МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Ю. В. КОВАЛЬ
А. Б. СТАВРОВСЬКИЙ

ІНФОРМАЦІЙНІ МЕРЕЖІ

Навчальний посібник

Київ
2021

УДК [004.55+004.7+004.67](075-028.42)=161.2

К56

Рецензенти:
д-р фіз.-мат. наук, проф. О. О. Марченко,
канд. фіз.-мат. наук, доц. Т. В. Панченко

Рекомендовано до друку
вченою радою факультету комп’ютерних наук та кібернетики
(протокол No 14 від 17 квітня 2021 року)

Коваль Ю. В.

К56 Інформаційні мережі: навчальний посібник / Ю. В. Коваль, А. Б. Ставровський. – Київ, 2021. – 84 с.

Розглянуто основні поняття, постановки задач в інформаційних системах та мережах, їх структуру та організацію. Кожен розділ проілюстровано прикладами задач. Для студентів спеціальності "Інформатика", аспірантів, наукових працівників та інженерів, що спеціалізуються в галузі досліджень з інформаційних мереж та систем.

УДК [004.55+004.7+004.67](075-028.42)=161.2

ПЕРЕДМОВА

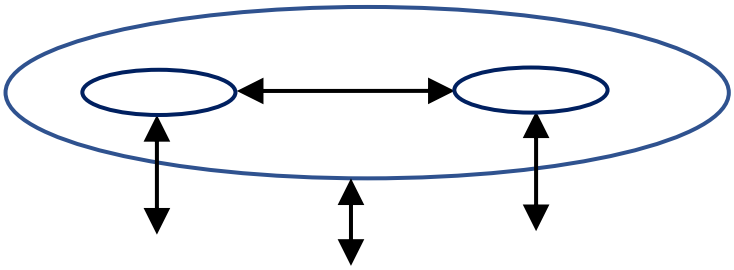
Навчальний посібник підготовлено на основі семестрового курсу лекцій для студентів спеціальності "Інформатика" факультету комп’ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка. Викладено основоположні поняття, розглянуто постановки задач інформаційних мереж. У першому розділі наведено загальне поняття мережа. У другому розглянуто стек протоколів TCP/IP, який став результатом багаторічної роботи зі створення програмного забезпечення для керування передачею даних спочатку в межах однієї глобальної мережі ARPANET, потім у об’єднанні трьох глобальних мереж з різною архітектурою, до яких пізніше були приєднані інші численні мережі. Зрештою, розроблені програми стали «серцем» Інтернета в його сучасному стані. Третій розділ присвячений питанням передачі даних у локальних мережах. Четвертий розділі розглядає передачу даних між мережами, п'ятий – транспортні протоколи. У шостому розділі представлено систему доменних імен й організацію даних у ній, які дозволяють за доменним ім’ям хоста отримати його IP-адресу, і навпаки, за IP-адресою – доменне ім’я. Сьомий розділ присвячено таким прикладним питанням, як передача файлів, технологія «клієнт-сервер» та протоколи передачі даних FTP та гіпертекстових даних HTTP.

Розділ 1. Огляд основних понять

1.1. Загальні поняття

Мережа – це загальне поняття, яке має різні втілення й тлумачення.

Мережа – це об'єднання однорідних об'єктів, яке має правила взаємодії його членів між собою й з об'єктами поза ним. Однорідність членів мережі дозволяє взаємодіяти з ними в єдиний спосіб, об'єднання – взаємодіяти з мережею як єдиним об'єктом.



Приклади мереж: господарські (водопровід, каналізація), енергетичні, транспортні, поштові (об'єкти поштового зв'язку й поштові маршрути), торговельні (сукупність торговельних підприємств, що забезпечують рух товарів), суспільні.

✓ Мережі можуть бути елементами більших, загальніших мереж, утворюючи ієрархії.

Дещо вужче поняття мережі тлумачать як сукупність шляхів, ліній зв'язку, каналів, однорідних закладів, підприємств, пристроїв тощо, розташованих на певній території та зв'язаних в єдину систему. У цьому розумінні мережа – це **інфраструктурна система** для забезпечення діяльності учасників та споживачів мережі. Стосовно інформаційної діяльності учасників, інфраструктурну систему інколи розглядають як окреме поняття.

Телекомунікаційна мережа – це система каналів зв'язку, призначена для передачі інформації шляхом передачі даних у вигляді повідомлень.

Інформаційна взаємодія в телекомунікаційних мережах -- це **обмін сигналами** (електричними, оптичними або електромагнітними хвилями в етері).

Сигнал – це **фізичний процес**, що містить у собі деяку інформацію. Носієм інформації є та чи інша характеристика носія сигналу, що **змінюється в часі**, наприклад, в електричному ланцюзі сигналом є зміна струму або напруги.

Інформація – відомості про факти, явища, події, які можуть становити інтерес і які можна зобразити, обробити та передати.

Дані – зображення інформації у вигляді, який дозволяє її передати й отримати. У інформаційній взаємодії дані являють собою послідовності сигналів.

Повідомлення – це дані, об'єднані змістом і придатні для обробки та передачі.

Телекомунікаційна мережа здійснює необхідні перетворення даних під час передачі. Різновиди повідомлень, що передаються: текстові, звукові, зображення та інші, а також спеціальні повідомлення для керування мережею (повідомлення про зміни, помилки, несправності тощо). Повідомленнями обмінюються **об'єкти** мережі – **джерело** (передавач, відправник) й **адресат** (приймач, одержувач).

✓ Телекомунікаційна мережа надає об'єктам, взагалі, віддаленим один від одного, можливість **інформаційної взаємодії**. Об'єктами інформаційної взаємодії є як обчислювальні пристрої, так і окремі мережі.

Приклади сучасних телекомунікаційних мереж:

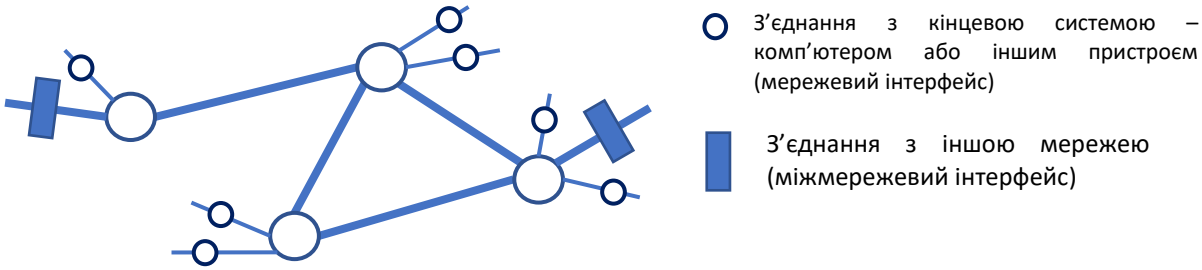
- комп’ютерна мережа,
- телефонна мережа,
- мережа стільникового зв’язку,
- мережа кабельного телебачення.

В якості каналів передачі даних у телекомунікаційних мережах використовуються різноманітні засоби й середовища: телефонні кабелі, спеціальні мережеві кабелі (коаксіальні кабелі, кручені пари, волоконно-оптичні кабелі), радіохвилі, світлові сигнали.

Далі замість слів «телекомунікаційні системи» вживаються просто «комунікаційні системи».

За типом сигналу відрізняють **цифрові** та **аналогові** мережі.

Комп’ютерна мережа – мережа, утворена з ліній зв’язку та мережевого обладнання так, що приєднані до них комп’ютери можуть обмінюватися один з одним ресурсами (процесорний час, пам’ять, дані тощо).



Комп'ютерні мережі здатні забезпечувати:

- швидку передачу даних на великі відстані;
- оперативний пошук інформації на віддалених комп’ютерах;
- обмін текстової, звукової та відеоінформації у режимі реального часу;
- інтерактивний та оперативний зв'язок між віддаленими комп’ютерами.

Вузол (англ. node) є одним з елементів комп’ютерних мереж – це пристрій, з’єднаний з іншими пристроями як частина комп'ютерної мережі. Вузлом може бути комп'ютер, мобільний засіб зв’язку або спеціальний мережевий пристрій.

Хост (англ. host – господар) – це комп’ютер, який має власне підключення до мережі й містить зображення інформації, призначене для відправки іншим об’єктам мережі.

Вузли в мережах обмінюються даними за певними алгоритмами, які відповідають призначенню вузлів і реалізовані апаратно або програмно. Реалізацію алгоритмів обміну даними в мережах називають **мережевими протоколами**. Існують сотні різноманітних протоколів і робота зі створення нових протоколів та вдосконалення існуючих ведеться постійно. Точніше про термін «мережевий протокол» див. п. 2.1.

Кожен комп’ютер з’єднується з комунікаційною мережею за допомогою спеціального пристрою – мережевого адаптера, який називається (мережевим) **інтерфейсом**.

✓ В основі передачі даних лежить **адресація мереж і вузлів**: кожна мережа й кожен мережевий інтерфейс кожного вузла мають унікальний номер – **адресу**. Кожна одиниця даних, що передається, містить адресу вузла призначення, за якою визначається маршрут передачі в мережі.

1.2. Класифікація комп’ютерних мереж за охопленням простору

Комп’ютерні мережі класифікуються за багатьма різними критеріями, тобто певними наборами ознак або характеристик: охоплення фізичного простору, топологія, апаратне забезпечення, середовище передачі, розділення даних під час їх передачі, доступність даних для учасників, швидкість передачі, рівень гарантій передачі даних та інші.

Критерієм **класифікації за охопленням простору** є спосіб керування мережею та розміри простору. Базовою є така класифікація.

LAN (Local Area Network, **локальна мережа**, **ЛМ**) керується одноосібно й зв’язує пристрої однієї особи або організації в межах квартири, будинку або кількох будівель. Локальні мережі відносно прості, тому легкі в керуванні й надійні.

MAN (Metropolitan Area Network – **мегаполісна мережа**, **регіональна (міська) мережа**, мережа «метрополітен») – це транспортна інфрастуктурна мережа, яка забезпечує передачу даних між локальними мережами, зазвичай, у межах величезного міста. Назва походить від того, що такі мережі починали створювати, прокладаючи дротові та кабельні лінії зв’язку тонелями метро.

WAN (Wide Area Network – **глобальна мережа**) – це мережа, яка має структуроване централізоване керування з єдиною політикою й накриває країни та континенти. Історичні приклади: корпоративна мережа фірми Digital, мережа Пентагона. Побудовані свого часу WAN продовжують функціонувати, але замість них тепер створюють віртуальні приватні мережі (VPN).

Структура комп'ютерних мереж ієрархічна й має кілька рівнів мереж: локальні, регіональні та магістральна (базова) мережа, що включає міжміські й міжнародні канали зв'язку та високошвидкісні магістральні вузли комутації (рис. 1.1). Сучасні базові мережі забезпечують транспортний сервіс для територіально віддалених локальних і регіональних мереж. За логічною та фізичною структурою регіональні мережі практично збігаються з глобальними, а відрізняються лише охопленням простору.

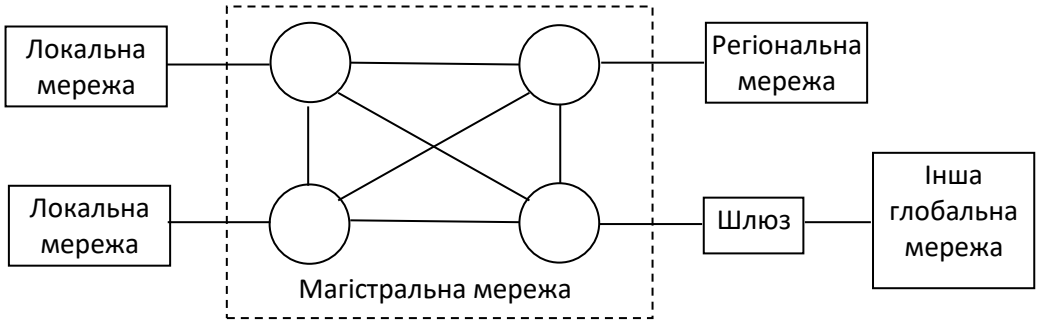


Рис. 1.1. Загальна структура глобальної мережі

- ✓ Глобальні та регіональні мережі **орієнтовані на з'єднання**: перед початком передачі даних абоненти підтверджують і переконуються, що обидва вони готові до прийому або передачі даних. У локальних мережах дані передаються без попереднього підтвердження готовності одержувача до обміну.

Окрім зазначених, відрізняють ще два типи мереж.

PAN (Person Area Network, персональна мережа) зв’язує пристрої окремої особи (в межах кількох метрів) без використання засобів інших мереж. Тепер ці мережі часто є безпровідними (безпровідний зв’язок комп’ютера з клавіатурою, мишею або принтером, Bluetooth).

CAN (Campus Area Network, «кампусна», університетська мережа) зв'язує пристрої тимчасових добровільних учасників, наприклад, мешканців гуртожитків у студентському містечку, без використання засобів інших мереж.

1.3. Топології мереж

Топологія (компонування, конфігурація, структура) комп'ютерної мережі – це розташування комп'ютерів мережі один відносно іншого й спосіб їх з'єднання лініями зв'язку.

Існують різні поняття топології, відповідно до різних рівнів мережевої архітектури:

- **фізична** топологія – схема розташування комп'ютерів і прокладки кабелів;
- **логічна** топологія – структура зв'язків, характер розповсюдження сигналів мережею;
- топологія **керування обміном** – правила надання мережевих ресурсів окремим комп'ютерам;
- **інформаційна** топологія – напрямки потоків даних, що їх передає мережа.

Є два основні класи топологій – широкомовні та послідовні. У першу чергу, це класи фізичних топологій локальних мереж.

Широкомовні топології: кожен комп'ютер передає сигнали, які мають сприйматися всіма іншими. Для цього потрібні відносно потужні приймачі й передавачі, здатні працювати з сигналами у широкому діапазоні рівнів, а також повторювачі та підсилювачі сигналів. Найчастіше зустрічаються широкомовні топології таких типів: спільна шина, дерево та зірка (рис. 1.2).

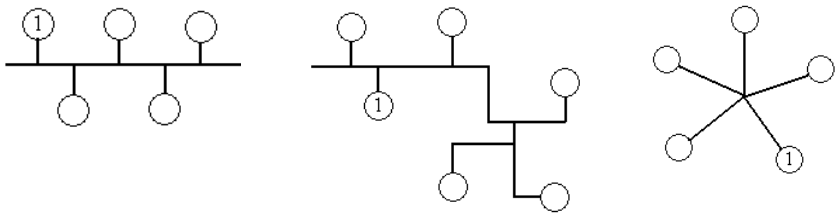


Рис. 1.2. Приклади широкомовних топологій: спільна шина, дерево, зірка

Спільна шина значно спрощує логічну й програмну структуру мережі, знижує витрати кабелів.

Дерево – це розвинений варіант спільної шини, в якому кілька шин з'єднані активними повторювачами або мережевими концентраторами (**хабами**). Завдяки активним повторювачам відмова одного сегмента не призводить до виходу з ладу інших. У разі відмови повторювача дерево розділяється на два піддерева або на дві шини.

Зірка – це варіант дерева, що має корінь з гілками до кожного підключеного комп'ютера. У центрі зірки може бути пасивний хаб, що є простим і надійним пристроєм. Зазвичай, зірка вимагає багато кабелю. *Завдяки простоті «зірка» поступово витісняє інші різновиди топологій*.

Послідовні топології: кожен комп'ютер передає дані тільки одному з усіх інших. До передавачів та приймачів систем тут менше вимог, ніж у широкомовних структурах, і різні ділянки мережі можуть бути реалізовані різними типами фізичного середовища.

Найбільш поширені послідовні топології: «довільні з'єднання», ієрархічна, кільце, ланцюжок, зірка з «інтелектуальним» центром, «сніжинка» (рис. 1.3).

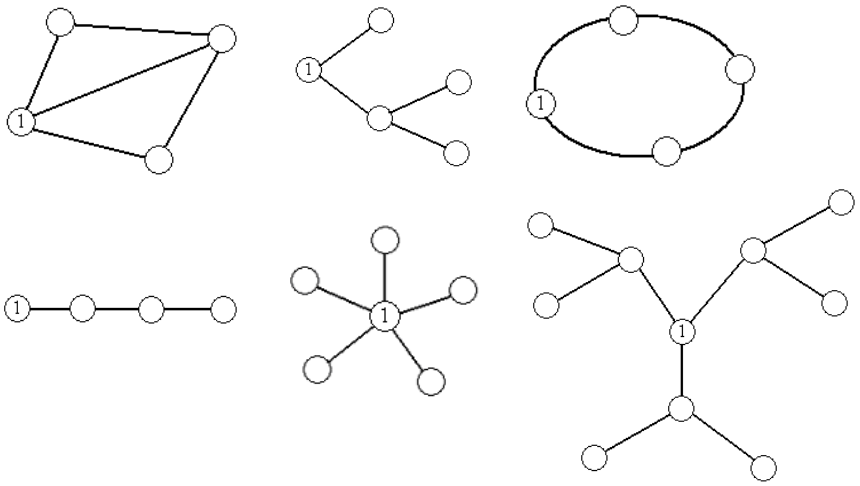


Рис. 1.3. Приклади послідовних топологій

«**Довільні з'єднання**»: деякі пристрої зв'язано з іншими безпосередньо. У лініях можуть бути різні методи передачі. Цей спосіб з'єднання прийнятний для мереж з обмеженою кількістю абонентів, є відносно простим, але має високу вартість і велику кількість каналів зв'язку. Також він потребує визначення маршрутів передачі даних.

Ієрархічне з'єднання: проміжні вузли працюють за принципом «збери й передай». Перевага – оптимальне поєднання елементів. Недоліки – складність логічної та програмної структури, різна швидкість передачі на різних рівнях.

Кільце, **ланцюг**, **зірка з «інтелектуальним» центром**, «**сніжинка**» – для правильного функціонування мережі необхідна постійна робота всіх її елементів. Щоб зменшити цю залежність, у кожен елемент включають реле, яке блокує його, коли він вийшов з ладу. У кільці сигнали передаються тільки в одному напрямку. Переваги: прості методи керування, низькі енерговитрати, легкість розширення мережі. Недоліки: порівняно повільна передача даних (залежно від кількості вузлів) та невисока надійність.

1.4. Однорангові та клієнт-серверні мережі

За правами учасників комп'ютерні мережі підрозділяються на однорангові та мережі типу «клієнт-сервер».

«**P2P**», **пірингова**, або **однорангова мережа** – це мережа, в якій усі учасники мають однакові права, наприклад, у мережах типу «шина» або «довільні зв'язки».

Переваги однорангових мереж: простота організації та відносна дешевизна.

Для керування ними не потрібні додаткові дорогі комп'ютери та системне адміністрування (керування розподілом ресурсів, правами доступу й захистом даних, резервне копіювання файлів тощо); користувачі комп'ютерів можуть самі керувати своїми ресурсами.

Недоліки однорангових мереж: слабка система контролю й протоколювання роботи мережі, відносно низька ефективна швидкість передачі даних, труднощі з резервним копіюванням розподіленої інформації. Вихід з ладу будь-якого комп'ютера, який надає послуги іншим, призводить до втрати частини загальних даних.

Однорангові мережі зазвичай невеликі (до 10 комп'ютерів), а за більшої кількості комп'ютерів мережеві операції відчутно уповільнюють роботу комп'ютерів і створюють інші проблеми. Найбільш поширені однорангові мережі на основі ОС Windows.

Мережа типу «клієнт-сервер» – це мережа, в якій є головний та підпорядковані учасники, наприклад, у мережах типу «зірка» або «дерево».

Клієнт-серверні локальні мережі створюються, коли в мережу треба об'єднати багато користувачів. У мережу включають спеціалізований комп'ютер – сервер.

Сервер – це абонент мережі, який надає свої ресурси іншим абонентам, не використовуючи їхніх ресурсів.

Виділений сервер – це сервер, що займається тільки мережевими задачами.

Клієнт, або **робоча станція** – це абонент мережі, який використовує мережеві ресурси, а свої ресурси не надає.

Будь-який комп'ютер може бути одночасно як клієнтом, так і сервером.

✓ Під сервером і клієнтом часто розуміють не власне комп'ютери, а програми, що на них працюють.

Зазвичай, сервери спеціально оптимізовані для швидкої обробки мережевих запитів на колективні ресурси та для керування захистом файлів і каталогів. У мережах великих розмірів може бути кілька серверів. Сервери можуть виконувати й інші завдання: мережевий друк, вихід в глобальну мережу, зв'язок з іншою локальною мережею, обслуговування електронної пошти тощо. Зазвичай, на сервері встановлюється спеціальна мережева ОС, оптимізована для виконання специфічних операцій з організації мережевого обміну. Клієнтами можуть керувати будь-які ОС, що підтримують взаємодію в мережі.

Переваги мереж типу клієнт-сервер.

- У них може бути до кількох тисяч користувачів, і кількість підключених комп’ютерів можна легко змінювати.
- Вони надійні, якщо надійним є сервер. Проте, відмова сервера паралізує всю мережу, на відміну від одноранговоі мережі, де відмова деяких комп'ютерів може не порушити роботу всієї мережі.
- Цим мережам притаманна висока швидкість обміну даними, оскільки на сервер завжди ставлять швидкий процесор, можливо, не один, оперативну пам'ять великого об'єму та швидкі носії даних. Це дозволяє ефективно керувати доступом до даних, їх захистом та протоколюванням обміну.
- Централізоване адміністрування в цих мережах полегшує обслуговування мережі й дозволяє оперативно вирішувати всі питання. Особливо це важливо для надійного захисту даних від несанкціонованого доступу.

Недоліки мереж на основі сервера.

- Громіздкість за невеликої кількості комп'ютерів.
- Залежність усіх клієнтів від сервера.
- Висока вартість мережі через дорогі сервери.
- Для адміністрування мережі потрібен спеціаліст з відповідною високою кваліфікацією.

1.5. Ще кілька класифікацій комп’ютерних мереж

За апаратурою

- **ZAN** (Zero Area Network, «нульова» мережа) – мережа всередині комп’ютера, яка керується засобами ОС.
- **DAN** (Direct Area Network, мережа з безпосередніми зв’язками) – «прямий зв’язок» між елементами. **Приклад**: Wi-Fi на відстані в кілька десятків метрів без проміжних вузлів.

- **Middle Range WAN** – транспортна апаратура посередника (мережа комутаторів), однотипне обладнання.
- **Long Range WAN** – транспортна апаратура різних технологій, інших ніж у локальних мережах (оптика, радіохвилі на десятки й сотні тисяч кілометрів).

За обмеженістю середовища передачі

- **Обмежені** – наприклад, дріт або кабель. Вважається, що сигнал не виходить за межі дроту (хоча, насправді, це не так).
- **Необмежені** – наприклад, Wi-Fi, WiMAX, мережі звукового діапазону. Сигнал передається в просторі.

За розділенням даних

- **Мережі без розділення даних** – інформаційний об'єкт передається як цілісний.
- **Мережі з розділенням даних** – інформаційний об'єкт передається фрагментами.

Кожен канал зв'язку має ненульову ймовірність помилки. Що довше повідомлення, то вище ймовірність помилки під час його передачі. У багатьох ситуаціях, якщо помилку виявлено, то необхідно передати повідомлення знову. Щоб цього уникнути, повідомлення розбивають на короткі фрагменти – **пакети**, які передають і обробляють окремо. Окрім того, з суто технічних причин передача коротких пакетів ефективніше використовує можливості каналів зв'язку й дозволяє досягти вищої їх продуктивності.

За доступністю пакетів

- **Широкомовні мережі** (Broadcast-мережі) – всім учасникам мережі доступні всі пакети. Кожен пакет має адресу вузла призначення.
- **Послідовні мережі (**Point-to-point-мережі, peer to peer) – адреса вузла призначення у пакеті може бути, але вона не потрібна.

✓ Доступність пакетів для учасників мережі ніяк не зв'язана з її топологією.

1.6. Огляд типів ліній зв'язку

Вузли з'єднуються за допомогою **пасивного** та **активного** мережевого обладнання.

Пасивне мережеве обладнання – це з'єднувальне обладнання , яке не потребує джерел електроживлення (кабелі, телекомунікаційні роз'єми, комутаційні шнури, монтажні елементи).

Активне мережеве обладнання – це мережева апаратура, яка функціонує, споживаючи електроенергію від зовнішніх джерел живлення.

Активне та пасивне обладнання разом утворюють **фізичне мережеве середовище**.

У комп'ютерних мережах використовуються **лінії зв'язку** (**ЛЗ**) кількох типів:

- мідні дроти та кабелі – симетричні («кручена пара»), коаксіальні та оптоволоконні,
- наземні радіосигнали WiMAX (Worldwide Interoperability for Microwave Access) та Wi-Fi (Wireless Fidelity – «безпровідна точність»),
- супутникові радіосигнали.

«**Кручена пара**» (twisted pair). Зазвичай, це дві або чотири кручених пари мідних дротів з діаметром у межах 0,4 - 0,65 мм у ізоляторі, але пар може бути й набагато більше. Існують кілька різновидів з різними степенями захисту від електромагнітних перешкод.

Залежно від різновиду та технології, за якою передаються дані, швидкість передачі може досягати 100 Мбіт/с, 1 Гбіт/с, 10 Гбіт/с, 100 Гбіт/с.

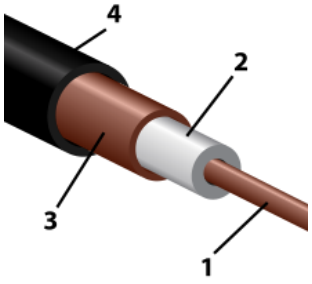
Кручені пари застосовують, у першу чергу, в локальних мережах, але також і в регіональних (мережах провайдерів). Їх прокладають як усередині будівель, так і зовні, в телефонних каналах та для монтажу повітряних ліній (для цього вони виробляються з додатковим сталевим дротом діаметром 1.2 мм). Підключаються до мережевого обладнання через роз'єм **8P8C** («народна назва» **RJ45**, хоча це насправді назва не роз'єму, а стандарту інтерфейсу (Registered Jack), а роз'єм просто найбільш популярний серед тих, які цьому стандарту відповідають).

Кабелі «кручена пара» **помилково** називають кабелями Ethernet. Насправді, Ethernet – це *сім'я технологій* передачі даних. Такі технології, неформально й абстраговано, визначають способи передачі даних, які включають кабельні з'єднання та електричні сигнали, способи формування даних для їх передачі, правила керування доступом до середовища. Технології Ethernet в якості носія сигналів включають не тільки кручені пари, але й коаксіальні та оптичні кабелі.

Переваги кручених пар порівняно з іншими різновидами кабелів: дешеві, надійні, стійкі до перешкод, легко деформуються. Це зумовлює їх надзвичайну популярність у локальних мережах. Локальні мережі з технологіями Ethernet вже багато років займають більше 90% ринку («без Ethernet немає Internet»).

Недоліки кручених пар: відносно мала пропускна спроможність, порівняно жорсткі обмеження на кількість кінцевих пунктів, сигнали від яких передаються через кабель, порівняно мала довжина, на якій послаблення сигналу залишається прийнятним і не вимагає підсилювачів.

Коаксіальний кабель з мідною жилою («коаксіал»). Коаксіальний кабель має структуру, зображену на рисунку: 1 – внутрішній провідник, 2 – ізоляція (суцільний поліетилен), 3 – зовнішній провідник (екран), 4 - оболонка (спеціальний поліетилен, стійкий до сонячного світла).

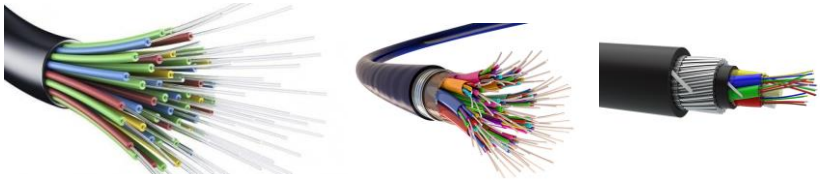


Коаксіальний означає **співвісний** (англ. coaxial). Осі внутрішнього та зовнішнього провідників збігаються, і завдяки цьому електромагнітне поле, що виникає в кабелі під час передачі сигналів внутрішнім провідником, не виходить за межі кабелю. Завдяки цьому електромагнітна енергія не випромінюється за межі кабелю, а кабель захищений від зовнішніх електромагнітних наведень.

Коаксіальний кабель порівняно з крученою парою допускає довші ділянки з прийнятним послабленням сигналу. Проте натепер коаксіал практично цілком витіснений крученою парою. Основне використання коаксіалу – не в мережах, а в телебаченні.

Волоконно-оптичний (оптоволоконний) кабель, або діелектричний хвилевід виробляється на основі волоконних світловодів і передає оптичні сигнали у вигляді світла (фотонів). Має найвищу пропускну здатність із усіх існуючих засобів зв'язку і забезпечує найвищу швидкість передачі даних (залежно від типу активного обладнання та технології передачі це сотні Гбіт/с і навіть кілька Тбіт/с). Отже, швидкість передачі тут на порядки перевищує швидкості мідних кабелів. Випускається в численних різноманітних варіантах, які залежно від призначення містять від кількох світловодів до кількох їх десятків і навіть сотень.

Приклади кабелів наведено на зображеннях.



Сигнал передається кварцовим склом, яке має дуже мале згасання сигналу й нечутливе до зовнішніх електромагнітних полів. Ці кабелі дозволяють утворювати неперервні ланки без підсилювачів завдовжки більше 100 км і надійно захищають від несанкціонованого доступу та перехоплення даних.

Оптоволоконні кабелі використовуються, зокрема, для зовнішніх магістралей, які з'єднують об'єкти на відстані до сотень і тисяч км, розташовані, можливо, на різних континентах.

Безпровідні лінії зв'язку. Обмежимося розглядом двох типів радіосигналів, не торкаючися супутникових.

Wi-Fi (Wireless Fidelity, «безпровідна точність») – це технологія безпровідних **локальних мереж**, а також сім'я стандартів передачі цифрових потоків даних по радіоканалах. Носієм сигналу є радіохвилі високої частоти, які мають обмежений простір розповсюдження (до 300 м, а в будівлях зазвичай десятки метрів). Саме на основі технології Wi-Fi створюються **безплатні точки доступу в загальнодоступних місцях**. Пропускна здатність за діючим стандартом IEEE 802.11n – до 300 Мбіт/с.

WiMAX (Worldwide Interoperability for Microwave Access) – жаргонна назва технології, метою розробки якої є надання універсального бездротового зв'язку на великих відстанях і для різноманітних пристроїв (робочих станцій, портативних комп'ютерів, мобільних телефонів). Заснована на стандарті IEEE 802.16, назва якого Wireless MAN. Насправді, це два стандарти.

Стандарт WiMAX IEEE 802.16d: безпровідний зв'язок між стаціонарними вузлами, до яких територіально прив'язані користувачі. Пропускна здатність до 70 Мбіт/с, радіус дії радіохвиль до 80 км. На базі таких ліній зв'язку створюють муніципальні мережі.

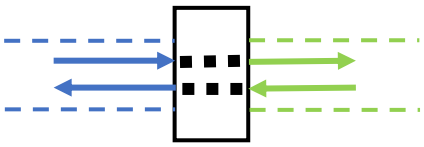
Стандарт WiMAX IEEE 802.16e: для мобільних користувачів. Пропускна здатність до 40 Мбіт/с, радіус дії радіохвиль до 5 км.

1.7. Огляд активного мережевого обладнання

Активне мережеве обладнання – це мережева апаратура, яка функціонує, споживаючи електроенергію від зовнішніх джерел живлення.

Активні пристрої розділяються на такі, що передають сигнали між двома своїми портами, і такі, що можуть мати більш ніж два порти.

Пристрої, які передають сигнали між двома своїми портами. До портів приєднано дві лінії зв’язку, або дві ланки однієї лінії зв’язку, або до одного з портів безпосередньо комп’ютер.



Мережевий адаптер (network interface controller, NIC) – це плата, яка встановлюється в комп'ютер і забезпечує його під'єднання до локальної мережі.

Мережевий адаптер може бути:

- вбудований (вбудований у материнську плату комп’ютера),
- внутрішній (окрема плата, яка вставляється в спеціальний роз’єм на МП,
- зовнішній (окремий пристрій, що приєднується до зовнішнього роз’єму).

Приклади адаптерів на зображенні: ліворуч і в центрі внутрішні, праворуч зовнішні.

Повторювач (**ретранслятор**) – це прилад, як правило, з **двома однотипними портами**: очищує, підсилює й повторює сигнал, що з’являється на одному з портів, і передає на інший порт.

Головне призначення повторювачів – подовжити лінію зв'язку.

Медіаконвертер – це прилад, який перетворює й передає сигнали **між *різними* середовищами**.

Типові приклади пар середовищ: коаксіальний кабель-кручена пара, кручена пара-оптоволоконний кабель.

Ретранслятор – у бездротовій мережі (тут немає ліній зв’язку як таких) сприймає радіосигнал від іншого мережевого пристрою й передає його на певну частину простору.

Головне призначення ретрансляторів – збільшити площу покриття й охопити раніше недоступні ділянки.

Пристрої, до яких може бути приєднано дві або більше ліній зв'язку. Пристрій отримує дані однією з ліній і або передає їх на всі інші лінії, або взагалі не передає, або передає на одну з них, «приймаючи рішення» за певним алгоритмом на основі даних, що передаються.

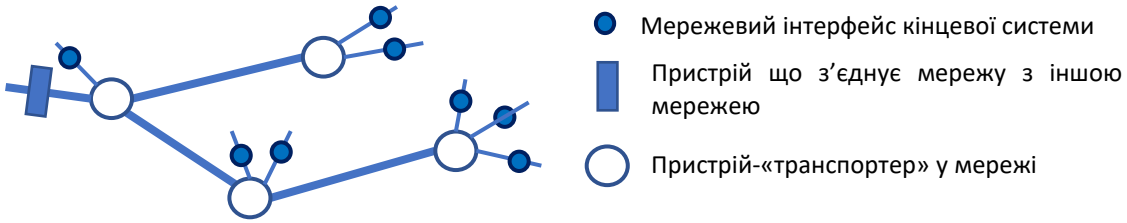
✓ Дані в лініях зв’язку передаються порціями – так званими **кадрами**, або **фреймами** (frame). Кожен кадр містить заголовок, в якому вказано місце призначення та місце відправлення цих даних.

Мережеві пристрої позначаються їхніми фізичними адресами.

Фізична адреса (Hardware Address) – це унікальний числовий ідентифікатор, що присвоюється кожній одиниці активного обладнання комп'ютерних мереж у процесі її виробництва.

Формат фізичної адреси стандартний. Найпоширеніший формат MAC-48 – Media Access Control (керування доступом до середовища), 48 біт.

Місцями призначення та відправлення в кадрах є або мережеві інтерфейси кінцевих систем, або пристрої «на межі» мережі (на рисунку залиті темним кольором).



Різновиди пристроїв-«транспортерів» у провідникових мережах.

Концентратор (активний **хаб**, або багатопортовий ретранслятор) має кілька портів, об'єднаних електросхемою. Кадр, що прибуває на який-небудь з портів, передається **на всі інші порти**. Концентратор узгоджує роботу своїх портів, до яких приєднані лінії, що мають працювати **з однаковими швидкостями**. Портів не менше ніж 4.

✓ Натепер концентратори «відходять у минуле», оскільки ефективнішими є комутатори.

Комутатор, або **світч** (switch), або **багатопортовий міст** має не менш ніж 4 порти (максимум, що його бачив один з авторів – 288 портів). До портів приєднані лінії зв'язку, що ведуть до інтерфейсів кінцевих систем або до інших комутаторів у ЛМ або до межі ЛМ.

Комутатор, на відміну від концентратора, за адресою в кадрі визначає, чи потрібно відправляти кадр далі, і якщо так, то в який саме порт (або порти). Комутатор може одночасно передавати дані між кількома парами портів.

Комутатор є мультипроцесорним пристроєм, здатним просувати дані одночасно між будь-якими парами своїх портів. Його центральний процесор координує роботу портів і підтримує спільні дані, потрібні для просування. Завдяки мультипроцесорності комутатори набагато продуктивніші ніж двоточкові мости.

Сучасні комутатори забезпечують буферизацію даних, необхідну, наприклад, коли дані передаються між портами, приєднаними до ЛЗ з різними швидкостями передачі. Існує проблема вичерпання буферу й відмови від передачі даних – коли дані приходять швидше, ніж їх можна передати. Це можливе, коли з ЛЗ, що має велику швидкість, дані передаються на ЛЗ, яка має малу швидкість, наприклад, з ЛЗ Gigabit Ethernet в 10-Мбіт/с Ethernet. Ця небезпека також існує, коли порти працюють з однією швидкістю, але в порт призначення кадри надходять одночасно з кількох інших портів.

✓ Історично спочатку вироблялися мости з двома портами для територіальних мереж, потім мости з більшою кількістю портів. Їх стали називати світчами (перемикачами) та використовувати також у локальних мережах. Отже, за своїми функціями це одне й те саме. Водночас, пристрої з двома портами, що з'єднують різнотипні мережі й приймають рішення, чи передавати кадри, як і раніше, називаються мостами.

Маршрутизатор, або роутер. Мережі (локальні, регіональні, належні різним володарям, глобальні) з'єднуються між собою за допомогою ЛЗ та маршрутизаторів.

Маршрутизатор, або **роутер** (router) залежно від призначення та устрою, може з'єднувати як окремі сегменти всередині мережі, так і різні мережі, можливо, з різними фізичними носіями сигналів.


✓ Часто роутером називають спеціалізований пристрій, а маршрутизатором – програмне забезпечення, що керує роботою роутера.

Роутер має кілька портів (можливо, десятки й сотні портів) для ліній зв'язку. На відміну від комутатора, обробляє не фізичні адреси пристроїв, а так звані мережеві адреси, а також розв'язує багато інших задач.

Роутер порівняно з комутатором має набагато більшу обчислювальну потужність та зовсім інше програмне забезпечення. Спектр функцій та можливостей маршрутизаторів дуже широкий. Роутер не лише передає дані між мережами, з якими він з'єднаний, а й визначає напрямки передачі, тобто **керує передачею даних** між мережами.

✓ Однією з задач керування передачею даних є **задача маршрутизації** – визначення мережі, в яку потрібно спрямувати дані.

Маршрутизатор часто розглядають як окремий випадок мережевого шлюзу, поняття якого історично виникло раніше.

Мережевий шлюз (gateway) – це програмне забезпечення, яке встановлено на роутер або на комп'ютер і керує перетворенням та передачею даних між мережами різних типів або рівнів (локальна та регіональна тощо). Головна функція шлюзів – передача даних між мережами й визначення маршрутів передачі.

1.8. Інформаційні мережі

Інформаційні мережі мають забезпечувати дистанційний обмін даними як окремих фізичних осіб, так і організацій. Вони створюються на основі комп'ютерних мереж, інформаційних систем та програмного забезпечення, яке керує передачею даних і функціонуванням мереж у цілому.

Уточнимо зазначені тут і деякі інші поняття.

Інформаційна система (Information system) – це сукупність взаємозв'язаних організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів [ДСТУ 2392-94].

Інформаційна система є постачальником або споживачем інформації, тобто це об'єкт, здатний зберігати, обробляти або передавати дані. Інформаційні системи складаються з комп'ютерів, програм, даних та інших складових, потрібних для обробки й передачі даних.

Інформаційна мережа – це мережа для обробки, зберігання й передачі даних, утворена інформаційними системами, які з'єднуються за допомогою комунікаційної (комп'ютерної) мережі.

Абонент системи або мережі – це об'єкт, який **має право** взаємодіяти з системою або мережею.

За різними тлумаченнями, абонентами можуть бути як фізичні або юридичні особи, так і об'єкти, підключені до мережі, які можуть брати участь в обміні даними, – комп'ютери, інші пристрої, локальні мережі, програмні системи. З погляду безпеки даних важливу роль відіграє **реєстрація абонентів**.

Зазвичай, юридичну або фізичну особу, яка є абонентом інформаційної мережі й використовує ресурси мережі, називають **користувачем**. Користувач як фізична особа є **фізичним абонентом**, підприємство або установа – **юридичним**. Користувач взаємодіє з системою або мережею за допомогою певного інтерфейсу.

- ✓ Отже, інформаційна мережа утворюється, коли до комп'ютерної мережі приєднано абонентські інформаційні системи. При цьому на базі деякої комп'ютерної мережі може бути побудована більш ніж одна інформаційна мережа.

Робоча станція, або **клієнт** (client) – це інформаційна система, призначена для вирішення завдань користувачів.

Зазвичай клієнт звертається за необхідними даними до сервера.

Сервер (server) – це комп'ютер, який надає іншим комп'ютерам свої ресурси та доступ до встановлених сервісів (сервер баз даних, файловий сервер, поштовий сервер тощо).

Клієнтами та серверами також називають програми, які за допомогою мережі розв'язують завдання користувача та, відповідно, надають послуги клієнтам.

- ✓ Інформаційна система на боці сервера в багатьох випадках організується у вигляді **веб-сайта** (англ. Website – «місце в павутині»), тобто системи зв'язаних між собою файлів даних і коду, яка має унікальну адресу й сприймається користувачем як єдине ціле.

Інформаційні мережі є відкритими системами.

Відкрита система (за означенням комітету IEEE) – це система, яка реалізує відкриті специфікації (стандарти) на інтерфейси, служби та формати даних й забезпечує:
– можливість перенесення прикладних систем з мінімальними змінами на широкий діапазон систем (**мобільність систем**);
– спільну роботу з іншими прикладними системами на локальних і віддалених платформах (**інтероперабельність**);
– взаємодію з користувачами в системі, яка полегшує перехід від системи до системи (**мобільність користувачів**).

1.9. Приклади інформаційних мереж

Всесвітня павутина (англ. World Wide Web, або WWW) – це найбільше всесвітнє багатомовне сховище інформації в електронному вигляді: множина зв'язаних між собою документів, розташованих на комп'ютерах по всій земній кулі.

Всесвітня павутина дозволяє отримувати доступ до даних незалежно від місця їх розташування. Її також називають **гіпертекстовою**, **гіпермедійною**, **розподіленою**, **інтегруючою** або **глобальною**.

GAN (Global Area Network, **глобальна мережа**) – інформаційна мережа, утворена багатьма різними мережами, що взаємодіють і накривають необмежені території.

Прикладом глобальної мережі є Інтернет.

Інтернет (англ. Internet) – це глобальна мережа, призначена для зберігання й передачі даних. Вона побудована зі з'єднаних між собою комп'ютерних мереж; мережеве програмне забезпечення включає комплект Інтернет-протоколів, які реалізують певні принципи взаємодії комп'ютерів та інших пристроїв у мережах.

У повсякденній мові слово **Інтернет** найчастіше позначає Всесвітню павутину й доступні в ній дані, а не саму фізичну мережу. Інтернет часто згадується як всесвітня мережа, глобальна мережа, або навіть просто «Мережа». Інтернет складається з мільйонів локальних і глобальних приватних, публічних, академічних, ділових і урядових мереж, зв'язаних між собою з використанням різноманітних дротових, оптичних і бездротових технологій. Інтернет утворює основу для розміщення величезної кількості інформаційних ресурсів і послуг, таких як взаємопов'язані гіпертекстові документи Всесвітньої павутини, електронна пошта та інші інформаційні мережі.

- ✓ Відмінною особливістю мережі Інтернет є висока надійність: якщо частина комп'ютерів і ліній зв'язку виходить з ладу, то мережа продовжує функціонувати й передає дані іншими лініями зв'язку. Така надійність забезпечується тим, що **в Інтернет відсутній єдиний центр керування**.

Організації, з'єднані одна з одною якнайшвидшими лініями зв'язку, утворюють магістральну частину мережі. Вони є крупними поставщиками послуг Інтернету (**провайдерами**); до них приєднуються інші провайдери та окремі користувачі (вузли, локальні та регіональні мережі). Різниця між користувачами та провайдерами умовна – будь-який користувач може надати послуги підключення до мережі іншим користувачам.

VPN (Virtual Private Network, віртуальна приватна мережа) – інформаційна мережа, яка об'єднує географічно віддалені мережі організації в єдину мережу, використовуючи для зв'язку між ними непідконтрольні загальнодоступні або віртуальні канали інших мереж.

Інтранет – це різновид VPN: корпоративна інформаційна мережа, що використовує стандарти, технології та програмне забезпечення Інтернету.

Безпека передачі даних через загальнодоступні мережі зазвичай реалізується за допомогою шифрування, завдяки чому створюється канал обміну інформацією, закритий для сторонніх (**тунель**). Це набагато дешевше ніж будувати власні канали зв'язку.

IAN (Internet Area Network), або «хмарна мережа» – мережа зв'язку, яка надійно з'єднує кінцеві точки передачі даних засобами глобальної мережі так, що вони можуть обмінюватися даними, не будучи прив'язаними до певного місця.

Глобальна інформаційна інфраструктура (ГІІ, Global information infrastructure, GII) – це якісно нове інформаційне утворення, яке, за задумом, має являти собою інтегровану загальносвітову інформаційну мережу масового обслуговування населення планети на основі інтеграції глобальних і регіональних інформаційно-комунікаційних систем, а також систем цифрового телебачення і радіомовлення, супутникових систем і мобільного зв'язку. ГІІ використовують різноманітні технології: Інтернет, телекомунікаційні, побутових електронних приладів, інформаційних застосунків або сервісів (індустрія змісту або застосунків – content or application industry).

Соціальна мережа (англ. Social networks) – це вебсайт, який дозволяє зареєстрованим на ньому користувачам розміщувати інформацію про себе й спілкуватися між собою, встановлюючи соціальні зв'язки. Інформація на цьому сайті створюється самими користувачами.

Соціальні мережі об'єднують людей різних країн, релігій, професій, соціальних або вікових груп тощо. Зазвичай вони мають інструменти для створення співтовариств за інтересами, де спілкування відбувається у вужчих колах. У світі найбільш поширені соціальні мережі Facebook, MySpace, Twitter, LinkedIn, Google+. Найбільші темпи зростання кількості користувачів у 2018 р. має мережа Instagram.

1.10. Розширюваність і масштабованість мереж

Розширюваність мережі – це можливість порівняно просто додавати окремі компоненти мережі (користувачі, комп'ютери, служби), збільшувати довжину сегментів кабелів і заміняти існуючу апаратуру потужнішою.

Масштабованість мережі – це можливсть нарощувати кількість вузлів і протяжність зв'язків в дуже широких межах **без зниження продуктивності мережі**.

Розширюваність і масштабованість мережі іноді може забезпечуватися лише в певних межах. Наприклад, локальна мережа Ethernet, побудована на основі одного розділюваного сегмента коаксіального кабелю, є розширюваною, тобто дозволяє легко додавати нові робочі станції. Однак така мережа зазвичай обмежується кількістю станцій у три-чотири десятки, оскільки підключення ще кількох десятків станцій різко знижує продуктивність мережі. Отже, поза вказаними межами мережа залишається легко розширюваною, але стає погано масштабованою.

- ✓ Для забезпечення масштабованості мереж застосовують додаткове комунікаційне обладнання та спеціальну структуризацію мережі. Зазвичай масштабованими є мережі з багаторівневою ієрархічною структурою, яка дозволяє додавати елементи на кожному рівні ієрархії.

Приклад. Інтернет є масштабованою мережею, оскільки технологія його протоколів здатна підтримувати мережу в масштабах земної кулі. Організаційна структура Інтернету має кілька ієрархічних рівнів: мережі користувачів, мережі локальних провайдерів послуг і далі вгору до мереж міжнаціональних постачальників послуг. Технологія протоколів TCP/IP, на якій побудовано Інтернет, дозволяє також створювати ієрархічні мережі. Основний протокол Інтернету IP ґрунтується на дворівневій моделі: нижній рівень – окремі мережі (найчастіше, мережі корпоративних користувачів), верхній – складена мережа, яка об'єднує окремі мережі. Технологія Інтернету підтримує також автономні системи (АС). В АС входять всі мережі одного постачальника послуг, тому АС являє собою вищий рівень ієрархії порівняно з локальними мережами. Наявність АС в Інтернеті дозволяє спростити визначення оптимальних маршрутів передачі даних – спочатку визначаються оптимальні маршрути між автономними системами, а потім всередині окремих АС незалежно одна від одної (див. розділ 4). ◄

Масштабованою має бути не тільки мережа, але й мережеві пристрої, зокрема, ті, що працюють на магістралі мережі. Через зростання мережі неможливо постійно змінювати обладнання, тому магістральні комутатори і маршрутизатори будуються зазвичай за модульним принципом, дозволяючи нарощувати кількість інтерфейсів і продуктивність обробки пакетів.

Контрольні запитання

Поняття: мережа (загальне поняття), комп’ютерна мережа, інформація, дані, повідомлення, комунікаційна мережа, вузол, хост, мережевий протокол, топологія мережі, однорангова мережа, мережа типу «клієнт-сервер», мережа з розділенням даних, широкомовна мережа, послідовна мережа, шлюз, задача маршрутизації, політика маршрутизації, інформаційна система, інформаційна мережа,

абонент та користувач інформаційної системи або мережі, клієнт і сервер, веб-сайт, відкрита система, всесвітня павутина, Інтернет, віртуальна приватна мережа, Інтранет, хмарна мережа, соціальна мережа.

Типи комп’ютерних мереж за простором доступності.

Загальна структура глобальної мережі.

Основні типи топологій мереж, їх переваги та недоліки.

Основні переваги та недоліки одноранговых мереж та мереж типу «клієнт-сервер».

Охарактеризувати функції основних типів мережевих пристроїв (повторювач, конвертер, концентратор, комутатор, маршрутизатор).

Різниця між концентратором і комутатором (мостом).

Різниця між комутатором (мостом) і маршрутизатором.

У чому полягає відкритість інформаційних мереж.

Основні необхідні властивості інформаційних мереж.

Назвати кілька чинників, які впливають на надійність функціонування інформаційних мереж.

Пояснити різницю між інформаційною та фізичною топологією мережі.

Основні області застосування бездротових ліній зв'язку.

Розширюваність і масштабованість інформаційних мереж.

Пояснити, чим зумовлена масштабованість Інтернету.

Розділ 2. Огляд організації мереж на основі TCP/IP

2.1. Стек протоколів TCP/IP

Стек протоколів TCP/IP став результатом багаторічної роботи зі створення програмного забезпечення для керування передачею даних спочатку в межах однієї глобальної мережі ARPANET, потім у об'єднанні трьох глобальних мереж з різною архітектурою, до яких пізніше були приєднані інші численні мережі. Зрештою, розроблені програми стали «серцем» Інтернета в його сучасному стані.

В основі створеного програмного забезпечення лежала ідея *відокремити передачу даних всередині мереж від передачі даних між мережами*. Мережі, які з'єднувалися, були різнотипними й мали різне апаратне та програмне забезпечення, тому *міжмережева передача мала бути незалежною від особливостей з'єднаних мереж*. Так виникли окремі *рівні передачі даних* всередині мереж і між мережами і окремі програми, що забезпечують цю передачу.

Розділення програмного забезпечення на рівні, які передають дані один одному, стало основою для формування стеку протоколів TCP/IP та моделі взаємодії відкритих систем, яка ще називається *модель DoD* (це скорочення від Department of Defence – Міністерство Оборони США). Саме в його структурному підрозділі DARPA (Defense's Advanced Research Projects Agency – Агентство передових оборонних дослідницьких проектів) було створено глобальну мережу ARPANET і розроблено модель взаємодії відкритих систем.

- ✓ Керівниками й авторами розробки моделі DoD та стеку протоколів TCP/IP є Вінтон Грей Серф (Vinton Gray Cerf) і Роберт Елліот Кан (Robert Elliot Cahn). У 2004 р. вони отримали премію Тьюринга (неофіційно, це аналог Нобелівської премії в галузі інформатики) з формулюванням «…за піонерську роботу з проблеми міжмережевого обміну (англ. internetworking), включно з розробкою та реалізацією основних Інтернет-протоколів TCP/IP, і за провідну роль у галузі комп'ютерних мереж».

Стек протоколів TCP/IP складається з чотирьох рівнів (згори донизу):
– рівень процесів і застосунків, або **прикладний** рівень (англ. Process/Application),
– **транспортний** рівень (англ. Transport),
– **міжмережевий** рівень, або рівень міжмережевих інтерфейсів (англ. Internet),
– рівень **мережевого доступу**, або **мережевий** (англ. Network Access).

Мережевий рівень часто називають **канальним** або **фізичним**, а інколи канальний та фізичний рівні розглядають окремо, тобто, фактично, розглядають п'ять рівнів.

Кожен з рівнів має своє призначення й функції, на кожному працюють програми, що реалізують мережеві протоколи та інтерфейси.

Протокол – це заснований на стандартах формалізований набір правил, що визначають послідовність і формат повідомлень, якими обмінюються процеси *одного рівня* в мережевих компонентах, розташованих, зазвичай, *на різних вузлах*. Програмну чи апаратну реалізацію зазначених правил теж називають протоколом.

Інтерфейс – це формалізований набір правил, що визначають послідовність і формат повідомлень, якими обмінюються процеси *сусідніх рівнів на одному вузлі*.

Слова «протокол» та «інтерфейс» мають по два різних значення – набір правил і програма, набір правил і мережевий пристрій. Що саме вони позначають у тому чи іншому реченні, зазвичай зрозуміло з контексту.

✓ Протоколи публікуються в нумерованій серії документів, яка має назву RFC (Request for Comments – запит коментарів). Документи RFC містять технічні специфікації та стандарти, які застосовуються в Інтернеті.

Публікацією документів RFC займається відкрите міжнародне співтовариство проектувальників, учених, мережевих операторів і провайдерів IETF (Internet Engineering Task Force). Воно було створене в 1986 році під егідою відкритої організації ISOC (Internet Society – Товариство Інтернету) і займається розвитком протоколів і архітектури Інтернету.

Мережеві протоколи можуть бути реалізовані як програмно, так і апаратно. Протоколи нижчих рівнів зазвичай реалізуються комбінацією програмних та апаратних засобів, протоколи верхніх рівнів – суто програмно. Протокол надає сервіс протоколу суміжного вищого рівня й використовує сервіс, наданий протоколом суміжного нижчого рівня.

На кожному рівні є два різновиди протоколів:

– протоколи обробки даних користувачів,

– протоколи керування обміном даних у мережах.

✓ Протоколи обробки даних на вузлах забезпечують передачу даних мережами, протоколи керування формують на всіх вузлах мереж дані про стан зв'язків, необхідні для прийняття рішень зі спрямування даних користувачів.

У протоколах Інтернет використовуються три рівні адрес вузлів (згори вниз): *доменні імена*, *IP-адреси*, *фізичні адреси*. Для користувачів, які взаємодіють з програмами прикладного рівня наприклад браузерами, природні й зрозумілі імена хостів (доменні імена), записані літерами, як `univ.kiev.ua`. Проте на транспортний рівень передаються не доменні імена вузлів, а IP-адреси – числові адреси інтерфейсів вузлів і мереж. Саме IP-адресами оперують протоколи транспортного та міжмережевого рівнів. На канальному та фізичному рівнях фігурують фізичні адреси (Hardware Address) інтерфейсів вузлів, присвоєні їм у процесі їх виробництва.

IP-адреси представлені в подальших параграфах цієї теми, доменні імена та фізичні адреси – в наступних розділах.

2.2. Огляд передачі даних

Один з базових принципів передачі повідомлень у мережах: повідомлення розбивається на фрагменти й до кожного з них додаються дані, необхідні для передачі лініями зв'язку в мережах та між мережами. Отже, повідомлення переворюється на послідовність **пакетів** (packet). Неформально, пакети аналогічні конвертам з листами звичайної пошти, а дані, додані в пакетах, – адресам на конвертах.

IP (Internet Protocol) – це протокол міжмережевого рівня, який обробляє кожен пакет з IP-адресою призначення й передає його між мережами або з вузла в мережу або з мережі на вузол.

Протокол IP працює на кожному вузлі мережі (комп'ютері, роутері або шлюзі). За IP-адресою одержувача в пакеті та розміщеними на вузлі даними про стан зв'язків він визначає інтерфейс наступного вузла на шляху до місця призначення й передає пакет у відповідний канал зв'язку (або на вузлі призначення IP передає пакет протоколу транспортного рівня).

Протоколи транспортного рівня працюють тільки на кінцевих вузлах. Основними транспортними протоколами в Інтернеті є TCP та UDP.

TCP (Transmission Control Protocol, протокол керування передачею) – протокол транспортного рівня, який забезпечує гарантовану доставку даних користувача

(звісно, якщо не вийшов з ладу вузол мережі, без якого немає шляху між джерелом і отримувачем).

UDP (User Datagram Protocol, протокол дейтаграм користувача) – протокол транспортного рівня, який швидко передає дані користувача, але не гарантує їх доставку.

Протокол TCP контролює передачу послідовності пакетів, за необхідності він здатен прискорювати або уповільнювати надсилання даних, а в разі втрати даних надсилати їх повторно. UDP передає дані, не контролюючи їх доставку.

На комп'ютерах, зазвичай, IP, TCP та UDP реалізовані в ядрі операційної системи (ОС), хоча й існують реалізації TCP та UDP у вигляді бібліотечних підпрограм.

Розглянемо схему роботи протоколів стеку TCP/IP та перетворення даних у відправника та одержувача на типовому прикладі.

Прикладний рівень. На комп'ютері користувача є повідомлення, яке потрібно відправити на віддалений хост. Процес протоколу прикладного рівня передає IP-адресу віддаленого хоста та повідомлення на транспортний рівень протоколу TCP.

Транспортний рівень. TCP розбиває повідомлення на сегменти й до кожного з них додає TCP-заголовок, утворюючи послідовність TCP-пакетів. TCP-заголовок містить дані, що вказують на програму прикладного рівня, для якої призначено повідомлення, та дані, за якими одержувач може відновити сегмент даних, відновити порядок сегментів, або виявити пошкодження та вирішити, що пакет прийняти неможливо. Нехай далі S позначає сегмент даних, T – TCP-заголовок, TS – TCP-пакет. Отже, TCP виконує таку роботу.

Повідомлення $\rightarrow S_1, S_2, \ldots \rightarrow T_1S_1, T_2S_2, \ldots$

TCP передає протоколу міжмережевого рівня IP послідовність TCP-пакетів та IP-адресу одержувача.

Міжмережевий рівень. IP додає до кожного отриманого пакета IP-заголовок, що містить IP-адреси відправника й одержувача, необхідні для доставки, та інші дані. Кожен утворений IP-пакет, позначений нижче як ITS, де I – IP-заголовок, протокол IP передає на канальний рівень.

$T_1S_1, T_2S_2, \ldots \rightarrow I_1T_1S_1, I_2T_2S_2, \ldots$

Канальний (мережевий) рівень. В апаратурі канального рівня до IP-пакета додається заголовок, який потрібен для передачі пакета лініями зв'язку всередині мережі й відповідає різновиду мережі (Ethernet, Wi-Fi, SONET або іншій). Пакет на канальному рівні має назву **кадр** (**фрейм**, **дейтаграма**) – він є, власне, «одиницею передачі». Сформований кадр, позначений нижче $EITS$, де E – заголовок, спрямовується на мережеве обладнання.

$I_1T_1S_1, I_2T_2S_2, \ldots \rightarrow E_1I_1T_1S_1, E_2I_2T_2S_2, \ldots$

Кадр через інтерфейс комп'ютера надходить у фізичну лінію зв'язку (ЛЗ). Далі за допомогою мережевої апаратури та програмного забезпечення кадр передається спочатку всередині мережі, а потім між мережами та в інших мережах. При цьому заголовки кадра змінюються відповідно до типів ЛЗ. На рис. 2.1, для прикладу, заголовки кадрів позначено літерами E та W відповідно до різновидів мереж Ethernet та Wi-Fi на боці джерела та одержувача.

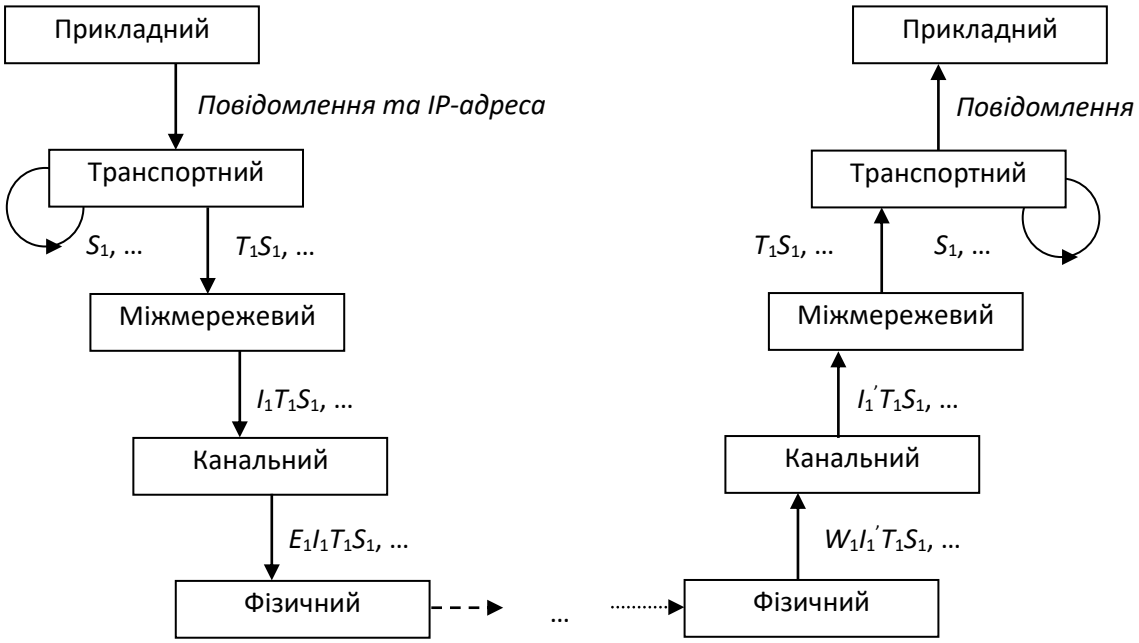


Рис. 2.1. Передача даних між рівнями взаємодії

Під час передачі даних *між мережами* кадр надходить на інтерфейс шлюзу, приєднаний до однієї з мереж, і має потрапити на інтерфейс іншої мережі. Протокол канального рівня передає протоколу IP пакет з кадру, а той за IP-адресою одержувача й своїми даними про стан мереж визначає інтерфейс у тій мережі, яка буде *наступною на шляху* до одержувача. IP-пакет передається на канальний рівень, той утворює кадр, відповідний типу ЛЗ в наступній мережі, і кадр надсилається (рис. 2.2., де `802.11` позначає заголовок кадру, відповідний мережі Wi-Fi, `Ether` – мережі Ethernet).

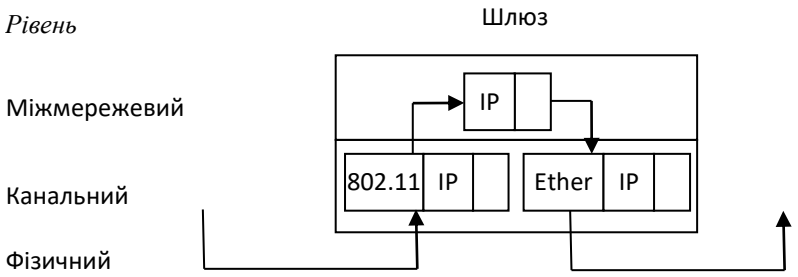


Рис. 2.2. Обробка пакетів на шлюзі

Нарешті, кадри приходять на вузол призначення, протокол канального рівня вилучає заголовки фізичного рівня (*WITS* → *ITS*) і передає IP-пакети протоколу IP. Той вилучає IP-заголовки (*ITS* → *TS*) й передає TCP-пакети (*TS*) протоколу TCP. Протокол TCP збирає TCP-пакети, витягує з них сегменти даних (*S*) і розташовує їх у правильному порядку. Коли якихось сегментів немає або вони пошкоджені, він відправляє запит відправнику передати їх ще раз. Після розміщення всіх даних у правильному порядку TCP передає утворене повідомлення процесу прикладного рівня (див. рис. 2.1., праворуч).

2.3. IP-адреси вузлів і мереж

IP-адреса – це натуральне число, що займає, залежно від версії протоколу міжмережевого рівня, 4 або 16 байт.

Протокол міжмережевої передачі даних IP створювався й удосконалювався протягом багатьох років. Відповідно, змінювалася й система адресації вузлів. Спочатку використовувалася так звана **класова адресація**, заснована на класах мереж. У цій системі адреса мала довжину 32 біт і складалася *з номера мережі та номера вузла*. Було три основні класи мереж A, B, C, у адресах вузлів яких номер мережі займав, відповідно, 1, 2, або 3 байти, а номер вузла – навпаки, 3, 2, або 1 байт.

Для того, щоб діапазони номерів мереж різних класів не перетиналися між собою, на номери мереж було накладено певні обмеження, тому мереж класу A могло бути лише 126, класу B – дещо більше ніж 16 тисяч, класу C – порядку 2 млн. Всього адрес у мережах класу A могло бути близько 2,1 млрд, класу B – порядку 1 млрд, класу C – понад 0,5 млрд.

IPv4-адреси. Адреси класів A, B, C теоретично дозволяли ідентифікувати близько 3,6 млрд вузлів, проте насправді вузлів могло бути значно менше через нераціональний розподіл номерів, зумовлений саме поділом на класи. Проблема вичерпання адрес була виявлена давно, і першим способом її пом'якшення стала так звана **безкласова адресація** (CIDR – Classless Inter-Domain Routing) в межах чотирибайтових адрес. Ця адресація використовується в нині діючій версії 4 протоколу IP, тому адреси називаються **IPv4-адресами**. Саме вони розуміються далі як IP-адреси.

- ✓ Для зручності сприйняття людиною IP-адреса зазвичай відображається четвіркою десяткових чисел від 0 до 255, що вказують значення кожного з чотирьох байтів (від старшого до молодшого), наприклад, `192.168.0.1` або `9.33.15.254`.

IP-адреса має два поля бітів – номер мережі та номер вузла в мережі. Номер мережі в адресі займає старші біти в деякій кількості від 0 до 32; решта бітів зображують номер вузла в мережі. Для того, щоб відокремлювати номери мережі та вузла в ній, необхідно знати довжину поля бітів з номером мережі. Для зручності сприйняття людиною цю довжину позначають числом від 0 до 32 після четвірки чисел для байтів. Наприклад, запис `200.170.2.251/28` вказує, що номер мережі займає 28 біт, номер вузла – останні чотири біти (вони містять «`1011`», тобто це номер 11).

Усі можливі адреси вузлів у мережі, номер якої має довжину n, утворюють **блок адрес** – 2^{32-n} номерів від `00…0` до 11…1 довжини 32–n.

В роботі з IPv4-адресами використовується **маска мережі** – 4 байти, в яких старші біти, відповідні бітам номера мережі, містять 1, решта бітів – 0. Наприклад, маска мережі для адреси `200.170.2.251/28`, наведеної вище, «для людини» позначається `255.255.255.240` (останні чотири біти «`0000`»), а маска для адреси `200.170.2.251/24` позначається `255.255.255.0` (останні вісім бітів нульові).

Описана система адресації, теоретично, допускає порядку 100 млрд адрес. Проте під час передачі в IP-пакетах маски мереж відсутні, тому доступні лише близько 4,2 млрд адрес. Це спричинило розробку способів збільшити реальну кількість адрес, якими можуть позначатися вузли, а також перехід на адреси більшої довжини.

Стандартні узгодження. Якщо номери мережі й вузла в пакеті мають вигляд 00…0, то адреса позначає вузол, що створив пакет. Адреса з номером мережі вигляду 00…0 та звичайним номером вузла позначає мережу, якій належить вузол, що створив пакет. Якщо номер вузла має вигляд 11…1 (так званий **широкомовний номер**, або **бродкаст**), то пакет має розсилатися всім вузлам мережі, номер якої задано в адресі. Якщо як номер мережі, так і номер вузла мають вигляд 11…1, то пакет має розсилатися всім вузлам мережі, якій належить вузол, що створив пакет.

З наведених узгоджень випливає:

- ✓ – пакет не може бути відправлений усім вузлам усіх мереж, тобто широкомовність обмежена однією мережею;

✓ – адреси вузлів вигляду 00…0 та 11…1 використовуються в спеціальний спосіб і не можуть бути адресами вузлів у мережі.

З останнього випливає, що максимальною довжиною маски для підмережі з вузлами є 30. Саме мережі /30 найчастіше використовуються для адресації двохточкових зв'язків між маршрутизаторами. Проте більшість сучасних маршрутизаторів, хоча й не всі, працюють і з масками /31, використовуючи адресу підмережі (останній біт 0) і бродкаст (останній біт 1) як адреси інтерфейсів вузлів. Маска /32 теж використовується – головним чином, для того, щоб вузол відправляв пакети сам собі.

Для довідки наведемо деякі спеціальні IPv4-адреси, визначені в стандартах з протоколу IPv4. Далі представлено використання деяких з них.

Мережа (адреса)	Опис	Стандарт
0.0.0.0/8	Джерело адрес поточної мережі	RFC5735
10.0.0.0/8	Для організації приватних мереж	RFC1918
100.64.0.0/10	Для використання в мережі провайдера	RFC6598
127.0.0.0/8	Інтерфейс комутації всередині хоста	RFC5735
169.254.0.0/16	Для автоматичного конфігурування (наприклад, за відсутності DHCP)	RFC3927
172.16.0.0/12	Для організації приватних мереж	RFC1918
192.0.0.0/24	Для спеціального призначення (зарезервовано IETF)	RFC5735
192.0.2.0/24	Тестова мережа 1, для використання в якості прикладів у документації	RFC5735
192.88.99.0/24	Для трансляції з IPv6 у IPv4	RFC3068
192.168.0.0/16	Для організації приватних мереж	RFC1918
198.18.0.0/15	Для тестування продуктивності	RFC2544
198.51.100.0/24	Тестова мережа 2, для використання в якості прикладів у документації	RFC5737
203.0.113.0/24	Тестова мережа 3, для використання в якості прикладів у документації	RFC5737
224.0.0.0/4	Для багатоадресної (групової) розсилки	RFC5771
240.0.0.0/4	Зарезервовано для можливих потреб у майбутньому	RFC1700
255.255.255.255	Широкомовна адреса	RFC919

Розподілом IP-адрес керує американська некомерційна організація IANA (Internet Assigned Numbers Authority – Адміністрація адресного простору Інтернет). Вона делегує свої повноваження з розподілу адрес у вигляді діапазонів по 2^{24} адрес регіональним реєстраторам, а ті надають менші діапазони інтернет-провайдерам.

Довжина адреси в IPv6. Наприкінці 1980-х років була усвідомлена проблема майбутньої нестачі IPv4-адрес. У 1992–1994 роках робоча група в IETF (Internet Engineering Task Force – Інженерна Рада Інтернету) розробила IPv6 – Інтернет-протокол наступного покоління з номером версії 6. *Адреса в IPv6 має довжину 16 байт* (128 біт), тому кількість адрес незрівнянно більша ніж в IPv4. Така довжина не лише знімає проблему дефіциту адрес, але й спрощує структуру IP-заголовків та обробку пакетів на маршрутизаторах.

Впровадження IPv6 почалося в 1996 р. і йшло дуже повільно, але з тенденцією прискорення. За даними Google[https://www.google.com/intl/en/ipv6/statistics.html], ріст частки користувачів, які з'єднуються з Google на основі IPv6, є експоненційним. Так, у січні 2014 р. ця частка була 2,5%, у січні 2015 р. – близько 6%, у січні 2016 р. – майже 10%, ще через рік – майже 15%, а влітку 2017 р. – майже 20%. За даними Facebook, станом на середину 2018 р. частка IPv6 у користувачів Facebook була найбільшою в США – більше ніж 50%, Бельгії – 49,5%, Індії – 49,36%, Німеччині – 39,2%. В Україні ця частка тоді була на два порядки менше – 0,26%.

2.4. Порти й з'єднання

Протокол транспортного рівня на боці відправника отримує повідомлення від процесу прикладного рівня, а на боці одержувача передає повідомлення певному процесу прикладного рівня. Дані між прикладними процесами та транспортними протоколами передає операційна система за допомогою спеціальних програмних каналів, які називаються портами.

Порт – це програмне поняття, аналогічне файловій змінній для файлу. Порти позначаються *двобайтовими номерами* від 1 до 65535 і призначаються операційною системою процесам прикладного рівня.

Існують стандарти призначення номерів портів протоколам прикладного рівня, наприклад, порт 25 відповідає протоколу відправки повідомлень електронної пошти SMTP (Simple Mail Transfer Protocol), 110 – протоколу прийому пошти POP3 (Post Office Protocol, version 3). Стандартизацією займається IANA (Internet Assigned Numbers Authority – Адміністрація адресного простору Інтернет). Ця американська некомерційна організація керує просторами IP-адрес, доменами верхнього рівня й проводить іншу роботу.

Порти 1–1023 зарезервовано для відомих мережевих сервісів (**системні** або **загальновідомі** порти). Порти 1024–49151 призначені для служб, зареєстрованих IANA. Порти 49152–65535 офіційно не призначені й зазвичай використовуються як тимчасові, наприклад, клієнтами для зв'язку з серверами. Номер 0 у транспортних протоколах або не використовується, або позначає відсутність порту.

Коли на боці відправника процес прикладного рівня передає дані транспортному протоколу, разом з IP-адресою одержувача він передає й номер порту, призначений потрібному прикладному процесу на боці одержувача.

Пара (IP-адреса хоста, порт) називається **сокетом** і визначає протокол прикладного рівня на хості.

Основним транспортним протоколом в Інтернеті є TCP. Він здійснює надійну доставку повідомлень завдяки *двосторонньому обміну даними* між вузлами-учасниками. Для цього перед початком обміну TCP забезпечує, що кожен з учасників має обидві пари вигляду (*адреса*, *порт*).

Структура з полями (*IP-адреса хоста1*, *порт хоста1*, *IP-адреса хоста2*, *порт хоста2*) визначає два хости та прикладні протоколи на них і називається **з'єднанням**.

Процес обміну спеціальними повідомленнями, після якого обидва учасники обміну даними створюють з'єднання й стають готовими до подальшого двостороннього обміну даними, називається **встановленням з'єднання**. Відповідно, TCP називається **протоколом, орієнтованим на з'єднання**.

На відміну від TCP, протокол UDP передає дані тільки в один бік і надає *ненадійний* сервіс: до отримувача може дійти неповна послідовність пакетів і не в порядку їх надсилання. Перед початком передачі даних UDP *не* встановлює з'єднання, тому UDP називають протоколом, **не орієнтованим на з'єднання**. Заголовок UDP-пакета також містить двобайтові номери портів відправника та одержувача, але *номер порту відправника в дійсності не використовується*.

2.5. Приватні, або локально унікальні IP-адреси

Одним з методів підвищення ефективності використання множини IP-адрес є **трансляція мережевих адрес**, NAT (Network Address Translation), описана в RFC 3022. Трансляцію адрес зазвичай здійснює **NAT-блок** (NAT box), який відстежує вхідний та вихідний трафік ЛМ. Він може бути інтегрований з маршрутизатором або перетворювачем сигналів між зовнішньою та внутрішньою лініями зв'язку.

Основна ідея трансляції мережевих адрес: присвоїти абоненту (фірмі або групі осіб) одну або декілька IP-адрес («*зовнішні*», чи абонентські адреси). Кожен вузол всередині абонентської ЛМ отримує іншу, *локальну IP-адресу, яка в межах цієї ЛМ є унікальною*, хоча водночас цю саму адресу можуть мати скільки завгодно вузлів у інших мережах. Коли пакет залишає ЛМ й проходить NAT-блок, ця локальна IP-адреса заміняється IP-адресою абонента.

Для реалізації цієї ідеї було виділено три діапазони так званих **приватних IP-адрес** (внутрішніх, локальних, або «сірих»), які використовуються тільки всередині локальних мереж:

`10.0.0.0 – 10.255.255.255/8` (16 777 216 адрес),

`172.16.0.0 – 172.31.255.255/12` (1 048 576 адрес),

`192.168.0.0 – 192.168.255.255/16` (65 536 адрес).

Пакети з цими адресами *не можуть* з'являтися в Інтернеті й використовуються лише в локальних мережах, кожна з яких, умовно, має один з номерів `10`, `172.16`, `192.168`.

Коли прикладний процес на вузлі ЛМ встановлює TCP-з'єднання з віддаленим процесом, воно зв'язується з вільним TCP-портом на вузлі (див. п. 2.4). Для трансляції локальних адрес в адреси абонента NAT-блок веде **таблицю перетворення**, кожен елемент якої містить локальну IP-адресу й номер порту. Пакет, що залишає ЛМ, проходить NAT-блок, який у IP-заголовку пакета замінює локальну IP-адресу вузла-джерела адресою абонента, а у TCP-заголовку поле *Порт джерела* – індексом того елемента таблиці, що містить локальну адресу джерела. Окрім того, переобчислюються й вставляються в пакет контрольні суми змінених заголовків TCP і IP.

Коли пакет прибуває ззовні на NAT-блок, тоді з TCP-заголовка береться значення поля *Порт одержувача*, що насправді є індексом у таблиці перетворення NAT-блоку. За елементом таблиці з цим індексом визначаються внутрішня IP-адреса й справжній порт джерела. Ці два значення вставляються в пакет, і потім наново обчислюються контрольні суми TCP і IP. Пакет передається в ЛМ для доставки за внутрішньою адресою.

Приклад. Нехай інтерфейс маршрутизатора зв'язує абонентську ЛМ з мережею провайдера й має IP-адресу `90.80.125.10`, а від порту `2048` вузла з внутрішньою адресою `192.168.0.100` надходить пакет. NAT-блок шукає у своїй таблиці перетворення ці адресу й порт і, якщо не знаходить, записує їх у таблицю. Отже, нехай ці адреса й порт записані в елемент таблиці з індексом 99. Тоді NAT-блок транслює адресу й порт так (оновлюючи контрольні суми TCP і IP).

Пакет всередині ЛМ		—>	Пакет за межами ЛМ	
Адреса джерела	Порт джерела	Індекс у таблиці NAT-блоку	Адреса джерела	Порт джерела
192.168.0.100	2048	99	90.80.125.10	99

Після цього, коли ззовні на маршрутизатор надходить пакет із адресою одержувача 90.80.125.10 та портом 99, тоді NAT-блок бере з елемента таблиці з індексом 99 локальну адресу й порт і формує пакет з локальною адресою й справжнім портом (також оновлюючи контрольні суми TCP і IP).

Пакет всередині ЛМ		<—	Пакет за межами ЛМ	
Адреса одержувача	Порт одержувача	Індекс у таблиці NAT-блоку	Адреса одержувача	Порт одержувача
192.168.0.100	2048	99	90.80.125.10	99

◄

✓ Поле *Порт джерела* має 16 біт, тому на одну справжню IP-адресу можна відобразити майже 61440 локальних адрес вузлів (перші 4096 портів зарезервовані для службових потреб).

Описана схема пом'якшує проблему нестачі IP-адрес, проте порушує деякі принципи (див. RFC 2993).

1. За принципами IP, кожна IP-адреса позначає тільки один вузол, а тут багато вузлів у різних ЛМ можуть мати одну й ту саму адресу, наприклад, 192.168.0.100.

2. За «наскрізним» принципом, кожен вузол повинен мати можливість відправити пакет будь-якому іншому вузлу будь-коли. Проте відображення адрес в NAT-блоці створюється за пакетами, що виходять з ЛМ, тому вхідні пакети не приймаються доти, доки не відправлені вихідні. Через це, користувач домашньої мережі, за допомогою NAT може створити TCP/IP-з'єднання з віддаленим сервером, але віддалений користувач не може підключитися до, наприклад, ігрового серверу в домашній мережі. Щоб це було можливим, необхідні спеціальні налаштування або технологія NAT Traversal.

3. NAT перетворює Інтернет з мережі без установлення з'єднання на щось подібне до *мережі, орієнтованої на з'єднання*. Проблема в тому, що NAT-блок має підтримувати таблицю відображення для всіх з'єднань, що проходять через нього. Запам'ятовувати стан з'єднання – справа мереж, орієнтованих на з'єднання, а не мереж без встановлення з'єднань. Якщо NAT-блок виходить з ладу й втрачається його таблиця перетворення, то нищаться всі TCP-з'єднання, що проходять через нього. Проте, якщо трансляції мережевих адрес немає, то вихід з ладу або перезавантаження маршрутизатора ніяк не впливає на TCP-з'єднання – процес-джерело просто вичікує кілька секунд і знову надсилає все непідтверджені пакети.

4. Одне з правил побудови багаторівневих протоколів: рівень *k* ніяк не взаємодіє зі змістом поля пакета на рівні *k*+1. Цей принцип визначає незалежність рівнів один від одного. Ідея багаторівневих протоколів полягає в тому, щоб зміни в одному з рівнів не могли вплинути на інші рівні. NAT руйнує цю незалежність.

5. Процеси в Інтернеті не зобов'язані використовувати TCP або UDP. Якщо хтось буде створювати новий протокол транспортного рівня, то йому доведеться боротися з тим, що NAT-блок не зможе коректно обробити поле *Порт джерела* TCP.

6. Деякі програми передбачають використання множинних TCP/IP-з'єднань або UDP-портів. Наприклад, FTP (File Transfer Protocol – стандартний протокол передачі файлів) вставляє IP-адресу в тіло пакета, щоб потім одержувач витягнув його звідти та обробив. NAT про це нічого

не знає, тому він не зможе переписати адресу або якось інакше її обробити. Звідси, FTP та деякі інші програми, такі як протокол інтернет-телефонії H.323, не зможуть працювати в умовах трансляції мережевих адрес, якщо не вжити спеціальних заходів. Можна було б покращити метод NAT і змусити його коректно працювати в окремих випадках, але ж неможливо дороблятиого для кожного нового додатку.

Попри всі недоліки, NAT є найбільш дієвим способом боротьби з дефіцитом IPv4-адрес, тому він дуже поширений у мережах організацій та домашніх мережах. NAT забезпечує міжмережеві екрани й засоби конфіденційності, оскільки за стандартних налаштувань блокує всі незапрошені вхідні пакети.

2.6. Протокол динамічної конфігурації вузла DHCP

Звідки на вузлі стає відомою його IP-адреса?

Деяким вузлам IP-адреси можуть виділятися провайдерами, після чого адміністратор мережі має відповідно налаштувати вузли за допомогою певних утиліт або засобами зі складу ОС. Проте в багатьох випадках, наприклад у «сірих» мережах (див. п. 2.5), вузли не мають фіксованих адрес, і отримують їх за спеціальними протоколами.

- ✓ Найчастіше для отримання адрес використовується протокол прикладного рівня **DHCP** (Dynamic Host Configuration Protocol – протокол динамічної настройки вузлів, див. RFC 2131, RFC 2132).

DHCP-сервер виділяє IP-адреси на деякий час, тобто в **оренду**, або **лізинг** (leasing). Перед закінченням терміну дії оренди вузол має надіслати на DHCP-сервер запит щодо продовження строку користування IP-адресою. Якщо цей запит не зроблено або в ньому відмовлено, оренда IP-адреси вузлом припиняється.

DHCP-сервер може працювати в таких режимах.

Динамічний розподіл. Адміністратор на сервері DHCP встановлює діапазон IP-адрес. Кожен вузол мережі отримує IP-адресу на правах оренди. Коли оренда припиняється, адреса стає вільною.

Автоматичне виділення. DHCP-сервер зберігає записи попередніх присвоювань IP-адрес і намагається надати вузлу ту адресу, яка давалася йому раніше.

Статичний розподіл. DHCP-сервер призначає IP-адреси тільки за таблицею MAC-адрес, яку зазвичай заповнює адміністратор мережі. Якщо MAC-адреси вузла немає в таблиці, то IP-адреса йому не буде призначена.

DHCP дозволяє отримати IP-адресу та інші параметри, потрібні для роботи в мережі (IP-адреса стандартного маршрутизатора, маска підмережі тощо). За цим протоколом інтернет-провайдери надають в тимчасове використання (оренду) «зовнішні» адреси, а в мережах підприємств і домашніх мережах розподіляються внутрішні адреси. За ним пристрої налаштовуються автоматично, тому абонентам не потрібно звертатися за даними до провайдера.

Кожен інтерфейс має вбудовану в нього *фізичну адресу* (MAC-адресу або іншу адресу канального рівня – див. п. 3.2), але під час запуску може не мати IP-адреси. Тоді він отримує IP-адресу від *DHCP-сервера*, який має бути в кожній мережі.

Виявлення DHCP-сервера. Для визначення своєї IP-адреси вузол широкомовно надсилає свою фізичну адресу в спеціальному повідомленні DHCP DISCOVER (**пакет виявлення**). У ньому в якості IP-адреси джерела записано `0.0.0.0`, адже вузол ще не має власної IP-адреси, а адресою призначення – широкомовна адреса `255.255.255.255`. Цей пакет має прибути на

DHCP-сервер. Якщо сервер не приєднаний до мережі безпосередньо, то пакет має бути ретрансльовано на DHCP-сервер незалежно від того, де він знаходиться.

Пропозиція DHCP-сервера. Отримавши пакет виявлення, DHCP-сервер виділяє вільну IP-адресу й надсилає її та свою IP-адресу вузлу в повідомленні DHCP OFFER (**пакет пропозиції**, який також може ретранслюватися). Тут вузол ідентифіковано його фізичною адресою з пакета виявлення.

У мережі може бути декілька DHCP-серверів, тому вузол може отримати кілька IP-адрес.

Запит до DHCP-сервера. Вибравши одну з адрес, запропонованих DHCP-серверами, вузол широкомовно розсилає **запит DHCP** (DHCP REQUEST). Запит містить фізичну адресу вузла та IP-адресу DHCP-сервера, вибраного клієнтом.

Підтвердження від DHCP-сервера. Отримавши запит DHCP, сервер надсилає **підтвердження** (DHCP ACK) клієнту. За даними з нього вузол налаштовує свій мережевий інтерфейс.

Контрольні запитання

Поняття: чотири рівні протоколів стеку TCP/IP; протокол (два значення); інтерфейс (два значення); протокол обробки даних; протокол керування; протоколи IP, TCP, UDP; порт; сокет; з'єднання; встановлення з'єднання; фізична адреса; IP(IPv4)-адреса; блок IP-адрес; маска мережі; приватна IP-адреса; діапазони приватних IP-адрес; NAT-блок; головне призначення протоколу DHCP; домен; доменне ім'я; система доменних імен; домен верхнього рівня; родовий домен; домен держави; доменна зона.

Що таке ARPANET?

Основні функції протоколів кожного з рівнів взаємодії в моделі DoD.

Основні відмінності властивостей протоколів TCP та UDP.

Як позначається маска мережі?

Чому перехід на IPv6 відбувається не так швидко, як це передбачали його розробники?

Які дані містить NAT-таблиця (таблиця трансляції мережевих адрес)?

Як використовуються індекси записів NAT-таблиці?

Чому під час трансляції мережевих адрес необхідна заміна порту джерела на індекс у таблиці трансляції?

Які принципи побудови Інтернет порушує використання приватних (локальних) мережевих адрес?

Чимало організацій дотримуються стратегії установки двох і більше маршрутизаторів, що з'єднують їх із провайдером. Це дає певний запас надійності, коли один з маршрутизаторів вийде з ладу. Чи можна застосувати цю політику при використанні NAT? Відповідь пояснити.

Головний успішний сценарій протоколу DHCP.

Чому в протоколі DHCP пакети транспортуються за допомогою протоколу UDP, а не TCP?

Які адреси вказуються в пакеті DISCOVER за протоколом DHCP і чому саме такі?

Опишіть загальну організацію та принципи розподілу доменних імен.

Задачі (*типові*)

Мережа задана номером та довжиною маски. Потрібно розбити її на задану кількість підмереж із заданими кількостями вузлів. Указати адреси та маски цих підмереж.

Мережа задана номером та довжиною маски. Потрібно розбити її на підмережі з заданою кількістю вузлів. Визначити кількість підмереж, їх адреси та маски.

Блок адрес задано першою адресою та кількість адрес у ньому. Потрібно виконати кілька запитів на виділення адрес; кожен запит – це задана кількість адрес. Запити розташовано в порядку спадання їхньої пріоритетності (перший найважливіший). Для кожного запиту вказати першу й останню видані адреси з довжиною маски у вигляді *w.x.y.z/s*. Потрібно задовольнити якомога більше запитів. Чи можна задовольнити всі ці запити? Скільки адрес залишиться після задоволення цих потреб?

Задано маску мережі, IP-адресу деякого вузла в цій мережі. Визначити, чи належить мережі інша задана IP-адреса.

Задано дві IP-адреси вузлів та довжини масок підмереж, яким вони належать. Визначити, чи належать вони одній і тій самій підмережі.

Задано IP-адресу деякого вузла та довжину маски мережі, якій він належить. Визначити адресу мережі та її широкомовну адресу.

Задано адресу вузла та маску мережі. Визначити, чи належить вузол мережі.

Задано мінімальні кількості вузлів, які мають бути в підмережах мережі. Визначити мінімальну кількість адрес, потрібну для цієї мережі.

Розділ 3. Передача даних у локальних мережах

3.1. Задача канального рівня

✓ *Головна задача канального рівня* – забезпечити сервіс міжмережевому рівню, а саме, передати IP-пакет від міжмережевого рівня одного вузла на міжмережевий рівень іншого вузла в *межах однієї ЛМ*. Для цього пакет «загортається в конверт» – кадр канального рівня, що містить цей пакет.

Сучасні протоколи канального рівня реалізують стандарти Ethernet, Wi-Fi (Wireless Fidelity, «безпровідна вірність»), WiMAX (Worldwide Interoperability for Microwave Access) та інші.

Протоколи канального рівня забезпечують кілька рівнів надійності сервісу.

Сервіс без підтверджень, без установки з'єднання реалізує принцип: якщо кадр втрачено, то нічого не робити. Застосовується в мережах з надійними каналами, наприклад, Ethernet, або коли дані мають передаватися без затримок (відео- або аудіодані).

Сервіс із підтвердженнями, без установки з'єднання підтверджує доставку кожного кадру. Застосовується у відносно надійних каналах, наприклад 802.11 (Wi-Fi).

Сервіс із підтвердженнями, орієнтований на з'єднання, застосовується в каналах з високою ймовірністю помилок. Спочатку встановлюється з'єднання, потім кадри нумеруються й кожний має отримати підтвердження. Приклади: супутниковий зв'язок та міжміські телефонні лінії.

У надійних каналах, наприклад, оптоволоконних, застосовується виявлення помилок і повторна передача кадрів, у ненадійних (як супутникові) – коди з виправленням помилок (Хеммінга, Ріда-Соломона та інші).

✓ *Головна проблема канального рівня* – конкуренція за канал між кадрами, які передаються з перетином у часі. Розв'язання цієї проблеми має тривалу історію, протягом якої були створені численні засоби й методи передачі. Натепер ця проблема розв'язана.

Окрім конкуренції кадрів за канал, є й інші проблеми. Ось деякі з них.

- Різні формати та різні максимальні довжини кадрів у мережах різних типів призводять до необхідности *переформатування кадрів*, а це уповільнює їх передачу.
- Мережі різних типів забезпечують різні рівні безпеки та якості обслуговування, наприклад, у мережах 802.11 (Wi-Fi) є шифрування канального рівня, а в Ethernet його немає. Звідси, коли кадр передається між різнотипними ЛМ, очікувана безпека чи якість обслуговування може й не бути дотримана.

Через зазначені причини сучасні мости (комутатори) зазвичай працюють з мережами одного типу, а мережі різних типів з'єднуються за допомогою роутерів.

3.2. Фізичні адреси, або адреси канального рівня

Фізична адреса (Hardware Address) – це унікальний числовий ідентифікатор, що присвоюється кожній одиниці активного обладнання комп'ютерних мереж у процесі її створення.

У широкомовних мережах фізична адреса дозволяє унікально ідентифікувати кожен вузол мережі й доставляти дані тільки цьому вузлу. Більшість мережевих протоколів канального рівня, хоча й не всі, використовують один з трьох просторів фізичних адрес: MAC-48, EUI-48 і

EUI-64. MAC – це Media Access Control (керування доступом до середовища), EUI – Extended Unique Identificator (розширений унікальний ідентифікатор).

Адреси типу MAC-48 найбільш поширені й використовуються в мережах Ethernet, Token ring, FDDI, WiMAX та інших. Вони складаються з 48 біт (6 байт), тому в адресному просторі MAC-48 є порядку $2*10^{14}$ адрес. За підрахунками IEEE, цього запасу має вистачити щонайменше до 2100 року.

Далі під фізичними адресами розумітимемо саме MAC-адреси. Вони були розроблені під час проектування стандарту Ethernet – кожна мережева карта або вбудований мережевий інтерфейс були повинні мати унікальний шестибайтовий номер, прошитий у процесі вироблення.

Старші 3 байти MAC-адреси – це унікальний ідентифікатор організації (Organizationally Unique Identifier, OUI), або код виробника (MFG), який виробник одержує в IEEE. Наступні три байти визначаються виробником для кожного екземпляра пристрою. Для сприйняття людиною MAC-адреси записують у вигляді шести пар шістнадцяткових цифр через дефіс, наприклад, `08-1A-37-CD-2F-E3`.

EUI-48, на відміну від MAC-48, застосовується не для мережевого обладнання, а для інших його типів. Ідентифікатори EUI-64 містять 64 біт і використовуються в FireWire, а також в IPv6 як молодші 64 біт IP-адреси вузла.

Адреси в кожному з зазначених просторів, теоретично, мають бути глобально унікальними. Проте, практично, унікальність необхідна тільки в межах кожної локальної мережі.

Розподілом адрес керує спеціальний підрозділ у складі IEEE (Institute of Electrical and Electronics Engineers – Інститут інженерів електротехніки та електроніки).

3.3. Кадр канального рівня

Протокол канального рівня, що працює на вузлі, отримує IP-пакет від протоколу міжмережевого рівня й формує **кадр канального рівня** (фрейм, або дейтаграма). Формати кадрів відповідають різноманітним типам ліній зв'язку, але їх загальна структура одна й та сама. Кадр має заголовок, дані, що передаються (це IP-пакет), та закінчення, призначене для виявлення помилок.

Для прикладу розглянемо базовий формат кадрів **Ethernet II**, найбільш поширений у мережах Ethernet. Він створений у співпраці компаній DEC, Intel і Xerox, тому його ще називають **DIX Ethernet**.

- Префікс та обмежувач початку – 7+1 байт.
- Заголовок – 14 байт:
    - MAC-адреса одержувача (DMAC-адреса, від **D**estination) – 6 байт,
    - MAC-адреса джерела, (SMAC-адреса, від **S**ource) – 6 байт,
    - Код типу Ethernet – 2 байти.
- Вкладені дані, тобто пакет мережевого рівня – від 46 до 1500 байт.
- Закінчення (код для виявлення помилок) – 4 байти.
- Проміжок між кадрами – 12 байтів, заповнених нулями.

Префікс потрібен для синхронізації роботи електричної схеми одержувача кадрів з сигналами, що надходять, і містить сім байтів спеціального вигляду. Обмежувач початку вказує, що після нього починаються вкладені дані.

SMAC-адреса відома на вузлі-відправнику, DMAC-адреса визначається за IP-адресою отримувача з IP-пакета та ARP-таблицею (див. п. 3.4) або встановлюється в спеціальне значення для відправки кадру в усі порти комутатора. Код типу зображується цілим числом. Число 1536 і

більше позначає протокол мережевого рівня, від якого отримано вкладені дані (взагалі, цим протоколом може бути не тільки IP). Число в межах до 1500 позначає довжину вкладених даних у байтах.

За специфікаціями Ethernet, довжина кадру має бути в межах від 64 до 1518 байт, тобто довжина вкладених даних – від 46 до 1500 байт. Кадр, що має довжину менше 64 байт або більше 1518 байт, вважається помилковим. Також помилковим є кадр, в якому під час перевірки на боці одержувача не збігається код для виявлення помилок, сформований під час відправлення кадру.

У каналів інших типів кадри можуть відрізнятися максимальною довжиною та закінченням, потрібним, можливо, не лише для виявлення, а й виправлення помилок (у ненадійних каналах передачі).

3.4. ARP-таблиці та протокол ARP

Кожен вузол, приєднаний до ЛМ, має унікальну фізичну адресу (MAC-адресу). IP-адресу вузол отримує від адміністратора мережі або від DHCP-сервера локальної мережі за протоколом DHCP. Відповідність IP-адрес і фізичних адрес пристроїв мережі зберігається в ARP-таблицях на кожному вузлі мережі (ARP – скорочення від Address Resolution Protocol, протокол визначення адреси).

Записи в ARP-таблицях можуть бути статичними або динамічними. Адміністратор мережі може створювати статичні записи. Інші записи є динамічними, тобто кожен запис «живе» певний час і потім видаляється. Насправді, час зберігання й метод зберігання встановлюються програмно, наприклад, операційною системою, і за бажання їх можна змінити.

Елемент ARP-таблиці має структуру, визначену в RFC 1213, у якій є поля:

- IP-адреса,
- MAC-адреса,
- тип відповідності адрес (ціле число).

Тип відповідності може бути таким: 4 – динамічна; 3 – статична; 2 – «застаріла» (запис уже не відповідає дійсності); 1 – недопустима.

✓ Зміст ARP-таблиці дозволяє побачити виклик утиліти `arp` (`arp-scan` у деяких дистрибутивах Linux): `arp -a`.

ARP-таблиці дозволяють за IP-адресами визначати фізичні адреси й навпаки. Ці задачі виникають, коли IP-пакет передається від IP на канальний рівень або в зворотному напрямку, адже IP працює тільки з IP-адресами, а протоколи канального рівня – тільки з фізичними адресами. Ці задачі розв'язуються за спеціальними протоколами.

✓ Протокол **ARP** (Address Resolution Protocol, див. RFC 826) за IP-адресою вузла визначає його MAC-адресу.

✓ Протокол **InARP** (Inverse Address Resolution Protocol, див. RFC 2390), навпаки, за MAC-адресою іншого вузла визначає його IP-адресу.

✓ Протоколи **RARP** (Reverse Address Resolution Protocol, див. RFC903), **BOOTP** (Bootstrap Protocol) та DHCP повертають IP-адресу самого вузла; тепер з них використовується тільки DHCP.

ARP є основним у мережах Ethernet, і завдяки цьому є дуже поширеним. У сім'ї протоколів IPv6 ARP немає, його функції покладено на протокол NDP (Neighbor Discovery Protocol – протокол виявлення сусідів).

Розглянемо роботу ARP на прикладі. Нехай ЛМ містить вузли з IP-адресами I_1 та I_2. Вузол I_1 має переслати пакет даних на вузол I_2. Мережа, якою вони з'єднані, *не працює* з IP-адресами, тому вузлу I_1 потрібно знати MAC-адресу вузла I_2.

Для визначення MAC-адреси протокол канального рівня на вузлі I_1 проглядає ARP-таблицю цього вузла. Якщо в ній для потрібної IP-адреси I_2 є MAC-адреса, то вона записується в кадр, що передається, інакше виконується протокол ARP.

За протоколом ARP, вузли обмінюються **ARP-повідомленнями** у вигляді ARP-запитів та ARP-відповідей. У повідомленнях обох типів записуються MAC-адреса та IP-адреса відправника (джерела), MAC-адреса та IP-адреса отримувача.

За ARP, вузол I_1 відправляє широкомовний **ARP-запит** усім вузлам ЛМ. Запит містить адреси M_1, I_1 та I_2. В якості MAC-адреси призначення записано нулі. Суть цього запиту: «*вузол з IP-адресою I_2, повідомте свою MAC-адресу вузлу з IP-адресою I_1 та MAC-адресою M_1*». Мережа доставляє цей запит усім іншим вузлам ЛМ, зокрема, й вузлу I_2.

Кожен вузол, отримавши запит, порівнює вказану в ньому IP-адресу I_2 з ввласною IP-адресою. У вузла I_2 ці адреси збігаються, тому він формує **ARP-відповідь**, в якій вказує свою IP-адресу I_2 та свою MAC-адресу M_2. Отримавши MAC-адресу M_2, протокол ARP на вузлі I_1 вносить в його ARP-таблицю пару (I_2, M_2). Після цього вузол I_1 може передавати дані через ЛМ на вузол з MAC-адресою M_2.

Але вузла з потрібною IP-адресою I_2 у ЛМ може не бути. Тоді не буде ARP-відповіді, через що у вузла-відправника в ARP-таблиці не буде запису з цією IP-адресою, і протокол рівня IP буде знищувати IP-пакети, призначені для відправки за цією адресою. Проте в багатьох реалізаціях ARP, якщо IP-адреса I_2, вказана в запиті, не належить ЛМ, то на запит з адресою I_2 відгукується зовнішній інтерфейс мережі (шлюз) і повертає свою фізичну адресу (режим проксі-ARP). Після цього пакети з IP-адресою I_2 надсилаються за межі ЛМ.

На основі ARP-запитів розв'язуються й інші задачі.

Приклади

1. Під час завантаження мережевого забезпечення ARP-запит може з'ясувати, чи не присвоєна певна IP-адреса якомусь іншому об'єкту в мережі. Це дозволяє зробити *запит самозвернення* (gratuitous ARP), в якому IP-адресою одержувача є адреса відправника. Якщо на такий запит прийде відгук, то це означає, що в мережі вже є вузол із цією IP-адресою.

2. У разі зміни свого фізичного інтерфейсу вузол може ініціювати необхідні зміни в ARP-таблицях інших вузлів. Для цього вузол видає широкомовний ARP-запит самозвернення. Якщо інший вузол, отримавши цей запит із IP-адресою, вже має її в своїй ARP-таблиці, то він оновлює запис таблиці із цією IP-адресою, вписуючи в нього нову MAC-адресу вузла-відправника.

3. Резервний файловий сервер може замінити основний, видавши запит самозвернення зі своєю MAC-адресою та IP-адресою основного сервера в якості IP-адреси відправника, наприклад, коли основний сервер вийшов з ладу. У відповідь на цей запит таблиці на всіх вузлах змінюються так, що IP-адресі основного сервера в їх записах відповідає MAC-адреса резервного сервера. Після цього кадри, адресовані основному серверу, спрямовуються на резервний.

3.5. Алгоритм роботи комутатора

За адресою в отриманому кадрі комутатор визначає, куди спрямувати кадр, виконуючи **алгоритм прозорого мосту IEEE 802.1D**. Слово «прозорий» зумовлено тим, що комутатор працює незалежно від того, чи є в мережі адаптери кінцевих вузлів, концентратори й

повторювачі, тобто так, ніби «Л3 прозорі для комутатора». У свою чергу, робота цих пристроїв теж не залежить від комутаторів.

Комутатор має **адресну таблицю** (**таблицю комутації**, **таблицю просування**), кожен елемент якої зберігає такі дані:

- MAC-адреса вузла,
- порт комутатора, зв'язаний з цією MAC-адресою,
- останній час появи кадру з цією MAC-адресою.

У таблиці можуть бути статичні записи, створені адміністратором мережі, та динамічні, що додаються, коли в комутатор надходять кадри.

Коли комутатор вмикається, у його таблиці немає динамічних записів. Коли в комутатор надходить кадр з DMAC-адресою, якої немає в таблиці, тоді кадр спрямовується в усі порти комутатора, окрім порту прибуття. При цьому в таблицю вносяться SMAC-адреса з кадру, порт прибуття й час появи. Завдяки цьому «протипотоковому навчанню» (*backward learning*), у подальшому кадр, DMAC-адреса якого є в таблиці, спрямовується тільки в порт, вказаний у таблиці, тобто в потрібний сегмент мережі.

Отже, кожен вхідний кадр обробляється в такий спосіб.

1. Якщо порт призначення для кадру невідомий, то використовується **заливка** (flooding) – кадр спрямовується в усі порти, окрім порту прибуття.

2. Якщо порт призначення відомий і відрізняється від порту прибуття, то відбувається **просування** (forwarding) кадру – кадр спрямовується в порт призначення.

3. Якщо порт призначення відомий і збігається з портом прибуття, то кадр ігнорується, тобто відбувається **фільтрація** (filtering) кадру.

Таблиця періодично (порядку хвилини) оновлюється: з неї викидаються динамічні рядки, час яких «застарів» більше ніж на певний ліміт (до кількох хвилин). Це оновлення забезпечує обмежену довжину таблиці й швидкий пошук у ній, проте, якщо протягом кількох хвилин не було кадрів для якого-небудь вузла, то наступний кадр для цього вузла буде надсилатися шляхом заливки.

У сучасних комутаторах реалізують один з трьох режимів обробки кадрів:

- комутація з проміжним зберіганням (store-and-forward);
- комутація без буферизації (cut-through);
- комутація з виключенням фрагментів (fragment-free).

Комутація з проміжним зберіганням. Комутатор, перш ніж передати кадр, повністю копіює його в буфер і перевіряє на відсутність помилок. Кадр, що містить помилки (не збігається контрольна сума або його довжина менше 64 байт або більше 1518 байт), відкидається. Кадр без помилок передається через відповідний порт пристрою призначення.

Цьому способу передачі властиві певні затримки через зберігання та перевірку помилок, проте він дозволяє працювати з портами, що підтримують різні технології та швидкості передачі (наприклад, 10/100 Мбіт/с, 1000 Мбіт/с, 10 Гбіт/с), та запобігати просуванню пошкоджених кадрів.

Комутація без буферизації. Комутатор копіює в буфер тільки DMAC-адресу (перші 6 байт після префікса) і відразу починає передавати кадр, не чекаючи його повного прийому. Затримка мінімальна, але контролю помилок немає. Цей спосіб працює, тільки якщо порти комутатора мають однакову швидкість. Саме він був реалізований у першому комутаторі Ethernet у 1990 р.

Комутація з виключенням фрагментів. Комутатор приймає в буфер перші 64 байти кадру, і це дозволяє уникнути просування занадто коротких кадрів, які є помилковими, адже довжина

кадру має бути не менше 64. Затримка передачі визначається обробкою перших 64 байтів кадру.

3.6. Петлі на канальному рівні

Створення надлишкових з'єднань у ЛМ – це звичайна практика для того, щоб мережа не «розпадалася» на незв'язані частини, коли якесь з'єднання порушено. Проте алгоритм роботи комутатора, описаний вище, породжує проблему, коли в ЛМ між комутаторами є два або більше шляхів, які, можливо, проходять через інші комутатори (**петля**, або **кільце**).

Приклад. Нехай два комутатори, К1 і К2, з'єднані двома лініями зв'язку Л1 і Л2. Від вузла А на К1 надходить кадр f0 з невідомою раніше адресою призначення. К1 застосовує заливку, тому по обох лініях Л1 і Л2 на К2 приходять дві копії кадру, відповідно, f1 і f2. Комутатору К2 теж невідома адреса призначення, тому він відправляє копії цих кадрів у всі порти, окрім тих, в які кадри надійшли. Звідси, лінією Л1 на К1 надходить копія кадру f2 – кадр f4, лінією Л2 – кадр f3, що є копією кадру f1. Комутатор К1, отримавши ці нові кадри з невідомою адресою призначення, відправляє їх копії (кадри f5 і f6). Так само робить К2, тому утворений цикл триває далі. ◄

Описане нескінченне надсилання копій завантажує лінії зв'язку настільки, що мережа стає практично непрацездатною.

Спосіб розв'язання цієї проблеми винайшла «мати Інтернету» Радья Перлман (Radia Perlman) у 1985 р., створивши **алгоритм кістякового** (**сполучного**) **дерева** (spanning tree). Ідея полягає в тому, що від реальної топології мережі залишається кореневе орієнтоване сполучне дерево, тобто деякі ЛЗ між комутаторами ігноруються (блокуються) й утворюється ациклічна топологія. Для того, щоб побудувати дерево, комутатори обмінюються даними про себе, на основі яких один з них визначається як кореневий (root – корінь). Решта комутаторів визначають найкоротші шляхи до кореня; дуги цих шляхів утворюють сполучне дерево. Усі порти комутаторів з приєднаними ЛЗ, що не належать цим шляхам, блокуються.

Алгоритм Перлман став основою **протоколу STP** (Spanning Tree Protocol – протокол сполучного дерева). Комутатори, виконуючи STP, кожні 2 с широкомовно надсилають **конфігураційні пакети** (**BPDU** – Bridge Protocol Data Units), які містять такі поля:

- **ідентифікатор**, або **код комутатора**-відправника (Bridge ID),
- **код** кореневого комутатора (Root Bridge ID),
- **код порту**, з якого відправлено пакет (Port ID),
- **вартість** маршруту до кореневого комутатора (Root Path Cost).

Код комутатора має два поля: пріоритет (Bridge Priority) та MAC-адреса (2 та 6 байт).

Вартість маршруту є сумарною вартістю окремих ліній зв'язку в маршруті. Що більше пропускна здатність лінії, то менше ця вартість, наприклад, для швидкості 10 Mbps вартість 100, 100 Mbps – вартість 19, 1 Gbps – 4.

Дії комутаторів за протоколом STP

1. Після ввімкнення в мережу кожен комутатор вважає себе кореневим, тобто в його BPDU код кореневого комутатора є його власним кодом, а вартість маршруту до нього нульова.

2. Кожен комутатор надсилає BPDU в усі порти кожні 2 с.

3. Якщо комутатор отримує BPDU, в якому код кореневого комутатора менше ніж його власний, то він припиняє генерувати свої BPDU й починає ретранслювати пакети з цим меншим кодом. Отже, зрештою в мережі залишається тільки один комутатор, який продовжує генерувати й передавати власні BPDU. Він стає **кореневим комутатором**.

4. Решта комутаторів ретранслюють BPDU кореневого комутатора, записуючи в них власний код і збільшуючи лічильник вартості шляху. Кожен комутатор, що не є кореневим, за кількістю посередників і швидкістю ліній визначає свій порт, найближчий до кореневого комутатора, – **кореневий порт**.

5. Комутатори, порти яких приєднані до одного сегмента мережі, за вартостями шляхів визначають, до якого з їх кореневих портів веде найкоротший шлях від кореневого комутатора. Цей комутатор стає **призначеним комутатором** для сегмента мережі, а його порт, приєднаний до сегмента – **призначеним портом**.

6. Після цього в сегментах, до яких приєднані два або більше портів, блокуються всі порти, окрім кореневих та призначеного.

7. Кореневий комутатор продовжує надсилати свої BPDU кожні 2 с. Протокол продовжує виконуватися, виявляючи зміни в топології та оновлюючи структуру дерева.

За протоколом STP, у порівнянні кодів комутаторів спочатку порівнюються їхні пріоритети, а за рівності – їхні MAC-адреси. Коли пріоритети рівні, меншим є код з меншою MAC-адресою, тобто, можливо, вироблений раніше і тому, вірогідно, з гіршими характеристиками. Робити застарілий пристрій кореневим комутатором не завжди бажано, тому адміністратори мережі зазвичай вручну встановлюють потрібні пріоритети комутаторів.

Протокол STP працює вже багато років; на його основі були створені кілька вдосконалень. Одним з них є протокол Rapid STP (**RSTP**), який порівняно з STP швидше збігається та є стійкішим до вимкнень комутаторів. Спеціально для використання у віртуальних ЛМ були розроблені Per-VLAN STP (**PVSTP**), Multiple STP (**MSTP**) та інші. Всі вони описані в сім'ї стандартів IEEE 802.1.

3.7. Фільтрація трафіку на канальному рівні

ЛМ забезпечує взаємодію кожного вузла з кожним. Водночас, у мережі можливі ситуації, коли така загальна доступність вузлів небажана. Наприклад, у ЛМ деякої організації є вузол з сервером, доступ до якого бажано дозволити тільки з комп'ютерів певного підрозділу організації. Це можна зробити на рівні операційної системи або системи управління базою даних самого сервера, але для надійності варто обмежити доступ ще й на рівні мережевого трафіку.

Багато моделей комутаторів дозволяють адміністраторам мереж задавати додаткові умови фільтрації кадрів разом із стандартними умовами їх фільтрації відповідно до адресної таблиці. Ці фільтри називаються **фільтрами користувача**, або **списками доступу** (access list).

Фільтр користувача призначений для створення додаткових бар'єрів на шляху кадрів, що дозволяє обмежувати доступ певних груп користувачів до окремих служб мережі. Він являє собою набір умов, які обмежують звичайну логіку передачі кадрів у комутаторі.

Найпростішими є фільтри на основі MAC-адрес вузлів, з якими працює комутатор. У таблиці комутації замість порту може бути вказана деяка операція обробки кадру, наприклад, відкидання. Тоді з комп'ютера, MAC-адреса якого вказана в цьому рядку таблиці, повністю забороняється доступ до ресурсів інших сегментів мережі.

Контрольні запитання

Поняття: головна задача канального рівня; сервіс без підтверджень і без установлення з'єднання; сервіс із підтвердженнями і без установленням з'єднання; сервіс із підтвердженнями і з установленням з'єднання; головна проблема канального рівня; фізична

адреса; складові частини кадру канального рівня; дані в заголовку кадру; ARP-таблиця та її призначення; таблиця просування комутатора та її призначення; комутація з проміжним зберіганням; комутація без буферизації; комутація з виключенням фрагментів; петля на канальному рівні.

Чому комутатори використовують для з’єднання однотипних, а не різнотипних ліній зв’язку?

Чому MAC-адреси вузлів можуть не бути унікальними?

Для чого використовується ARP-таблиця й які дані містить запис у ній?

Основна задача протоколу ARP.

Головний успішний сценарій протоколу ARP.

Що таке запит самозвернення й для чого він надсилається?

Скільки ARP-таблиць може мати комп'ютер і скільки маршрутизатор?

Указати, які з наступних тверджень правильні, а які хибні, й чому:

а) кожен комутатор має власну мережеву адресу;

б) кожен інтерфейс кожного комутатора має MAC-адресу;

в) кожен інтерфейс кожного комутатора має мережеву адресу;

г) кожен роутер має власну мережеву адресу;

д) кожен інтерфейс кожного роутера має мережеву адресу.

е) кожен інтерфейс кожного роутераі має MAC-адресу.

Які саме дані використовуються для автоматичної побудови таблиці просування комутатора?

Як комутатор обробляє кадр залежно від порту прибуття та порту призначення (якщо його задано в кадрі)?

До яких негативних наслідків призводить наявність петель в локальній мережі, побудованій на комутаторах, що працюють за алгоритмом прозорого мосту?

Призначення протоколу STP.

За якою ознакою протокол STP упорядковує комутатори для побудови дерева?

Чому динамічні записи таблиці комутації мають обмежений термін життя?

Для чого можуть бути потрібні статичні записи в таблиці комутації?

До яких наслідків може призвести недостатній обсяг таблиці комутації комутатора?

Як реалізується фільтрація кадрів у комутаторі?

Задачі (типові)

Задано схему ЛМ, яка містить комп’ютери, концентратори та комутатори з указаними на них портами та спочатку порожніми таблицями комутації. За послідовністю пар (джерело кадру, отримувач кадру) визначити, яку послідовність портів пройде кожен кадр.

Розділ 4. Передача даних між мережами

4.1. З'єднання мереж та маршрутизація

Для того, щоб користувачі різних мереж могли спілкуватися між собою, мережі необхідно з'єднати, тобто створити «інтермережу», або *інтернет*. Мережі можуть мати різні типи ліній зв'язку (Ethernet, кабелі, радіо тощо), і на кожному рівні в них можуть працювати різні протоколи. Ця різнотипність зумовлює складність передачі пакетів між мережами. Окрім того, виникають проблеми збільшення розміру інтермережі.

Коли впроваджувалися мости, передбачалося, що вони мають з'єднувати мережі різних типів, перетворюючи кадри з одного формату в інший. Проте сучасні мості (комутатори) виконують лише мінімальні перетворення кадрів, наприклад, між мережами Ethernet, які працюють з різними швидкостями. Комутатори дозволяють з'єднувати мережі лише окремих різних типів на канальному рівні, а для з'єднання різнотипних мереж і передачі даних між ними потрібні маршрутизатори. Маршрутизатор обробляє пакети з даними й визначає їх подальший шлях (маршрут) за IP-адресами, записаними в них, та даними про зв'язки в мережах.

Маршрутизація – це процес визначення маршруту передачі даних у інформаційній мережі.

Маршрутизація здійснюється за допомогою протоколів двох різновидів, які працюють на кожному вузлі мережі (хості або роутері).

Протокол маршрутизації (Routing Protocol) – це протокол, який створює дані, необхідні для маршрутизації. Він працює на вузлі мережі, взаємодіє з іншими вузлами й за певним алгоритмом динамічно формує зображення топології мережі у вигляді *таблиці маршрутизації (ТМ)*.

Маршрутизований протокол (Routed Protocol) – це протокол, який передає дані з інтерфейсу вузла в мережу та навпаки або з однієї мережі через шлюз у іншу мережу. Для цього йому потрібна ТМ, дані в якій створює протокол маршрутизації.

У сучасних мережах працюють кілька протоколів маршрутизації; деякі з них представлено в цій темі. Основним, хоча й не єдиним, маршрутизованим протоколом є IP. Формат IP-пакетів розпізнається *усіма маршрутизаторами* й дозволяє передавати пакети *практично будь-якою мережею*. Це зумовлює визначну роль IP в інтернеті. У протоколах TCP та IP, які утворюють основу сучасної мережі Інтернет, реалізована *ідея створення спільного шару* (або *рівня*), що «згладжує» відмінності існуючих мереж. Саме протоколи TCP та IP були головними в розв'язанні проблеми міжмережевого обміну, за яке В. Серф і Р. Кан у 2004 р. були удостоєні премії Тьюринга.

4.2. Формат IPv4-пакета

Одиницею обробки для протоколу IP є IP-пакет. IPv4-пакет має **заголовок** та **основну** (корисну) частину. У заголовку є обов'язкова 20-байтова частина та необов'язкова частина змінної довжини від 0 до 40 байт. Заголовок має такі поля.

Версія (4 біти) – версія протоколу, до якого належить пакет.

Довжина заголовка (4 біти) – це кількість 4-байтових слів від 5 до 15.

Диференційоване обслуговування (1 байт) – раніше називалося *Тип служби*. Це позначення **класу обслуговування**, що є певною комбінацією надійності та швидкості. Це поле задає характеристики передачі (терміновість, гарантованість та інші). Наприклад, для оцифрованого голосу швидкість доставки важливіше точності, а для файлу, навпаки, головне – передача без помилок.

Повна довжина (2 байти) – довжина всього пакета, включно з даними. Зазвичай через особливості Ethernet довжина пакета набагато менше ніж максимально можлива.

Ідентифікатор пакета (2 байти) – це номер пакета, який потрібен, коли під час передачі маршрутизатори розбивають пакет на фрагменти. Всі фрагменти одного пакета містять його ідентифікатор: він дозволяє одержувачу визначити пакет, якому належать отримані фрагменти.

Фрагментації стосуються ще три поля.

DF – біт заборони фрагментування пакета: 0 у ньому означає, що пакет можна фрагментувати, 1 – що це заборонено.

MF – біт продовження пакета: 0 у ньому позначає останній фрагмент пакета, 1 – не останній.

Ще один біт зарезервовано.

Зсув фрагмента (13 біт) – зсув фрагмента від початку пакета, вказаний у байтах.

Час життя (1 байт) – це лічильник, що обмежує час життя пакета. На практиці він враховує кількість переходів через мережі. Спочатку він отримує певне максимальне значення, а далі на кожному маршрутизаторі зменшується. Якщо він стає рівним нулю, то пакет знищується, а відправнику пакета надсилається пакет з повідомленням про це. Це знищення пакета дозволяє уникнути вічних мандрів пакетів, наприклад, через вихід з ладу таблиці маршрутизації.

Протокол (1 байт) – повідомляє, якому процесу транспортного рівня передати цей пакет (TCP, UDP або інший). Нумерація процесів стандартизована (див. RFC 1700 та www.iana.org).

Контрольна сума заголовка (2 байти) – потрібна для виявлення помилок у заголовку, що можуть виникати, коли пакет проходить мережею. Це додатковий код суми двобайтових полів. Перевірка: сума полів разом з цим значенням має давати 0. Контрольна сума заголовка обчислюється заново на кожній транзитній ділянці, оскільки поле «Час життя» щоразу змінюється.

Адреса відправника та **Адреса одержувача** (по 4 байти) – IP-адреси мережевих інтерфейсів відправника та одержувача.

Необов'язкова частина – була створена для того, щоб з появою нових варіантів протоколу не довелося вносити в заголовок поля, відсутні в цьому форматі; схоже, так і залишиться непотрібною.

✓ Маска мережі *не передається* в заголовках IP-пакетів.

4.3. Таблиця маршрутизації

На кожному маршрутизаторі та хості операційна система веде **таблицю маршрутизації** (**ТМ**). ТМ містить дані, за якими IP приймає рішення про спрямування пакетів. Кожен запис таблиці відповідає певній мережі (*мережі призначення*) й має такі поля:

- **адреса мережі** призначення – адреса інтерфейсу маршрутизатора в мережі призначення,

- **маска мережі** призначення (маска /32 або 255.255.255.255 вказує на вузол мережі),
- **адреса шлюзу** – адреса інтерфейсу маршрутизатора або хоста, на який слід передавати пакети, спрямовані в мережу призначення (це *наступний*, або *перший* шлюз на шляху до мережі призначення),
- **інтерфейс** – залежно від ОС, це може бути порядковий номер, деякий унікальний ідентифікатор або символьне ім'я інтерфейсу вузла, через який досяжний наступний шлюз (в ОС Windows інтерфейси позначаються їх IP-адресами),
- **метрика** – числовий показник якості маршруту, потрібний для обчислення найкоротших маршрутів до приймачів (що метрика менше, то краще маршрут),
- **таймер** – кількість часу з моменту надходження останнього оновлення,
- *інші поля*.

Залежно від моделі маршрутизатора та протоколів маршрутизації, у таблиці можуть бути й інші дані.

✓ У таблиці, зазвичай, зображено **стандартний**, або **основний шлюз** (default gateway). Він використовується, коли адреса вузла призначення в IP-пакеті не відповідає жодній з мереж призначення в таблиці. У деяких ОС стандартних шлюзів може бути декілька.

Приклади

1. Нехай на хості є робоча станція з Windows і з мережевим інтерфейсом, який має IP-адресу 192.168.1.101. Інтерфейс приєднаний до підмережі з локальною адресою 192.168.1.0/24 та основним шлюзом, адреса якого 192.168.1.1.

Якщо в командному рядку Windows ввести команду route print, то вона виведе на екран список інтерфейсів (номер типу, MAC-адреса, марка адаптера), далі рядки таблиці маршрутизації, після яких, можливо (залежно від версії ОС Windows), основний шлюз, тобто адресу інтерфейсу роутера, приєднаного до підмережі. Буде виведена так звана **мінімальна ТМ**, тобто приблизно таке.

```
Interface List (список інтерфейсів)
10...64 66 26 b1 5e 0a ...... TP-Link 300Mbps Wireless N Adapter
IPv4 Route Table
```

Network Destination (адреса мережі призначення)	Netmask (маска мережі)	Gateway (адреса шлюзу)	Interface (інтерфейс)	Metric (метрика)
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.101	20
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	306
192.168.1.101	255.255.255.255	127.0.0.1	127.0.0.1	276
192.168.1.0	255.255.255.0	192.168.1.101	192.168.1.101	306
192.168.1.255	255.255.255.255	192.168.1.101	192.168.1.101	276
224.0.0.0	240.0.0.0	192.168.1.101	192.168.1.101	276
255.255.255.255	255.255.255.255	192.168.1.101	192.168.1.101	306

```
Default Gateway (основний шлюз): 192.168.1.1
```

Перший рядок таблиці: адреса мережі 0.0.0.0/0 позначає «всі інші мережі, які не відповідають іншим рядкам цієї таблиці маршрутизації». Дані, призначені для цієї адреси, спрямовуються на інтерфейс робочої станції (192.168.1.101), щоб той відправив їх на основний шлюз підмережі (192.168.1.1).

Другий та третій рядки з адресами мереж призначення 127.0.0.0/8 та 192.168.1.101/32 визначають маршрут для відправки пакетів з вузла самому собі. Адреса інтерфейсу 127.0.0.1 (так звана мережа 127/8, або *loopback*, або *localhost*) – це IP-адреса, за якою комп'ютер може звернутися до самого себе взагалі без звернення до мережевих засобів. Адреса 127.0.0.1 дозволяє передавати дані для програм-серверів, які працюють на тому самому комп'ютері, що й програма-клієнт.

Четвертий рядок (підмережа 192.168.1.0/24, якій належить вузол) визначає, що пакет призначено для відправки всередині підмережі; він передається на канальний рівень на інтерфейсі вузла й на маршрутизатор може не надійти.

П’ятий рядок (широкомовна адреса 192.168.1.255/32) аналогічно попередньому визначає маршрут для широкомовної відправки пакетів всередині підмережі, якій належить ця робоча станція.

Шостий рядок з адресою 224.0.0.0/4 визначає маршрут для так званої *групової розсилки* (*мультикаст*), яка в цьому тексті не розглядається.

Сьомий рядок (широкомовна адреса 255.255.255.255/32): аналогічно четвертому й п'ятому рядкам, широкомовний IP-пакет відправляється на всі вузли локальної мережі Ethernet у складі широкомовних Ethernet-кадрів.

- ✓ Сучасні протоколи Інтернету вже практично не використовують широкомовні IP-пакети. Натомість, застосовується групова розсилка.

2. Нехай є локальна мережа, схематично зображена на рис. 4.1. У ній є два роутери, позначені кругами R_1 та R_2. Підмережі, які вони з’єднують, мають такі локальні адреси та довжини масок:

192.168.1.0/26,

192.168.1.64/27,

192.168.1.96/28,

192.168.1.128/25.

На рис. 4.1 ці підмережі позначені скорочено молодшим байтом їх адрес. Фізичні порти (інтерфейси) роутерів зображені малими чорними кружечками і мають адреси, також позначені їх молодшими байтами 97, 129, 130, 65, 1. Один з інтерфейсів роутера R_1 приєднаний до іншої мережі й має зовнішню адресу 98.11.5.46.

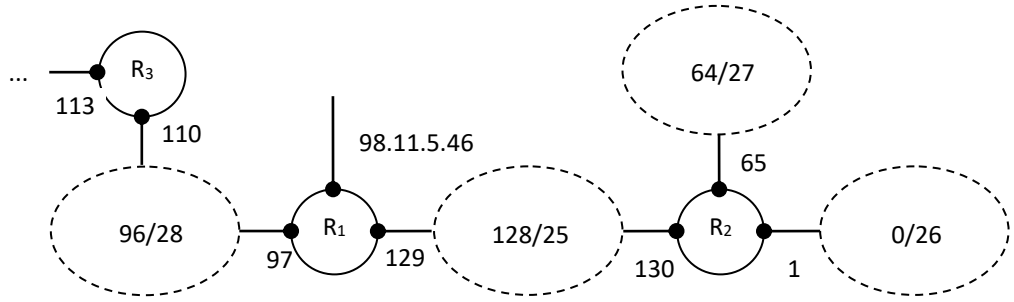


Рис. 4.1. Схема частини локальної мережі

Наведемо частину скороченого змісту ТМ роутера R_1, який утворюється через деякий час після його ввімкнення. Адреса мережі призначення вказана разом з довжиною маски, хоча, насправді, маска займає окреме поле. В якості інтерфейсу записано IP-адресу порту маршрутизатора, в якості метрики – кількість проходжень через проміжні мережі. Метрика 0 означає, що маршрутизатор приєднано до відповідної підмережі. Припустимо, що роутер R_1 через іншу мережу, до якої він приєднаний, обмінюється даними з деяким інтерфейсом з адресою 98.11.3.1.

Network Destination (адреса мережі призначення)/Mask	Gateway (адреса шлюзу)	Interface (інтерфейс)	Metric (метрика)
0.0.0.0/0	98.11.3.1	98.11.5.46	0
192.168.1.96/28	local	192.168.1.97	0
192.168.1.128/25	local	192.168.1.129	0
192.168.1.0/26	192.168.1.130	192.168.1.129	1
192.168.1.64/27	192.168.1.130	192.168.1.129	1
...			

Таблиця маршрутизації може заповнюватися в два способи – статичний та динамічний.

Статичний – записи додаються й змінюються адміністратором мережі вручну, коли встановлюється або змінюється топологія мережі.

Динамічний – записи оновлюються регулярно одним або кількома протоколами маршрутизації: OSPF, EIGRP, IS-IS та іншими. Ці протоколи реалізують той чи інший алгоритм визначення та спосіб зображення оптимальних шляхів до мереж за певним критерієм – кількість проміжних вузлів, пропускна здатність каналів, затримка передачі даних тощо. Деякі з цих протоколів представлено нижче.

4.4. Огляд протоколу IP

Функції протоколу IP визначені в стандарті RFC 791 так: «Протокол IP забезпечує передачу блоків даних (пакетів, або дейтаграм), від відправника до одержувача, де відправник і одержувач – це комп'ютери, ідентифіковані адресами фіксованої довжини (IP-адресами). Протокол IP забезпечує за необхідності також фрагментацію й збирання дейтаграм для передачі даних через мережі з малим розміром пакетів».

- ✓ **Головне завдання протоколу IP** – визначити наступний крок на шляху проходження пакета на адресу одержувача.

Протокол IP *обробляє кожен пакет як незалежну одиницю*, яка не має зв'язку з іншими пакетами. Після відправки пакета його шлях на рівні протоколу IP ніяк не контролюється. Якщо пакет не може бути доставлений, він знищується. Вузол, що знищив пакет, може відправити за зворотною адресою спеціальне повідомлення про причину знищення.

Протокол IP є *ненадійним протоколом без встановлення з'єднання*: він не підтверджує доставку даних, не контролює цілісність отриманих даних і не виконує обміну службовими повідомленнями, що підтверджують установку з'єднання з вузлом призначення та його готовність до прийому даних.

Загальний алгоритм протоколу IP

1. Можливі два випадки: отримано дані «згори» від транспортного рівня (на хості) або отримано IP-пакет від інтерфейсу нижнього рівня (на хості або роутері). Отримавши дані «згори», протокол формує пакет, записуючи в його заголовок адреси відправника та одержувача. Ці адреси разом з деякими іншими даними отримуються від протоколу транспортного рівня в спеціальному так званому **псевдозаголовку пакета**. У пакеті, отриманому «знизу», аналізуються адреса призначення в заголовку.

2. Якщо пакет призначений цьому вузлу, то дані з пакета передаються протоколу транспортного рівня, вказаному в заголовку пакета. Якщо пакет є фрагментом більшого пакета, то очікуються інші фрагменти, після чого з них збирається весь пакет.

3. Якщо пакет не спрямовано на жодну з IP-адрес цього вузла, то подальші дії залежать від того, чи дозволена ретрансляція (forwarding) «чужих» пакетів. Якщо дозволена, то за ТМ визначається *наступний вузол* мережі, якому має бути переправлено пакет, та відповідний йому інтерфейс вузла. Пакет передається цьому інтерфейсу для відправки на наступний вузол. За необхідності пакет може бути фрагментовано.

4. Якщо в пакеті є помилка або з деяких причин його не можна відправити, то він знищується. Зазвичай, відправнику пакета надсилається спеціальне повідомлення про помилку.

Правила визначення інтерфейсу, якому передається пакет у п. 3 алгоритму

Для кожного рядка ТМ визначається, чи збігається адреса мережі призначення (цільова адреса) з початком (префіксом) адреси в пакеті, який «вирізано» шляхом кон'юнкції з маскою мережі.

1. Якщо збігається цільова адреса, довжина маски якої 32, то береться інтерфейс із цього рядка.

2. Якщо кілька рядків дають збіг адреси мережі призначення з відповідними префіксами адреси в пакеті, то береться інтерфейс у тому рядку, маска мережі в якому найдовша (**правило найдовшого спільного префіксу** – longest matching prefix). Якщо таких рядків кілька, то береться інтерфейс із рядка, який містить найменшу метрику.

3. Якщо не знайдено жодного рядка, то береться інтерфейс із рядка з цільовою адресою `0.0.0.0/0` (цей інтерфейс веде до основного шлюзу).

Приклад. Нехай у мережі, зображеній на рис. 4.1, на інтерфейс `192.168.1.97` роутера R_1 надходить пакет. Якщо в ньому адреса призначення `91.202.128.77`, то жодна з адрес мереж у таблиці не збігається з початком цієї адреси, тому вибирається рядок з адресою `0.0.0.0`, і пакет передається інтерфейсу `98.11.5.46`. Якщо в пакеті цільова адреса `192.168.1.140`, то адреса мережі `192.168.1.128/25` в межах її маски збігається з цією адресою, тому пакет передається в мережу `192.168.1.128/24`. ◄

Агрегація маршрутних записів

На відміну від маршрутизаторів локальних мереж, для маршрутизаторів інтернет-провайдерів і магістралей не існує поняття «стандартний шлюз» (шлюз за умовчанням). Про них кажуть, що вони перебувають у вільній від умовчань зоні (default-free zone) Інтернету. Їхні ТМ мають зберігати адреси сотень тисяч і мільйонів мереж. ТМ переглядається при відправці кожного пакету, тому розмір ТМ має критичне значення для швидкості роботи мережі в цілому.

✓ Для зменшення розмірів ТМ застосовується заміна кількох довгих адрес мереж однією короткою – **агрегація (агрегування) адрес**, або **агрегація маршрутних записів**.

Розглянемо агрегування на прикладі локальної мережі, хоча насправді воно реалізується в магістральних маршрутизаторах.

Приклад. У мережі на рис. .1 присутні мережі з адресами `192.168.1.0/26` і `192.168.1.64/27`. Молодші байти цих адрес містять розряди, відповідно, **00**`000 0000` і **010**`0 0000` (розряди в межах маски виділено жирним шрифтом), а наступна адреса на шляху від роутера R_1 до цих мереж спільна – `192.168.1.130`. Адреси цих мереж мають спільний початок з 25 розрядів: `192.168.1.0/25`. Тоді замість двох записів у ТМ роутера R_1 можна утворити «агрегований» запис із цільовою адресою та довжиною маски `192.168.1.0/25` (у ТМ його виділено).

Network Destination (адреса мережі призначення)/Mask	Gateway (адреса шлюзу)	Interface (інтерфейс)	Metric (метрика)

0.0.0.0/0	98.11.3.1	98.11.5.46	0
192.168.1.96/28	local	192.168.1.97	0
192.168.1.128/25	local	192.168.1.129	0
192.168.1.0/25	**192.168.1.130**	**192.168.1.129**	**1**
...			

Чи не суперечать один одному записи ТМ, адже нова цільова адреса 192.168.1.0/25 є префіксом адреси 192.168.1.96/27? Наприклад, в яку мережу має йти пакет із адресою 192.168.1.100, якщо перші 25 розрядів цієї адреси збігаються з *двома* цільовими адресами? Проте адреса 192.168.1.100 з новою агрегованою адресою збігається у 25 розрядах, а з адресою 192.168.1.96/27 – у 27. *Друга довжина більше першої*, тому, за правилами визначення інтерфейсу в протоколі IP, пакет іде в мережу 192.168.1.96 через інтерфейс 192.168.1.97. Пакети ж із адресами, наприклад 192.168.1.33 та 192.168.1.77, призначені для вузлів у мережах, відповідно, 192.168.1.0/26 і 192.168.1.64/27, передаються на адресу 192.168.1.130. Далі вже роутер R_2 за своєю ТМ визначає, в яку з двох мереж надіслати кожен з цих пакетів.

Аналогічно, у ТМ на роутері R_3 (див. рис. 4.1) можна агрегувати три записи з цільовими адресами й довжинами масок 192.168.1.128/25, 192.168.1.0/26, 192.168.1.64/27, адже першим на шляху до всіх цих мереж є роутер R_1, точніше, його інтерфейс 192.168.1.97. Спільний префікс цих трьох адрес має довжину 24, тому агрегований запис має такий вигляд.

192.168.1.0/24	192.168.1.97	192.168.1.101	2

◄

4.5. Протокол ICMP

Протокол IP доставляє дані, не гарантуючи передачі. Ця ненадійність компенсується на рівні транспортних протоколів, наприклад, TCP, шляхом нумерації пакетів, підтвердження доставки та повторного надсилання даних. Протокол мережевого рівня ICMP також деякою мірою компенсує ненадійність IP, повідомляючи відправнику про втрату його пакетів.

Протокол ICMP (Internet Control Message Protocol, протокол керуючих повідомлень Інтернету) відправляє спеціальні повідомлення, коли неможливо доставити пакет або відбулася інша подія.

Приклад. Протокол IP на деякому шлюзі виявив, що пакет для подальшої передачі за маршрутом необхідно розбити на фрагменти, а в пакеті встановлено біт DF (*не розбивати*). IP не може передати пакет далі, але, перш ніж відкинути його, він має повідомити про це вузлу-джерелу. ◄

ICMP-повідомлення має такі поля:

- тип повідомлення (1 байт),
- код (1 байт),
- контрольна сума всього повідомлення (2 байти),
- поле даних (4 байти) – інтерпретація цього поля залежить від типу повідомлення.

ICMP-повідомлення передається як поле даних IP-пакета. IP-адреса вузла-джерела визначається з заголовка IP-пакета, що викликав подію. ICMP-повідомлення, яке прибуло на вузол-джерело, обробляється або ядром операційної системи, або протоколами транспортного рівня, або прикладними процесами, або ігнорується.

ICMP працює в багатьох ситуаціях, з якими зв'язано близько тридцяти типів повідомлень (ще більше десяти зарезервовані). Ось деякі з них.

Тип повідомлення	*Опис*
Адресат недосяжний	Пакет не може бути доставлений
Час вичерпано	Час життя пакета зменшився до нуля
Проблема з параметром	Помилкове поле заголовка
Переадресувати	Потрібно оновити маршрутні дані
Запит відгуку й відгук	Перевірити, чи працездатний вузол
Запит мітки часу й відповідь	Те ж, що й попереднє, але з міткою часу
Оголошення маршрутизатора / Запит до маршрутизатора	Знайти довколишній маршрутизатор

Повідомлення DESTINATION UNREACHABLE (**адресат недосяжний**) надсилається, коли маршрутизатор не може знайти пункт призначення, або коли пакет з бітом DF (*не фрагментувати*) не може бути доставлений, оскільки його шлях проходить через мережу з малим розміром пакетів.

Повідомлення TIME EXCEEDED (**час вичерпано**) потрібно, коли пакет ігнорується через те, що лічильник «час життя» в ньому зменшився до нуля. Ця подія є ознакою того, що пакети рухаються замкненим колом або що встановлено занадто низьке значення таймера.

✓ Повідомлення «час вичерпано» використовується в утилітах `ping` і `traceroute`. Наприклад, утиліта **`traceroute`** знаходить маршрутизатори, розташовані у вузлах шляху від хоста до адреси призначення. Вона відправляє на адресу призначення послідовність пакетів з часом життя 1, 2, 3 тощо. Шлюзи, на яких лічильники досягають нуля, розташовуються в тому порядку, в якому пакет проходить їх при пересиланні. Вони відправляють на хост повідомлення «час вичерпано». За цими повідомленнями хост визначає їх IP-адреси й отримує дані про шлях. Утиліта `traceroute` є дуже корисним інструментом налагоджування мережі.

Повідомлення PARAMETER PROBLEM (**проблема з параметром**) вказує, що виявлено помилкове значення в полі заголовка. Це свідчить про помилку в програмному забезпеченні хоста-відправника пакета або в проміжному шлюзі.

Повідомлення REDIRECT (**переадресувати**) надсилається хосту-відправнику пакета, коли шлюз виявляє, що пакет адресований помилково. У цей спосіб шлюз пропонує хосту оновити маршрут.

Повідомлення ECHO (**запит відгуку**) і ECHO REPLY **(відгук**) передаються, щоб визначити, чи досяжний і чи працездатний конкретний адресат. Отримавши повідомлення ECHO, хост має відправити назад повідомлення ECHO REPLY. Ці повідомлення використуються утилітою **`ping`**, яка перевіряє, чи ввімкнений хост і чи приєднаний він до Інтернету.

Повідомлення TIMESTAMP REQUEST (**запит мітки часу**) і TIMESTAMP REPLY (**відгук з міткою часу**) аналогічні попереднім, але відповідь містить час отримання повідомлення й час відправлення відповіді. Це повідомлення використовуються для вимірювання продуктивності мережі.

Повідомлення ROUTER ADVERTISEMENT (**оголошення маршрутизатора**) та ROUTER SOLICITATION (**запит до маршрутизатора**) дозволяють хостам знаходити шлюз для того, щоб він міг передавати пакети за межі локальної мережі.

Повний список повідомлень можна знайти за адресою www.iana.org/assignments/icmp-parameters.

ICMP-повідомлення генеруються не в усіх особливих ситуаціях. Наприклад, якщо втрачено пакет з ICMP-повідомленням, то новий не генерується. Також ICMP-повідомлення не створюються, коли обробляються IP-пакети з широкомовною або груповою адресою, щоб не викликати перевантаження в мережі (так званий «широкомовний шторм»).

4.6. Оновлення таблиць маршрутизації

У мережі працюють сотні тисяч маршрутизаторів, тому існує ненульова ймовірність виходу з ладу хоча б одного з них. Збій апаратного чи програмного мережевого забезпечення загрожує втратою даних під час їх передачі, оскільки дані в таблицях маршрутизації стають помилковими. На виправлення помилок потрібен певний час, протягом якого дані, що передаються, можуть бути втрачені.

Отже, структура зв'язків у мережах може непередбачувано змінюватися, тому необхідні засоби, які автоматично й оперативно приводять дані в таблицях у відповідність до поточного стану зв'язків. Цими засобами є протоколи маршрутизації. Плата за їх роботу – додаткове навантаження на мережеві пристрої.

Протоколи маршрутизації реалізують алгоритми двох основних типів: *дистанційно-векторні алгоритми* (distance vector algorithm) та *алгоритми стану зв'язків* (link state algorithm). За протоколом першого типу, маршрутизатор періодично передає всім сусіднім маршрутизаторам зміст своєї ТМ, після чого вони оновлюють свої таблиці. За протоколом другого типу маршрутизатори обмінюються даними про стан зв'язків між ними й на основі цих даних підтримують зображення спільного для всіх графа складеної мережі. За цим зображенням маршрутизатори визначають оптимальні за певним критерієм маршрути до кожної з мереж, що входять у складену мережу.

У наступних параграфах представлено алгоритми обміну даними про відстані або стан зв'язків та їх оновлення, а також протоколи, за якими маршрутизатори обмінюються даними в межах безпосередньої досяжності в локальній мережі й узгоджують свої таблиці маршрутизації.

4.7. Вектори відстаней і протокол RIP

Історично перший алгоритм маршрутизації відомий за іменами його авторів як **розподілений алгоритм Беллмана-Форда** (Bellman-Ford). Цей алгоритм є **дистанційно-векторним алгоритмом** (**ДВА**) і був застосований у мережі ARPANET.

За алгоритмом Беллмана-Форда, маршрутизатори обробляють ТМ, записи яких відповідають мережам. Кожна ТМ ініціалізується даними про сусідні маршрутизатори, а потім періодично оновлюється відповідно до повідомлень, отриманих від них. Для цього маршрутизатори виконують такі дії.

– Періодично відправляють оновлення маршрутизації до кожного сусіда. **Оновлення** – це набір повідомлень, що містять усі дані з ТМ, тобто записи для кожної мережі з відстанню до неї.

– Оновлюють відстані до мереж. Коли від сусіда R прибуває оновлення, стосовне деякої мережі N, тоді для мережі N обчислюється *можлива* нова відстань d: відстаньвід R до N, вказана в оновленні, плюс вартість лінії зв'язку, якою оновлення прийшло від R: $d = d_R(N)+d(R)$. Якщо в ТМ

наступним маршрутизатором для *N* вказано *R*, то *d* зберігається в ТМ як відстань до *N*, навіть якщо вона більше ніж поточна. Якщо *R* не є наступним маршрутизатором на шляху до *N*, то, коли *d* менше поточної відстані, тоді *d* зберігається в ТМ, а маршрутизатор *R* стає наступним на шляху до *N*.

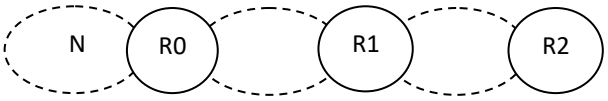
Деякі особливості дистанційно-векторного алгоритму

Встановлення в таблицях маршрутів, які дійсно відповідають найкоротшим шляхам у мережі, називається **конвергенцією** (convergence).

Конвергенція на практиці може тривати досить довго, оскільки обмін оновленнями між сусідніми маршрутизаторами ніяк не синхронізовано.

Маршрутизатори в мережі можуть з'являтися та зникати. Якщо найдовший шлях у мережі утворений *N* транзитними ділянками, то поява нового маршрутизатора стає відомою всім іншим через *N* обмінів. Зникнення маршрутизатора може створити *проблему «нескінченного обчислення»* (count-to-infinity). Головна причина – коли маршрутизатор повідомляє про те, що в нього є якийсь шлях, він сам *не гарантує достовірності* цих даних.

Приклад. Нехай до мережі N приєднано маршрутизатор R0, маршрутизатори R0 і R1, R1 і R2 є сусідами, причому R2 надсилає пакети у мережу N через R1, тому в ТМ маршрутизатора R2



записана відстань 2 до R0, а R1 встановлено як перший маршрутизатор на шляху до N..

Припустимо, що R0 виходить з ладу. R1 швидко це виявляє й відмічає в своїй ТМ. Нехай обмін оновлень має період 30 сек. Протягом цього часу цілком може статися, що R2 надсилає оновлення маршрутизатору R1. Тоді R1 оновлює відстань до N як 3=2+1 й зберігає її, а першим шлюзом на шляху до N у своїй ТМ встановлює R2. Так виникає маршрутна петля: пакети від R1 до R0 починають циркулювати між R1 і R2, поки не вичерпається їх час життя. Більш того, обмін відстанями, що збільшуються, продовжується в оновленнях між R1 і R2, тому ці недостовірні дані розходяться по інших сусідах, сусідах їх сусідів тощо. Лише тоді, коли відстань дійде до 16 (у маршрутизатора R2), він прийме рішення, що шляху до R0 немає. За конкретних параметрів мережі (період 30 сек і максимальна відстань 15) помилковий стан даних про шляхи між мережами триває кілька десятків хвилин. ◄

Проблема з петлею, що утворюється між сусідніми маршрутизаторами, розв'язується за допомогою **методу розщеплення горизонту** (split horizon) – маршрутні дані про деяку мережу, присутні в ТМ, ніколи не передаються маршрутизатору, від якого вони отримані. Цей метод реалізований в усіх дистанційно-векторних протоколах.

Проте можливі петлі, утворені більш ніж двома маршрутизаторами, – *складені петлі*. Для того, щоб їх уникати, є кілька методів. Розглянемо так звані тригерні оновлення та заморожування змін.

Тригерні оновлення: маршрутизатор, отримавши дані про зміну метрики до якої-небудь мережі, не чекає закінчення періоду передачі таблиці маршрутизації, а передає дані про зміну маршруту негайно. Це в багатьох випадках запобігає передачі застарілої інформації про пошкоджені маршрути, але завантажує мережу, тому тригерні оновлення також виконуються з деякою затримкою. Тоді також можлива ситуація, коли регулярне оновлення в якому-небудь маршрутизаторі трохи випереджає в часі тригерне оновлення від маршрутизатора, попереднього в ланцюжку, і маршрутизатор встигає передати по мережі застарілі дані, наприклад, про маршрут, якого насправді вже немає.

Заморожування змін дозволяє позбутися ситуацій, описаних вище. Припустимо, мережа стає недоступною через те, що приєднаний до неї маршрутизатор вийшов з ладу. Тоді маршрутизатор-сусід виявляє це й на певний час припиняє приймати дані про цю мережу від

інших маршрутизаторів, що лежать на деякій відстані від маршрутизатора, що вийшов з ладу, оскільки ці дані недійсні. Відповідно, ці «віддалені» маршрутизатори не отримують даних про мережу, яка стала недоступною, через певний час викидають запис про неї зі своїх таблиць і далі не поширюють застарілі дані.

На основі алгоритму Беллмана-Форда був розроблений протокол маршрутизації, який свого часу широко використовувався, – **протокол RIP** (Routing Information Protocol, RFC 2453). У мережах із класовою адресацією працювала його версія RIPv1, у мережах без класів – RIPv2 (див. п. 2.4). У RIP відстанню є кількість **хопів** – «стрибків», тобто кількість ділянок між сусідніми маршрутизаторами, обмежена числом 15. Оновлення надсилаються кожні 30 с. RIP є протоколом прикладного рівня; для передачі даних він використовує транспортний протокол UDP. Регулярний і частий обмін повідомленнями оновлення значно збільшує трафік мережі, тому RIP має проблеми масштабування й придатний тільки в невеликих локальних мережах. Він уже давно не використовується. Для IPv6 було розроблено протокол **RIPng** (Next-Generation RIP), який використовує такі самі типи повідомлень, відстань та інші параметри, що й RIP, але не підтримує IPv4.

4.8. Стисло про IGRP та EIGRP

Окрім протоколів сім'ї RIP, є й інші протоколи на основі ДВА. Зокрема, американська компанія CISCO свого часу розробила протокол **IGRP** (Interior Gateway Routing Protocol) та його покращення **EIGRP** (Enhanced Interior Gateway Routing Protocol), яке є дуже популярним, але працює тільки на маршрутизаторах CISCO. Протокол IGRP є дистанційно-векторним, EIGRP – гібридним, обробляючи як вектори відстаней, так і стани зв'язків.

Протокол IGRP переважає RIP у багатьох аспектах. Він оновлює дані в ТМ кожні 90 секунд, тому навантажує мережі значно менше ніж RIP. Він підтримує відстань до 100 хопів, тому застосовний у мережах більших розмірів. Він використовує 24-бітові значення метрики маршруту, яка узагальнює поняття відстані й дозволяє краще вибирати шляхи призначення. У обчисленні метрики маршруту можуть використовуватися від двох до п'яти з таких компонентів:

– мінімальна пропускна здатність маршруту (bandwidth),

– сумарна затримка на всьому маршруті (delay),

– найгірший показник надійності на маршруті (reliability),

– навантаження (load),

– величина MTU (Maximum transmission unit) – максимальна довжина даних, які можна передати протоколом за одну ітерацію, наприклад, для Ethernet це 1500.

IGRP був повністю замінений протоколом EIGRP, який порівняно з IGRP забезпечує швидшу конвергенцію таблиць маршрутизації до узгодженого стану, набагато менше навантажує мережу, добре масштабується на мережі великих розмірів і підтримує IPv6. EIGRP підтримує відстані до 255 хопів та маски змінної довжини, використовує 32-бітові значення метрики шляхів. Конвергенція з EIGRP відбувається набагато швидше ніж із IGRP завдяки алгоритму розповсюдження оновлень Diffusing Update ALgorithm – DUAL. Завдяки DUAL у EIGRP немає зациклення маршрутів.

Протокол EIGRP складається з чотирьох основних компонентів:

- виявлення/відновлення сусідів (Neighbor Discovery/Recovery),
- надійний транспортний протокол RTP (Reliable Transport Protocol),

- Блок станів алгоритму DUAL (DUAL Finite State Machine),
- Модулі, залежні від протоколів (Protocol Dependent Modules).

Виявлення/відновлення сусіда – це процес, в якому маршрутизатор динамічно розпізнає інші маршрутизатори в мережах, до яких вони підключені, а також відсутність доступу до сусіда або припинення його роботи. Для цього маршрутизатор кожні 5 с розсилає дуже малі пакети Hello, які створюють незначний додатковий трафік. Отримуючи пакети Hello, маршрутизатор визначає, що його сусід функціонує нормально й може обмінюватися маршрутними даними.

Надійний транспортний протокол RTP забезпечує гарантовану й упорядковану доставку пакетів EIGRP всім сусідам, хоча для підвищення ефективності надійність надається тільки в разі потреби. Наприклад, пакет Hello надсилається всім сусідам із зазначенням, записаним у пакеті, що прийняття цього пакета не потрібно підтверджувати. Проте пакет Update (оновлення) вимагає підтвердження отримання, і про це вказується в пакеті.

Блок станів алгоритму DUAL реалізує процес прийняття рішень для обчислення маршрутів. Для того, щоб визначати нові маршрути до адресатів, що можуть зникнути, блок відстежує всі маршрути, оголошені всіма сусідами. Дистанційні дані оновлюються не періодично, а тільки тоді, коли маршрутизатор виявляє, що деякий адресат стає недоступним. Тоді маршрутизатор шукає серед сусідів **наступний елемент** – сусід, який має найкоротший шлях до зниклого одержувача, за умови, що цей шлях не є частиною циклу маршрутизації. Коли ймовірних наступних елементів немає, але є сусіди, що мали маршрути до пункту призначення, тоді проводяться переобчислення дистанційних даних і визначається новий наступний елемент. Час переобчислень впливає на загальний час конвергенції.

Модулі, залежні від протоколів, відповідають за міжмережевий рівень і «приймають на себе» вимоги специфічних протоколів. Наприклад, модуль IP-EIGRP відповідає за відправку та отримання пакетів EIGRP, інкапсульованих у протокол IP. Цей модуль відповідає за розбиття на компоненти пакетів EIGRP та повідомлення алгоритму DUAL про отримання нових даних. Він також запитує алгоритм DUAL про прийняття рішень з маршрутизації, результати яких зберігаються в IP-таблиці маршрутизації.

4.9. Алгоритм стану зв’язків і схема протоколу OSPF

Інший підхід до формування даних у таблицях маршрутизації, за якими визначається спрямування пакетів, реалізовано в алгоритмах стану зв’язків.

Алгоритм стану зв’язків (Link-State Algorythm) забезпечує кожен маршрутизатор даними, достатніми для побудови графа зв’язків мережі. Для цього маршрутизатори обмінюються даними про стан зв’язків між ними й на основі цих даних підтримують зображення спільного для всіх графа складеної мережі. За цим зображенням маршрутизатори визначають оптимальні за певним критерієм маршрути до кожної з мереж, що входять в складену мережу.

За алгоритмом, маршрутизатор періодично з’ясовує стан ліній зв’язку, приєднаних до його портів, надсилаючи короткі пакети. Об’ємні повідомлення з даними про топологію – **оголошення про стан зв'язків** (Link State Advertisement, **LSA**) мережі – передаються тільки після того, як встановлено факт зміни стану якогось із зв’язків. Звідси, протоколи на основі LSA порівняно з протоколами на основі DVA створюють набагато менший службовий трафік.

На відміну від дистанційно-векторних алгоритмів, в алгоритмах стану зв’язків під час передачі топологічні дані не змінюються. Звідси, всі маршрутизатори мережі зберігають однакові дані про поточну конфігурацію графа зв'язків мережі.

Протокол OSPF (Open Shortest Path First – «спершу найкоротший маршрут») був створений у 1988 р. в IETF (Internet Engineering Task Force) і визнаний стандартом в 1990 р (див. RFC 2328). Його ідея запозичена з протоколу IS-IS (Intermediate System to Intermediate System – зв'язок між проміжними системами), що став стандартом ISO. Ці два протоколи підтримуються численними виробниками маршрутизаторів. OSPF зазвичай використовується в корпоративних мережах, IS-IS – в мережах інтернет-провайдерів.

OSPF використовується як у окремих мережах, так і в автономних системах, зазвичай, складених з окремих областей (див. п. 4.10). Для кожної області визначається окремий граф, і топологія області невідома за її межами. Завдяки цьому обсяг службового трафика протоколу значно менше ніж коли розглядається вся АС цілком.

OSPF є відкритим, враховує кілька параметрів ліній зв'язку (фізичну відстань, затримку тощо), автоматично й швидко адаптується до змін топології. Він підтримує вибір маршрутів на основі типу сервісу, тобто по-різному вибирає маршрути для трафіку реального часу та інших його різновидів. Він вміє розподіляти навантаження на декілька ліній – у багатьох ситуаціях це дає кращу продуктивність. Він надає певний рівень безпеки й підтримує тунельні з'єднання (попередні протоколи цього практично не забезпечували).

OSPF розв'язує дві задачі.

1. **Побудова та підтримка бази даних про стан зв'язків мережі**. Зв'язки мережі зображуються графом, вершинами якого є маршрутизатори, їх інтерфейси та підмережі. Залежно від типів мереж, до яких приєднані маршрутизатори, ребро зв'язує маршрутизатор і підмережу, до якої він приєднаний, або два маршрутизатори, приєднані до спільної мережі, або маршрутизатор та інтерфейс іншого маршрутизатора. Докладніше див. RFC 2328.

2. **Визначення оптимальних маршрутів і заповнення таблиці маршрутизації**. Кожен маршрутизатор шукає оптимальні маршрути від своїх інтерфейсів до всіх відомих йому підмереж за допомогою алгоритму Дейкстри. У знайденому маршруті запам'ятовується перший крок – до наступного маршрутизатора. Дані про цей крок записуються в таблицю маршрутизації.

За алгоритмом OSPF кожен маршрутизатор:

– виявляє своїх сусідів і контролює їх працездатність;

– визначає метрику відстані або вартості зв'язку з кожним із своїх сусідів;

– створює **пакет стану зв'язків** (Link State Advertisement, **LSA**), що містить усі зібрані дані;

– надсилає пакети стану зв'язків маршрутизаторам-сусідам і приймає пакети від них;

– обчислює найкоротші шляхи до інших маршрутизаторів.

Розглянемо описані дії докладніше.

Побудова та підтримка бази даних про стан зв'язків мережі

Виявлення та контроль працездатності сусідів

Для контролю стану зв'язків та сусідніх маршрутизаторів OSPF-маршрутизатори посилають короткі пакети HELLO по всіх лініях зв'язку кожні 10 секунд.

Коли маршрутизатор завантажується, його перша задача – отримати дані про сусідні маршрутизатори (neighboring routers), приєднані до спільних з ним мереж. Він посилає пакети HELLO в усі лінії зв'язку. Маршрутизатори на інших кінцях ліній мають надіслати у відповідь пакети, що містять їхні адреси та оголошення стану зв'язків (LSA).

Якщо маршрутизатори приєднані до широкомовної ЛМ, наприклад, Ethernet, то всі маршрутизатори в цій ЛМ будуть сусідніми. Це може призвести до занадто великої кількості

LSA. Щоб запобігти цьому, вся ЛМ вважається окремим вузлом, **суміжним** (adjacent) з усіма іншими. Роль цього вузла в протоколі маршрутизації відіграє один з маршрутизаторів ЛМ – **призначений маршрутизатор** (designated router, DR), а також ще один – **резервний призначений** (backup designated router, BDR). Всі маршрутизатори мережі вважаються суміжними з обома призначеними. Вузли обмінюються даними з вузлами, *суміжними до них, а не сусідніми*. Відношення суміжності й сусідства – це *різні відношення*.

У ситуації, коли пакети HELLO перестають надходити на маршрутизатор від суміжного йому, маршрутизатор робить висновок, що зв'язок став непрацездатним, і корегує свої топологічні дані. Він відразу розсилає всім суміжним маршрутизаторам LSA з цими змінами, після чого вони виправляють свої бази даних і розсилають цей LSA своїм «суміжникам», і так далі.

Встановлення вартості зв'язку

Алгоритм вимагає, щоб кожна лінія зв'язку мала метрику відстані або вартості, необхідну для обчислення найкоротшого шляху. Вартість шляху до сусідніх маршрутизаторів може бути визначена автоматично або задана адміністратором мережі. Зазвичай, вартість тим менше, чим вищу пропускну здатність має лінія зв'язку. Наприклад, мережа Ethernet зі швидкістю 1 Гбіт/с може мати вартість 2, а зі швидкістю 100 Мбіт/с – вартість 10.

Створення LSA

Після того як дані про суміжні вузли зібрано в маршрутизаторі, він створює пакет із цими даними (LSA). Для мереж ці пакети створює призначений маршрутизатор. Взагалі, LSA створюються періодично кожні 30 хвилин або коли відбувається якась подія, наприклад, лінія зв'язку або суміжний маршрутизатор виходить з ладу, або з'являється в мережі, або змінює свої властивості.

LSA містить:

- ідентифікатор (адреса) відправника,
- порядковий номер пакета,
- часовий ліміт життя пакета,
- список маршрутизаторів або мереж, суміжних з відправником, та вартостей зв'язків з ними.

Порядковий номер та часовий ліміт життя пакета розглядаються в наступному пункті.

Розповсюдження LSA

Кожен маршрутизатор веде **таблицю пакетів стану зв'язків**. Кожен запис таблиці відповідає вже отриманому, але ще не повністю обробленому пакету. Запис містить такі дані:

- адреса відправника,
- порядковий номер пакета,
- часовий ліміт життя пакета,
- прапорці розсилки й підтверджень для кожної лінії маршрутизатора,
- список маршрутизаторів, суміжних з даним, та вартостей зв'язків з ними.

Пакети стану зв'язків розповсюджуються на основі заливки. У кожен пакет відправник вставляє **порядковий номер**, що збільшується на 1 з кожним наступним пакетом. Коли приходить новий LSA, маршрутизатор шукає адресу його відправника та порядковий номер пакета в своєму списку. Якщо це новий пакет, то він розсилається далі по всіх лініях, крім тієї, якою прийшов. Якщо порядковий номер прибулого пакета не більше, ніж номер вже отриманого пакета від того ж відправника, то пакет видаляється як дублікат або як застарілий.

Описаний спосіб розповсюдження LSA має кілька проблем, але вони є розв'язуваними.

1. Якщо послідовний номер, досягнувши максимально можливого значення, стане нулем через переповнення, то виникне плутанина. Проте порядкові номери є 4-байтовими: якщо розсилати пакети кожні 10 с, то переповнення настане через 1370 років.

2. Якщо маршрутизатор виходить з ладу, то втрачається поточний порядковий номер його пакетів. Після його перезавантаження цей номер буде 0, тому наступний відправлений ним пакет буде проігнорований як застарілий. Також порядковий номер може бути зіпсований при передачі, наприклад, через помилку в одному біті замість номера 2 буде прийнято число 32770 – тоді пакети з номерами з 3 по 32770 будуть ігноруватися маршрутизаторами як застарілі.

Ці проблеми вирішуються тим, що пакет містить поле **ліміт життя пакета** (наприклад, 60 секунд). У таблиці ліміт щосекунди зменшується на 1. Коли він зменшується до нуля, дані від цього маршрутизатора видаляються. Зазвичай, пакет HELLO від маршрутизатора надходить кожні 10 с. Звідси, відомості про маршрутизатор видаляються, якщо він вимкнений або було втрачено шість пакетів стану зв'язків підряд, що малоймовірно. Це гарантує, що через деякий час після виникнення помилки нові пакети не сприймаються як застарілі.

Для підвищення надійності алгоритм має певні вдосконалення. Коли LSA приходить на маршрутизатор для розповсюдження, він відправляється не відразу, а зберігається протягом певного часу в області проміжного зберігання на випадок появи нових зв'язків або розриву старих. Якщо за цей час від того ж відправника встигає прийти ще один пакет, маршрутизатор порівнює їх порядкові номери, і старіший пакет видаляється.

Для захисту від помилок на лініях зв'язку факти одержання пакетів стану зв'язків підтверджуються. **Прапорці розсилки** відповідають лініям зв'язку й означають, що пакет слід відіслати у відповідну лінію зв'язку. **Прапорці підтверджень** означають, що потрібно підтвердити отримання цього пакета з цієї лінії.

Визначення оптимальних маршрутів і заповнення таблиці маршрутизації

Зібравши повний комплект пакетів стану зв'язків, маршрутизатор може побудувати граф мережі. Насправді, кожна лінія зв'язку зображена двічі, по одному значенню для кожного напрямку. У різних напрямків може бути різна вартість, тому найкоротші шляхи від *A* до *B* і від *B* до *A* можуть не збігатися.

Кожен маршрутизатор на основі алгоритму Дейкстри будує дерево найкоротших шляхів від себе до всіх можливих адресатів. Результат обчислень каже маршрутизатору, яке з'єднання слід вибрати, щоб дістатися до потрібної адреси призначення. Ці дані додаються в таблицю маршрутизації, після чого поновлюється нормальна робота маршрутизатора.

- ✓ Порівняно з дистанційно-векторними протоколами, OSPF вимагає більших потужності процесора та об'єму пам'яті маршрутизатора, проте забезпечує швидшу конвергенцію таблиць маршрутизації до узгодженого стану й менше навантажує мережу.

4.10. Протокол зовнішнього шлюзу BGP

Світова мережа, утворена сукупністю існуючих мереж, має ієрархічну структуру й складові частини різних рівнів. Одну з таких складових частин і основу для об'єднання мереж в Інтернеті утворюють автономні системи.

Автономна система (**АС**) – це система мереж і шлюзів, керованих одним оператором або кількома, якщо вони мають спільну політику маршрутизації, тобто спільні правила передачі даних на маршрутизаторах.

Автономні системи позначаються унікальними номерами, які присвоює IANA (Internet Assigned Numbers Authority – Адміністрація адресного простору Інтернет) – американська некомерційна організація, що керує просторами IP-адрес, доменами верхнього рівня й проводить іншу роботу. До 2007 р. номери мали довжину 16 біт, з 2007 р. діє стандарт на 32-бітові номери AS RFC 4893. За кожною АС закріплено неперервний блок IP-адрес у вигляді IP-префікса. Станом на початок 2017 р. було зареєстровано більше 56 тисяч АС.

Наявність АС спрощує розв'язання задачі визначення маршрутів передачі даних. Задача розв'язується у двох рівнях – на одному рівні шукається найкращий маршрут між автономними системами, на іншому – маршрут всередині кожної АС. Протоколи маршрутизації всередині АС називаються протоколами внутрішнього шлюзу (Interior Gateway Protocol), або ПВШ, наприклад, OSPF чи EIGRP. У кожній АС може працювати будь-який ПВШ, хоча поширені лише декілька.

Між АС дані передаються **прикордонними маршрутизаторами автономних систем** (AS boundary router). Маршрутні дані між АС поширюються за протоколом зовнішнього шлюзу (exterior gateway protocol) – **ПЗШ**. В Інтернеті цим ПЗШ є **BGP** (Border Gateway Protocol – протокол прикордонного шлюзу), описаний у RFC 4271. Для надійності та безпечності передачі маршрутних даних у BGP-сеансі використовується TCP та автентифікація маршрутизаторів.

Маршрут у BGP – це послідовність номерів АС на шляху до мережі з заданою адресою. Розглянемо роботу BGP за допомогою прикладу.

Приклад. Нехай є три автономні системи, позначені AS_1, AS_2 і AS_3, та в них маршрутизатори EG_1–EG_6, які працюють як зовнішні шлюзи (External Gateway) й виконують протокол BGP (рис. 4.3).

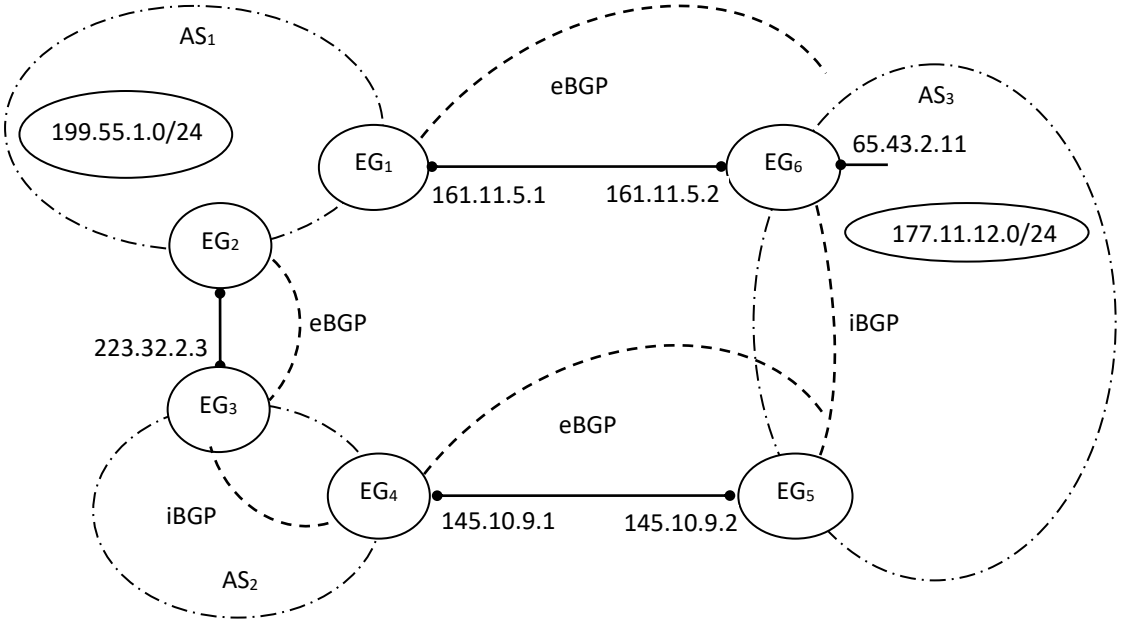


Рис. 4.3. Автономні системи та прикордонні маршрутизатори

✓ Маршрутизатор взаємодіє з іншими маршрутизаторами за протоколом BGP тільки тоді, коли в його конфігурації вони **призначені як сусіди** адміністратором АС.

Наприклад, шлюз EG_1 взаємодіє з EG_6 не тому, що між ними є двоточковий канал, а тому, що EG_6 призначений як сусід у конфігурації EG_1, а EG_1 – у конфігурації EG_6.

Основним повідомленням протоколу BGP є повідомлення UPDATE (оновити), яким шлюз повідомляє шлюзам сусідньої АС про досяжність мереж в його власній АС. Це повідомлення є *тригерним*, тобто надсилається тільки тоді, коли в автономній системі що-небудь змінюється: з'являються нові мережі або нові шляхи до мереж, чи навпаки, існуючі мережі або шляхи зникають.

В одному повідомленні UPDATE оголошується один новий маршрут або анулюються кілька маршрутів, що припинили існування. Дані про новий маршрут записуються рядком такого вигляду.

BGP Route = *AS_Path*; *NextHop*; *Network/Mask_length*;

Тут *AS_Path* – список номерів автономних систем, *NextHop* – IP-адреса шлюзу, через який потрібно передавати пакети в мережу *Network/Mask_length*. Наприклад, шлюз EG_1 повідомляє шлюзу EG_6 у AS_3, що в AS_1 з'явилася нова мережа 199.55.1.0/24, таким рядком.

AS_1; 161.11.5.1; 199.55.1.0/24;

Шлюз EG_6, отримавши повідомлення UPDATE, зберігає в своїй ТМ дані про мережу 199.55.1.0/24, адресу наступного шлюзу 161.11.5.1 і відмітку про те, що ці дані отримані за BGP. Якщо у EG_6 встановлено режим перерозподілу маршрутів BGP у маршрути, наприклад, протоколу OSPF, то за цим протоколом EG_6 передає нові маршрутні дані внутрішнім шлюзам у AS_3. Після цього на всіх внутрішніх шлюзах в AS_3 стає відомо про існування мережі 199.55.1.0/24. В якості адреси наступного шлюзу EG_6 вказує адресу свого внутрішнього інтерфейсу 65.43.2.11 (див. рис. 4.3).

Формат даних про мережі у BGP відрізняється від формату даних протоколів маршрутизації всередині АС, зокрема, дані про мережу містять номер АС, якій належить мережа. Тому повідомлення про мережі для інших зовнішніх шлюзів всередині АС також передаються за протоколом BGP. Наприклад, для того, щоб шлюз EG_5 отримав повідомлення UPDATE про мережу 199.55.1.0/24 потрібного формату, шлюзи EG_6 і EG_5 встановлюють між собою BGP-сеанс. Ця реалізація протоколу BGP називається **внутрішньою** (Interior BGP, iBGP), на відміну від основної, **зовнішньої** (Exterior BGP, eBGP). В результаті BGP-сеансу шлюз EG_5 отримує потрібні дані від EG_6, після чого може передати їх зовнішньому сусідові EG_4.

На основі даних, отриманих від EG_6, шлюз EG_5 формує нове повідомлення: на початку списку АС він додає власну AS_3, а в якості адреси наступного шлюзу вказує адресу свого інтерфейсу.

AS_3, AS_1; 145.10.9.2; 199.55.1.0/24;

Номери АС у повідомленнях UPDATE запобігають їх зацикленню. Наприклад, EG_3 передає EG_2 таке повідомлення про мережу 199.55.1.0/24.

AS_2, AS_3, AS_1; 223.32.2.3; 199.55.1.0/24;

Проте EG_2 його відкине, оскільки в списку АС вже є номер AS_1 його власної АС.

4.11. З'єднання різнотипних мереж

Окрім IP, існують інші протоколи міжмережевого рівня, наприклад, IPX, SNA або AppleTalk. Для сумісної роботи різних протоколів необхідні **мультипротокольні маршрутизатори** (multiprotocol router). Вони мають *перетворювати дані між протоколами* або *забезпечувати з'єднання на рівні протоколу транспортного рівня*. Обидва ці способи мають недоліки.

Перетворення даних між протоколами різних мереж. Якщо пакети різних форматів мають різні інформаційні поля, то правильне їх перетворення практично неможливе. Наприклад, IPv6-адреса має довжину 128 біт і не може вміститися в 32 біти адреси IPv4. Через це, до речі,

проблема використання IPv4 і IPv6 в одній мережі є головною перешкодою на шляху впровадження IPv6.

З'єднання на транспортному рівні, наприклад за допомогою TCP, припускає, що TCP реалізовано у всіх мережах, але це не так. Окрім того, тоді з мережею можуть працювати тільки програми, що використовують TCP, проте багатьом програмам реального часу потрібні інші протоколи, наприклад UDP.

Приклад. Розглянемо взаємодію різнорідних мереж у спільному мережевому шарі. На рис. 4.4 зображено інтермережу, складену з мереж 802.11 (Wi-Fi), MPLS і Ethernet. Нехай передавач у мережі 802.11 надсилає пакет приймачу, приєднаному до мережі Ethernet. Пакет має пройти мережами трьох різних типів, тому на їх кордонах він додатково обробляється.

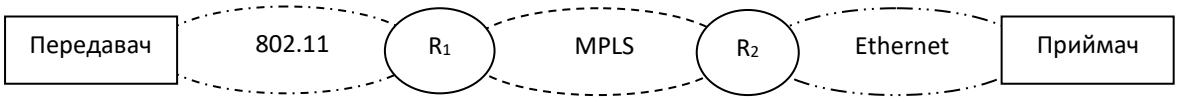


Рис. 4.4. Інтермережа з трьох мереж

Протокол IP на хості передавача отримує дані від протоколу транспортного рівня й створює IP-пакет, заголовок якого містить IP-адресу приймача. За IP-адресою та ТМ на хості протокол IP визначає, що пакет потрібно відправити через маршрутизатор R_1, розташований на кордоні мереж 802.11 і MPLS. Пакет передається на канальний рівень, де вставляється в кадр 802.11 з MAC-адресою маршрутизатора R_1. Цей кадр передається на R_1 (рис. 4.5).

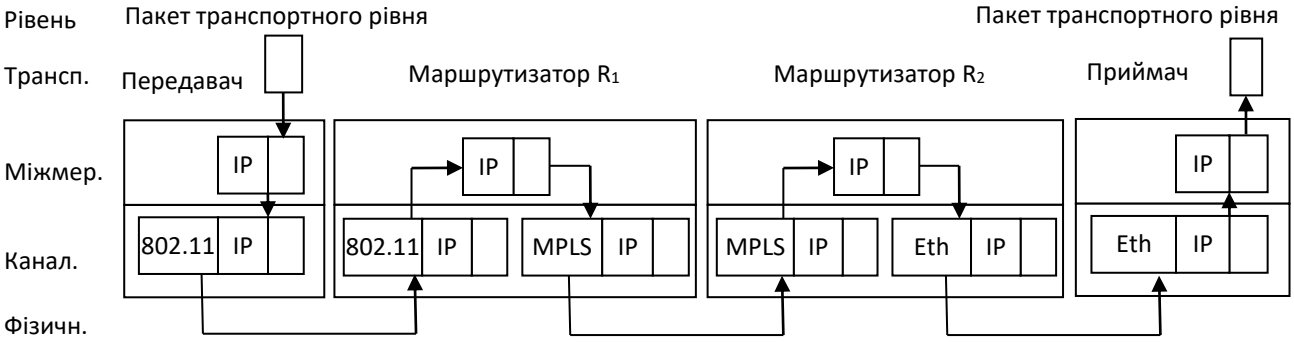


Рис. 4.5. Шлях та обробка пакетів

Протокол IP на R_1 витягує з кадру IP-пакет, а з нього – IP-адресу мережі одержувача. За цією адресою й своєю ТМ він визначає, що пакет треба відправити на маршрутизатор R_2 мережею MPLS. Мережа MPLS орієнтована на з'єднання, тому, щоб пакет пройшов нею, в ній спочатку створюється віртуальний канал MPLS, що веде до R_2. Пакет, упакований в кадр із заголовком MPLS, пройшовши цим каналом, досягає R_2 – кордону мережі Ethernet.

На маршрутизаторі R_2 протокол IP видаляє заголовок MPLS і за IP-адресою й своєю ТМ визначає наступну ділянку мережевого рівня, яка веде до вузла призначення. Пакет може виявитися занадто великим, адже 802.11 працює з кадрами більшого розміру, ніж Ethernet. Тоді пакет ділиться на фрагменти, і кожен з них записується в поле даних кадру Ethernet і передається приймачу. На кінцевому вузлі з прибулих фрагментів протокол IP збирає пакет і передає протоколу транспортного рівня. ◄

Існує окремий випадок з'єднання мереж, який відносно легко реалізується навіть за різних мережевих протоколів: передавач і приймач належать мережам одного типу, між якими є мережа іншого типу. Один з методів з'єднання мереж у цій ситуації називається **тунелюванням** (tunneling).

Приклад. Припустимо, що організація має локальні мережі з IPv6 у Манчестері та Ліверпулі, а між ними є мережа з IPv4. Щоб надіслати IP-пакет хосту в Ліверпулі, хост у Манчестері формує пакет, що містить ліверпульську IPv6-адресу й відправляє його на мультипротокольний маршрутизатор, який з'єднує манчестерську мережу IPv6 і мережу IPv4. Отримавши пакет IPv6, маршрутизатор записує його *як дані* в пакет з IPv4-адресою маршрутизатора, що з'єднує мережу IPv4 і ліверпульську мережу IPv6. Коли пакет потрапляє на цю адресу, мультипротокольний маршрутизатор у Ліверпулі витягує початковий IPv6-пакет і надсилає його на хост призначення.

Шлях через мережу IPv4 утворює свого роду *тунель*, що з'єднує два мультипротокольні маршрутизатори – IPv6-пакет в IPv4-упаковці переміщається цим тунелем. Перепаковують і переадресовують пакети мультипротокольні маршрутизатори, які обробляють як IPv6-, так і IPv4-пакети. Хостам у Манчестері та Ліверпулі не потрібно дбати про взаємодію з мережею IPv4. ◄

Тунелювання широко використовується для з'єднання ізольованих вузлів і мереж через мережу-посередник. В результаті з'являється нова мережа, ніби накладена на вже існуючу – **оверлейна мережа** (overlay). Недоліком тунелювання є те, що пакет не може бути доставлений на жоден з хостів, розташованих у мережі-посереднику. Проте ця властивість стає перевагою у віртуальних приватних мережах.

Контрольні запитання

Поняття: маршрутизація, протокол маршрутизацій, маршрутизований протокол, таблиця маршрутизації, стандартний шлюз, основні властивості IP, призначення протоколу ICMP, автономна система, прикордонний маршрутизатор автономної системи, протокол зовнішнього шлюзу.

Призначення поля IPv4-заголовка «Диференційоване обслуговування».

Призначення поля IPv4-заголовка «Час життя».

Як і в яких ситуаціях обчислюється контрольна сума заголовка IP-пакета?

Чому в IP обчислення контрольної суми включає заголовок й не включає дані?

Які адреси містить IPv4-заголовок?

Основні поля записів у таблиці маршрутизації.

Що означає адреса мережі призначення 0.0.0.0/0 у таблиці маршрутизації?

Для чого потрібен стандартний шлюз?

Головне завдання протоколу IP.

Що є одиницею обробки в протоколі IP?

Як IP-адреси отримувача та джерела повідомлення передаються з транспортного рівня до протоколу IP?

Опишіть загальний алгоритм роботи протоколу IP.

Правило найдовшого спільного префікса.

Назвіть кілька ситуацій, в яких маршрутизатор знищує IP-пакет.

Призначення протоколу ICMP.

Для яких протоколів призначені повідомлення, надіслані протоколом ICMP?

Про що може свідчити ICMP-повідомлення «час вичерпано»?

Як ICMP-повідомлення «час вичерпано» використовується в утилітах **`ping`** і **`traceroute`**?

Чому алгоритми обміну векторів відстаней використовуються лише у невеликих мережах?

На яких маршрутизаторах працює протокол BGP?

Задачі (типові)

Задано абстрагований вміст ТМ (послідовність пар вигляду «адреса/маска-інтерфейс»). Визначити інтерфейси, на які маршрутизатор відправить пакети з заданими IP-адресами.

Задано схему мережі з кількома роутерами. Визначити вміст таблиці маршрутизації на тому чи іншому роутері, можливо, з урахуванням агрегації маршрутних записів.

Розділ 5. Транспортні протоколи

Протоколи транспортного рівня працюють на хостах. Головна їх задача – забезпечити сервіс із передачі даних процесам прикладного рівня. На боці відправника протокол транспортного рівня отримує повідомлення від процесу прикладного рівня, а на боці одержувача передає повідомлення потрібному процесу прикладного рівня. Передачу даних між процесами прикладних і транспортних протоколів виконує операційна система за допомогою спеціального програмного каналу, що називається портом (див. п. 2.4).

Основними транспортними протоколами в Інтернеті є TCP та UDP. TCP контролює передачу послідовності пакетів, за необхідності він здатен прискорювати або уповільнювати надсилання даних, а в разі втрати даних надсилати їх повторно. UDP передає дані, не контролюючи їх доставку. На хостах реалізація TCP та UDP, зазвичай, вбудована в ядро операційної системи (ОС), хоча й існують їх реалізації бібліотечними засобами.

5.1. Загальна схема роботи TCP

TCP (Transmission Control Protocol, протокол керування передачею) – протокол транспортного рівня, який забезпечує гарантовану доставку даних користувача (звісно, якщо не вийшов з ладу вузол мережі, без якого немає шляху між джерелом і отримувачем).

Нагадаємо, що TCP *орієнтовано на з'єднання*: вузол, який ініціює обмін даними, спочатку зв'язується з іншим вузлом, отримує від нього підтвердження готовності до обміну та підтверджує власну готовність.

TCP має таку загальну схему.

Встановлення з'єднання (connection).

Передача даних (data transmission).

Завершення з'єднання (disconnection).

З'єднання описується четвіркою (*адреса відправника*, *порт відправника*, *адреса одержувача*, *порт одержувача*). Встановлюючи з'єднання, учасники обмінюються даними вигляду (*адреса*, *порт*) і підтверджують готовність до подальшого обміну даними. З'єднання дозволяє TCP вести двосторонній обмін даними між вузлами-учасниками, який є засобом що забезпечує надійну доставку повідомлень.

Під час передачі даних TCP розбиває повідомлення на послідовність сегментів і до кожного додає TCP-заголовок, утворюючи TCP-пакет. Заголовок містить номери портів учасників та спеціальні дані, за якими одержувач може відновити повідомлення з окремо одержаних сегментів, або виявити пошкодження та повідомити відправнику, щоб він повторно надіслав втрачений сегмент даних.

З'єднання завершується також шляхом обміну спеціальними даними, які сигналізують про закінчення роботи та підтверджують отримання цих даних.

5.2. Заголовок TCP-пакета

Протокол TCP розбиває повідомлення на сегменти, довжина яких визначена в ОС. Перед кожним сегментом додається TCP-заголовок, тобто формується TCP-пакет. У заголовку є **обов'язкова частина** та **додаткова частина**. Перша має довжину 20 байт, друга – від 0 до 40 байт, кратну 4.

- ✓ Під час передачі даних від TCP до IP та навпаки перед TCP-заголовком ще додається **IP-псевдозаголовок**. Для IPv4 він має довжину 20 байт і містить такі дані: IP-адреса відправника (4 байт), IP-адреса одержувача (4 байт), нульовий байт (1 байт), поле «Протокол» з кодом транспортного протоколу (1 байт), довжина пакета (2 байт), порт відправника (2 байт), порт одержувача (2 байт), довжина сегмента даних у пакеті (2 байт), контрольна сума (2 байт).

Обов'язкові поля в TCP-заголовку такі.

Порт відправника (Source Port, 2 байт) – номер порту, що визначає процес-джерело повідомлення.

Порт одержувача (Destination Port, 2 байт) – номер порту, що визначає процес, якому на боці одержувача призначено повідомлення.

Порядковий номер SN (Sequence Number, 4 байт) – номер першого байта сегмента у потоці даних у цьому з'єднанні. Якщо в заголовку пакета встановлено біт SYN (на фазі встановлення з'єднання), то в цьому полі передається початковий номер *ISN* (див. нижче).

Номер підтвердження AN (Acknowledgment Number, 4 байт) – якщо в заголовку пакета встановлено біт ACK (див. нижче), то це поле містить порядковий номер байта, на який чекає відправник пакета, тобто всі попередні байти з номерами від *ISN*+1 до *AN*–1 включно він успішно отримав.

- ✓ Порядкові номери займають 4 байт, тому довжина повідомлення на практиці не може бути більше 2^{32} байт.

Зсув даних (Data Offset, 4 біти) – довжина TCP-заголовка, тобто кількість 4-байтових слів (від 5 до 15).

Зарезервовано (Reserved, 6 біт) – це поле заповнюється нулями.

Керувальні біти (Control Bits, 6 біт), встановлення яких (в 1) відповідає певному режиму роботи:

URG – обробляється поле вказівника термінових байтів (Urgent Pointer);

ACK – обробляється поле номера підтвердження;

PSH – здійснити «проштовхування»: TCP має негайно передати процесу-одержувачу всі дані з буфера прийому для обробки, навіть якщо буфер не заповнений;

RST – перезавантажити поточне з'єднання;

SYN – запит на встановлення з'єднання;

FIN – даних для передачі більше немає.

Довжина вікна (Window, 2 байт) – кількість байтів у вікні (див. нижче).

Контрольна сума CS (Checksum, 2 байт) – потрібна для виявлення можливих помилок у пакеті, коли він проходить мережею. Це додатковий код суми двобайтових полів даних пакета (у це поле перед обчисленням суми записується 0). Перевірка: сума полів разом з цим значенням має давати 0. Контрольна сума, крім заголовка й даних пакета, враховує перші 12 байт

псевдозаголовка – це запобігає прийому пакетів, під час передачі яких трапилася помилка в адресах.

Вказівник термінових байтів (Urgent Pointer, 2 байт) – кількість термінових байтів, розміщених на початку поля даних сегмента. Протокол TCP не обробляє термінові дані, але сигналізує про них прикладному процесу. Поле Urgent Pointer обробляється, якщо встановлено біт URG.

5.3. Установлення з'єднання

Процес установлення з'єднання називається **рукостисканням** (handshake) і має *три кроки* обміну повідомленнями у вигляді сегментів певного розміру. Головний успішний сценарій такий.

1. Відправник *A* надсилає одержувачу *B* TCP-пакет, заголовок якого містить *початковий номер у послідовності ISN(A)* та *запит зв'язку*. Початковий номер *ISN(A)*, тобто Initial Sequence Number, є значенням поля *SN*. Запит зв'язку – це встановлений біт SYN. Даних у пакеті немає.

2. Одержувач *B*, отримавши запит, зберігає номер і намагається створити сокет для прийому повідомлення. Якщо сокет створено, то *B* надсилає *A* TCP-пакет, також без даних, у заголовку якого встановлено біт ACK, що підтверджує встановлення з'єднання для отримання даних від *A*. Оскільки TCP забезпечує двосторонній обмін даними, у цьому ж заголовку встановлено біт SYN, що означає запит зв'язку для передачі даних від *B* до *A*, і міститься *ISN(B)*. Далі *B* переходить у стан SYN-RECEIVED (*синхронізацію отримано*). Якщо сокет не створено, то *B* надсилає *A ознаку неуспіху*, встановивши в заголовку прапорець RST.

3. Якщо *A* отримує від *B* ознаку успіху (біт ACK=1), то він переходить у стан ESTABLISHED (*з'єднання встановлено*). Він зберігає *ISN(B)* і надсилає пакет, у заголовку якого встановлено біт ACK. Ця ознака підтвердження дозволить *B* далі включити в свій наступний сегмент корисні дані для *A*. Цей пакет уже містить перший сегмент повідомлення, адже від *B* вже отримано підтвердження встановленого з'єднання. Нумерація байтів сегмента починається з *ISN(A)*+1. Якщо ж *A* отримує ознаку неуспіху (прапорець RST), то він припиняє спроби з'єднатися. Якщо *A* не отримує відповіді від *B* протягом 10 секунд, то він повторює процес з'єднання.

4. Якщо *B* в стані SYN-RECEIVED отримує ознаку підтвердження, то він переходить в стан ESTABLISHED, інакше після певної затримки закриває сокет і переходить в стан CLOSED.

- ✓ Після встановлення з'єднання взаємодія учасників стає симетричною, оскільки кожен з них як передає, так і приймає дані.
- ✓ Протокол TCP має вразливість: зловмисник може встановити фальшиву IP-адресу відправника й надіслати серверу багато пакетів SYN. Отримавши пакет SYN, сервер виділяє певний об'єм своїх ресурсів для встановлення нового з'єднання. Обробка довгої серії пакетів SYN може зайняти всі ресурси сервера, що призводить до відмови в обробці нових запитів (ця атака називається «відмова в обслуговуванні» – Denial of Service, DoS).

5.4. Передача даних

Розглянемо властивості передачі сегментів повідомлення за протоколом TCP, засновані на тому, що кожен TCP-пакет містить усі дані з'єднання, потрібні для керування процесом передачі.

Номери сегментів. При обміні даними одержувач використовує номери сегментів у отриманих пакетах для відновлення їх порядку. Одержувач повідомляє відправнику номер, до якого він успішно отримав дані. Всі нові пакети, номери яких уже підтверджені, ігноруються. Якщо отриманий пакет містить номер більше за очікуваний, то дані з пакета зберігаються в буфері, але підтверджений номер не змінюється. Якщо згодом надходить пакет, номер у якому є очікуваним, то дані з нього додаються до відновленої частини повідомлення.

Керування швидкістю передачі. TCP керує процесом передачі так, щоб відправник не надсилав пакети інтенсивніше, ніж їх може обробити одержувач. Для цього одержувач у пакеті підтвердження вказує поточний розмір приймального буфера. Відправник зберігає цей розмір і відправляє даних не більше ніж указав одержувач. Якщо одержувач указав нульовий розмір вікна, то передача даних припиняється доти, доки одержувач не надішле більший розмір вікна.

Прийом сегментів без буферизації. У певних ситуаціях відправник явно вимагає, щоб одержувач просував дані в певній послідовності у своєму процесі без їх буферизації, тобто «проштовхував» їх. Для цього використовується прапорець PSH (push). Якщо в одержаному пакеті виявлено прапорець PSH, то TCP віддає всі буферизовані на поточний момент дані процесу-одержувачу. «Проштовхування» використовується, зокрема, в інтерактивних застосунках. Наприклад, у мережевих терміналах немає сенсу очікувати введення користувача після того, як він закінчив набирати команду. Тому останній пакет, що містить дані команди, має прапорець PSH для того, щоб процес на боці одержувача зміг відразу почати її виконання.

Контрольні суми. Пакети під час передачі лініями зв’язку не тільки губляться, але й пошкоджуються. Для виявлення пошкоджень TCP використовує *контрольну суму* (КС) у заголовку, яку відправник обчислює за даними пакета. Коли пакет прибуває в пункт призначення, одержувач обчислює КС за отриманим пакетом і перевіряє КС у пакеті. Якщо перевірка виявила помилку, то одержувач відкидає TCP-пакет і запитує повторну передачу.

- ✓ Головна властивість протоколу TCP: *передати все, що є, нехай і не відразу*. Після того, як встановлено з'єднання, TCP гарантує доставку даних. Якщо частина даних виявляється пошкодженою, то вона передається повторно. Бітова послідовність даних, доставлена одержувачу, відновлюється в такому самому вигляді, в якому передавалася.
- ✓ TCP реалізує надійний *віртуальний канал* між вузлами в з’єднанні й цим значно спрощує протоколи прикладного рівня.

5.5. Завершення з’єднання

За безпомилкової роботи з’єднання завершується шляхом обміну повідомленнями, який ініціює відправник. Головний успішний сценарій аналогічний установленню з’єднання й має три кроки обміну TCP-пакетами.

1. Відправник *A* надсилає одержувачу *B* TCP-пакет, у заголовку якого встановлено біти FIN і ACK, тобто підтвердження того, що даних для передачі більше немає. Даних у пакеті немає.

2. *B* надсилає *A* пакет, в якому так само встановлено біти FIN і ACK.

3. Отримавши це підтвердження від *B*, *A* закриває з’єднання і відправляє *B* пакет із встановленим бітом ACK (підтвердження завершення з’єднання).

4. Отримавши підтвердження, *B* закриває з’єднання.

5.6. Огляд UDP

UDP (User Datagram Protocol, протокол дейтаграм користувача) – протокол транспортного рівня, який швидко передає дані користувача, але не гарантує їх доставку.

UDP описаний в RFC 768. UDP надає *ненадійний* сервіс: пакети можуть надходити одержувачу не в порядку надсилання, дублюватися або зникати. Протокол UDP використовується для передачі коротких повідомлень, коли додаткові витрати на встановлення з'єднання та перевірку успішності доставки даних перевищують витрати на повторне, в разі невдачі, надсилання повідомлення. Його також застосовують, коли прикладний процес сам встановлює з'єднання й перевіряє доставку пакетів, як, наприклад, протокол мережевого доступу до файлових систем NFS (Network File System).

Отже, з погляду UDP, перевірка помилок і виправлення або не потрібні, або мають виконуватися на прикладному рівні. UDP використовується в програмах, чутливих до часу, коли краще скинути пакети, ніж чекати пакетів, що затрималися.

UDP використовується серверами, які відповідають на невеликі запити від величезної кількості клієнтів. Зокрема, UDP дозволяє процесам відправляти інкапсульовані IP-пакети без встановлення з'єднань. Приклади протоколів прикладного рівня, які використовують UDP: NFS, TFTP (Trivial File Transfer Protocol – простий протокол передачі файлів), SNMP (Simple Network Management Protocol – простий протокол керування мережею), DNS (Domain Name Service – служба доменних імен). Його також використовують потокові мультимедійні програми типу IPTV, Voice over IP, протоколи тунелювання IP і чимало онлайн-ігор.

✓ Головна властивість UDP: передати все, не контролюючи доставку.

UDP передає пакети, які містять заголовок з чотирьох 2-байтових полів

(*порт відправника*, *порт одержувача*, *довжина*, *контрольна сума*)

та поле даних. Номери портів визначають прикладні процеси відправника та одержувача. Коли прибуває пакет UDP, дані передаються процесу, зв'язаному з портом отримувача.

Дані про порт відправника потрібні, перш за все, коли створюється відповідь для відправника. Копіюючи значення поля *порт відправника* (Source Port) з вхідного пакета в поле *порт призначення* (Destination Port) вихідного сегмента, процес, який надсилає відповідь, може вказати, якому саме процесу на протилежному боці вона призначається. Поле *довжина* містить довжину заголовка й даних (кількість байтів). Мінімальне значення – 8, максимальне – 65515 (через обмеження на розмір IP-пакета). Поле *контрольна сума* заповнюється відправником і використовується одержувачем (аналогічно TCP). UDP-пакет повинен мати парну довжину, тому, при обчисленні контрольної суми за непарної довжини до нього додається нульовий байт.

✓ Як і в протоколі TCP, під час передачі даних від UDP до IP та навпаки перед UDP-пакетом додається IP-псевдозаголовок (див. п. 5.2).

З погляду UDP, дані після заголовку в пакеті є цілісним повідомленням. UDP *не* розбиває повідомлення на сегменти й не об'єднує кілька повідомлень в одному пакеті. Кожен виклик UDP з боку прикладного процесу-відправника веде до формування й відправки UDP-пакета. Відповідно, щоб отримати всі повідомлення, процес-одержувач має стільки ж разів викликати свій UDP.

На боці одержувача UDP отримує пакет від мережевого рівня, перевіряє КС і передає повідомлення прикладному процесу через *порт відправника*.

Якщо перевірка КС виявила помилку або процесу, підключеного до вказаного порту, не існує, то пакет ігнорується. Якщо пакети надходять швидше, ніж UDP встигає їх обробляти, то нові

пакети також ігноруються. Протокол UDP ніяк не підтверджує успішний прийом даних і не повідомляє про помилку, не забезпечує надходження повідомлень у порядку їх відправки, не встановлює з’єднання між прикладними процесами, тому він є *ненадійним протоколом без встановлення з'єднання*.

Максимальна довжина UDP-повідомлення дорівнює 65507 байт – це максимальна довжина IP-пакета (65535 байт) мінус мінімальна довжина IP-заголовка (20 байт) і мінус довжина UDP-заголовка (8 байт). На практиці зазвичай використовуються повідомлення довжиною 8 Кбайт.

Контрольні запитання

Поняття: TCP, UDP, сокет, з’єднання

Загальна схема TCP

Призначення полів TCP-заголовка: порти відправника та одержувача, порядковий номер, номер підтвердження, довжина заголовка, керувальні біти, довжина вікна, контрольна сума.

Головний принцип TCP.

Головний успішний сценарій встановлення з'єднання в TCP.

Як можна змінити головний успішний сценарій встановлення з'єднання в TCP, щоб протидіяти DoS-атаці?

Описати, як використовуються порядкові номери в сегментах.

Описати керування швидкістю передачі.

Головний успішний сценарій завершення з'єднання.

Опишіть можливу взаємодію учасників TCP-з’єднання, якщо під час розриву з’єднання учасник *B* надсилає учаснику *A* пакет, в якому встановлено тільки біт ACK.

Поля в UDP-заголовку.

Головний принцип UDP.

Чому UDP називається ненадійним протоколом без встановлення з’єднання?

Розділ 6. Сервери імен

З системою адресації хостів за допомогою імен знайомить розділ 2. У цьому розділі описується реалізація цієї системи й організація даних у ній, які дозволяють за доменним ім'ям хоста отримати його IP-адресу, і навпаки, за IP-адресою – доменне ім'я.

6.1. Доменні імена

IP-адреси, тобто номери, що позначають хости мережі, легко обробляються протоколами, але незручні для людини, якій природно позначати вузли мережі символьними іменами.

Використання імен та адрес потребує визначення IP-адреси за символьним ім'ям і навпаки, імені за адресою. Спочатку основою для перетворення імен в адреси й навпаки був файл **`hosts.txt`**, змістом якого була відповідність між всіма зареєстрованими символьними іменами хостів та їх IP-адресами. Цей файл формувався **в єдиному екземплярі** на сервері, розміщеному в Стенфордському дослідницькому інституті (SRI International), і кожен комп'ютер, підключений до мережі, для обміну з іншими мав отримати копію цього файлу. Файл `hosts.txt` і тепер є в операційних системах, в яких реалізовано TCP/IP, але містить тільки один запис `localhost – 127.0.0.1`.

Система з єдиним місцем реєстрації імен дозволяла легко керувати іменуванням хостів, проте значно ускладнювала роботу Інтернету в цілому. Інший підхід до реалізації іменування хостів було винайдено в 1983 р. за ініціативою Джонатана Постела (Jonathan Postel): Пол Мокапетріс (Paul Mockapetris) розробив систему символьних імен DNS (Domain Name System – система доменних імен) та організацію даних і програмного забезпечення, які дозволяли давати хостам символьні імена та швидко отримувати за іменами хостів їх IP-адреси й навпаки, за адресами визначати імена.

> **Система доменних імен DNS** (Domain Name System) – це організація даних і програмне забезпечення, які дозволяють давати хостам символьні імена та швидко отримувати за іменами хостів їх IP-адреси й навпаки, за адресами визначати імена.

Система символьних імен – **ієрархічна схема** іменування хостів та множин хостів, заснована на доменах (Domain – зона, область, територія).

> **Домен** – це деяка **підмножина імен** мережі Інтернет, які спільно централізовано адмініструються та обслуговуються спеціальними серверами системи доменних імен. Домен ідентифікується ім'ям.
>
> **Доменне ім'я** (domain name) хоста, або **доменна адреса** (domain address), або **ім'я хоста** (host name) – це позначення конкретного мережевого хоста, яке ідентифікує власника цього хоста.

Домени утворюють ієрархію множин імен. На верхньому рівні є дві групи доменів (toplevel domains – TLD) – родові домени та домени держав. Усі вони мають різні імена.

> **Родові домени** та їх імена відповідають певним сферам діяльності, наприклад, домени з іменами **`com`** (комерція), **`edu`** (навчання), **`net`** (мережі).
>
> **Домени держав** та їх імена відповідають державам, наприклад, **`us`** (США), **`jp`** (Японія), **`ua`** (Україна).

За кожною державою згідно з міжнародним стандартом ISO3166 закріплено один домен держави, тому зараз їх майже 300. У 2010 р. були введені додаткові імена країн у алфавітах, відмінних від латинського (арабський, китайський, кирилічні та інші).

Зареєстрованих родових доменів верхнього рівня в 1985 р. було 7, у 2010 р. – 25, на початку 2016 р. – майже 900, на початку 2017 р. – більше 1500.

Кореневий домен (**домен нульового рівня**) – це об'єднання доменів верхнього рівня.

Кожен домен верхнього рівня розбивається на **піддомени** (subdomains), які мають власні імена й, у свою чергу, можуть розбиватися на піддомени, і так далі. Так утворюються домени другого рівня, третього рівня тощо.

Повне ім'я піддомену записується як його ім'я перед повним ім'ям домену вищого рівня через крапку, наприклад, повні імена доменів другого рівня **google.com**, **edu.ua**, **knu.ua**, третього рівня **univ.kiev.ua** або четвертого **cyb.univ.kiev.ua**.

Імена доменів можуть бути **абсолютними** або **відносними**. Абсолютне ім'я домену закінчується крапкою (як **univ.kiev.ua.**), відносне – ні (як **univ.kiev.ua**).

- ✓ Імена доменів утворюють **дерево**, коренем якого є ім'я «**.**», що позначає кореневий домен. Ім'я домену являє собою **позначення шляху від цього домену до кореневого**.

Листками в дереві є імена доменів, які не мають піддоменів (**кінцеві домени**). Кінцевий домен складається з одного імені, яке може бути ім'ям деякого веб-сайту, сервера електронної пошти або іншої служби, і за цим ім'ям може «ховатися» один хост, а можуть і тисячі хостів організації, яка веде цей сайт або службу.

Імена доменів нечутливі до регістрів символів і зазвичай записуються малими латинськими літерами. Імена також можуть містити цифри від 0 до 9 та дефіс «-», але він не може бути першим або останнім символом. Довжина імені від 2 до 63, довжина повного імені не більше 255, а рівнів може бути до 127.

- ✓ Структура доменів відображає не фізичну будову мережі, а **логічний поділ** імен між країнами, групами організацій, їх підрозділами тощо. Наприклад, два факультети, розташовані в одній будівлі, користуються спільною локальною мережею, але доменні імена їх хостів належать різним доменам. Або навпаки, деяка служба університету розміщена в різних корпусах з різними локальними мережами, але логічно всі її хости належать одному домену.

6.2. Реєстрація імен доменів

Реєстрацією імен верхнього рівня тривалий час керувала IANA (Internet Assigned Numbers Authority – Адміністрація адресного простору Інтернет). З жовтня 2016 р. – Public Technical Identifiers, філія ICANN (Internet Corporation for Assigned Names and Numbers – Інтернет-корпорація з присвоєння імен та адрес).

Кожен домен верхнього рівня має **доменного реєстратора** – організацію, яка затверджена в ICANN і керує доменом. Наприклад, доменом **.ua** керує національний реєстратор – компанія **Imena.UA**. Доменами другого рівня керують доменні реєстратори, затверджені відповідним доменним реєстратором першого рівня.

- ✓ У кожному домені є своя адміністрація, яка розподіляє імена піддоменів незалежно від домену вищого рівня. Після реєстрації імені піддомену, для створення в ньому піддоменів не потрібен дозвіл від адміністрації доменів вищих рівнів.

Майже всі організації в США реєструються під родовими доменами, а за межами США – під доменами своїх держав. Наприклад, у Японії домени **ac.jp** і **co.jp** відповідають американським доменам **edu** і **com**, у Нідерландах всі домени організацій поміщаються безпосередньо під доменом **nl**, а в Україні єдиного підходу немає.

Приклад. Доменні імена факультетів комп'ютерних наук (computer science) кількох університетів:

cs.stanford.edu – університет Стенфорда, США;

cs.vu.nl – університет міста Врійє, Нідерланди;

cs.keio.ac.jp – університет Кейо, Японія;

ami.lnu.edu.ua – Львівський національний університет;

fotius.cdu.edu.ua – Черкаський національний університет;

cyb.univ.kiev.ua та **unicyb.kiev.ua** – Київський національний університет.

6.3. Зони та сервери імен

Система доменних імен DNS – це ієрархічна схема іменування хостів та множин хостів, заснована на доменах, плюс розподілена база даних, яка реалізує цю схему. Ця розподілена система формує відповідність між іменами доменів, зручними для користувачів, та IP-адресами хостів.

В основі роботи DNS лежить поняття доменної зони.

> **Доменна зона** – це множина імен піддоменів одного рівня деякого (**батьківського**) домену.

Доменну зону часто позначають ім'ям батьківського домену з крапкою попереду, наприклад **.ua**. Зона **.ua** містить імена доменів другого рівня вигляду ***ім'я*.ua**, зона **.kiev.ua** – імена доменів третього рівня вигляду ***ім'я*.kiev.ua** тощо.

Імена доменів верхнього рівня утворюють **кореневу зону**, яка ніяк не позначається або позначається крапкою «**.**».

Імена піддоменів мають бути унікальними **лише в межах зони**, наприклад, ім'я **edu** може позначати як домен верхнього рівня, так і піддомен домену будь-якої країни: **edu.pl**, **edu.ua** тощо.

- ✓ Отже, множина імен DNS розділена на зони, які не перетинаються й кожна з яких містить певну частину загального дерева імен доменів.

DNS реалізована у вигляді розподіленої бази даних, розташованої на мільйонах спеціальних спільно працюючих DNS-серверів. У пам'яті цих серверів зберігаються великі таблиці, в яких кожному доменному імені ставиться у відповідність IP-адреса.

> **DNS-сервер**, або **сервер імен** (name server) – це програма, призначена для відповідей на DNS-запити за відповідним протоколом; також це хост, на якому запущена вказана програма.

DNS-сервери створюються для доменних зон. У кожній зоні має бути **не менше двох** серверів імен – хостів, які зберігають базу даних для зони. Це **основний (первинний) сервер імен** і не менш ніж один **резервний сервер** (**вторинний**, або так зване **дзеркало**), який отримує копії даних від основного. Для надійності сервери імен будь-якої зони повинні бути в різних мережах і мати окремі з'єднання з Інтернетом.

Для будь-якої зони імен на основному DNS-сервері зони ведеться **база даних DNS зони** – набір текстових файлів, даними в яких є так звані **записи ресурсів** (про це далі). Резервні сервери періодично копіюють собі базу даних DNS (виконують **трансфер зони**).

DNS-сервери зон усіх рівнів і окремих мереж протягом певного часу зберігають дані, отримані від інших серверів імен, і це дублювання даних значно прискорює виконання запитів. Насправді, звернення до одного з кореневих серверів імен або їх дзеркал відбувається, приблизно, у 1/5–1/4 частині випадків визначення доменних імен. Частіше запити використовують дані, раніше отримані від кореневих та інших серверів імен.

✓ Трансфер зони знижує навантаження на основний DNS-сервер і прискорює доступ до бази для клієнтів, віддалених від основного сервера.

У конфігурації сервера вказується, чи є сервер основним або резервним, а також, які файли на ньому містять опис зони.

✓ Зміна даних про зону відповідальності сервера, наприклад, про те, що додається або видаляється домен чи вузол, проводиться **тільки на основному сервері**.

Підтверджений запис (authoritative record) – це запис у базі даних основного DNS-сервера.

Кешований запис (cached record) – це запис, отриманий від інших DNS-серверів.

Підтверджений запис, на відміну від кешованого, завжди вважається правильним.

Особливості кореневої зони. Цю зону імен обслуговують 13 **кореневих серверів імен** (root name servers) з іменами **`a.root-servers.net`**, **`b.root-servers.net`**, …, **`m.root-servers.net`**. Вони реалізовані на потужних комп'ютерах, кожен з них зберігає **дані про всі домени вищого рівня**, тобто дані багаторазово продубльовано. Сервери розташовані в різних географічних точках, доступні за адресою будь-якого з них, тому запит передається найближчому з них. Насправді, кожен кореневий сервер має кілька десятків дзеркал, тому справжня кількість серверів верхнього рівня більше 500 (станом на початок 2020).

✓ Інформаційні середовища сучасних соціальних мереж складають конкуренцію DNS, важливим обмеженням у якій є вимога **унікальності імені**. Соціальні мережі дозволяють користувачам мати однакові імена й розрізняють користувачів за іншими властивостями. Отже, новітні технології соціальних мереж у перспективі можуть витіснити технологію DNS.

6.4. Визначення IP-адреси за доменним ім'ям

✓ Головна задача сервера імен – визначення IP-адреси за доменним ім'ям хоста (**визначення імен** – name resolution).

Для того, щоб отримати IP-адресу хоста за його доменним ім'ям, прикладний процес передає ім'я бібліотечній процедурі – **розпізнавачу** (resolver). Процес-розпізнавач звертається із запитом визначення імені до DNS-сервера ЛМ, який шукає у своїй базі даних це ім'я. Далі можливі дві ситуації.

1. Потрібний хост належить домену, за який відповідає цей локальний DNS-сервер. DNS-сервер у відповідь передає запис із IP-адресою хоста розпізнавачу, а той передає отриману IP-адресу прикладному процесу.

2. Потрібний домен є віддаленим і в кеші локального DNS-сервера даних про нього немає. DNS-сервер надсилає **віддалений запит** на один із кореневих серверів або його дзеркало.

Подальші дії з отримання адреси розглянемо на прикладі.

Приклад

Прикладному процесу на вузлі `serv.univ.edu.ua` потрібна IP-адреса для імені `cs.stanford.edu` в університеті Стенфорда й у кеші локального сервера імен немає даних про цей домен.

Крок 1. Процес-розпізнавач на вузлі `serv.univ.edu.ua` звертається із запитом визначення імені домену до локального сервера імен (рис. 6.1).

Крок 2. Локальний сервер імен надсилає запит, що містить ім'я потрібного домену та деякі інші дані. Запит надходить на один з кореневих серверів імен, скажімо, `a.root-servers.net`.

Крок 3. На кореневому сервері імен немає ані адреси вузла в університеті Стенфорда, ані адреси сервера імен університету, проте там має бути адреса сервера імен зони `.edu`, якій належить `cs.stanford.edu`. Отже, кореневий сервер повертає ім'я та IP-адресу сервера імен зони `.edu` (`a.edu-servers.net`). Ця відповідь називається **частковою**.

Кроки 4, 5. Локальний сервер імен направляє запит серверу імен edu (`a.edu-servers.net`), і той повертає ім'я сервера імен університету Стенфорда.

Кроки 6, 7. Локальний сервер імен відсилає запит на сервер імен університету Стенфорда. Цей сервер відповідає за зону `.stanford.edu`, тому дає потрібну відповідь – IP-адресу `171.64.64.64`.

Крок 8. Локальний сервер імен передає отриману IP-адресу `171.64.64.64` процесу-розпізнавачу на `serv.univ.edu.ua`.

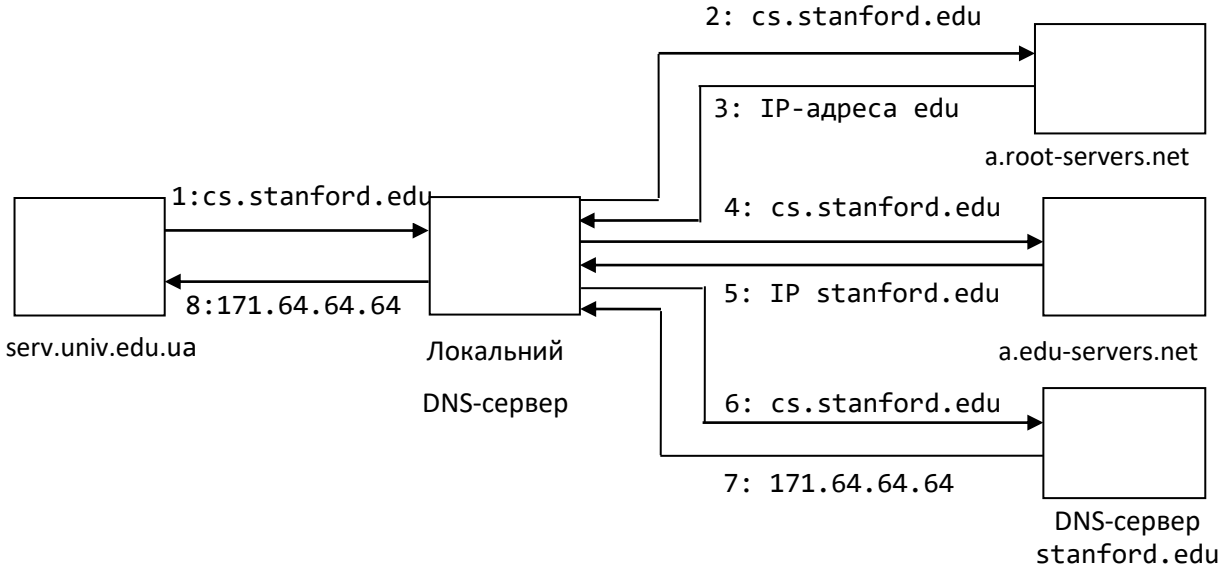


Рис. 6.1. Послідовність запитів і відповідей

Нарешті, розпізнавач на вузлі `serv.univ.edu.ua` передає отриману IP-адресу прикладному процесу. ◄

✓ Отже, вузлу достатньо звернутися до локального сервера імен на своєму домені.

У сценарії, наведеному в прикладі, є три технічні моменти.

1. **Два механізми запитів**.

Коли процес-вирішувач на хості `serv.univ.edu.ua` надсилає запит на локальний сервер імен, той надсилає запити віддаленим серверам імен, поки не отримає відповідь, яку можна повернути розпізнавачу.

Запит розпізнавача до локального сервера імен називається **рекурсивним запитом** (recursive query).

Проте кореневий і всі наступні віддалені сервери імен **не продовжують** рекурсивно запит локального сервера імен, а повертають часткові відповіді. Локальний сервер імен продовжує надсилати запити (за кожною новою отриманою IP-адресою).

Запит за IP-адресою з часткової відповіді називається **ітеративним запитом** (iterative query).

2. **Кешування відповідей**.

Усі відповіді, включно з частковими, зберігаються в **кеші локального сервера імен**. Звідси, якщо інший хост в зоні `univ.edu.ua` запросить `cs.stanford.edu`, то локальному серверу імен відповідь буде вже відома. Також, якщо буде запит до іншого хоста на домені `stanford.edu`, наприклад `mathematics.stanford.edu`, то цей запит буде надісланий на сервер імен цієї зони.

✓ Використання відповідей, збережених у кеші, значно скорочує кількість кроків у визначенні відповідей на запити.

Відповіді, збережені в кеші, не є підтвердженими, адже дані про зміни в зоні `cs.stanford.edu` не розповсюджуються автоматично по всіх кешах, в яких може зберігатися копія цих даних. Тому записи кеша зазвичай «довго не живуть»: кожен запис ресурсу має поле, яке визначає тривалість зберігання запису в кеші.

Якщо вузол роками має постійну адресу, то вважається надійним зберігати дані про нього в кеші протягом одного дня. Для більш мінливої інформації варто було б видаляти всі записи через кілька секунд або хвилин, адже застарілі дані небезпечні помилковим визначенням імен. Проте на практиці, навпаки, ці дані зберігаються на своєму хості, тобто кешуються. Для цього існують спеціальні програми, що підтримують міні-бази DNS для тих адрес, які використовуються часто.

3. **Транспортування повідомлень**.

Запити й відповіді надсилаються за транспортним протоколом UDP, тобто в UDP-пакетах. Якщо протягом деякого (нетривалого) часу відповідь не приходить, то DNS-клієнт повторює запит і перевіряє інший сервер зони після кількох таких спроб. Взагалі ж, процес визначення IP-адрес побудовано так, щоб робота не припинялася, якщо вийшов з ладу один із серверів або загубився один з пакетів.

✓ Дізнатися IP-адресу за доменним ім’ям хоста, можна, наприклад, викликавши в командному рядку утиліту **`ping`** або **`nslookup`**. Наприклад, на платформах Windows або Linux команди **`ping univ.kiev.ua`** та **`nslookup univ.kiev.ua`** серед інших даних виведуть IP-адресу у вигляді `91.202.128.77`.

6.5. Записи ресурсів

✓ **База даних DNS** являє собою набір текстових файлів на основному DNS-сервері зони.

У кожній зоні базу веде адміністратор. Вторинні сервери періодично копіюють до себе файли бази. У файлах конфігурації сервера вказується, в яких файлах які зони описано, і чи є сервер первинним або вторинним для цієї зони.

✓ Текст у файлі, що описує зону, являє собою **послідовність записів ресурсів** (resource records).

Запис ресурсу є рядком, що має такі п’ять частин.

`Domain_name  TTL  Class  Type  Value`

`Domain_name` (ім’я домену) позначає домен, якого стосується запис. Воно може бути абсолютним (з крапкою в кінці) або відносним (без крапки). До відносного імені під час обробки файлу автоматично дописується ім'я зони, яка описується. Наприклад, якщо в описі

зони knu.ua записати ім'я csc, то ім'я буде оброблятися як «csc.knu.ua.». У відповідь на запит про домен, заданий ім'ям, сервер повертає всі записи ресурсів, відповідні цьому імені.

✓ Ім'я домену може бути відсутнім – тоді воно береться з попереднього запису ресурсу.

TTL (*Time_to_live* – час життя) позначає тривалість (кількість секунд) збереження запису в кешованій копії файлу – ціле число від 1 до 2^{31}–1, наприклад 86400 (доба), 3600 (година) або 604800 (тиждень).

Class (клас) для даних Інтернета дорівнює IN. Для інших даних застосовуються інші коди, які на практиці зустрічаються рідко.

Type (тип) означає тип запису. Узагалі типів близько півсотні, але найчастіше зустрічаються лише декілька; їх представлено нижче.

Value (значення) – це текстове зображення значення (число, ім'я домену тощо), яке відповідає типу запису.

✓ Символи «#» та «;» у записах починають коментар до кінця рядка, @ позначає ім'я поточної зони, дужки () служать для розташування запису на кількох рядках. В імені допускається метасимвол *.

Найважливіші типи записів ресурсів.

Перший запис опису зони: типу **SOA** (Start Of Authority – початкова точка повноважень). Опис закінчується, коли зустрінеться новий запис типу SOA (це свого роду **заголовок опису зони**). Запис повідомляє параметри зони: ім'я основного сервера імен зони, адреса електронної пошти його адміністратора, унікальний порядковий номер і тривалості.

Приклад. Розглянемо запис типу SOA, яким починається опис гіпотетичної зони. Час життя в цьому записі після імені зони відсутній – він задається останнім параметром запису.

```
univ.edu.ua.  IN  SOA  ns  admin.univ.edu.ua.  (
      2016090101 ;   порядковий номер (serial)
           28800 ;   період оновлення (refresh): 8 годин
            7200 ;   період повторних спроб (retry): 2 години
          604800 ;   термін закінчення (expire): сім діб
           86400 ;   мінімальний термін зберігання (minimum TTL): 24 години
)
```

Ім'я зони (тут це univ.edu.ua.) записується з крапкою. Ім'я ns (без крапки) – це префікс повного імені основного сервера імен зони ns.univ.edu.ua., повне ім'я admin.univ.edu.ua. – поштова адреса адміністратора з крапкою замість символу «@» після імені admin. Далі в круглих дужках ідуть поля, необхідні для правильної обробки зони іншими серверами.

Порядковий номер – це «номер версії» опису зони. Коли в опис у файлі вносяться зміни, цей номер необхідно збільшити – якщо вторинний сервер зони побачить, що в його файлі цей номер менше, ніж у первинного сервера, то він оновить свою копію. Часто цей номер беруть як дату створення опису: тут 2016090101 – це 1 вересня 2016 р., перше оновлення.

Період оновлення повідомляє вторинним серверам, як часто їм треба перевіряти порядковий номер.

Період повторних спроб повідомляє, як часто вторинний сервер має намагатися прочитати дані, коли первинний сервер не відповідає.

Термін закінчення – це час, протягом якого вторинний сервер повинен обслуговувати зону, якщо первинний сервер не відповідає. Після закінчення цього часу вторинні сервери вважатимуть свої дані застарілими.

Мінімальний термін зберігання задає час життя записів для цієї зони за промовчанням, тобто коли в записах не вказано час життя. ◄

- ✓ Після запису SOA порядок інших записів про зону неважливий, хоча існують певні шаблони запису, наприклад, спочатку описуються сервери імен, потім хости, далі псевдоніми та поштові сервери.

Розглянемо кілька важливих типів записів та їх зміст.

Запис типу NS зображує сервер імен зони, який зберігає базу даних для домену.

- ✓ Записи типу NS і типу SOA для імені зони **критично важливі та обов'язкові** в описі зони; записи інших типів можуть бути відсутніми.
- ✓ Час життя в усіх записах типу NS для зони має бути одним і тим самим.

Запис типу A (Address – адреса) містить 32-розрядну IPv4-адресу інтерфейсу хоста.

- ✓ Саме запис типу A встановлює відповідність між доменним іменем та IP-адресою хоста.

Запис типу AAAA аналогічний типу A, лише містить 128-розрядну IPv6-адресу. Якщо на хості встановлено кілька мережевих з'єднань, то йому потрібні два або більше записів типу A або AAAA. Відповідно, відповідь на запит DNS дає кілька адрес на одне ім'я.

Запис типу MX вказує ім'я хоста, здатного приймати електронну пошту для поточної зони, та його пріоритет у цьому. Таких записів у описі зони може бути кілька.

Запис типу CNAME означає псевдонім для хоста, ім'я якого вказано в записі типу A (так зване **канонічне ім'я**). Зазвичай, записи цього типу забезпечують доступ до серверів за допомогою імен, характерних для відповідних Інтернет-сервісів.

Приклад. Наведемо уявні дані для гіпотетичної зони univ.edu.ua.

```
univ.edu.ua. IN SOA ns admin (
    2020090101  28800  7200  604800  86400  ; див. попередній приклад
)
univ.edu.ua. 86400 IN NS ns          ;   ns.univ.edu.ua -- сервер імен зони
univ.edu.ua. 86400 IN MX 1 mail      ;   перший та другий хости для доставки
univ.edu.ua. 86400 IN MX 2 exim      ;      пошти на ...@univ.edu.ua
ns      86400 IN A 91.201.128.100    ;   IP-адреса хоста ns.univ.edu.ua
mail    86400 IN A 91.201.128.82     ;   IP-адреса хоста mail.univ.edu.ua
exim 86400 IN A 91.201.128.86        ;   IP-адреса хоста exim.univ.edu.ua
www 86400 IN CNAME ns.univ.edu.ua    ;   псевдонім DNS-сервера
ftp 86400 IN CNAME mail.univ.edu.ua  ;   псевдонім хоста mail.univ.edu.ua
somecomp IN A 195.68.211.130         ;   IP-адреса хоста somecomp.univ.edu.ua
kubik IN A 194.68.211.6              ;   дві IP-адреси хоста
kubik IN A 91.201.128.36             ;      kubik.univ.edu.ua
IN MX 1 kubik                        ;   перший та другий хости для доставки
IN MX 2 mail                         ;      пошти на ...@kubik.univ.edu.ua
```

За означеннями в цьому файлі, електронна пошта на адресу вигляду *person*@univ.edu.ua спочатку має надходити на хост mail.univ.edu.ua, а в разі невдачі – на хост exim.univ.edu.ua. Аналогічно й пошта на адресу вигляду *person*@kubik.univ.edu.ua. Відсутність запису типу MX для імені somecomp свідчить, що хост somecomp.univ.edu.ua не може сам отримувати пошту.

Завдяки псевдоніму www, зміна хоста, на якому розташовано Web-сервер (разом з DNS-сервером), не буде вимагати змінювати прив'язку імені www. Те саме стосується FTP-сервера.

Останні чотири рядки містять типові записи для вузла з ім'ям kubik.univ.edu.ua. Доменні імена в останніх двох записах відсутні – це означає, що поточним доменом для них є kubik. ◄

6.6. Зворотний пошук

Зворотний пошук (reverse lookups) – пошук доменного імені за IP-адресою.

Зворотний DNS-запит – запит на пошук доменного імені за IP-адресою.

Зворотний пошук потрібен, наприклад Web- і FTP-серверам, які не дають доступ клієнтам з певних доменів. Отримавши від клієнта запит на установку з'єднання, сервер передає IP-адресу клієнта DNS-серверу як зворотний DNS-запит. DNS-сервер має повернути ім'я клієнтського хоста. За цим іменем приймається рішення, допустити клієнта на сервер чи ні.

Для того, щоб відображати IP-адреси в простір імен, свого часу було створено спеціальний **домен верхнього рівня ARPA** (Address and Routing Parameter Area). Піддоменами в домені ARPA керує організація IAB (Internet Architecture Board – Рада з архітектури Інтернету).

У домені ARPA виділено три піддомени:

- in-addr.arpa для відображення IPv4-адрес у простір доменних імен;
- ip6.arpa для відображення IPv6-адрес у простір доменних імен;
- e164.arpa для відображення телефонних номерів формату E.164.

Зона ARPA підтримується кореневими серверами й серверами зон верхнього рівня, хоча за рекомендаціями RFC 2870 для ARPA бажано виділяти окремі сервери.

Імена в домені in-addr.arpa утворюють ієрархію, відповідну IP-адресам, складові числа яких записано **у зворотному порядку** відносно звичних IP-адрес. Зворотний пошук імен ведеться на основі записів типу **PTR** (від англ. *pointer* – вказівник), які зв'язують IP-адреси хостів з їх канонічними іменами. Наприклад, для хоста knu.ua, що має адресу 91.202.128.77, в описі «зворотної» зони 211.68.194.in-addr.arpa може бути такий запис.

```
77.128.202.91.in-addr.arpa IN PTR knu.ua
```

✓ Зворотний порядок запису дозволяє виконувати зворотний пошук тими самими програмними засобами, що й прямий пошук.

На платформі Windows можна дізнатися доменне ім'я хоста за заданою IP-адресою викликом утиліти **nslookup**, указавши тип запису PTR та IP-адресу. У відповідь утиліта виводить ім'я хоста, а також, можливо, й інші імена.

Приклад. Дамо таку команду з реальною IP-адресою.

```
nslookup  -type = PTR 91.202.128.77
```

У відповіді буде вказано доменне ім'я хоста knu.ua, який має цю адресу.

```
77.128.202.91.in-addr.arpa      name = knu.ua
```

Також будуть виведені доменні імена серверів імен зворотної зони 91.in-addr.arpa – tinnie.arin.net, pri.authdns.ripe.net, sec3.apnic.net, sns-pb.isc.org, а потім їх IP-адреси.

Контрольні запитання

Поняття: DNS-сервер, кореневий DNS-сервер, дзеркало, підтверджений запис, кешований запис, розпізнавач (resolver), рекурсивний запит до сервера імен, віддалений запит, ітеративний запит, часткова відповідь на запит.

Чому кешовані записи не вважаються підтвердженими?

Чому запити до серверів імен і відповіді від них передаються за допомогою UDP, а не TCP?

Описати структуру записів ресурсів.

Записи ресурсів яких типів зустрічаються найчастіше?

Призначення запису типу SOA та параметри зони, які він повідомляє.

Що зображує запис типу NS?

Чому записи типу SOA та NS критично важливі в описі зони?

За яких умов одне й те саме доменне ім’я хоста вказується в кількох записах типу A або AAAA?

Розділ 7. Передача файлів і гіпертекстових даних

7.1. URL

Передача гіпертекстових документів вимагає правильного вказування місцеположення таких документів. Ця задача вирішується за допомогою посилання URL.

URL (англ. *Uniform Resource Locator* – уніфікований локатор ресурсу) – це стандартизований спосіб позначення розташування інтернет-ресурсів (файлів, сторінок, веб-сайтів).

URL був запропонований у 1990 р. Тімом Бернерсом-Лі з CERN (Conseil Européen pour la Recherche Nucléaire, Швейцарія). Стандарт URL описано в RFC 3986, він регулюється організацією IETF та її підрозділами.

✓ URL дозволяє позначати розташування (**локацію**) майже всіх ресурсів Інтернету й не тільки.

Приклади.

1. Локатор україномовної головної сторінки сайту КНУ `http://univ.kiev.ua/ua/#stud` вказує, що ресурс досяжний за протоколом `http`, розташований на хості з доменним ім'ям `univ.kiev.ua` і є україномовним варіантом. Закінчення `#stud` позначає розділ на сторінці, перший рядок якого браузер розміщує на початку вікна відображення.

2. Локатор `https://en.wikipedia.org/wiki/URL` вказує на ресурс, досяжний за протоколом `https`, розташований на хості з доменним ім'ям `en.wikipedia.org`. Шлях `wiki/URL` вказує на розташування ресурсу на хості.

Загальна форма запису URL.

<*схема*>:[//[<*логін*>:<*пароль*>@]<*хост*>[:<*порт*>]][/]<*URL-шлях*>[?<*параметри*>][#<*якір*>]

Тут *схема* – це схема звернення до ресурсу, зазвичай це ім'я мережевого протоколу, найчастіше `http` або `https` (популярні також `ftp`, `file`, `mailto`, `data`, `irc`);

логін – ім'я користувача, що використовується для доступу до ресурсу;

пароль – пароль цього користувача;

хост – повне доменне ім'я хоста в DNS або IP-адреса хоста;

порт – порт хоста для підключення;

URL-шлях – дані про місце розташування ресурсу (вигляд цих даних залежить від протоколу);

параметри – рядок запиту з переданими на сервер (методом GET) параметрами. Рядок починається символом ?, параметри розділяються знаком &;

якір – ідентифікатор «якоря» з попереднім символом #. Якорем може бути вказаний заголовок всередині документа або атрибут `id` елемента. За таким посиланням браузер відкриває сторінку й переміщує вікно до зазначеного елементу.

Повна адреса завжди починається з мережевого протоколу, наприклад, `https` або `http`. Взагалі, існує ще близько 30 протоколів, які використовуються рідше (`ftp`, `irc`, `xmpp` тощо). У більшості сучасних браузерів `http` опускається.

Щоб переконатися, що перед Вами правильно складений URL, можна скопіювати його й вставити у відповідне поле в браузері.

Початково URL було розроблено для використання в браузерах та html-документах. Втім на сьогодні він використовується й у багатьох інших випадках.

✓ URL є частиною більш загальної системи ідентифікації ресурсів URI.

7.2. Передача файлів: протокол FTP

Одним із стандартних протоколів передачі файлів у Інтернеті є **FTP** (**File Transfer Protocol**), розроблений у 1971 р. і зафіксований у RFC 959. FTP є протоколом прикладного рівня й використовується для розповсюдження ПЗ, доступу до віддалених хостів, передачі веб-сторінок та інших документів з місць їх розробки на сервери. Він заснований на архітектурі «клієнт-сервер»: на одному з комп'ютерів працює програма FTP-сервер, на іншому – програма FTP-клієнт, запущена користувачем. Клієнт з'єднується з сервером, передає йому запити на отримання або передачу файлів та отримує файли від нього або передає йому.

Робота FTP-клієнта спочатку керувалася за допомогою командного рядка. Натепер створені численні FTP-клієнти для різних ОС із графічним інтерфейсом: Microsoft Expression Web, FileZilla, Total Commander, Far Manager, Krusader, WinSCP, Cyberduck, SmartFTP, FreshFTP та інші.

Основною особливістю FTP є **множинне з'єднання**. Одне з'єднання є керуючим (**канал керування**) і передає команди з параметрами від клієнта сервера на отримання файлів та відповіді сервера через порт 21. Для передачі даних створюються окремі з'єднання (**канали даних**), по одному на кожну передачу. Звідси, в рамках одного сеансу FTP можна передавати одночасно декілька файлів, причому в обох напрямках. Для кожного каналу даних відкривається свій TCP-порт, номер якого вибирається або сервером, або клієнтом, залежно від режиму передачі – активного або пасивного.

Активний режим. Клієнт створює керуюче TCP-з'єднання з сервером, передає серверу свою IP-адресу й деякий номер свого порту, а далі чекає, коли сервер створить TCP-з'єднання з цією адресою і номером порту для передачі даних.

Пасивний режим. Потрібен, коли клієнт відокремлений брандмауером з блоком NAT, через що він не може отримати вхідне TCP-з'єднання. У цьому режимі клієнт надсилає серверу через потік керування команду PASV й у відповідь на неї отримує IP-адресу та номер порту сервера. Далі клієнт відкриває канал даних з деяким своїм портом та отриманими адресою й портом сервера.

Раніше для передачі даних FTP-сервер використовував тільки порт 20, у сучасних реалізаціях сервер у пасивному режимі може призначати порт з номером більш ніж 1024.

FTP може передавати дані лише кількох різновидів – тексти в кодуванні ASCII, тексти в кодуванні EBCDIC, бінарні дані як послідовність байтів (цей режим є основним). FTP також дозволяє двом комп'ютерам з однаковими налаштуваннями надсилати дані у власному форматі. Користувач FTP може пройти автентифікацію, передавши логін і пароль відкритим текстом у спеціальних командах, або підключитися анонімно, якщо це дозволено на сервері.

FTP також є ресурсом Internet. URL для ftp виглядає так:

ftp: // <*user*>: <*password*> @ <*host*>: <*port*> / <*url-path*>

У цьому локаторі:
user - ім'я користувача,
password - його пароль,
host - доменне ім'я або IP-адреса сервера,
url-path - шлях до файлу.

Синтаксис URL FTP описано в RFC1738. Його формат такий:

ftp: // [<*користувач*> [: <*пароль*>] @] <*хост*> [: <*порт*>] / <*шлях*>

Параметри в квадратних дужках необов'язкові.

Приклади

ftp://public.ftp-servers.example.com/mydirectory/myfile.txt

ftp: // user001: secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt

Більш детально про надання імені користувача та пароля написано в документації браузерів.

На практиці найчастіше використовується **анонімний варіант** ftp. Він нічим не відрізняється від інших, лише в якості імені користувача достатньо вказати **anonymous**, а в якості свого пароля - свою поштову адресу. Для анонімного ftp в url реалізовано спрощений синтаксис:

ftp: // <host> / <url-path>

Отже, за відсутнього імені автоматично вставляється ім'я anonymous. Порт також зазвичай не вказується, а використовується стандартний 21.

Приклади адрес ftp в формі url:

ftp://ftp.cdrom.com/pub/music/songs/1996

ftp://ds.internic.net/rfc/rfc1738.txt

Як host можна вказувати також IP адреси.

Недоліки FTP:

- двопортовий, активний і пасивний режими - протокол важко різати на мережевому екрані (необхідний stateful firewall);
- відсутність поняття кодування що призводить, наприклад, до того, що при скачуванні файлу з сервера в кодуванні cp1251 клієнтом в koi8-r імена і вміст необхідно перекодувати;
- відсутність об'єктно-орієнтованої команди LIST. Існують десятки різних форматів виведення лістингу, і їх доводиться парсити, при цьому висока ймовірність помилки. Наприклад, якщо клієнт і сервер перебувають у різних часових зонах, то одна помилка гарантована - в обробці часу модифікації файлу;
- відсутність вбудованих механізмів шифрування;
- проблеми пересилки файлів, у яких в імені є символ з кодом 255 (у cp1251 це буква "я"), оскільки в стандарті FTP написано, що FTP використовує в якості транспортного протоколу `telnet`. Багато UNIX-серверів це реалізують, але жоден сервер під Windows це не реалізує, а з клієнтів реалізують лише `lftp` і `squid`.

7.3. Протоколи передачі гупертекстових документів HTTP та HTTPS

HTTP (англ. HyperText Transfer Protocol) - стандартний інтернет-протокол прикладного рівня, який використовується відповідними серверами та клієнтами.

Типовим клієнтом для виконання таких завдань є браузер. Втім, його можно замінити стандартним клієнтом telnet. Це може використовуватися для тестування мережі. На жаль, відображення документів за допомогою telnet неможливе.

При виконанні протокола відбувається обмін повідомленнями за схемою «запит-відповідь». Кожен запит/відповідь складається з трьох частин:

- стартовий рядок;
- заголовки;
- тіло повідомлення, що містить дані запиту, запитаний ресурс або опис проблеми, якщо запит не виконано.

Обов'язковим мінімумом запиту є стартовий рядок. Починаючи з HTTP/1.1, обов'язковим став заголовок Host: оскільки необхідно розрізнити кілька доменів, які мають одну й ту саму IP-адресу.

Рядок запиту виглядає так:

<Метод> <URI> HTTP/<Версія>.

Поле <Метод> може набувати таких значень: OPTIONS, HEAD, GET, PUT, POST, PATCH, DELETE, TRACE, CONNECT.

OPTIONS - повертаються методи HTTP, які підтримуються сервером. Цей метод може служити для визначення можливостей вебсервера.

HEAD - аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло. Це корисно для добування метаінформації, заданої в заголовках відповіді, без пересилання всього вмісту. Зокрема, клієнт чи проксі, перевіривши заголовок Last-Modified: (останній час модифікації), таким чином може переконатися, що сторінка на сервері не змінилася від часу попереднього запиту.

GET- отримується вміст вказаного ресурсу. Запитаний ресурс може приймати параметри, наприклад, пошукова система отримує шуканий рядок як параметр. Параметри передаються в рядку URI, наприклад, **http://univ.kiev.ua/resource?p1=v1&p2=v2**. Згідно зі стандартом HTTP, запити типу GET вважаються ідемпотентними — багаторазове повторення одного і того ж запиту GET повинне приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами). Це дозволяє кешувати відповіді на запити GET. Якщо назва ресурсу не вказана (у URI наявні лише схема та доменне ім'я), то вебсервер повертає індекс директорії вебсервера.

PUT - завантажує вказаний ресурс на сервер.

POST - отримується вміст вказаного ресурсу та передає дані користувача (наприклад, з HTML-форми) заданому ресурсу. При цьому передані дані включаються в тіло запиту (request body). На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багаторазове повторення одних і тих же запитів POST може повертати різні результати.

PATCH - завантажує певну частину ресурсу на сервер.

DELETE - видаляє вказаний ресурс.

TRACE - повертає отриманий запит так, що клієнт може побачити, що проміжні сервери додають або змінюють в запиті.

CONNECT - Для використання разом з проксі-серверами, які можуть динамічно перемикатися в тунельний режим SSL.

Перший рядок відповіді має наступний вигляд: HTTP/<Версія> <Код статусу> <Опис статусу>

Коди статусу поділяються на такі класи:

- 1xx — інформаційний: запит прийнятий, продовжуй процес.
- 2xx — успіх: дія була успішно передана, зрозуміла, та прийнята.
- 3xx — перенаправлення: наступні дії мають бути успішно виконані для реалізації запиту.

- 4xx — помилка клієнта: запит містить синтаксичні помилки або не може бути виконаний.
- 5xx — помилка сервера: сервер не зміг виконати правильно сформований запит.

Приклад HTTP діалогу.

Запит:

```
HEAD / HTTP/1.1
```

Host: unicyb.kiev.ua.

Відповідь:

```
HTTP/1.1 301 Moved Permanently
```

Server: nginx

```
Date: Sun, 16 May 2021 13:09:58 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
Location: http://www.unicyb.kiev.ua/
```

✓ HTTPS (англ. HTTP over TLS, HTTP over SSL, чи HTTP Secure) — схема URI, що синтаксично ідентична HTTP. Така схема вказує, що протокол HTTP має використовуватися, але з іншим портом за замовчуванням (443) і додатковим шаром шифрування/автентифікації між HTTP і TCP.

7.4. Протоколи електронної пошти

В термінології електронної пошти виділяються такі компоненти.

MTA (англ. Mail Transfer Agent - агент пересилки пошти) - відповідає за пересилання пошти між поштовими серверами; як правило, перший MTA в ланцюжку отримує повідомлення від MUA, останній передає повідомлення до MDA; можлива реалізація з відправкою пошти через smart host.

MDA (англ. Mail Delivery Agent - агент доставки пошти) - відповідає за доставку пошти кінцевому користувачеві.

MUA (англ. Mail user agent - поштовий агент користувача; в російській нотації закріпився термін поштовий клієнт) - програма, що забезпечує інтерфейс, що відображає отримані листи і надає можливість відповідати, створювати, перенаправляти листи.

MRA (англ. Mail retrieve agent) - поштовий сервер, що забирає пошту з іншого сервера по протоколам, призначеним для MDA.

У разі використання виділених серверів для зберігання пошти користувачів взаємодія користувача з сервером може відбуватися за протоколами, які не вкладаються в цю схему.

Поштові сервери зазвичай виконують роль MTA і MDA. Деякі поштові сервери виконують роль як MTA, так і MDA, деякі мають поділ на два незалежних сервери: сервер-MTA і сервер-MDA. При цьому, якщо для доступу до скриньки використовуються різні протоколи - наприклад, POP3 і IMAP, - то MDA в свою чергу може бути реалізований або як єдиний додаток, або як набір додатків, кожен з яких відповідає за окремий протокол.

Поштовий сервер, сервер електронної пошти, мейл-сервер - в системі пересилки електронної пошти так зазвичай називають агент пересилки повідомлень.

- ✓ Зазвичай поштовий сервер працює «за кулісами», а користувачі мають справу з іншою програмою - клієнтом електронної пошти.

Приклад. Поширеним клієнтом електронної пошти є Outlook Express, однак останнім часом часто використовуються повноцінні версії поштового клієнта від Microsoft - Outlook, а також клієнта від Mozilla - Thunderbird. Коли користувач набрав повідомлення і надсилає його одержувачу, поштовий клієнт взаємодіє з поштовим сервером, використовуючи протокол SMTP. Поштовий сервер відправника взаємодіє з поштовим сервером одержувача (безпосередньо або через проміжний сервер - **релей**). На поштовому сервері одержувача повідомлення потрапляє в поштову скриньку за допомогою агента доставки повідомлень MDA. MDA може бути частиною POP/IMAP сервера (наприклад, deliver і lmtpd у Cyrus-IMAP), або частиною SMTP сервера (наприклад, mail.local чи Procmail у Sendmail), або окремим ПЗ (наприклад Procmail). Для фінальної доставки отриманих повідомлень використовується не SMTP, а інший протокол (часто це POP3 або IMAP), який також підтримується більшістю поштових серверів. Хоча в найпростішої реалізації MTA досить покласти отримані повідомлення в особистий каталог користувача в файлової системі центрального сервера («поштову скриньку»).

Часто поштовий сервер має програмне забезпечення для організації розсилок електронної пошти.

Наведемо більш детальний опис протоколів, що застосовуються.

SMTP (англ. Simple Mail Transfer Protocol, простий протокол передачі пошти) - це широко використовуваний мережевий протокол, призначений для передачі електронної пошти в мережах TCP/IP.

SMTP вперше був описаний в RFC821 (1982). Останнє оновлення в RFC5321 (2008) включає розширення ESMTP (англ. Extended SMTP). Дуже часто під «протоколом SMTP», як правило, мають на увазі і його розширення. Протокол SMTP як правило для передачі вихідної пошти використовує порт TCP за номером 25. В той час, як електронні поштові сервери та інші агенти пересилання повідомлень використовують SMTP для відправки та отримання поштових повідомлень клієнтські поштові програми, що працюють на призначеному для користувача рівні зазвичай використовують SMTP тільки для відправки повідомлень на поштовий сервер для ретрансляції. Для отримання повідомлень клієнтські програми зазвичай використовують або POP3, або IMAP, або патентовані системи (такі як Microsoft Exchange і Lotus Notes / Domino) для доступу до облікового запису своєї поштової скриньки на сервері.

POP3 (англ. Post Office Protocol Version 3, протокол поштового відділення, версія 3) – це стандартний інтернет-протокол прикладного рівня, який використовується клієнтами електронної пошти для отримання електронної пошти з віддаленого сервера по TCP/IP-з'єднанню.

Протокол POP був розроблений в декількох версіях, нинішнім стандартом є третя версія (POP3). Альтернативним протоколом для збору повідомлень з поштового сервера є IMAP.

IMAP (Internet Message Access Protocol) – це стандартний інтернет-протокол прикладного рівня, який використовується клієнтами електронної пошти для отримання електронної пошти з віддаленого сервера по TCP/IP-з'єднанню.

Концепція поштового терміналу має на увазі, що вся кореспонденція, пов'язана з поштовою скринькою (включно з копіями відправлених листів), зберігається на сервері, а користувач звертається до сховища (іноді його за традицією також називають «поштовою скринькою») задля перегляду кореспонденції (як нової, так і архіву) і написання нових листів (включаючи відповіді на інші листи). На цьому принципі діє протокол IMAP і більшість веб-інтерфейсів безкоштовних поштових служб. Подібне зберігання поштового листування вимагає значно

більших потужностей від поштових серверів, в результаті, в багатьох випадках відбувається поділ між поштовими серверами, які пересилають пошту, і серверами зберігання листів.

- ✓ POP3 та IMAP – найбільш поширені Інтернет-протоколи для отримування пошти. Практично всі сучасні клієнти і сервера електронної пошти підтримують обидва стандарти. Більшість постачальників послуг електронної пошти (такі як Hotmail, Gmail, та Yahoo! Mail) також підтримують ці протоколи.

Наведемо відмінності цих двох протоколів. Грунтуючись на роботі протоколів можна розділити їх за двома основними критеріями.

Продуктивність сервера. IMAP більш вимогливий до ресурсів ніж POP3, так як вся робота з обробки пошти (така як пошук) лягає на плечі сервера, POP3 тільки передає пошту клієнта;

Пропускна здатність каналу. Тут IMAP у виграші; POP3 передає тіла всіх листів цілком, тоді як IMAP може передавати окремі частини повідомлень, наприклад, тільки текстову, а решту – за запитом.

В певних умовах сервер зберігання листів може бути налаштований на поведінку, подібну клієнту: такий сервер звертається до сервера електронної пошти по протоколу POP3 і забирає пошту собі. Подібні рішення використовуються зазвичай в малих організаціях, в яких немає інфраструктури для розгортання повноцінних поштових серверів. Тоді використовується локальний сервер для зберігання пошти й поштовий сервер провайдера, що надає послугу отримання пошти за POP3, наприклад, за допомогою fetchmail. Основним недоліком такого рішення є затримка в доставці, оскільки ПЗ, що забирає пошту, звертається до сервера з деякою затримкою. Наприклад, POP3 connector з Exchange 2003 Server в складі Windows SBS не дозволяє через інтерфейс розширеного налаштування виставити інтервал менше 15 хвилин, оскільки надто часті перевірки викликають проблеми з навантаженням на поштовий сервер. Деякі поштові сервери мають засоби захисту від надто частих перевірок.

Контрольні запитання

Які є поштові сервери?

В чому відмінність методів Get та POST?

Чи може бути браузер фтп-клієнтом?

Що таке FTP?

Що таке POP3?

Що таке IMAP?

Що таке HTTP?

Що таке HTTPS?

Посилання на джерела

RFC821 Jonathan B. Postel, “SIMPLE MAIL TRANSFER PROTOCOL”, August 1982, http://www.faqs.org/rfcs/rfc821.html

RFC822 David H. Crocker, “STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES”, August 13, 1982, http://www.faqs.org/rfcs/rfc822.html

RFC826 David C. Plummer, “An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware”, November 1982, http://www.faqs.org/rfcs/rfc826.html

RFC903 Finlayson, Mann, Mogul, Theimer, “A Reverse Address Resolution Protocol”, June 1984, http://www.faqs.org/rfcs/rfc903.html

RFC919 Jeffrey Mogul, “BROADCASTING INTERNET DATAGRAMS”, October 1984, http://www.faqs.org/rfcs/rfc919.html

RFC959 J. Postel, J. Reynolds, “FILE TRANSFER PROTOCOL (FTP)”, October 1985, http://www.faqs.org/rfcs/rfc959.html

RFC1213 K. McCloghrie, M. Rose, “Management Information Base for Network Management of TCP/IP-based internets: MIB-II”, March 1991, http://www.faqs.org/rfcs/rfc1213.html

RFC1700 J. Reynolds, J. Postel, “ASSIGNED NUMBERS”, October 1994, http://www.faqs.org/rfcs/rfc1700.html

RFC1738 T. Berners-Lee, L. Masinter, M. McCahill, “Uniform Resource Locators (URL)”, December 1994, http://www.faqs.org/rfcs/rfc1738.html

RFC1918 Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, “Address Allocation for Private Internets”, February 1996, http://www.faqs.org/rfcs/rfc1918.html

RFC2131 R. Droms, “Dynamic Host Configuration Protocol”, March 1997, http://www.faqs.org/rfcs/rfc2131.html

RFC2132 S. Alexander, R. Droms, “DHCP Options and BOOTP Vendor Extensions”, March 1997, http://www.faqs.org/rfcs/rfc2132.html

RFC2328 J. Moy, “OSPF Version 2”, April 1998, http://www.faqs.org/rfcs/rfc2328.html

RFC2390 T. Bradley, C. Brown, A. Malis, “Inverse Address Resolution Protocol”, September 1998, http://www.faqs.org/rfcs/rfc2390.html

RFC2453 G. Malkin, “RIP Version 2”, November 1998, http://www.faqs.org/rfcs/rfc2453.html

RFC2544 S. Bradner, J. McQuaid, “Benchmarking Methodology for Network Interconnect Devices”, March 1999, http://www.faqs.org/rfcs/rfc2544.html

RFC2821 J. Klensin, “Simple Mail Transfer Protocol”, April 2001, http://www.faqs.org/rfcs/rfc2821.html

RFC2822 P. Resnick, “Internet Message Format”, April 2001, http://www.faqs.org/rfcs/rfc2822.html

RFC2870 R. Bush, D. Karrenberg, M. Kosters, R. Plzak, “Root Name Server Operational Requirements”, June 2000, http://www.faqs.org/rfcs/rfc2870.html

RFC2993 T. Hain, “Architectural Implications of NAT”, November 2000, http://www.faqs.org/rfcs/rfc2993.html

RFC3022 P. Srisuresh, K. Egevang, “Traditional IP Network Address Translator (Traditional NAT)”, January 2001, http://www.faqs.org/rfcs/rfc3022.html

RFC3068 C. Huitema, “An Anycast Prefix for 6to4 Relay Routers”, June 2001, http://www.faqs.org/rfcs/rfc3068.html

RFC3219, Rosenberg, J., Salama, H. and M. Squire, "Telephony Routing over IP (TRIP)", January 2002, http://www.faqs.org/rfcs/rfc3219.html

RFC3508 O. Levin, “H.323 Uniform Resource Locator (URL) Scheme Registration”, April 2003, http://www.faqs.org/rfcs/rfc3508.html

RFC3927 S. Cheshire, B. Aboba, E. Guttman, “Dynamic Configuration of IPv4 Link-Local Addresses”, May 2005, http://www.faqs.org/rfcs/rfc3927.html

RFC3986 T. Berners-Lee, R. Fielding, L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax”, January 2005, http://www.faqs.org/rfcs/rfc3986.html

RFC4271 Y. Rekhter, T. Li, S. Hares, “A Border Gateway Protocol 4 (BGP-4)”, January 2006, http://www.faqs.org/rfcs/rfc4271.html

RFC5321 J. Klensin, “Simple Mail Transfer Protocol”, October 2008, http://www.faqs.org/rfcs/rfc5321.html

RFC5735 M. Cotton, L. Vegoda, “Special Use IPv4 Addresses”, January 2010, http://www.faqs.org/rfcs/rfc5735.html

RFC5737 J. Arkko, M. Cotton, L. Vegoda, “IPv4 Address Blocks Reserved for Documentation”, January 2010, http://www.faqs.org/rfcs/rfc5737.html

RFC5771 M. Cotton, L. Vegoda, D. Meyer, “IANA Guidelines for IPv4 Multicast Address Assignments”, March 2010, http://www.faqs.org/rfcs/rfc5771.html

RFC6598 J. Weil, V. Kuarsingh, C. Donley, C. Liljenstolpe, M. Azinger, “IANA-Reserved IPv4 Prefix for Shared Address Space”, April 2012, http://www.faqs.org/rfcs/rfc6598.html