).
4. Виконання команд обміну (get, mget, dir, mdel, mput або put).
5. Завершення процедури (quit або close).

Незважаючи на розповсюдженість, у FTP є багато недоліків:

- програми-клієнти FTP не завжди зручні і прості у користуванні;
- користувач не завжди може зрозуміти який файл перед ним: той, що необхідно, чи ні;
- не існує простого та універсального засобу для пошуку на серверах anonymous FTP;
- програми FTP доволі старі, деякі їхні особливості, які були потрібні в часи їхнього створення, не зовсім зрозумілі і потрібні зараз. Наприклад, для передавання файлів існує два режими – двійковий та текстовий, і якщо користувач неправильно обрав режим передавання, то файл, який необхідно передати, може бути пошкодженим.
- опис файлів на сервері видається у форматі операційної системи сервера;
- сервери FTP нецентралізовані.

6.5.5 Протокол TFTP (Trivial File Transfer Protocol). Це простий протокол передавання файлів. Він, як правило, використовується при завантаженні бездискових систем. На відміну від протоколу FTP, що використовує як транспортний протокол TCP, TFTP використовує UDP. Це зроблено для того, щоб протокол був якомога простішим. Опис TFTP версії 2 можна знайти в RFC 1350.

Обмін між клієнтом і сервером починається з того, що клієнт відправляє запит на запис або читання сервера. У стандартному варіанті завантаження бездискової системи перший запит – це запит на читання (RRQ). На рис. 6.11 показано формат кількох повідомлень TFTP (opcode – код операції, коди операцій 1 і 2 мають однаковий формат).

Перші 2 байти TFTP-повідомлення – це код операції. У запиті на читання (RRQ) і в запиті на запис (WRQ) ім'я файлу (filename) вказує файл на сервері, який клієнт хоче або зчитати, або записати. Ім'я файлу закінчується нульовим байтом. Режим (mode) – це ASCII-рядок: netascii або octet (будь-яка комбінація великих або маленьких літер), який також закінчується байтом 0. netascii означає, що дані є рядками ASCII-тексту, причому ко-

жний рядок закінчується 2-символьною послідовністю повернення каретки, за якою слідує пропуск рядка (позначається CR/LF).

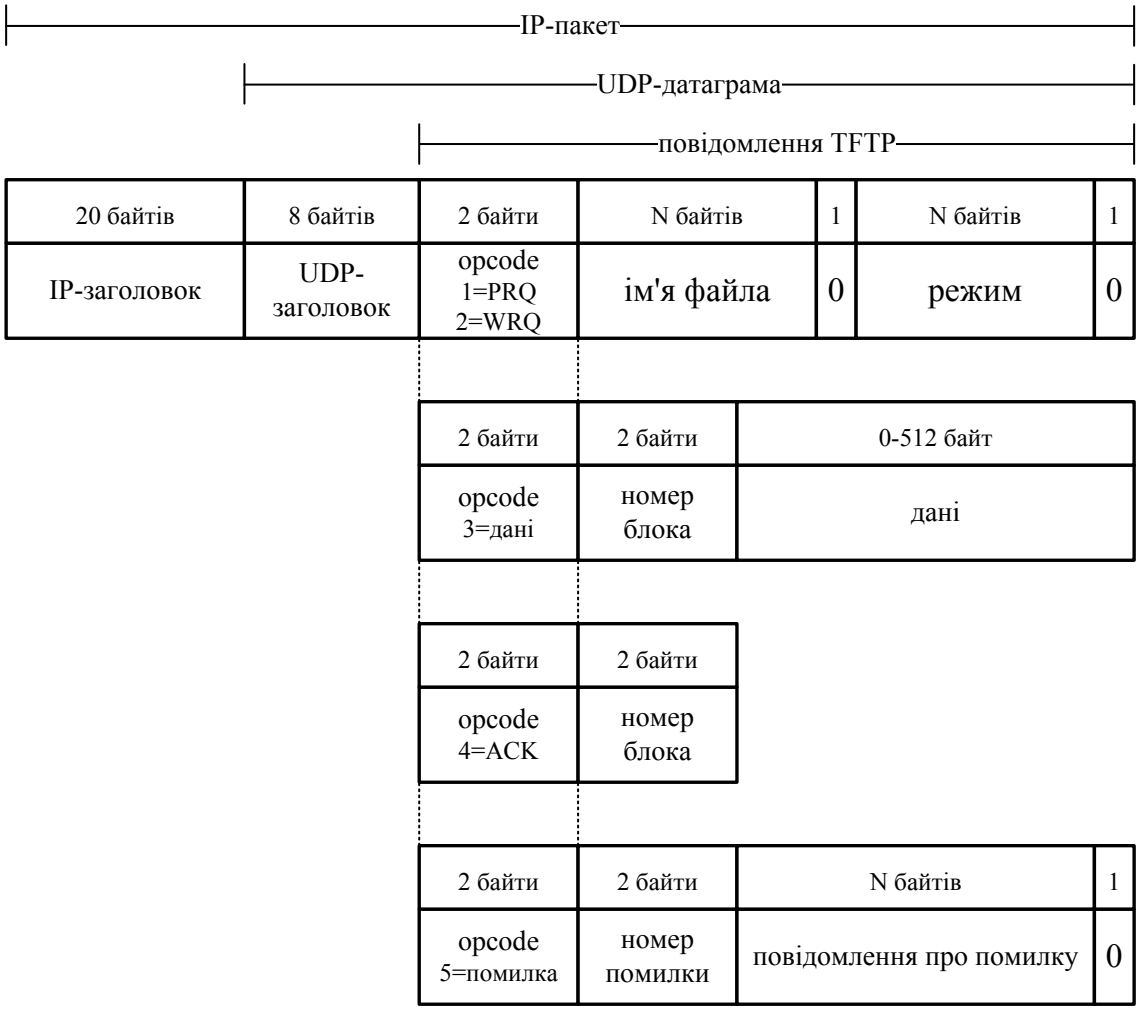


Рисунок 6.11 – Формат повідомлень TFTP

І клієнт, і сервер мають мати можливість здійснити перетворення між цим форматом і яким-небудь іншим, що використовується на локальному хості. Передача octet означає, що дані будуть передаватися у вигляді 8-бітових байтів без інтерпретації.

Кожен пакет даних містить номер блока (block number), який потім використовується в пакеті підтвердженні. Наприклад, коли потрібно здійснити читання файлу, клієнт посилає запит на читання (RRQ), вказуючи ім’я файлу і режим. Якщо файл може бути прочитаний клієнтом, сервер відповідає пакетом даних з номером блока, що дорівнює 1. Клієнт посилає підтвердження (ACK) на номер блока 1. Сервер відповідає наступним пакетом даних з номером блока, що дорівнює 2. Клієнт підтверджує номер блока 2. Це продовжується до тих пір, поки файл не буде переданий. Кожен пакет даних містить 512 байтів даних, за винятком останнього пакета, який містить від 0 до 511 байтів даних. Коли клієнт отримує пакет даних, який містить менше ніж 512 байтів, він вважає, що отримав останній пакет.

У разі запиту на запис (WRQ) клієнт посилає WRQ, вказуючи ім'я файлу і режим. Якщо файл може бути записаний клієнтом, сервер відповідає підтвердженням (ACK) з номером блока, що дорівнює 0. Клієнт посилає перші 512 байтів файлу з номером блока, що дорівнює 1, сервер відповідає ACK з номером блока, що дорівнює 1.

Такий тип передавання даних називається протоколом із зупинкою та очікуванням підтвердження (stop-and-wait). Він використовується тільки в таких простих протоколах, як TFTP.

Останній тип TFTP-повідомлень – це повідомлення про помилки, код операції (opcode) дорівнює 5. Це якраз те, чим сервер відповідає в тому випадку, якщо запит на читання або запис не може бути оброблений. Помилки читання або запису під час передавання файлу також призводять до того, що відправляється повідомлення про помилку, при цьому передавання припиняється. Номер помилки (error number) містить цифровий код помилки, за яким слідує повідомлення про помилку в ASCII-форматі, яке може містити додаткову інформацію, надану операційною системою.

Оскільки TFTP використовує ненадійний UDP, то саме від TFTP залежить, як будуть оброблені втрачені і дубльовані пакети. У разі втрати пакета відправник відпрацьовує тайм-аут і здійснює повторне передавання. Можлива поява проблеми, що називається «синдром новачка», яка може виникнути, якщо з обох сторін буде відправлено тайм-аут і здійснено повторне передавання. Як і в більшості UDP-програм, контрольна сума TFTP-повідомлення не розраховується, а це означає, що будь-яке пошкодження даних може бути визначено тільки за допомогою контрольної суми UDP.

TFTP-пакети не містять ніяких даних про ім'я користувача або пароль. Тобто, даний протокол не використовує ніякого захисту. Він був розроблений для використання в процесі завантаження.

Для додаткової безпеки, наприклад, на Unix-системі TFTP-сервер встановлює такі значення в ідентифікатор користувача (UID) і ідентифікатор групи (GID), які не можуть бути призначені реальному користувачеві. Це дозволяє доступ тільки до файлів, що доступні для читання і запису всім.

6.6 Протокол HTTP

HTTP (HyperText Transfer Protocol) – це символьно-орієнтований клієнт-серверний протокол прикладного рівня без збереження стану, що використовується сервісом WWW (World Wide Web). Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URI (Uniform Resource Identifier – унікальний ідентифікатор ресурсу) в запиті клієнта. URI може вказуватися як місцезнаходження ресурсу (Universal Resource Locator, URL) або універсальне ім'я (Universal Resource Name, URN). Основними ресурсами є файли, що зберігаються на сервері, але ними можуть бути й інші логічні (каталог на сервері) або абстрактні об'єкти (ISBN). Протокол HTTP дозволяє вказати спосіб подання (кодування) одного і того ж ресур-

су за різними параметрами: mime-типу, мови і т. д. Завдяки цій можливості клієнт і web-сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим.

Структура протоколу визначає, що кожне HTTP-повідомлення складається з трьох частин, що передаються у такому порядку:

1. Стартовий рядок (starting line) визначає тип повідомлення;
2. Заголовки (headers) характеризують тіло повідомлення, параметри передавання та інші відомості;
3. Тіло повідомлення (message body) – безпосередньо дані повідомлення. Обов'язково має відділятися від заголовків порожнім рядком.

Стартовий рядок є обов'язковим елементом, оскільки вказує на тип запиту/відповіді, заголовки і тіло повідомлення можуть бути відсутні. Стартовий рядок відрізняється для запиту і відповіді. Рядок запиту має такий вигляд:

Метод URI HTTP/Версія протоколу.

Приклад запиту

GET /web-programming/index.html HTTP/1.1

Стартовий рядок відповіді сервера має такий формат:

HTTP/Версія Код Стану [Пояснення]

Наприклад, на попередній запит клієнтом даної сторінки сервер відповідає таким рядком:

HTTP/1.1 200 Ok

Метод HTTP – це послідовність із будь-яких символів, крім керівних і розділювачів, що вказує на основну операцію над ресурсом. Як правило, метод являє собою коротке англійське слово, що записується великими літерами. Основні методи HTTP та їх пояснення наведено у табл. 6.10. Назва методу чутлива до регістра.

Таблиця 6.10 – Основні методи HTTP

Назва методу	Опис методу
OPTIONS	Використовується для визначення можливостей web-сервера або параметрів з'єднання для конкретного ресурсу. Передбачається, що запит клієнта може містити тіло повідомлення для визначення відомостей, що його цікавлять. Формат тіла і порядок роботи з ним у даний момент не визначені. Сервер поки має його ігнорувати. Аналогічна ситуація і з тілом у відповіді сервера. Для того, щоб дізнатися можливості всього сервера, клієнт має вказати в URI зірочку «*». Запити «OPTIONS * HTTP/1.1» можуть також застосовуватися для перевірки працездатності сервера (аналогічно «пінгуванню») і тестування підтримки сервером протоколу HTTP версії 1.1. Результат виконання цього методу не кешується.
GET	Використовується для запиту вмісту зазначеного ресурсу. За допомогою методу GET можна також розпочати будь-який процес. У цьому випадку в тіло відповідного повідомлення слід вносити інформацію про хід виконання процесу. Клієнт може передавати параметри виконання запиту в URI цільового ресурсу після символу «?»:

	GET / path / resource? Param1 = value1 m2 = value2 HTTP/1.1. Відповідно до стандарту HTTP, запити типу GET вважаються ідемпотентними – багаторазове повторення одного і того ж запиту GET має приводити до однакових результатів (за умови, що сам ресурс не змінився). Це дозволяє кешувати відповіді на запити GET. Крім звичайного методу GET, розрізняють ще умовний GET і частковий GET. Умовні запити GET містять заголовки If-Modified-Since, If-Match, If-Range і подібні. Часткові GET містять у запиті Range. Порядок виконання подібних запитів визначено стандартами окремо.
HEAD	Аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло. Запит HEAD звичайно застосовується для вилучення метаданих, перевірки наявності ресурсу (валідація URL) і щоб дізнатися, чи не змінився він з моменту останнього звертання. Заголовки відповіді можуть кешуватися. При розбіжності метаданих ресурсу з відповідною інформацією в кеші копія ресурсу позначається як застаріла.
POST	Застосовується для передавання даних користувача заданому ресурсу. Наприклад, якщо на html-сторінці реалізовано блок для введення коментарів у відповідну форму, то введені дані передаються серверу методом POST, і він поміщає їх на сторінку. При цьому передані дані (у прикладі це текст коментаря) вносяться у тіло запиту. Аналогічно за допомогою методу POST завантажуються файли. На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багаторазове повторення одних і тих же запитів POST може повертати різні результати. При результаті виконання 200 (Ok) і 204 (No Content) у тіло відповіді слід помістити повідомлення про результат виконання запиту. Якщо був створений ресурс, то серверу слід повернути відповідь 201 (Created) із зазначенням URI нового ресурсу в заголовку Location. Повідомлення відповіді сервера на виконання методу POST не кешується.
PUT	Застосовується для завантаження вмісту запиту на вказаний у запиті URI. Якщо за заданим URI не існувало ресурсу, то сервер створює його і повертає статус 201 (Created). Якщо ж було змінено ресурс, то сервер повертає 200 (Ok) або 204 (No Content). Сервер не має ігнорувати некоректні заголовки Content-*, що передаються клієнтом разом з повідомленням. Якщо якийсь з цих заголовків не може бути розпізнаний або неприпустимий при поточних умовах, то необхідно повернути код помилки 501 (Not Implemented). Фундаментальна відмінність методів POST і PUT полягає в розумінні призначень URI-ресурсів. Метод POST припускає, що за вказаним URI буде проводитися обробка переданого клієнтом вмісту. Використовуючи PUT, клієнт припускає, що завантажені дані відповідають ресурсу, що знаходиться за даним URI. Повідомлення відповідей сервера на метод PUT не кешуються.
PATCH	Аналогічно PUT, але застосовується тільки до фрагмента ресурсу.
DELETE	Знищує вказаний ресурс.
TRACE	Повертає отриманий запит так, що клієнт може побачити, що проміжні сервера додають або змінюють у запиті.
LINK	Встановлює зв’язок зазначеного ресурсу з іншими.
UNLINK	Знищує зв’язок зазначеного ресурсу з іншими.

Кожен сервер зобов'язаний підтримувати, як мінімум, методи GET і HEAD. Якщо сервер не розпізнав зазначений клієнтом метод, то він має повернути статус 501 (Not Implemented). Якщо серверу метод відомий, але він не застосовується до конкретного ресурсу, то повертається повідомлення з кодом 405 (Method Not Allowed). В обох випадках серверу слід внести в повідомлення відповіді заголовок Allow зі списком методів, що підтримуються.

Код стану інформує клієнта про результати виконання запиту і визначає його подальшу поведінку. Набір кодів стану є стандартним, і всі вони описані у відповідних документах RFC. Кожен код подається цілим тризначним числом. Перша цифра вказує на клас стану, наступні – порядковий номер стану (рис. 6.12). За кодом відповіді, зазвичай, слідує короткий опис англійською мовою. Класи кодів стану, що використовуються сьогодні, та приклади відповідей сервера наведено у табл. 6.11.

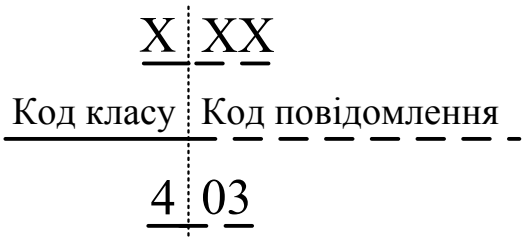


Рисунок 6.12 – Структура коду стану HTTP

Таблиця 6.11 – Коди стану протоколу HTTP

Клас коду	Опис коду
1xx Informational (інформаційний)	У цей клас виділені коди, що інформують про процес передавання. В HTTP/1.0 повідомлення з такими кодами мають ігноруватися. В HTTP/1.1 клієнт має бути готовий прийняти цей клас повідомлень як звичайну відповідь, але нічого відправляти серверу не потрібно. Самі повідомлення від сервера містять тільки стартовий рядок відповіді і, якщо потрібно, декілька специфічних для відповіді полів заголовка. Приклади відповідей сервера: 100 Continue (продовжувати); 101 Switching Protocols (перемикання протоколів); 102 Processing (іде опрацювання).
2xx Success (успішно)	Повідомлення даного класу інформують про випадки успішного приймання та обробки запиту клієнта. Залежно від статусу сервер може ще передати заголовки і тіло повідомлення. Приклади відповідей сервера: 200 OK (успішно); 201 Created (створено); 202 Accepted (прийнято); 204 No Content (нема вмісту); 206 Partial Content (неповний вміст).

Продовження табл. 6.11

3xx Redirection (перенаправлення)	Повідомляють клієнтові, що для успішного виконання операції потрібно здійснити наступний запит до іншого URI. У більшості випадків нова адреса вказується в полі Location-заголовка. Клієнт у цьому випадку має, як правило, зробити автоматичний перехід («редірект»). При зверненні до наступного ресурсу можна отримати відповідь з цього ж класу кодів. Може вийти навіть довгий ланцюжок з перенаправлень. Тому розробники протоколу HTTP рекомендують після другої такої відповіді обов'язково запитувати підтвердження на перенаправлення у користувача (раніше рекомендувалося після 5-го). Клієнт має запобігати потраплянню в колові перенаправлення, оскільки поточний сервер може перенаправити клієнта на ресурс іншого сервера.. Приклади відповідей сервера: 300 Multiple Choices (множинний вибір); 301 Moved Permanently (переміщена назавжди); 304 Not Modified (не змінювалося).
4xx Client Error (помилка клієнта)	Призначені для вказання на помилки з боку клієнта. При використанні всіх методів, крім HEAD, сервер має повернути у тілі повідомлення гіпертекстове пояснення для користувача. Приклади відповідей сервера: 401 Unauthorized (неавторизований); 402 Payment Required (потрібна оплата); 403 Forbidden (заборонено); 404 Not Found (не знайдено); 405 Method Not Allowed (метод не підтримується); 406 Not Acceptable (не прийнятно); 407 Proxy Authentication Required (потрібна аутентифікація проксі).
5xx Server Error (помилка сервера)	Використовуються у випадках, коли невдало виконана операції з вини сервера. Для всіх ситуацій, крім використання методу HEAD, сервер має вносити у тіло повідомлення пояснення, яке клієнт відобразить користувачеві. Приклади відповідей сервера: 500 Internal Server Error (внутрішня помилка сервера); 502 Bad Gateway (поганий шлюз); 503 Service Unavailable (сервіс недоступний); 504 Gateway Timeout (шлюз не відповідає).

Заголовок HTTP (HTTP Header) – це рядок в HTTP-повідомленні, що містить розділену двокрапкою пару виду «параметр-значення». Формат заголовка відповідає загальному формату заголовків текстових мережних повідомлень ARPA (RFC 822). Як правило, браузер і web-сервер містять у повідомленнях більше одного заголовка. Заголовки мають відправлятися раніше тіла повідомлення і відокремлюватися від нього хоча б одним пустим рядком (CRLF).

Назва параметра має складатися мінімум з одного друкованого символа (ASCII-коди від 33 до 126). Після назви відразу має слідувати символ двокра-

пки. Значення може містити будь-які символи ASCII, крім переведення рядка (CR, код 10) і повернення каретки (LF, код 13). Символи пропусків на початку і наприкінці значення обрізаються. Послідовність декількох пропусків всередині значення може сприйматися як один пропуск. Регістр символів у назві не має значення (якщо інше не передбачено форматом поля).

Усі HTTP-заголовки поділяються на чотири основні групи.

1. General Headers (основні заголовки) мають вноситися в будь-яке повідомлення клієнта і сервера.
2. Request Headers (заголовки запиту) використовуються тільки в запитах клієнта.
3. Response Headers (заголовки відповіді) присутні тільки у відповідях сервера.
4. Entity Headers (заголовки сутності) супроводжують кожну сутність повідомлення.

Сутності – це корисна інформація, що передається у запиті або відповіді. Сутність складається з метаінформації (заголовки) і безпосередньо вмісту (тіло повідомлення). Сутності виділено в окремий клас заголовків, щоб не плутати їх із заголовками запиту або заголовками відповіді при передаванні множинного вмісту (multipart / *). Заголовки запиту і відповіді, як і основні заголовки, описують все повідомлення в цілому і розміщуються тільки в початковому блоці заголовків. У той час як заголовки сутності характеризують вміст кожної частини окремо, розташовуючись безпосередньо перед її тілом. У табл. 6.12 наведено опис деяких HTTP-заголовків.

Таблиця 6.12 – Заголовки HTTP

Заголовок	Група	Опис
Allow	Entity	Список методів, що застосовуються до ресурсу, до якого йде звертання.
Content-Encoding	Entity	Застосовується при необхідності перекодування вмісту (наприклад, gzip/deflated).
Content-Language	Entity	Локалізація вмісту (мова (и)).
Content-Length	Entity	Розмір тіла повідомлення (в октетах).
Content-Range	Entity	Діапазон (використовується для підтримки багатопотокового завантаження чи дозавантаження).
Content-Type	Entity	Вказує тип вмісту (mime-type, наприклад text/html). Часто містить вказання на таблицю локальних символів (charset).
Expires	Entity	Дата/час, після якої ресурс вважається застарілим. Використовується проксі-серверами.
Last-Modified	Entity	Дата/час останньої модифікації сутності.
Cache-Control	General	Визначає директиви керування механізмами кешування. Для проксі-серверів.
Connection	General	Задає параметри, необхідні для конкретного з’єднання.
Date	General	Дата і час формування повідомлення.

Продовження табл. 6.12

Pragma	General	Використовується для спеціальних вказівок, що можуть (опційно) застосовуватися до будь-якого одержувача з усього ланцюжка запитів/відповідей (наприклад, pragma: no-cache).
Transfer-Encoding	General	Задає тип перетворення, що застосовується до тіла повідомлення. На відміну від Content-Encoding цей заголовок поширюється на всі повідомлення, а не тільки на сутність.
Via	General	Використовується шлюзами і проксі для відображення проміжних протоколів і вузлів між клієнтом і web-сервером.
Warning	General	Додаткова інформація про поточний статус, яка не може бути подана в повідомленні.
Accept	Request	Визначає застосовані типи даних, що очікуються у відповіді.
Accept-Charset	Request	Визначає кодування символів (charset) для даних, що очікуються у відповіді.
Accept-Encoding	Request	Визначає формати кодування/декодування вмісту (наприклад, gzip), що використовуються.
Accept-Language	Request	Відповідні мови. Використовується для узгодження передачі.
Authorization	Request	Облікові дані клієнта, що запитує ресурс.
From	Request	Електронна адреса відправника.
Host	Request	Ім’я/мережна адреса [і порт] сервера. Якщо порт не вказаний, використовується 80.
If-Modified-Since	Request	Використовується для виконання умовних методів. Якщо запитуваний ресурс змінився, то він передається з сервера, інакше – з кешу.
Max-Forwards	Request	Подає механізми обмеження кількості шлюзів і проксі при використанні методів TRACE і OPTIONS.
Proxy-Authorization	Request	Використовується при запитах, що проходять через проксі та вимагають авторизації.
Referer	Request	Адреса, з якої виконується запит. Цей заголовок відсутній, якщо перехід виконується з адресного рядка або, наприклад, за посиланням з js-скрипта.
User-Agent	Request	Інформація про користувача агента (клієнта).
Location	Response	Адреса перенаправлення.
Proxy-Authenticate	Response	Повідомлення про статус з кодом 407.
Server	Response	Інформація про програмне забезпечення сервера, що відповідає на запит (це може бути як web-, так і проксі-сервер).

Тіло HTTP-повідомлення (message-body), якщо воно є, використовується для передавання сутності, що пов’язана з запитом або відповіддю. Тіло повідомлення (message-body) відрізняється від тіла сутності (entity-

body) тільки в тому випадку, коли при передаванні застосовується кодування, вказане в заголовку Transfer-Encoding. В інших випадках тіло повідомлення ідентично тілу сутності.

Заголовок Transfer-Encoding має відправлятися для вказання на будь-яке кодування передачі, що використовується з метою гарантування безпечного й правильного передавання повідомлення. Transfer-Encoding – це властивість повідомлення, а не сутності, і воно може бути додано або видалено будь-яким застосуванням у ланцюжку запитів/відповідей.

Присутність тіла повідомлення у запиті зазначається додаванням до заголовків запиту поля заголовка Content-Length або Transfer-Encoding. Тіло повідомлення (message-body) може бути додано до запиту лише тоді, коли метод запиту може містити тіло об'єкта (entity-body). Всі відповіді містять тіло повідомлення, можливо нульової довжини, крім відповідей на запит методом HEAD і відповідей з кодами статусу 1xx (інформаційні), 204 (немає вмісту), і 304 (не модифікований).

6.7 Протокол SNMP

Керування TCP/IP-мережами будується на взаємодії між станцією керування мережею (менеджер) і елементами мережі. Елементами мережі можуть бути будь-які об'єкти, що використовують сімейство протоколів TCP/IP: хости, маршрутизатори, термінали, термінальні сервери, принтери і т. д. На елементах мережі має бути запущено програмне забезпечення, що називається агентом.

Обмін даними, як правило, двосторонній: менеджер просить агента повідомити йому певне значення, або агент повідомляє менеджеру про будь-яку важливу подію. У менеджера має бути можливість встановити змінні в агента і зчитувати їх. Для керування мережами TCP/IP потрібні 3 складові:

1. Інформаційна база керування (MIB – Management Information Base), що вказує, які значення елементів мережі необхідно обслуговувати (інформація, яка може бути запрошена і встановлена менеджером);
2. Визначення загальної структури й схеми ідентифікації, що використовується для звертання до змінних у MIB. Це називається структурою інформації керування (SMI – Structure of Management Information);
3. Протокол, що функціонує між менеджером і елементом. Він називається простим протоколом керування мережею (SNMP – Simple Network Management Protocol). RFC 1157 описує цей протокол. Як транспортний протокол можуть бути використані різні протоколи, зазвичай з SNMP використовується UDP.

SNMP визначає лише п'ять типів повідомлень, якими обмінюються менеджер і клієнт:

- get-request (отримати значення однієї або кількох змінних);
- get-next-request (отримати наступну змінну після даної або кілька вказаних змінних);

- set-request (встановити значення однієї або кількох змінних);
- get-response (видати значення однієї або кількох змінних. Це повідомлення повертається агентом менеджеру у відповідь на оператори get-request, get-next-request і set-request);
- trap (повідомити менеджера, якщо щось сталося з агентом).

Перші три повідомлення відправляються від менеджера до агента, а останні два від агента до менеджера. На рис. 6.13 наведено оператори SNMP.

Оскільки повідомлення SNMP реалізуються через послідовність запит-відгук з використання протоколу UDP, то можлива ситуація, коли запит не надходить до агента або відповідь не повертається до менеджера. У цьому випадку менеджер відпрацює тайм-аут і здійснить повторне передавання.

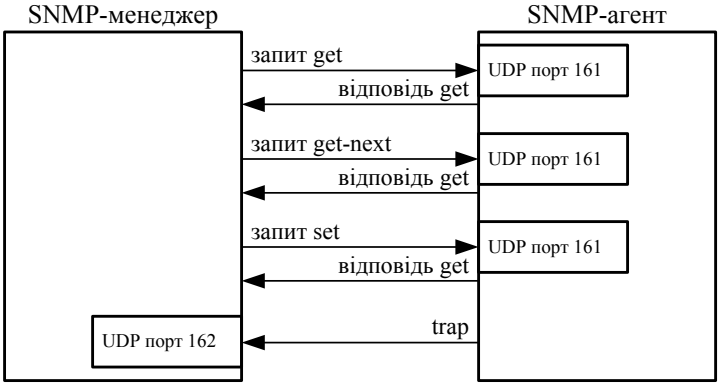


Рисунок 6.13 –Оператори протоколу SNMP

На рис. 6.14 показано формат SNMP повідомлення.

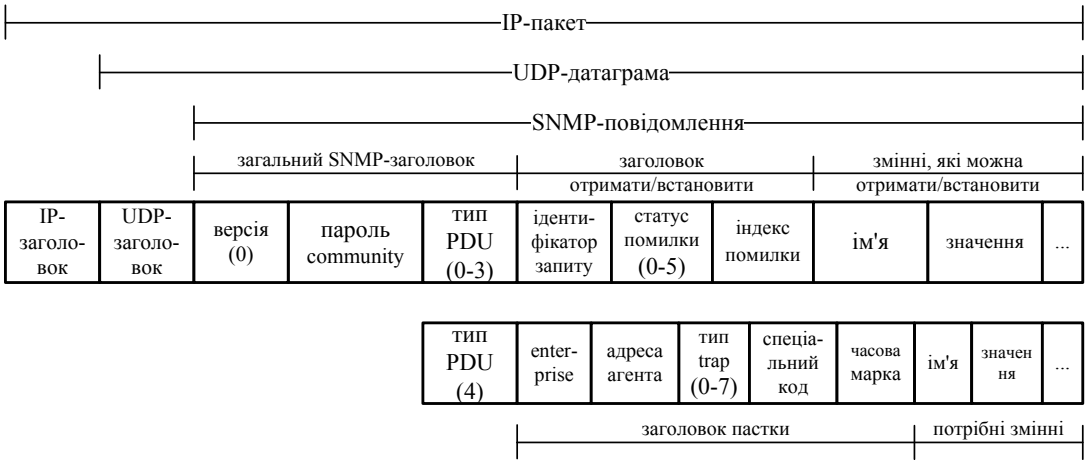


Рисунок 6.14 – Формат повідомлення протоколу SNMP

Значення поля version дорівнює 0. Це значення дорівнює номеру версії протоколу мінус 1.

Поле тип блока даних протоколу (тип PDU) вказує на тип SNMP-повідомлення (табл. 6.13).

Таблиця 6.13 – Типи PDU повідомлень SNMP

Тип PDU	SNMP-повідомлення	Тип PDU	SNMP-повідомлення
0	get-request	3	get-response
1	get-next-request	4	trap
2	set-request		

Пароль (community) – це рядок символів, у якому міститься пароль у відкритому вигляді. Пароль використовується при взаємодії між менеджером і клієнтом. Як правило, це 6-символьний рядок public.

В операторах get, get-next і set менеджер встановлює ідентифікатор запиту (request ID), який повертається агентом у повідомленні get-response. Це дозволяє клієнтові (менеджеру в даному випадку) зіставити відгуки від сервера (агент) з запитами, що були відправлені клієнтом (менеджером). Це поле також дозволяє менеджеру видати декілька запитів одному або кільком агентам, а потім відсортувати отримані відгуки.

Статус помилки (error status) – це ціле число, що повертається агентам і вказує на помилку. У табл. 6.14 наведено значення, імена та опис помилок.

Таблиця 6.14 – Значення статусу помилки SNMP

Статус помилки	Ім’я	Опис
0	noError	Усе в порядку.
1	tooBig	Клієнт не може помістити відповідь в одне SNMP-повідомлення.
2	noSuchName	Оператор вказує на неіснуючу змінну.
3	badValue	В операції встановлення використано недійсне значення або зроблено помилку у синтаксисі.
4	readOnly	Менеджер намагався змінити змінну, що має помітку «лише для читання».
5	genErr	Невідома помилка.

Якщо виникла помилка, індекс помилки (error index) – це ціле зміщення, що вказує на те, в якій змінній відбулася помилка. Це значення встановлюється агентом тільки для помилок noSuchName (нема такого імені), badValue (неправильне значення) і readOnly (тільки для читання).

Список імен змінних і значень знаходиться в get-, get-next- і set-запитах. Розділ значень ігнорується в операторах get і get-next.

Для оператора trap (PDU type дорівнює 4) формат SNMP-повідомлення змінюється.

Для ЕОМ під керуванням UNIX можна використати програму snmpi (SNMP initiator або SNMPWALK, NETGUARD, SNMPMAN). Вона дозволить отримати багато корисної інформації про мережу. Синтаксис звертання до snmpi: snmpi [-a agent] [-c community] [-f file] [-p portno] [-d] [-v] [-w].

Програма snmpi вкрай проста. Опція -a пропонує можливість ввести адресу SNMP-об’єкта: ім’я ЕОМ, IP-адресу або транспортну адресу. Опція -p дозволяє задати номер UDP-порту. За замовчуванням це порт 61.Опція

-c дозволяє задати груповий пароль (community) для SNMP-запиту. За замовчуванням це public, тобто, вільний доступ. Опція -f дозволяє вибрати файл, що містить відкомпільований опис MIB-модулів. За замовчуванням це objects.defs. Опція -w активує режим спостереження, видаючи на термінал всі службові повідомлення. Вихід з програми виконується за командою quit (q). Якщо ви працюєте на ЕОМ, що підключена до локальної мережі, отримайте допуск до одного з інших хостів мережі і приступайте.

6.8 Протокол NFS

NFS (Network File System) надає клієнтам прозорий доступ до файлів і файлової системи сервера. На відміну від FTP, який забезпечує передавання файлів і їх повне копіювання, даний протокол здійснює доступ лише до тих частин файлів, до яких звертається процес. Основна перевага NFS в тому, що він робить цей доступ прозорим, тобто будь-яке застосування клієнта, яке може працювати з локальним файлом, з таким же успіхом може працювати і с NFS-файлом без будь-яких змін для самої програми.

NFS – це клієнт-серверне застосування, що побудоване з використанням Sun RPC (Remote Procedure Call – виклик віддаленої процедури). NFS-клієнт отримує доступ до файлів на NFS-сервері шляхом відправлення RPC-запитів на сервер. Звертання NFS-клієнта здійснюються операційною системою клієнта від імені користувацького процесу клієнта. У свою чергу NFS-сервер реалізовано всередині операційної системи для підвищення ефективності його роботи.

Існує два види Sun RPC. Одна версія побудована з використанням API-сокета і працює з TCP і UDP. Інша називається TI-RPC (Transport Independent) і побудована з використанням TLI API, працює з будь-якими транспортним рівнями. Формат повідомлення виклику процедури RPC з використанням UDP показано на рис. 6.15.

Ідентифікатор транзакції (XID) встановлюється клієнтом і повертається сервером. Коли клієнт отримує відгук, він порівнює XID сервера і власний. Якщо вони не збігаються, то клієнт відкидає повідомлення і чекає надходження наступного. Кожного разу, коли клієнт видає новий RPC, він змінює XID. Але якщо клієнт передає RPC повторно (якщо відгук не було отримано), XID не змінюється.

Змінна call дорівнює 0 для запиту і 1 для відповіді. Поточна версія RPC (RPC-version) дорівнює 2, три наступні змінні – це номер програми (program number), номер версії (version number) і номер процедури (procedure number) – ідентифікують конкретну процедуру, яка має бути викликана на сервері.

Повноваження (credentials) ідентифікують клієнта. Перевірка (verifier) використовується для захищеного RPC (Secure RPC), яке використовує DES-шифрування. Незважаючи на те, що поля повноваження і перевірки змінної довжини, їх довжина передається як частина поля.

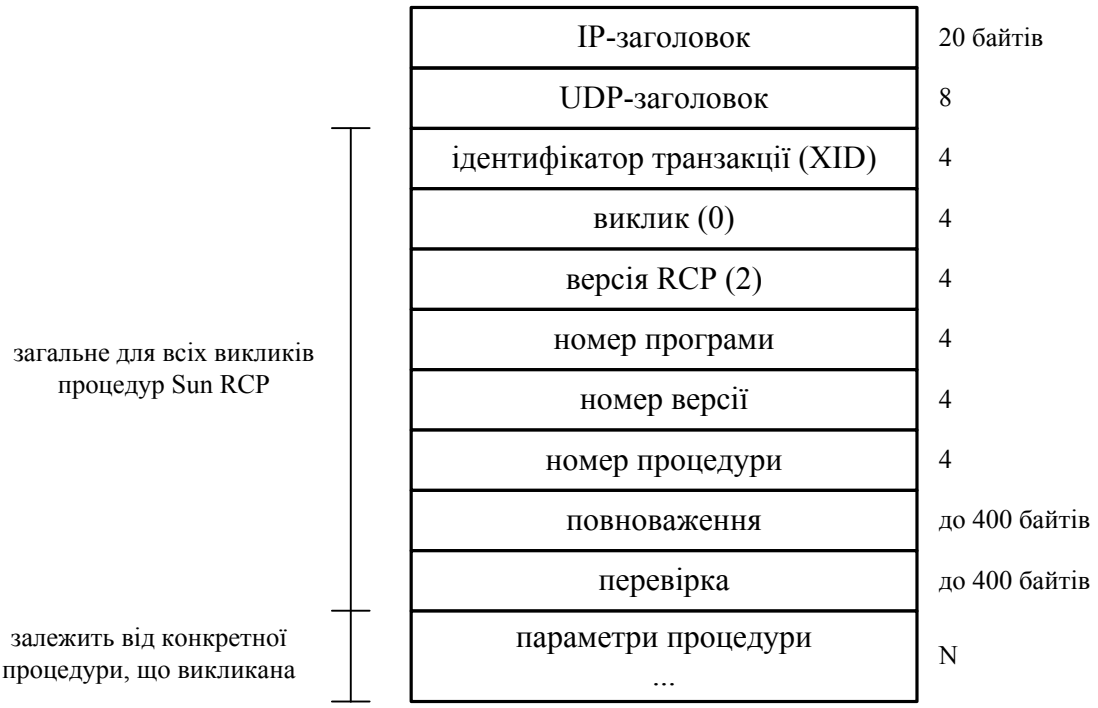


Рисунок 6.15 – Повідомлення виклику процедури RPC у форматі UDP-дейтаграми

Далі йдуть параметри процедури. Їх формат залежить від того, як застосування визначає віддалену процедуру. Для UDP-дейтаграми довжина цього поля розраховується як розмір UDP-дейтаграми мінус довжина всіх полів. Коли замість UDP використовується TCP, то поняття фіксованої довжини не існує. У подібних випадках між TCP-заголовком і XID з'являється 4-байтове поле довжини, з якого отримувач дізнається довжину RCP-виклику у байтах.

На рис. 6.16 показано формат RCP-відгуку. Він відправляється від server stub до client stub, коли віддалена процедура закінчує свою роботу.

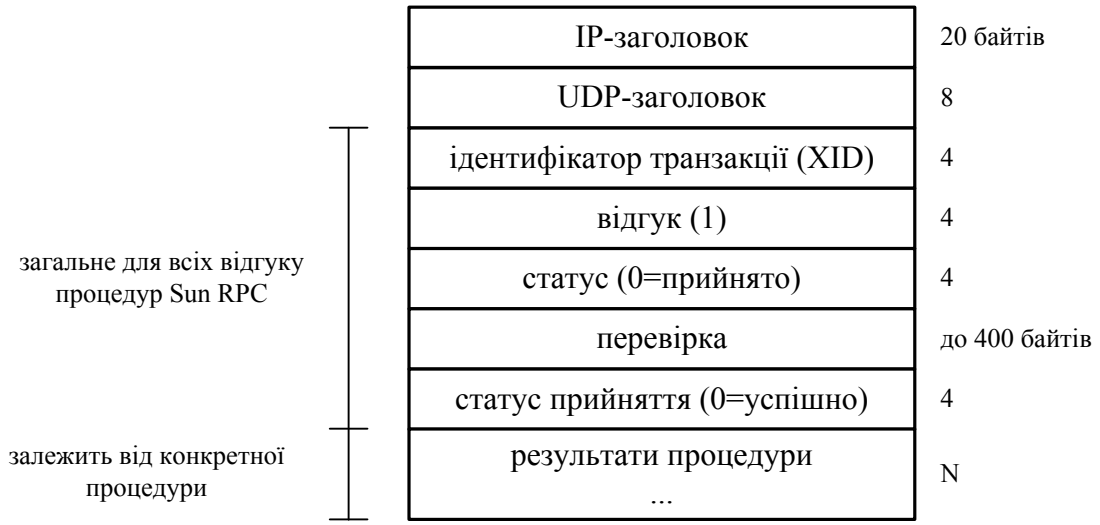


Рисунок 6.16 – Формат відгуку процедури RPC у форматі UDP-дейтаграми

XID виклику просто копіюється з XID-відгуку. У поле reply встановлюється 1. Поле статусу (status) містить нульове значення, якщо повідомлення відгуку було прийнято. Повідомлення може бути відкинуто, якщо номери версії RPC різні або сервер не може ідентифікувати клієнта. Поле перевірки (verifier) використовується у випадку захищеного RPC, щоб вказати сервер. Поле статус прийняття (accept status) має нульове значення, якщо все нормально. Ненульове значення може вказувати, наприклад, на неправильний номер версії або неправильний номер процедури. Якщо замість UDP використовується TCP, то між TCP-заголовком і XID розташовується 4-байтове поле довжини.

На рис. 6.17 показано налаштування NFS клієнта та NFS сервера. Після відкриття файлу ядро системи передає всі звернення або до локальних файлів, або NFS-айлів. NFS відправляє всі RPC запити NFS-серверу через модуль TCP/IP. NFS зазвичай використовує UDP протокол.

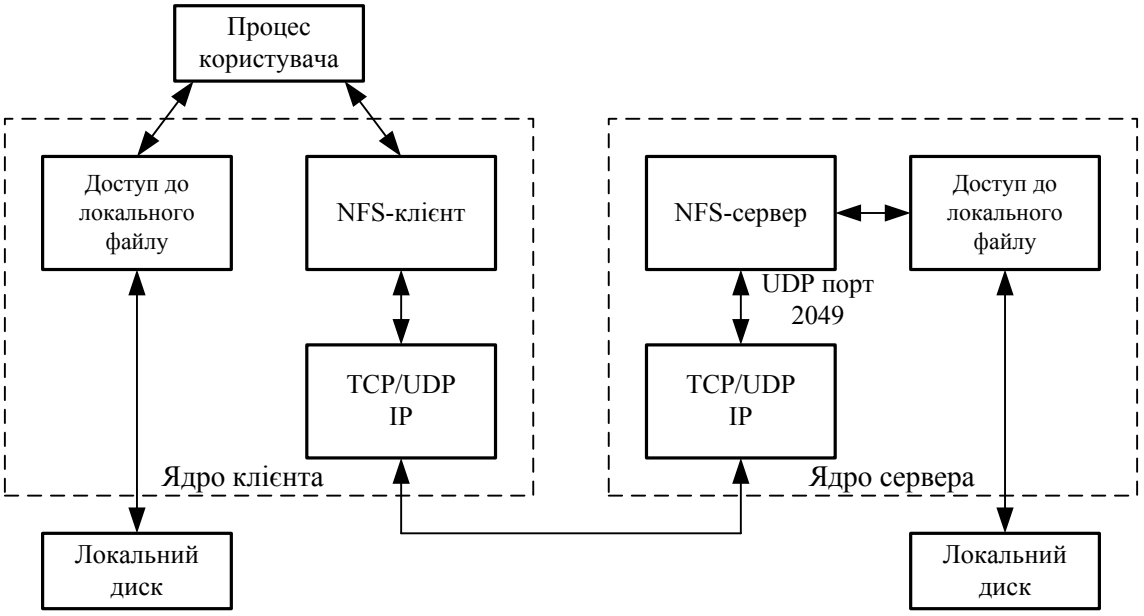


Рисунок 6.17 – З’єднання NFS-клієнта та NFS-сервера

NFS-сервер отримує запити від клієнта як UDP-дейтаграми на порт 2049. NFS може працювати з перетворювачем портів, що дозволяє серверу використовувати динамічне призначення портів.

Коли NFS-сервер отримує запит від клієнта, він передає локальній підпрограмі доступ до файлу, і вона забезпечує доступ до локального диска на сервері.

Серверу може бути потрібен деякий час, щоб опрацювати запит клієнта. Протягом цього часу сервер не блокує запити від інших клієнтів. Щоб відпрацювати дану ситуацію, більшість NFS-серверів запускають кілька разів, тобто всередині ядра існує кілька NFS-серверів. Конкретні методи розв’язання залежать від операційної системи. У більшості ядер Unix-систем не «живе» кілька NFS-серверів, замість цього запускається кілька

процесів користувача, що здійснюють один системний виклик і залишаються всередині ядра як процес ядра.

NFS-клієнту потрібен час, щоб опрацювати запит від користувального процесу на хості клієнта. RPC відправляється на хост сервера, після чого очікується відгук. Для того, щоб процеси користувача на хості клієнта могли у будь-який час використовувати NFS, існує кілька NFS-клієнтів, що запущені всередині ядра. Конкретна реалізація залежить від операційної системи. У Unix-системах, як правило, процес користувача здійснює один системний виклик і залишається всередині ядра як процес ядра.

Одна з основ NFS реалізується описувачами файлів. Для звертання до файлу або директорії на сервері використовується opaque. Термін opaque означає, що сервер створює описувач файлу, передає його назад клієнту, який клієнт потім використовує при звертанні до файлу. Вміст описувача файлу використовується лише сервером.

NFS-клієнт отримує описувач файлу кожного разу, коли відкривається файл, який знаходиться на NFS-сервері. Коли відбувається зчитування-запис у файл на сервері, описувач файлу знову передається серверу. Це вказує на те, що відбувся доступ до файлу.

Зазвичай застосування користувача не працюють з описувачами файлів. Обмін ними здійснюють NFS-клієнт і NFS-сервер. У NFS версії 2 описувач файлу займає 32 байти, а у версії 3 – 64 байти.

Перед тим як отримати доступ до NFS-файлів, клієнт використовує NFS-протокол монтування, щоб змонтувати файлову систему сервера. Зазвичай це відбувається при завантаженні клієнта. У результаті клієнт отримує описувач файлу файлової системи сервера.

NFS-сервер надає 15 процедур, опис яких наведено у табл. 6.15. Деякі з процедур можуть функціонувати лише на Unix-системах. Для NFSv3 розроблено додаткові процедури.

NFS спочатку було розроблено, щоб використовувати UDP, і це використовують різні виробники. Але новіші реалізації також підтримують TCP. Підтримка TCP використовується для роботи у глобальних мережах. Тому використання NFS сьогодні не обмежується локальними мережами.

Таблиця 6.15 – Опис процедур NFS-сервера

Назва процедури	Опис
NFSPROC_GETATTR	Повертає атрибути файлів: тип (звичайний файл, папка і т. д.), права доступу, розмір, власника, час останнього звертання і т. д.
NFSPROC_SETATTR	Встановлює атрибути файлу. Встановлено може бути лише певний набір атрибутів, зокрема: права доступу, власника, групова власність, розмір, час останнього звертання та ін.
NFSPROC_STATFS	Повертає статус файлової системи: розмір вільного простору, оптимальний розмір для передавання та ін.

NFSPROC_LOOKUP	Викликається клієнтом кожного разу, коли процес користувача відкриває файл, який знаходить на NFS-сервері. Повертає описувач файлу разом з атрибутами файлу.
NFSPROC_READ	Читає з файлу. Клієнт вказує описувач файлу, початковий зсув у байтах і максимальну кількість байтів, які потрібно зчитати (до 8192 для NFSv2).
NFSPROC_WRITE	Записує у файл. Клієнт вказує описувач файлу, початковий зсув у байтах, кількість байтів, які потрібно записати, і дані, які потрібно записати.
NFSPROC_CREATE	Створює файл.
NFSPROC_REMOVE	Знищує файл.
NFSPROC_RENAME	Перейменовує файл.
NFSPROC_LINK	Робить постійний лінк на файл. Постійний лінк – це концепція Unix, що визначає точку входу до файлу. Конкретний файл на диску може мати довільну кількість точок входу (імен, які також називаються постійними лінками), що вказують на цей файл.
NFSPROC_SYMLINK	Створює символічний лінк на файл. Символічний лінк – це файл, що містить ім’я іншого файлу. Більшість операцій, що здійснюється над символічним лінком (наприклад, відкривання), насправді працюють з тим файлом, на який вказує символічний лінк.
NFSPROC_ READLINK	Читання символічного лінку повертає ім’я файлу, на який вказує символічний лінк.
NFSPROC_MKDIR	Створює папку.
NFSPROC_RMDIR	Знищує папку.
NFSPROC_READDIR	Читає папку.

6.9 Питання для самоперевірки

1. У чому полягає основне призначення протоколу DHCP?
2. Визначте формат DHCP-повідомлення і призначення основних полів.
3. Визначте механізми виділення IP-адрес при використанні протоколу DHCP.
4. Перерахуйте основні конфігураційні параметри, що визначаються протоколом DHCP.
5. Визначте процес взаємодії клієнта і сервера при передаванні мережної адреси.
6. Який транспортний протокол використовує протокол DHCP?
7. Перерахуйте основні стани DHCP-клієнта і сервера.
8. Яке основне призначення протоколу DNS? Дайте короткий аналіз.
9. На які зони поділено домени верхнього рівня?
10. Визначте і охарактеризуйте простір DNS-імен.
11. Чим відрізняється основний сервер DNS від вторинного?

12. Для чого використовується кешування при роботі DNS-сервера?
13. Який формат має DNS-повідомлення. Охарактеризуйте основні поля.
14. Як DNS-сервер може опрацьовувати запити?
15. Яке призначення мають прапорці заголовка DNS-повідомлення?
16. Як зберігається ім’я домену у DNS-запиті?
17. Визначте базові концепції протоколу Telnet.
18. Назвіть та охарактеризуйте сценарії обговорення опцій при telnet-з’єднанні.
19. Вкажіть основні команди Telnet-клієнта.
20. Яка відмінність протоколу SSH від Telnet?
21. Які способи аутентифікації клієнта надає протокол SSH?
22. Визначте протоколи електронної пошти.
23. Яке призначення має протокол SMTP?
24. Визначте команди протоколу SMTP.
25. Охарактеризуйте протоколи POP3 та IMAP. Яка між ними відмінність?
26. Дайте загальний аналіз протоколам FTP та TFTP.
27. Визначте процедуру встановлення FTP-з’єднання.
28. Як подаються дані у FTP?
29. Визначте команди і відповіді протоколу FTP.
30. Назвіть та охарактеризуйте режими роботи FTP-сервера.
31. Який формат має повідомлення TFTP?
32. Охарактеризуйте протокол HTTP.
33. Визначте основні методи HTTP.
34. На які основні групи поділяють HTTP-заголовки. Охарактеризуйте кожну з них.
35. Визначте основні коди стану протоколу HTTP.
36. Який протокол використовується для керування мережами TCP/IP?
37. Визначне і охарактеризуйте типи повідомлень протоколу SNMP.
38. Який формат має повідомлення протоколу SNMP? Визначте призначення основних полів.
39. Охарактеризуйте протокол NFS.
40. Визначте порядок встановлення з’єднання NFS-клієнта та NFS-сервера.
41. Визначте та охарактеризуйте основні процедури NFS-сервера.

7 СУЧАСНІ НАПРЯМКИ РОЗВИТКУ КОМП'ЮТЕРНИХ МЕРЕЖ

7.1 Інтернет речей

Інтернет речей (Internet of Things, IoT) – це концепція обчислювальної мережі, що об'єднує фізичні предмети (речі), які обладнано вбудованими технологіями для взаємодії одне з одним або з зовнішнім середовищем. Ця концепція розглядає організацію таких мереж як явище, що здатне перебудувати економічні й суспільні процеси, усунувши з частини дій і операцій необхідність участі людини. Іншими словами – це концепція обчислювальної мережі фізичних предметів, які оснащено певними технологіями для взаємодії між собою.

У 2013 році компанія Cisco започаткувала ще один термін, що описує сучасний стан розвитку мережних технологій – **всеосяжний Інтернет** (Internet of Everything (IoE)). Відповідно до визначення компанії IoE об'єднує людей, процеси, дані та речі (рис. 7.1), щоб зробити мережні з'єднання більш значущими та цінними, ніж будь-коли раніше; перетворюючи інформацію на дії, які створюють безпрецедентні нові можливості для підприємств, окремих осіб і країн.

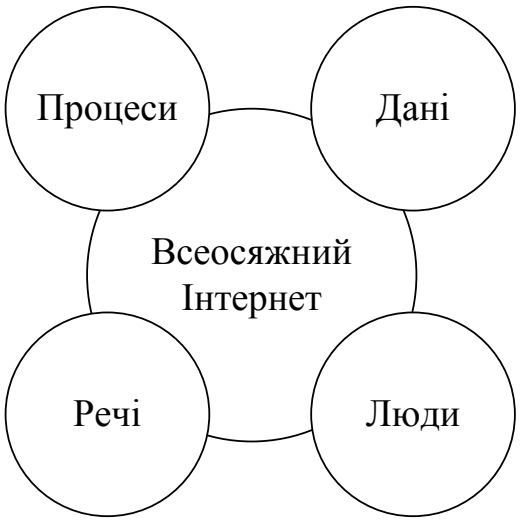


Рисунок 7.1 – Структура всеосяжного Інтернету

7.1.1 Речі як елемент IoE

На даний момент до складу «речей» відносять, головним чином, різноманітні комп'ютери і комп'ютерні пристрої, зокрема настільні комп'ютери, ноутбуки, смартфони, планшетні комп'ютери, універсальні ЕОМ, кластери обчислювальних машин тощо. Однак Інтернет речей містить й об'єкти і пристрої, які не завжди підключені. Такі об'єкти містять вбудовані технології, що забезпечують взаємодію з внутрішніми серверами або контролерами та зовнішнім середовищем, наприклад, системи відеос-

постереження, пожежної сигналізації, кондиціювання, комп'ютерна електроніка тощо. Ці об'єкти мають можливість підключення до мережі і можуть обмінюватися даними в захищеній, надійній та доступній мережній платформі. Інтернет Речей – це перехід до єдиної технології, можливість підключити об'єкти, які раніше не були підключені, щоб вони могли обмінюватись інформацією в мережі.

Окрім звичайних або портативних комп'ютерів і смартфонів користувачі все частіше використовують інтелектуальні побутові пристрої, наприклад, інтелектуальні годинники, холодильники, телевізійні системи, системи водонагрівання тощо. IoT є основою побудови так званого «розумного дому».

Всі пристрої «розумного дому» можна поділити на три категорії:

- сенсори або давачі,
- актуатори,
- контролери.

Використання **сенсорів** – це один із способів збирання даних з пристроїв, які не є комп'ютерами. Вони перетворять фізичні властивості нашого середовища в електричні сигнали, які можуть бути оброблені комп'ютерами. Як приклад можна навести сенсори ґрунтової вологи, температури повітря, радіації і руху.

Популярний тип сенсорів використовує радіочастотні мітки (Radio Frequency IDentification, RFID-мітки). RFID використовує радіочастотні електромагнітні поля для передавання інформації між невеликими мітками з кодом (RFID-мітки) і радіочастотним зчитувачем. Як правило, радіочастотні мітки використовуються для ідентифікації та відстеження об'єктів, в які вони вбудовані, наприклад, товари в складських приміщеннях. Оскільки ці мітки зовсім невеликі, їх можна прикріпити практично до будь-яких предметів. Деякі радіочастотні мітки не вимагають батарейок. Живлення, необхідне для передавання інформації, мітки отримують від електромагнітних сигналів, які відправляє зчитувач радіочастотних міток. Мітка отримує цей сигнал і використовує частину енергії сигналу для відправлення відповіді.

Актуатор (actuator) – це виконавчий пристрій або його активний елемент, що перетворює один вид енергії (електричну, магнітну, теплову, хімічну) в інший (найчастіше – в механічну), що приводить до виконання певної дії, заданої керівним сигналом. Призначається для того, щоб впливати на навколишнє середовище, або на певний об'єкт в ньому, наприклад, закрити або відкрити вікно.

Сенсори можна запрограмувати на виконання вимірювань, перетворення даних в сигнали і відправлення цих даних на основний пристрій, який називається **контролером**. Контролер відповідає за збір даних від сенсорів і за підключення до Інтернету. Контролери можуть самостійно приймати негайні рішення або ж відправляти дані для аналізу більш потужним комп'ютером. Такий більш потужний комп'ютер або знаходиться в

тій же локальній мережі, що і контролер, або доступний через Інтернет-підключення.

Контролери – це пристрої, які відповідають за збирання даних від сенсорів і за підключення до Інтернету. На них, зазвичай, покладають логіку поверхневого аналізу інформації, що надходить від підключених до них сенсорів. У певних ситуаціях аналіз даних може вимагати малої кількості обчислювальних ресурсів, так що контролери цілком здатні приймати деякі рішення самостійно. В цьому випадку вони відправляють певні команди керування на актуатори. Якщо ж обробка інформації потребує великих обчислювальних витрат, або ця інформація має десь накопичуватись, контролери відправляють її на сервери для подальшої обробки.

Актуальним питанням є вибір технології підключення пристроїв до мережі. Залежно від обсягу даних, відстані, способу живлення пристрою застосовується одна або кілька з технологій, наведених на рис. 7.2.

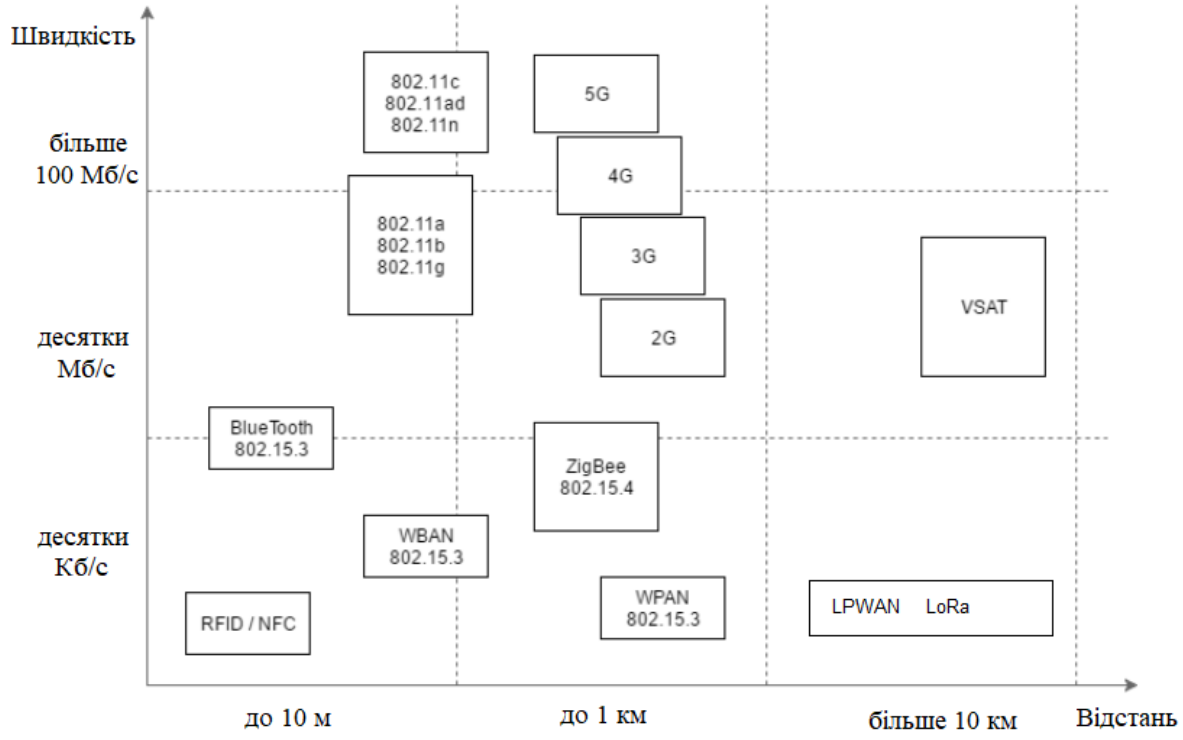


Рисунок 7.2 – Технології підключення пристроїв IoE до мережі

7.1.2 Дані як елемент IoE

Дані – це подання інформації у формалізованому вигляді, придатному для передавання, зберігання або обробки. Все, що нас оточує, породжує дані. Однак самі по собі дані можуть бути безглуздими. Дані стають корисними в міру того, як їх співвідносять або порівнюють. Корисні дані стають інформацією. Після застосовування або розуміння інформація перетворюється на знання.

Дані поділяються на структуровані і неструктуровані. **Структуровані дані** вводяться і зберігаються в фіксованих полях всередині файлу або за-

пису. Прикладом структурованих даних є електронна анкета, яку заповнює користувач при реєстрації на певних сайтах. Комп'ютер легко вводить, класифікує, шукає та аналізує структуровані дані.

Неструктуровані дані не оброблені. У них немає тієї складової, що ідентифікує значення даних. Для неструктурованих даних відсутній особливий спосіб введення або групування даних для їх подальшого аналізу. До прикладів неструктурованих даних відносять фотографії, аудіо- та відеофайли. Левова частка даних Інтернету є неструктурованими або слабкоструктурованими.

Структуровані і неструктуровані дані – це активи, які мають цінність для людей, організацій, галузей та урядів. Як і інші активи, інформація, зібрана зі структурованих і неструктурованих даних, є величиною, що може бути виміряна. Цінність таких даних може збільшуватися або зменшуватися залежно від способу керування ними, плину часу, поширюваності тощо.

За формою зберігання виділяють:

- **локальні дані** доступні безпосередньо, з локальних пристроїв. До локальних сховищ даних належать жорсткі диски, USB-сховища, CD- / DVD-диски;
  централізовані дані зберігаються й використовуються з єдиного централізованого сервера. Доступ до такої інформації можна отримати віддалено за допомогою різних пристроїв в локальній мережі або мережі Інтернет. Сервер та його інфраструктура в даному випадку є критичною точкою системи зберігання;
- **розподілені дані** дублюються і зберігаються в кількох місцях, що забезпечує просте та ефективне спільне використання даних. Крім того в цій системі немає єдиної точку збою, що збільшує рівень доступності даних.

Історично більшість даних є **статичними**, або нерухомими. Як правило, вони збираються протягом певного часу і мають певне значення для організації. Використання великої кількості різноманітних давачів створює неперервний потік даних, які мають цінність доки з ними взаємодіють користувачі, після чого їх цінність стрімко втрачається. Такі дані називають **динамічними**, або рухомими. Саме динамічні дані є основою Інтернету речей.

У зв'язку з бурхливим зростанням обсягів даних перед організаціями постає задача обробки та керування великими масивами даних. Великі масиви даних розглядаються у трьох вимірах: обсяг, різноманіття і швидкість передавання. **Великі масиви даних (big data)** – це процес збирання та аналізу великих обсягів даних організаціями для того, щоб визначати тенденції, передбачати поведінку і надавати максимальні можливості тим, хто приймає рішення.

Додатки для великих масивів даних мають вміти збирати ці дані і структурувати їх таким чином, щоб організації отримували від цього користь.

Наприклад, додатки для великих масивів даних мають враховувати мінливі джерела й тенденції даних, до яких належать:

- мобільний зв'язок – мобільні пристрої, події, поширення інформації та інтеграція сенсорів;
- доступ до даних і їх використання – Інтернет, з'єднані між собою системи, соціальні мережі і моделі доступу;
- можливості екосистеми – головні зміни в моделі обробки інформації та доступності відкритого середовища.

Результатом є збільшення вартості і складності моделей даних, змінення принципів зберігання та аналізу великих масивів даних, а також доступу до них. Внаслідок цього організації все частіше використовують віртуалізацію і хмарні обчислення для підтримки потреб великих масивів даних.

7.1.3 Люди і процеси як елементи IoE

Кінцевою метою організації даних, що породжуються, передаються та обробляються в межах Всеосяжного Інтернету, є допомога людям приймати обґрунтовані рішення й виконувати потрібні дії. Саме тому люди є центральним елементом IoE. Процеси сприяють взаємодії між людьми, речами і даними, комбінуючи підключення «машина-машина» (M2M), «машина-людина» (M2P) і «людина-людина» (P2P).

Підключення **«машина-машина» (M2M)** відбуваються, коли дані передаються по мережі від однієї машини або «речі» до іншої без участі людини. До машин відносять давачі, роботи, комп'ютери та мобільні пристрої. Часто підключення M2M власно і називають Інтернетом Речей.

Підключення **«машина-людина» (M2P)** відбуваються, коли інформація передається між машиною (наприклад, сервером або web-камерою) і людиною. Якщо людина отримує інформацію з бази даних або віддалено керує іншим пристроєм, то це підключення M2P. Підключення M2P сприяють передаванню, керуванню, обробці даних й отриманню звітності від машин для прийняття людьми обґрунтованих рішень. Дії, що виконуються людьми, виходячи з обґрунтованих рішень, завершують петлю зворотного зв'язку Всеосяжного Інтернету.

Підключення **«людина-людина» (P2P)** відбуваються, коли інформація передається від однієї людини до іншої. Це традиційні сервіси електронної пошти, програми обміну миттєвими повідомленнями. Останнім часом все ширше застосовуються інтерактивні комунікаційні сервіси, що працюють в режимі реального часу і дозволяють встановлювати через Інтернет відео- та аудіосеанси, наприклад Ip-телефонія, системи типу Skype, Viber, Telegram тощо. Дуже активно P2P підключення розвивають сервіси соціальних мереж.

7.1.4 Протоколи Інтернету речей

M2M є основою Інтернету речей і передбачає взаємодію різноманітних давачів, актуаторів, контролерів та шлюзів. Таким чином виникає необхідність в «особливих» протоколах для забезпечення взаємодії цих пристроїв один з одним і верхніми рівнями. Стандартні прикладні протоколи не під-

ходять через їх непристосованість до умов мережі Інтернету речей. Давач, зазвичай мініатюрний, з невеликою пам'яттю, вимірює фізичні параметри в режимі реального часу, найчастіше в умовах низького енергозабезпечення. Результати вимірювань обробляються сенсорним вузлом і передаються на сервер. З іншого боку, обсяг інформації, що формується одним сенсорним вузлом, порівняно невеликий, проте більшість сервісів Інтернету речей побудовано на принципі обробки інформації від значної кількості вузлів, що принципово відрізняється від архітектури клієнт-сервер.

Для такого рівня задач краще підходить шаблон **видавник-підписник (publisher-subscriber)**. Видавник є джерелом інформації, а підписник – її одержувачем. Термін «підписка» пов'язаний з певною операцією, що виконується учасниками взаємодії з метою отримання інформації підписником від конкретного видавника. Топологія реалізації M2M за схемою видавник-підписник наведена на рис. 7.3.

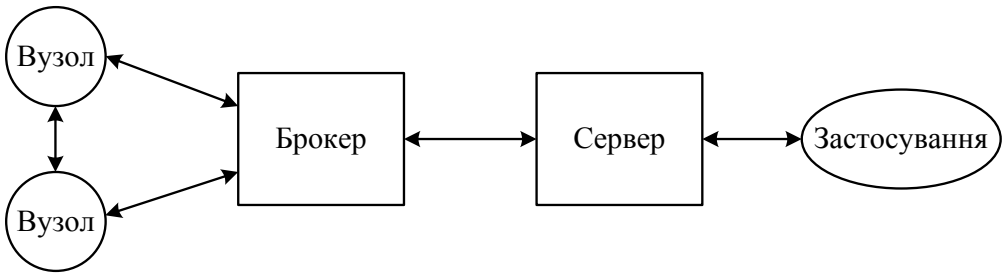


Рисунок 7.3 – Топологія реалізації M2M
за схемою видавник-підписник

Складовими топології є:

- **сенсорний вузол**, який збирає, а в деяких випадках і об'єднує, інформацію від кількох сенсорів та спрямовує її на сервер. Оскільки сенсори в більшості випадків є примітивними пристроями, єдиною задачею яких є постійне передавання контрольованого параметра, сенсорні вузли оснащують мікроконтролерами, які будуть відповідати за зчитування вимірюваних даних і відправляти їх за заздалегідь визначеними алгоритмамих далі на сервер;
- **сервер** – це місце збирання та обробки інформації з сенсорних вузлів;
- **застосування** – це спеціалізоване програмне забезпечення, що слугує для візуалізації одержуваної з давачів або вже обробленої сервером інформації та керування системою;
- **брокер** – це сервер, який приймає інформацію від видавників (сенсорних вузлів) і передає її відповідним підписникам (серверам). Крім того брокер може додатково виконувати різні операції, пов'язані з аналізом і обробкою отриманих даних, встановлювати пріоритети повідомленням і формувати черги для їх передавання.

Таким чином, для взаємодії елементів подібної системи необхідні спеціальні прикладні протоколи. Відповідно до схеми на рис. 7.3 поділимо ці протоколи на чотири групи, що будуть відповідно забезпечувати взаємодію: вузол – вузол, вузол – брокер, брокер – сервер та сервер – застосування.

Протоколи «вузол – вузол» використовуються для обміну інформацією між сенсорними вузлами, наприклад, розподіл інформації між сенсорними вузлами для тимчасового зберігання або перенаправлення. Для забезпечення зв'язку між сенсорними вузлами (давачами) може бути використано протокол **DDS (Data Distribution Service)**.

DDS забезпечує обмін даними, регламентований вимогами до якості обслуговування (QoS). Всі пристрої, що беруть участь у обміні даними, створюють так званий DDS-домен (рис. 7.4). Елементи домену взаємодіють шляхом публікації даних і підписання на теми, визначені за назвою теми. При створенні підписки можуть бути вказані фільтри часу та вмісту, тобто підписка на отримання частини даних, опублікованих за темою у визначені моменти часу.

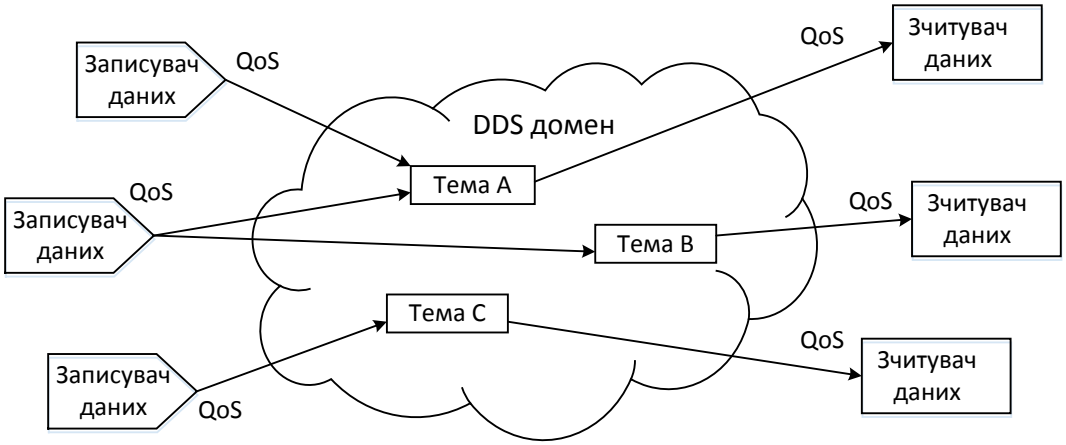


Рисунок 7.4 – Схема зв'язку між сенсорними вузлами з використанням протоколу DDS

Протокол DDS розподіляє дані, реалізуючи прямий шинний зв'язок між пристроями на базі реляційної моделі даних. DDS реалізує багатоадресну систему, використовуючи на транспортному рівні протокол UDP. Даний протокол орієнтований на взаємодію за шаблоном «видавник-підписник», при цьому передавання повідомлень здійснюється шиною з використанням методу «запит – відповідь».

На ділянці «вузол – брокер» реалізується кілька задач, наприклад, реєстрація сенсорного вузла, конфігурування та налаштування вузлів, передавання і розподіл інформації тощо. Ці задачі можуть бути реалізовані протоколами XMPP та CoAP.

XMPP (Extensible Messaging and Presence Protocol) – розширюваний протокол обміну повідомленнями та інформацією про присутність. Забез-

печує простий спосіб адресації пристроїв Інтернету речей. Для ідентифікації використовуються ідентифікатори, що за форматом схожі на адреси електронної пошти наприклад, username@jabber.com. Протокол XMPP використовує текстовий формат XML, на транспортному рівні застосовується протокол TCP. XMPP підтримує різні комунікаційні моделі: запит – відповідь, публікація – підписка, асинхронні повідомлення тощо.

Адресація XMPP особливо зручна у випадках, коли дані передаються між віддаленими, найчастіше незалежними точками, наприклад в разі взаємодії двох абонентів. За допомогою XMPP, наприклад, можливе підключення домашнього термостата до Web-серверу для отримання до нього доступу з телефону. Головними перевагами цього протоколу є також безпека й масштабованість.

CoAP (Constrained Application Protocol) – це спеціалізований протокол передавання, розроблений робочою групою IETF – CORE, створений для мереж і пристроїв з обмеженими ресурсами, M2M-додатків тощо. CoAP можна розглядати як доповнення до HTTP, але, на відміну від HTTP, CoAP націлений на використання в пристроях з певними обмеженнями. CoAP використовує транспортний протокол UDP.

Протокол CoAP використовує повідомлення, більшість яких аналогічні відповідним повідомленням протоколу HTTP, зокрема: GET, PUT, HEAD, POST, DELETE, CONNECT. Клієнти (додатки користувача) використовують повідомлення для керування й спостереження за ресурсом. За запитом встановлюється прапорець спостереження, і сервер продовжує відповідати після того, як первинне повідомлення було передано. Це дозволяє серверам організовувати потокове передавання змін станів сенсорів.

Вибір конкретного протоколу (XMPP або CoAP) залежить від особливості реалізовуваної мережі. XMPP знайшов своє застосування в системах освітлення та клімату, також використовується для адресації пристроїв в невеликих персональних мережах. Протокол CoAP здебільше застосовується в системах давачів температури та інших давачів «розумного» будинку.

Виходячи з призначення брокера в мережі Інтернету речей, можна виділити задачі, що мають вирішуватися на ділянці «брокер – сервер», а саме: збирання та агрегація даних; організація черг повідомлень; розподіл і зберігання інформації «до запитання». Для завантажених мереж з великою кількістю пристроїв раціональніше застосовувати протокол, що знижує навантаження на канал за рахунок організації черг, зокрема протокол MQTT.

Протокол **MQTT (Message Queue Telemetry Transport)** призначений для телеметрії та дистанційного моніторингу. Використовується для обміну повідомленнями між пристроями за принципом «видавник – підписник», дає їм змогу надсилати і отримувати дані при виникненні певної події. MQTT є бінарним протоколом, що працює з використанням транспорту TCP.

Спрощена схема, що ілюструє обмін повідомленнями MQTT, наведена на рис. 7.5. Протокол використовує чотирнадцять повідомлень у форматі

запит – відповідь, зокрема: CONNECT, CONNACK, PUBLISH, PUBACK, PUBREC, PUBREL, PUBCOMP, SUBSCRIBE, SUBACK, UNSUBSCRIBE, UNSUBACK, PIN-GREQ, PINGRESP, DISCONNECT. Відповідно до специфікації за допомогою вказаних повідомлень існує можливість реалізувати механізм пріоритетів. В даному випадку керувати відправленням повідомлень за допомогою трьох класів QoS.

Взаємодія між сервером і застосуваннями (див. рис. 7.3) найчастіше реалізується за допомогою протоколу SOAP, оскільки у цього протоколу є виділений механізм доступу RPC (Remote Procedure Call), який відповідає за віддалений виклик функцій.

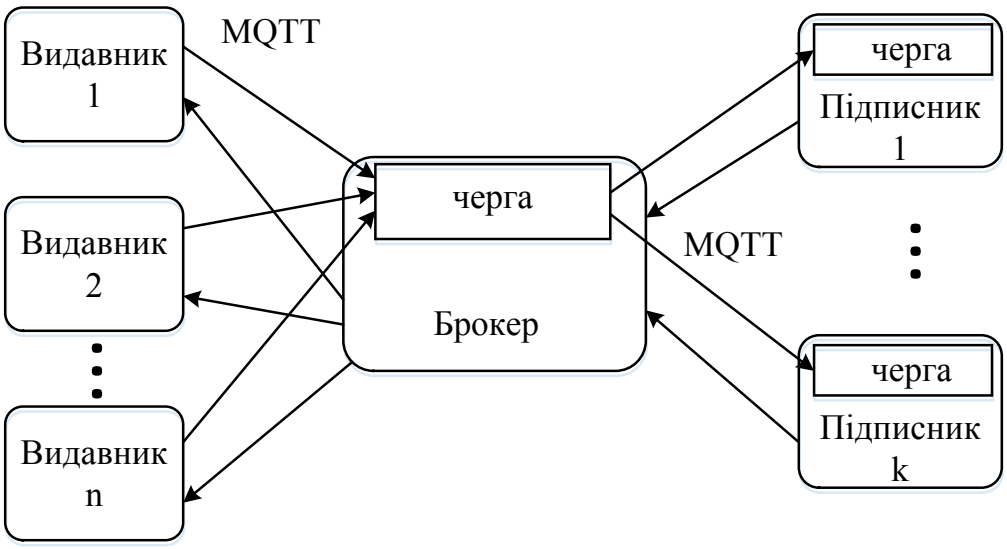


Рисунок 7.5 – Схема обміну повідомленнями за протоколом MQTT

SOAP (Simple Object Access Protocol) – це протокол обміну структурованими та довільними повідомленнями у форматі XML в розподіленому обчислювальному середовищі. SOAP використовує базову модель з'єднання, що забезпечує узгоджене передавання повідомлення від відправника до одержувача, потенційно допускає наявність посередників, які можуть обробляти частину повідомлення або додавати до нього додаткові елементи.

SOAP підтримує два механізми доступу – SOAP RPC і SOAP Message. SOAP RPC являє собою простий протокол «запит – відповідь» і використовується для синхронного віддаленого виклику процедур за допомогою XML. SOAP Message – це протокол для відсилання та обробки SOAP-повідомлень, який може використовуватися для асинхронних комунікацій і передбачає можливість як негайної, так і відкладеної відповіді на запит. Завдяки всього кількох повідомленням (Get, SOAPAction, SOAPAction-Response) протокол може використовуватися з будь-яким протоколом прикладного рівня: SMTP, FTP, HTTP, HTTPS.

7.2 Основи хмарних технологій

Хмарні обчислення (cloud computing) – це технологія розподіленої обробки даних в якій комп'ютерні ресурси та потужності надаються користувачеві як Інтернет-сервіс. Хмарні обчислення припускають наявність великої кількості підключених через мережу комп'ютерів, які фізично можуть розміщуватися в будь-якій точці Земної кулі. Ця технологія допомагає скоротити операційні витрати за рахунок більш ефективного використання ресурсів.

Хмарні технології дозволяють вирішувати різні завдання керування даними, забезпечуючи:

- доступ до даних організації в будь-який час та з будь-якого місця;
- оптимізацію IT-інфраструктури в організації за рахунок підписання тільки на необхідні послуги;
- скорочення або уникнення витрат на розгортання, підтримку та оновлення обладнання й програмного забезпечення;
- скорочення витрат на електроенергію, зменшення вимог до матеріальної частини та потреби в навчанні персоналу;
- поява можливості швидкого масштабування обчислювальних ресурсів, адаптації до нових бізнес-моделей підприємства.

З іншого боку, не слід забувати і про такі особливості хмарних технологій:

- необхідність постійного та якісного доступу до мережі Інтернет;
- наявність обмежень на програмне забезпечення, що можна розгорнути в хмарі, в більшості випадків є обмеження щодо налагодження його під конкретні задачі користувачів;
- проблема конфіденційності – в більшості випадків провайдери хмарних сервісів не дають стовідсоткової гарантії конфіденційності даних.

Існує багато різновидів хмарних сервісів, орієнтованих на різні вимоги клієнтів. Три основних типи послуг хмарних сервісів, згідно з визначенням Національного інституту зі стандартизації та технологій США, такі.

- **Програмне забезпечення як послуга (Software as a Service – SaaS)**. Постачальник хмарних сервісів надає доступ до послуг електронної пошти, обміну даними, створення та редагування офісних документів тощо. Прикладами таких сервісів є Google Docs та Office 365**.**
- **Платформа як послуга (Platform as a Service – PaaS)**. Користувачеві надається комп'ютерна платформа зі встановленою операційною системою і певним програмним забезпеченням. Постачальник хмарних сервісів відповідає за надання доступу до засобів розробки, тестування та розгортання програмного забезпечення.

- **Інфраструктура як послуга (Infrastructure as a Service – IaaS)**. Постачальник хмарних сервісів відповідає за доступ до мережного обладнання, віртуалізованих мережних сервісів та підтримку мережної інфраструктури.

Існують різні моделі хмарних сервісів, зокрема: публічні, приватні, гібридні та колективні.

Хмарні програми та служби, що надаються в **публічній** (загальнодоступній) хмарі, доступні практично всім користувачам. Сервіси можуть бути безкоштовними або пропонуватися за моделлю «оплата за фактом використання», як у випадку купівлі місця в хмарному сховищі. Загальнодоступна хмара використовує Інтернет для надання послуг.

Хмарні програми та служби, що надаються в **приватній** хмарі, призначені для певної організації або юридичної особи, наприклад, для державної установи. Приватна хмара може бути організована на базі приватної мережі компанії. Однак створення та обслуговування такої хмари вимагає значних витрат. Саме тому керування приватною хмарою, як правило, доручають іншій організації, яка здатна забезпечити необхідний рівень безпеки доступу до ресурсів компанії.

Гібридна хмара складається з двох або більше хмар (наприклад, приватної і публічної), причому кожна з частин залишається окремим об'єктом, але вони пов'язані між собою в рамках єдиної архітектури. Користувачі, підключені до гібридної хмари, можуть мати різні рівні доступу до сервісів залежно від своїх прав доступу.

Колективні хмари створюються для виняткового використання певною спільнотою. Відмінності між загальнодоступною та колективною хмарами полягають у функціональних потребах, налагоджених для певної спільноти. Наприклад, медичні установи мають дотримуватися певної політики захисту персональної інформації пацієнтів, що накладає додаткові вимоги на процеси автентифікації і конфіденційності.

7.3 Центри обробки даних

Фізичною основою хмарних технологій, як правило, є **дата центри (data center)** або їх ще називають **центрами (зберігання і) обробки даних (ЦОД/ЦЗОД)**. ЦОД спеціалізоване приміщення, в якому розташовано серверне і мережне обладнання, підключене до мережі Інтернет. ЦОД призначено для забезпечення єдиного цілісного інформаційного ресурсу підприємства з гарантованими рівнями достовірності, доступності та безпеки даних. Ядром ЦОД є IT-інфраструктура, що складається з серверів, систем зберігання даних і допоміжних систем для керування внутрішнім передаванням даних і захисту інформації.

До складу ЦОД, як правило, входять дві різні мережі передавання даних: **фронтальна (front-end)** – мережні служби і **внутрішня (back-end)**, або операційні служби. Пристрої та системи фронтального ряду відпові-

дають за безпечний доступ до ресурсів ЦОД і містять веб-сервери, брандмауери, проксі-сервери, VPN-концентратори тощо. Внутрішня мережа містить власне сервери та обладнання для комунікації між серверами та масивами даних.

Основним критерієм оцінювання якості роботи будь-якого дата-центру є час доступності серверів (uptime). Сучасні ЦОД забезпечують час доступності серверів на рівні 99.999% і вище. Рівень доступності 99.999% (п'ять дев'яток) визначає максимальний час недоступності сервера (downtime) 5 хвилин і 15 секунд на рік.

Характерною рисою ЦОД є велика щільність серверів, мережного обладнання та систем зберігання даних. Саме тому основними функціями ЦОД є:

- захист обладнання від впливу навколишнього середовища;
- забезпечення обладнання якісним і безперебійним електроживленням;
- відведення тепла, що виділяється;
- керування фізичним доступом до обладнання та його охорона.

Можна виділити два різновиди ЦОД:

- **ЦОД комерційного типу** – це цілі комплекси, які споруджують для подальшої оренди обчислювальних потужностей. Вони відрізняються високою продуктивністю і максимальною швидкістю обміну даними. Користувач, скориставшись такою послугою, отримує віртуальний ЦОД, який фактично може перебувати в іншому місті або навіть країні.
- **Внутрішні (корпоративні) ЦОД** – центри, які встановлюються в межах конкретного підприємства. Вони призначені тільки для внутрішнього корпоративного користування. Незважаючи на витрати, пов'язані зі створенням і введенням в експлуатацію такого центру, головна його перевага полягає в прямому керуванні всією системою. Компанія не залежить від стороннього власника обладнання (постачальника послуг) і зможе більш ефективно забезпечити безпеку даних і збереження комерційної таємниці.

До структури будь-якого ЦОД входять такі блоки:

- IT-інфраструктура;
- інженерні системи різного рівня складності;
- комплексна система захисту і безпеки;
- система управління та моніторингу.

IT-інфраструктура являє собою комплекс високотехнологічного обладнання, який об'єднаний в загальну систему. Це своєрідне ядро будь-якого сучасного ЦОДа, яке складається з серверного обладнання та систем передавання, обробки та зберігання даних.

Щоб забезпечити безперебійну роботу потужного високотехнологічного обладнання, потрібно створити **ефективну інженерну систему**. Основні

інженерні системи ЦОД відносять до двох категорій: електропостачання та охолодження.

Спеціальне обладнання має забезпечувати не тільки безперебійне подання електрики до обладнання, а й у разі аварій на ЛЕП перейти на автономне живлення. Для здійснення цього використовують різні джерела безперебійного живлення і додаткові генератори.

Потужні сервери під час роботи виділяють величезну кількість тепла, яке відводиться за допомогою спеціальних вбудованих радіаторів. Це не вирішує проблему повністю, оскільки серверні станції знаходяться в окремих закритих приміщеннях. Щоб забезпечити надійне охолодження, використовують спеціальні системи кондиціонування та вентиляції.

Одна з найважливіших складових працездатності будь-якого ЦОД – т це правильно розроблена **система безпеки**. Для того щоб забезпечити зберігання даних і запобігти доступу до них сторонніх осіб через Інтернет, компанії використовують новітнє антивірусне ПЗ та інші програми захисту даних. Щоб запобігти проникненню сторонніх осіб, розробляється і впроваджується система допуску певної категорії працівників, методи протидії злому і фізичному проникненню до обладнання, відеоспостереження, протипожежна система. Комплексний підхід до безпеки забезпечує збереження важливої інформації.

Комплексний підхід до **керування і моніторингу** є важливою складовою будь-якого сучасного ЦОДа. Такі системи в автоматичному режимі контролюють працездатність всього обладнання й параметри навколишнього середовища (температура, вологість, напруга і частота струму). Невід'ємною частиною моніторингу є системи прогнозування ймовірної відмови обладнання та раннього оповіщення.

Окреме місце займає диспетчеризація – це важлива складова робочого процесу, яка дозволяє організувати інформування персоналу про позаштатну ситуацію за допомогою сучасних технологій зв’язку (відправлення даних на електронну адресу, можливість автоматичного дозвону абонентам певної групи або короткі СМС-повідомлення). Комплексний підхід суттєво знижує ризик виникнення аварійних ситуацій, а в разі настання форс-мажору забезпечує максимально ефективну методику боротьби та відновлення пошкоджених ділянок системи.

Дизайн ІТ-інфраструктури більшості центрів обробки даних базується на перевіреному багаторівневому підході (рис. 7.6), який дозволяє покращити масштабованість, продуктивність, гнучкість, стійкість та спростити обслуговування.

ІТ-інфраструктура ЦОД складається з магістрального рівня (core), рівнів агрегації (Aggregation) та доступу (Access). До рівня магістралі під’єднуються відповідні магістралі мережі Інтернет та кампусних мереж (на рисунку не показані).

Магістраль забезпечує швидкісну комутацію пакетів для всіх потоків, що надходять і виходять з ЦОД. Підключення до декількох модулів агре-

гації забезпечує надійний зв'язок з рівнем агрегації, уникаючи єдиної точки відмови. Також на цьому рівні здійснюється балансування трафіку між кампусною мережею і рівнем агрегації.

Рівень агрегації забезпечує інтеграцію сервісних модулів, обмежує кордони широкомовного домену, формує STP-дерево та забезпечує надмірність для шлюзу за замовчуванням. На цьому рівні можуть виконуватись сервіси брандмауера та балансування навантаження серверів, що дозволяє підвищити захищеність й ефективність використання ресурсів серверів.

Рівень доступу забезпечує фізичне підключення до мережі серверів різного типу: одноюнітових серверів, блейд-серверів, серверних кластерів і мейнфреймів. Мережна інфраструктура рівня доступу складається з різноманітних комутаторів.

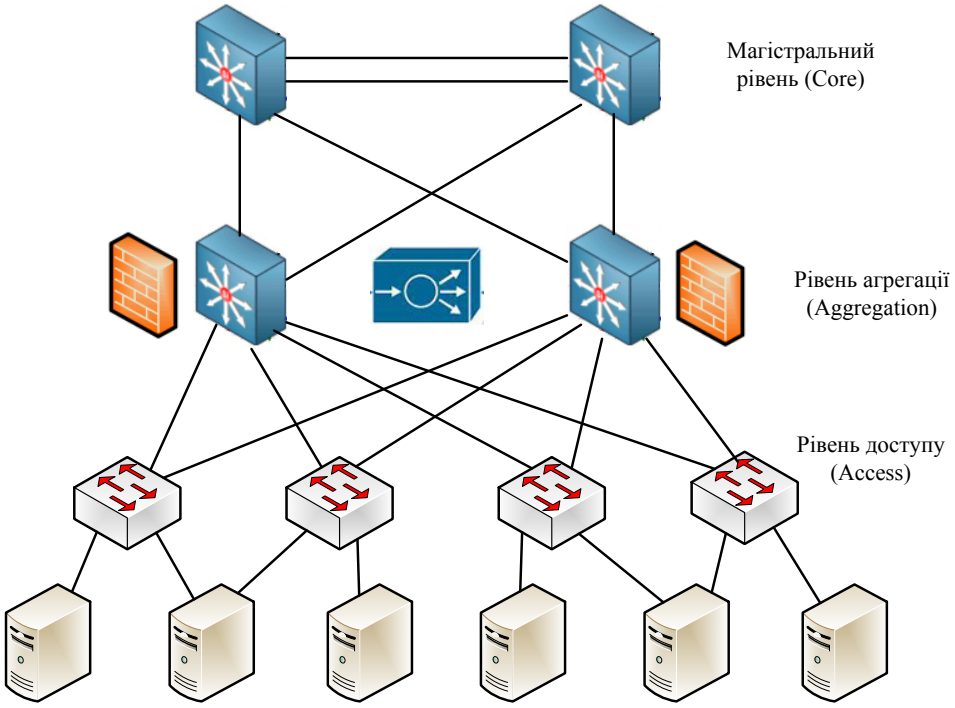


Рисунок 7.6 – Багаторівневий підхід реалізації мережі ЦОД

При створенні сучасних ЦОД використовується, як правило, одна з двох моделей: багатошарова (multi-tier) та кластерна (claster) моделі.

Багатошарова модель (рис. 7.7) є більш поширеною як в корпоративних ЦОД, так і комерційних. Вона передбачає використання чотирьох незалежних шарів:

- **ресурсні сервери**, або сервери інформаційних ресурсів, відповідають за зберігання і надання даних серверам додатків, наприклад, файл-сервери або сервери баз даних;
- **сервери додатків** виконують обробку даних відповідно до бізнес-логіки системи; наприклад, сервери обробки високоякісних зображень або відео в режимі реального часу;

- **сервери подання інформації** надають інтерфейс між користувачами і серверами додатків; наприклад, web-сервери;
- **службові сервери** (на рисунку не показані) забезпечують роботу інших підсистем ЦОД; наприклад, сервери керування системою резервного копіювання.

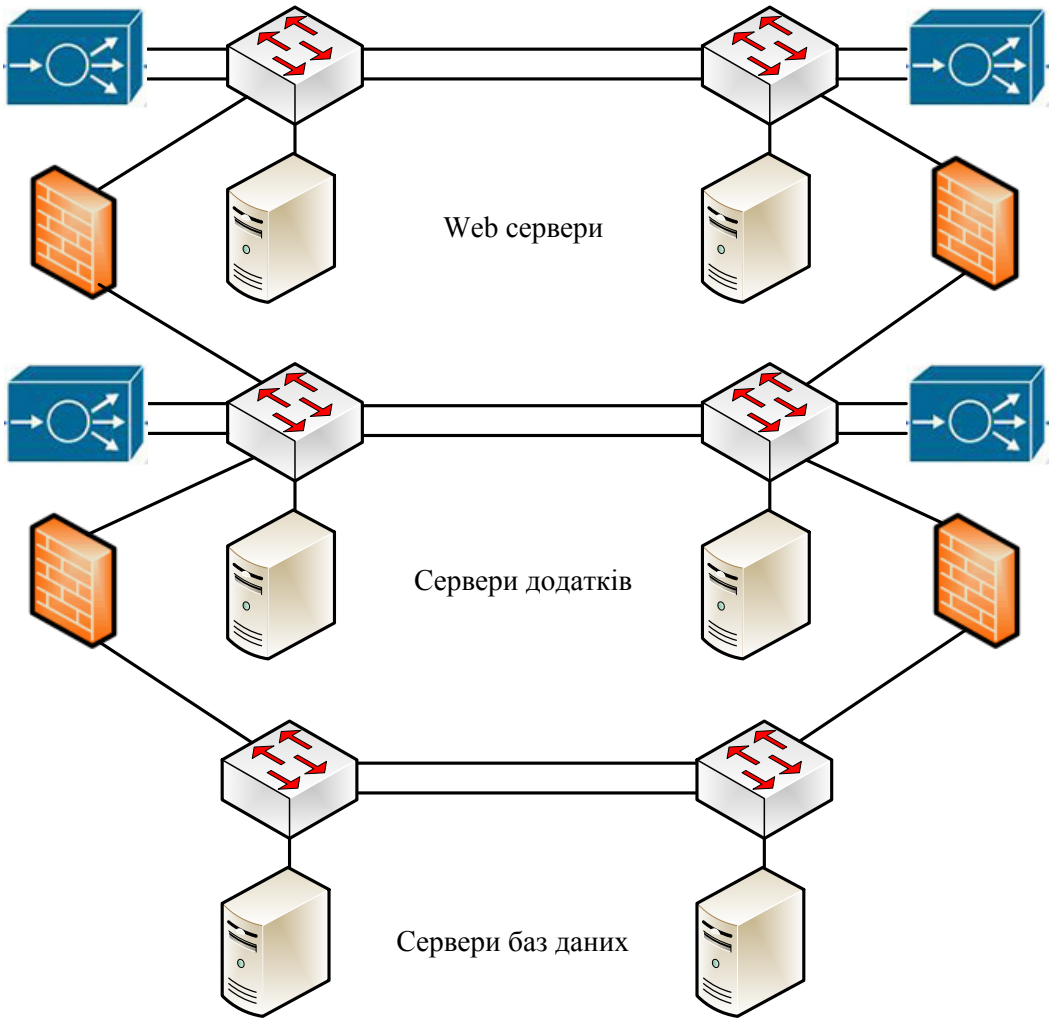


Рисунок 7.7 – Реалізації мережі ЦОД за багатошаровою моделлю

Багатошарова модель забезпечує високий рівень гнучкості, масштабованості, надійності, захищеності та еластичності. **Гнучкість** реалізується за рахунок використання програмного забезпечення, що запускається у вигляді окремих процесів, які взаємодіють між собою на одному комп'ютері через механізм міжпроцесорної взаємодії (IPC), або через мережу на різних машинах. **Масштабованість** може бути реалізована шляхом додавання нових серверів до відповідного шару. **Надійність** забезпечується за рахунок використання кількох серверів в межах кожного шару, що забезпечує працездатність системи при виході з ладу одного з них. **Високий рівень безпеки** реалізується завдяки ізоляції окремих шарів

між собою за допомогою брандмауерів. Наприклад, зловмисник може скомпрометувати веб-сервер, не отримавши доступу до серверів додатків або серверів баз даних. **Еластичність** досягається за рахунок балансування навантаження, що врівноважує мережний трафік між ярусами.

У сучасному середовищі центру обробки даних **кластери серверів** використовуються для багатьох цілей, зокрема для забезпечення високого рівня доступності, балансування навантаження та збільшення обчислювальної потужності. Усі кластери мають загальну мету – об'єднати декілька процесорів, щоб подати їх як єдину високоефективну систему, використовуючи спеціальне програмне забезпечення та високошвидкісні мережні з'єднання. Серверні кластери історично асоціювалися з університетськими дослідженнями, науковими лабораторіями та військовими дослідженнями для таких унікальних застосувань, як метеорологія (моделювання погоди), сейсмологія (сейсмічний аналіз), військові дослідження.

Сьогодні серверні кластери застосовуються до більш широкого спектра застосувань: аналіз фінансовий тенденцій – аналіз цін на облігації в реальному часі та з урахуванням історичного тренду; анімація фільмів – рендеринг мультигігабайтових файлів; виробництво – моделювання автомобільного дизайну та аеродинаміки; пошукові системи – швидкий паралельний пошук.

Кластерна модель інформаційної системи ЦОД наведена на рис.7.8.

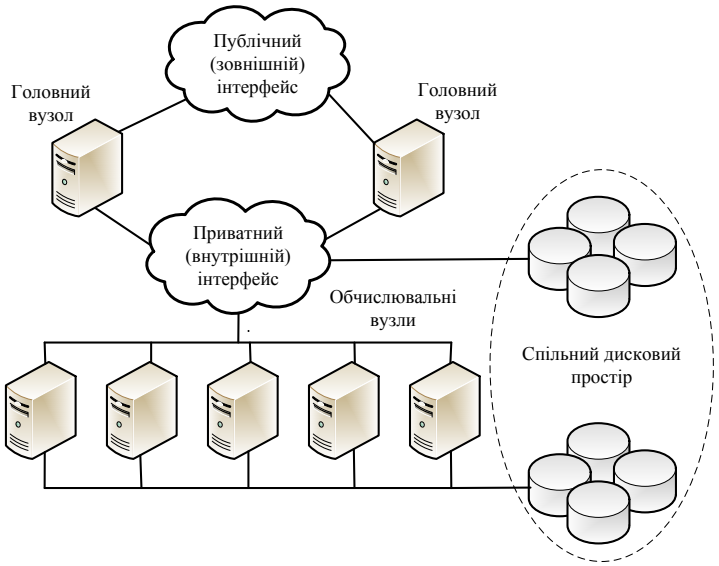


Рисунок 7.8 – Реалізації мережі ЦОД за кластерною моделлю

Зовнішній інтерфейс (front end) – це інтерфейс, який використовується для зовнішнього доступу до кластера, до якого можуть звертатися сервери додатків або користувачі, які відправляють завдання або отримують результати роботи з кластера. Зазвичай, це Ethernet IP інтерфейс, підключений до рівня доступу інфраструктури серверної ферми.

Основний або головний вузол (master node) відповідає за керування обчислювальними вузлами в кластері та оптимізацію загальної обчислювальної ємності. Зазвичай головний вузол – єдиний вузол, який спілкується з зовнішнім світом. Кластеризація проміжного програмного забезпечення, що працює на головних вузлах, надає інструменти для керування ресурсами, планування завдань та моніторингу стану обчислювальних вузлів кластера. Основні вузли, як правило, розгортаються надлишковим способом (у кількості два або більше) і, як правило, є більш потужними порівняно з обчислювальними вузлами.

Внутрішній інтерфейс основного вузла (back end) через високошвидкісну комунікаційну фабрику з'єднаний з обчислювальними вузлами. Типові вимоги до фабрики: низька затримка та висока пропускна спроможність. Найчастіше реалізується комутаторами з портами Gigabit та 10 Gigabit Ethernet. Альтернативною до технології Ethernet є відносно нова технологія **Infiniband**.

На обчислювальних вузлах встановлено оптимізоване або повне ядро операційної системи, і вони, як правило, виконують операції, що потребують інтенсивних обчислень, зокрема обробка великих масивів чисел, рендерінг, компіляція або інші маніпуляції файлами.

Інтерфейс до системи зберігання може використовувати інтерфейси Ethernet або Fiber Channel. Інтерфейси Fiber Channel складаються з інтерфейсів 1G/2G/4G і зазвичай підключаються до комутатора SAN.

Кластер серверів використовує спільну паралельну файлову систему, яка надає швидкісний одночасний доступ до файлових ресурсів всім обчислювальним вузлам. Типи файлової системи залежать від операційної системи (наприклад, PVFS або Luster).

Базовою технологією організації доступу до спільної файлової системи ЦОД є **технологія SAN (Storage Area Network)**. Асоціація промислових мереж зберігання даних (SNIA) визначає мережу сховищ SAN як мережу, основною метою якої є передавання даних між комп'ютерними системами та елементами зберігання. SAN складається з комунікаційної інфраструктури, яка забезпечує фізичне з'єднання та рівень керування, який організовує з'єднання елементів зберігання з комп'ютерними системами та забезпечує безпечне й надійне передавання даних.

Фактично SAN – це спеціалізована швидкісна мережа, яка з'єднує сервери та пристрої зберігання даних. Традиційно сервер має обмежену кількість пристроїв зберігання даних. Натомість SAN надає можливість підключити один або кілька гетерогенних серверів до спільного сховища даних. Мережа може містити багато пристроїв зберігання, зокрема дискові, стрічкові та оптичні накопичувачі. Крім того ці пристрої можуть розташовуватися віддалено від серверів, що їх використовують. Структуру SAN мережі наведено на рис. 7.9.

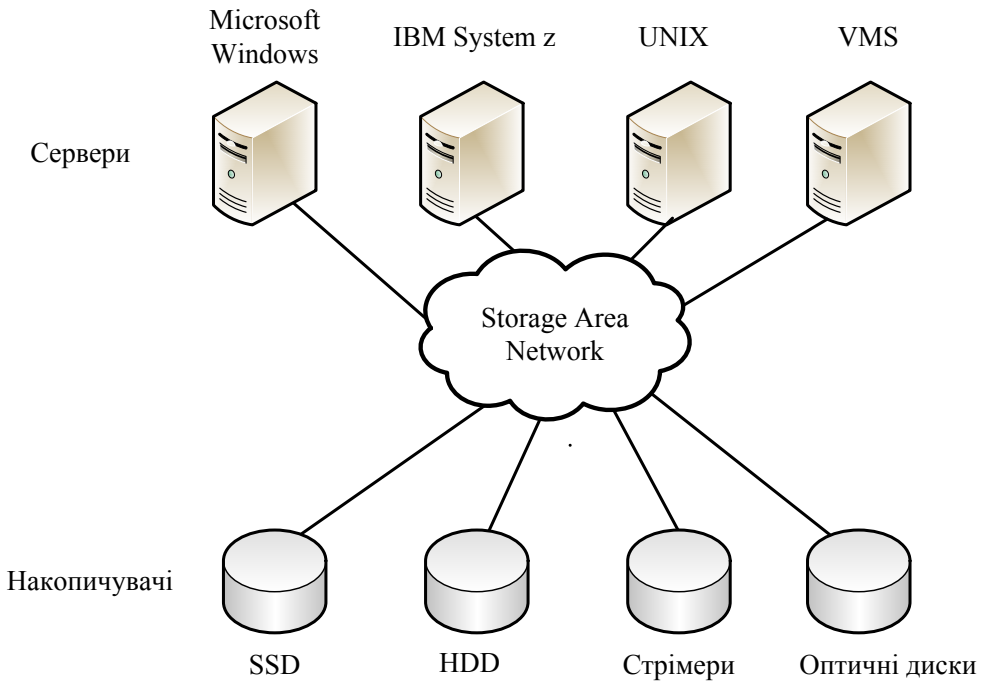


Рисунок 7.9 – Структура SAN-мережі

Основними елементами SAN-мережі є:

- комунікаційна структура, яку реалізовано на базі протоколу Fibre Channel;
- пристрої зберігання (магнітні диски, плівка, оптичні носії);
- сервери (Microsoft Windows, UNIX, IBM VMware vSphere, VMS та інші).

SAN-мережі будуються на основі таких топологій:

- Точка – тточка (Point-to-point);
- Петля з арбітражем (Arbitrated loop);
- Комутована фабрика (Switched fabric).

З'єднання «точка – точка» – це найпростіша топологія. Вона використовується тоді, коли є всього два вузли і майбутнє розширення не прогнозується. Середовище використовується пристроями монопольно з залученням всієї смуги пропускання.

Петля з арбітражем передбачає утворення фізичної петлі до 126 вузлів и працює як загальна шина. Трафік передається в одному напрямку. Арбітр допускає встановлення тільки одного з'єднання в кожний момент часу, в межах якого відбувається обмін даними між відправником та одержувачем. Після завершення з'єднання арбітр дозволяє встановлення наступного. Внаслідок низької пропускної спроможності сьогодні майже не використовується.

Комутована фабрика – це найпопулярніша топологія, яка передбачає використання SAN-комутаторів, які, аналогічно Ethernet-комутаторам, дозволяють одночасне передавання трафіку між окремими портами. Це дає можливість досягти максимальної пропускної спроможності.

7.4 Технології віртуалізації

Однією з найбільш істотних технологічних новацій, що лежать в основі хмарних обчислень, є технології віртуалізації. Віртуалізація – це надання набору обчислювальних ресурсів або їх логічного об'єднання, абстраговано від апаратної реалізації, й забезпечення при цьому логічної ізоляції обчислювальних процесів, що виконуються на одному фізичному ресурсі.

Основні поняття технології віртуалізації

Віртуальна машина – ізольований програмний контейнер, який працює з власною ОС і додатками подібно фізичному комп'ютеру. Віртуальна машина діє так само, як фізичний комп'ютер і містить власні віртуальні ОЗУ, жорсткий диск і мережний адаптер.

Основними особливостями віртуальних машин є: сумісність (віртуальні машини сумісні з усіма стандартними комп'ютерами, віртуальна машина працює під керуванням власної гостьовий оперативної системи і виконує власні застосування); ізольованість (віртуальні машини повністю ізольовані одна від одної, ніби це різні фізичні пристрої); інкапсуляція (віртуальні машини повністю інкапсулюють обчислювальне середовище).

Хостова операційна система – це операційна система, яку встановлено на реальне обладнання. В рамках цієї операційної системи встановлюється програмне забезпечення віртуалізації як звичайна програма.

Емулятор віртуальної машини – це програмне забезпечення, яке встановлюється на хостову операційну систему і складається з монітора віртуальних машин і графічної оболонки.

Монітор віртуальних машин (**Virtual Machine Monitor**, **VMM**) або **гіпервізор** є програмою, що забезпечує всі взаємодії між віртуальним і реальним обладнанням, підтримує роботу однієї або декількох створених віртуальних машин і встановлених гостьових операційних систем. Графічна оболонка забезпечує взаємодію користувача з застосуванням віртуальної машини, дозволяючи налагоджувати віртуальні машини під свої потреби та керувати їх роботою.

Гостьова операційна система – це операційна система, яка встановлюється на створену віртуальну машину. Як гостьова операційна система може бути Microsoft Windows, UNIX, Linux тощо.

При використанні технології віртуалізації отримують ієрархічну структуру взаємодії віртуальних ЕОМ і реальної апаратури. На нижньому шарі цієї ієрархії знаходиться реальне обладнання, керування яким розподіляється між хостовою операційною системою та емулятором віртуальних машин.

Хостова операційна система та емулятор розподіляють між собою ресурси реальної ЕОМ і складають другий рівень ієрархії. Хостова операційна система керує програмами, що на ній запущені, і розподілом між ними ресурсів реальної ЕОМ.

Емулятор віртуальних машин керує віртуальними машинами зі встановленими на них гостьовими операційними системами і розподіляє між ними ресурси реальної ЕОМ.

Гостьові операційні системи керують роботою своїх застосувань в рамках виділених емулятором ресурсів.

Основні різновиди віртуалізації:

- віртуалізація серверів (повна віртуалізація і паравіртуалізація);
- віртуалізація на рівні операційних систем;
- віртуалізація мережі;
- віртуалізація застосувань;
- віртуалізація робочих місць;
- віртуалізація апаратних засобів.

Архітектура сучасних серверів x86 передбачає виконання тільки однієї ОС на сервері. Подолати такі структурні обмеження можна за допомогою віртуалізації серверів x86. Ця технологія абстрагує операційну систему і застосування від рівня фізичного обладнання, що робить середовище серверів менш складним. Завдяки віртуалізації на одному фізичному сервері можна виконувати кілька операційних систем у вигляді віртуальних машин, у кожній з яких є доступ до обчислювальних ресурсів сервера.

Технологія віртуалізації серверів дозволяє запускати на одному сервері декілька логічних одиниць – віртуальних машин, які повністю відтворюють роботу незалежних фізичних серверів. Це дає змогу розміщувати на одній одиниці обладнання кілька десятків незалежних операційних систем і корпоративних застосувань, ефективніше використовуючи IT-інфраструктуру.

В основі рішення віртуалізації на рівні машини лежить монітор віртуальних машин VMM. VMM відповідає за створення та ізоляцію віртуальної машини і збереження її стану, а також за організацію доступу до системних ресурсів. При реалізації VMM для створення інтерфейсу між віртуальними машинами і віртуалізованими системними ресурсами використовуються три можливих методи: повна віртуалізація, власна віртуалізація і паравіртуалізація.

При використанні **повної віртуалізації** (рис. 7.10) монітором VMM (для абстрагування віртуальної машини від реального обладнання) створюється і підтримується повна віртуальна система. Цей підхід дозволяє виконувати у віртуальній машині операційну систему без будь-якої її модифікації. Це дає змогу легко переносити віртуальні машини між серверами з різними фізичними конфігураціями. Така гнучкість досягається ціною втрати продуктивності через накладні витрати на обслуговування станів віртуальних машин і затримок при двійковій трансляції.

Рисунок 7.10 – Модель повної віртуалізації

Власна віртуалізація залежить від архітектури процесора, на якому здійснюється віртуалізація. Наприклад, в серіях AMD-V і Intel VT процесори мають в своїй апаратній частині нові режими виконання, інструкції та структури даних, які призначені для зменшення складності VMM. При власній віртуалізації VMM не потрібно підтримувати в програмному забезпеченні характеристики ресурсів віртуальної машини та її стан. Перевагами власної віртуалізації є спрощення архітектури VMM й істотне підвищення продуктивності.

Паравіртуалізація має деяку схожість з повною віртуалізацією. Цей метод використовує VMM для поділу доступу до основних апаратних засобів, але додає код, що стосується віртуалізації, безпосередньо в операційну систему. Цей підхід усуває потребу в будь-якій перекомпіляції або перехоплюванні команд, що не можуть бути віртуалізовані. Однак паравіртуалізація вимагає змінення гостьової ОС для роботи з VMM, що є основним недоліком методу. Натомість, паравіртуалізація забезпечує високу продуктивність, майже як у реальної системи.

Віртуалізація на рівні операційної системи (рис. 7.11) віртуалізує сервери безпосередньо над операційною системою. Цей метод підтримує єдину операційну систему і, в найзагальнішому випадку, просто ізолює незалежні віртуальні сервери (контейнери) один від одного. Для поділу ресурсів одного сервера між контейнерами необхідним є внесення змін в ядро операційної системи (наприклад, як у випадку з OpenVZ), водночас забезпечується «рідна» продуктивність без «накладних витрат» на віртуалізацію пристроїв. Природно, що такий різновид віртуалізації унеможливлює запуск різних гостьових операційних систем.

Віртуалізація мережі – це повне відтворення фізичної мережі програмним методом. Віртуалізовані мережі аналогічні фізичним мережам з точки зору надійності і можливостей. Однак вони мають безліч таких додаткових експлуатаційних переваг, як незалежність від обладнання, швидка ініціалізація, можливість розгортання без переривання роботи систем, автоматизоване обслуговування та підтримка як сучасних, так і застарілих застосувань. Віртуалізовані мережі забезпечують підключення робочих навантажень до таких логічних мережних пристроїв і служб, як логічні порти, комутатори, маршрутизатори, брандмауери, засоби балансування на-

вантаження, мережі VPN тощо. Застосування у віртуалізованих мережах працюють так само, як і в фізичних.

Рисунок 7.11 – Модель віртуалізації на рівні операційної системи

Віртуалізація застосувань – це технологія, яка спрямована на розділення та ізоляцію застосувань на боці клієнта, що працює з локальною операційною системою. Застосування ізолюються в віртуальному середовищі, що знаходиться між операційною системою і стеком застосувань. Віртуальне середовище завантажується до застосування, ізолює його від інших застосувань і операційної системи, а також запобігає модифікації застосуванням локальних ресурсів (таких, як файли і налагоджування реєстра). Застосування можуть читати інформацію з локального системного реєстра і файлів, але придатні для запису версії цих ресурсів підтримуються всередині віртуального середовища. Дана технологія дозволяє використовувати на одному комп'ютері, а точніше в одній і тій самій операційній системі кілька несумісних між собою застосувань одночасно. Віртуалізація застосувань дозволяє користувачам запускати заздалегідь сконфігуроване застосування або групу застосувань з сервера.

Віртуалізація робочих місць передбачає емуляцію інтерфейсу користувача, тобто користувач бачить застосування і працює з ним на своєму терміналі.

Застосування цієї технології дозволяє відокремити призначене для користувача ПЗ від апаратної частини, а також забезпечити доступ до клієнтських застосувань через термінальні пристрої.

Найбільш популярним різновидом віртуальних робочих місць є віртуалізація робочих столів, що реалізується за допомогою інфраструктури Virtual Desktop Infrastructure (VDI). Віртуалізацію робочих столів може бути реалізовано у статичному або динамічному вигляді. В першому варіанті фізичний настільний комп'ютер замінюється віртуальним, у другому – кінцеві користувачі динамічно підключаються до одного з віртуальних робочих столів з пулу. Віртуальні робочі столи розташовуються на надійних відмовостійких серверах в ЦОД, завдяки чому користувач завжди має повний контроль над усіма своїми ресурсами.

Найскладнішим різновидом віртуалізації є віртуалізація, або точніше **емуляція апаратних засобів**. У цьому методі на хост-системі створюється віртуальна машина, в межах якої емулюється те чи інше обладнання. Цей підхід реалізовано в пакеті GNS (General Network Simulation), який дозволяє здійснювати симуляцію комп'ютерних мереж, побудованих на реальному обладнанні. Користувачі працюють з реальними образами операційних систем мережних пристроїв. Перевагою є можливість отримати досвід роботи з пристроєм за його фізичної відсутності. Головною проблемою емуляції апаратних засобів є суттєве уповільнення виконання програм в такому середовищі (деколи в 100 разів), оскільки кожна команда має моделюватися на основних апаратних засобах.

7.5 Програмно керовані мережі

У стандартній архітектурі сучасних мережних пристроїв виділяють три основних рівні (рис. 7.12):

- **рівень керування пристроєм (management plane)** – неуніфікований та незалежний від виробника рівень для керування пристроєм. Це може бути інтерфейс командного рядка (CLI), вбудований веб-сервер або API та протоколи керування;
- **рівень керування трафіком (control plane)** – набір процедур та алгоритмів, що реалізують базові інтелектуальні функції пристрою, наприклад, побудову таблиць маршрутизації в маршрутизаторі;
- **інфраструктурний рівень (data plane)** – рівень передавання даних, на якому здійснюється безпосередньо передавання трафіку через пристрій.

Рисунок 7.12 – Модель стандартної архітектури сучасних мережних пристроїв

Особливістю стандартного підходу є те, що всі рівні реалізовано автономно в кожному окремому пристрої. Це створює низку обмежень при побудові сучасних мереж на основі таких пристроїв, зокрема статичність структури мережі, складність розподіленого адміністрування великої кількості окремих пристроїв, складність масштабування мережі тощо.

Основна ідея **програмно керованих мереж SDN (software-defined networking)** полягає в тому, щоб:

- відділити керування мережним обладнанням від керування передаванням даних за рахунок розроблення спеціального програмного забезпечення, яке може працювати на окремому обладнанні під контролем адміністратора мережі;
- перейти від керування окремими екземплярами мережного обладнання до керування мережею в цілому;
- створити інтелектуальний програмно-керований інтерфейс між мережним застосуванням та транспортним середовищем мережі.

Архітектура програмно керованих мереж (рис. 7.13) передбачає реалізацію мережі передавання даних таким чином, що рівні керування віддаляються від рівня передавання даних і реалізуються програмно у вигляді спеціального контролера керування мережею. Ця концепція є однією з форм віртуалізації обчислювальних ресурсів.

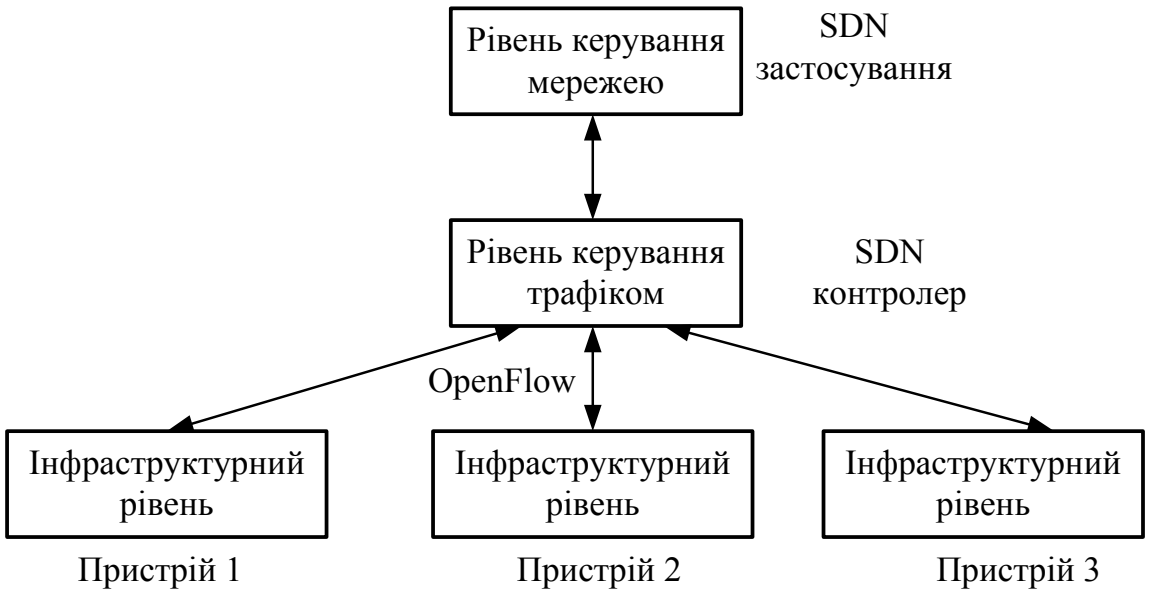


Рисунок 7.13 – Архітектура програмно керованих мереж

Застосування такого підходу має низку переваг, зокрема:

- централізоване управління мережею через SDN-контролер;
- спрощення процесу налагоджування та керування за рахунок переходу від керування мережними пристроями до керування цілими мережами;
- можливості програмування як обладнання (OpenFlow), так і застосувань (API – Контролер SDN);
- швидка реакція на зміни в мережі;
- покращена можливість оптимізації маршруту передавання трафіку, повний контроль над трафіком, що передається;
- істотне скорочення часу розгортання застосувань;

- можливість централізованого застосування політик, збільшення продуктивності, зменшення затримок дозволяє підвищити ефективність взаємодії користувачів і застосувань як в корпоративних мережах, так і в мережах ЦОД.

Як видно з архітектури SDN, що наведена на рис. 7.13, крім класичного керування мережею через команди системного адміністратора до контролера, останній підтримує запуск спеціальних SDN-застосувань керування мережею. Кожне SDN-застосування насправді є інтерфейсом оптимізації мережі під конкретне бізнес-застосування і його основна роль – модифікація мережі в реальному часі під поточні потреби обслуговуваної програми. Це може бути, наприклад, зміна QoS мережі між двома телефонними абонентами, щоб створити належні умови для високоякісного відеодзвінка в реальному часі або створення VPN-тунелю між двома абонентами.

Аналіз інформаційних потоків в архітектурі SDN (рис. 7.14) дозволяє виділити два основних напрямки обміну інформацією: перший – між SDN-застосуваннями і другий для керування фізичними мережними пристроями. Перший потік отримав назву «північний міст», а другий – «південний міст». В ролі «північного мосту» виступає протокол на основі REST (Representational State Transfer) API, а «південний міст» реалізується протоколом OpenFlow.

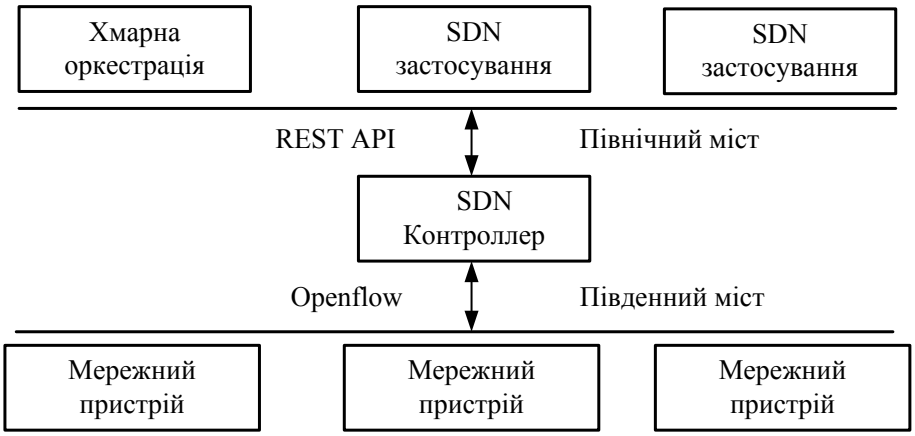


Рисунок 7.14 – Інформаційні потоки в мережі SDN

Openflow – це стандартний протокол, що є основним елементом концепції SDN і забезпечує взаємодію контролера з мережними пристроями. Контролер використовується для керування таблицями потоків комутаторів, на підставі яких приймається рішення про передавання прийнятого пакета на конкретний порт комутатора. Таким чином, в мережі формуються прямі мережні з'єднання з мінімальними затримками передавання даних і необхідними параметрами.

Комутатор OpenFlow (рис. 7.15) зазвичай містить одну або більше таблиць потоків, а також одну групову таблицю і, використовуючи її, контролер може маніпулювати таблицями потоків, встановлювати обмеження на

максимальну кількість потоків тощо. Кожна таблиця потоків складається з набору потоків, а кожен потік містить правила, лічильники та набір інструкцій. Структурні елементи записів у таблиці потоків зображено на рис.7.16.

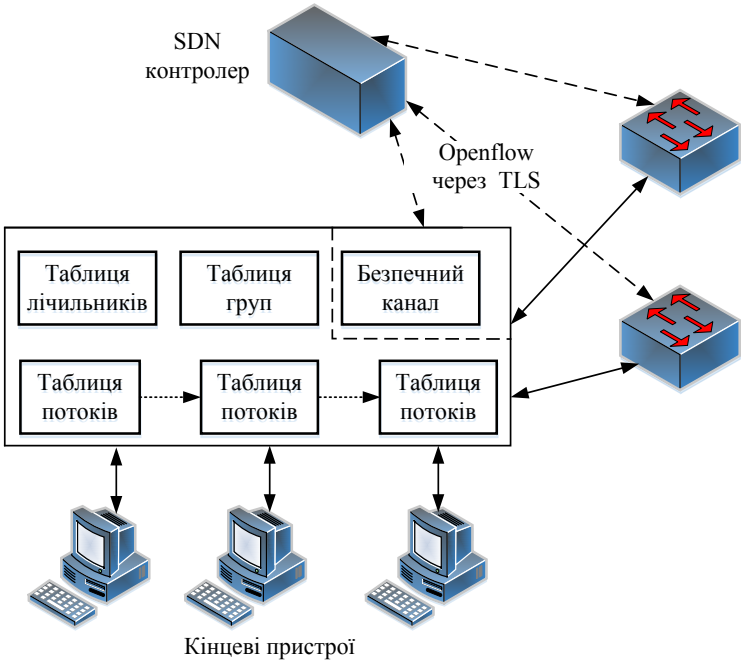


Рисунок 7.15 – Структура комутатора OpenFlow

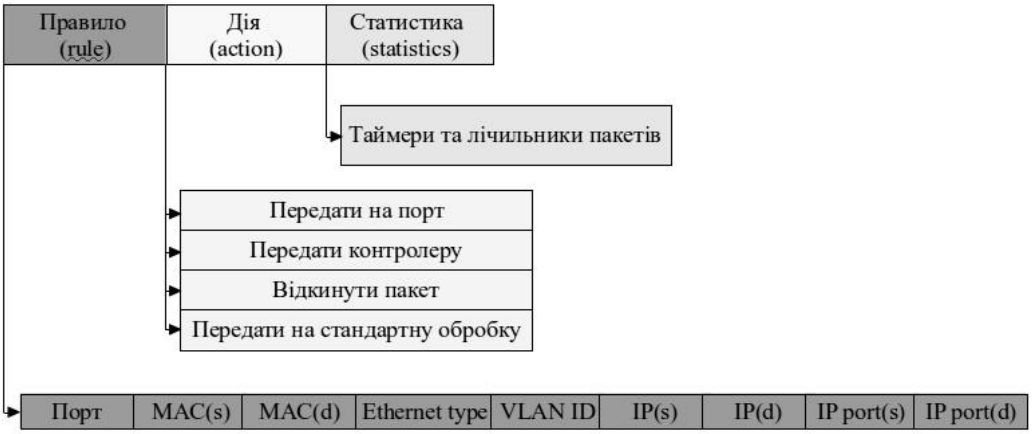


Рисунок 7.16 – Елементи таблиці потоків комутатора OpenFlow

У тому разі, коли на інтерфейс комутатора надходить новий пакет, комутатор відокремлює його заголовок, який утворений протоколами канального, мережного та транспортного рівнів. Після цього комутатор порівнює ці заголовки з усіма потоками, що записані в таблиці потоків. Якщо відповідний потік знайдено, то комутатор виконує інструкції, призначені для цього потоку та оновлює лічильники статистики. До інструкцій належать: передавання пакета,у на вихідний інтерфейс, модифікація полів заголовка, відкидання, широкомовне розсилання, відправлення на контролер.

Всі потоки в таблиці потоків характеризуються пріоритетом. Якщо для одного й того самого пакета знайдено два правила, то комутатор вибирає правило з вищим пріоритетом, і до пакета застосовуються відповідні інструкції.

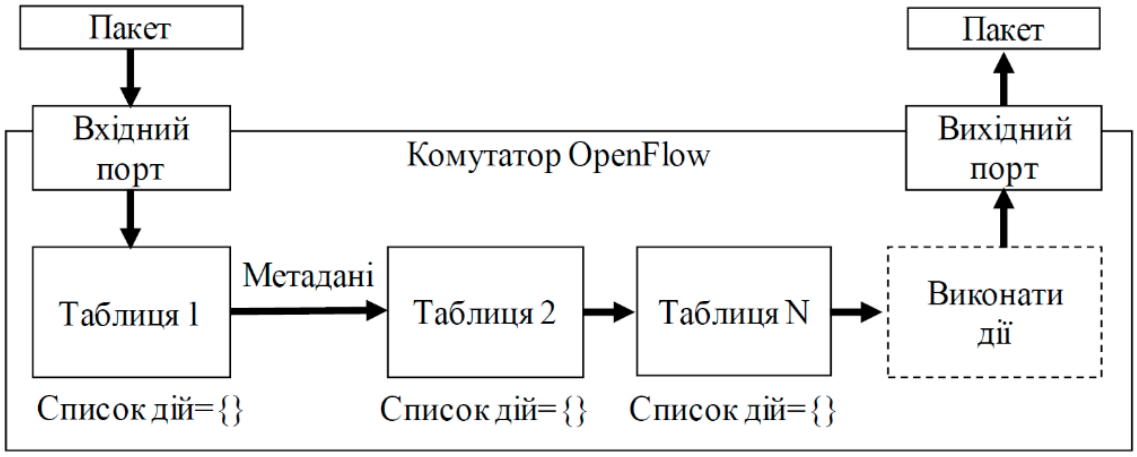


Рисунок 7.17 – Модель гнучкої обробки пакетів на основі каскадування таблиць потоків

Інструкції каскадної обробки забезпечують комутатору можливість пересилати пакети в наступні таблиці потоків для подальшої обробки. Разом з пакетом між таблицями може передаватися службова інформації у форматі метаданих. Каскадна обробка зупиняється тоді, коли набір інструкцій, пов'язаних з відповідним потоком, не вказує на наступну таблицю. Це означає, що з пакетом виконано всі необхідні інструкції, і він передається на вихідний інтерфейс. Як правило, ним є фізичний порт, але він також може бути віртуальним портом, визначеним комутатором або зарезервованим віртуальним портом, визначеним специфікацією.

Зарезервовані віртуальні порти можуть використовуватися для відправлення на контролер, на всі порти або ж для відправлення з використанням традиційних методів обробки. Порти, визначені комутатором як віртуальні, можуть створювати групи агрегації каналів. Потоки можуть також вказувати на групову таблицю, що означає додаткову обробку.

Відповідно до принципу каскадування комутатор починає виконувати пошук пакета в першій таблиці потоків. Поля, що використовуються для зіставлення, залежать від типу пакета. Пакет відповідає запису таблиці потоків, якщо значення полів у пакеті збігається з відповідними полями в правилі потоку. Якщо поле в правилі містить значення ANY(«*»), тоді відповідне поле в заголовку пакета може містити довільне значення.

Така модель обробки пакета відкриває унікальні можливості. Комутатор може використовуватися як в ролі комутатора, так і маршрутизатора чи мережного екрана. Вся функціональність закладена в таблиці потоків, приклад якої з правилами для реалізації різних ролей функціонування комутатора наведено в табл.7.1.

Таблиця 7.1 – Приклади реалізації різноманітних спеціалізованих функцій комутатора за допомогою таблиці потоків

	Порт	Eth src	Eth dst	Eth type	VLAN ID	IP src	IP dst	TCP src	TCP dst	Дія
Ethrnet комутатор	*	00: 1F	*	*	*	*	*	*	*	Порт № 6
Мережний екран (TCP)	*	*	*	*	*	*	*	*	22	Відкинути
IP маршрутизатор	*	*	*	*	*	*	1.0. 0.1	*	*	Порт № 1
OpenFlow комутатор	#3	00: 20	00: 1F	0800	1	1.0. 0.2	1.0. 0.1	27	80	Порт № 4

Таблиця груп містить записи груп, а кожен запис містить також список інструкцій зі специфічною семантикою, що залежить від типу групи. Вказані в записі дії застосовуються для всіх пакетів, що належать до певної групи. Модель комутації з використанням групової таблиці наведено на рис. 7.18.

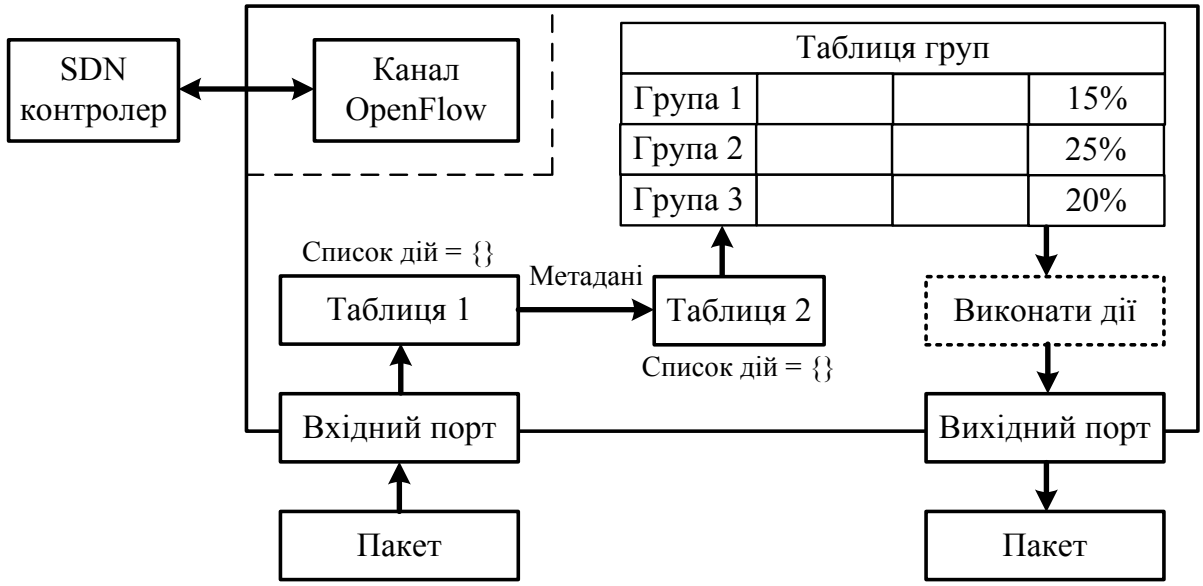


Рисунок 7.18 – Модель комутації з використанням групової таблиці потоків та реалізація балансування навантаження

Збір статистики реалізується за допомогою лічильників. Лічильники можуть бути призначені для кожної таблиці, потоку, порту, черги або групи. Таблиця 7.2 містить набори лічильників, визначені спеціалізацією OpenFlow. OpenFlow-сумісні лічильники можуть бути реалізовані програмно і відображатимуть інформацію на основі опитування апаратних лічильників, які мають більш обмежений діапазон значень.

Кожен потоковий запис містить набір інструкцій, які виконуються в тому разі, коли пакет відповідає правилу. Всі інструкції можна розділити на такі типи:

- для негайного виконання,
- для очищення списку інструкцій,
- для додавання нової інструкції,
- для запису метаданих,
- для переходу до наступної таблиці.

Інструкції для негайного виконання дають вказівку застосувати певні дії негайно, без будь-яких змін у наборі дій. Така інструкція може бути використана для модифікації пакета при передаванні його до іншої таблиці. Інструкції для очищення видаляють всі інструкції з набору інструкцій для окремого потоку. Інструкції для додавання нової інструкції доповнюють наявний набір інструкцій для окремого правила новими інструкціями. Інструкція переходу до наступної таблиці містить номер таблиці, в яку пакет буде переданий після обробки в поточній таблиці. Номер наступної таблиці обов'язково має бути більшим від номера поточної таблиці.

Для кожного потоку визначено список інструкцій, який є за замовчуванням порожнім. Надалі список інструкцій для окремого потоку може змінюватися. Список інструкцій містить максимум одну інструкцію кожного типу. Інструкції в списку застосовуються в порядку, вказаному нижче, незалежно від порядку, в якому вони були додані до списку. Комутатор може підтримувати довільний порядок виконання інструкцій, який визначається реалізацією функцій застосування інструкцій.

Таблиця 7.2 – Список лічильників

Підрахунок	Лічильник	Підрахунок	Лічильник
По таблицях	Підрахунок заголовків	По портах	Отриманих пакетів
	Пакетів-запитів		Переданих пакетів
	Пакетів-відповідей		Отриманих байт
По потоках	Отриманих пакетів		Переданих байт
	Отриманих байт		Отриманих частин
	Тривалість		Переданих частин
			Отриманих помилок
По чергах	Переданих пакетів		Переданих помилок
	Переданих байт		Отримання циклічних помилок
	Переданих помилок перевантаження		Отримання помилок переповнення
			Отримання CRC помилок
			Кількості колізій

Стандартний порядок інструкцій є таким:

1 копіювання зовнішнього TTL,
2 видалення мітки,
3 додавання мітки,
4 копіювання внутрішнього TTL,
5 зниження TTL,
6 застосування всіх дій,
7 застосування всіх маніпуляцій з QoS,
8 застосування всіх групових дій,
9 пересилання пакета у вказаному напрямку.

7.6 Питання для самоперевірки

1. Назвіть основні складові Інтернету речей.
2. На які категорії поділяють пристрої Інтернету речей та яке призначення кожної з них?
3. Які різновиди даних асоціюються з Інтернетом речей?
4. Назвіть основні протоколи, що реалізують підключення M2M.
5. Поясніть призначення основних компонентів топології M2M, яку реалізовано за схемою «видавник-підписник».
6. Опишіть схему зв'язку між сенсорними вузлами з використаним протоколу DDS.
7. Який протокол Інтернету речей дозволяє знизити навантаження на канал, зокрема шляхом організації черг?
8. Назвіть основні переваги, а також ризики застосування хмарних технологій.
9. Які основні типи хмарних послуг визначено сучасними стандартами?
10. В чому полягають особливості мереж, що реалізовані в центрах обробки даних?
11. Які мережні пристрої застосовуються в ЦОД?
12. Назвіть основні задачі, які мають бути реалізовані в ЦОД.
13. Які функціональні блоки містить будь-який ЦОД?
14. В чому полягає багаторівнева архітектура мережної інфраструктури сучасного ЦОД?
15. В чому полягають основні відмінності багатошарової та кластерної моделей ЦОД?
16. Назвіть характерні риси технології SAN.
17. Дайте означення терміна «віртуальна машина», які спільні риси та відмінності між віртуальною та реальною обчислювальними машинами?
18. Що таке «монітор віртуальних машин», які основні функції цього програмного засобу?
19. Назвіть основні різновиди віртуалізації та поясніть особливості кожного з них.

20. Охарактеризуйте основні різновиди віртуалізації серверів та порівняйте їх між собою за критеріями простоти реалізації, ефективності використання обчислювальних ресурсів, підтримки різних операційних систем тощо.

21. Які основні переваги віртуалізації мереж?

22. Порівняйте між собою стандартну архітектуру реалізації мережних пристроїв та програмно керовану.

23. В чому полягають основні переваги застосування програмно керованого підходу порівняно зі стандартним?

24. Які функції виконує протокол OpenFlow в архітектурі програмно керованих мереж?

25. Які таблиці використовуються при керуванні SDN-комутатором?

26. Назвіть основні поля таблиці потоків, що використовується SDN комутатором.

27. В чому полягає каскадування таблиць потоків SDN-комутатора?

28. Які лічильники використовує протокол OpenFlow в своїй роботі?

29. Назвіть основні типи інструкцій, що використовуються в потоковому записі SDN-комутатора.

8 СУЧАСНІ ЦИФРОВІ МЕРЕЖІ

8.1 Ієрархія цифрових каналів

Ієрархією цифрових каналів називають послідовність групоутворення, впорядковану таким чином, щоб забезпечити можливість об'єднувати на одному рівні ієрархії певну кількість цифрових потоків нижнього рівня в один потік, який разом з іншими цифровими потоками такої ж швидкості міг бути мультиплексований у потік більш високого рівня.

Цифрові виділені канали утворюються шляхом постійної комутації в первинних мережах, що використовують апаратуру комутації, яка працює на основі часового ущільнення TDM, розглянутого в підрозділі 2.6. Існують два покоління технологій цифрових первинних мереж:

- технологія плезіохронної цифрової ієрархії **PDH** (Plesiochronous Digital Hierarchy);
- синхронна цифрова ієрархія **SDH** (Synchronous Digital Hierarchy), якій в Америці відповідає стандарт **SONET** (Synchronous Optical Network***).***

8.2 Плезіохронна технологія PDH

Ця технологія була розроблена компанією AT&T наприкінці 60-х років минулого століття для зв'язку великих комутаторів телефонних мереж. Канали з частотним ущільненням FDM, які використовувались до цього часу, вже не могли забезпечити високошвидкісне багатоканальне передавання одним фізичним кабелем.

В технології плезіохронної ієрархії розрізняють три системи стандартизації:

- американська (канали типу T);
- європейська, яка є міжнародною (канали типу E);
- японська (канали типу J).

Ці класифікації відрізняються тільки кратністю ущільнення (мультиплексування) каналів попереднього рівня на відповідному рівні ієрархії. Американська версія розповсюджена на американському континенті, а в Європі використовується міжнародний стандарт. Загальна структура утворення каналів цієї ієрархії наведена на рис. 8.1.

Незважаючи на розбіжності американської, європейської (міжнародної) та японської версій технології PDH, канали відповідного рівня ієрархії швидкостей мають узагальнене визначення DS-n (Digital Signal level n).

Спочатку організацією ANSI були розроблені та стандартизовані принцип формування та апаратура для каналів американської версії T1, які забезпечували передавання даних від 24 абонентів. Для обслуговування

кожного абонента виділяється базовий канал DS-0 з пропускною спроможністю 64 Кбіт/с. Частота стробування каналів дорівнює 8 КГц (тривалість циклу – 125 мкс), тому канал T1 забезпечує передавання кадру, який містить мультиплексовані дані від 24 абонентів (базових каналів). При формуванні кадру DS-1 виконується процедура **байт-інтерлівінгу** (byte-interleaving), тобто на вихід TDM-комутатора послідовно комутується по одному байту від потоку даних кожного абонентського каналу. Для забезпечення синхронізації додається ще один біт F, тому кадр каналу T1 містить 193 біти, що веде до загальної пропускної спроможності 1,544 Мбіт/с (193×8000).

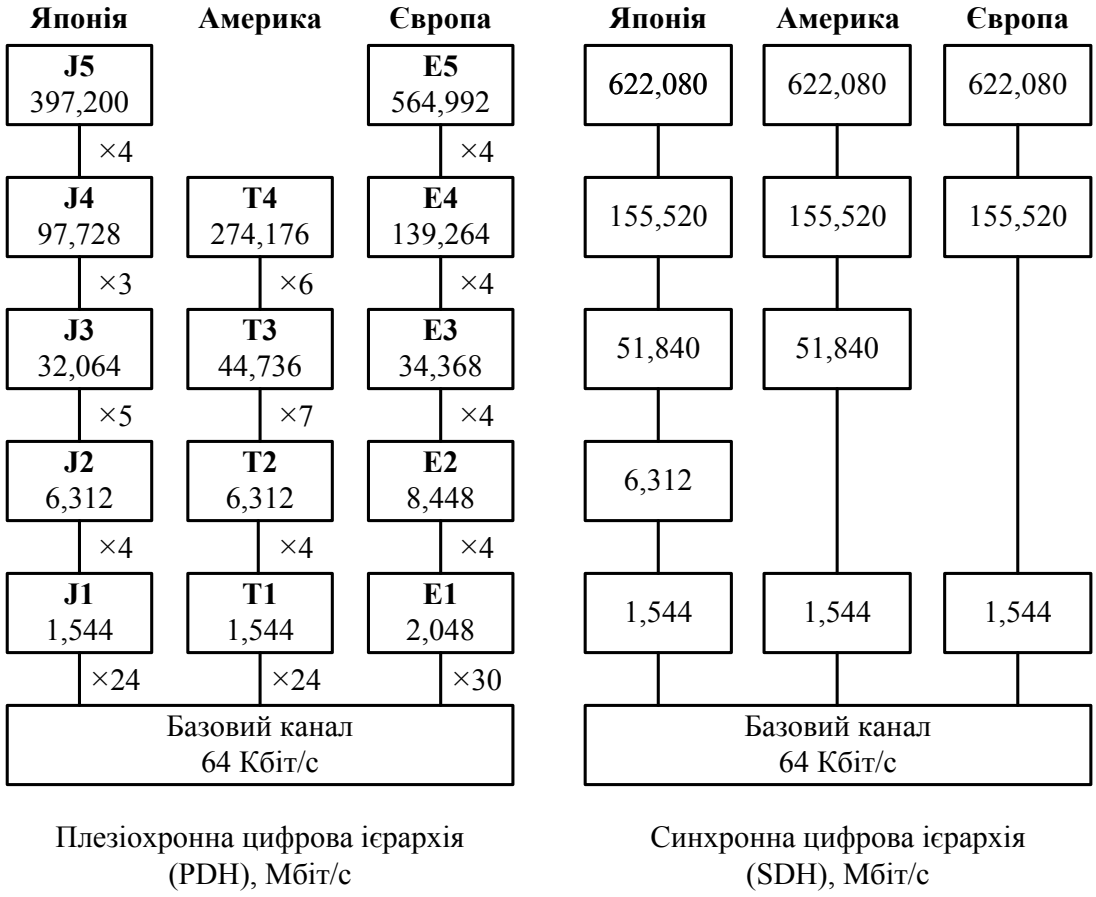


Рисунок 8.1 – Ієрархія цифрових каналів

Для передавання службової інформації використовуються різні підходи, які залежать від типу передаваних даних і покоління апаратури. При передаванні голосового потоку для керівної інформації використовується найменш значущий восьмий біт заміру голосу, тому реальна швидкість передавання даних користувача становить 56 Кбіт/с. Пізніше для службових цілей стали використовувати тільки кожний шостий кадр, тобто в п’ятьох кадрах дані користувача використовують вісім бітів, а в шостому – сім. Така процедура використання восьмого біта називається **«крадіжкою біта»** (bit robbing).

При передаванні комп'ютерних даних в каналі Т1 для даних користувача використовується 23 базових канали, а 24-й відводиться для службових цілей.

Для створення каналу наступного рівня ієрархії Т2 необхідно мультиплексувати (об'єднати) 4 канали Т1, а для створення каналу Т3 – 7 каналів попереднього рівня Т2. При цьому в каналі Т2, в якому передається кадр типу DS-2 і який складається з 4-х послідовних кадрів DS-1, між кадрами DS-1 зберігається біт синхронізації F, а самі кадри DS-2 розділяються 12 службовими бітами, які призначені як для розмежування кадрів, так і для їх синхронізації. Відповідно, кадри DS-3 складаються з семи кадрів DS-2, розділених службовими розрядами.

На рис. 8.1 наведено принцип формування ієрархії цифрових каналів, при цьому в прямокутниках наведено позначення (тип) каналу та його пропускна спроможність в Мбіт/с, а між ними – кратність мультиплексування (кількість) каналів попереднього рівня.

Розроблена американська ієрархія цифрових каналів була стандартизована (з деякими змінами) міжнародною організацією CCITT і описана в стандартах G.700–G.706. Аналогом каналів T_i є канали типу E_i з іншими структурою та швидкостями передавання. Канали Е1 містять 32 байти (таймдомени), з яких 30 використовуються для передавання даних користувача, а 2 – для передавання службової інформації та даних сигналізації між компонентами мережі. Міжнародна система використовує постійне 4-кратне ущільнення каналів попереднього рівня для отримання каналу наступного рівня ієрархії (див. рис. 8.1).

Формати кадрів Т1 та Е1 наведено на рис. 8.2.

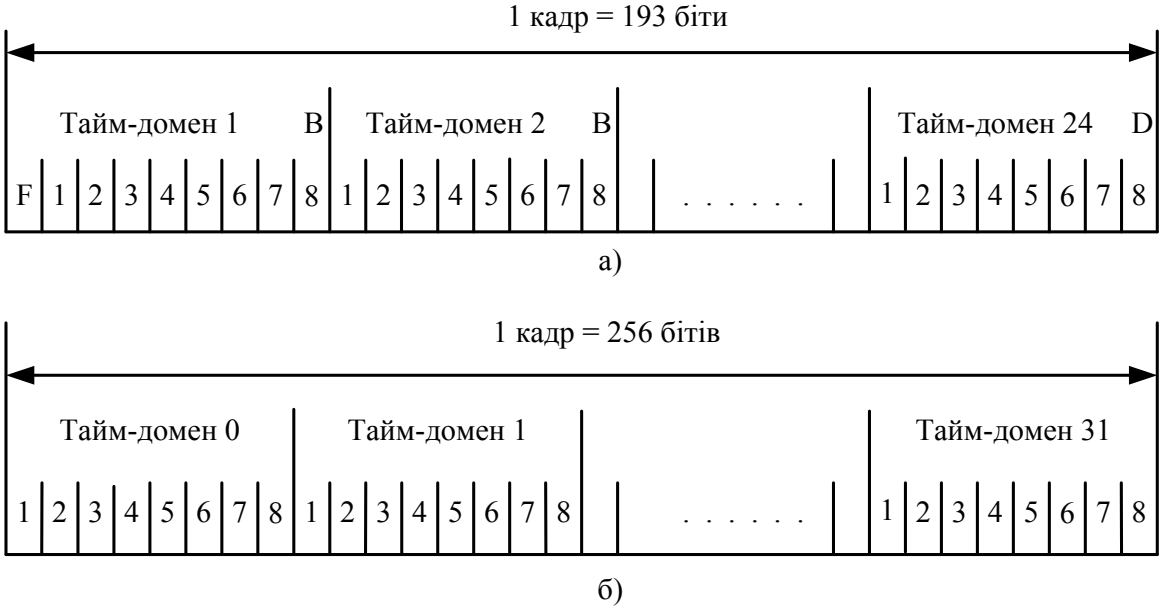


Рисунок 8.2 – Структура кадрів Т1 та Е1 відповідно для американського (а) та європейського (б) стандартів

У таблиці 8.1 наведена характеристика каналів всіх стандартизованих рівнів швидкостей американського та європейського стандартів технології.

На фізичному рівні технології PDH, який визначається стандартом G.703, використовуються різні типи кабелів: скручена пара, коаксіальний та оптоволоконний кабелі. При передавання сигналів у каналах T1 використовується біполярний потенційний код B8ZS, а у каналах E1 – HDB3, принципи формування яких описані в пункті 2.4.3. Для підсилення сигналу в каналах T1/E1 через кожні 1,8 км встановлюються регенератори.

Слід зазначити, що з наведеної ієрархії каналів технології PDH найбільше використання знайшли канали T1/E1 та T3/E3. Канали другого рівня T2/E2 за сукупними затратами на створення та обслуговування не дуже суттєво відрізняються від каналів T3/E3, які мають значно більшу швидкість передавання.

Таблиця 8.1 – Ієрархія цифрових каналів технології PDH

Позначення каналу	Американський стандарт			Європейський (міжнародний) стандарт		
	Кількість голосових каналів	Кількість каналів попередньо-ого рівня	Швидкість, Мбіт/с	Кількість голосових каналів	Кількість каналів по-переднього рівня	Швидкість, Мбіт/с
DS-0	1	1	64 Кбіт/с	1	1	64 Кбіт/с
DS-1	24	24	1,544	30	30	2,048
DS-2	96	4	6,312	120	4	8,488
DS-3	672	7	44,736	480	4	34,368
DS-4	4032	6	274,176	1920	4	139,264
DS-5	—	—	—	7680	4	564,992

Всі версії технології PDH мають суттєві недоліки.

По-перше, одним з основних недоліків є складність процедур мультиплексування та демультиплексування даних користувача, що необхідно виконувати дуже часто. Це випливає з самої «плезіохронності» (тобто майже синхронності) цієї технології, яка передбачає відсутність повної синхронності потоків даних при об’єднанні низькошвидкісних каналів у канали з більшою пропускною спроможністю. В результаті для одержання даних користувача з ущільненого (об’єднаного) каналу необхідно поступово, крок за кроком, повністю демультиплексувати кадри цього каналу. Наприклад, для того, щоб виділити дані абонентського каналу 64 Кбіт/с з кадрів ущільненого каналу E3, необхідно спочатку демультиплексувати їх до рівня кадрів E2, потім – до рівня E1, і лише потім, демультиплексувавши кадри E1, отримати дані користувача. Після цього для можливості передавання всіх даних інших користувачів необхідно виконати зворотні процедури, тобто поступово мультиплексувати потоки до рівня E3. Це призводить як до часових затримок, необхідних для виконання цих перетворень, так і до суттєвого збільшення апаратних затрат. Тому для подолання цього недоліку

в мережах PDH реалізують деякі додаткові прийоми, які ведуть до зменшення кількості операцій демультиплексування при вилученні даних користувача з високошвидкісних каналів. Одним з таких прийомів є «зворотна доставка» (back hauling), яка передбачає реалізацію особливих механізмів взаємодії сусідніх мультиплексорів, що призводить до значного ускладнення роботи всієї мережі і вимагає складного конфігурування її комунікаційних модулів, що, в свою чергу, призводить до помилок внаслідок значного обсягу ручної роботи адміністраторів мережі.

По-друге, в технології PDH практично відсутні розвинені вбудовані процедури керування мережею та контролю передавання потоків. Існуючі службові біти не несуть достатньо інформації про стан каналу, не дозволяють його конфігурувати тощо. Крім того, в технології відсутні процедури підтримки відмовостійкості, що є принципово необхідним для сучасних первинних мереж.

По-третє, в технології PDH стандартизовано швидкості передавання, які є занадто низькими для вимог сучасних застосувань. Крім того, оптоволоконні канали дозволяють передавати дані зі швидкостями в декілька гігабітів за секунду, в той час коли ієрархія швидкостей PDH теоретично завершується рівнем 564 Мбіт/с, а практично не перевищує 100 Мбіт/с.

Враховуючи всі ці недоліки була розроблена нова технологія первинних цифрових мереж – синхронна цифрова ієрархія SDH.

8.3 Синхронна технологія SDH

Технологія синхронної цифрової ієрархії була розроблена компанією Bellcore (**Bell Co**mmunication **Re**search, на сьогодні відома як Telcordia Technologies) під назвою «Синхронні оптичні мережі» SONET (Synchronous Optical NETs) на початку 1980-х років. У 1986 р. у CCITT (зараз ITU**)** розпочалась робота зі створення нового стандарту на основі SONET, яка у 1988 р. привела до затвердження нових Рекомендацій з синхронної цифрової ієрархії SDH (G.707, G.708, G.709), які пізніше були об’єднані в одну Рекомендацію G.707. Основною метою міжнародного стандарту було створення такої технології, яка б дозволила передавати трафіки всіх існуючих цифрових каналів (як американського, так і європейського стандартів), використовуючи високошвидкісні магістральні канали на оптоволоконних кабелях, і забезпечила б ієрархію швидкостей, що продовжує ієрархію технології PDH, до швидкостей в декілька гігабітів в секунду.

Технології SDH і SONET сумісні за апаратурою та стеками протоколів і можуть мультиплексувати вхідні потоки практично будь-якого стандарту PDH, часто вважаються єдиною технологією SDH/SONET, проте мають деякі відмінності. Технологія SDH, порівняно з SONET, має такі характеристики:

- швидкість передавання першого ієрархічного рівня SDH дорівнює 155,520 Мбіт/с (тоді як для мереж SONET втричі менша, а саме: 51,840 Мбіт/с), що дозволяє завантажувати потоки E4 технології PDH (139,263 Мбіт/с);
- на фізичному рівні для стандарту SDH визначено три різні типи середовищ передавання: оптоволоконний і коаксіальний кабелі та радіорелейні системи, тоді як для SONET – оптоволоконний і коаксіальний кабелі.

У стандарті SDH всі рівні швидкостей і типи кадрів для цих рівнів мають загальну назву: **STM-n** (Synchronous Transport Module level n). В технології SONET використовується два позначення для рівнів швидкостей і типів каналів: **STS-n** (Synchronous Transport Signal level n), яка вживається при передавання даних електричним сигналом, і **OC-n** (Optical Carrier level n), яка використовується при передавання даних оптоволоконним кабелем. Формати кадрів STS і OC ідентичні. Треба зазначити, що кадри даних технологій SONET та SDH називають **циклами**, і, починаючи з загального для технологій рівня STS-3/STM-1, їх формати повністю збігаються.

Ієрархія швидкостей при обміні даними між модулями та типи кадрів, які використовуються в технології SDH/SONET, наведені в табл. 8.2. Треба зазначити, що хоча технологія SDH і починається з каналу STM-1 з пропускною спроможністю 155,520 Мбіт/с, однак для полегшення її використання Рекомендацією G.708 було прийнято нульовий рівень синхронної технології STM-0, який повністю збігається за структурою і параметрами з кадрами STS-1 і має втричі меншу пропускну спроможність.

Таблиця 8.2 – Канали технології SDH/SONET

Кадри технології SONET (оптичний носій)	**Кадри технології SONET**	**Кадри технології SDH**	**Корисна пропускна спроможність (Мбіт/с)**	**Загальна пропускна спроможність (Мбіт/с)**
OC-1	STS-1	(STM-0)	48,960	51,840
OC-3	STS-3	STM-1	150,336	155,520
OC-12	STS-12	STM-4	601,344	622,080
OC-24	STS-24	STM-8	1202,688	1244,160
OC-48	STS-48	STM-16	2405,376	2488,320
OC-96	STS-96	STM-32	4810,752	4976,640
OC-192	STS-192	STM-64	9621,504	9953,280
OC-768	STS-768	STM-256	38486,016	39813,120
OC-1536	STS-1536	STM-512	76972,032	79626,120
OC-3072	STS-3072	STM-1024	153944,064	159252,240

Для побудови мереж SDH/SONET використовуються модулі, які відрізняються за призначенням та функціями, що їх вони виконують (на практиці іноді складно провести чітку межу між цими пристроями, оскільки використовуються багатофункціональні пристрої):

- **термінальні пристрої T** (Terminal) або сервісні адаптери SA (Service Adapter), які приймають дані користувача від низькошвидкісних каналів технології PDH (типу T1/E1 або T3/E3) і перетворюють їх у кадри STS-n (або STM-n);
- **мультиплексори MUX** (Multiplexers), які приймають дані від термінальних пристроїв і мультиплексують потоки кадрів швидкостей STS-n у кадри більш високої ієрархії STS-m (m > n);
- **мультиплексори «введення-виведення» Add-Drop MUX** (Add-Drop Multiplexers) – основні пристрої мережі, які можуть приймати і передавати транзитом потоки певної швидкості STS-n, вставляючи або видаляючи «на ходу», без повного демультиплексування, дані користувача, що прийняті з низькошвидкісних каналів;
- **цифрові кросс-конектори DCC** (Digital Cross-Connect), які називають також апаратурою оперативного переключення (АОП) і є різновидом мультиплексора; призначені для мультиплексування та постійної комутації високошвидкісних потоків STS-n різного рівня між собою і утворюють магістраль мережі SDH/SONET;
- **регенератори сигналів Reg** (Regenerator), які використовуються для відновлення форми s потужності сигналів, які передаються в кабелі.

Узагальнена структура мережі наведена на рис. 8.3.

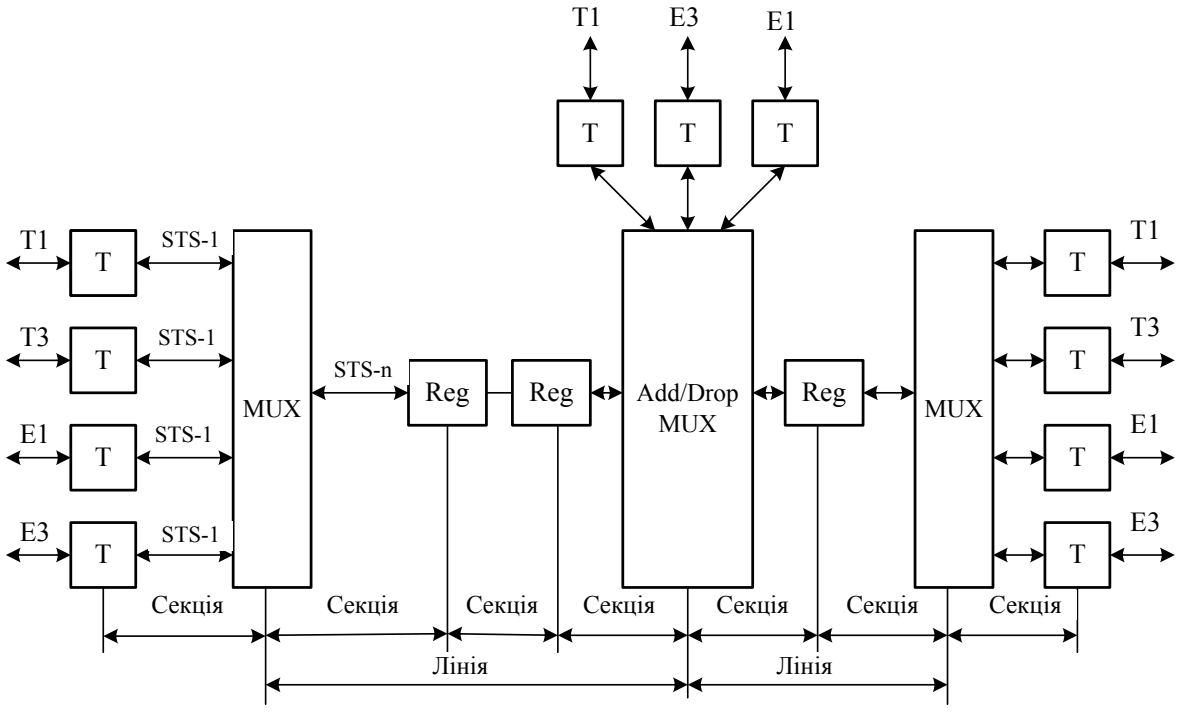


Рисунок 8.3 – Узагальнена структура мережі SDH/SONET

Стек протоколів мережі SDH/SONET має 4 рівні. В різних модулях мережі реалізується свій стек протоколів (рис. 8.4).

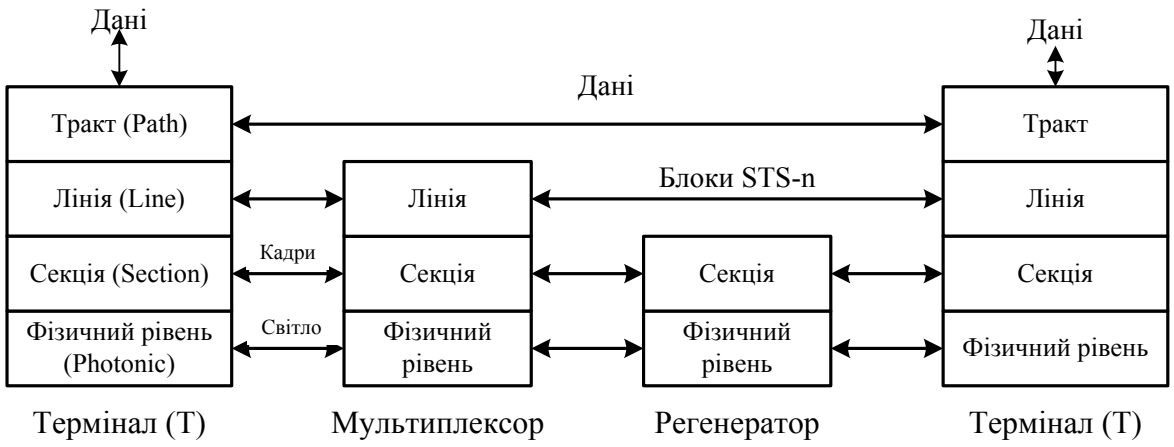


Рисунок 8.4 – Стек протоколів

Фізичний рівень (або фотонний) забезпечує кодування інформації за допомогою модуляції світла. Для кодування використовується метод NRZ, недоліки якого нівелюються завдяки наявності зовнішньої тактової частоти.

Рівень секції підтримує фізичну цілісність мережі і забезпечує передавання кадрів, використовуючи службову інформацію, яка виконується для тестування секції та керування мережею. Заголовок секції завжди починається з двох байтів: 11110110.00101000, які є синхросигналами початку кадру, а наступний байт визначає рівень кадру: STS-1, STS-2 тощо. **Секцією** в технології SDH/SONET називають неперервний відрізок кабелю, який з'єднує будь-які пристрої мережі.

Рівень лінії забезпечує передавання даних між двома будь-якими мультиплексорами мережі. **Лінією** називають потік кадрів одного рівня між двома мультиплексорами мережі. Протокол цього рівня не тільки забезпечує обробку кадрів різних рівнів STS-n для виконання операцій мультиплексування і демультиплексурування, вставляння та вилучення даних користувача, а також виконує реконфігурацію лінії у випадку відмови будь-якого її елемента.

Рівень тракту відповідає за доставку даних між кінцевими користувачами мережі. **Трактом** називають складене віртуальне з'єднання між користувачами. Протокол цього рівня забезпечує прийом даних в форматі технології PDH і їх перетворення у кадри STS-n/STM-m.

Базовий формат кадру STM-1 (рис. 8.5) складається з 2430 байтів (для STS-1 – з 810 байтів), розміщених на інтервалі між двома синхросимволами, частота яких становить 8 КГц (тривалість циклу – 125 мкс). Кожний байт кадру на визначеній позиції в межах циклу може передавати дані одного телефонного каналу або, що еквівалентно, одного цифрового каналу з

пропускною спроможністю 64 Кбіт/с. Для зручності цикли сигналів подають у вигляді прямокутної матриці. Число рядків матриці завжди дорівнює 9, а кількість стовпців залежить від ієрархічного рівня STM і складає 270 × N (де N – номер рівня).

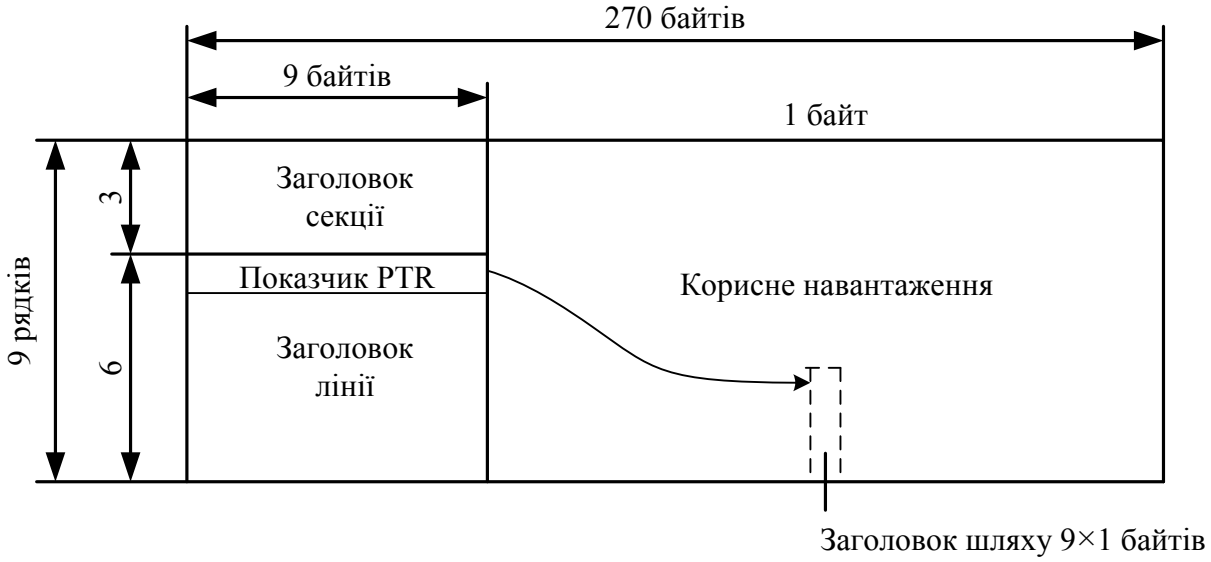


Рисунок 8.5 – Структура кадру STM-1

Кожен кадр має заголовок **OH** (Overhead), який складається з секційного заголовка **SOH** (Section Overhead) і покажчиків адміністративних блоків **PTR** (Pointer), та інформаційне поле корисного навантаження. Секційний заголовок, в свою чергу, містить заголовок секції регенерації **RSOH** (Regenerator Section Overhead), інформація якого обробляється регенераторами і використовується для контролю та реконфігурації секції, та заголовок лінії або секції мультиплексування **MSOH** (Multiplex Section Overhead), службова інформація якого використовується в модулях, які формують і розформовують синхронні транспортні модулі, тобто використовуються для контролю, керування лінією та для її реконфігурації. Покажчики **PTR** ідентифікують перший байт віртуального контейнера, тобто, де в полі корисного навантаження реально починається корисне навантаження. Всього використовується три покажчика по три байти кожний.

Як було вказано раніше, кадр STM-1 містить 2430 байтів, з яких для корисного навантаження використовується 2349 (261 × 9). Таким чином, корисна пропускна спроможність циклу становить 150,336 Мбіт/с. Загальна і корисна пропускні спроможності кадрів STM-n наведені в табл. 8.2.

Байти матриці передаються рядками, зліва направо, у кожному байті першим передається найбільш значущий біт (рис. 8.6). Цикли синхронних транспортних модулів вищих рівнів формуються з використанням байт-синхронного мультиплексування циклів STM нижчих рівнів (рис. 8.7).

Потрібно зазначити, що в поле корисного навантаження циклів STM можуть завантажуватись не тільки кадри PDH, але і комірки ATM, кадри Ethernet, пакети IP тощо.

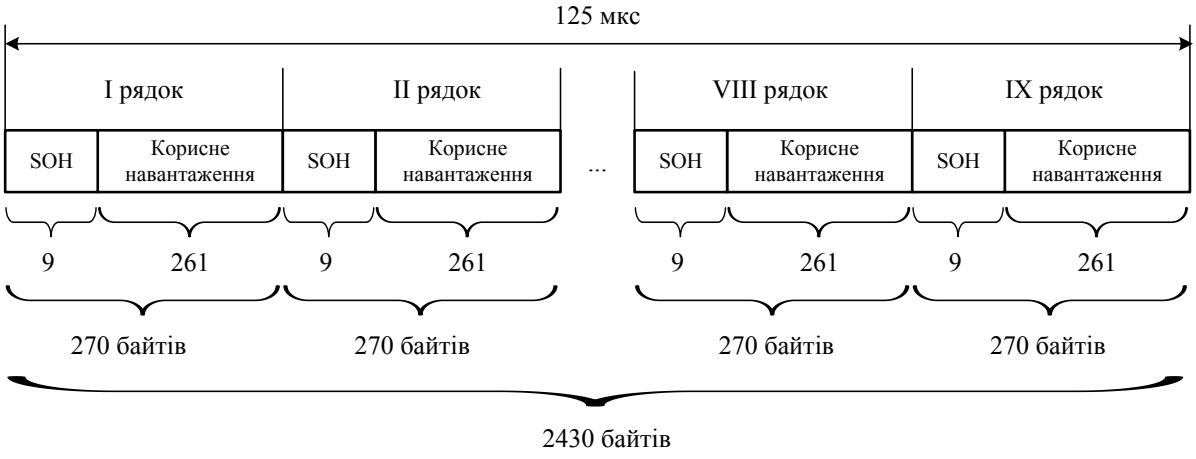


Рисунок 8.6 – Цифровий потік кадру STM-1
(структура циклу у розгорнутому вигляді)

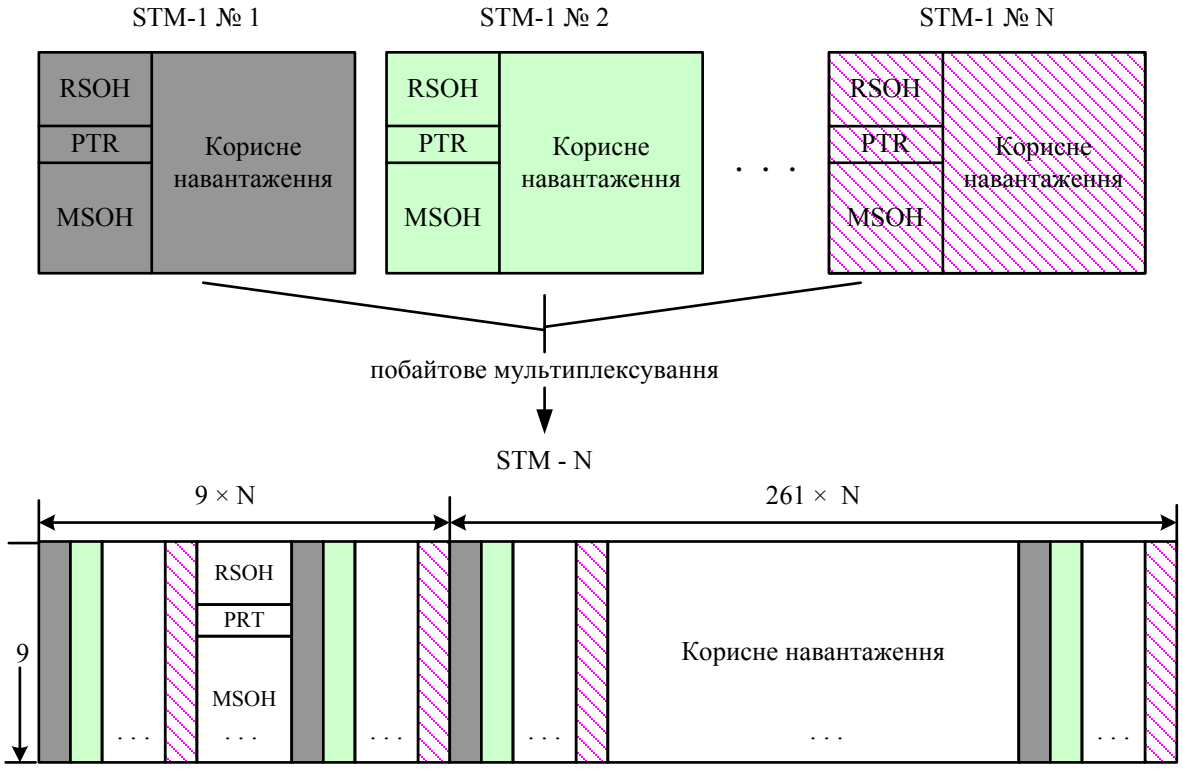


Рисунок 8.7 – Принцип формування кадру STM-n

Відмовостійкість мереж SDH/SONET вбудована в її основні протоколи. Існують 2 способи її реалізації: схеми 1:1 та 1:n. Перший варіант передбачає, що для одного робочого каналу (і порту, який його обслуговує) призначається додатковий резервний канал. Дані одночасно

передаються двома каналами, а у разі виявлення помилок визначається, який канал має бути робочим. Реалізація відмовостійкості мережі за схемою 1:n передбачає, що для захисту n робочих каналів вводиться тільки один резервний, який при виявленні помилок в роботі основного каналу замінює його.

Керування, конфігурування та адміністрування мереж SDH/SONET також вбудовано в протоколи. Службова інформація протоколу дозволяє централізовано і дистанційно конфігурувати шляхи між кінцевими користувачами мережі, змінювати режим комутації потоків, а також збирати статистику про роботу мережі.

8.4 Мережі ISDN

Мережі **ISDN** (Integrated Services Digital Network) – це цифрові мережі, що надають комплексні послуги при передаванні голосових повідомлень, відео, даних та іншої інформації в одному каналі. Перші стандарти та рекомендації серії I.x для мереж ISDN з'явились в 1984 році, але тільки в середині 1990-х років ці мережі стали використовуватись на практиці, оскільки саме в цей період з'явились комунікаційні пристрої з підтримкою сервісів ISDN.

Архітектура мережі ISDN забезпечує реалізацію декількох сервісних служб:

- виділені цифрові канали (некомутовані канали);
- комутована телефонна мережа загального використання;
- комутація каналів;
- комутація пакетів;
- ретрансляція кадрів (Frame Relay);
- контроль і керування роботою мережі.

При цьому основним способом комутації в мережах ISDN є комутація каналів, а дані обробляються в цифровій формі. На практиці, однак, не обов'язково всі мережі підтримують всі стандартні служби. Наприклад, ретрансляція кадрів, хоча і була розроблена для мереж ISDN, на сьогодні реалізується як незалежна мережа на основі сукупності комутаторів.

Базова швидкість мережі ISDN – це швидкість каналу DS-0 з пропускною спроможністю 64 Кбіт/с, але використання диференціального кодування дозволяє передавати голосові потоки з тією ж якістю зі швидкістю 32 Кбіт/с та 16 Кбіт/с.

Цифровий зв'язок між центральним комутатором і користувачем ISDN підтримує одночасне передавання декількох каналів з часовим розподілом. В стандартах ISDN визначають базові типи каналів, з яких формують різні інтерфейси користувача. Типи каналів, які стандартизовано відповідними рекомендаціями, наведено в табл. 8.3.

Таблиця 8.3 – Типи каналів технології ISDN

Тип каналу	Пропускна спроможність	Призначення
A		Аналогова телефонна лінія 4 КГц.
B	64 Кбіт/с	Цифровий канал з кодово-імпульсною модуляцією для передавання голосу та масивів даних, інтерактивний обмін даними, відео з низькою роздільною здатністю.
C	8 Кбіт/с або 16 Кбіт/с	Передавання даних.
D	16 Кбіт/с або 64 Кбіт/с	Канал позаканальної сигналізації, керування іншими каналами.
E	64 Кбіт/с	Внутрішня сигналізація ISDN.
H0	384 Кбіт/с	Передавання цифрових даних, високоякісне аудіо.
H10	1472 Кбіт/с	Передавання цифрових даних, високоякісне аудіо.
H11	1536 Кбіт/с	Передавання цифрових даних, теле- та відеоконференції.
H12	1920 Кбіт/с	Передавання цифрових даних, теле- та відеоконференції.
H4	до 150 Мбіт/с	Інтерактивне відео.

На практиці найчастіше використовуються канали типу B и D.

Канали типу **B (**Bearer – носій) забезпечують передавання даних користувача будь-якого типу: комп'ютерних даних, голосового потоку тощо.

Канали типу **D** (Device – пристрій) використовуються, по-перше, для передавання адресної інформації, на основі якої виконується комутація каналів типу B, і по-друге, в разі її відсутності – для передавання даних користувача в режимі комутації пакетів.

Канали типу **H** (Higher rate channel – канал підвищеної швидкості) використовуються, якщо ресурсів B-каналу не вистачає для коректної роботи даного застосування користувача. Канали типу H утворюються об'єднанням необхідної кількості каналів і забезпечують швидкість передавання, яка необхідна для даного прикладного сервісу. Наприклад, канал H0 з загальною пропускною спроможністю 384 Кбіт/с утворено об'єднанням шести каналів типу B, а канал H11 пропускної спроможності 1,536 Мбіт/с – об'єднанням чотирьох каналів H0.

Розрізняють 2 типи мереж ISDN: вузькосмугові мережі **N-ISDN** (Narrowband ISDN), які використовують канали з базовою пропускною спроможністю 64 Кбіт/с, тобто B, D та H, і широкосмугові мережі **B-ISDN** (Broadband ISDN), які підтримують швидкість передавання порядку 622 Мбіт/с, що не може бути забезпечена каналами логічної групи H. Служби B-ISDN розподілено в дві групи: слабозв'язні комунікаційні служби (communications services), які аналогічні традиційним телефонним діалоговим службам, і служби спілкування (conversational services), які забезпечують двостороннє передавання інформації (відеоконференції) або високошвидкісне передавання даних. Засоби B-ISDN використовують технологію комутації комірок, яку називають режимом асинхронного передавання **ATM** (Asynchronous Transfer Mode), який буде розглянуто пізніше.

Інтерфейси доступу забезпечують підключення користувачів мережі ISDN до самої мережі і складаються з логічно згрупованих каналів, які надаються користувачу мережею або постачальником послуг. Ці інтерфейси визначають допустимі швидкості передавання й відрізняються кількістю каналів B, D та H в даному інтерфейсі. Стандарти визначають два типи інтерфейсів доступу:

- **BRI** (Basic Rate Interface) – інтерфейс базового рівня або інтерфейс передавання даних з номінальною швидкістю;
- **PRI** (Primary Rate Interface) – інтерфейс основного рівня або інтерфейс передавання даних з основною швидкістю.

Інтерфейс передавання даних з номінальною швидкістю BRI (рекомендація I.430) надає користувачу два канали типу B для передавання даних та один канал типу D з пропускною спроможністю 16 Кбіт/с для передавання керівної інформації, завдяки цьому цей інтерфейс визначають як 2B+D. Всі канали – дуплексні. Таким чином BRI забезпечує загальну швидкість передавання 144 Кбіт/с в кожному напрямку, а з урахуванням додаткових бітів, які не несуть інформацію, фізична швидкість становить 192 Кбіт/с. Сферою використання інтерфейсу BRI є невеликі компанії і персональні (домашні) мережі. Треба зазначити, що інтерфейс BRI, залежно від потреб і вимог користувача, може підтримувати не тільки зазначену схему, а й B+D та просто D.

Інтерфейс передавання даних з основною швидкістю PRI (рекомендація I.431) підтримує схеми 30B+D або 23B+D (відповідно для європейської та американської версій), при цьому канал D має швидкість передавання 64 Кбіт/с. Загального варіанта інтерфейсу PRI не існує, оскільки швидкості найбільш поширених каналів, з одного боку, в Європі (E1), а з іншого, в Америці (T1) та Японії (J1), не збігаються. Залежно від вимог користувачів існують й інші схеми інтерфейсу PRI, які використовують меншу кількість каналів B, наприклад, 20B+D або 18B+D. Якщо користувач використовує декілька інтерфейсів PRI, то всі вони можуть мати тільки один канал D, а кількість каналів даних B відповідно збільшується.

Інтерфейс PRI може бути реалізовано і на каналах типу H, однак при цьому загальна пропускна спроможність не має перевищувати 2,048 Мбіт/с або 1,544 Мбіт/с для відповідних версій. Таким чином, можливі, наприклад, схеми інтерфейсів PRI 3HO+D для американської версії та 5HO+D для європейської.

Кадри інтерфейсів PRI мають структуру кадрів DS-1 для каналів T1 або E1 (залежно від сфери використання).

Інтерфейси BRI та PRI отримує користувач, а самі магістралі ISDN реалізуються і підтримуються постачальником послуг. Стандарти ISDN визначають інтерфейс «користувач – мережа» (або «абонент – мережа») як сукупність **пристроїв різних функціональних груп** і **опорних точок**, які є логічними точками взаємодії між цими функціональними групами.

Підключатися до мережі ISDN можуть будь-які пристрої: як фізичні модулі ISDN, так і віртуальні. Всі пристрої залежно від функцій, які вони реалізують, відносять до однієї з нижченаведених функціональних груп.

1. Кінцева станція типу 1 **NT1** (Network Termination type 1) – це фізичний пристрій інтерфейсу користувача ISDN, який виконує функції першого рівня моделі OSI: фізичне з'єднання між пристроями користувача і мережею, обслуговування і моніторинг лінії. NT1 підтримує декілька каналів в інтерфейсах BRI та PRI і виконує їх мультиплексування в режимі TDM.
2. Кінцева станція типу 2 **NT2** (Network Termination type 2) реалізує функції перших трьох рівнів моделі OSI. Станції NT2 – це комунікаційні пристрої (шлюзи локальних мереж, комутатори тощо).
3. Термінальне обладнання типу 1 **TE1** (Terminal Equipment type 1) являє собою будь-який модуль кінцевого користувача, який підтримує протоколи і сервіси мережі ISDN.
4. Термінальне обладнання типу 2 **TE2** (Terminal Equipment type 2) – це пристрої кінцевого користувача, несумісні з мережею ISDN.
5. Термінальний адаптер **TA** (Terminal Adapter) дозволяє підключати пристрої, що не підтримують протоколи ISDN, до мережі.

Підключення обладнання користувача до мережі ISDN реалізується відповідно до стандартної схеми, яка запропонована міжнародною організацією CCITT (рис. 8.8). Всі пристрої, враховуючи функції, що їх вони виконують, відносять до конкретної функціональної групи, кожна з таких груп підключається за допомогою своїх інтерфейсів, які називаються **опорними точками** (**reference points**). Кожний інтерфейс з'єднання реалізується конкретними протоколами.

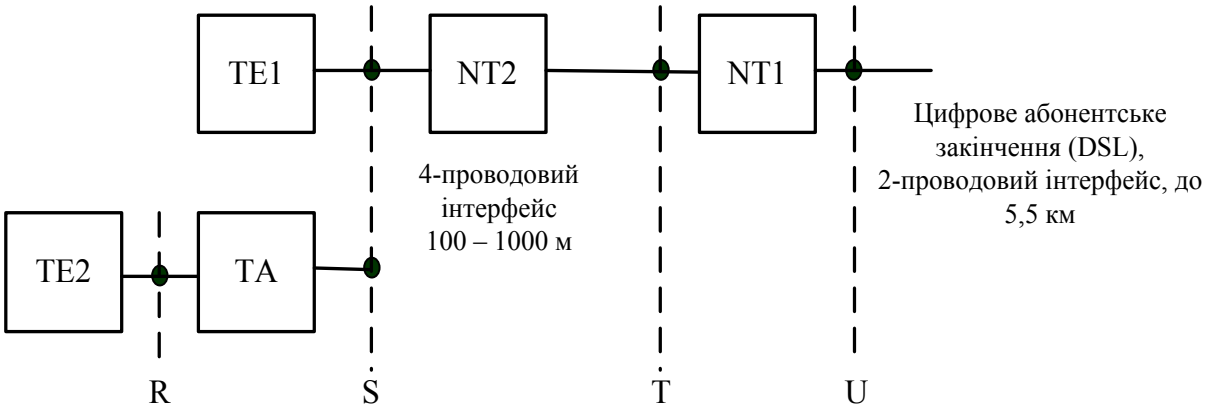


Рисунок 8.8 – Підключення обладнання користувача ISDN

Стандарти визначають 4 найбільш важливі опорні точки ISDN:

- **R** визначає інтерфейс між кінцевими пристроями TE2, які не підтримують ISDN, і термінальними адаптерами TA;

- **S** визначає інтерфейс «користувач – мережа» між термінальним обладнанням TE1 або термінальним адаптером TA і будь-яким кінцевим пристроєм NT1 або NT2;
- **T** описує інтерфейс між локальним пристроєм NT2 і кінцевим модулем абонентської лінії NT1;
- **U** визначає стандарт комунікацій між обладнанням NT1 та місцевою телефонною мережею (зазвичай в цьому інтерфейсі використовується код 2B1Q).

Для реалізації міжмережної взаємодії, а також підключення до ISDN-мереж з іншим принципом функціонування в стандартах описуються додаткові опорні точки і відповідні їм протоколи:

- **K** визначає інтерфейс мережі ISDN з будь-якою мережею, яка функціонує на основі інших принципів та протоколів, міжмережне з'єднання при цьому реалізується мережею ISDN;
- **L** аналогічна опорній точці K, але міжмережне з'єднання виконує не ISDN, а інша мережа;
- **M** описує інтерфейс зі спеціалізованими функціями адаптації мережі, що відрізняється від ISDN, які реалізовано в цій мережі;
- **N** – це інтерфейс між двома мережами ISDN, при цьому протоколи визначають сумісність служб взаємодійних мереж;
- **P** визначає спеціалізоване з'єднання в ISDN, яке забезпечує доступ до окремих компонентів.

Для передавання в мережі ISDN використовуються три типи сервісів:

- комутація каналів у B-каналі;
- комутація пакетів у B-каналі;
- комутація каналів у D-каналі.

Канали типу B, які використовуються для передавання даних, можна використовувати для комутації каналів, комутації пакетів, а також організації напівпостійних каналів. D-канал використовується для обміну повідомленнями між користувачем і мережею при встановленні з'єднання, керуванні передаванням та при моніторингу ресурсів мережі.

У мережі ISDN визначають два стеки протоколів: для каналів типів B та D (рис. 8.9).

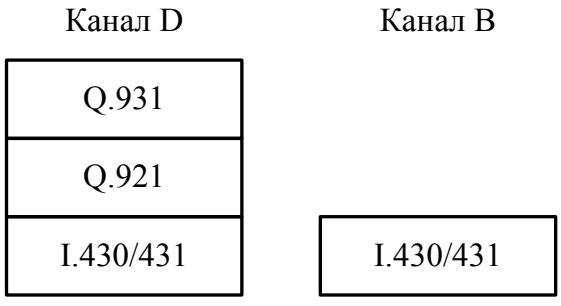


Рисунок 8.9 – Стеки протоколів каналів мережі ISDN

Канали типу B, для яких визначено тільки протокол фізичного рівня, створюють мережу з комутацією цифрових каналів. Комутація (створення складеного каналу між взаємодійними абонентами) виконується на основі вказівок, які отримано з каналів D.

Для каналів типу D визначено три нижніх рівні протоколів. Фізичний рівень описано в стандартах I.430/431 (залежно від типу інтерфейсів доступу), на канальному рівні використовується протокол LAP-D, який визначається стандартом Q.921, а на мережному рівні зазвичай використовується протокол Q.931, який стандартизує маршрутизацію виклику абонента. Протокол LAP-D є модифікацією протоколу HDLC, враховує особливості використання D-каналів і забезпечує передавання як з встановленням з'єднання між абонентами, так і без нього. Відмінність структури кадру LAP-D від кадру HDLC полягає в тому, що поле адреси містить 2 байти, перший з яких визначає код служби, для якої передаються дані пакета, а другий використовується для адресації термінала користувача.

Сукупність каналів типу D дозволяє організувати в мережі ISDN централізовану **систему сигналізації SS7 (**Signal System Number 7). Ця система була розроблена для внутрішнього моніторингу ресурсів мережі, керування комутаторами мережі загального призначення, а також для підтримки каналу між абонентськими станціями. І хоча служба SS7 належить до прикладного рівня, кінцевому користувачу її послуги недоступні, а повідомлення SS7 передаються тільки між комутаторами мережі.

Треба зазначити, що на сьогодні існує декілька різновидів протоколів ISDN, наприклад, National ISDN-1 (США), AT&T Custom; Euro-ISDN (Net3) та інші.

Згідно з означенням CCITT архітектура мережі ISDN містить 4 площини:

- площина керування **C** (Control, C-plane);
- площина користувача **U** (User, U-plane);
- транспортна площина **T** (Transport, T- plane);
- площина адміністрування **M** (Management, M- plane).

Протоколи C-площини реалізують встановлення та завершення з'єднання, керування запитами на інформаційний канал. Протоколи площини U визначають передавання інформації між прикладними сервісами (застосуваннями) користувача. Протоколи T-площини керують фізичними з'єднаннями, а протоколи площини адміністрування M контролюють взаємодію як в самій площини, так і зі з'єднаннями.

Таким чином, основними **перевагами** мереж ISDN є:

- можливість передавання одними каналами як голосових повідомлень, так і комп'ютерних даних в цифровому форматі;
- висока швидкість передавання інформації та надійність зв'язку;
- висока якість передавання голосових повідомлень;

- скорочення часу встановлення з'єднання завдяки використанню виділеного каналу сигналізації та передавання в ньому сигналів керування, моніторингу та взаємодії станцій;
- можливість проведення аудіо- та відеоконференцій між абонентами мережі, які територіально віддалені;
- надання додаткових послуг, наприклад, ідентифікація абонентів, передавання та переадресація викликів, блокування вхідних викликів тощо.

8.5 Мережі Frame Relay

Спочатку мережа **Frame Relay** (FR або FRN – Frame Relay Network) розроблялась як окремий сервіс для використання в мережах ISDN і є мережею з комутацією кадрів (мережею з ретрансляцією кадрів), яка орієнтована на використання цифрових каналів достатньо високої якості. Перша рекомендація була представлена у 1984 році Міжнародним союзом телекомунікацій ITU, а з 1990 року почала активно розвиватися завдяки створенню компаніями Cisco Systems, StrataCom, Northern Telecom та Digital Equipment Corporation консорціуму, основною метою якого є концентрація зусиль з розвитку даної технології. На сьогодні розробкою стандартів та дослідженням технології FR займаються 3 організації:

- Frame Relay Forum (FRF) – міжнародний консорціум, який охоплює більше 300 організацій, серед яких Cisco, 3Com, Digital, Zilog, Newbridge Networks та інші;
- американський національний інститут стандартів American National Standards Institute (ANSI);
- міжнародний союз телекомунікацій International Telecommunication Union (ITU).

Основна відмінність мережі Frame Relay від звичайної комп'ютерної мережі полягає в способі корекції помилок. В комп'ютерній мережі комунікаційний вузол обов'язково перевіряє, чи був пошкоджений кадр в процесі передавання. За результатами цієї перевірки формується позитивне або негативне підтвердження про прийом кадру, яке передається тому комунікаційному вузлу, що його надіслав. В разі наявності помилок в кадрі, які не можуть бути виправлені з використанням коду корекції, його необхідно повторно передати. Така процедура передавання суттєво знижує загальну пропускну спроможність мережі.

У мережах Frame Relay вважається, що використовується середовище передавання з низьким рівнем помилок, тобто надійні канали зв'язку. Тому при передаванні кадрів підтвердження про їх прийом не передаються, а пошкоджений кадр просто ліквідується. Виявлення помилок і, в разі необхідності, повторне передавання повідомлення забезпечується приклад-

ним програмним забезпеченням користувача, тобто протоколами вищих рівнів. В результаті в мережах FR для передавання використовується до 90% смуги пропускання, в той час як, наприклад, в мережах X.25 – до 40%.

На сьогодні максимальна швидкість передавання в мережах FR становить 34,368 Мбіт/с та 44,736 Мбіт/с відповідно для каналів E3 і T3.

Фізично мережа Frame Relay складається з сукупності комунікаційних вузлів, зв'язаних фізичними каналами, та пристроїв доступу (рис. 8.10). В мережі FR розрізняють два типи пристроїв:

- термінальні пристрої **DTE (**Data Terminal Equipment**)** – зовнішні модулі для доступу до мережі, наприклад, робочі станції, термінали, мультиплексори, маршрутизатори, мости тощо;
- мережні пристрої **DCE (**Data Circuit-terminating Equipment**)** – комунікаційні вузли у складі мережі, призначені для синхронізації та формування каналу між кінцевими взаємодійними пристроями; зазвичай, це комутатори кадрів.

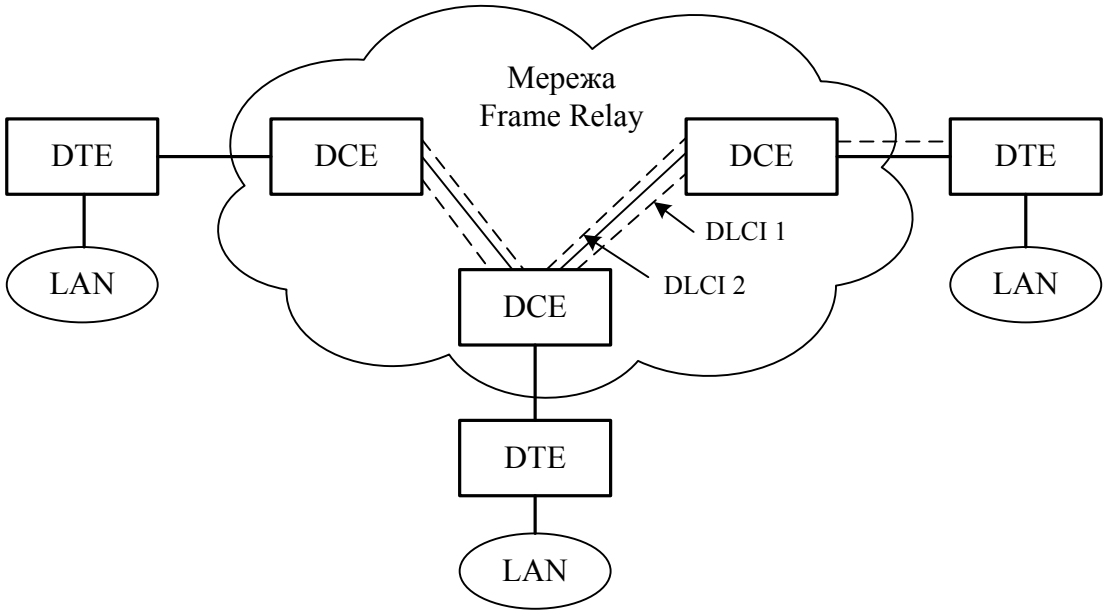


Рисунок 8.10 – Узагальнена структура мережі Frame Relay

Для передавання даних від відправника до отримувача в мережі Frame Relay створюються віртуальні канали **VC** (Virtual Circuit), які можуть бути двох типів:

- постійний віртуальний канал **PVC** (Permanent Virtual Circuit), який створюється між взаємодійними станціями і існує достатньо довго навіть при відсутності передавання даних;
- комутований віртуальний канал **SVC** (Switched Virtual Circuit), який створюється між взаємодійними станціями безпосередньо перед передаванням даних і анулюється одразу після закінчення сеансу зв'язку.

Створеному віртуальному каналу надається відповідний ідентифікатор з'єднання каналу даних **DLCI** (Data Link Connection Identifier). Комутатори виконують мультиплексування віртуальних каналів, які розпізнаються за унікальними номерами DLCI, у фізичний канал.

Стек протоколів FR (рис. 8.11) реалізує два нижні рівні: фізичний – для передавання даних між пристроями доступу, канальний – для сигналізації та керування, для передавання інформації про стан віртуальних каналів тощо. На канальному рівні, який реалізується протоколом LAP-F core і є спрощеною версією протоколу LAP-D, передається потік кадрів змінної довжини, але поле даних не може перевищувати 4096 байтів. Протокол LAP-F core функціонує в будь-яких каналах ISDN і каналах типу T1/E1. При використанні постійних віртуальних каналів PVC необхідно підтримувати тільки протокол LAP-F core. При передаванні даних комутованими віртуальними каналами SVC необхідно використовувати протокол LAP-F control, який виконує функції контролю доставки кадрів і керування потоком.

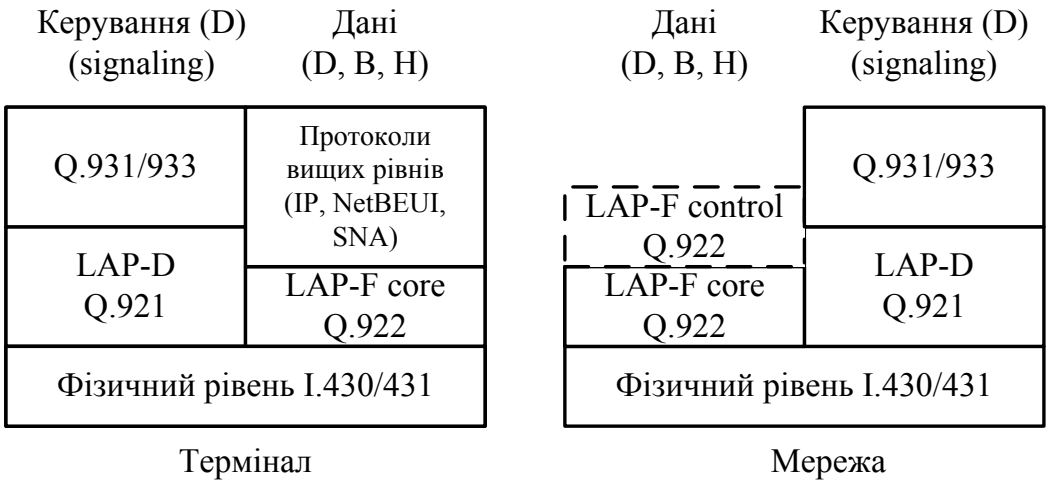


Рисунок 8.11 – Стек протоколів мережі Frame Relay

Канал типу D інтерфейсу користувача використовується для створення комутованих віртуальних каналів. У стеку протоколів цього каналу функціонує протокол LAP-D, над яким функціонує протокол Q.931 або Q.933, на основі якого встановлюється віртуальне з'єднання, використовуючи адреси кінцевих абонентів, а також ідентифікатори віртуального з'єднання DLCI.

Враховуючи сказане вище можна виділити **переваги** мереж FR:

- незначна затримка доставки повідомлень в мережі;
- можливість встановлення пріоритетів для різних типів трафіку;
- динамічне розподілення пропускної спроможності каналу зв'язку;
- можливість надання необхідної пропускної спроможності каналу за вимогою кінцевого прикладного процесу.

Недоліками мереж FR є:

- орієнтація на якісні канали зв'язку, вартість яких достатньо висока;
- відсутність гарантованої доставки кадрів і повідомлення про доставку в кінцевий термінал.

8.6 Мережі ATM

Технологія асинхронного передавання **ATM** (Asynchronous Trasnfer Mode) була розроблена ще у 1970-х роках у Франції та США незалежно у France Telecom та Bell Labs відповідно. Технологія ATM являє собою єдиний транспортний механізм для мереж з інтеграцією послуг та дозволяє передавати в мережі різні типи трафіку (комп'ютерні дані, аудіо- та відеопотоки) забезпечуючи при цьому для кожного типу трафіку достатню пропускну спроможність та якість обслуговування і гарантуючи вчасну доставку чутливої до затримок інформації. Тобто, технологія ATM забезпечує обслуговування всіх типів трафіку відповідно до їх вимог таким чином, щоб затримки і переривання в потоці даних були непомітні користувачу. Для цього в мережах ATM реалізовані різні способи передавання даних: як орієнтованих на з'єднання, так і без встановлення з'єднання. Технологія ATM – це технологія комутації комірок фіксованої довжини з використанням методу асинхронного мультиплексування з часовим ущільненням **ATDM** (Asynchronous Time Division Multiplexing).

При введені даних в мережу ATM пакет розбивається на комірки фіксованої довжини, кожна з яких маркується ідентифікатором (рис. 8.12). Комутація комірок, які можуть містити будь-яку інформацію, виконується апаратно.

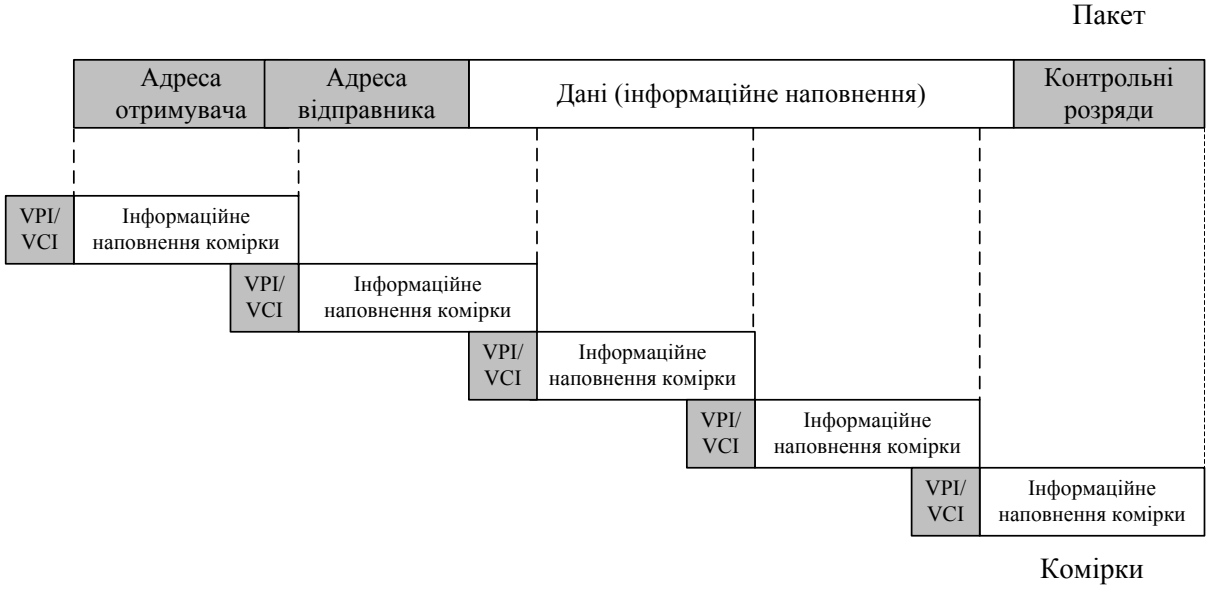


Рисунок 8.12 – Розподіл пакета на комірки

Модель ATM. Модель ATM подібна до моделі взаємодії відкритих систем OSI, але повної відповідності між рівнями немає. Протоколи ATM охоплюють тільки нижні рівні (рис. 8.13), а програмне забезпечення користувача виконує всі функції вищих рівнів, які пов'язані як з роботою мережі, так і з наскрізним контролем коректності передавання та керування потоком. Інтерфейс програмного забезпечення користувача зі службами ATM виконується на підрівні MAC канального рівня OSI, і тому більшість функцій, що пов'язані з мережним рівнем моделі OSI (маршрутизація, ретрансляція тощо), виконується на рівні ATM, що відповідає канальному рівню.

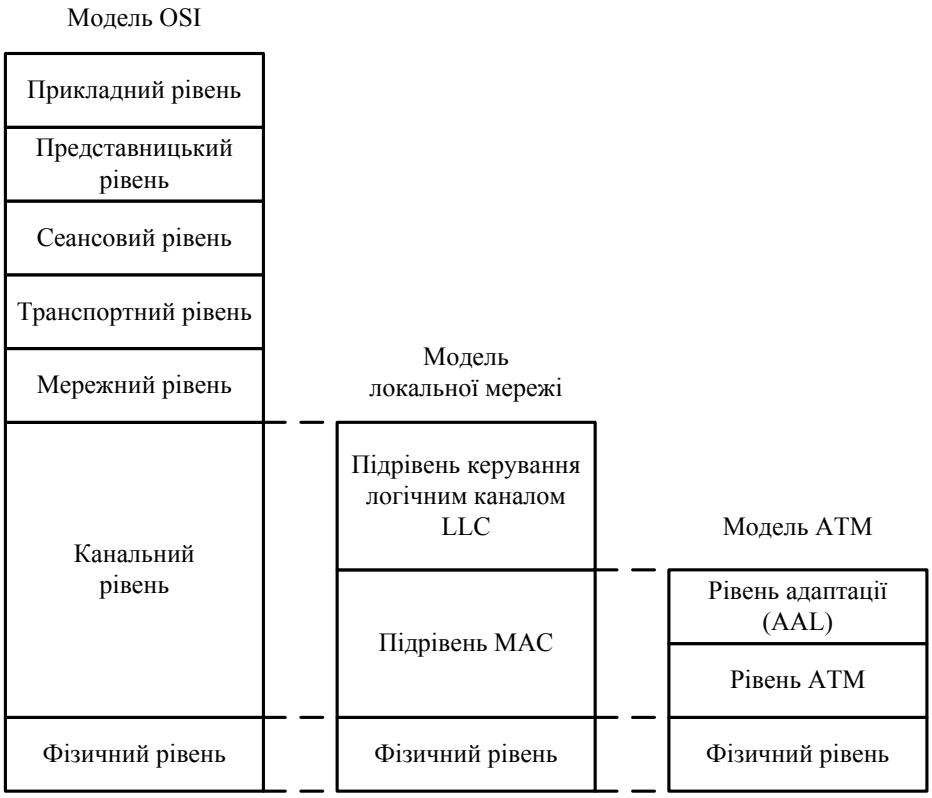


Рисунок 8.13 – Модель ATM та її відповідність моделі OSI

У моделі ATM виділяють чотири платформи: для передавання інформації користувача, для обробки контрольної інформації, для забезпечення основних мережних функцій, для керування цими платформами.

Всі платформи є сукупністю чотирьох рівнів (рис. 8.14), з яких три нижніх рівні, відповідно до визначення організацій ANSI, ATM Forum та ITU, відносять безпосередньо до моделі ATM:

- рівень користувача (**User Layer**), на якому реалізовані відповідні протоколи верхніх рівнів для прикладних задач;
- рівень адаптації ATM (**ATM Adaptation Layer – AAL**) для з'єднання кінцевих пристроїв з мережею ATM;
- рівень ATM (**ATM Layer**) – для комутації та мультиплексування потоків та їх керування;

- фізичний (**Physical Layer**) – для передавання фізичними каналами.

Рівень адаптації AAL забезпечує доступ застосувань користувача до комутаторів мережі ATM, оскільки багато прикладних сервісів не мають прямого доступу до сервісів ATM. Рівень адаптації являє собою набір протоколів AAL1–AAL5, які, залежно від типу трафіку і параметрів його передавання, формують з пакетів користувача блоки стандартного розміру. Кожний з протоколів AAL обробляє трафік відповідного класу:

- AAL1 – гарантована доставка, постійна швидкість, використовується для класу A;
- AAL2 – гарантована доставка, змінна швидкість, використовується для класу B;
- AAL3/4 – гарантована і негарантована доставки, змінна швидкість, керування потоком;
- AAL5 – негарантована доставка, керування потоком, підтримка IP зверху ATM.

<table>
<tr><td colspan="3">Протоколи верхніх рівнів</td></tr>
<tr><td rowspan="3">Рівень адаптації ATM
(AAL1 – 5)</td><td rowspan="2">Підрівень конвергенції (CS)</td><td>Спільна частина підрівня конвергенції</td></tr>
<tr><td>Специфічна частина для сервісу</td></tr>
<tr><td colspan="2">Підрівень сегментації і реасемблювання (SAR)</td></tr>
<tr><td colspan="3">Рівень ATM
(маршрутизація, мультиплексування, керування потоком, обробка пріоритетів)</td></tr>
<tr><td rowspan="2">Фізичний рівень</td><td colspan="2">Підрівень узгодження передавання</td></tr>
<tr><td colspan="2">Підрівень, який залежить від фізичного середовища</td></tr>
</table>

Рисунок 8.14 – Стек протоколів мережі ATM

Протоколи AAL при передаванні потоків користувача функціонують тільки на кінцевих станціях мережі і тому аналогічні транспортним протоколам більшості мережних технологій (рис. 8.15). Треба зауважити, що жодний з протоколів AAL не виконує процедури відновлення втрачених та пошкоджених даних, але, зазвичай, повідомляє кінцевий вузол про виникнення такої ситуації. Корекція пошкоджених даних виконується протоколами вищих рівнів, які не входять в стек протоколів технології ATM.

Цей рівень має два підрівні: конвергенції або перетворення **CS** (Convergence Sublayer) та сегментації і реасемблювання **SAR** (Segmentation And Reassembly). Підрівень **CS** залежить від класу трафіку передавання і забезпечує синхронізацію взаємодійних станцій, контролює коректність передавання потоку користувача і, по можливості, виправляє помилки.

Підрівень SAR не залежить від протоколів AAL, тобто від типу трафіку передавання, а формує комірки з пакетів і виконує обернені перетворення.

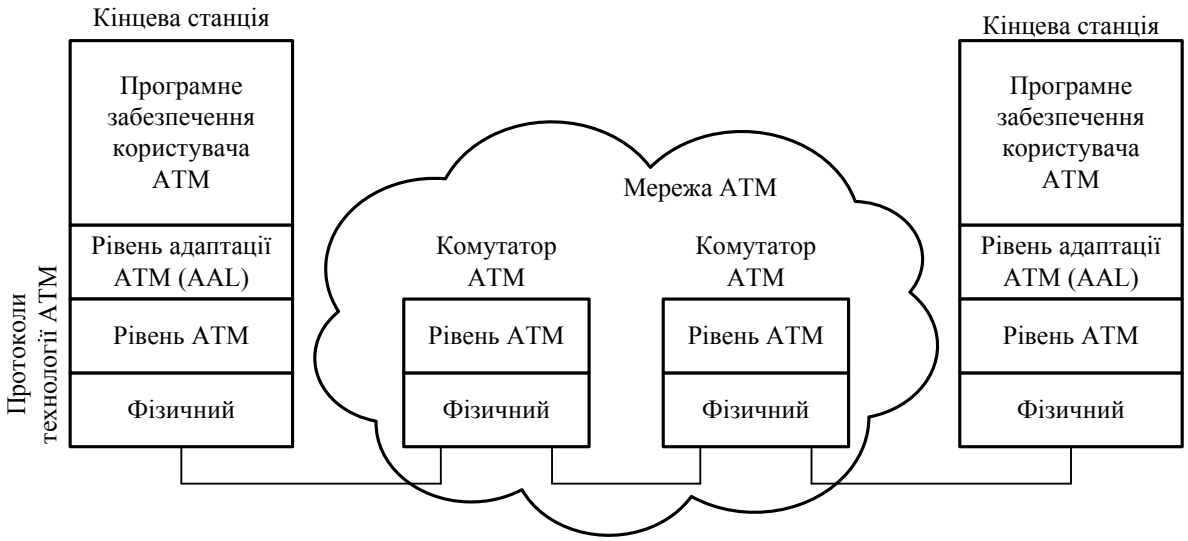


Рисунок 8.15 – Розподілення протоколів по вузлах і комутаторах АТМ

АТМ передбачає чотири **класи служб** (обслуговування), які розрізняються за трьома характеристиками: синхронізація пристроїв між кінцевими точками передавання, швидкість передавання потоку бітів та необхідність встановлення з'єднання (табл. 8.4). П'ятий клас визначається користувачами або виробниками обладнання АТМ.

Таблиця 8.4 – Характеристика класів служб АТМ

	Клас A	Клас B	Клас C	Клас D	X
Синхронізація	Потрібна		Не потрібна		
Швидкість	Постійна	Змінна			
Встановлення з'єднання	Встановлення з'єднання			Без з'єднання	

Характеристика класів, що наведена в таблиці 8.4, дозволяє визначити сферу їх застосування. Служба **класу A** використовується для передавання аудіо- та відеопотоків замість звичайного комунікаційного з'єднання в режимі комутації каналів. Служба **класу B** призначена для передавання ущільнених аудіо- та відеоінформації та використовується, зазвичай, в телеконференціях, при організації яких незначні затримки і змінна швидкість передавання є допустимими. **Клас C** забезпечує обмін даними аналогічно передаванню в режимі віртуальних каналів в звичайних комп'ютерних мережах (протоколи LLC2, Frame Relay, TCP, X.25). Служба **класу D**, зазвичай, використовується в застосуваннях, які не вимагають встановлення з'єднання між взаємодійними станціями, наприклад, протоколи IP, DNS, SNMP, Ethernet.

Рівень ATM відповідає за створення комірок стандартного формату, для чого приймає з рівня адаптації 48-байтові блоки і додає 5-байтовий заголовок. Протокол ATM виконує передавання комірок через комутатори на основі готових таблиць комутації портів і комутації за номером віртуального з'єднання, який складається з двох частин: номера віртуального маршруту і номера віртуального каналу. Крім цієї основної функції протокол ATM виконує також керування потоком комірок (при виконанні умов домовленості між кінцевими станціями).

Для забезпечення та підтримки необхідної якості обслуговування різних віртуальних з'єднань і ефективного використання ресурсів мережі в протоколі ATM реалізовано декілька служб, які надають послуги різних категорій обслуговування трафіку користувача. Ці служби призначені для підтримки трафіку різних класів в тандемі з протоколами AAL і є внутрішніми службами мережі ATM, які розподілені між всіма комутаторами мережі. Послуги цих служб разом з параметрами трафіку та необхідним рівнем якості обслуговування QoS (Quality of Service) задаються кінцевою станцією при встановленні з'єднання і розділені на категорії, які відповідають класам трафіку.

Визначено такі **категорії обслуговування**, які використовуються для забезпечення різних рівнів якості сервісу QoS для трафіку різного типу:

- постійна швидкість передавання **CBR** (Constant Bit Rate) – послуга для трафіку з постійною бітовою швидкістю;
- змінна швидкість передавання **VBR** (Variable Bit Rate) – послуга для трафіку з середньою бітовою швидкістю;
- невизначена швидкість передавання **UBR** (Unspecified Bit Rate) – послуга для трафіку, що не висуває вимог до швидкості передавання даних і синхронізації взаємодійних станцій;
- доступна швидкість передавання **ABR** (Available Bit Rate) – послуга для трафіку зі змінною бітовою швидкістю, який вимагає встановленої мінімальної бітової швидкості, але не вимагає синхронізації взаємодійних станцій.

Послуги категорії **CBR** призначено для підтримки трафіку синхронних застосувань: голосового, цифрових виділених каналів тощо. У випадку, коли прикладний сервіс встановлює з'єднання цієї категорії, жорстко задаються максимальна швидкість, яку може підтримувати з'єднання без втрати комірок, і параметри якості обслуговування QoS: максимальна затримка комірок, інтервали значення затримки та максимальна кількість втрачених комірок.

В категорії змінної швидкості передавання **VBR** виділяють два типи:

- **rt-VBR** (Real-time VBR) – послуги для трафіку зі змінною швидкістю в реальному часі, який вимагає забезпечення середньої швидкості передавання даних і синхронізації взаємодійних станцій;

- **nrt-VBR** (Non-real-time VBR) – послуги для трафіку зі змінною швидкістю, який вимагає забезпечення середньої швидкості передавання даних, але не вимагає синхронізації взаємодійних станцій.

Порівняно з категорією CBR передавання зі змінною швидкістю вимагає більш складної процедури запиту з'єднання між прикладним сервісом і мережею. І якщо категорія **rt-VBR** висуває жорсткі вимоги до затримки, але відносно низькі вимоги до втрати комірок, то для категорії **nrt-VBR** значення затримки не є основним, але втрата комірок має бути незначною.

На відміну від категорії CBR та служб VBR, категорія **UBR** не підтримує ні параметри трафіку, ні параметри якості обслуговування. Категорія **UBR** забезпечує доставку даних «по можливості» без будь-яких гарантій. Для передавання даних надаються найкращі послуги з доступних в мережі на цей час. Головним недоліком цієї категорії є відсутність можливості керувати потоком даних і неспроможність враховувати інші типи трафіку. При перевантаженні мережі з'єднання UBR продовжує передавати дані, і, оскільки в такому режимі не враховуються параметри ні трафіку, ні QoS, то саме такі комірки відкидаються в першу чергу.

Категорія доступної швидкості передавання **ABR** використовує керування потоком, завдяки чому при перевантаженні мережі забезпечуються гарантії збереження комірок, а не їх відкидання. Трафік з'єднання за категорією ABR отримує гарантовану якість послуг за параметрами пропускної спроможності та можливої частки втрачених комірок. При цьому затримки передавання комірок мережа намагається звести до мінімуму, однак гарантій цього не дає. Це призводить до того, що служба ABR не може бути використана для сервісів і застосувань реального часу.

Зауважимо, що при передаванні трафіку категорій CBR, VBR і UBR відсутні можливості керування перевантаженнями в мережі, а використовуються тільки процедури відкидання комірок. Це призводить до того, що станції, які використовують послуги CBR і VBR, намагаються не порушувати умови з'єднання і тому не використовують додаткову пропускну спроможність навіть пза її наявності в мережі.

Служба ж ABR дозволяє використовувати резерв пропускної спроможності мережі, оскільки повідомляє кінцевій станції про наявність на даний час пропускної спроможності, яка не використовується іншими станціями, за допомогою зворотного зв'язку. Цей механізм дозволяє службі ABR кінцевої станції знизити швидкість передавання даних (до мінімально встановленого значення) в мережу, якщо вона перевантажена. При цьому вузол, що використовує послуги ABR, має періодично відправляти в мережу крім комірок даних ще й спеціальні службові комірки керування ресурсами **RM** (Resource Management). Комірки RM, які вузол відправляє в тому ж напрямку, що і потік даних, називають прямими комірками **FRM** (Forward Recource Management), а комірки, які передають-

ся в зворотному напрямку (відносно потоку даних), – зворотними комірками **BRM** (Backward Recource Management). Передаванням між кінцевими станціями прямих і зворотних комірок узгоджується доступна на даний момент швидкість передавання комірок в мережу.

Таким чином, категорія ABR не тільки забезпечує підтримку вимог до обслуговування конкретного віртуального з'єднання, але й дозволяє більш раціонально розподіляти ресурси мережі між її абонентами, що, в свою чергу, веде до підвищення якості обслуговування всіх абонентів мережі.

Пріоритетне обслуговування трафіку базується на категоріях послуг кожного віртуального з'єднання. Спочатку більшість комутаторів ATM використовувало однорівневу схему обслуговування, згідно з якою кожному трафіку надавався свій пріоритет: CBR – перший пріоритет, VBR – другий, а UBR – третій. Такий підхід не забезпечує необхідну мінімальну швидкість передавання комірок трафіку ABR, оскільки для цього необхідно виділити деяку гарантовану смугу пропускання.

Для гарантованої підтримки служби ABR на сьогодні реалізована дворівнева схема обслуговування, згідно з якою кожному класу служб надається відповідна частина пропускної спроможності: трафіку CBR надається частина пропускної спроможності, яка необхідна для підтримки максимальної (пікової) швидкості, трафіку VBR – частина пропускної спроможності, що необхідна для підтримки середньої швидкості, а трафіку ABR – частина пропускної спроможності, яка необхідна для підтримки мінімальної швидкості передавання комірок. Такий підхід гарантує, що кожне з'єднання буде функціонувати без втрат комірок і не буде передавати комірки ABR за рахунок трафіку CBR або VBR. На другому рівні трафіки CBR та VBR можуть (якщо це необхідно) використовувати всю пропускну спроможність мережі, яка залишилась після того, як з'єднання ABR вже отримали гарантовану їм пропускну спроможність.

У березні 1999 р. ATM-форум розробив специфікацію 4.1 керування трафіком, в якій визначена ще одна службова категорія – гарантована швидкість передавання фреймів **GFR** (**Guaranteed Frame Rate**). Ця категорія використовується тільки в віртуальному канальному з'єднанні VCC для підтримки непотокових застосувань. Головною метою даної категорії є забезпечення прикладних застосувань мінімальною гарантованою швидкістю передавання даних. При використанні даної категорії можна динамічно збільшувати пропускну спроможність, для чого від кінцевої станції необхідно задати значення максимальної та мінімальної швидкостей передавання та максимального розміру фрейму (кадру). Застосування цієї категорії вимагає об'єднання комірок даних користувача у фрейми, розміри яких встановлюються на рівні ATM. Така організація передавання більш ефективна за рахунок меншої збитковості, але при виникненні перевантаження в мережі ATM буде відкинута не одна комірка, а цілий фрейм.

Фізичний рівень виконує кодування комірок і їх передавання через середовище передавання. Він складається з двох підрівнів: конвергенції передавання **TC** (Transmission Convergence) та адаптації до фізичного середовища **PMD** (Physical Media Dependent).

Підрівень TC реалізує різні протоколи передавання через фізичні канали, наприклад, SDH/SONET, T1/E1, T3/E3 тощо, і виконує такі функції:

- генерацію контрольних бітів комірки при її передаванні в мережу та перевірка комірки на наявність помилок при її прийомі з мережі;
- виділення меж комірок в неперервному потоці бітів при його прийомі з мережі;
- формування та передавання в канал структурованих кадрів, які складаються з вихідних комірок, що необхідно для коректної роботи деяких інтерфейсних протоколів, та реалізація зворотних процедур.

Підрівень PMD приймає потік даних від підрівня TC і передає його як електричний або оптичний сигнал у фізичну лінію. При прийомі отримує фізичний сигнал і формує потік бітів, який і передає на підрівень TC. При цьому виконуються функції:

- кодування бітового потоку відповідно до способу, який використовується в даному фізичному інтерфейсі, наприклад, коди 4B/5B, NRZ, RZ та інші;
- синхронізація та часове узгодження передавання бітів;
- передавання та прийом електричного або оптичного сигналу в фізичний канал та прийом з нього.

Структура комірки. Розмір комірки, що передається в мережі ATM, становить 53 байти, з них 5 байтів виділяється для заголовка, а 48 – для інформаційного наповнення комірки. При встановленні розміру комірки враховувались особливості використання різних, вже існуючих каналів зв’язку.

Мережі європейського континенту зазвичай не дуже великі, що не вимагає використання технології компенсації відлуння, під якою розуміють процес усунення (або значного послаблення) відлуння з голосового сигналу для покращення якості передавання голосового сигналу в каналах зв’язку. І оскільки головним параметром була затримка, то запропоновано використовувати комірки, які б складались з 4 байтів заголовка і 32 байтів поля даних (інформаційного наповнення).

У мережах американського континенту часто використовуються канали, які мають значну довжину, тому технологія компенсації відлуння в них застосовується досить давно. І, щоб не втрачати ефективність передавання при достатньо великому заголовку в каналі, було запропоновано використовувати 64 байти для поля даних і 5 байтів для заголовка комірки.

Враховуючи такі два підходи міжнародна організація ITU-T встановила існуючу структуру комірки ATM фіксованої довжини. Розрізняють такі

формати комірок для інтерфейсів двох типів: користувач – мережа **UNI** (User-Network Interface) та мережа – мережа **NNI** (Network-Network Interface).

На рис. 8.16 показана структура комірки ATM для інтерфейсів різних типів. Узагальнене керування потоком **GFC** (Generic Flow Control) використовується тільки для комірок інтерфейсу користувач – мережа (UNI). Стандарт ATM визначає поки один некерований режим, для якого всі біти цього поля дорівнюють нулю.

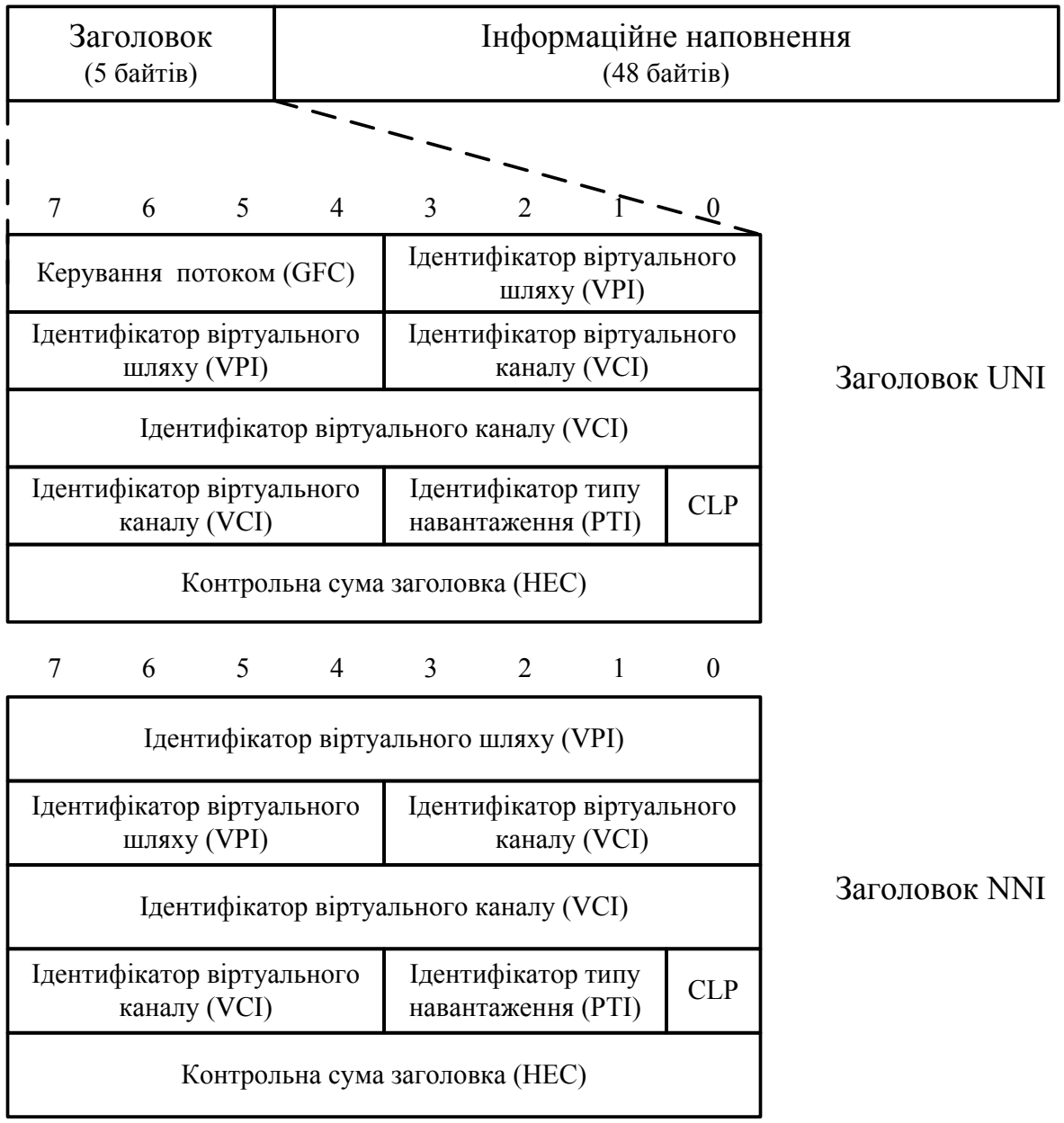


Рисунок 8.16 – Структура комірки ATM

Ідентифікатор віртуального шляху (маршруту) **VPI** (Virtual Path Identifier) використовується для об’єднання віртуальних каналів при маршрутизації.

Ідентифікатор віртуального каналу **VCI** (Virtual Channel Identifier) визначає конкретний віртуальний канал в віртуальному шляху.

Ідентифікатор типу навантаження **PTI** (Payload Type Identifier) ідентифікує тип даних, які містяться в полі інформаційного наповнення комірки.

Пріоритет втрати комірки **CLP** (Cell Loss Priority) дозволяє обладнанню АТМ визначити, які комірки необхідно обробляти в першу чергу при виникненні перевантаження. Одиничне значення означає високий пріоритет втрати комірок, тобто, при перевантаженні мережі такі комірки відкидаються в першу чергу. Нульове значення означає низький пріоритет втрат.

Контрольна сума заголовка **HEC** (Header Error Check) контролює коректність передавання тільки заголовка. Реалізовано два режими: виявлення помилок за допомогою циклічного коду та корекція однократних помилок. В першому випадку виявлення помилки в заголовку приведе до відкидання комірки, а в другому – корегуються однократні помилки (проте якщо виникла багатократна помилка, то вона також трактується як однократна і неправильно виправляється). Поле HEC вираховується в кожному з'єднанні, оскільки при проходженні через мережу постійно змінюється значення VPI/VCI.

Віртуальні канали, які створюються в мережі АТМ, можуть бути трьох типів:

- постійні віртуальні канали **PVC** (Permanent Virtual Circuit), які організовуються при конфігуруванні мережі та являють собою постійні з'єднання між двома кінцевими станціями;
- комутовані віртуальні канали **SVC** (Switched Virtual Circuit), які встановлюються динамічно кожний раз, коли будь-якій станції необхідно передати дані іншій станції, при цьому станція-відправник передає запит на встановлення з'єднання, а мережа АТМ розповсюджує адресні таблиці і передає цій станції значення VPI та VCI, які записуються в заголовок комірки; після передавання даних між станціями встановлене з'єднання анулюється;
- інтелектуальні постійні віртуальні з'єднання, які налагоджуються автоматично, **SPVC (**Smart або Soft Permanent Virtual Circuits) насправді є каналами PVC, які ініціалізуються за вимогами в комутаторах АТМ, при цьому задаються тільки кінцеві станції, а для кожного передавання, залежно від завантаженості каналів, мережа визначає, через які комутатори будуть передаватися комірки.

Різниця між віртуальними каналами і віртуальними шляхами показана на рис. 8.17.

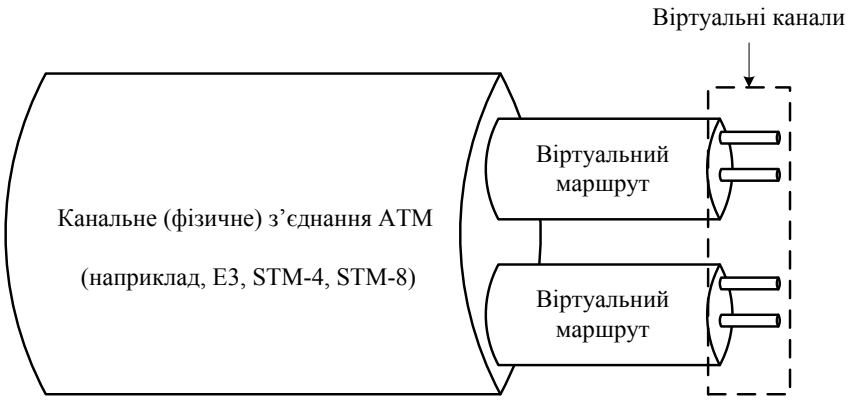


Рисунок 8.17 – Віртуальні канали та шляхи

Віртуальний шлях (маршрут) – це сукупність віртуальних каналів. Кожний віртуальний шлях, загальна кількість яких, залежно від інтерфейсу, дорівнює відповідно 256 або 4096, містить 256 віртуальних каналів, кожний з яких має свій унікальний ідентифікатор VCI. Комбінація цих двох ідентифікаторів є ідентифікатором даного з'єднання. Віртуальне канальне з'єднання VCC (Virtual Circuit Connection) містить сукупність віртуальних маршрутів VP.

Узагальнена характеристика класів і типів трафіку, їх параметрів та особливостей використання наведена в табл. 8.5.

Таблиця 8.5 – Основні характеристики класів трафіку АТМ

Клас QoS	1	2	3	4	5
Клас обслуговування	A	B	C	D	x
Тип трафіку	CBR	VBR	VBR	ABR	UBR
Тип рівня	AAL1	AAL2	AAL3/4	AAL3/4	
Синхронізація	Потрібна		Не потрібна		
Швидкість передавання	Постійна	Змінна			
Режим встановлення з'єднання	З встановленням з'єднання			Без встановлення з'єднання	
Приклади використання	Трафік T1/E1, голосовий трафік, TV-трафік	Стиснуті відео та голосовий трафік	Frame Relay, X.25, LLC2, TCP	IP, Ethernet, DNS, SNMP	Передавання даних

Існують два основних **фізичних інтерфейси**, особливості яких відображаються у форматах комірок, що передаються між пристроями мережі за допомогою функцій АТМ:

- інтерфейс «користувач – мережа» **UNI** (User-Network Interface);
- інтерфейс «мережа – мережа» **NNI** (Network-Network Interface);

- інтерфейс B-ISDN між мережами операторів зв'язку **B-ICI** (B-ISDN Inter-Carrier Interface);
- міжмережний інтерфейс ATM **AINI** (ATM Internetwork Interface).

Інтерфейс «користувач – мережа» UNI – це інтерфейс між кінцевою точкою мережі ATM і комутатором ATM, який є проміжною системою. Розрізняють суспільні та приватні інтерфейси UNI, які, однак, використовують один формат комірок. **Суспільний інтерфейс UNI** визначає формати комірок, які передаються між комутатором ATM (мережним вузлом) постачальника комунікаційних послуг і пристроєм ATM в приватній мережі. Крім того суспільний інтерфейс використовується між суспільним комутатором ATM і пристроєм кінцевої точки ATM, а також між суспільним комутатором ATM мережі загального користування і приватним комутатором ATM в організації користувача. **Приватний інтерфейс UNI** визначає формати комірок, які передаються між приватним комутатором ATM і пристроєм кінцевої точки ATM (кінцевою станцією).

Інтерфейс «мережа – мережа» NNI – це інтерфейс між будь-якими комутаторами ATM. Як і у випадку інтерфейсу UNI, розрізняють суспільні та приватні інтерфейси NNI, які також використовують однаковий формат. **Суспільний інтерфейс NNI** визначає формати комірок, які передаються між двома комутаторами ATM загального користування. Цей інтерфейс може бути реалізовано як між двома комутаторами ATM в одній мережі колективного користування, так і між комутаторами ATM в різних суспільних мережах. Що стосується **приватного інтерфейсу NNI**, то він може існувати тільки між двома приватними комутаторами ATM, які можуть належати як мережі однієї організації, так і мережам різних компаній.

Інтерфейс B-ICI визначає взаємозв'язок декількох провайдерів служб ATM і дуже схожий на інтерфейс NNI. Різниця полягає в тому, що інтерфейс NNI реалізує функції фізичного рівня та рівня ATM, тоді як специфікація B-ICI містить і всі вищі рівні, а саме: рівень AAL та спеціальні рівні міжоператорних служб. При цьому підтримка інших міжоператорних служб є обов'язковою для інтерфейсів B-ICI, оскільки дозволяє технології ATM підключатися до інших служб, наприклад, Frame Relay, служб передавання голосу тощо.

Міжмережні інтерфейси AINI забезпечують обмін між мережами, в одній з яких використовується приватний протокол NNI, а в іншій – протокол **B-ISUP (Broadband ISDN User Part)** широкосмугової мережі користувача ISDN, або між двома мережами, в яких використовується приватний протокол NNI. Особливості організації всіх цих інтерфейсів показано на фрагменті мережі (рис. 8.18).

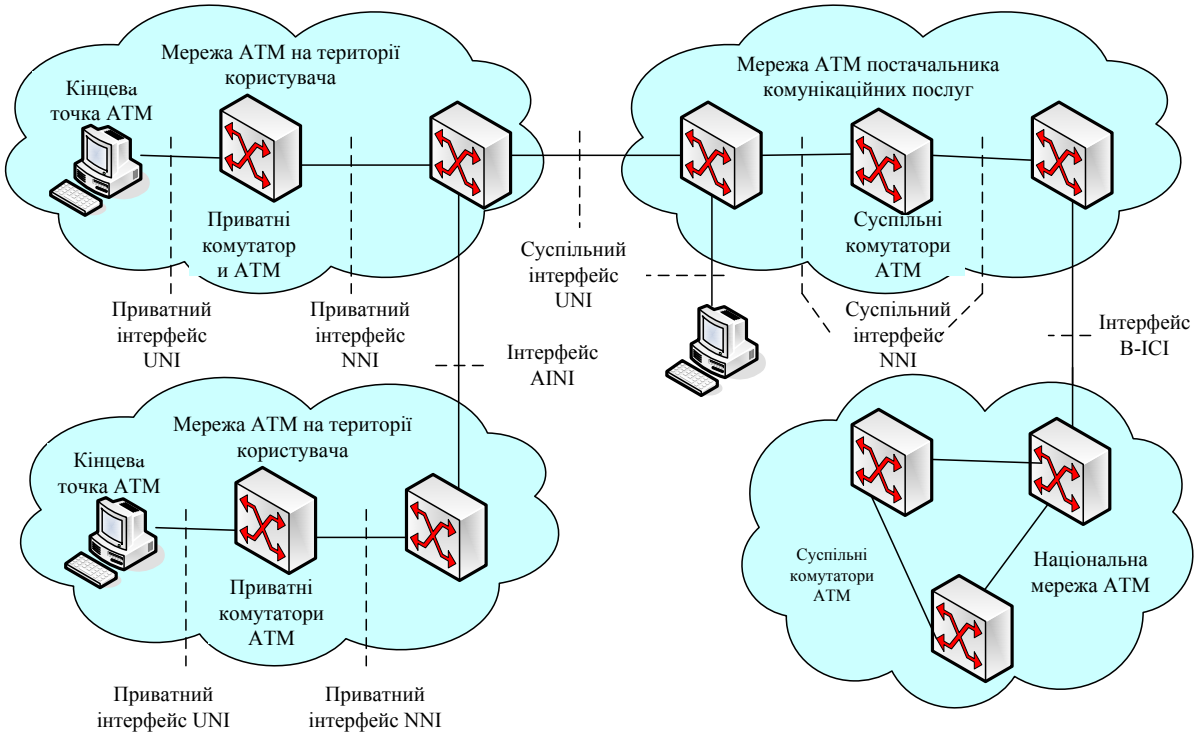


Рисунок 8.18 – Мережні інтерфейси АТМ

Мережні компоненти АТМ. Як показано на рис. 8.18, мережа АТМ складається з компонентів трьох основних типів: комутатори АТМ; кінцеві точки АТМ; маршрути пересилання.

Комутатори АТМ виконують функції, що пов’язані з маршрутизацією інформації від станції-відправника до станції-отримувача. Іноді такий комутатор називають проміжною системою або проміжним вузлом **IS** (**Intermediate System**). Всі комутатори можуть бути як суспільними, так і приватними. Суспільні комутатори АТМ являють собою частину суспільної мережі постачальника комунікаційних послуг і за стандартами АТМ називаються мережним вузлом **NN** (**Network Node**). Приватні комутатори належать і обслуговуються компанією користувача, й, згідно зі стандартами АТМ, називаються вузлом на території користувача **CPN** (**Customer Premises Node**). Існують також комутатори, які можуть виконувати в мережі функції як суспільних, так і приватних комутаторів.

Кінцеві точки АТМ – це пристрої, які виступають у ролі станції-відправника або станції-отримувача даних користувача, часто називаються кінцевими системами **ES** (**End System**). Як кінцева точка, зазвичай, використовуються комп’ютери з відповідним програмним і апаратним забезпеченням, крім того це може бути спеціальний мережний пристрій, до якого підключаються комп’ютерні системи.

Кінцеві точки можуть підключатися до суспільного комутатора АТМ мережі комунікаційних послуг або до приватного комутатора АТМ мережі користувача.

Маршрути пересилання ТР (**Transmission Path**) забезпечують взаємодію кінцевих точок і комутаторів АТМ за допомогою різних типів фізичних комунікаційних каналів.

Особливості функціонування мережі АТМ. Технологія АТМ, яку іноді називають технологією ретрансляції комірок (Cell Relay), є технологією передавання даних з комутацією пакетів, в якій використовуються блоки фіксованого розміру – комірки довжиною 53 байти. Ці мережі орієнтовані на використання надійного фізичного носія для передавання даних. Комутатори АТМ не виконують процедури виявлення помилок і не відправляють підтвердження при передаванні комірок через мережу. Все це веде до мінімальних втрат при обробці комірки в комутаторі, а передавання даних невеликими порціями комірок дозволяє достатньо ефективно передавати дані з постійною швидкістю. Крім того, технологія АТМ надає користувачам переваги комутації каналів, що гарантує надання пропускної спроможності для передавання даних і відповідний рівень обслуговування. Швидкість передавання в мережі АТМ суттєво залежить від типу каналу зв'язку, і на сьогодні середня швидкість передавання даних становить 622 Мбіт/с, а максимальна – 5 Гбіт/с (при використанні оптоволоконного кабелю).

Основні переваги технології АТМ полягають у нижченаведеному.

1. Технологія АТМ ефективно використовує смугу пропускання мережі, яка надається тільки за вимогою користувача, а принцип статичного мультиплексування дає можливість колективно використовувати смугу пропускання каналу.
2. АТМ підтримує багатофункціональні послуги для передавання інформації різних типів в будь-якій формі.
3. Висока ефективність передавання, яка характеризується зменшенням затримки передавання, необхідною якістю обслуговування та можливістю динамічного перерозподілу смуги пропускання між кінцевими системами.
4. Мережі АТМ характеризуються невеликою затримкою передавання, що пояснюється тим, що контроль помилок та їх корекція виконують протоколи більш високих рівнів. Постійний розмір комірки приводить до того, що комутація і маршрутизація реалізуються апаратно, що, в свою чергу, визначає конкретну, передбачену затримку передавання даних в мережі.
5. Висока надійність мережі, яка пояснюється тим, що мережі АТМ швидко реагують на похибки передавання, змінюючи при цьому виділення смуги пропускання без зупинки і при перевантаженні каналу передавання, виконуючи перенаправлення трафіку на інше з'єднання.
6. Технологія АТМ характеризується високою гнучкістю. Віртуальні з'єднання дають можливість користувачу отримати гарантований мінімум смуги пропускання для кожного з'єднання і забезпечити його необхідною

якістю обслуговування з різними вимогами до затримки передавання та втрати комірок для кожного класу.

7. Забезпечення єдиного мережного транспорту. Технологія ATM підтримує єдиний спосіб передавання даних, який дозволяє з'єднувати мережі будь-яких розмірів і технологій. При цьому немає необхідності виконувати додаткову трансляцію і застосовувати шлюзові системи між мережами різних типів.

8.7 Технологія xDSL

Технологія **xDSL** (**Digital Subscribers Line**) була розроблена компанією Bell Operating Company у 1987 році і являє собою високошвидкісну технологію, яка дозволяє максимально використовувати наявні абонентські лінії. xDSL – це сукупність технологій, які дозволяють суттєво розширити пропускну спроможність абонентської лінії за рахунок використання ефективних лінійних кодів і адаптивних методів корекції спотворень сигналу. Швидкість передавання даних в xDSL становить від 64 Кбіт/с до 52 Мбіт/с, що дозволяє користувачу вибрати, залежно від потреб, необхідну модифікацію.

Технологія xDSL забезпечує передавання голосу, високошвидкісне передавання даних і відео, а також одночасне передавання даних і голосу по одній мідній парі. Існуючі типи технології xDSL розрізняються, в основному, типом модуляції та значенням швидкості і дальності передавання.

Існує два типи технології xDSL: симетричні та асиметричні, параметри яких наведено в табл. 8.6.

Таблиця 8.6 – Загальна характеристика технології xDSL

	Технологія xDSL	Максимальна швидкість	Максимальна відстань, км	Лінійне кодування	К-ть телеф. пар	Основне використання
Симетричні	HDSL	2,048 Мбіт/с	6,5	2B1Q або CAP	2	Об'єднання мереж, послуги E1
	SDSL	2,048 Мбіт/с	3,0	2B1Q або CAP	1	Об'єднання мереж, послуги E1
	SHDSL (G.shdsl)	4,6 Мбіт/с	7,5	TC-PAM	2	Об'єднання мереж, передавання даних
	MDSL	2,3 Мбіт/с		2B1Q	1	Передавання даних
	MSDSL	2,064 Мбіт/с	8,8	від CAP8 до CAP128	1	Доступ до Internet, об'єднання LAN, доступ до мереж SDH, об'єднання голосу та даних
	VDSL2	100 Мбіт/с	0,15		1	Передавання даних
	IDSL	144 Кбіт/с	10,8	2B1Q	1	Передавання даних

Несиметричні	ADSL	8 Мбіт/с – прийом 1,5 Мбіт/с – передавання	5,5	CAP та DTM (OFDM)	1	Доступ до Internet, віддалений доступ до LAN, голос, відео
	ADSL.Lite	1,5 Мбіт/с – прийом 512 Кбіт/с – передавання	5,5	CAP та DTM (OFDM)	1	Доступ до Internet, віддалений доступ до LAN, голос, відео
	VDSL	52 Мбіт/с – прийом 16 Мбіт/с – передавання	1,2	QAM або DTM	1	Об'єднання мереж, телебачення високої роздільної здатності
	RADSL	7 Мбіт/с – прийом 1 Мбіт/с – передавання	5,5	CAP та DTM (OFDM)	1	Доступ до Internet, голос, відео
	UADSL	1,5 Мбіт/с – прийом 384 Кбіт/с – передавання	3,5	CAP та DTM (OFDM)	1	Доступ до Internet, голос, відео

Симетричні технології xDSL забезпечують однакову швидкість передавання в напрямку мережа – користувач (низхідний потік) і в напрямку користувач – мережа (висхідний потік). Асиметричні технології xDSL забезпечують швидкість передавання низхідного трафіку значно вищу, ніж висхідного. Симетричні модифікації зазвичай використовуються в корпоративних мережах, асиметричні – для забезпечення доступу абонентів до мультимедійної мережі. На сьогодні найбільше використання знаходять асиметричні модифікації, з яких, в свою чергу, – технологія **ADSL.**

Пояснюється це тим, що для більшості користувачів низхідний потік (вхідний трафік) значно більш важливий, ніж висхідний (вихідний трафік). Тому для низхідного трафіку надається канал з більшою смугою пропускання, що? за теоремою Г. Найквіста? веде до більшої пропускної спроможності.

Організація обміну з використанням технології xDSL реалізується за допомогою DSL-модема, який підключається до модулів користувача, та мультиплексора доступу **DSLAM** (DSL Access Multiplexer), який знаходиться у провайдера. Модеми DSL відрізняються від звичайних телефонних модемів значно більшим діапазоном частот (від декількох КГц до одиниць МГц), що і привело до визначення xDSL як технології **широкосмугового доступу**. Звичайна телефонна лінія, як було розглянуто в розділі 2, для передавання голосу використовує смугу частот в інтервалі 0,3–3,4 КГц. А, наприклад, в ADSL діапазон частот знаходиться в інтервалі від 26 КГц до 1,1 МГц, причому ця смуга частот розділяється на два підканали: для висхідного потоку виділяють частоти в інтервалі 26 КГц — 138 КГц, а для низхідного – інтервал від 138 КГц до 1,1 МГц.

Основні розбіжності технологій xDSL за параметрами пропускної спроможності та максимальної відстані полягають у способах модуляції, які використовуються для кодування даних. Залежно від цього розрізняють:

- системи DSL з послідовним передаванням сигналів, тобто «системи модуляції однієї несучої» **SCM** (Single Carrier Modulation), які використовують способи кодування 2B1Q, CAP тощо;
- системи з паралельним передаванням сигналів на декількох несучих частотах або «системи з багатьма несучими» **DTM** (Discrete Multitone), які використовують в асиметричних DSL.

Дамо скорочену характеристику наведених в таб. 8.6 технологій.

ADSL (Asymmetric Digital Subscriber Line) – асиметрична DSL, яка на сьогодні використовується найчастіше, – використовує два типи лінійного кодування: CAP і DTM і дозволяє використовувати одну пару проводів для передавання даних і підключення телефонного апарата до міської АТС. Обладнання ADSL можна конфігурувати для досягнення максимальної швидкості з мінімальним рівнем помилок.

ADSL.Lite (відома також під назвою **G.Lite**) подібна до ADSL і забезпечує менші швидкості, але характеризується простотою інсталяції та низькою вартістю створення і підтримки в робочому стані.

VDSL (Very High Bit-Rate Digital Subscriber Line) – перспективна високошвидкісна технологія, яка забезпечує роботу виділених каналів невеликої довжини (300–1200 м) в синхронному режимі, причому при збільшенні відстані зменшується швидкість. В термінології ITU-T цей стандарт називається **G.vdsl**.

RADSL (Rate Adaptive ADSL) – варіант технології ADSL з автоматичним налагоджуванням швидкості передавання залежно від стану лінії. На сьогодні практично не використовується, оскільки всі стандартні варіанти ADSL забезпечують налагоджування швидкості передавання.

UADSL (Universal ADSL або DSL Lite), порівняно з технологією ADSL, забезпечує менші швидкості передавання, але характеризується незначною вартістю і простотою налагоджування.

HDSL (High Bit-Rate DSL) – високошвидкісне симетричне дуплексне передавання даних, яка використовує чотирипроводові лінії, з можливістю налагоджування швидкості обміну. Використовується для доступу до трактів SDH або PDH, а також підтримує роботу мережних сервісів в реальному часі.

SDSL (Single Digital Subscriber Line) аналогічна HDSL, але, завдяки використанню двопроводової абонентської лінії, більш економічна.

SHSDL (Symmetric High Bit-Rate DSL) – стандарт високошвидкісного симетричного передавання даних (в термінології ITU-T називається **G.shdsl)**, який є єдиним стандартизованим варіантом xDSL.

MDSL (Multi-Rate DSL) забезпечує невелику дальність передавання і, завдяки використанню модуляції 2B1Q, забезпечує якісне з'єднання в каналах з високим рівнем шумів.

MSDSL (Multi-Rate Single-Pair DSL) забезпечує автоматичне регулювання швидкості передавання залежно від стану лінії та якості сигналу. Використовується для високошвидкісного доступу до мереж Internet та SDH, об'єднання локальних мереж тощо.

IDSL (ISDN DSL) – технологія, яка функціонує на основі технології ISDN і характеризується невеликими швидкостями та вартістю. На відміну від мереж ISDN забезпечує зв'язок з абонентами через з'єднання типу «точка-точка».

Треба зауважити, що суттєвою перевагою технології xDSL є забезпечення високошвидкісного доступу до сервісів та послуг мережі з використанням існуючої кабельної інфраструктури місцевих телефонних мереж.

8.8 Технологія MPLS

Технологія багатопротокольної комутації за мітками **MPLS** (**MultiProtocol Label Switching**) з'явилась наприкінці 90-х років XX ст., об'єднує можливості комутації пакетів і керування трафіком, які реалізуються ресурсами канального рівня та маршрутизації, що виконується на мережному рівні. Багатопротокольність MPLS полягає в тому, що вона може функціонувати з будь-яким протоколом вищих рівнів. При цьому сама технологія MPLS є незалежною від протоколів канального та мережного рівнів мереж IP, ATM та інших, тобто MPLS – це своєрідний транспортний ресурс, який передає інформацію багатьох протоколів вищих рівнів. **Головна особливість технології MPLS** полягає у відділенні процесу комутації пакетів від аналізу ідентифікаторів (наприклад, IP-адрес), які передаються в його заголовку, що дозволяє виконувати комутацію пакетів значно скоріше.

Таким чином, технологія MPLS є технологією комутації потоків на основі міток. Для забезпечення цієї можливості для кожного потоку пакетів відповідного протоколу, наприклад, IP, створюється своя мітка, яка записується між заголовками канального та мережного рівнів (рис. 8.19), і подальша обробка виконується на основі не складної маршрутизації, а швидкої комутації, тобто наступний модуль в маршруті передавання визначається без процедури пошуку його адреси.

Заголовок протоколу канального рівня (Ethernet, PPP, FDDI тощо)	Заголовок MPLS	Заголовок протоколу мережного рівня (IPv4, IPv6, AppleTalk тощо)	Дані верхніх рівнів

Рисунок 8.19 – Розміщення заголовка MPLS в кадрі

Мітка призначається маршрутизатором в кожній точці входу потоку в магістраль MPLS, і її значення передається сусіднім маршрутизаторам. Будь-який потік пакетів асоціюється з конкретним класом еквівалентності пересилання **FEC** (Forwarding Equivalence Class). Клас еквівалентності визначає групу пакетів, які переадресовуються однаковим чином і передаються одним маршрутом з такою ж процедурою обробки. Клас FEC може відповідати не тільки адресі мережі призначення, а й будь-якому класу трафіку, тобто до одного класу відносять пакети всіх потоків, маршрути передавання яких через мережу MPLS і процедури обробки збігаються, тобто при виборі наступного маршрутизатора пакети цих потоків неможливо розрізнити.

Кожний з класів ідентифікується конкретною міткою, значення якої є унікальним для конкретного фрагмента шляху між двома сусідніми вузлами мережі MPLS – маршрутизаторами, які комутують за мітками **LSR** (Label Switching Router).

Структура мітки, яка має фіксовану довжину, що дорівнює 4 байтам, наведена на рис. 8.20, де:

Мітка – значення мітки, за якою виконується комутація; це 20-бітове число в діапазоні $0–2^{20-1}$, за винятком зарезервованих значень від 0 до 15 (відповідно до RFC-3032);

CoS (Class of Service) – клас обслуговування блока даних;

S – біт, який визначає, чи є дана мітка останньою в стеку міток, для останньої мітки в стеку біт S дорівнює 1, для всіх інших – 0;

TTL (Time to Live) – час життя (повний аналог поля TTL в IP-дейтаграмі).

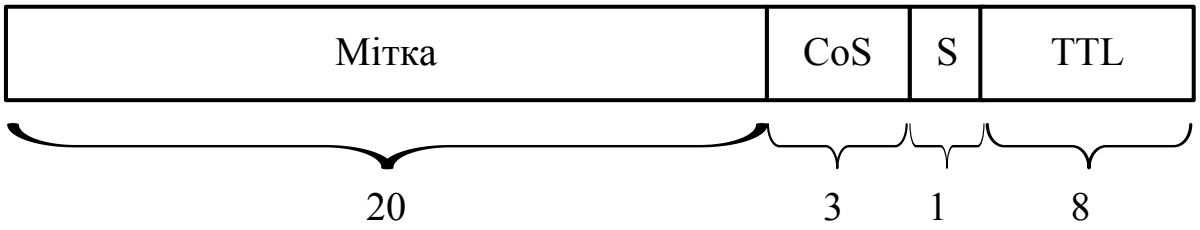


Рисунок 8.20 – Формат заголовка MPLS

При передаванні потоків даних через магістралі MPLS виділяють дві різні ділянки: мережу користувача і мережу MPLS або MPLS-домен, що складається з сукупності маршрутизаторів, які підтримують комутацію за мітками і знаходяться під єдиним адміністративним керуванням (рис. 8.21).

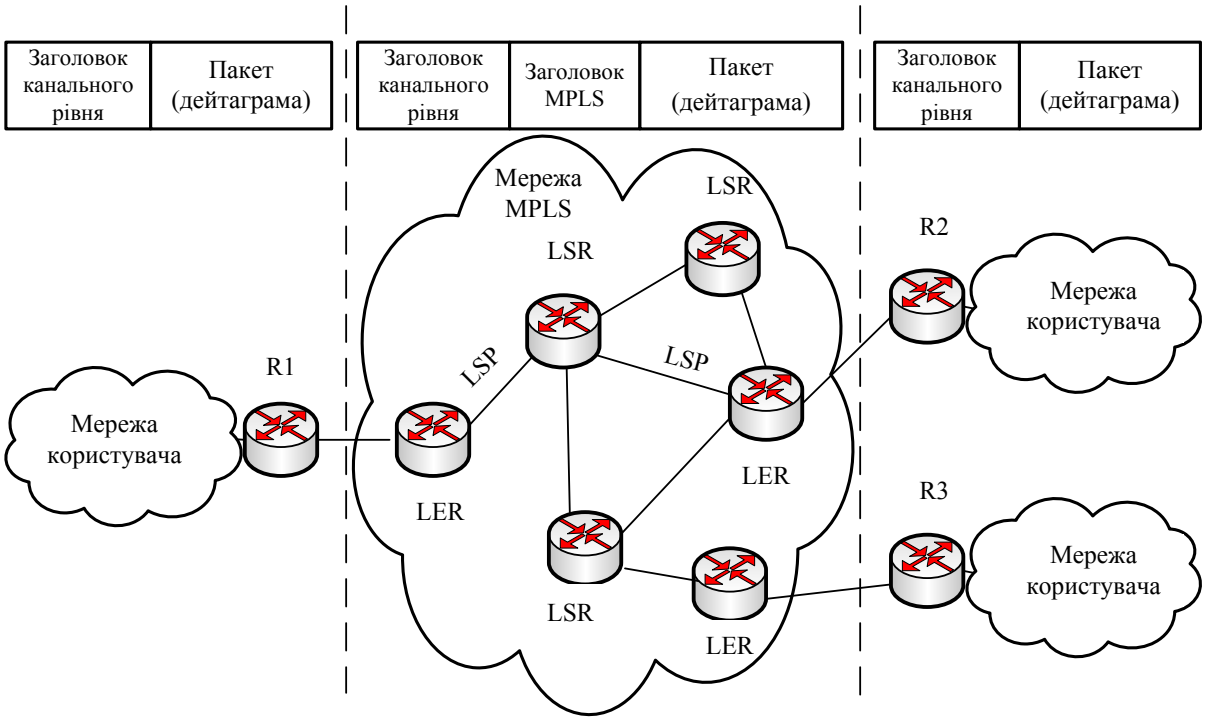


Рисунок 8.21 – Фрагмент мережі MPLS

Мережа MPLS складається з комунікаційних вузлів двох типів:

- межевих комутаторів **LER** (Label Edge Router), які об'єднують домен MPLS з комунікаційними вузлами, що знаходяться поза цим доменом (в даній структурі R1, R2 і R3);
- внутрішніх (транзитних) маршрутизаторів комутації за мітками **LSR** (Label Switch Router).

Процедура передавання пакета між кінцевими користувачами через магістраль MPLS реалізується таким чином. Будь-який пакет передається в мережу MPLS з маршрутизатора суміжної мережі R1, яка може бути як мережею користувача, так і іншою MPLS-мережею. Перший межевий маршрутизатор LER на вході в мережу аналізує адресу отримувача і необхідну якість обслуговування **QoS** (Quality of Service), що знаходяться в заголовку пакета. На основі цієї інформації визначається значення мітки, яка присвоєна відповідному класу FEC, і вихідний інтерфейс для передавання наступному, транзитному маршрутизатору LSR. Ця мітка, значення якої є унікальним для каналу між сусідніми маршрутизаторами, і необхідна якість (клас) обслуговування записуються між заголовками канального і мережного рівнів. Пакет проходить декілька маршрутизаторів LSR мережі MPLS поки не передається на кінцевий межевий маршрутизатор LER, який видаляє з нього мітку і передає адресату, який знаходиться не в домені MPLS. Послідовність маршрутизаторів LER, LSR, LSR, … LSR, LER, через які передається пакет, що належить одному класу FEC, створює віртуальний, комутований за мітками, однонаправлений тракт **LSP** (Label Switch Path).

Кожний транзитний маршрутизатор LSR містить таблицю **LIB** (Label Information Base), в ній зберігається інформація, на основі якої виконується комутація та заміна вхідної мітки на вихідну. Структура таблиці наведена на рис. 8.22, в якій кожній парі «вхідний інтерфейс, вхідна мітка» ставиться у відповідність пара інших значень, а саме: «вихідний інтерфейс, вихідна мітка». При цьому значення префікса використовується тільки при побудові та заповненні таблиці, а в самому процесі комутації не бере участі. Для кожного отриманого пакета маршрутизатор LSR за значенням «вхідний інтерфейс, вхідна мітка» визначає вихідний інтерфейс, замінюючи при цьому вхідну мітку в пакеті на нову, вихідну, яка зберігається у відповідному полі таблиці, і пересилає пакет наступному маршрутизатору LSR.

Вхідний інтерфейс	Вхідна мітка	Префікс	Вихідний інтерфейс	Вихідна мітка
1	25	197.35.78.0	3	21
1	12	205.14.91.0	3	16
2	17	147.56.101.0	4	28

Рисунок 8.22 – Структура таблиці LIB

Зауважимо, що для вхідного та вихідного межевих маршрутизаторів LER в таблиці відсутні пари значень «вхідний інтерфейс, вхідна мітка» та «вихідний інтерфейс, вихідна мітка» відповідно. Треба зазначити, що пакет, який передається в мережі MPLS, зазвичай має не одну, а декілька міток, які утворюють стек (рис. 8.23), що дає можливість створювати ієрархію міток і дозволяє суттєво розширити функціональні можливості технології за рахунок організації тунелів. Особливості комутації задає тільки верхня мітка стека (на рис. 8.23 це MPLS 3), а інші (нижні) мітки передають прозоро, без обробки, до видалення мітки зі стека. Мітка заголовка MPLS 1 є останньою, тому S = 1, і хоча й записана в стек першою, оброблятися буде останньою.

Заголовок протоколу канального рівня	Заголовок MPLS 3	Заголовок MPLS 2	Заголовок MPLS 1	Заголовок протоколу мережного рівня	Дані верхніх рівнів

Рисунок 8.23 – Стек міток

Зі стеком міток можна виконувати такі операції: записувати мітку в стек, видаляти мітку зі стека і здійснювати заміну мітки. Операція запису мітки в стек додає нову мітку поверх вже існуючого стека міток, а операція видалення мітки зі стека видаляє верхню мітку стека. Ці операції дозволяють виконувати як об’єднання (злиття) потоків, так і їх розгалуження.

Функціональні можливості стека MPLS дозволяють об'єднувати декілька віртуальних шляхів LSP, які мають спільну частину маршруту, в один. Для цього до стека міток кожного з об'єднуваних LSP додається спільна мітка, в результаті чого утворюється єдиний агрегований тракт MPLS. В точці закінчення цього тракту він розгалужується на індивідуальні тракти LSP.

Основні переваги технології MPLS полягають у нижчевикладеному.

1. Незалежність від особливостей таких різних технологій канального рівня, як Ethernet, Frame Relay, SDH та інших, що дозволяє використовувати магістралі MPLS для передавання різного типу трафіку: кадрів Ethernet, IP-дейтаграм, комірок ATM тощо.
2. Вибір маршруту передавання виконується не на основі аналізу IP-адреси, а мітки, що дозволяє скоротити час пошуку маршруту в таблиці, а за рахунок цього комутація пакетів реалізується скоріше і забезпечується більш висока швидкість передавання трафіку в мережі.
3. Проста процедура створення віртуальних приватних мереж **VPN** (Virtual Private Network), яка дозволяє реалізувати швидкісні та надійні з'єднання з високим рівнем захисту інформації користувачів за рахунок створення для кожної VPN незалежних тунелів.
4. Забезпечення необхідного рівня якості обслуговування за рахунок того, що кожному інформаційному потоку призначається необхідний клас обслуговування CoS, який обробляється окремо. Крім того процедури та механізми оптимізації, керування трафіком і мережею відокремлено від передавання інформаційного потоку, що приводить до більшої ефективності використання таких алгоритмів в великих мережах передавання даних.
5. Розподілення функціональності між ядром мережі (доменом MPLS) та мережею користувача.
6. Підтримка каналів з високою пропускною спроможністю, які функціонують на швидкостях в 2,488 Гбіт/с (канали STM-16 або OC-48) і більше, що не можуть забезпечити інші мережні технології.

На сьогодні виділяють 3 основні сфери застосування технології MPLS:

- керування потоком;
- підтримка класів та якості обслуговування;
- створення віртуальних приватних мереж.

Керування інформаційним потоком в мережі MPLS дозволяє передавати потоки даних через найменш завантажені канали зв'язку і маршрутизатори, а не найкоротшим маршрутом, який визначається за допомогою відповідного протоколу маршрутизації. В результаті мережа передавання даних функціонує більш стабільно та ефективно, а адміністратори мережі можуть задавати маршрути, що відповідають специфічним умовам обробки, і отримувати статистичну інформацію про кожен віртуальний маршрут LSP.

Підтримка необхідної якості обслуговування забезпечується завдяки виділенню ресурсів в кожному маршрутизаторі на маршруті передавання і гарантії того, що даний трафік матиме ресурси для отримання необхідної якості обслуговування даного потоку. Оскільки кожний потік обробляється самостійно, технологія MPLS надає конкретному потоку користувача диференційовані послуги **DiffServ** (Differential Services) при його передаванні та обробці.

При створенні **віртуальних приватних мереж** виконується тунелювання для кожної VPN. І оскільки маршрутизація в MPLS реалізується на основі міток, а не адрес, достатньо легко кожній створеній віртуальній мережі надати необхідні їй послуги. Для VPN особливо важливо забезпечити захист даних корпоративних користувачів при їх передаванні через мережу.

На сьогодні існують модифікації технології MPLS, які орієнтовані на використання в конкретних застосуваннях. До таких модифікацій відносять: **T-MPLS** (Transport Multiprotocol Label Switching), **MPLS-TP** (MPLS Transport Profile), **G-MPLS** (Generalized MPLS), **Multiprotocol Lambda Switching**.

Технологія **транспортної багатопротокольної комутації T-MPLS** розроблялась організацією ITU-T в 2006–2008 роках для використання в транспортних пакетних мережах операторів зв'язку і передбачає спрощення деяких функцій базової технології. Основні відмінності від технології MPLS полягають у використанні двонаправлених віртуальних маршрутів LSP, а також можливості об'єднання віртуальних маршрутів LSP, коли для всього трафіку, який передається з даного маршрутизатора в одному напрямку, використовується одна мітка.

Стандарт **MPLS-TP** розробляється організаціями ITU-T та IETF, починаючи з 2008 року. На сьогодні знаходиться на стадії розробки і передбачається відмова від зроблених раніше спрощень в технології T-MPLS.

Технологія **G-MPLS** також знаходиться на стадії розробки і передбачає розширення сфери дії технології MPLS для будь-яких транспортних технологій та передбачає її перенесення від маршрутизаторів на оптичний рівень, на якому вибір маршруту подальшого передавання виконується на основі часових інтервалів, довжини хвилі та фізичних портів, які в термінології даної технології називають неявними мітками.

Основна ідея технології **Multiprotocol Lambda Switching** полягає в тому, що при передаванні світлового потоку по оптоволоконному кабелю функції міток виконують різні довжини оптичних хвиль. Це дозволяє додавати сервіси віртуальних приватних мереж та підвищувати якість обслуговування фактично на фізичному рівні, на якому обробка виконується з оптичною швидкістю.

8.9 Питання для самоперевірки

1. Сформулюйте основні характерні ознаки плезіохронної цифрової ієрархії.
2. Охарактеризуйте основні принципи, які покладено в основу ієрархії цифрових каналів.
3. Охарактеризуйте основні принципи, які використовуються при утворенні каналів плезіохронної цифрової ієрархії.
4. Поясніть термін «крадіжка біта» та особливості використання цієї процедури.
5. Поясність особливості структури кадрів Т1 та Е1. Яким чином в них передається службова інформація?
6. Скільки голосових каналів передається каналом DS-5 європейської, американської та японської систем стандартизації?
7. Охарактеризуйте та проаналізуйте основні переваги та недоліки плезіохронної цифрової ієрархії.
8. Охарактеризуйте основні відмінності і переваги синхронної та плезіохронної ієрархії цифрових каналів.
9. Проаналізуйте структуру кадрів STS-1 та STM-1, їх основні структурні компоненти та структуру кадрів в розгорнутому вигляді.
10. Поясніть, яким чином формуються синхронні транспортні модулі вищих рівнів.
11. Визначте, скільки каналів типу Е3 може передавати кадр STM-3.
12. Визначте, яку кількість каналів типу Т1 може передавати кадр STS-12.
13. Визначте, яку кількість каналів типу Е1 може передавати кадр STM-16, якщо в ньому вже мільтиплексовано 5 кадрів Т1.
14. Поясніть відмінності номінальної та корисної пропускних спроможностей кадрів синхронної цифрової ієрархії каналів. Визначте корисну пропускну спроможність кадру STM-12.
15. Охарактеризуйте типи модулів, які використовуються для побудови мереж SDH/SONET. Поясніть їх функціональні та процедурні характеристики.
16. Сформулюйте характерні особливості мереж з інтегрованим доступом ISDN.
17. Охарактеризуйте особливості функціональних груп пристроїв мережі ISDN. Поясніть особливості віртуальних і фізичних модулів.
18. Типи інтерфейсів доступу мережі ISDN, їх структура та характеристика.
19. Наведіть типи каналів технології ISDN, їх характеристику та особливості функціонування.
20. Що таке опорні точки? Типи опорних точок і особливості їх використання.
21. Проаналізуйте основні переваги мереж технології ISDN.

22. Охарактеризуйте основні функціональні та процедурні особливості мереж Frame Relay.
23. Охарактеризуйте стек протоколів мережі Frame Relay.
24. Проаналізуйте переваги і недоліки мереж Frame Relay.
25. Охарактеризуйте переваги технології ATM.
26. Проаналізуйте особливості стека протоколів технології ATM.
27. Проаналізуйте параметри категорій обслуговування технології ATM.
28. Який з протоколів AAL найбільш ефективно використовує пропускну спроможність каналу?
29. Охарактеризуйте типи інтерфейсів технології ATM. Що спільного, в чому різниця цих інтерфейсів?
30. Охарактеризуйте призначення та функціональні особливості мережних компонентів ATM.
31. Охарактеризуйте призначення та особливості технології xDSL.
32. Поясніть особливості симетричних і несиметричних технологій xDSL.
33. Сформулюйте особливості послідовного та паралельного передавання сигналів в системах xDSL?
34. Які типи лінійного кодування використовуються в технології xDSL?
35. Поясніть основні принципи організації та передавання потоків даних в мережах MPLS.
36. Сформулюйте основні функціональні та процедурні особливості маршрутизаторів, які використовуються в технології MPLS.
37. Охарактеризуйте структуру мітки MPLS та принципи її призначення.
38. Поясніть, що таке стек міток і принципи його використання та обробки.
39. Охарактеризуйте структуру таблиці LIB, її призначення та особливості використання.
40. Охарактеризуйте основні характеристики модифікацій технології MPLS.

9 БЕЗПРОВОДОВІ КОМП'ЮТЕРНІ МЕРЕЖІ

Останнім часом безпроводові комп'ютерні мережі та технології віддаленого доступу зазнають бурхливого розвитку. Це пов'язано з широким використанням різних мобільних пристроїв, які функціонують за принципом «anytime, anywhere», тобто отримувати послуги незалежно від місця знаходження та часового інтервалу.

На сьогодні безпроводові технології дозволяють створювати не тільки локальні мережі, але й регіональні та глобальні, забезпечуючи пропускну спроможність до 20 Гбіт/с. Крім того безпроводові мережі прості в розгортанні та забезпечують доступ в важкодоступних місцях.

9.1 Покоління безпроводового зв'язку

Безпроводові комп'ютерні мережі неможливо розглядати без аналізу типів безпроводового зв'язку та його характеристик. Технології мобільного (безпроводового) зв'язку зазвичай розділяють на декілька етапів або поколінь. Основним визначальним фактором, який покладено в ці класифікацію, є швидкість передавання даних. Розрізняють такі покоління мобільного зв'язку: 1G (Generation) – аналогові системи, 2G – цифрові системи, 3G – універсальні системи мобільного зв'язку, 4G – мобільний зв'язок з підвищеними вимогами.

До покоління 1G відносять:

- AMPS (Advanced Mobile Phone Service),
- NMT (Nordic Mobile Telephone),
- TACS (Total Access Communication System),
- C-450 RTMS (Radio Telephone Mobile System) тощо, які на сьогодні є застарілими.

Покоління 2G представляють стандарти:

- GSM (Global System for Mobile communication), який працює в діапазоні 900, 1800, 1900 МГц і є одним з найбільш розповсюджених світових стандартів;
- GPRS (General Packet Radio Service), який є розвитком технології GSM і забезпечує передавання максимум до 171,2 Кбіт/с (в середньому 40–50 Кбіт/с);
- EDGE (Enhanced Data Rates for Global (GSM) Evolution) також є розвитком технології GSM і забезпечує передавання даних максимум 473,6 Кбіт/с;
- CDMA (Code Division Multiple Access) – стандарт множинного доступу з кодовим розділенням, основні технічні характеристики якого визначені в низці стандартів: IS-95 (Interim Standard), IS-96, IS-97, IS-98 та IS-99; має модифікації CDMAOne та CDMA2000 1X, що забезпечує такі швидкості передавання даних:

- для абонентів з високою мобільністю (до 120 км/г) – не менше 144 Кбіт/с;
- для абонентів з низькою мобільністю (до 3 км/г) – 384 Кбіт/с;
- для нерухомих об'єктів на коротких відстанях – 2,048 Мбіт/с.

До 3G покоління відносять універсальні системи мобільного зв'язку, які забезпечують швидкість передавання від 2 Мбіт/с до 100 Мбіт/с, і разом з комутацією каналів забезпечують пакетне передавання даних. В дане покоління входить 5 стандартів, серед яких найбільш популярними на сьогодні є стандарти:

- UMTS (Universal Mobile Telecommunication System), який відомий ще під назвою WCDMA (Wideband Code Division Multiple Access) з подальшими покращеними модифікаціями HSDPA/HSUPA (High Speed Down-link Packet Access/High-Speed Uplink Packet Access) та HSPA+ (High Speed Packet Access), які забезпечують теоретично максимальну можливу швидкість передавання даних в низхідному каналі, тобто від базової станції до абонентів мережі 14,4 Мбіт/с, а в висхідному каналі (від абонента до базової станції) до 5,8 Мбіт/с; цей стандарт є подальшим розвитком європейських стандартів GSM/GPRS/EDGE;
- CDMA EV-DO (CDMA Evolution-Data Only), який є розвитком стандартів CDMAOne та CDMA2000 1X; швидкість передавання даних стандарту CDMA EVDO залежить від релізів (поколінь) стандарту і складає (в низхідному каналі/висхідному каналі відповідно):
  — CDMA2000 1x EV-DO Release 0 (rel.0) – 2,4/0,153 Мбіт/с (де 1x характеризує фазу розвитку стандарту мобільного зв'язку CDMA2000 1X, що належить до 2G);
  — CDMA2000 1x EV-DO Revision A (rev.A) – 3,1/1,8 Мбіт/с;
  — CDMA2000 1x EV-DO Revision B (rev.B) – 73,5/27 Мбіт/с (на практиці поки 7/5 Мбіт/с);
  — CDMA EV-DO Revision C (Rev.C) – 280/75 Мбіт/с;
  — CDMA EV-DO Revision D (Rev.D) – 500/120 Мбіт/с (останні два релізи є перспективними і потенційно можуть бути віднесені до наступного покоління 4G);
- **LTE (Long Term Evolution)** перші версії стандарту, які практично забезпечують швидкість передавання даних 11,8/4,8 Мбіт/с (теоретично 60/20 Мбіт/с);
- **WIMAX (Worldwide Interoperability for Microwave Access)**, крім останніх версій.

Покоління 4G представляють мережі, які забезпечують швидкість передавання даних більше 100 Мбіт/с для мобільних станцій і 1Гбіт/с для стаціонарних, суттєвою особливістю таких мереж є те, що передавання даних реалізується за протоколом IPv6. До таких мереж на сьогодні відносять мережі **WIMAX** стандарту IEEE 802.16m, які забезпечують визначені швидкості, та мережі **LTE-Advanced**, для яких теоретично визначені шви-

дкості 326,4/172,8 Мбіт/с відповідно для низхідного та висхідного каналів. Потенційно мережі **Wi-Fi** (Wireless Fidelity) стандартів IEEE 802.11n, 802.11ac та 802.11ad також можуть бути віднесені до мереж цього покоління. Базовими принципами мобільного зв'язку 4G є використання технологій мультиплексування з ортогональним частотним розділенням сигналу OFDM (Orthogonal Frequency Division Multiplexing) та одночасного передавання даних за допомогою N антен і їх одночасного прийому M антенами (технологія MIMO – Multiple Input/ Multiple Output).

Покоління 5G – це новий етап у розвитку технологій безпроводових мереж, який має розширити можливості доступу в мережу Інтернет через мережі радіодоступу. При розробці стандарту п'ятого покоління враховуються вдосконалені можливості вже існуючих стандартів. Задачами, які потрібно вирішити за допомогою технології 5G або **NR (New Radio)**, є постійне зростання мережного трафіку, збільшення кількості мобільних пристроїв, які підключаються до мережі, розширення частотного діапазону. Прийнято виділяти три основні послуги, для яких необхідне створення мереж нового покоління мобільного зв'язку 5G:

- надширокосмуговий мобільний зв'язок **eMBB (enhanced Mobile BroadBand)**;
- високонадійне з'єднання з дуже низькою затримкою передавання даних **URLLC (Ultra-Reliable and Low-Latency Communication)**;
- можливість підключення дуже великої кількості функціонально різних пристроїв **mMTC (massive Machine-Type Communication)**.

Мережі 5G (або мережі IMT-2020 в термінах Міжнародного союзу телекомунікацій (електрозв`язку) ITU) являють собою сукупність нових і існуючих радіоінтерфейсів і створюють єдину безпроводову інфраструктуру, що надає широкий спектр послуг. На сьогоднішній день, згідно з вимогами стандарту IMT-2020 технологія мобільного зв'язку має такі характеристики:

- теоретичний максимум пропускної спроможності складає 20 Гбіт/с у низхідному каналі (від базової станції до мобільного абонента) та 10 Гбіт/с – у висхідному каналі (в протилежному напрямку);
- реальна стабільна швидкість передавання в умовах міст 100 і 50 Мбіт/с відповідно з затримкою не більше 4 мс;
- збільшення спектральної ефективності в мережах 5G в 2–5 разів: в низхідному каналі – 30 біт/с/Гц, у висхідному – 15 біт/с/Гц;
- збільшення швидкості переміщення абонентів до 500 км/ч;
- збільшення загальної кількості підключених пристроїв до 1 млн/км^2;
- скорочення часової затримки на радіоінтерфейсі до 0,5 мс (для сервісів URLLC і до 4 мс (для сервісів eMBB);
- підвищення енергоефективності пристроїв на 2 порядки.

Покоління 6G. Якщо зміна поколінь 3G–4G–5G була обумовлена, в основному, необхідністю збільшення швидкості та зменшення затримок передавання даних, то поява мереж 6G приведе до змін самого підходу до створення мережної інфраструктури. Розробники вважають, що швидкість пере-

давання даних в мережах 6G складатиме від 100 Гбіт/с до 1 Тбіт/с, а для керування ними будуть використовуватись системи штучного інтелекту. Однією з основних задач мереж 6G буде вирішення проблеми лавиноподібного трафіку, який передається в світових мережах мобільного зв'язку. До задач, які будуть вирішувати мережі цього покоління, зазвичай відносять:

- віртуалізація процесів, інфраструктури мереж тощо;
- поява нових місць розміщення базових станцій безпроводових мереж (наприклад, на дронах, аеростатах, автомобілях тощо);
- забезпечення більш жорстких норм на безпеку передавання даних;
- надання принципово нових видів послуг: наприклад, передавання в електронному вигляді запахів тощо;
- необхідність забезпечення роботи супутникового зв'язку для масових споживачів та інші;
- необхідність обслуговування таких глобальних систем суспільства, як: колективної та індивідуальної безпеки (SafeNet); медичні (HealthNet); безпілотного транспорту (TransferNet); дронів (AeroNet); фінансові (FinNet); штучного інтелекту (NeuroNet) та інших.

Узагальнена порівняльна характеристика поколінь мобільного зв'язку наведена в таблиці 9.1.

Таблиця 9.1 – Покоління мобільного зв'язку

Покоління	1G	2G	3G	4G	5G
Принцип роботи	Аналогова телефонія.	Цифрова телефонія та передавання повідомлень/даних.	Широкосмугове передавання даних.	Широкосмугове передавання даних. Повністю IP-мережа.	Надширокосмугове передавання даних. Новий радіоінтерфейс. Віртуалізація мережних елементів.
Тип множинного доступу	FDMA	FDMA/ TDMA	CDMA	OFDMA	Конкурувальні технології: OFDMA, неортогональна схема доступу (NOMA), FBMC, UFMC, GFDM та інші.
Стандарти	AMPS, TACS, NMT	GSM, D-AMPS, JDC, 2.5G: GPRS, EDGE	WCDMA, CDMA200, UMTS, 3.5G: HSPA, HSPA+	3.9G: WIMAX, LTE, 4G: LTE-Advanced	5G
Ширина частотного каналу, МГц	0.03 (AMPS), 0.025 (NMT, TACS)	0.25	5	1.4/3/5/10/15/20 (LTE). До 100 (LTE-A) та більше в LTE-A Pro, де ширина окремих частотних каналів не перевищує 20 МГц.	Від 100 до декількох сотень МГц (в смузі до 20 ГГц); 500–1000 МГц (в діапазоні 20– 40 ГГц); більше 1000 МГц (більше 40 ГГц).
Максимальна швидкість, Мбіт/с	Не більше 0,002	Більше 0,4 (EDGE)	336 (HSPA)	1000 і більше (LTE-A)	20000

9.2 Класифікація безпроводових комп'ютерних мереж

Безпроводові мережі прийнято класифікувати за декількома ознаками: за характером з'єднання, типом середовища передавання, характером використання, типом модуляції тощо, але найбільш поширеною є класифікація за територіальною ознакою (рис. 9.1).

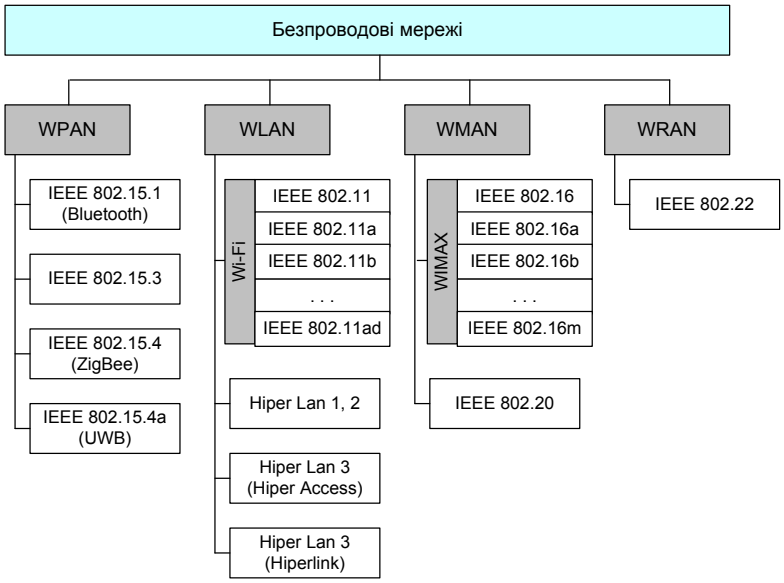


Рисунок 9.1 – Класифікація безпроводових комп'ютерних мереж

Серед безпроводових мереж зазвичай розрізняють такі типи:

- глобальні WWAN (Wireless Wide Area Network);
- регіональні WRAN (Wireless Regional Area Network);
- міські WMAN (Wireless Metropolitan Area Network);
- локальні WLAN (Wireless Local Area Network);
- персональні WPAN (Wireless Personal Area Network).

Безпроводові глобальні (всесвітні) мережі WWAN не мають територіальних обмежень і широко використовують супутникові канали зв'язку.

Безпроводова регіональна мережа WRAN – територіально-розподілена мережа, яка може покривати значні території.

Мережі мегаполісів WMAN – високошвидкісні мережі, які охоплюють територію діаметром (зазвичай) 50–70 км.

Безпроводова локальна мережа WLAN – мережа для обслуговування невеликих територій, зазвичай до 500 м.

Безпроводова персональна мережа WPAN (або домашня мережа) використовується для організації особистого простору (зазвичай радіусом 10–15 м). Суттєвою відмінністю таких мереж є використання пристроїв з малим енергоспоживанням, які можуть передавати дані від ноутбуків, мобільних телефонів, відеокамер, різних побутових пристроїв тощо.

9.3 Основні принципи передавання в безпроводових каналах зв'язку

В безпроводових мережах використовується технологія розширення спектра **SS** (**Spread Spectrum**), яка передбачає, що вузькосмуговий інформаційний сигнал подається таким чином, що його спектр стає значно більш широким, ніж початковий сигнал. Одночасно з цим виконується перерозподілення інформаційного сигналу по широкій смузі радіодіапазону, що дозволяє в результаті ускладнити перехоплення сигналу.

Техніка розширення спектра безпроводових мереж використовує два різних способи передавання сигналу:

- метод стрибкоподібної зміни частоти FHSS (Frequency Hopping Spread Spectrum);
- метод прямого послідовного розширення спектра DSSS (Direct Sequence Spread Spectrum).

Кожний з названих методів використовується в різних стандартах безпроводових мереж, але більш сучасною є технологія DSSS.

При використанні методу **стрибкоподібної зміни частоти FHSS** передавання відбувається при постійній зміні несучої в широкому діапазоні частот. В результаті потужність сигналу розподіляється по всьому діапазону частот, і прослуховування будь-якої певної частоти тільки визначає невеликий шум. Послідовність зміни несучих є псевдовипадковою, яка відома тільки модулю-передавачу та модулю-приймачу (рис. 9.2).

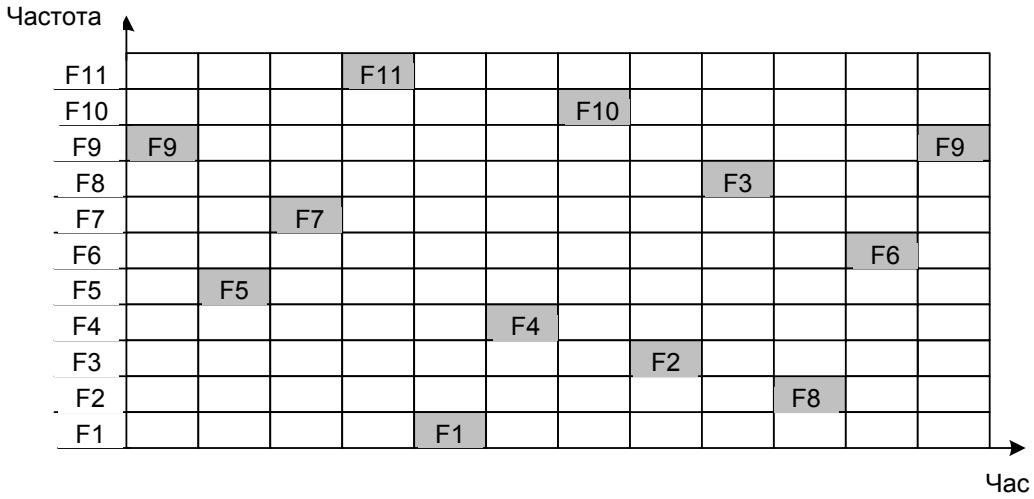


Рисунок 9.2 – Розширення спектра стрибкоподібної зміни частоти

Протягом фіксованого інтервалу передавання відбувається на незмінній несучій частоті, при цьому для передавання дискретної інформації використовуються стандартні методи модуляції, наприклад, фазової або частотної. Для синхронізації приймача та передавача для визначення початку кожного періоду передавання протягом деякого часу передаються синхробіти, які призводять до зменшення корисної пропускної спроможності. Для

прикладу, наведеного на рис. 9.2 послідовність зміни частот така: F9→F5→F7→F11→F1→F4→F10→F2→F3→F8→F6→F9.

Несуча частота змінюється відповідно до номерів частотних підканалів, які формуються алгоритмом псевдовипадкових чисел. Передавач і приймач синхронно переключаються з однієї частоти на іншу, при цьому зміна (стрибок) має бути не менша 6 МГц. Зміна частот може бути повільною (slow hopping), при якій частота несучої є постійною протягом передавання декількох інформаційних розрядів, та швидкою (fast hopping), коли зміна несучої відбувається декілька разів протягом передавання одного біта даних.

Метод швидкого розширення спектра більш стійкий до похибок, оскільки вузькосмугова похибка, яка пошкоджує сигнал в деякому підканалі, не призводить до втрати біта. Крім того, при реалізації цього методу відсутня міжсимвольна інтерференція. Однак метод повільного розширення спектра значно більш простий в реалізації і має значно менші накладні витрати.

Методи FHSS використовуються в технологіях IEEE 802.11 та Bluetooth.

В методі **прямого послідовного розширення спектра DSSS** також використовується весь частотний діапазон, який виділений для даного безпроводового каналу. Але, на відміну від методу FHSS, весь частотний діапазон використовується не за рахунок постійних переключень з однієї частоти на іншу, а за рахунок того, що кожний біт інформації подається N бітами таким чином, що тактова швидкість передавання збільшується в N разів. При такому підході достатньо вибрати відповідним чином швидкість передавання даних і значення N, щоб спектр сигналу зайняв весь діапазон.

Код, на який замінюється біт початкової інформації, називають **послідовністю розширення**, а кожний біт такої послідовності – **чипом**. Відповідно до цього швидкість передавання підсумкового коду називають **чиповою швидкістю**. Модулі передавача та приймача мають знати послідовність розширення для коректної взаємодії. Кількість бітів в цій послідовності визначає **коефіцієнт розширення коду**, який, зазвичай, має значення від 3 до 100. Для кодування бітів отриманого підсумкового коду використовується будь-який метод модуляції, наприклад, BPSK. При збільшенні коефіцієнта розширення збільшується і спектр підсумкового сигналу, і відповідно зменшується ймовірність його пошкодження.

В безпроводових мережах зазвичай значення кожного біта кодується за допомогою надлишкової k-бітової послідовності Баркера, довжина якої лежить в інтервалі 3–13. Таким чином бітовий інтервал t розбивається на k чипів, кожний тривалістю $\tau = t / k$. Таке подання інформаційного сигналу дозволяє сформувати широкосмуговий сигнал тривалістю τ та смугою пропускання $F = k / t$, що в k раз перевищує смугу пропускання початкового інформаційного сигналу.

В мережах стандарту IEEE 802.11 використовується послідовність, яка містить 11 елементів, і має вигляд 11100010010 (або, як альтернатива, 101101110000). Існують пряма послідовність Баркера, в якій одиничний

інформаційний біт передається прямим кодом, а нульовий – інверсним, та інверсна послідовність, за допомогою якої одиничний біт передається інверсним кодом, а нульовий – прямим.

Отримана послідовність чипів модулюється відповідно до способу, який використовується в даному каналі, наприклад, BFSK або BPSK.

Наприклад, якщо необхідно передати в канал тетраду 1011, а розширювальною послідовністю Баркера є послідовність 11100010010, то передавач формує і передає в канал таку послідовність бітів:

11100010010 00011101101 11100010010 11100010010.

Послідовність Баркера дозволяє приймачу швидко синхронізуватися з передавачем, тобто надійно виявляти початок послідовності, що передається в каналі. Для цього приймач порівнює прийняту послідовність зі зразком послідовності, і навіть при пошкодженні декількох бітів з великою ймовірністю приймач правильно визначить початок послідовності і тому коректно може інтерпретувати отриману послідовність.

Метод DSSS забезпечує меншу захищеність від похибок, ніж метод FHSS, оскільки вузькосмугова похибка впливає на частину спектра, а значить і на результат розпізнавання одиничних і нульових бітів.

Безпроводові мережі, в яких використовується метод DSSS, використовують канали шириною 22 МГц, завдяки чому багато WLAN можуть функціонувати в одній зоні покриття. В Північній Америці та більшій частині Європи в діапазоні 2,4–2,483 ГГц канали шириною 22 МГц (за смугою розфільтрування 3 МГц між ними) дозволяють створити три канали передавання, які не перекриваються.

В безпроводових мережах широко використовується ортогональне частотне мультиплексування **OFDM** (Orthogonal Frequency Division Multiplexing), яке передбачає, що весь частотний діапазон розбивається на деяку достатньо велику кількість підканалів (subcarrier), що називають «кишенями». Одному каналу (і передавачу, і приймачу) виділяють з цієї сукупності N таких підканалів. При передаванні вихідний потік розбивається на N підпотоків, і передавання відбувається паралельно по всіх підканалах, причому розподілення підканалів в процесі функціонування може динамічно змінюватись. Тобто, при використанні OFDM високошвидкісний потік даних конвертується в декілька паралельних бітових потоків меншої швидкості, кожний з яких модулюється своєю окремою несучою. Вся ця сукупність несучих передається одночасно. Такий підхід дозволяє знизити міжсимвольну інтерференцію, яка стає проблемою при високих швидкостях передавання даних. Головна перевага технології OFDM полягає в тому, що тривалість символа в допоміжній несучій значно більша порівняно з затримкою розповсюдження, ніж у традиційних схемах модуляції.

Переваги технології OFDM:

- адаптивність, тобто можливість використання різних схем та алгоритмів модуляції для різних підканалів, що дозволяє адаптуватись не тільки до умов розповсюдження сигналу в каналі, а й до різних

вимог до якості передавання сигналу (найчастіше при модуляції сигналів різних підканалів в технології OFDM використовуються методи фазової PSK і квадратурно-амплітудної QAM модуляцій);
- достатньо проста реалізація методами цифрової обробки;
- можливість зниження міжсимвольної інтерференції (між підканалами), що особливо важливо при багатопроменевому розповсюдженні;
- висока спектральна ефективність, яка характеризується тим, що при збільшенні кількості підканалів OFDM системи забезпечують майже вдвічі кращу спектральну ефективність порівняно з традиційними системами з частотним мультиплексуванням.

Недоліки OFDM:
- необхідна дуже точна синхронізація за часом та частотою;
- OFDM сигнал має відносно високе значення пік-фактора, що призводить до занадто високих енергетичних витрат;
- використання захисних інтервалів знижує спектральну ефективність методу;
- на метод ортогонального частотного мультиплексування суттєво впливає ефект Доплера, що призводить до додаткових ускладнень при його використанні в мобільних мережах.

Модуляція OFDM використовується в системах сотового зв'язку WiMAX, MobileWiMAX, системах цифрового телебачення, MBWA, системах типу «розумний дім» та «розумне місто» та багатьох інших системах. Цей принцип модуляції широко використовується в безпроводових мережах стандартів IEEE 802.11a,e,g,n; 802.16a,d,e; 802.20.

Існують такі модифікації технології OFDM, які використовуються різними компаніями в різних застосуваннях:
- **COFDM** (Coded OFDM). Даний вид OFDM відрізняється лише тим, що дані попередньо кодуються за допомогою кодів корекції. Використовуються в системах цифрового телебачення DVB-T;
- **Flash OFDM** (Fast low-latency access with seamless handoff OFDM). Ця модифікація розроблена компанією Flarion Technologies для мобільних пристроїв. Особливості модифікації полягають в алгоритмах роботи в режимі з комутацією пакетів даних;
- **OFDMA** – багатокористувацький варіант OFDM технології;
- **VOFDM** (Vector OFDM). Дана модифікація розроблена компанією Cisco Systems. В основі лежить концепція технології MIMO. Існує також варіант MIMO-OFDM;
- **WOFDM** (Wideband OFDM). Широкосмугова модифікація OFDM, яка розроблена Wi-LAN Inc. Дана модифікація забезпечує підвищення пропускної спроможності та завадостійкості. Основна відмінність полягає в більшій частотній відстані між несучими.

Слід відзначити, що серед розглянутих модифікацій найбільше розповсюдження отримали класична схема OFDM і модифікація COFDM.

9.4 Локальні мережі WLAN

Безпроводові локальні мережі **WLAN** (Wireless LAN) на сьогодні набувають все більшого використання, оскільки можуть функціонувати паралельно з проводовими мережами, а іноді є єдино можливим рішенням організації з'єднання модулів користувачів. Мережі WLAN часто називають мережами **Wi-Fi** (Wireless Fidelity – висока точність відтворення з використанням безпроводової технології). В безпроводових мережах інформація передається за допомогою електромагнітних хвиль високої частоти. А це призводить до нестійкої та непередбачуваної роботи мережі, оскільки завади від різних телекомунікаційних систем та модулів, відбиття сигналів, атмосферні завади призводять до ненадійного прийому інформації. В зв'язку з цим в мережах WLAN використовуються складні методи кодування інформації та сигналів, які зменшують вплив завад на корисний сигнал.

Безпроводові локальні мережі зазвичай створюються на основі сукупності стандартів IEEE 802.11, перший з яких був прийнятий в 1997 році, а також прийнятий ETSI під назвою ETS 300 328. У стандарті визначено параметри різних варіантів реалізації фізичного рівня протоколу, а також підрівнів доступу до каналу, які незалежні від безпроводового середовища передавання. Поточний перелік стандартів сімейства 802.11 та їх загальну характеристику наведено в додатку А.9.1, а параметри локальних мереж стандарту IEEE 802.11 наведені в таблиці 9.2.

Таблиця 9.2 – Параметри мереж стандарту 802.11

	802.11a	**802.11b**	**802.11g**	**802.11n**	**802.11y**	**802.11ac**	**802.11ad**
Рік прийняття	1999	1999	2003	2009		2012	2014
Частота, ГГц	5	2,4	2,4	2,4–5	3,65–3,7	5	2,4–5
Пропускна спроможність (реальна), Мбіт/с	54 (25)	11 (5)	54 (25)	74-248 (100)	54 (25)	1,3 Гбіт/с	до 7 Гбіт/с
Модуляція	QPSK, BPSK, QAM16, QAM64	QPSK-DSSS	QPSK, BPSK, QAM16, QAM64	OFDM		QAM256	
Ширина каналу, МГц	20	25	20	20, 40			60
Радіус дії, м	50	100	100	70–150	5 км (у відкритому середовищі)		

Мережі стандарту IEEE 802.11 підтримують такі топології:

- незалежні базові зони обслуговування IBSSs (Independent Basic Service Sets);
- базові зони обслуговування BSSs (Basic Service Sets);
- розширені зони обслуговування ESSs (Extended Service Sets).

Зоною обслуговування (Service Set) називають сукупність пристроїв, які знаходяться в радіусі дії (зоні покриття) безпроводової мережі. Зона покриття значною мірою залежить від потужності станції-передавача, наявності та типу перешкод між станціями, режиму роботи (топології) безпроводової мережі тощо. Технологія WLAN забезпечує доступ до мережі за рахунок передавання широкомовних сигналів через ефір на несучій частоті в діапазоні радіочастот. Станція-одержувач може отримувати сигнали в діапазоні роботи декількох станцій-передавачів.

Незалежні базові зони обслуговування IBSSs являють собою сукупність станцій, які функціонують відповідно до стандарту IEEE 802.11 і з'єднуються між собою безпосередньо (рис. 9.3). Оскільки такі мережі насправді є простими одноранговими WLAN, їх також називають спеціальними, неплановими Ad-Hoc мережами. Така незалежна база обслуговування створюється, коли окремі клієнтські пристрої формують мережу без використання точки доступу, тобто безпосередньо встановлюють з'єднання між собою та обмінюються даними. Обмежень на кількість клієнтських станцій в такій мережі стандартом не визначається, але оскільки розподілення доступу між станціями виконується децентралізовано, їх кількість не може бути великою. Для такого режиму необхідно, щоб кожна станція мала безпроводовий адаптер. Недоліками режиму Ad-Hoc є обмежений діапазон дії такої мережі та неможливість підключення до зовнішньої мережі, наприклад, до мережі Internet.

Базова зона обслуговування BSS передбачає наявність **точки доступу AP (Access Point)**, яка є центральним модулем для зв'язку всіх станцій створеної зони. Клієнтські станції не можуть обмінюватись інформацією безпосередньо між собою, це реалізується тільки через точки доступу, які, зазвичай, мають порт висхідного каналу (uplink port), за допомогою якого забезпечується підключення до проводової мережі. Завдяки цьому таку топологію називають **інфраструктурою BSS**.

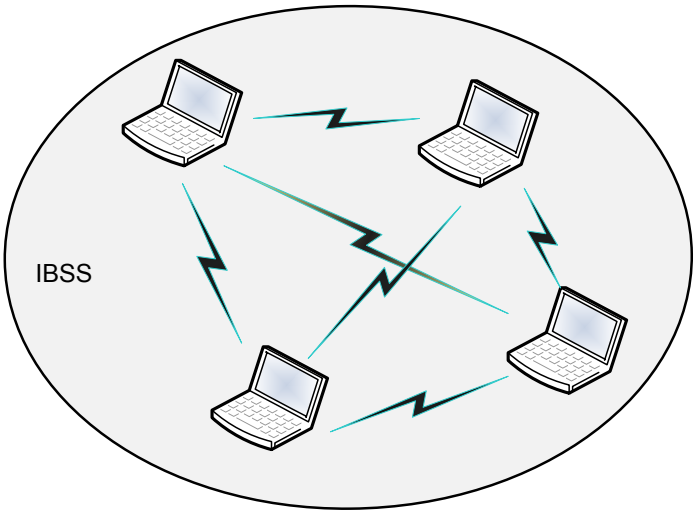


Рисунок 9.3 – Незалежна Ad-Hoc мережа (**IBSS)**

Декілька базових зон обслуговування BSS за допомогою портів висхідного каналу можуть підключатися до розподільної системи (зовнішньої мережі), створюючи при цьому **розширену зону обслуговування ESS**. На рис. 9.4 дві базові зони BSS об'єднані в єдину розширену зону ESS. Між собою точки доступу з'єднуються за допомогою проводових сегментів зовнішньої мережі або радіомостів.

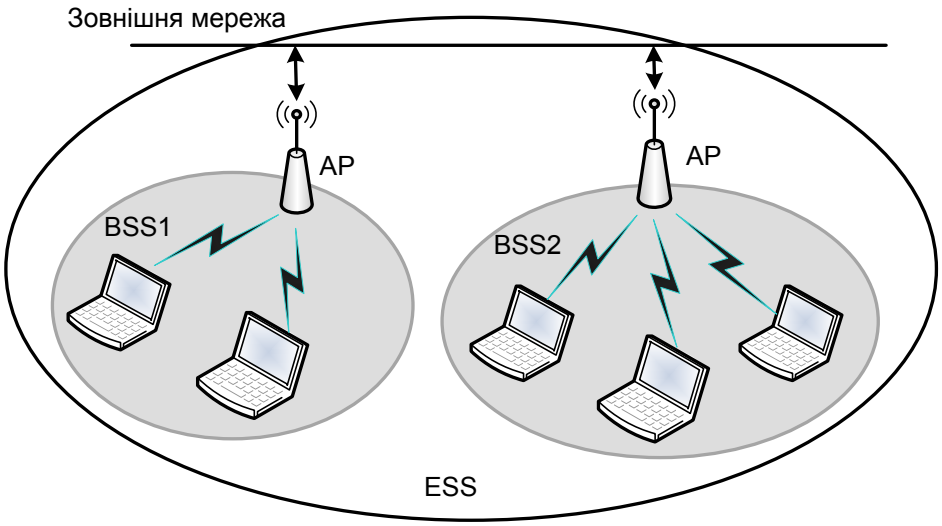


Рисунок 9.4 – Розширена зона обслуговування ESS безпроводової мережі

Стандарт IEEE 802.11 відповідає загальній структурі стандартів комітету 802 і визначає особливості реалізації двох нижніх рівнів: фізичного і канального з підрівнями LLC та MAC (рис. 9.5). При цьому підрівень LLC виконує стандартні, загальні для всіх технологій LAN функції.

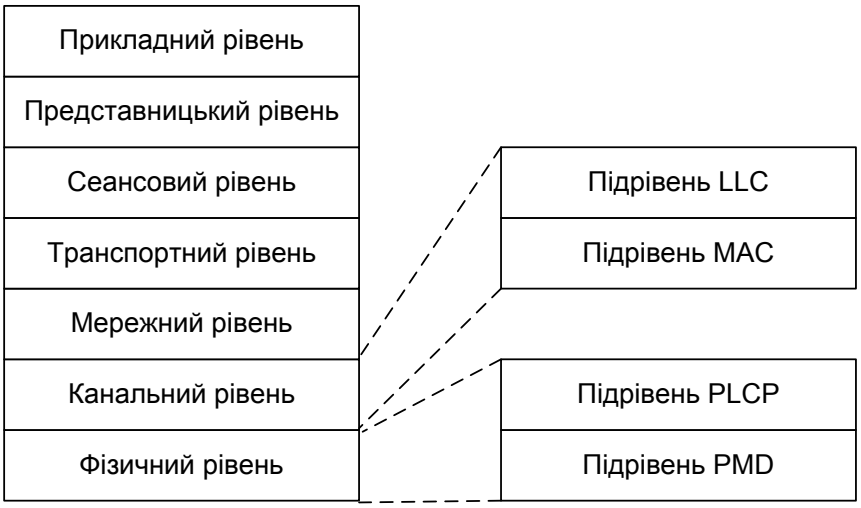


Рисунок 9.5 – Підрівні фізичного рівня IEEE 802.11

Кожний фізичний рівень стандарту IEEE 802.11, як і канальний, має два підрівні:

- підрівень визначення стану фізичного рівня PLCP (Physical Layer Convergence Procedure);
- підрівень фізичного рівня, який залежить від середовища передавання PMD (Physical Medium Dependent).

Підрівень PLCP є рівнем, який забезпечує передавання блоків даних протоколу MAC між MAC-станціями з використанням підрівня PMD, на якому реалізовано метод передавання та прийому даних через безпроводове середовище. Підрівні PLCP та PMD відрізняються для різних варіантів стандарту IEEE 802.11.

На фізичному рівні існує декілька специфікацій, які відрізняються частотним діапазоном, методом кодування та швидкістю передавання (рис. 9.6).

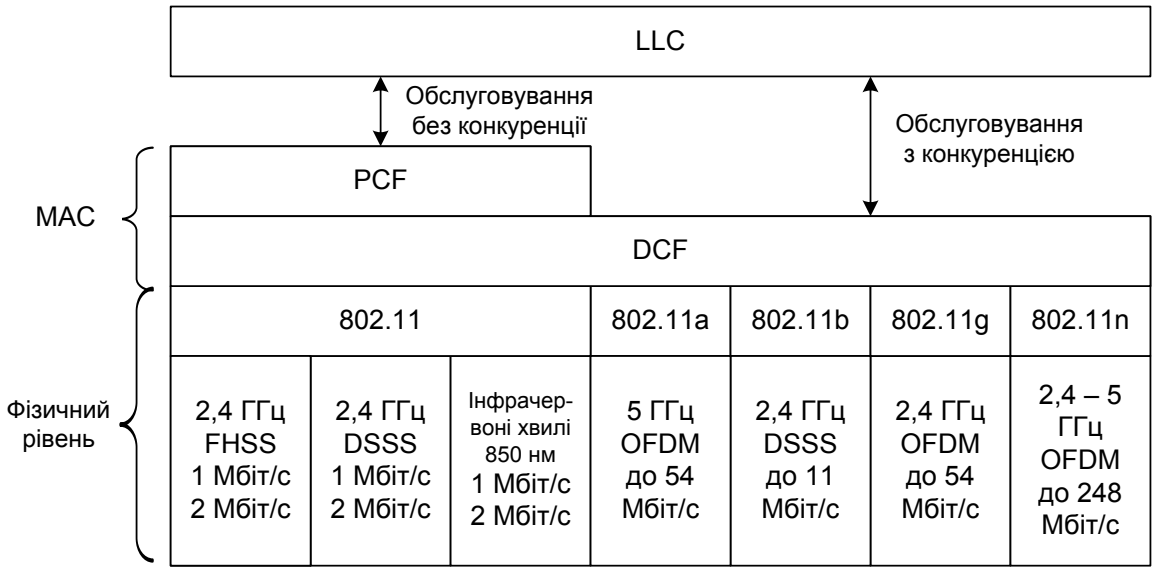


Рисунок 9.6 – Стек протоколів IEEE 802.11

На підрівні **MAC** в безпроводових мережах стандарту IEEE 802.11 реалізовано два режими доступу до середовища, що розподіляється між всіма користувачами:

- розподілений режим доступу **DCF (Distributed Coordination Function)**;
- централізований режим доступу **PCF (Point Coordination Function)**.

Розподілений режим DCF передбачає використання алгоритму множинного доступу з уникненням колізій **CSMA/CA**, який для безпроводових мереж є значно більш ефективним, ніж алгоритм **CSMA/CD**, який розпізнає колізії, а не попереджує їх. Для цього станція безпроводової мережі намагається передати кожний кадр таким чином, щоб знизити ймовірність його зіткнення з іншим кадром. Крім того, кожний переданий кадр необхідно підтверджувати кадром квитанції, який формується і відправляється станцією-одержувачем. Якщо після завершення тайм-ауту кадр квитанції не отримано, станція-відправник вважає, що виникла колізія, і буде намагатися відправити кадр повторно. Алгоритм передавання даних в канал полягає в нижчевикладеному.

Станція, якій потрібно передати кадр, спочатку обов'язково прослуховує середовище передавання. Для виявлення несучої в каналі (наявності передавання даних інших станцій) реалізовано два механізми: фізичний та віртуальний. Перший функціонує на фізичному рівні і визначає рівень сигналу в антені порівняно з його встановленим пороговим значенням. Віртуальний механізм виявлення несучої передбачає, що всі кадри (як даних, так і керівні) містять інформацію про час передавання кадру (або сукупності кадрів) та отримання підтвердження про його прийом. Тобто, всі станції мережі мають інформацію про поточне передавання і можуть визначити коли канал буде звільнено, і як тільки це відбувається, станція відраховує часовий інтервал, який дорівнює встановленому значенню міжкадрового інтервалу ***IFS*** (InterFrame Space). Якщо після завершення цього інтервалу середовище ще вільне, починається відлік часових слотів фіксованого значення (рис. 9.7). Кожна станція може розпочати передавання кадру тільки на початку будь-якого слоту, якщо середовище передавання вільне. Номер слоту вибирається випадково з множини значень, що визначає розмір конкурентного вікна CW (Contention Window).

Розмір слоту вибирається таким чином, щоб він перевищував час розповсюдження сигналу між двома будь-якими станціями разом з часом, який витрачається на розпізнавання зайнятості середовища. Розмір слоту залежить від способу кодування і становить для методу FHSS 28 мкс, а для методу DSSS – 1 мкс. При виконанні такої умови кожна станція, прослуховуючи слоти, зможе коректно розпізнати початок передавання кадру іншою станцією, який реалізується раніше того часову слоту, який станція обрала для свого передавання. Таким чином, в даному випадку колізія може виникнути тільки в тому випадку, якщо декілька станцій обирають один і той же слот для передавання.

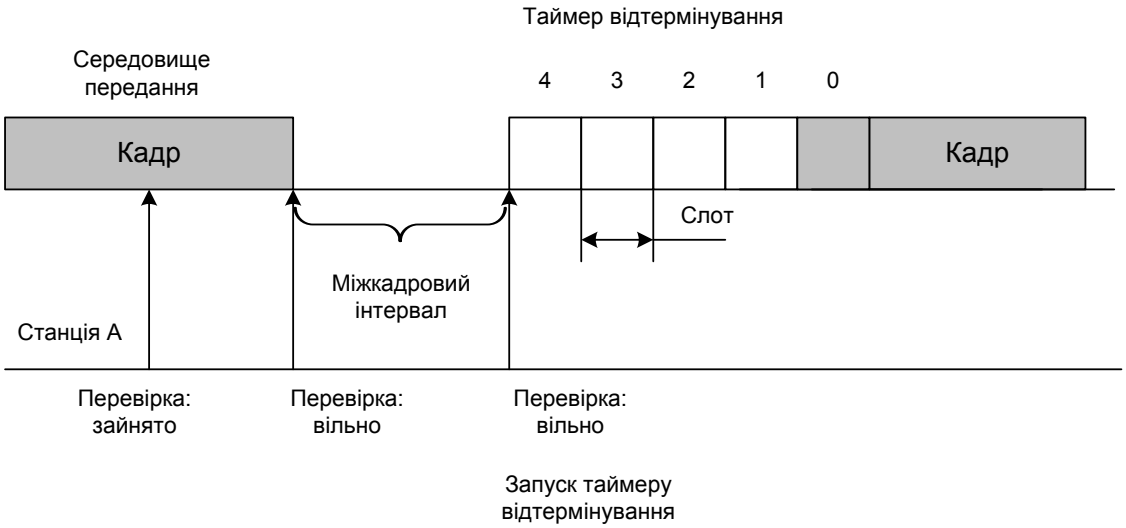


Рисунок 9.7 – Режим розподіленого доступу DCF

Для прикладу, наведеного на рис. 9.7, станція, що має дані для передавання, прослуховує канал і після виявлення вільного стану вибирає на основі відповідного алгоритму значення слоту, наприклад, 4, яке і присвоюється таймеру відтермінування. На початку кожного слоту стан середовища передавання знову перевіряється, а від поточного значення таймера відтермінування віднімається 1. Як тільки значення таймера дорівнюватиме нулю, розпочинається передавання кадру. Якщо ж на початку будь-якого слоту канал виявився зайнятим, значення таймера «заморожується» (без зменшення поточного значення слоту на 1), і починається новий цикл доступу до каналу. При цьому виконується контроль за станом середовища, і при його звільненні встановлюється пауза, що дорівнює міжкадровому інтервалу, а потім довільно встановлюється нове значення слоту. Якщо ж канал залишився вільним, то станція використовує значення «замороженого» таймера і виконує перевірку стану середовища на початку кожного слоту. Наприклад, якщо в першому циклі перевірки таймер було заморожено на значенні 1, то з цього значення починається новий цикл перевірки.

Потрібно зауважити, що на початку слоту станція не починає безпосереднє передавання кадру даних. Замість цього спочатку станція передає короткий службовий кадр запиту на передавання **RTS** (Request To Send). У відповідь на цей запит станція призначення формує і передає службовий кадр готовності до передавання **CTS** (Clear To Send), і тільки після цього станція може передати кадр даних. Кадр CTS, крім підтвердження готовності, має й інше призначення, а саме: він має попередити про захоплення середовища станції, які не знаходяться в зоні сигналу станції-відправника, але є в зоні сигналу станції-одержувача, тобто є прихованими вузлами для станції-відправника.

Централізований режим доступу PCF відрізняється від розподіленого тим, що точка доступу мережі виконує ще і функції арбітра та послідовно надає право доступу тим станціям, яким необхідно передати кадри. Реа-

лізується також пріоритетне обслуговування мережного трафіку, наприклад, за необхідності передавання з мінімальними затримками. Але для запобігання повного захоплення каналу таким потоком обмежується тривалість цього режиму.

Початковий стандарт IEEE 802.11 визначає **такі методи передавання на фізичному рівні**:

- передавання в діапазоні інфрачервоних хвиль;
- технологія розширення спектра шляхом стрибкоподібної зміни частоти FHSS в діапазоні 2,4 ГГц;
- технологія широкосмугової модуляції з розширенням спектра методом прямої послідовності DSSS в діапазоні 2,4 ГГц;
- технологія ортогонального частотного мультиплексування OFDM.

При передаванні в діапазоні **інфрачервоних хвиль** використовуються хвилі діапазону 850 нм, які генеруються або напівпровідниковим лазером, або світлодіодом. В цьому випадку ділянка покриття LAN обмежена зоною прямої видимості, оскільки інфрачервоні хвилі не проникають через стіни. Стандарт передбачає три варіанти розповсюдження випромінювання: ненаправлену антену, відбиття від стелі та фокусне направлене випромінювання, яке призначене для організації з'єднання типу «point-to-point», наприклад, між двома будовами.

Безпроводові локальні мережі з використанням **технології FHSS** підтримують швидкості передавання 1 Мбіт/с та 2 Мбіт/с. Пристрої FHSS розділяють призначену для їх роботи смугу частот від 2,402 ГГц до 2,48 ГГц на 79 каналів, які не перекриваються, що справедливо для Північної Америки та більшої частини Європи. Ширина кожного з 79 каналів становить 1 МГц, тому такі мережі забезпечують достатньо високу швидкість передавання символів та невелику швидкість зміни каналів.

Технологія **широкосмугової модуляції з розширенням спектра методом прямої послідовності DSSS** більш ефективна, ніж FHSS, але і більш складна в реалізації. При реалізації цього методу підвищується тактова частота модуляції і кожному символу повідомлення, що передається в канал, ставиться у відповідність деяка достатньо довга псевдовипадкова послідовність. Такі безпроводові мережі використовують канали шириною 22 МГц, завдяки чому декілька WLAN можуть функціонувати в одній зоні покриття. В Північній Америці та більшій частині Європи такі канали дозволяють створити в діапазоні від 2,4 ГГц до 2,483 ГГц три канали передавання, що не перекриваються. При швидкості 1 Мбіт/с використовується двійкова відносна фазова модуляція DBPSK (Differential Binary Phase Shift Keying), при цьому одиничний біт подається 11-елементним кодом Баркера 11100010010, а нульовий – інверсним кодом Баркера. Символи коду Баркера не переносять інформацію, біти передаються одразу всім кодом Баркера: прямим та інверсним, що дозволяє надати сигналу властивості шуму, що забезпечує завадостійкість. Для швидкості 2 Мбіт/с використовується квадратурна відносна фазова модуляція DQPSK (Quadrature Phase Shift

Keying), яка передбачає передавання чотирьох різних станів сигналу, що дозволяє за один цикл синхронізації передавати одразу 2 біти.

Технологія **ортогонального частотного мультиплексування OFDM** (Orthogonal Frequency Division Multiplexing) передбачає паралельне передавання корисного сигналу одночасно по декількох частотних діапазонах (на відміну від технологій FHSS та DSSS, в яких сигнали передаються послідовно), що дозволяє підвищити не тільки пропускну спроможність каналу, але й якість сигналу. В даній технології кожний високошвидкісний канал має ширину 20 МГц і складається з 52 підканалів шириною 300 кГц. Зазвичай дана технологія використовується в діапазоні 5 ГГц, в якому функціонує стандарт IEEE 802.11a, де передбачена швидкість передавання даних до 54 Мбіт/с (стандартом визначені три обов'язкові швидкості: 6, 12 та 24 Мбіт/с, та п'ять необв'язкових: 9, 18, 36, 48 та 54 Мбіт/с). Пропускна спроможність залежить від способу модуляції, який використовується в даному випадку.

Для створення безпроводової локальної мережі необхідне спеціальне обладнання, до якого відносять адаптери, точки доступу, маршрутизатори та антени Wi-Fi. Сучасні комп'ютери, ноутбуки, планшети тощо мають вбудовані безпроводові адаптери, а в разі їх відсутності адаптер підключається через USB-роз'єм. Наявність безпроводових адаптерів забезпечує взаємодію тільки в рамках локальної мережі. В разі потреби підключення до проводової мережі необхідно використовувати точку доступу, а в деяких випадках необхідне використання зовнішньої антени Wi-Fi. Безпроводові антени можуть бути різних конфігурацій та направленості. Направлена антена використовується для організації зв'язку типу мостового з'єднання в конфігурації «point-to-point». Кругова антена функціонує в конфігурації «point-to-multipoint», має кругову зону покриття і зазвичай використовується в маршрутизаторах та точках доступу.

Для організації мереж стандарту IEEE 802.11n використовуються достатньо нові підходи: багатоантенні системи множинного введення-виведення **MIMO** (Multiple Input Multiple Output), які передбачають використання мінімум двох систем, що передають інформацію, і двох систем, що її приймають (в даному випадку 2×2). При передаванні вихідний потік розділяється на декілька потоків, кількість яких залежить від кількості антен (рис. 9.8). Далі кожен з отриманих потоків буде передаватися через свою антену, причому, зазвичай, сигнал з кожної антени передається з різною поляризацією, що дозволяє достатньо легко ідентифікувати його при прийомі.

Розрізняють симетричні системи MIMO з однаковою кількістю передавальних і приймальних антен (наприклад, 2×2, 4×4, 8×8) та несиметричні з різною кількістю таких антен (2×4, 4×2, 2×8, 4×8 тощо).

Максимальна пропускна спроможність при використанні MIMO залежить як від кількості незалежних каналів, так і його ширини та якості параметрів передавання, й може бути розрахована за формулою

$$C = M\,B\,\log_2(1 + S/N),$$

де C – пропускна спроможність каналу;
M – кількість незалежних потоків даних;
B – ширина каналу;
S/N – співвідношення сигнал/шум.

Прикладом безпроводових мереж, які широко використовують MIMO, є технологія AirMAX. Суть її полягає в тому, що як передавання, так і прийом сигналу декількома антенами в одному каналі структурується та впорядковується TDMA з апаратним прискоренням: пакети даних рознесено по окремих часових слотах, а черги передавань координовано. Це дозволяє уникнути колізій в каналі, завдяки чому збільшуються як пропускна спроможність каналу, так і кількість абонентських систем до 120 (замість 20–25 при стандартному підключенні до точки доступу Wi-Fi).

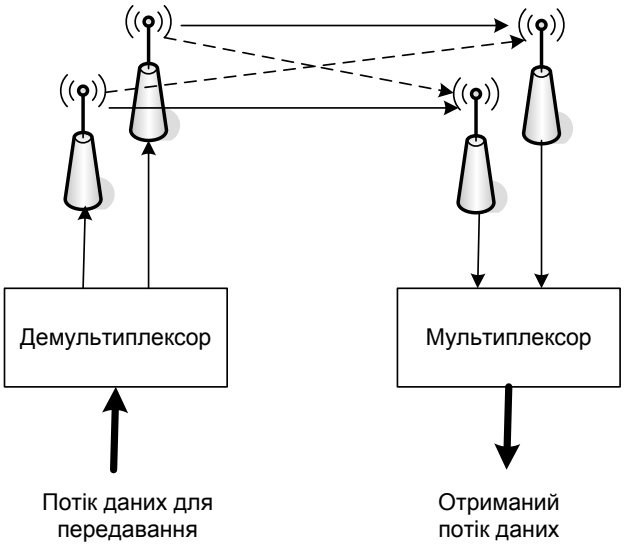


Рисунок 9.8 – Принцип роботи технології MIMO 2×2

Останнім часом набувають розповсюдження безпроводові широкосмугові **Mesh-мережі** стандарту **IEEE 802.11s** WMN (Wireless Mesh Network), які часто ще називають комірчастими, що використовуються для передавання мультимедійної інформації. Такі мережі називають також MBSS (Mesh BSS, Mesh Basis Service Set). Ці мережі знаходять широке застосування при створенні локальних і розподілених міських безпроводових мереж. Одним з головних принципів побудови Mesh-мереж є принцип самоорганізації архітектури, що забезпечує такі можливості, як реалізація топології мережі «кожний з кожним», живучість мережі при відмові окремих модулів, динамічну маршрутизацію трафіку, контроль стану мережі, просту масштабованість мережі, внаслідок чого збільшується зона покриття в режимі самоорганізації та інші можливості. В загальному випадку Mesh-мережі можуть бути як стаціонарними, так і мобільними, в яких окремі станції або і всі можуть змінювати своє розташування в процесі роботи. В

цьому випадку як термінальні станції (ТС) можуть використовуватись різноманітні пристрої: планшети, мобільні телефони тощо.

В класичних статичних конфігураціях безпроводових мереж стандарту IEEE 802.11 термінальні станції підключаються до точок доступу AP (Access Point) і можуть взаємодіяти тільки з ними, обмін даними між ТС і доступ до зовнішньої мережі можливий тільки через точки доступу (рис. 9.9).

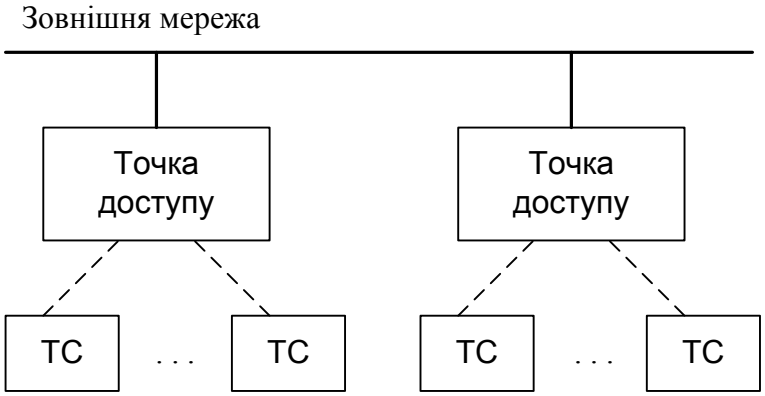


Рисунок 9.9 – Стандартна архітектура мережі стандарту IEEE 802.11

У Mesh-мережі, крім термінальних абонентських станцій (ТС), обов'язково присутні спеціальні пристрої – Mesh-портали, які з'єднують Mesh-мережу з зовнішніми мережами. WMN зазвичай містять окрему опорну мережу (backbone) з маршрутизаторів (Mesh-роутерів).

Mesh-портали забезпечують доступ до мереж різних типів і функціонально подібні до шлюзів. Ці системи обробляють великі інформаційні потоки від всіх клієнтських станцій Mesh-мережі.

Встановлення з'єднання IEEE 802.11s реалізується за рахунок періодичного відправлення стандартного повідомлення «Відкрити з'єднання», у відповідь на яке можуть бути отримані повідомлення «Підтвердження з'єднання» або «Закриття з'єднання». З'єднання між двома сусідніми Mesh-порталами вважається встановленим тільки тоді, коли взаємодійні портали відправили один одному команди «Відкрити з'єднання» та відповіли підтвердженням з'єднання в будь-якій послідовності. Для кожного встановленого з'єднання призначається час життя, протягом якого воно є дійсним і має бути використане.

Всі вузли мереж такого типу є багатофункціональними пристроями, які можуть взаємодіяти як один з одним, так і з термінальними станціями, виконуючи при цьому необхідні Mesh-функції. Узагальнена структура Mesh-мережі стандарту IEEE 802.11s наведена на рисунку 9.10, з якого видно, що термінальні станції можуть взаємодіяти безпосередньо між собою та комунікаційними вузлами, які об'єднують функції точок доступу і Mesh-служб.

Абонентські системи, що підключаються до Mesh-мереж, можуть бути двох типів: термінальними клієнтськими станціями мережі стандарту IEEE 802.11 будь-якої групи, які функціонують в режимі інфраструктури, та Mesh-станціями, що виконують функції трансляції та маршрутизації в сусідні вузли мережі.

Mesh-точки доступу, крім стандартних функцій точки доступу інфраструктурного режиму для підключених термінальних клієнтських станцій, виконують ще і функції маршрутизації з вузлами мережі маршрутизаторів (Mesh-роутери).

Mesh-роутери, зазвичай, мають декілька інтерфейсів і виконують функції маршрутизації.

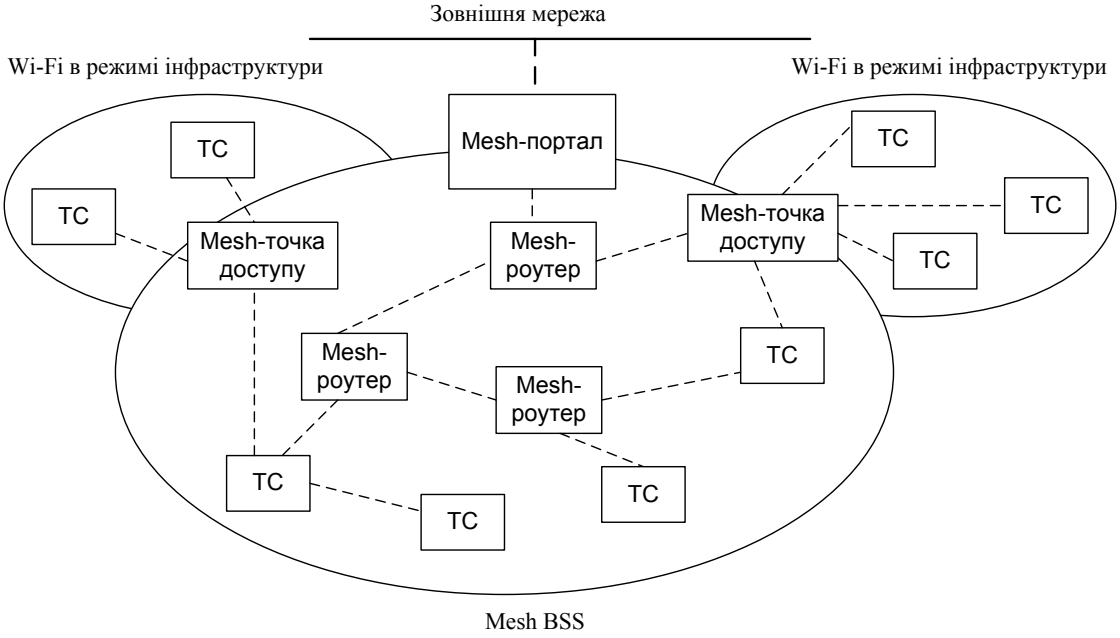


Рисунок 9.10 – Узагальнена архітектура Mesh-мережі стандарту IEEE 802.11s

Порівняно з мобільними мережами Ad Hoc Mesh-мережі мають переваги, які полягають в більшій надійності, пропускній спроможності тощо. Безперечною перевагою Mesh-мереж є мінімальний час і простота розгортання, що пояснюється її самоорганізацією, самоадапацією та самовідновленням в обхід пошкодженого каналу (або цілої ділянки), забезпечення зв'язку на території, де традиційні мережі не можуть це реалізувати.

Необхідно відмітити, що зміни в стандарті IEEE 802.11s не торкаються фізичного рівня – всі нововведення стосуються лише MAC-рівня. Структура кадрів MAC-рівня відрізняється від стандартного формату кадру мереж стандарту 802.11 тим, що в ньому, додатково до всіх стандартних опцій, присутній mesh-заголовок. Крім цього, в заголовку присутнє поле «Час життя», яке використовується при багатокроковому передаванні для кожного вста-

новленого з'єднання та запобігає зациклюванню. Для боротьби з можливими дублікатами пакетів передбачено поле нумерації пакетів.

Крім того, в стандарті IEEE 802.11s розглянуті питання маршрутизації пакетів в рамках Mesh-мережі (тобто мережний і транспортний рівні еталонної моделі OSI).

Недоліком даної технології є те, що мережа стандарту IEEE 802.11s визначена для невеликих комірчастих розмірів з максимальною кількістю вузлів до 64.

Розглянемо особливості та основні характеристики технології WiGig, яка описується стандартами 802.11ad і 802.11ay, та її відмінності від Wi-Fi 6. На відміну від стандартних мереж будь-якого типу стандарту Wi-Fi, технологія WiGig є більш універсальною, яка дозволяє об'єднувати в єдину мережу різні за своїм призначенням модулі (комп'ютери, відеокамери, сотові телефони тощо) та передавати великі обсяги даних з великою швидкістю. Поточна версія WiGig забезпечує швидкість передавання в реальних умовах 5 Гбіт/с, а в деяких умовах 7 Гбіт/с (покращена версія – до 10 Гбіт/с), на відміну від Wi-Fi 6, який дозволяє передавати дані з реальною швидкістю 2 Гбіт/с (з можливістю подальшого збільшення).

В технології WiGig для передавання використовується частота 60 ГГц, що є менш навантажена, на відміну від Wi-Fi 6 та інших стандартних версій Wi-Fi, які використовують частоти 2,4 ГГц і 5 ГГц, що і забезпечує можливість передавати значно більші обсяги даних.

Однак використання більш коротких довжин хвиль призводить до суттєвих обмежень щодо дальності передавання, яка на сьогодні дорівнює 10 м, і до проходження перешкод (стіни, стелі тощо), що в принципі, є перевагою при побудові систем «розумного дому».

9.5 Мережі WIMAX

Технологія **WIMAX** (Worldwide Interoperability for Microwave Access) забезпечує взаємодію станцій, які знаходяться на значно більшій відстані, ніж забезпечує стандарт IEEE 802.11, і є безпроводовою технологією, що забезпечує широкосмуговий доступ з високою пропускною спроможністю. Технологія WIMAX забезпечує створення регіональних мереж і описана в стандартах IEEE 802.16, для розробки яких в 2001 році був сформований WIMAX-форум, що об'єднує на сьогодні більше 230 компаній. Всесвітній саміт з інформаційного суспільства WSIS (World Summit on Information Society) визначив, що основне завдання технології WIMAX полягає в забезпеченні універсального безпроводового доступу для пристроїв різних типів та класів (робочих станцій, мобільних телефонів, побутових пристроїв «розумного дому» тощо), а також для локальних мереж (проводових і безпроводових).

Для створення мереж WIMAX необхідна наявність компонентів двох типів (рис. 9.11):

- **базової станції WIMAX**, яка має знаходитись на будь-якому висотному об`єкті (вишці або будівлі);
- **станція приймача WIMAX**: антени з приймальним пристроєм WIMAX.

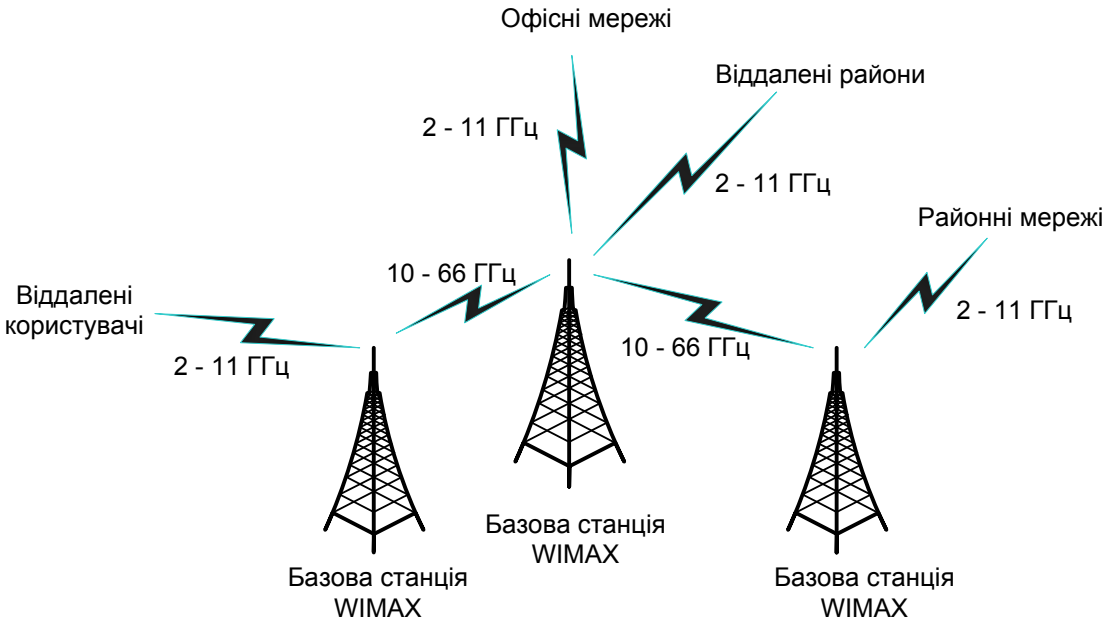


Рисунок 9.11 – Архітектура WIMAX

З'єднання базової станції з клієнтським пристроєм реалізується в надвисокому частотному діапазоні 2–11 ГГц, який дозволяє підтримувати канали з шириною смуги пропускання від 1,5 до 20 МГц (в деяких версіях стандарту до 50 МГц). Різні смуги частот використовуються в різних типах доступу для надання різних послуг. При цьому базова станція та клієнтські модулі не обов'язково мають знаходитись в зоні прямої видимості.

На сьогодні для мереж WIMAX використовується три основних діапазони частот: 2,5–2,7; 3,4–3,6 та 5–6 ГГц, кожний з яких має як обмеження, так і сфери використання з реалізацією відповідного типу доступу до ресурсів мережі.

Між сусідніми базовими станціями встановлюється постійне з'єднання прямої видимості в діапазоні 10–66 ГГц, що дозволяє забезпечити передавання даних (за ідеальних умов) зі швидкістю до 134 Мбіт/с. З сукупності базових станцій хоча б одна постійно з'єднана з мережею провайдера швидкісним каналом, наприклад, T3/E3. Зазвичай одна базова станція може одночасно обслуговувати більше 60 корпоративних користувачів на швидкостях, аналогічних каналам T1/E1, або більше 100 домашніх користувачів. Суттєвою перевагою мереж WIMAX є забезпечення стабільного передавання при відсутності прямої видимості базової станції, що дозволяє підт-

римувати стабільне швидкісне передавання в міських умовах щільної забудови.

Мережі WIMAX використовують нижчевказані **типи доступу**:

- **fixed access** – фіксований доступ;
- **nomadic access** – сеансовий доступ (доступ з різних місць);
- **portable access** – доступ в режимі переміщення;
- **mobile access** – мобільний доступ, який може біти двох типів:
  - simple mobile access – спрощений мобільний доступ;
  - full mobile access – повноцінний мобільний доступ.

Фіксований доступ використовує діапазон частот 10–66 ГГц і тому передбачає, що станція (або інший пристрій) користувача постійно знаходиться в одному місці протягом всього часу дії контракту з оператором, який надає послуги. При цьому абонентська станція може підключатися до мережі або відключатися від неї, вибираючи будь-яку базову станцію для входу в мережу. Зазвичай станція реалізує зв'язок з одним сектором базової станції, а в разі його відмови мережа забезпечує підключення до іншого сектора. Такий метод доступу є альтернативою таким широкосмуговим проводовим технологіям, як xDSL, T1/E1 тощо, і дозволяє забезпечити пропускну спроможність 134 Мбіт/с.

Сеансовий доступ nomadic access передбачає, що станція знаходиться в одному місці протягом всього сеансу зв'язку. Між сесіями клієнтська станція може вільно переміщуватись в зоні дії цієї ж безпроводової мережі, а з'єднання з мережею буде встановлюватись за допомогою інших базових станцій WIMAX. Цей режим використовується, зазвичай, портативними пристроями, наприклад, ноутбуками.

Доступ в **режимі переміщення portable access** дозволяється переміщення користувача мережі (зі швидкістю до 40 км/год) забезпечуючи при цьому автоматичне переключення клієнтського модуля від однієї базової станції до іншої без втрати зв'язку станції з мережею. Однак такий режим не забезпечує всі можливості керування при переміщенні станції з одного сектора базової станції до іншого сектора іншої базової станції.

Використання режиму **мобільного доступу** дозволяє збільшити швидкість переміщення клієнтської станції до більше 120 км/год і забезпечує стійке з'єднання при різкій зміні напряму переміщення станції. При цьому **спрощений мобільний доступ simple mobile access** забезпечує неперервний сеанс зв'язку для застосувань, які не вимагають реалізації режиму реального часу, при переміщенні станцій в зоні покриття даної безпроводової мережі. Реалізація **повноцінного мобільного доступу full mobile access** забезпечує неперервний сеанс зв'язку для всіх застосувань при переміщенні станції з високою швидкістю в зоні покриття даної безпроводової мережі, при цьому гарантується передавання керування як між секторами, так і між базовими станціями.

Параметри чинних стандартів технології WIMAX наведено в таблиці 9.3.

Таблиця 9.3 – Параметри стандартів технології WIMAX

	802.16	**802.16a**	**802.16b**	**802.16e**	**802.16d**	**802.16-2004**	**802.16e-2005**	**802.16.1**	**802.16.2**	**802.16.2a**	**802.16.3**	**802.16m**
Рік прийняття	2001	2003	2003	2002	2004	2004	2005	2000	2001	2002	2003	2010
Частотний діапазон, ГГц	10–66	2–11	5–6	10–66	2–11	2–11	2–6	10–66	10–66	2–11	< 11	2,3–3,8
Пропускна спроможність, Мбіт/с	32–134 (для каналу 28 МГц)	до 75 (для каналу 20 МГц)	до 70 (для каналу 20 МГц)	32–134 (для каналу 28 МГц)	до 75 (для каналу 20 МГц)	до 75 (для каналу 20 МГц)	до 15 (для каналу 5 МГц)	до 134	134,4	144	54-175	30, 100/1Гбіт/с
Модуляція	QPSK, QAM16 QAM64	OFDM 256, QPSK, QAM16 QAM64	масшта-бована OFDMA	QPSK, QAM16 QAM64	OFDM 256, QPSK, QAM16 QAM64	OFDM, OFDMA з TDD і FDD	масштабована OFDMA	QAM16 QAM64 з TDD і FDD	QPSK, QAM64 з TDD і FDD	QPSK, QAM16 з TDD і FDD	OFDM, QPSK, QAM16 QAM64	OFDMA з TDD і FDD
Ширина каналу, МГц	20; 25; 28	1,5 та 20	10; 20	20; 25; 28	1,5 та 20	20	1,5 та 20	12,5–50	20; 25; 28	28	3,5; 7; 10,5 та 14	5; 10; 20
Тип доступу	Fixed	Fixed	Fixed/ Normadic	Fixed	Fixed/ Normadic	Fixed/ Normadic	Fixed/ Normadic and Full mobility /Portable	Fixed	Fixed	Fixed	Fixed	Fixed/Mobile /Normadic
Радіус дії, км	1–3	4–6, до 30	7–10, до 50	1–3	4–6, до 30	7–10, до 50	1–3	< 10	8–10	12	30–50	30–100

Стандарти WIMAX також використовують технологію MIMO. Наприклад, в мережах стандарту IEEE 802.16e технологія MIMO розглядається як необов'язкова опція в найпростішій конфігурації: 2×2 (антени передавача × антени приймача). В стандарті IEEE 802.16m технологія MIMO є обов'язковою з можливою конфігурацією 4×4 (в низхідному каналі допустимі конфігурації: 2×2, 4×2, 4×4, 8×2, 8×4, 8×8), що дозволяє збільшити швидкість передавання до 1 Гбіт/с.

В різних стандартах WIMAX використовується й різна техніка мультиплексування: ортогональне частотне мультиплексування OFDM (або мультиплексування з розподілом по ортогональних частотах) та множинний доступ з розподілом по ортогональних частотах OFDMA (або масштабований OFDMA – SOFDMA, Scalable OFDMA).

Головний принцип OFDM полягає в тому, що основний потік розділяється на деяку кількість паралельних підпотоків з низькою швидкістю передавання, а весь частотний діапазон каналу розбивається на деяку достатньо велику кількість піднесучих, для кожної з яких в загальному випадку може застосовуватись свій метод модуляції.

При використанні методу OFDMA сукупність піднесучих розбивається на N груп, кожна з яких має M піднесучих, з яких формується M підканалів: по одній піднесучій з групи. Використання масштабованості приводить до змінної структури підканалів, тобто до змінної кількості піднесучих, які об'єднуються в канал.

Метод OFDMA, порівняно з OFDM, дозволяє забезпечити більшу гнучкість при керуванні модулями користувача з різними типами антен, а також більш ефективне використання ресурсів.

Таким чином, мережі технології 802.16 дозволяють більш ефективно (порівняно з провідними технологіями) не тільки надавати доступ в мережу новим клієнтам, але й розширювати спектр послуг і охоплювати нові важкодоступні території. Крім того, безпроводові технології значно більш прості у використанні, ніж традиційні проводові канали. Мережа WIMAX, як і мережі Wi-Fi, достатньо прості в розгортанні і в разі необхідності легко масштабуються.

9.6 Технологія LTE

Технологія **LTE** (Long Term Evolution**)** є технологією побудови безпроводових мереж на базі IP-технологій, що створена міжнародною організацією **3GPP** (Third Generation Partnership Project), яка розробляє перспективні стандарти мобільного зв'язку. Дана технологія була затверджена як стандарт в 2008 році і характеризується високими швидкостями передавання даних: номінальна швидкість передавання в низхідному каналі (від базової станції до користувача) – 326,4 Мбіт/с, а у висхідному каналі

(від користувача до базової станції) – 172,8 Мбіт/с. Мережі на основі стандарту LTE можуть функціонувати по всій ширині спектра частот: від 700 МГц до 2,7 ГГц.

LTE базується на трьох основних технологіях:

- ортогональне частотне мультиплексування OFDM;
- багатоантенні системи MIMO;
- еволюційна системна архітектура мережі.

Основною перевагою мереж LTE є значно менші затримки при передаванні як даних користувачів, так і керівної інформації, оскільки маршрут передавання пролягає через меншу кількість проміжних вузлів.

Узагальнена структура мережі LTE наведена на рис. 9.12.

Рисунок 9.12 – Архітектура мережі LTE

Мережа LTE складається з двох важливих компонентів: мережі радіодоступу та базової мережі. Обмін даними в ***базовій мережі*** виконується за протоколом IP з комутацією пакетів, що суттєво відрізняється від мереж попередніх поколінь, в яких використовується комутація каналів між окремими модулями. В базову мережу входять блоки, які виконують керування, маршрутизацію, комутацію та зберігання даних різних типів.

Мережа радіодоступу забезпечує зв'язок терміналів користувачів з базовою мережею і складається з сукупності базових станцій, які, на відміну від всіх інших поколінь безпроводових мереж, можуть обмінюватись інформацією з використанням інтерфейсу X2 та виконувати функції керування. Тобто, базова станція мережі LTE виконує функції як передавача інформації, так і контролера. В мережі передається два види трафіку: дані користувача, передавання яких наведено на рис. 9.12 неперервною лінією, та сигнальна інформація, що відображена пунктирною лінією.

Базові станції в мережі LTE виконують розподілення радіоканалів, динамічне розподілення ресурсів у висхідному та низхідному каналах, маршрутизацію потоку даних користувача для передавання на обслуговувальний шлюз, шифрування даних тощо.

Основним керівним модулем в мережі LTE є **вузол керування мобільністю**, який виконує тільки функції керування і не обробляє дані користувача, але за допомогою протоколу сигналізації має безпосередній зв'язок з терміналами користувача і виконує аутентифікацію клієнтів, керування каналами, роумінг, сигналізацію між базовою мережею і терміналами користувачів тощо.

Обслуговувальний шлюз виконує обробку та маршрутизацію даних, які передаються з (або в) базової станції, буферизацію пакетів в низхідному напрямку, керує мобільністю між мережею LTE та мережами інших технологій 3GPP.

Пакетний шлюз реалізує з'єднання терміналів користувачів з зовнішніми мережами і забезпечує фільтрацію пакетів користувача, розподілення IP-адрес для терміналів користувача, виконує маркування пакетів транспортного рівня в низхідному напрямку.

Вузол контролю та моніторингу виконує дві основні функції: контролю шлюзу і контролю за якістю, реалізуючи для цього постійний моніторинг стану мережі та підтримку заданого рівня якості обслуговування QoS для всіх типів з'єднань.

Сервер абонентських даних призначено для зберігання даних про абонентів мережі: ідентифікаторів користувача, місця його знаходження на мережному рівні на випадок, якщо користувач, вийшовши з даної мережі, буде знаходитись в іншій, а йому прийшло повідомлення, дані безпеки абонентів тощо. В мережі LTE може бути декілька таких серверів, їх кількість залежить від структури мережі та кількості абонентів.

Потрібно відмітити, що архітектура LTE, яка також називається SAE (System Architecture Evolution), забезпечує гнучкість мережної конфігурації та високий рівень доступності сервісів. Мережа LTE може достатньо просто взаємодіяти з такими мережами попередніх поколінь, як GSM, CDMA, WCDMA, WIIMAX та іншими. В подальшому планується реалізувати технологію LTE не тільки в мобільних телефонах, але й у ноутбуках та інших портативних модулях.

В таблиці 9.4 наведено характеристики технології LTE порівняно з аналогічними параметрами мереж WIMAX (зауважимо, що вказані в стандартах теоретично можливі характеристики на практиці можуть бути меншими).

Таблиця 9.4 – Порівняльна характеристика LTE та WIMAX

Параметр	**LTE/3GPP Rel.8**	**WIMAX/IEEE 802.16e**
Швидкість передавання даних, Мбіт/с	326 у низхідному каналі, 172 у висхідному каналі	до 75
Підтримка мобільності, км/год	до 500	
Технологія мультиплексування	OFDMA у низхідному каналі, SL-FDMA у висхідному каналі	OFDMA
Спектр, МГц	1,25; 2,5; 5; 10; 15 та 20	
Радіус дії, км	5–100 з невеликим послабленням після 30 км	до 50
Заголовки/службова інформація	Порівняно невеликі заголовки	Достатньо великі заголовки
Схема MIMO, кодування	MIMO зі зворотним зв’язком	MIMO без зворотного зв’язку
Схема MIMO у висхідному каналі	2×2	2×2
Схема MIMO у низхідному каналі	Рознесений прийом	Рознесений прийом
Висота антени базової станції, м	25	25
Сервіс	Передавання даних	Передавання даних

Наступним етапом розвитку мереж 4G LTE є стандарт LTE-A (LTE-Advanced), який часто помилково називають 4G+. Саме ця технологія є технологією четвертого покоління. Головною її особливістю є використання MIMO і можливість агрегації частотних діапазонів каналів передавання даних, що дозволяє суттєво підвищити швидкість передавання, яка залежить від кількості об’єднаних каналів.

Порівняльна характеристика радіоінтерфейсів мереж LTE та LTE-Advanced наведена в таблиці 9.5.

Таблиця 9.5 – Порівняльна характеристика технологій LTE та LTE-A

Параметр	LTE	LTE-A
Пікова швидкість передавання Даних	У низхідному каналі 326 Мбіт/с (4×4 MIMO), 173 Мбіт/с (2×2 MIMO). У висхідному каналі (20 МГц): 86 Мбіт/с (1×2 MIMO).	При 8×8 MIMO, 20+20 МГц, 64QAM: Низхідний канал: 1,2 Гбіт/с. Висхідний канал: 568 Мбіт/с.
Підтримка мобільності, км/год	до 250–300	до 500
Технологія мультиплексування	OFDMA у низхідному каналі, SL-FDMA у висхідному каналі	
Модуляція	QPSK, 16QAM, 64QAM, 256QAM	
Агрегація спектра		До 4 частот смуг різної масштабованої ширини в різних смугах (низхідний, висхідний) (реліз Rel`13). Розробка агрегації 5 та 8 смуг (в перспективі – до 32 смуг).
Ширина каналу	Змінна (до 20 МГц)	Змінна (до 100 МГц)

9.7 Стандарти мереж WPAN, WMAN та WRAN

Безпроводові персональні (домашні) мережі **WPAN** на сьогодні створюються на основі технології Bluetooth, яка була розроблена компанією Ericsson. Метою стандарту IEEE 802.15, в якому описується дана технологія, є забезпечення можливості обміну даними різним електронним пристроям, які знаходяться на відстані до 100 метрів. Стандарт використовує радіочастоти 2400–2483,5 МГц, що належать до діапазону, який має назву **ISM** (Industrial, Scientific, Medicine) та використовується в багатьох країнах для безліцензійного доступу. Весь діапазон в технології розділено на 78 каналів шириною 1 МГц кожний. Забезпечується передавання як синхронних, так і асинхронних даних.

Для організації дуплексного зв’язку використовується метод часового мультиплексування, тобто в одному часовому слоті передається один модуль, а в іншому – інший модуль. При симетричній організації обміну асинхронними даними швидкість передавання складає 433,9 Кбіт/с в кожну сторону. Максимальна швидкість досягається при асиметричному обміні і складає 723,2 Кбіт/с в одну сторону і 57,5 Кбіт/с – в іншу.

Стандарт IEEE 802.15.3 є розвитком попереднього стандарту і забезпечує швидкість передавання даних до 55 Кбіт/с на відстань до 100 м. Такі характеристики можна реалізувати для кількості модулів в мережі до 245. Крім вказаної швидкості передавання забезпечується підтримка швидкостей 11, 22, 33 та 44 Кбіт/с. Для захисту даних використовується шифрування.

Мережі ZigBee стандарту IEEE 802.15.4 орієнтовано на використання в тому випадку, коли необхідно організувати зв'язок з різними пристроями: датчиками, сенсорами, модулями систем автоматизації виробництва, систем безпеки тощо. Стандарт забезпечує пропускну спроможність до 250 Кбіт/с на відстань до 10 м в діапазоні 2,4 ГГц.

Стандарт IEEE 802.15.4a визначає технологію широкосмугового радіозв'язку **UWB** (Ultra Wide Band), яка розроблена корпорацією Intel для передавання на швидкості до 500 Мбіт/с на відстань до 5 (іноді до 10) м. Для передавання використовують надкороткі радіоімпульси (менше 1 нс) в широкому діапазоні частот: від 3,1 до 10,6 ГГц.

Стандарт IEEE 802.20 описує системи мобільного безпроводового широкосмугового доступу **MBWA** (Mobile Broadband Wireless Access) для розробки міських і регіональних мереж (WMAN та WRAN), які функціонують в діапазонах частот не вище 3,5 ГГц. В даному стандарті визначається мережа персонального асиметричного широкосмугового доступу з комутацією пакетів, яка забезпечує передавання даних на основі протоколу IP в низхідному каналі 16–20 Мбіт/с, а у висхідному – 1–3,2 Мбіт/с. Головною особливістю таких мереж є те, що вони гарантують стабільний зв'язок для мобільних користувачів, які можуть переміщуватись зі швидкістю до 250 км/год, зокрема і не в зоні прямої видимості базової станції. Радіус дії – до 15 км. На сьогодні технологія MBWA використовується більш як в 30 країнах під назвою **HC-SDMA** (High Capacity Spatial Division Multiple Access), але часто використовується і назва iBurst, як в Японії.

Для створення безпроводових регіональних мереж **WRAN** розроблено стандарт IEEE 802.22, який забезпечує безпроводове широкосмугове передавання на відстань до 100 км і орієнтований на частотний діапазон від 30 до 3000 МГц, які стали вільними після широкого використання цифрового телебачення. Забезпечує стабільний зв'язок при переміщенні користувача зі швидкістю до 114 км/год і пропускну спроможність до 22 Мбіт/с.

Особливістю стандарту IEEE 802.22 WRAN є те, що він використовує технологію когнітивного радіопередавання, що забезпечує налагоджування параметрів модулів мережі таким чином, щоб передавання даних відбувалась тільки в ліцензованому діапазоні частот.

Основні характеристики мереж стандарту IEEE 802.22 наведені в таблиці 9.5.

Таблиця 9.5 – Характеристика мереж стандарту IEEE 802.22

Параметр	**Значення**
Пропускна спроможність	до 22 Мбіт/с
Ширина каналу	6 МГц
Технологія мультиплексування	OFDM, OFDMA у висхідному каналі
Модуляція	QPSK, QAM16, QAM64
Швидкість передавання	1,5 Мбіт/с у низхідному каналі, 384 Кбіт/с у висхідному каналі
Підтримка мобільності	до 114 км/год

9.8 Супутникові системи та мережі

За оцінками International Telecommunications Union, більш ніж половина населення Землі не має доступу до мережі, і тільки 10% поверхні планети забезпечено безпроводовим (мобільним) зв'язком. Тому використання супутникового зв'язку, який охоплює (залежно від того, на якій висоті знаходиться супутник) значну територію планети, має суттєве значення і розвивається останні роки швидкими темпами.

До складу будь-якої супутникової системи, узагальнена структура якої наведена на рис. 9.13, входять такі компоненти:

- **космічний сегмент**, який складається з декількох супутників-ретрансляторів;
- **наземний сегмент**, який містить центр управління системою, центр управління зв'язком, шлюзові та командно-вимірювальні станції;
- **користувацький (абонентський) сегмент**, який реалізує зв'язок за допомогою персональних супутникових терміналів;
- **наземні мережі зв'язку**, з якими через інтерфейс зв'язку з'єднуються шлюзові системи космічного зв'язку.

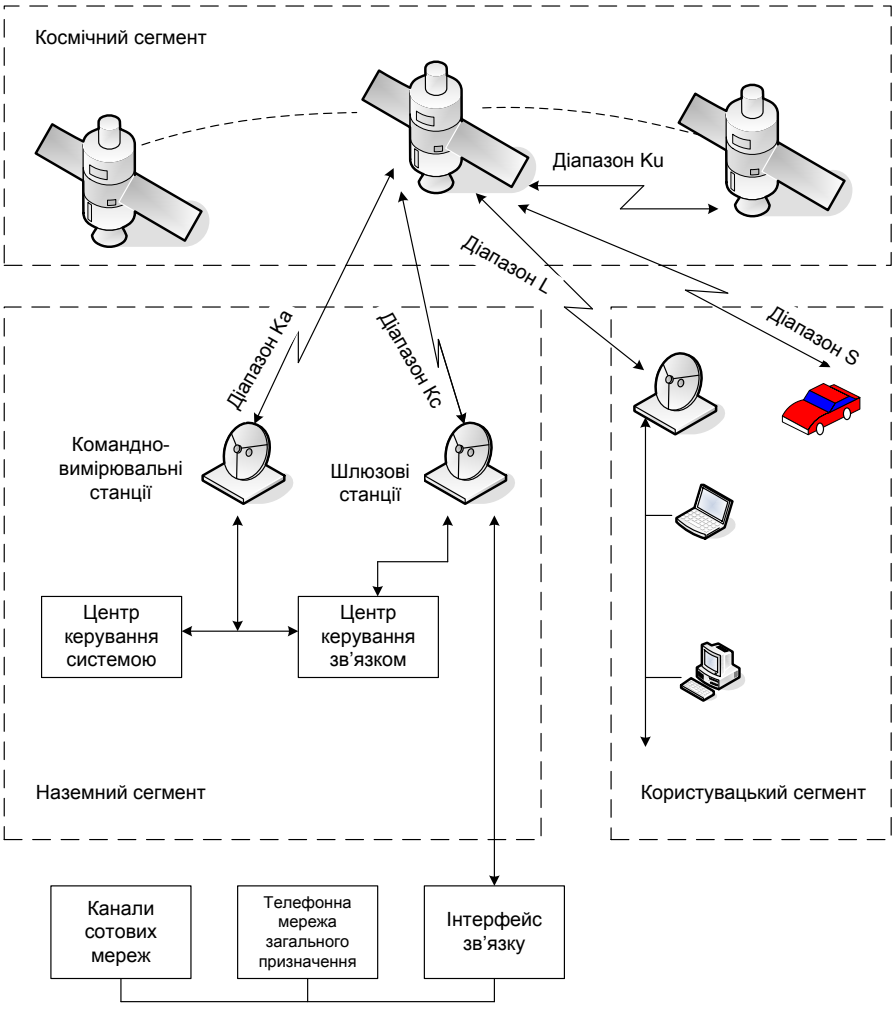


Рисунок 9.13 – Узагальнена структура супутникової системи

Для супутникових систем виділені смуги частот, які наведено в таблиці 9.6.

Таблиця 9.6 – Значення частот для супутникових систем

Діапазон	**Смуга частот, ГГц**
L	1, 452–1,500; 1,61–1,71
S	1,93–2,7
C	3,4–5,25; 5,725–7,075
Ku	10,70–12,75; 12,75–14,80
Ka	14,40–26,50; 27,00–50,20
Kc	84,00–86,00

Космічний сегмент складається з деякої кількості супутників-ретрансляторів, які об'єднані в групи. Орбіти супутників класифікуються за формою, нахилом до земної поверхні та періодичністю його проходження над точками земної поверхні. За формою для систем зв'язку розрізняють орбіти, форма яких наближена до кругової, а висоти апогея та перигея відрізняються на декілька десятків кілометрів, та еліптичні орбіти.

За класифікацією за величиною великої півосі, а точніше, висотою над поверхнею Землі, розрізняють такі типи супутників:

- низькоорбітальні LEO (Low Earth Orbit), які знаходяться на висоті від 160 до 2000 км і тому піддаються максимальному впливу гравітаційного поля Землі та її атмосфери;
- середньоорбітальні MEO (Medium Earth Orbit) супутники, які зазвичай, знаходяться на висоті від 2000 до 35786 км, тобто займають висоти між низькоорбітальними та геостаціонарними орбітами;
- геостаціонарні або геосинхронні GSO (Geosynchronous Orbit), що знаходяться на висоті 35786 км, і період їх обернення збігається з періодом обернення Землі навколо своєї осі;
- високоорбітальні HEO (High Earth Orbit) супутники знаходяться на висоті більше 35786 км.

Наземний сегмент – комплекс обладнання та систем, призначених для коректного функціонування всієї супутникової системи зв'язку. Центр керування системою виконує стеження за супутником, розрахунок його координат, корекцію часу (за необхідності), діагностику працездатності бортової апаратури, формування і передавання командної (службової) інформації тощо. Всі ці функції виконуються на основі телеметричної інформації, що надходить від кожного супутника орбітальної групи. Центр керування зв'язком забезпечує контроль за станом кожного супутника, контроль і керування орбітою окремих супутників, контроль і керування супутниками в нештатних ситуаціях тощо. При цьому передавання службової інформації виконується за допомогою рознесених територіально командно-вимірювальних систем (основних і резервних). Центр керування

зв’язком, крім контролю використання ресурсів супутника, виконує ще й аналіз та контроль зв’язку, реалізуючи це через шлюзові станції.

Шлюзова станція містить декілька приймально-передавальних комплексів, кожний з яких має свою антену. Використання декількох таких комплексів дозволяє практично без порушення зв’язку переходити послідовно від одного супутника до іншого.

Шлюзові станції зазвичай обробляють великі потоки інформації, задля чого до їх складу входять високошвидкісні комп’ютери, які містять банк даних персональних терміналів. Крім того, шлюзові станції містять комутаційне обладнання для з’єднання з різними наземними системами зв’язку. Основними задачами шлюзової станції є організація дуплексного з’єднання, передавання повідомлень і потоків даних великого розміру.

Користувацький сегмент складається з різних за своїми функціональними та технічними характеристиками систем (супутникові термінали та телефони, мобільні термінали для автотранспортних, авіаційних і морських систем тощо). В цьому забезпечується не тільки зв’язок між абонентами, які мають персональні супутникові термінали, а й їх зв’язок з абонентами інших мереж (сотова, телефонна тощо), визначається місцеположення абонентів.

Основні переваги супутникових систем полягають у:

- великій пропускній спроможності, яка обумовлена роботою супутників в широкому діапазоні гігагерцевих частот, що дозволяє підтримувати декілька тисяч каналів зв’язку одночасно;
- забезпеченні зв’язку між станціями, які знаходяться на дуже великих відстанях, і можливості обслуговування абонентів у важкодоступних точках;
- побудові мережі без комутаційних пристроїв, це обумовлено тим, що супутниковий зв’язок реалізується широкомовно;
- незалежності вартості передавання інформації від відстані між взаємодійними абонентами; вартість залежить від обсягу трафіку, що передається, та інтервалу передавання.

Недоліками супутникових систем зв’язку є:

- необхідність додаткових засобів та часу для забезпечення конфіденційності передавання даних, попередження можливості перехоплення даних іншими супутниками;
- затримка прийому радіосигналу наземною станцією, яка суттєво залежить від відстані між супутником і станцією, що призводить до складності реалізації канальних протоколів і необхідності контролю часу відповіді на запит;
- взаємне спотворення радіосигналів від наземних станцій, які працюють на суміжних частотах;
- вплив різних атмосферних явищ на передавання сигналів між наземними станціями і супутниками;

- виділення частот для коректного функціонування супутникових систем і розміщення супутників на орбітах вимагає координації дій та співробітництва багатьох країн, які використовують системи супутникового зв'язку.

Кількість супутників, які на сьогодні знаходяться на різних орбітах, коливається в достатньо широких інтервалах, однак вони призначені для вирішення різних класів задач. Для забезпечення глобального Інтернет-покриття створюється нова система зв'язку, яка зможе надавати доступ до широкосмугового (високошвидкісного) Інтернету у віддалених від комунікацій місцях. Для цього в 2019 році було виведено на орбіту майже 200 супутників, в до середини 2020-х років планується запуск ще 30000 супутників. Такий підхід забезпечить вирішення багатьох соціальних, економічних і освітніх задач майже в усіх точках Землі.

9.9 Питання для самоперевірки

1. Проаналізуйте можливості та характеристики поколінь безпроводового зв'язку.
2. Поясніть основні характеристики систем мобільного зв'язку покоління 3G.
3. Охарактеризуйте особливості безпроводових мереж та їх класифікацію.
4. Охарактеризуйте особливості систем покоління 4G.
5. Проаналізуйте особливості систем покоління 5G.
6. Поясніть особливості застосування технології розширеного спектра сигналу та її типи.
7. Поясніть особливості використання методу стрібкоподібної зміни частоти.
8. Поясніть особливості прямого послідовного розширення спектра.
9. Охарактеризуйте поняття розширювальної послідовності, коефіцієнта розширення, чипа та чипової швидкості.
10. Подайте підсумкову послідовність при передаванні даних 01101 та використанні інверсної послідовності коду Баркера.
11. Подайте підсумкову послідовність при передаванні даних 10001 та використанні прямої послідовності коду Баркера.
12. Поясніть принцип функціонування ортогонального частотного мультиплексування.
13. Охарактеризуйте особливості та стек протоколів безпроводових мереж стандарту IEEE 802.11.
14. Проаналізуйте основні топології безпроводових локальних мереж стандарту IEEE 802.11.
15. Проаналізуйте режими доступу до середовища передавання даних в безпроводових локальних мережах.
16. Поясніть процедуру передавання даних в мережах IEEE 802.11 з ви-

користанням розподіленого режиму доступу.
17. Поясніть особливості реалізації розширення спектра методом стрибкоподібної зміни частоти в мережах стандарту IEEE 802.11.
18. Поясніть особливості реалізації методу прямого послідовного розширення спектра в мережах стандарту IEEE 802.11.
19. Поясніть особливості використання ортогонального частотного мультиплексування в мережах стандарту IEEE 802.11.
20. Охарактеризуйте принцип функціонування та особливості використання технології MIMO в безпроводових локальних мережах.
21. Охарактеризуйте особливості структурної організації Mesh-мереж.
22. Поясність характеристики та особливості Mesh-мереж.
23. Охарактеризуйте особливості та сфери використання мереж WiGig.
24. Охарактеризуйте компоненти мереж WIMAX, їх призначення та особливості функціонування.
25. Охарактеризуйте особливості типів доступу, реалізованих в мережах WIIMAX.
26. Поясніть спільні риси та відмінності методів OFDM та OFDMA.
27. Які модифікації методу OFDM Вам відомі і в чому їх особливості?
28. Проаналізуйте характеристики стандартів IEEE 802.16.
29. Наведіть основні характеристики та особливості функціонування мереж LTE.
30. Проаналізуйте структурну організацію мереж LTE та функції її складових модулів.
31. Порівняйте характеристики мереж WiMAX та LTE.
32. Поясніть основні відмінності мереж LTE-A.
33. Охарактеризуйте особливості та можливості стандартів безпроводових персональних мереж.
34. Охарактеризуйте призначення, особливості та можливості стандарту IEEE 802.20.
35. Охарактеризуйте призначення, особливості та можливості стандарту IEEE 802.22.
36. Поясніть особливості супутникових систем та мереж.
37. Охарактеризуйте структуру супутникової системи.
38. Які типи супутникових систем Вам відомі?
39. Поясніть переваги супутникових систем.
40. Проаналізуйте недоліки супутникових систем.

СПИСОК ЛІТЕРАТУРИ

1. Азаров О. Д. Комп’ютерні мережі / О. Д. Азаров, С. М. Захарченко, О. В. Кадук та ін. – Вінниця : ВНТУ, 2013. – 370 с. ISBN 978-966-641-543-4
2. Джеймс Мартин. Архитектура и реализация АТМ / Джеймс Мартин, Кэтлин Кэвен Чапмен, Джо Лубен. – М. : Издательство «ЛОРИ», 2000. – 214 с. ISBN 5-85582-068-8
3. Бірюков М. Л. Транспортні мережі телекомунікацій / М. Л. Бірюков, В. К. Стеклов, Б. Я. Костік. – К. : Техніка, 2005. – 312 с.
4. Буров Є. Комп'ютерні мережі / Буров Є. — Львів : Магнолія, 2006. — 262 с. — ISBN 966-8340-69-8
5. Зайченко О. Ю. Комп’ютерні мережі / О. Ю. Зайченко, Ю. П. Зайченко. – К. : Видавничий Дім «Слово», 2010. – 520 с. ISBN 978-966-194-050-4
6. Дикер П. Сети АТМ корпорации CISCO / Дикер П. – М. : Издательский дом «Кильямс», 2004. – 880 с. ISBN 5-8459-0632-6
7. Камер Д. Є. Сети TCP/IP, том 1. Принципы, протоколы и структура / Камер Д. Є. – М. : Издательский дом «Вильямс», 2003. – 880 с. ISBN 5-8459-0419-6
8. Камер Д. Є. Сети TCP/IP, том 3. Разработка приложений типа клиент/сервер для Linux/POSIX / Д. Є. Камер, Д. Л. Стивенс. – М. : Издательский дом «Вильямс», 2003. – 592 с.
9. Карташевский В. Г. Сети подвижной связи / В. Г. Карташевский, С. Н. Семенов, Т. В. Фирстов. – М. : Эко-Трендз, 2001. – 299 с. ISBN 5-88405-028-3
10. Кулаков Ю. О. Комп’ютерні мережі / Ю. О. Кулаков, Г. М. Луцький. – К. : Юніор, 2005. – 397 с. ISBN 966-7323-27-7
11. Кулаков Ю. О. Комп’ютерні мережі / Ю. О. Кулаков, І. А. Жуков. – К. : Вид-во Нац. авіац. ун-ту «НАУ-друк», 2009. – 392 с.
12. Куроуз Дж. Компьютерные сети. Многоуровневая архитектура Интернета / Дж. Куроуз, К. Росс. – СПб. : Питер, 2016. – 912 с. ISBN 5-8046-0093-1.
13. Кульгин М. Компьютерные сети, практика построения. – СПб. : Питер. 2003. – 462 с. ISBN 5-94723-563-3.
14. Лаем Куин. Fast Ethernet / Лаем Куин, Ричард Рассел. – СПб. : Питер, 2008. – 788 с.
15. Назаров А. Н. АТМ : Технические решения создания сетей / А. Н. Назаров, И. А. Разживин, М. В. Симонов. – М. : Радио и связь, 2002. – 406 с. ISBN 978-5-93517-079-0
16. Ногл М. TCP/IP. Иллюстрированный учебник / М. Ногл. – М. : ДМК Пресс, 2001. – 480 с. ISBN 5-94074-044-8.

17. Оглтри Т. Модернизация и ремонт сетей / Т. Оглтри. – М. : Издательский дом «Вильяис», 2005 – 1328с. ISBN 5-8459-0688-1.

18. Одом Уэндел. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-105: маршрутизация и коммутация, академическое издание. / О. Уэндел – СПб. : ООО «Диалектика», 2018. – 1008 с. ISBN 978-5-9909446-5-7.

19. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : [учебник для ВУЗов] / В. Г. Олифер, Н. А. Олифер. – СПб. : Питер, 2018. – 992 с. ISBN 978-5-496-01967-5.

20. Палмер М. Проектирование и внедрение компьютерных сетей. Учебный курс. / М. Палмер, Р. Б. Синклер. – СПб. : БХВ, 2004. – 752 с. ISBN 5-94157-374-X.

21. Рошан П. Основы построения беспроводных локальних сетей стандарта 802.11 / Рошан П., Лиэри Дж. – М. : Издательский дом «Вильямс», 2004. – 304 с. ISBN 5-8459-0701-2.

22. Семенов Ю. А. Протоколы Интернет / Ю. А. Семенов. – М. : Горячая линия-Телеком, 2005. – 1100 с. ISBN 5-93517-044-2.

23. Степунин А. Н. Мобильная связь на пути к 6G. в 2-х т. / А. Н. Степунин, А. Д. Николаев. – 2-е изд. – Москва-Вологда : Инфра-Инженерия, 2018. – 796 с. ISBN 978-5-9729-0192-0

24. Стивенс У. Р. Протоколы TCP/IP. Практическое руководство / У. Р. Стивенс. – СПб. : «Невский проспект» – «БХВ-Петербург», 2003. – 672 с. ISBN 5-7940-0093-7.

25. Столлингс В. Беспроводные линии связи и сети / В. Столлингс. – М. : Издательский дом «Вильямс», 2003. – 640 с. ISBN 5-8459-0409-9.

26. Столлингс В. Компьютерные системы передачи данных / В. Столлингс. – М. : Издательский дом «Вильямс», 2002. – 928 с. ISBN 5-8459-0311-4.

27. Столлингс В. Передача данных / В. Столлингс. – СПб. : Питер, 2004. – 750 с. ISBN 5-94723-647-8.

28. Столлингс В. Современные компьютерные сети / В.Столлингс. – СПб. : Питер, 2003. – 783 с. ISBN 5-947233-27-4.

29. Таненбаум Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – СПб. : Питер, 2015. – 960 с. ISBN 5-318-00492-X.

30. Титтель Э. ISDN просто и доступно / Эд Титтель, Стив Джейс, Дэвид Пискителло, Лайза Пфайфер. – М. : Издательство «ЛОРИ», 1999. – 282 с. ISBN: 0-12-691412-5.

31. Тихвинский В. О. Сети мобильной связи LTE : технологии и архитектура / В. О.Тихвинский, С. В.Терентьев, О. Б. Юрчук. – М. : Эко-Трендз, 2010. – 284 с. ISBN 978-98595-032-8.

32. Уолтон Ш. Создание сетевых приложений в среде Linux. Руководство разработчика / Шон Уолтон. – М. : СПб. – Киев : «Вильямс». 2001. – 464 с. ISBN 5-8459-0193-6.

33. Хелд Г. Технологии передачи данных / Г. Хелд. – К. : Спб. : Питер, БХВ-Петербург, 2003. – 720 с. ISBN 5-94723-472-6 .

34. Шмалько А. В. Цифровые сети связи : основы планирования и построения / А. В. Шмалько. – М. : Эко-Трендз, 2001. – 282 с. ISBN 5-88405-032-1.

35. Филимонов А. Ю. Построение мультисервисных сетей Ethernet / А. Ю. Филимонов. – СПб. : БХВ-Петербург, 2007. – 592 с. ISBN 978-5-9775-0007-4.

36. Stallings William. Data and Computer Communication / William Stallings. – 1999. – 810 p. ISBN-10: 0130843709.

37. Stallings William. Computer Networking with Internet Protocols and Technology / William Stallings. – 2004. – 640 p. ISBN 10 9780131410985.

38. Thurwachter Jr. Data and telecommunication : systems and applications / Jr. Thurwachter, N. Charles. – 2000. – 630 p. ISBN 10: 0137939108.

ДОДАТКИ

Додаток А
Стандартизація мереж

На сьогодні існує ціла низка міжнародних і національних організацій, які займаються розробкою стандартів, рекомендацій та архітектур для комп'ютерних мереж і систем комунікацій. Залежно від статусу організацій розрізняють такі види стандартів:

- стандарти окремих компаній та фірм (наприклад, стек протоколу DECnet, фірми Digital Equipment тощо);
- стандарти спеціальних комітетів, об'єднань та форумів, які створюються декількома фірмами, наприклад, стандарти технології ATM, розроблені спеціально створеним об'єднанням ATM Форум, стандарти союзу Fast Ethernet та інші;
- національні стандарти, наприклад, мережа FDDI, яка розроблена Американським національним інститутом стандартизації ANSI;
- міжнародні стандарти, наприклад, модель і стек комунікаційних протоколів Міжнародної організації стандартизації ISO, численні стандарти Міжнародного союзу електрозв'язку MCE, зокрема стандарти на мережі з комутацією пакетів X.25, мережі Frame Relay, ISDN, модеми і багато інших.

Деякі стандарти, розвиваючись, можуть переходити з однієї категорії в іншу.

Розглянемо найбільш важливі міжнародні та національні організації.

Міжнародна організація зі стандартизації **ISO** (International Organization for Standardization), яка на сьогодні об`єднує 163 держави, створена в 1946 році. Сфера діяльності ISO стосується стандартизації в усіх галузях, крім електротехніки та електроніки, які відносять до компетенції Міжнародної електротехнічної комісії (IEC – International Electrotechnical Commission). Крім стандартизації ця організація займається проблемами сертифікації. ISO є домінуючою організацією зі стандартизації в галузі інформаційних технологій і розробила та затвердила безліч стандартів, зокрема для мережних технологій. Документи, які прийнято ISO, мають статус міжнародного стандарту і позначаються номером, наприклад, ISO 10026. Цій організації належить розробка еталонної моделі взаємодії відкритих систем OSI (Open System Interconnection) – абстрактної мережної моделі для комунікацій і розробки мережних протоколів, яка подає мережу як сукупність рівнів, кожний з яких визначає і обслуговує свою частину взаємодії кінцевих станцій та передавання даних через мережу.

Міжнародний союз телекомунікацій (електрозв`язку) **ITU** (International Telecommunication Union) – провідна організація з розробки стандартів та рекомендацій для телефонних і телекомунікаційних служб, яка також регулює питання міжнародного використання радіочастот (розподілення радіочастот за призначенням і для окремих держав). В ITU, який є органом ООН, на сьогодні входить майже 200 держав і більше 700 представників

різних асоціацій та бізнес-структур. До грудня 1992 року мав назву Міжнародного консультативного комітету з телефонії та телеграфії (МККТТ, CCITT – Consultative Committee for International Telephone and Telegraph). Стандарти (згідно з термінологією ITU – рекомендації) не є обов'язковими, але широко підтримуються, оскільки полегшують взаємодію між мережами зв'язку і дозволяють провайдерам надавати послуги в усьому світі.

ITU має 3 комітети (сектори), які виконують основну роботу даного союзу:

- **Telecommunication Standardization Sector (ITU-T)** – сектор стандартизації телекомунікації (електрозв'язку) – складається з 14 дослідницьких груп за окремими напрямами роботи й вивчає та досліджує технічні та робочі питання, виконує розробку та адаптацію більш ефективних методів роботи і взаємодії користувачів телекомунікаційних мереж;
- **Radiocommunication Sector (ITU-R)** – сектор радіозв'язку – займається розподіленням частотного спектра та орбіт супутникового зв'язку;
- **Telecommunication Development Sector (ITU-D)** – сектор розвитку електрозв'язку – створений для забезпечення доступу до найбільш вагомих розробок в галузі телекомунікацій та інформаційної інфраструктури.

Кожні 4 роки розробляються та приймаються нові стандарти, оновлюються старі рекомендації, створюються нові та ліквідуються старі дослідницькі групи. Найбільш відомим стандартом, який розроблено комітетом CCITT, є стандарт мережі X.25.

Європейська асоціація виробників комп'ютерів **ECMA** (European Computer Manufacturers Association) – некомерційна організація, яка була створена групою європейських компаній, але пізніше, завдяки входженню до її складу представників таких компаній, як IBM, Digital, AT&T, British Telecom і Toshiba, стала міжнародною організацією. І хоча ECMA розробляє стандарти інформаційних технологій для Європи, вони часто передаються в ISO для затвердження їх як міжнародних.

Європейський інститут зі стандартизації у сфері телекомунікацій **ETSI** (European Telecommunications Standards Institute) – незалежна некомерційна організація, яка реалізує стандартизацію в телекомунікаційній промисловості в Європі. Цією організацією були стандартизовані: система сотового зв'язку GSM та система професійного мобільного радіозв'язку TETRA. Крім того є одним з розробників системи 3GPP.

Інститут інженерів з електротехніки та електроніки **IEEE** (Institute of Electrical and Electronics Engineers) – міжнародна некомерційна асоціація професіоналів, членами якої є окремі інженери та спеціалісти, а не компанії. Створена у 1884 році і на сьогодні об'єднує майже 400 тисяч індивідуальних членів більш ніж зі 170 країн. IEEE видає більше 100 наукових журналів, 40 журналів для спеціалістів. Головна мета IEEE – розвиток науко-

вої діяльності в комп'ютерній техніці, інформатиці, телекомунікації, електроніці та електротехніці, інформаційна підтримка спеціалістів з цих напрямків. Комітет IEEE 802 (Computer Society Local Network Committee, Project 802) спеціалізується на питаннях, пов'язаних з локальними мережами, розробив і випустив цілу низку стандартів (IEEE 802.x), які в подальшому були прийняті і опубліковані ISO як міжнародні стандарти (ISO 8802.x). На сьогодні в групу стандартів IEEE 802.x входять не тільки стандарти для локальних мереж, а й ті, що присвячені іншим питанням, наприклад, безпеці, кабельним модемам тощо.

Американський національний інститут стандартів **ANSI** (American National Standards Institute) – некомерційна неурядова організація, яка розробляє та публікує стандарти для промисловості країни. Інформаційними технологіями займаються комітети:

- **JTC1 TAG** – технічна консультативна група **(Technical Advisory Group)**, яка представляє позицію США щодо стандартів в ISO;
- **ASC X.3**, який розробляє 90% стандартів США у сфері інформаційних технологій; підкомітет X.3 відповідає за стандартизацію технології **FDDI** (Fiber Distributed Digital Interface);
- **ASC T.1** – добровільний орган стандартизації для телекомунікаційної галузі США, який розробляє національні телекомунікаційні стандарти;
- **ASC X.12** – група відповідає за стандарти, які відносять до електронного обміну даними (EDI – Electronic Data Interchange) на території США.

Національні організації стандартизації інших країн:

Франція – Французька асоціація зі стандартизації AFNOR (Association Francaise Normalisation);

Великобританія – Британський інститут стандартів BSI (Britain Standard Institute);

Німеччина – Німецький інститут стандартів DIN (Deutsches Institut fur Normung e.V.);

Канада – Канадська асоціація стандартизації CSA (Canadian Standards Association);

Японія – Японський комітет промислових (галузевих) стандартів JISC (Japanese Industrial Standards Committee).

Найбільш відомим стандартом з мережних комунікацій є технологія FDDI.

Асоціація електронної промисловості **EIA** (Electronic Industries Alliance) – національна комерційна асоціація США, яка представляє американських виробників електронного обладнання в різноманітних організаціях зі стандартизації. EIA розроблено та опубліковано цілу низку стандартів, що стосуються фізичних комунікаційних інтерфейсів, електричних сигналів, кабельної системи, а також описують різні способи з'єднання

комп'ютерів з іншими електронними пристроями, наприклад, стандарти RS-232 (Recommended Standard 232), RS-422, RS-449.

Співтовариство Internet **ISOC** (Internet Society) – міжнародна професійна освітня організація, основною метою якої є забезпечення відкритого розвитку, еволюції та використання Internet як глобальної комунікаційної інфраструктури в усьому світі. Під керуванням ISOC працює організація **IAB** (Internet Activities Board, а з 1992 року – Internet Architecture Board), яка займається розробкою та розглядом стандартів і напрямків розвитку мережі Internet, а також її адмініструванням. IAB має два підкомітети, у кожного з яких є свій виконавчий комітет:

- науково-дослідницький – **IRTF** (Internet Research Task Force);
- інженерний (законодавчий) – **IETF** (Internet Engineering Task Force).

IRTF – робочий підкомітет, який займається достроковими дослідницькими проектами, тобто вирішенням науково-дослідницьких проблем. Виконавчий комітет **IRSG** (Internet Research Steering Group) займається вивченням проблем Internet науково-дослідницького характеру.

IETF – основна робоча структура Internet, яка відповідає за вирішення інженерних задач і за розробку стандартів для мережі та приймає документи **RFC (Request for Comments**). Виконавчий комітет **IESG Internet Engineering Steering Group** призначений для вивчення інженерних проблем Internet.

Кожний з підкомітетів має певну кількість робочих груп, які є мобільними структурами і створюються для вирішення конкретної інженерної задачі. В IETF існує певна практика прийняття проекту RFC, що базується на необхідності розгляду декількох незалежних реалізацій запропонованого стандарту.

Всі прийняті IETF стандарти RFC (а також інші матеріали, що заслуговують уваги) доступні усередині Internet через електронну пошту, файлові сервери тощо. Деякі з документів RFC, запропоновані IAB, прийняті як стандарти Internet. До них належать документи, в яких описані протоколи TCP/IP, SNMP тощо.

Адміністративна група мережі Internet **IANA** (Internet Assigned Numbers Authority) займається розподіленням адрес мереж, атрибутів тощо, а також виконує контроль за унікальністю адрес та ідентифікаторів.

В Internet також існує організація, яка відповідає за поширення технічної інформації про служби мережі, реєстрацію та підключення користувачів до мережі, призначення IP-адрес і доменних імен, а також підтримку бази даних RFC. Ця організація називається Центром мережної інформації **NIC** (Network Information Center). Спочатку це був єдиний центр, на сьогодні існує багато таких центрів на рівні локальних, регіональних і національних мереж.

ATM Forum – консорціум компаній-виробників комунікаційного обладнання для мереж ATM. Формально не є організацією зі стандартизації. Створено спеціально для розробки та стандартизації обладнання й розроб-

ки протоколів для мереж ATM. Документи, що випускаються цією організацією, називаються угодами з реалізації.

Frame Relay Forum – організація, яка об'єднує виробників комунікаційного обладнання для мереж Frame Relay.

MPLS Forum – консорціум, який об'єднує організації, що займаються розробкою принципів побудови та протоколів мереж MPLS, організації віртуальних мереж на основі MPLS.

У квітні 2005 року три організації – ATM Forum, Frame Relay Forum та MPLS Forum – об'єднались в один **MFA Forum** (MPLS–Frame Relay–ATM Forum), який з 2007 року називається **IP/MPLS Forum**. У квітні 2009 року IP/MPLS Forum ввійшов до консорціуму **Broadband Forum (BBF)**, який існує з 1994 р.

Додаток Б

Таблиця Б.1 – Стандартизовані типи модуляції

Абревіатура	Type of signal modulation	Тип модуляції
π/4 QPSK	Quaternary Phase Shift Keying	π/4 четвірково-фазова маніпуляція
ADM	Adaptive Delta Modulation	адаптивна дельта-модуляція
AFM	Amplitude-Frequency Modulation	амплітудно-частотна модуляція
APM	Amplitude Phase Modulation	амплітудно-фазова модуляція
BFSK	Binary Frequency Shift Keying	двійкова частотна маніпуляція
BPSK	Binary Phase Shift Keying	відносна фазова маніпуляція
CAP	Carrierless AM-PM	амплітудно-фазова модуляція без несучої
CDM	Companded Delta Modulation	комплементна дельта-модуляція
DFSK	Double Frequency Shift Keying	двійкова частотна маніпуляція
DM	Delta Modulation	дельта- модуляція
DPM	Differential Phase Modulation	диференціальна фазова модуляція
FM	Frequency Modulation	частотна модуляція
FM-PM	Frequency Modulation-Phase Modulation	частотно-фазова модуляція
FSK	Frequency Shift Keying	частотна маніпуляція
MFSK	Multiple or Multilevel FSK	багатократна або багаторівнева частотна маніпуляція
PAM	Phase Amplitude Modulation, Pulse-Amplitude Modulation	амплітудно-фазова модуляція, амплітудно-імпульсна модуляція AIM
PM	Phase Modulation	фазова модуляція
PSK	Phase Shift Keying	фазова маніпуляція
QAM N N=4 (16, 32, 64, 128)	Quadrature Amplitude Modulation	квадратично-амплітудна модуляція
QPSK	Quadrature Phase Shift Keying	квадратично-фазова маніпуляція
QPSK	Quaternary Phase Shift Keying	Четвірково-фазова маніпуляція

Таблиця Б.2 – Стандартні протоколи модемів

Рекомендація	Швидкість передавання біт/с	Режим передавання	Дуплекс/ напівдуплекс	Модуляція	Тип лінії
V.17(fax)	14400,9600, 7200, 1200	синхронний	дуплекс	СКК128. 64, 32, 16	комутована
V.21	300	синхронний/ асинхронний	дуплекс	FM	комутована, виділена
V.22	1200,6	синхронний/ асинхронний	напівдуплекс	QPSK, BPSK	комутована, виділена
V.22bis	2400,12	синхронний/ асинхронний	дуплекс	QAM I 6, QAM 4	комутована
V.23	1200,6	синхронний/ асинхронний	дуплекс	ЧМ	комутована
V.26	2400	синхронний	дуплекс	QPSK	виділена
V.26bis	2400,12	синхронний	дуплекс	QPSK, BPSK	комутована
V.26ter	2400,12	синхронний/ асинхронний	дуплекс	QPSK, BPSK	комутована
V.27(fax)	4800	синхронний	будь-який		вид.
V.27bis(fax)	4800, 2400	синхронний	будь-який	BPSK, QPSK	вид.
V.27ter(fax)	4800, 2400	синхронний	дуплекс	ОФМ8, QPSK	комутована
V.29(fax)	9600, 7200, 4800	синхронний	будь-який	QAM I 6, QAM 4	виділена.
V.32	9600, 4800, 2400	синхронний/ асинхронний	дуплекс	СКК32. 16. QAM 4, BPSK	комутована
V.32bis	14400, 12000, 9600. 7200, 4800	синхронний	дуплекс	СКК128, 64.32. 16	комутована
V.32ter	19200, 16800	синхронний	дуплекс	СКК 256, 512	комутована
V.33	14400, 12000	синхронний	дуплекс	СКК128, 64	виділена
V.34	28800, 26400, 24000,21600, 19200, 16800. 14400, 12000. 9600, 7200. 4800, 2400	синхронний	дуплекс	багатовимірні СКК	комутована, виділена
V.34bis (V.34+)	33600	синхронний	дуплекс	багатовимірні СКК	комутована, виділена
V.90	56000 (пр.напр), 33600 (зв.напр.)	асинхронний	дуплекс		
V.92	56000 (пр.напр), 48600 (зв.напр.)	асинхронний	дуплекс		
Belll03j	300	синхронний/ асинхронний	дуплекс	FM	комутована
Be11202	1200	синхронний/ синхронний	дуплекс	FM	комутована виділена

BeI1208	4800		дуплекс		комутована
Bell212a	1200		дуплекс		комутована
HST	300, 450/4800, 7200, 9600, 12000, 14400, 16800	Синхронний	асиметричний дуплекс		комутована
ZyX	7200, 9600, 12000, 14400, 16800, 19200	синхронний.	дуплекс	СКК,256	комутована
PEP	19600	синхронний	дуплекс	511×СКК 64	комутована

СКК – сигнально-кодова конструкція;

FM – частотна модуляція;

BPSK – відносна фазова модуляція;

QPSK – квадратично-фазова модуляція;

QAM – квадратично-амплітудна модуляція;

bis та ter означають, відповідно, другу та третю модифікації протоколів.

Додаток В

Таблиця В.1 – Перелік робочих груп IEEE 802.x

Назва	Опис
IEEE 802.1	Об’єднання мереж, керування мережними пристроями та їх взаємодія.
IEEE 802.2	Керування логічним передаванням даних (Logical Link Control, LLC).
IEEE 802.3	Технологія Ethernet з методом доступу CSMA/CD.
IEEE 802.4	Локальна мережа з методом доступу «маркерна шина» (Token Bus).
IEEE 802.5	Локальна мережа з методом доступу «маркерне кільце» (Token Ring).
IEEE 802.6	Мережі мегаполісів (MAN).
IEEE 802.7	Технічна консультативна група широкосмугового передавання по коаксіальному кабелю.
IEEE 802.8	Технічна консультативна група з оптоволоконних мереж.
IEEE 802.9	Інтегровані мережі передавання голосу і даних.
IEEE 802.10	Мережна безпека.
IEEE 802.11	Безпроводові локальні мережі.
IEEE 802.12	Локальні мережі з доступом на вимогу з пріоритетами.
IEEE 802.13	Офіційно не використовується.
IEEE 802.14	Кабельні модеми.
IEEE 802.15	Безпроводові персональні мережі (Wireless Area Personal Network, WPAN): • IEEE 802.15.1 – мережі Bluetooth; • взаємодія стандартів IEEE 802.15 і IEEE 802.11; • IEEE 802.15.3 – мережі High-Rate WPAN; • IEEE 802.15.4 – мережі ZigBee (Low Rate Wireless Personal Area Network); • IEEE 802.15.5 – технологія Mesh networking для WPAN.
IEEE 802.16	Безпроводова міська мережа WIMAX.
IEEE 802.17	Еластичне кільце пакетів.
IEEE 802.18	Технічна консультативна група з радіорегулювання.
IEEE 802.19	Технічна консультативна група з взаємодії мереж.
IEEE 802.20	Мобільний широкосмуговий безпроводовий доступ.
IEEE 802.21	Технологія Media Independent Handoff.
IEEE 802.22	Регіональні безпроводові мережі (Wireless Regional Area Network).
IEEE 802.23	Робоча група надзвичайних сервісів.
IEEE 802.24	Smart Grid TAG. Інтелектуальніі мережі.
IEEE 802.25	Omni-Range Area Network. Інтерфейс мобільного зв’язку

Таблиця В.2 – Поточний список стандартів IEEE 802.1

Назва	Опис
IEEE 802.1b	Керування локальними/регіональними мережами (LAN/MAN).
IEEE 802.1D	Об’єднання локальних мереж за допомогою MAC мостів (містить стандарти 802.1p, 802.12e, 802.1j, 802.6k, 802.1t і 802.1w).
IEEE 802.1e	Стандарт на протоколи системного навантаження (System Load Protocol) для локальних и регіональних мереж.
IEEE 802.1f	Інформація про загальні визначення та процедури керування IEEE 802.
IEEE 802.1G	Віддалені MAC-мости.
IEEE 802.1H	Правила організації MAC-мостів в мережах Ethernet.
IEEE 802.1p	Доповнення до логіки MAC-мостів LAN та MAN для забезпечення пріоритезації трафіку та динамічної багатоадресної фільтрації.
IEEE 802.1Q	Віртуальні мережі (VLAN).
IEEE 802.1r	Передавання нестандартних атрибутів за допомогою GARP-протоколу.
IEEE 802.1s	Multiple Spanning Trees. Застосування алгоритму Spanning Trees (STP) для VLAN (внесено в 802.1Q).
IEEE 802.1v	Класифікація VLAN за протоколами і портами (внесено в 802.1Q).
IEEE 802.1w	Протокол RSTP (Rapid Spanning Tree Protocol) на заміну STP (внесено в 802.1Q).
IEEE 802.1X	Контроль доступу та аутентифікації на основі порту, що обмежує права неавторизованих станцій, підключених до комутатора.
IEEE 802.1AB	Протокол LLDP – незалежний протокол для ідентифікації та передавання пристроями їх параметрів і можливостей.
IEEE 802.1ad	Використання VLAN понад існуючої VLAN (доповнення до 802.1Q).
IEEE 802.1AE	Безпека MAC (MACSec) MACsec дозволяє ідентифікувати неавторизовані підключення до LAN і вилучати їх з комунікації в мережі.
IEEE 802.1af	Media Access Control (MAC) Key Security.
IEEE 802.1ag	Керування помилками з’єднання (Connectivity Fault Management). Цей стандарт полегшить виявлення і перевірку маршрутів через мости та LAN 802.1.
IEEE 802.1ah	Мости опорних провайдерських мереж (PBB – Provider Backbone Bridge).
IEEE 802.1aj	Двопортові пристрої (TPMR – Two Port MAC Relay), які є більш простими, ніж VLAN-мости.
IEEE 802.1ak	Протокол Multiple Registration Protocol (MRP), розрахований на великі мережі для збільшення їх швидкості та пропускної спроможності.

IEEE 802.1ap	Визначення Management Information Base (MIB) для VLAN-мостів.
IEEE 802.1aq	Стандарт визначає алгоритми обчислення найкоротшого маршруту і підтримку VLAN за допомогою ідентифікаторів VLAN (VID), які прив'язані до топології мережі (SPB – Shortest Path Bridging – для невеликих VLAN і SPBB – Shortest Path Backbone Bridging – для великих PBB).
IEEE 802.1AR	Secure Device Identity (DevID) – визначає унікальні ідентифікатори модуля, а також керування і криптографічні прив'язки станції до її ідентифікаторів.
IEEE 802.1AS	Визначає процедури і протокол синхронізації для аудіо- та відеозастосувань і реконфігурування мережі при відмові її елементів (Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks).
IEEE 802.1Qat	Розширення протоколу Ethernet для резервування смуги пропускання (SRP – Stream Reservation Protocol).
IEEE 802.1Qau	Congestion Management.
IEEE 802.1Qav	Forwarding and Queuing Enhancements for Time-sensitive Streams.
IEEE 802.1Qaw	Протокол керування втратами для підтвердження передавання та ізоляції маршруту, де виникли втрати (Management of Data-Driven and Data-Dependent Connectivity Faults).
IEEE 802.1Qay	Provider Backbone Bridge Traffic Engineering (PBB-TE).
IEEE 802.1Qaz	Розширення механізму вибору маршруту передавання для підтримки виділення смуги пропускання для конкретних класів трафіку (Enhanced Transmission Selection).
IEEE 802.1BA	Визначає профілі, що вибирають конфігурацію, протоколи мостів, станцій і мереж для передавання аудіо- та відеотрафіку.

Таблиця В.3 – Поточний список стандартів IEEE 802.3

Назва	Опис
IEEE 802.3	10BASE5 10 Мбіт/с з використанням товстого коаксіального кабелю.
IEEE 802.3a	10BASE2 10 Мбіт/с з використанням тонкого коаксіального кабелю.
IEEE 802.3b	10BROAD36.
IEEE 802.3c	10 Мбіт/с, специфікації повторювача.
IEEE 802.3d	FOIRL (Fiber-Optic Inter-Repeater Link, оптоволоконні лінії між повторювачами).
IEEE 802.3e	1BASE5 або StarLAN.
IEEE 802.3i	10BASE-T 10 Мбіт/с з використанням скрученої пари (категорія 3).
IEEE 802.3j	10BASE-F 10 Мбіт/с з використанням оптоволокна.
IEEE 802.3u	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet, 100 Мбіт/с, автоузгодження швидкостей (сумісність з IEEE 802.3i).
IEEE 802.3x	Підтримка дуплексного зв’язку; сумісність зі стандартом DIX.
IEEE 802.3y	100BASE-T2 100 Мбіт/с з використанням низькоякісної скрученої пари.
IEEE 802.3z	1000BASE-X GigabitEthernet з використанням оптоволоконного кабелю; 1 Гбіт/с.
IEEE 802.3-1998	Версія, що містить в собі всі попередні стандарти з виправленими помилками.
IEEE 802.3ab	1000BASE-T GigabitEthernet з використанням скрученої пари; 1 Гбіт/с.
IEEE 802.3ac	Збільшення максимального розміру кадру до 1522 байтів (для підтримки інформації про VLAN стандарту IEEE 802.1Q і пріоритету стандарту IEEE 802.1p).
IEEE 802.3ad	Агрегація каналів
IEEE 802.3-2002	Версія, що містить в собі всі попередні стандарти з виправленими помилками.
IEEE 802.3ae	10 ГБіт/с Ethernet з використанням оптоволоконного кабелю: 10GBASE-SR, 10GBASE-LR, 10GBASE-ER, 10GBASE-SW, 10GBASE-LW, 10GBASE-EW.
IEEE 802.3af	PoE – електроживлення через Ethernet (Power over Ethernet).
IEEE 802.3ah	Ethernet in the First Mile («Перша миля»).
IEEE 802.3ak	10GBASE-CX4 10 Gbit/s.

IEEE 802.3-2005	Версія, що містить в собі всі попередні стандарти з виправленими помилками.
IEEE 802.3an	10GBASE-T 10 Gbit/s Ethernet з використанням неекранованої скрученої пари (UTP).
IEEE 802.3ap	Ethernet (1 and 10 Gbit/s).
IEEE 802.3aq	10GBASE-LRM 10 Gbit/s (1,250 МБайт/с) Ethernet з використанням мультимодового оптоволокна.
IEEE 802.3ar	Congestion management
IEEE 802.3as	Розширення кадру.
IEEE 802.3at	Покращення живлення через Ethernet.
IEEE 802.3au	Вимоги ізоляції для живлення через Ethernet (802.3-2005/Cor 1).
IEEE 802.3av	10 Gbit/s EPON.
IEEE 802.3ax	Link aggregation з 802.3 в IEEE 802.1
IEEE 802.3ay	Оновлення базового стандарту.

Таблиця В.4 – Поточний перелік стандартів IEEE 802.11

Назва	Рік прийняття	Опис
IEEE 802.11	1997	Початковий стандарт передавання зі швидкістю 1Мбіт/с та 2Мбіт/с в частотному діапазоні 2,4 ГГц.
IEEE 802.11a	1999	Стандарт передавання зі швидкістю 54 Мбіт/с, частотний діапазон 5 ГГц.
IEEE 802.11b	1999	Модифікація 802.11 для підтримки швидкості 5,5 Мбіт/с та 11 Мбіт/с.
IEEE 802.11c	2001	Процедури мережних мостів (внесено в стандарт IEEE 802.1D).
IEEE 802.11d	2001	Розширення 802.11 для регулювання роботи в нових інформаційних галузях (країнах), тобто, інтернаціональні роумінгові розширення.
IEEE 802.11e	2005	Підтримка функцій забезпечення якості обслуговування QoS (Quality of Service).
IEEE 802.11F	2003	Рекомендація протоколу взаємодії між точками доступу Inter-Access Point Protocol (відкликано 2006 р.).
IEEE 802.11g	2003	Розширення стандарту IEEE 802.11b, передавання зі швидкістю 54 Мбіт/с на відстань до 50 м, частотний діапазон 2,4 ГГц.
IEEE 802.11h	2004	Зміни частотного діапазону 5 ГГц стандарту 802.11a для сумісності з європейськими вимогами.
IEEE 802.11i	2004	Розширенні функцій безпеки (WPA2). Розвиток стандарту IEEE 802.11a. Базується на концепції захищеної мережі, компонентами якої є аутентифікація (за допомогою стандарту IEEE 802.1X разом з сервером RADIUS) та технологія шифрування TKIP.
IEEE 802.11j	2004	Модифікації відповідно до вимог Японії.
IEEE 802.11k		Розширення 802.11 для підвищення продуктивності LAN за рахунок введення процедур керування радіоресурсами.
IEEE 802.11l		Зарезервовано.
IEEE 802.11m		Підтримка стандарту 802.11, що не ввійшла в інші розділи.
IEEE 802.11n	2009	Високошвидкісні локальні мережі Wi-Fi 4. Припускається підвищення номінальної швидкості передавання до 600 Мбіт/с на відстань до 100 м за рахунок більш раціонального використання частотного діапазону (2,4–2,5 ГГЦ або 5 ГГц), технології MIMO, а також удосконалених механізмів керування на фізичному рівні.
IEEE 802.11o		Зарезервовано.
IEEE 802.11p		WAVE (Wireless Access for the Vehicular Environment) – стандарт мобільного доступу до мережі з транспортних засобів.
IEEE 802.11q		Зарезервовано (не плутати зі стандартом IEEE 802.11Q).
IEEE 802.11r		Стандарт швидкого роумінгу, який забезпечує прискорення процедури передавання клієнтів між зонами обслуговування (радіокомірками).

IEEE 802.11s		Розширена зона обслуговування для мереж з топологією mesh (Extended Service Set Mesh Network) – багатозв'язні Mesh-мережі, в яких реалізований принцип самоорганізації.
IEEE 802.11T		Рекомендація WPP (Wireless Performance Prediction), методи тестів за вимірів безпроводового обладнання.
IEEE 802.11u		Взаємодія з мережами інших стандартів (не 802.11), наприклад, сотовими.
IEEE 802.11v		Керування безпроводовими мережами.
IEEE 802.11x		Зарезервовано і не буде використовуватись (не плутати зі стандартом IEEE 802.11X).
IEEE 802.11X		Стандарт захисту безпроводових мереж, які сумісні зі стандартом IEEE 802.11, містить протокол розширеної аутентифікації EAP, протокол захисту транспортного рівня TLS (Transport Level Security), сервер RADIUS.
IEEE 802.11y		Додатковий стандарт зв'язку для частотного діапазону 3,65–3,7 ГГц зі швидкістю передавання даних 54 Мбіт/с на відстань до 5 км (в відкритому середовищі, без перешкод).
IEEE 802.11w		Стандарт захисту безпроводових мереж на рівні керування доступом до середовища.
IEEE 802.11ac	2014	Wi-Fi 5. Стандарт передавання зі швидкістю до 6,77 Гбіт/с для пристроїв з 8 антенами зі зниженим енергоспоживанням.
IEEE 802.11ad	2014	Модифікація стандарту IEEE 802.11ac, передавання зі швидкістю до 7 Гбіт/с, функціонує в частотному діапазоні 60 ГГц, яка не вимагає ліцензування. Стандарт описує технологію WiGig.
IEEE 802.11ah	2017	Має також назву Wi-Fi HaLow і працює в неліцензованому частотному діапазоні 900 МГц, шо дозволяє передавати дані, не враховуючи перешкоди, зі швидкістю від 100 Кбіт/с до 340 Мбіт/с (з використанням 4 потоків).
IEEE 802.11ay	2019	Розширення стандарту IEEE 802.11ad зі збільшенням частотного діапазону в 4 рази та використанням MIMO з 4 потоками. Другий стандарт, який регламентує технологію WiGig.
IEEE 802.11ax	2019	Wi-Fi 6. Новий стандарт, який визначає швидкість передавання до 10,7 Гбіт/с.
IEEE 802.11az	2021	Перспективний стандарт, який знаходиться на стадії розробки.

ПРЕДМЕТНИЙ ВКАЗІВНИК, ЛАТИНСЬКИЙ АЛФАВІТ

ПРЕДМЕТНИЙ ВКАЗІВНИК, КИРИЛИЧНИЙ АЛФАВІТ

Н

О

П

Р

С

Т

Ц

Ш

Навчальне видання

Азаров Олексій Дмитрович
Захарченко Сергій Михайлович
Кадук Олександр Володимирович
Орлова Марія Миколаївна
Тарасенко Володимир Петрович

КОМП'ЮТЕРНІ МЕРЕЖІ

Підручник

Рукопис оформив *С. Захарченко*

Редактор *В. Дружиніна*

Оригінал-макет підготував *О. Ткачук*

Підписано до друку 13.07.2020.
Формат 29,7×42 ¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різографічний. Ум. друк. арк. 22,68.
Наклад ____ пр. Зам. № 2020-076.

Видавець та виготовлювач
Вінницький національний технічний університет,
інформаційний редакційно-видавничий центр.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95,
м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
E-mail: kivc.vntu@gmail.com
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.