Розділ 11. Інтермережі

11.1. IP-мережі і TCP/IP-мережі

Практично всі мережі є на сьогодні складеними, тобто містять у собі декілька мереж, кожна з яких може працювати на основі власної технології канального рівня. Це обумовлено тим, що більшість мереж створювали поступово, об'єднуючи та долучаючи щоразу більшу кількість ізольованих раніше цього сегментів.

З розвитком глобальної мережі Інтернет окремих мереж, які не є складеними, можна казати май же зовсім не стало. Навіть мережі малих офісів та домашні мережі все частіше стають постійними або тимчасовими (на час під'єднання по лінії доступу) членами цієї найбільшої в світі складеної мережі.

Відносно складеної мережі ми вживаємо термін «**інтермережа**», який визначає сукупність логічних мереж, що взаємодіють між собою на основі протоколів та устаткування мережевого рівня (див. підрозділ 5.4). У межах інтермережі логічні мережі з'єднують за допомогою маршрутизаторів, основне призначення яких полягає в передаванні даних із однієї логічної мережі в іншу. Функції маршрутизаторів можуть виконувати як спеціалізовані пристрої, так і універсальні комп'ютери з відповідним програмним забезпеченням. Компонентами інтермережі можуть бути як локальні, так і глобальні сегменти, внутрішня структура яких не має принципового значення для протоколу мережевого рівня.

Для стеку TCP/IP основним протоколом мережевого рівня, як уже зазначено, є **протокол міжмережевої взаємодії – інтернет- протокол** (Internet Protocol, **IP**), що дає підстави називати інтермережу також **IP-мережею**.

Мережевий рівень функціонує як *координатор роботи* всіх логічних мереж на шляху проходження пакету по складеній IP-мережі. Для переміщення даних у межах окремих логічних мереж мережевий рівень звертається до використовуваних у них технологій канального рівня. Протоколи мережевого рівня реалізуються, як правило, у вигляді програмних модулів і виконуються на кінцевих вузлах, які називають також хостами, та на проміжних вузлах – шлюзах, інакше – маршрутизаторах.

IP-мережа за функціональною ознакою є *телекомунікаційною мережею*, в якій передавання трафіку здійснюється IP-пакетами.

Реалізація функцій прикладного рівня на базі IP-мережі забезпечує її *сервісні можливості* з *формування послуг* та *застосовань*. Зосереджуючи увагу саме на цьому, будемо використовувати термін «**TCP/IP-мережа**». За функціональною ознакою TCP/IP-мережа є *інформаційною мережею,* класичним прикладом якої є глобальний Інтернет, де **інтернет-сервіс-провайдинг** (Inetrnet Service Providing, **ISP**) – це особливий вид діяльності, відокремлений від діяльності мережевих операторів.

11.2. Протокол міжмережевого взаємодії

Протокол міжмережевої взаємодії (Internet Protocol, **IP**), описаний у документі REF 791, є *основним протоколом мережевого рівня* стеку протоколів TCP/IP.

IP – це неорієнтований на налаштування з'єднання та ненадійний протокол передавання. Термін «*неорієнтований на налаштування з'єднання*» означає, що сеанс для обміну даними не встановлюється. Термін «*ненадійний*» означає, що доставка не гарантується. Хоча IP докладає всіх зусиль, щоб доправити пакет, IP-пакет може бути втрачено, доправлено поза чергою, продубльовано або затримано. Протокол IP не може виправляти помилки таких типів. Підтвердження про отримання пакетів і повторне звернення за втраченими пакетами є обов'язками протоколу більш високого рівня, наприклад, TCP.

IP-адреса

Формат IP-адреси стандартний і визначений протоколом IP, тому адреси комп'ютерів ще називають IP-адресами.

IP-адреса комп'ютера складається з чотирьох полів, які відокремлюють крапкою. Кожне поле містить число, значення якого лежить у межах від 0 до 255. Такий формат називають *точково-десятковою нотацією*. Для зберігання даних у обчислювальній техніці засьосовують двійкові числа, тому IP-адресу можна подати в двійковому вигляді:

двійковий формат – 11000000 10101000 00000011 00011000

(десятковий формат – 192.168.3.24)

У двійковому форматі IP-адреса складається з 32 бітів, які розбиті на чотири октети (поля по 8 біт). Щоб точно вказувати місцезнаходження комп'ютера в мережі, IP-адресу розділено на дві частини, одна містить номер мережі, інша – номер комп'ютера в цій мережі.

Для того, щоб відокремити в IP-адресі поля, пов'язані з номером мережі від полів номера вузла, комп'ютерні мережі поділяють на три основні класи: A, B і C. Класи істотно відрізняються один від одного за розмірами та складністю. Вони визначають, скільки біт в IP-адресі відводиться під номер мережі та скільки під номер вузла.

Клас A. Мережа класу A має адреси, які починаються з числа від 1 до 127 для першого октету, інша частина адреси – це адреса сайту. Таким чином клас A допускає максимально 126 мереж, а в кожній з них до 16 777 214 комп'ютерів. Як правило, це мережі величезних компаній, яких у світі небагато, що об'єднують велику кількість мережевих пристроїв.

Клас B. У мережі класу B для опису адреси мережі використовують перші два октети, а інша частина – це адреси вузлів. Перший октет приймає значення від 128 до 191, що дає максимально 16 384 мережі, в кожній з яких до 65 534 вузла. Адреси класу B призначено для мереж великого й середнього розміру.

Клас C. Адреси мереж класу C починаються з цифри від 192 до 223 та використовують три перших октет для опису адреси мережі. Останній октет позначає адресу сайту. Таким чином, клас C допускає максимально 2 097 152 мережі, по 254 комп'ютери в кожній. Адресу цього класу призначають малим мереж.

Адреса мережі класу A, що починається на 127, зарезервовано для тестування і є недоступною для використання.

Адреси класу D – це групові адреси, які закріплюють за групами вузлів. Це використовують деякі служби для так званої багато адресної розсилки. Діапазон адрес класу E зарезервовано, в даний час його не застосовують.

Один і той самий фізичний пристрій (комп'ютер та ін) може мати декілька IP-адрес, тобто відповідати декільком логічним вузлам. Зазвичай, така ситуація виникає, якщо пристрій має кілька мережевих адаптерів і/або модемів, оскільки з кожним з них повинен бути пов'язаний як мінімум одна унікальна IP-адреса. Хоча нерідко комп'ютеру, який має один мережевий адаптер або модем, може бути присвоєно декілька IP-адрес. Якщо фізичний пристрій має кілька IP-адрес, то говорять, що він має декілька інтерфейсів, (точок під'єднання до логічної мережі).

Підмережі та маски підмереж

Підмережу в даному випадку розуміють як окрему, самостійно функціонуючу частини IP-мережі одного класу, яка під'єднується, як правило, через маршрутизатор. Мережа класу A допускає наявність понад 16 мільйонів вузлів. Уявити собі таку мережу дуже складно, а працювати в ній неможливо через те, що мережеве обладнання просто не впорається з такою кількістю переданих пакетів. У зв'язку з цим IP-мережу можна, як відомо, розбити на кілька логічних мереж (підмереж), об'єднавши їх маршрутизаторами та присвоївши

кожній із них свій ідентифікатор мережі. В одному мережевому класі може існувати безліч підмереж.

Для налаштування підмережі використовують **маску підмережі**, що призначена для визначення адреси мережі незалежно від класу мережі. Формат запису маски підмережі однаковий з форматом IP-адреси: це чотири двійкових октети або чотири поля, розділених крапкою. Значення полів маски задають так: усі біти, встановлені в 1, відповідають ідентифікатором мережі; всі біти, встановлені в 0, відповідають ідентифікатору вузла (див. таблицю 11.1).

Таблиця 11.1

Клас мережі	Біти маски підмережі	Маска підмережі
A	11111111 00000000 00000000 00000000	**255.0.0.0**
B	11111111 11111111 00000000 00000000	**255.255.0.0**
C	11111111 11111111 11111111 00000000	**255.255.255.0**

Будь-який вузол у мережі потребує наявності маски підмережі. Маска не є IP-адресою вузла, вона лише описує адресний простір підмережі: з якої адреси починається підмережа та якою закінчується. Якщо в одній фізичній мережі працюватимуть комп'ютери з різною маскою, то вони не побачать один одного.

Наприклад, IP-адреса комп'ютера – 192.168.0.1 і маска під мережі – 255.255.255.0, тоді номер мережі –192.168.0, а номер комп'ютера – 1. Якщо локальна мережа складається з п'яти комп'ютерів, то IP-адреси комп'ютерів записують так:

192.168.0.1 маска 255.255.255.0
192.168.0.2 маска 255.255.255.0
192.168.0.3 маска 255.255.255.0
192.168.0.4 маска 255.255.255.0
192.168.0.5 маска 255.255.255.0

Використання в парі з IP-адресою маски підмережі дає змогу відмовитися від застосування класів адрес та зробити більш гнучкою всю систему IP-адресації. Так, наприклад, маска 255.255.255.240 (11111111 11111111 11111111 11110000) допомагає розбити діапазон в 254 IP-адреси, які належать одній мережі класу С, на 14 діапазонів, які можуть призначатися різним мережам.

Таким чином, якщо номер мережі – 192.168.0 і маска мережі – 255.255.255.240, то діапазони підмереж будуть:

192.168.0.16 – 192.168.0.30
192.168.0.32 – 192.168.0.46
192.168.0.48 – 192.168.0.62
192.168.0.64 – 192.168.0.78
192.168.0.80 – 192.168.0.94
192.168.0.96 – 192.168.0.110
192.168.0.112 – 192.168.0.126
192.168.0.128 – 192.168.0.142
192.168.0.144 – 192.168.0.158
192.168.0.160 – 192.168.0.174
192.168.0.176 – 192.168.0.190
192.168.0.192 – 192.168.0.206
192.168.0.208 – 192.168.0.222
192.168.0.224 – 192.168.0.238

Загальні та приватні адреси

Усі IP-адреси поділяють на дві групи: загальні та приватні. Загальні адреси використовують на комп'ютерах, безпосередньо під'єднаних до Інтернету. Приватні IP-адреси – на комп'ютерах, під'єднаних до внутрішніх локальних мереж.

Доступ до Інтернету для всіх комп'ютерів локальної мережі в більшості випадках забезпечує тільки один комп'ютер. Такий комп'ютер налаштовано відразу на дві IP-адреси: одну – приватну, іншу – загальну.

Приватний адресний простір визначають такими адресними блоками:

від 10.0.0.1 до 10.255.255.254
від 172.16.0.1 до 172.31.255.254
від 192.168.0.1 до 192.168.255.254

Ці адреси використовують у локальних мережах невеликих організацій, що не потребує реєстрації. Комп'ютерні мережі з приватними адресами можуть під'єднуватися до Інтернету через Інтернет-провайдера.

Якщо кількість комп'ютерів у мережі не перевищуватиме 254, то рекомендовано використовувати адреси з діапазону від 192.168.0.1 до 192.168.0.254 з маскою підмережі 255.255.255.0. Тоді 192.168.0 буде номером мережі, а адреси комп'ютерів – від 1 до 254.

Якщо комп'ютерів буде понад 254, то можна використовувати діапазон від 192.168.0.1 до 192.168.255.254 з маскою підмережі 255.255.0.0. Тоді 192.168 буде номером

мережі, а адреси комп'ютерів – від 0.1 до 255.254 (це більше 65 000 адрес).

Адресні блоки 10.0.0.1 і 172.16.0.1 призначено для більш великих комп'ютерних мереж.

Якщо в комп'ютері налаштовано кілька мережевих адаптерів, то кожен адаптер повинен мати свою унікальну IP-адресу. Такі комп'ютери використовуються для з'єднання декількох локальних мереж і називаються маршрутизаторами (Router).

Для побудови мереж, що складають глобальну мережу Інтернет, вибирають чітко визначені діапазони адресів, які призначаються **Адміністрацією адресного простору Интернет** (Internet Assigned Numbers Authority, **IANA**). IANA, яка є підконтрольною ICANN, «найвищій інстанції» в питаннях резервування діапазонів адрес, що має свої представництва по всьому світу. Наприклад, у Європі розподіл адрес координує RIPE NCC.

Динамічні та статичні IP-адреси. DHCP

Основною аксіомою IP-адресації є необхідність дотримання унікальності IP-адрес у всьому просторі мережі, оскільки, перш за все, цим забезпечується коректність доставки даних і маршрутизації. Закріплюють IP-адресу комп'ютера або вручну (статична адреса), або комп'ютер отримує її автоматично з серверу (динамічна адреса). Статичну адресу прописує адміністратор мережі в налаштуваннях протоколу TCP/IP на кожному комп'ютері мережі та жорстко закріплює за комп'ютером. У цьому процесі є певні незручності, а саме:

- адміністратор мережі повинен вести облік усіх використовуваних адрес, щоб уникнути повторів;
- якщо кількість комп'ютерів у локальній мережі є значною, то налаштування IP-адрес стає довготривалим.

Однак, незважаючи на зазначені незручності, статичні адреси мають одну важливу перевагу – постійну відповідність IP-адреси певному комп'ютера. Це дає змогу ефективно застосовувати політику IP-безпеки й контролювати роботу користувачів у мережі. Наприклад, можна заборонити певному комп'ютеру виходити в Інтернет або з'ясувати з якого комп'ютера виходили в Інтернет та ін.

Якщо комп'ютеру не присвоєно статичну IP-адресу, то адреса призначається автоматично. Цю адресу називають динамічною, тому що при кожному під'єднанні комп'ютера до локальної мережі адреса може змінюватися. Перевагами динамічних адрес є такі:

- централізоване керування базою IP-адрес;
- надійне налаштування, що вилучає ймовірність дублювання IP-адрес;
- спрощення мережевого адміністрування.

Динамічну IP-адреса призначає спеціальна **серверна служба -динамічений протокол конфігурування хоста** (Dynamic Host Configuration Protocol, **DHCP**). У параметрах служби DHCP адміністратор мережі прописує IP-діапазон, адреси з якого, видаватимуть іншим комп'ютерам.

Серверна служба DHCP, яка поширює IP-адреси, називаэться **DHCP-сервером**, а комп'ютер, який отримує (орендує) IP-адресу з мережі, називаэться **DHCP-клієнтом**.

Оскільки протокол DHCP призначено для функціонування в мережах з неналаштованою IP-взаємодією, то він є немаршрутувальним. Щоб забезпечити можливість проходження DHCP-пакетів через маршрутизатори, використовують додаткові функціональні модулі (реалізовані програмно чи апаратно), які називаються агентами ретрансляції BOOTP (BOOTP relay agent). Маршрутизатор, який виконує функції такого ретранслятора, приймає з мережі DHCP-пакети й направляє їх у інші мережі.

Зауважимо, що операційна система Windows XP Professional не містить служби DHCP-сервер. До складу Windows XP входить локальна **служба автоматичного призначення IP-адрес** (Internet Assigned Numbers Authority, **IANA**). У зв'язку з відсутністю в мережі DHCP-серверу комп'ютер із налаштованою ОС Windows XP Professional звертається до вбудованої функції автоматичного призначення IP-адреси та здійснює самоналаштування IP-адреси й маски підмережі, використовуючи одну із зарезервованих адрес. Зарезервовані адреси призначаються з діапазону 169.254.0.0 до 169.254.255.255 з маскою підмережі 255.255.0.0.

Функція автоматичного призначення IP-адреси гарантує унікальність IP-адреси, яку надають. Ця функція працює на локальному комп'ютері та не забезпечує IP-адресами інші комп'ютери мережі.

Доменні імена

Аппаратне і програмне забезпечення в IP-мережі для ідентифікації комп'ютерів використовує IP-адреси. Але користувачи завжди віддають перевагу роботі з найбільш зручними *символьними іменами*, так званими **доменними іменами** комп'ютерів.

Символьні ідентифікатори мережевих інтерфейсів в межах складеної мережі будуються за *ієрархічною ознакою*. Складові повного доменного (символьного) імені в IP-мережі разподіляються кропкой і перелічуються у наступному порядку: спочатку просте ім'я хоста, далі ім'я групи хостів (наприклад, ім'я організації), далі ім'я більш крупної групи (домена) і так до рівня домена найвишчого рівня (наприклад, домена, який об'єднює організації за географічним принципом: UA – Україна, UK – Великобританія, US – США).

Для регламентації й забезпечення використання доменних імен в Інтернеті створена спеціальна **DNS-служба**, яка грунтується на разподіленій базі відображення «доменне ім'я - IP-адреса», тому доменні імена называют ще **DNS-іменами.** Крім того, існує узгодження про використання міжднародних DNS-імен.

Служба DNS застосовує у своей роботі **DNS-сервери** и **DNS-клієнти**. DNS-сервери підтримують расподилену базу відображень, а DNS-клієнти звертаються до серверів з запитом про розв'язання доменного імені в IP-адресу. Служба DNS спирається на ієрархію доменів, і кожний сервер службы DNS зберегає лише частину імені, а не всі імена. При зростанні кількості вузлів в мережі проблема маштабування

вирішується шляхом створення нових доменів і піддоменів імен та долучення в службу DNS нових серверів.

Між доменним іменем і IP-адресою вузла немає ніякої функціональної залежності, тому єдиний спосіб встановлення відповідності – це таблиця відображень, яка створюється адміністраторами мережі.

У загальному випадку мережевий інтерфейс може мати декілька локальних адрес, мережевих адрес і доменних імен.

Формат IP-пакету (датаграми)

IP-пакет – це форматований блок інформації, який передається IP-мережею, складений із заголовка й текстової частини. Для IP протоколу розроблено дві версії: четверта – **IPv4** і шоста – **IPv6**. На рисунках 11.1 и 11.2 наведено формати IP-пакетів відповідно зазначених версій.

Рисунок 11.1. Формат пакету версії IPv4

Призначення полей:

- **Версія** – інформація про версію протоколу. Для IPv4 значення поля має дорівнювати 4 бітам.

- **IHL** – довжина заголовка IP-пакету в 32-бітових словах (dword). Саме це поле вказує на початок блоку даних у пакеті. Мінімальне коректне значення для цього поля дорівнює 5бітам.
- **Ідентифікатор** – значення, яка призначається відправником пакету та призначене для з'ясування коректної послідовності фрагментів при складанні датаграми. Для фрагментованого пакету всі фрагменти мають однаковий ідентифікатор.
- **Прапори** – 3 біти. Перший біт повинен завжди дорівнювати нулю, другий біт DF (don't fragment) визначає можливість фрагментації пакета, а третій біт MF (more fragments) показує, чи не є цей пакет останнім у ланцюжку пакетів.
- **Зміщення фрагмента** – значення, що визначає позицію фрагмента в потоці даних.
- **Протокол** – ідентифікатор Інтернет-протоколу наступного рівня. В IPv6 називається «Next Header».

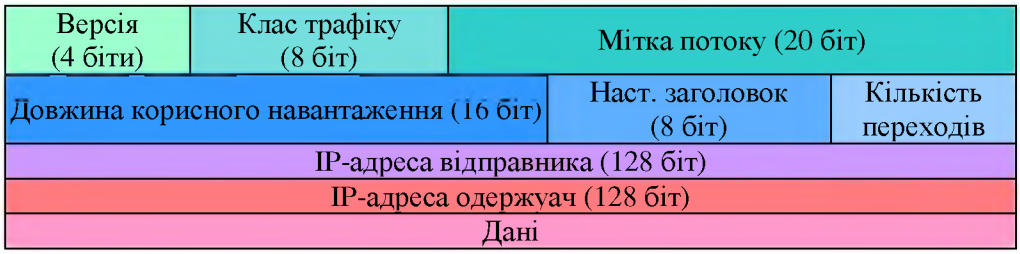

Рисунок 11.2. Формат пакету версії IPv6

Призначення полей:

- **Версія** – для IPv6 значення поля має дорівнювати 6 біт.

- **Клас трафіку** визначає пріоритет трафіку (QoS, клас обслуговування).
- **Мітка потоку** – унікальне число, однакове для однорідного потоку пакетів.
- **Довжина корисного навантаження** – довжина даних (заголовок IP-пакету не враховується).
- **Наступний заголовок** визначає наступний інкапсульований протокол.
- **Кількість переходів** – максимальна кількість маршрутизаторів, які може пройти пакет. При проходженні маршрутизатора це значення зменшується на одиницю, досягнувши 0,пакет пакет відкидається.

Версію IPv6 призначено для роботи в мережах значно більших, ніж ті, в яких сьогодні працює IPv4.

Версію IPv4 розроблено в 1981 році й подано в специфікації RFC 791. Вона є найбільш розповсюдженою. Її відминна властивість полягає у використанні 32-бітової IP-адреси для глобальної адресації пакетів. Однак, як було з'ясовано пізніше, 32-бітових адрес недостатньо для стрімкого зростання Інтернету. З цього приводу в 1995 році з'явилась версия IPv6 (RFC 1752), де використовуються 128-бітові IP-адреси. Шоста версія IP-протоколу застосовується лише в окремих країнах, де Інтернет з'явився пізніше, і яким не вистачило 32-бітових IP-адрес.

Протокол розв'язування адрес ARP

Обмін усередині окремої логічної мережі, яка є частиною IP-мережі, здійснюється так саме, як і в глобальній IP-мережі, тобто з використанням IP-адреси. При цьому виникає необхідність втановлення відповідності між IP-адресою і фізичною адресою (MAC-адресою) пристрою. Ця процедура має назву **розв'язування адрес**. Її функції реализуються спеціальним **протоколом разв'язування адрес** (Address Resolution Protocol, **ARP**), який затверджено в документі RFC 826.

Протокол ARP формально складається з двох частин. Одна частина протокола визначає фізичні адреси при посиланні ARP-запита (служебового пакету), друга відповідає на ARP-запипти від інших пристроїв мережі. У свойому функціонуванні протокол ARP передбачає, що кожний присрій знає як свою IP-адресу, так і свою фізичну адресу.

Для того, чтоб зменьшити кількість посилань ARP-запитов, кажний пристрій в мережі, який використовує протокол ARP, повинен мати спеціальну буферну пам'ять. У цій пам'яті зберегаються відомості про пари IP- і фізичних адресо інших пристроїв в мережі. Кожного разу, коли пристрій получае ARP-відповідь, він зберегає у цей пам'яти дані про зв'вязок адрес. Як що такі дані є, то немає потреби надсилати ARP-запит. Така буферна пам'ять має назвау «**ARP-таблиця**». В ARP-таблиці можуть бути як статичні, так і динамічни записи.

Динамічні записи додаються і вилучаються автоматично. Кожний запис має свій потенційний час існування. Після того як запис був додан до таблиці, йому надається часовий таймер.

Як що запис не використовується протягом перших двох хвилин, він вилучається. Як що навпоки - час його існування становить 10 хвилин.

Статичні записи можуть бути додані користувачем й будуть залишитися в таблиці до перезавантаження комп'ютера.

11.3. Організаційна структура Інтернету

Більшість протоколів маршрутизації, які застосовують у сучасних мережах з комутацією пакетів, виникли завдяки Інтернету. Для того, щоб зрозуміти їхнє призначення та особливості, корисно ознайомитися з організаційною структурою цієї глобальної мережі.

Інтернет від самого початку створено як мережу, яка об'єднує велику кількість незалежних систем. У його структурі виокремлювали **магістральну мережу – ядро** (Core Backbone Network, **CBN**), а під'єднані до магістралі мережі розглядали як **автономні системи** (Autonomous System, **AS**) (див. рис. 11.3).

Магістраль і кожна з автономних систем мали своє власне адміністративне керування та протоколи маршрутизації. Необхідно звернути увагу на те, що розподілення на автономні системи не пов'язано прямо з розподіленням Інтернету на мережі та домени імен. **Автономна система** об'єднує мережі, де маршрутизація здійснюється під загальним адміністративним керівництвом однієї організації, а **домен імен** - єдиний для комп'ютерів (можливо тим, які належать різним мережам), для яких

призначення унікальних символьних імен відбувається під таким же керівництвом.

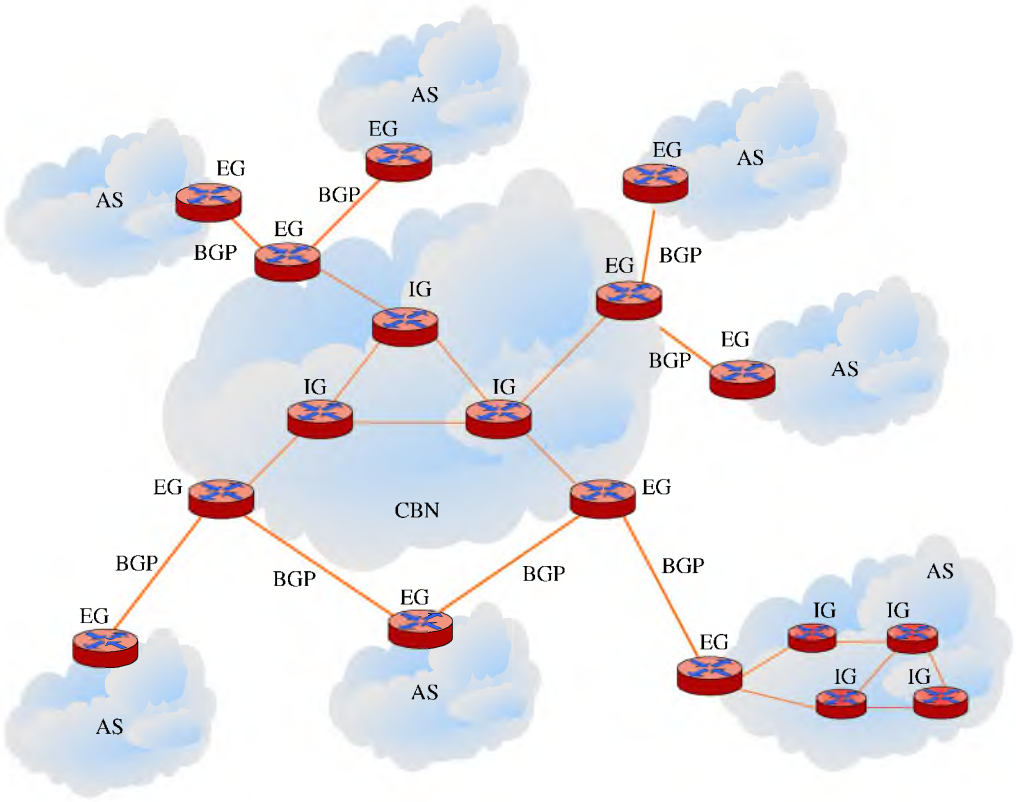

Рисунок 11.3. Організаційна структура Інтернету

CBR - маршрутна мережа
AS - автономна система
IG - внутрішній шлюз
EG - зовнішний шлюз
BGP -протокол суміжної маршрутизації

Природно, що сфера дії автономної системи та домену імен можуть іноді збігатися, якщо одна організація виконує обидві зазначені функції.

Маршрутизатори, які застосовуються для формування інтермережі усередині автономної системи, називаються **внутрішніми шлюзами** (Interior Gateway, **IG**), а ті, за допомогою яких автономні системи під'єднуються до магістралі CBN, – **зовнішніми шлюзами** (Exterior Gateway, **EG**). Сама магістраль CBN також є автономною системою.

Усі автономні системи мають спеціальний унікальний 16-розрядний номер, який присвоюється централізовано відповідним адміністративним органом Інтернету, де реєструють усі AS.

Протоколи, що використовуються всередині автономних систем маршрутизації називаються **протоколами внутрішніх шлюзів** (Interior Gateway Protocol, **IGP**), а протоколи обміну маршрутної інформацією між зовнішніми шлюзами автономних систем і шлюзами магістральної мережі CBN – **протоколами зовнішніх шлюзів** (Exterior Gateway Protocol, **EGP**). Усередині магістральної мережі CBN також може функціонувати будь-який власний внутрішній протокол IGP.

Поділ усього Інтернету на автономні системи є необхідним для *багаторівневої модульної організації*, що уможливлює розширення будь-якої великої системи. Зміна протоколів маршрутизації всередині якої-небудь автономної системи не повинна впливати на роботу інших автономних систем. Крім того, поділ Інтернет на автономні системи сприяє агрегуванню інформації на магістральних та зовнішніх шлюзах. Внутрішні шлюзи можуть використовувати для внутрішньої маршрутизації досить детальні графи взаємних зв'язків, щоб вибрати найбільш раціональний маршрут. Однак, якщо інформація такого ступеня деталізації зберігатиметься в

усіх маршрутизаторах мережі, то топологічні бази даних настільки розростуться, що буде потрібною пам'ять гігантських розмірів, а час прийняття рішень про маршрутизацію стане неприйнятно тривалим. Тому детальна топологічна інформація залишається всередині автономної системи, яку зовнішні шлюзи подають для іншої частини Інтернет як єдине ціле. Вони повідомляють про внутрішній склад автономної системи мінімально необхідні відомості: кількість IP-мереж, їх адреси та внутрішню відстань до цих мереж від даного зовнішнього шлюзу.

Структура Інтернет з єдиною магістраллю, яка наведена на рисунку 11.3, була такою досить довго, тому спеціально для неї було розроблено протокол маршрутного обміну інформацією між AS, названий EGP. Однак з розвитком інтернет-сервіс-провайдингу структура Інтернету ускладнилася і зв'язкі між автономними системами стали довільними. В наслідок цього протокол EGP поступився місцем **протоколу прикордонної маршрутизації** (Border Gateway Protocol, **BGP**), який дає змогу розпізнавати наявність петель між автономними системами та вилучати їх з міжсистемних маршрутів.

Великомасштабні автономні системи, що складаються із сотень вузлів, можна поділяти на підсистеми (більш дрібні AS). Таке утворення називають **конфедерацією**. Організовувати конфедерації рекомендовано в тих випадках, коли в роботу за протоколом BGP залучено велику кількість маршрутизаторів, що викликає лавиноподібне наростання кількості BGP-сеансів на окремому маршрутизаторові. Всередині кожної такої підсистеми AS чинними є всі правила

маршрутизації за IGP. Оскільки, кожна підсистема AS має власний номер, вони можуть взаємодіяти із зовнішнім протоколом BGP.

11.4. Методи та протоколи маршрутизації

Найважливішим завданням мережевого рівня є маршрутизація – передавання пакетів між двома кінцевими вузлами в складеній мережі. Завдання щодо вибору маршрутів із декількох можливих вирішують маршрутизатори, а також кінцеві вузли. Маршрут вибирають на підставі наявної у цих пристроїв інформації про поточну конфігурацію мережі, а також з урахуванням зазначеного критерію вибору маршруту. Зазвичай, таким критерієм є затримка проходження маршруту окремим пакетом або середня пропускна здатність маршруту для послідовності пакетів. Часто використовують досить простий критерій, який враховує тільки кількість транзитних маршрутизаторів. Завдання маршрутизації вирішується на основі аналізу відповідних таблиць, розміщених на всіх маршрутизаторах і кінцевих вузлах мережі.

Таблиці маршрутизації

Рішення про просування пакету маршрутизатор приймає на основі **таблиці маршрутизації**. Побудова коректних таблиць маршрутизації на всіх маршрутизаторах у великій складеній мережі є трудомістким завданням. Таблицю маршрутизації для відображення поточної мережевої топології необхідно періодично коригувати, тобто керувати динамічно.

Маршрутизатор виконує це завдання, беручи участь у процесі обмінювання інформацією про маршрути з іншими маршрутизаторами на основі **протоколів маршрутизації**. Протоколами маршрутизації, які забезпечують обмінювання інформацією про маршрути в мережах на базі протоколу IP, є протоколи RIP, OSPF, EGP і BGP, які розглядатимемо в п. 11.4. Залежно від структури складеної мережі деякі маршрутизатори можуть одночасно підтримувати роботу декількох протоколів маршрутизації.

Будь-який із маршрутизаторів має не менше ніж два порти для під'єднання різних логічних мереж. Кожен порт можна розглядати як окремий вузол мережі, оскільки він має власну *мережеву адресу* та власну *локальну адресу* в кожній мережі, яку до нього під'єднано.

Маршрутизатори самі ініціюють обмін спеціальними службовими пакетами, повідомляючи сусідам про відомі їм мережі, маршрутизатори та про зв'язки цих мереж з маршрутизаторами. Зазвичай до уваги береться не тільки топологія зв'язків, але й їхня пропускна здатність і стан. Це дає змогу маршрутизаторам швидше адаптуватися до змін конфігурації мережі, а також правильно передавати пакети в мережах з довільною топологією, що допускає наявність замкнутих контурів.

За допомогою протоколів маршрутизації маршрутизатор складає «карту» зв'язків мережі різної деталізації. На підставі цієї інформації для кожного номера мережі приймається рішення про те, якому наступному маршрутизатору треба передавати пакети, які прямують за цією адресою, щоб маршрут був раціональним. Результати таких рішень заносять

до таблиці маршрутизації. У таблиці 11.2 наведено приклад простої таблиці маршрутизації. У цій таблиці розміщено типові записи для протоколів маршрутизації, таких, наприклад, як RIP, які використовують в якості метрики маршруту кількість переходів, так званих **хопів** (hop count). У деяких технічних джерелах можна зустріти також термін «транзитні вузли», під яким розуміють кількість маршрутизаторів, через які повинен пройти пакет до приходу до одержувача.

Таблиця 11.2

Номер мережі одержувача 128.3.0.0			
Наступний маршрутизатор на шляху	**Кількість переходів**	**Протокол маршрутизації**	**Часовий таймер**
128.5.3.2	3	RIP	145
128.5.4.7	3	RIP	170
128.5.3.9	6	RIP	25

Кожен запис у таблиці маршрутизації містить таку інформацію:

- *наступний маршрут на шляху* містить IP-адресу віддаленого маршрутизатора, якому необхідно послати пакети для доставки їх за адресою призначення;
- *кількість переходів* визначає кількість переходів (хопів) між даним маршрутизатором і одержувачем пакетів;
- *протокол маршрутизації* визначає протокол маршрутизації, за допомогою якого запис з'явився в таблиці маршрутизації;

- *часовий таймер* визначає час з моменту останнього поновлення запису в таблиці. Цей таймер скидує показання щоразу після отримання оновлення.

В основному маршрутизатори керують таблицею маршрутизації, яка містить тільки один маршрут для кожної мережі призначення. Деякі протоколи маршрутизації можуть керувати більш ніж одним маршрутом у певну мережу (прикладом є протокол OSPF).

Маршрутизатори повинні перевіряти свої таблиці маршрутизації, відшукуючи шлях для кожного пакету. Зміна конфігурації мережі деякі записи в таблиці спростовує. У подібних випадках пакети можуть бути відправлені за помилковими маршрутами, можуть зациклюватися та губитися. Якщо маршрут неможливо знайти, то маршрутизатору, який виконує пошук, необхідно видалити пакет з обігу.

Існує спеціальна IP-адреса 0.0.0.0, яка позначає маршрут за замовчуванням. Якщо потрібний шлях не знайдено й визначено маршрут за замовчуванням, маршрутизатор не буде видаляти пакет, а передасть його за цим маршрутом. Введення поняття маршруту за замовчуванням дає змогу значно зменшити розмір таблиць маршрутизації. У результаті процес маршрутизації спрощується, оскільки таблиця маршрутизації містить кілька записів для локальних мереж і маршрут за замовчуванням для всіх інших. Особливо помітною є перевага для під'єднання до Інтернету організацій. Окрім того використання маршруту за замовчуванням значно зменшує обсяги інформації, якими обмінюються маршрутизатори.

Недоліком маршруту за замовчуванням є потенційна можливість утворення петель маршрутизації.

Слід зазначити, що таблиця маршрутизації існує не тільки в маршрутизаторів із декількома портами, а й у хостів, які під'єднуються до мережі через один мережевий адаптер. Відмінність у цій ситуації полягає в тому, що всі пакети необхідно видавати тільки через один-єдиний порт, незалежно від їх адреси призначення.

Автоматична побудова таблиць маршрутизації

Від того, наскільки швидко вміст таблиці маршрутизації приводиться у відповідність до реального стану мережі, залежить якість роботи всієї мережі. Для автоматизації цього процесу в складених мережах розроблено вищезазначені **протоколи маршрутизації** (або маршрутизувальні протоколи). Ці протоколи слід відрізняти від власне *мережевих протоколів* (наприклад, IP). Як перші, так другі забезпечують функції мережевого рівня: беруть участь у доставці пакетів адресатові через різнорідну складену мережу. Але в той час, як перші збирають і передають мережею тільки службову інформацію, другі призначено для передавання даних користувачеві. Протоколи маршрутизації використовують мережеві протоколи як транспортний засіб.

Маршрутизувальні протоколи можуть бути реалізовані на основі різних алгоритмів, які відрізняються методами побудови таблиць маршрутизації, способами вибору найкращого маршруту та іншими особливостями.

Раніше вважалося, що в таблицях маршрутизації для кожної адреси призначення вказано тільки наступний (найближчий) маршрутизатор, а не весь їх ланцюжок від початкового до кінцевого вузла. Відповідно до цього підходу маршрутизація виконується за розподіленою схемою: кожен маршрутизатор відповідає за вибір тільки одного етапу шляху, а остаточний маршрут складається в результаті роботи всіх маршрутизаторів, через які проходить цей пакет. Такі алгоритми маршрутизації називають **однокроковими**.

Існує також протилежний, багатокроковий підхід – **маршрутизація від джерела** (Source Routing). Відповідно до нього вузол-джерело вказує в пакеті, який відправляють у мережу, повний маршрут його проходження через усі проміжні маршрутизатори. Такий спосіб не вимагає побудови та аналізу таблиць маршрутизації. Це прискорює проходження пакета мережею та розвантажує маршрутизатори, але при цьому велике навантаження лягає на кінцеві вузли. Дану схему застосовують набагато рідше, ніж схему розподіленої однокрокової маршрутизації.

Алгоритми та протоколи маршрутизації

Залежно від способу формування таблиць маршрутизації однокроковий алгоритми поділяють на три класи:

- алгоритми фіксованої (або статичної) маршрутизації;
- алгоритми простої маршрутизації;
- алгоритми адаптивної (або динамічної) маршрутизації.

Якщо **маршрутизація є фіксованою,** то всі записи в таблиці маршрутизації – статичні. Адміністратор мережі сам вирішує, на які маршрутизатори треба передавати пакети з тими чи іншими адресами, і заносить відповідні записи до таблиці маршрутизації вручну (наприклад, за допомогою утиліти route ОС UNIX або Windows NT).

Таблицю, як правило, створюють у процесі завантаження та редагують, якщо це необхідно. Такі виправлення можуть знадобитися, зокрема, якщо в мережі відмовляє якийсь маршрутизатор, а його функції передають іншому.

Таблиці розрізняють одномаршрутні, в яких для кожного адресата задано один шлях, та багато маршрутні, коли пропонується декілька альтернативних шляхів. У разі багатомаршрутних таблиць повинно бути задано правило вибору одного з маршрутів. Найчастіше один шлях є основним, а інші – резервними.

Очевидно, що алгоритм фіксованої маршрутизації з його способом формування таблиць маршрутизації вручну є доцільним тільки в невеликих мережах з простою топологією. Однак йог можна ефективно застосовувати також на магістралях великих мереж із простою структурою та очевидними найкращими шляхами проходження пакетів у мережі.

У алгоритмах **простої маршрутизації** таблиця маршрутизації або зовсім не використовується, або будується без протоколів маршрутизації. Виокремлюють три типи простої маршрутизації:

- випадкова маршрутизація, коли прибувший пакет посилається в будь-якому свободному напрямку, крім вихідного;
- лавинна маршрутизація, коли пакет широкомовно надсилається у всіх можливих напрямках, крім вихідного (аналогічно тому, як мости обробляють кадри з невідомою адресою);
- маршрутизація з урахуванням накопиченого досвіду, коли вибір маршруту здійснюється за таблицею, але таблиця будується так само, як і у випадку мосту, шляхом аналізу адресних полів пакетів, які надходять.

Найбільшого поширення набули алгоритми **адаптивної,** або **динамічної** маршрутизації. Вони забезпечують автоматичне оновлення таблиць маршрутизації після зміни конфігурації мережі. Використовуючи протоколи адаптивних алгоритмів, маршрутизатори можуть збирати інформацію про топологію зв'язків у мережі та оперативно реагувати на всі зміни конфігурації зв'язків. У таблиці маршрутизації, звичайно, заносять інформацію про інтервал часу, протягом якого даний маршрут залишатиметься чинним. Цей час називають *часом життя маршруту* (Time To Live, **TTL**).

Адаптивні алгоритми мають розподілений характер, тобто в мережі немає спеціально виділених маршрутизаторів для збору та узагальнення топологічної інформації: цю роботу розподілено між усіма маршрутизаторами.

У глобальних мережах використовують так звані **сервери маршрутів**. Вони збирають маршрутну інформацію, а

потім за запитами роздають її маршрутизаторів. У цьому випадку останні або звільняються від функції створення таблиці маршрутизації, або створюють тільки частину таблиці. Взаємодія маршрутизаторів із серверами маршрутів здійснюється за спеціальними протоколами, наприклад, **протоколом вибору наступного кроку** (Next Hop Resolution Protocol, **NHRP**).

Адаптивні алгоритми маршрутизації повинні відповідати декільком важливим вимогам. Перш за все, вони зобов'язані забезпечувати вибір якщо не оптимального, то хоча б раціонального маршруту. Друга умова – це їх неодмінна простота, щоб відповідні реалізації не споживали значні мережеві ресурси, зокрема вони не повинні породжувати занадто великий обсяг обчислень або інтенсивний службовий трафік. І, нарешті, алгоритми маршрутизації повинні мати властивість збіжності, тобто завжди приводить до однозначного результату за прийнятний час.

Сучасні адаптивні протоколи обміну інформацією про маршрути, у свою чергу, поділяють на дві групи, кожна з яких пов'язана з одним із наступних типів алгоритмів:

- дистанційно-векторні алгоритми (Distance Vector Algorithm, **DVA**);
- алгоритми стану зв'язків (Link State Algorithm, **LSA**).

В алгоритмах **дистанційно-векторного** типу кожен маршрутизатор періодично й широкомовно розсилає мережею вектор, компонентами якого є відстані від даного маршрутизатора до всіх відомих йому мереж. Під відстанню,

зазвичай, розуміють кількість транзитних вузлів. Метрика може бути й такою, яка враховує не тільки кількість проміжних маршрутизаторів, але й час проходження пакетів між сусідніми маршрутизаторами або надійність шляхів.

Одержавши вектор від сусіда, маршрутизатор збільшує відстань до зазначених у ньому мереж на довжину шляху до даного сусіда та додає до нього інформацію про відомі йому інші мережі, про які він дізнався безпосередньо (якщо вони під'єднані до його портів) або з аналогічних оголошень інших маршрутизаторів, а потім розсилає нове значення вектора мережею. Врешті-решт, кожен маршрутизатор дізнається інформацію про всі наявні в об'єднаній мережі мережах і про відстань до них через сусідні маршрутизатори.

Дистанційно-векторні алгоритми добре працюють тільки в невеликих мережах. У великих же мережах вони завантажують лінії зв'язку інтенсивним широкомовним трафіком. Зміни конфігурації відпрацьовуються з цього алгоритму не завжди коректно, так як маршрутизатори не мають точного уявлення про топологію зв'язків у мережі, а лише узагальнену інформацію (вектор відстаней), до того ж отриманої через посередників. Робота маршрутизатора відповідно до дистанційно-векторних протоколом нагадує роботу моста, так як точної топологічної картини мережі такий маршрутизатор не має.

Найбільш поширеним протоколом на базі дистанційно-векторного алгоритму є **протокол маршрутної інформації** (Routing Information Protocol, **RIP**).

Протокол RIP для IP-мереж подано двома версіями. Перша версія RIPv.1 не підтримує маски, тобто передбачає

поширення між маршрутизаторами тільки інформації про номери мереж і відстані до них, а інформація про маски цих мереж не розсилається. Вважається, що всі адреси належать до стандартних класів A, B або C. Протокол другої версії RIP v.2 передає інформацію про міські мережі, тому він більшою мірою відповідає вимогам сьогодення.

Алгоритми стану зв'язків дають змогу кожному маршрутизатору отримати достатню інформацію для побудови точного графа зв'язків мережі. Всі маршрутизатори працюють на основі однакових графів, у результаті чого процес маршрутизації виявляється більш стійким до змін конфігурації. «Широкомовна» розсилка (передавання пакету всім найближчим сусідам маршрутизатора) проводиться тут тільки при змінах стану зв'язків, що в надійних мережах відбувається не так часто. Вершинами графа є як маршрутизатори, так об'єднані ними мережі. Поширювана умережі інформація складається з опису зв'язків різних типів: маршрутизатор-маршрутизатор, маршрутизатор-мережа.

Для того щоб зрозуміти, в якому стані знаходяться лінії зв'язку, під’єднані до портів маршрутизатора, він повинен періодично обмінюватися короткими пакетами HELLO зі своїми найближчими сусідами. Цей службовий трафік також засмічує мережу, але не так, як, наприклад, пакети RIP, оскільки як пакети HELLO мають набагато менший обсяг.

Прикладом протоколу на основі алгоритму стану зв'язків може бути протокол першочергового вибору найкоротшого шляху «**відкрити найкоротший шлях першим**» (Open Shortest Path First, **OSPF**) стеку TCP/IP.

Протокол OSPF розроблено з урахуанням застосування у великих гетерогенних мережах. Обчислювальна складність протоколу OSPF швидко зростає із збільшенням розмірності інтермережі, тобто збільшенням кількості вхідних у неї IP-мереж, маршрутизаторів і зв'язків між ними. Для вирішення цієї проблеми в протоколі OSPF уведено поняття «область» мережі (area) (не плутати з автономною системою Інтернет). Маршрутизатори, які належать певній області, будують граф зв'язків тільки для неї, що скорочує розмірність мережі. Між областями інформація про зв'язки не передається, а прикордонні маршрутизатори обмінюються певною інформацією про адреси IP-мереж, які знаходяться в кожній з областей, і про відстань від прикордонного маршрутизатора до кожної IP-мережі. Для передавання пакетів між областями вибирають один із прикордонних маршрутизаторів області, а саме той, у якого відстань до потрібної IP-мережі є найменшою. Передаючи адресу в іншу область, маршрутизатори OSPF агрегують декілька адрес загальним префіксом в один.

Маршрутизатори OSPF можуть брати адресну інформацію від інших протоколів маршрутизації, наприклад від протоколу RIP, що є корисним для роботи в гетерогенних мережах. Така адресна інформація обробляється так само, як і зовнішня інформація між різними областями.

Протоколи RIP і OSPF є *внутрішніми шлюзовими протоколами IGP*, вони працюють на маршрутизаторах корпоративних мереж.

Безкласова міждоменна маршрутизація CIDR

Оскільки біти ідентифікатора мережі починаються зі старших розрядів IP-адреси, маску підмережі можна подати коротше, просто *вказавши число бітів ідентифікатора мережі*. Наприклад, запис 192.168.0.1/**24** відповідає IP-адресі 192.168.0.1 з маскою підмережі 255.255.255.0. Такий вид *запису маски* називають **префіксом мережі.** У таблиці 11.3 наведено префікси мереж для класів A, B і C.

Таблиця 11.3

Клас мережі	Біти маски підмережі	Префікс мережі	Маска підмережі
A	11111111 00000000 00000000 00000000	/8	255.0.0.0
B	11111111 11111111 00000000 00000000	/16	255.255.0.0
C	11111111 11111111 11111111 00000000	/24	255.255.255.0

Подання маски підмережі у вигляді префікса мережі називається технологією **безкласової міждомеої маршрутизації** (Classless Interdomain Routing, **CIDR**). Розподілення IP-адреси на номер мережі і номер вузла в технологии CIDR виконується не на основе декількох старших бітів, які визначають клас мережі, а на основе маски перемінної довжини, що надається Інтернет-провайдером. Використання префікса мережі дозволяет центрам разповсюдження адрес запобігти видачи абонентам надлишкових адрес.

Технологія CIDR описана в RFC 1519 і полягає в об'єднанні адрес, які залишено, в блоки змінного розміру, незалежно від класу. Провайдер при цьому отримує

можливість «нарізати» блоки з виділеного йому адресного простору відповідно до реальних потреб кожного клієнта. Це дає змогу уникнути видачі клієнтам зайвих адрес. Наприклад, якщо декому потрібно 2000 адрес, йому виділяють блок з 2048 адрес на межі, кратній 2048 байтам. Усі адреси блоку мають однаковий *префікс*, тобто однакові цифри в декількох вищих розрядах.

Використання CIDR дає змогу значно скоротити обсяги маршрутної інформації, що передають між автономними системами. Маршрутизація на магістралях Інтернету здійснюється на основі префіксів, а не повних адрес мереж. Так, якщо всі мережі всередині деякої автономної системи починаються з загального префікса, скажімо 194.27.0.0/16, то зовнішній шлюз автономної системи повинен робити оголошення тільки про цю адресу, не повідомляючи окремо про існування всередині даної автономної системи, наприклад, мережі 194.27.32.0/19 або 194.27.40.0/21, так як ці адреси агрегуються в адресі блоку 194.27.0.0/16.

У CIDR застосовують розширення всіх записів таблиці маршрутизації за рахунок додавання 32-бітної маски. Таким чином, утворюється єдина таблиця для всіх мереж, що складаються з набору трійок: *IP-адреса*, *маска підмережі*, *вихідна лінія*. Після надходження пакету, у разі застосування CIDR, з нього витягується IP-адреса призначення. Потім він маскується, а після сканування таблиці маршрутизації порівнюється зі значеннями записів. Може виявитися, що за значенням підійде декілька записів з різними довжинами масок підмережі. У цьому випадку використовується найдовша маска.

Технологію CIDR успішно застосовують у четвертій версії IPv4, вона підтримується такими протоколами маршрутизації, як OSPF, RIP.

Протокол BGP

Прикордонний шлюзовий протокол (Border Gatewey Protocol, **BGP**) є сьогодні основним протоколом маршрутного обміну інформацією *між автономними системами*, створеним для застосування в *Інтернеті*. Зважаючи на цю його особливість, BGP ще називають протоколом зовнішніх маршрутизаторів або протоколом міждоменної маршрутизації. **Доменом маршрутизації** в термінології Інтернету прийнято називати автономну систему (AS).

Хоча BGP розроблено як протокол маршрутизації між AS, його можна використовувати для маршрутизації всередині AS. Два сусідніх BGP, які сполучаються з різних AS, повинні знаходитися в одній і тій же фізичній мережі. Маршрутизатори BGP, які знаходяться в межах однієї і тієї ж AS, повідомляють про себе, щоб забезпечити узгоджуване уявлення про дану AS та визначити, який з її маршрутизаторів BGP виконуватиме роль точки з'єднання під час передавання повідомлень у зовнішні AS та під час їх приймання.

Деякі AS є тільки транзитними для проходження через них трафіку, джерело якого не знаходиться в їх межах і який не призначено для них. BGP, таким чином, повинен взаємодіяти з будь-якими протоколами маршрутизації всередині кожної з транзитних AS.

Повідомлення про коригування BGP складаються з пар «*мережевий номер/тракт AS*». **Тракт AS** є *своєрідним маршрутом, який описано як послідовність AS*, через які можна досягти мережі з зазначеними у повідомленні номером. Повідомлення про коригування для забезпечення надійної доставки відправляють за допомогою механізму транспортування TCP, описаного в розділі 11.5.

Обмін інформацією між двома маршрутизаторами відбивається в змісті маршрутної таблиці BGP. На відміну від деяких інших протоколів маршрутизації BGP не вимагає періодичного оновлення всієї маршрутної таблиці. Замість цього маршрутизатори BGP зберігають останню версію маршрутної таблиці кожного рівноправного члена. Хоча BGP підтримує маршрутну таблицю всіх можливих трактів до будь-якої конкретної мережі, у своїх повідомленнях про коригування він оголошує тільки про основні (оптимальні) тракти. Позначка про ступінь оптимальності тракту називається «**показником**». Він являє собою довільну кількість одиниць, що характеризує ступінь переваги якого-небудь конкретного тракту. Показники звичайно встановлюють адміністратори мережі за допомогою конфігураційних файлів. Ступінь переваги може базуватися на будь-якій кількості критеріїв, зокрема кількості AS (тракти з меншим числом AS. як правило, кращі), типі каналу (стабільність, швидкодія та надійність каналу) та інших факторах.

Формат пакету BGP подано на рисунку 11.4.

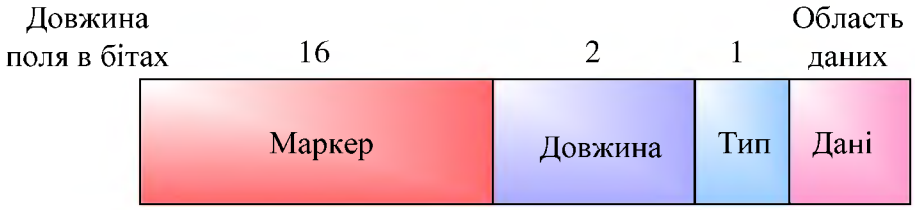

Рисунок 11.4. Формат пакету BGP

Пакети BGP мають загальний 19-байтовий заголовок, що складається з трьох полів:

- **Marker** – поле маркера має довжину 16 байтів, а його вміст може легко інтерпретувати одержувач. Маркер можна застосовувати для виявлення втрати синхронізації в роботі BGP-партнерів. Це поле використовують для встановлення автентичності;
- **Length** – поле довжини займає два байти і визначає повну довжину повідомлення в байтах разом із заголовком. Значення цього поля, зазвичай, лежить в межах 19–4096;
- **Type** є кодом різновиду повідомлення і може приймати такі значення: OPEN – відкрити; UPDATE – змінити; NOTIFICATION – увага; KEEPALIVE – ще живий.

Повідомлення протоколу BGP. У документі RFC 1163 визначено 4 типи повідомлень для BGP:

- відкривне повідомлення;
- повідомлення про коригування;

• сповіщення;
• повідомлення keepalive (продовжуй діяти).

Після того, як з'єднання протоколу транспортного рівня організовано, першим повідомленням, яке відправляє кожна сторона, є відкривне. Якщо відкривне повідомлення є прийнятним для одержувача, то відправнику надсилається повідомлення keepalive, яке підтверджує отримання відкривного повідомлення. Після успішного підтвердження прийняття початкового повідомлення можна здійснювати обмін коригуваннями, повідомленнями keepalive та сповіщеннями.

Відкривне повідомлення. Додатково до звичайного заголовка пакету BGP у відкривному повідомленні виділяють кілька полів. У полі версії (*version*) вказують номер версії BGP, що дає можливість одержувачеві перевіряти, чи збігається його версія з версією відправника. У полі автономної системи (*autonomous system*) поміщається номер AS відправника. Поле часу утримування (hold time) вказує максимальне число секунд, які можуть пройти без отримання будь-якого повідомлення від передавального пристрою, перш ніж вважати його таким, що відмовив. Поле коду підтвердження (*authentication code*) вказує на використовуваний код підтвердження (якщо він є). Поле даних підтвердження (autentication data) містить фактичні дані підтвердження (у разі їх наявності).

Повідомлення про коригування. Повідомлення про коригування BGP забезпечують коректування маршрутизації для інших систем BGP. Інформацію цих повідомлень

використовують для побудови графіка, який описує взаємодію між різними AS.

Окрім звичайного заголовка BGP повідомлення про коригування мають декілька додаткових полів. Ці поля забезпечують маршрутну інформацію шляхом перерахування атрибутів трактів, відповідних кожній мережі. На сьогодні час BGP визначає 5 атрибутів:

- *Origin* – джерело. Може мати одне з трьох значень: *IGP, EGP* і *incomplete* (незавершений). Атрибут IGP означає, що ця мережа є частиною даної AS. Атрибут EGP вказує на те, що початкові відомості про дану інформації отримано від протоколу EGP. Реалізації BGP надають перевагу маршрутами IGP над маршрутами EGP, тому що маршрут EGP відмовляє у разі наявності маршрутних петель. Атрибут incomplete використовують для з'ясування того, що про дану мережу відомо завдяки іншим засобам.
- *AS path* – шлях AS. Забезпечує фактичний перелік AS на шляху до вузла призначення;
- *Next hop* – наступне пересилання. Забезпечує адреса IP маршрутизатора, який необхідно використовувати як наступний транзитний вузол для пересилання до мереж, які перераховано в повідомленні про коригування.
- *Unreachable* – недосяжний. Вказує (у разі його наявності), що певний маршрут неможливо застосувати.

- *Inter-AS metric* – покажчик у повідомленні між AS. Забезпечує для маршрутизатора BGP можливість рекламувати свої витрати на маршрути до пунктів призначення, що знаходяться в межах його AS. Цю інформацію можуть використовувати маршрутизатори, які є зовнішніми відносно AS «рекламодавця», для вибору оптимального маршруту до конкретного вузла призначення, що знаходиться в межах даної AS.

Сповіщення відправляють у разі виявлення збійної ситуації, коли один маршрутизатор хоче повідомити іншому, чому він перериває з'єднання між ними. Крім звичайного заголовка BGP сповіщення містять поле коду помилки (*error code*), поле підкоду помилки (*error subcode*) та дані помилки (*error data*). Поле коду помилки вказує тип помилки, який може бути одним з перерахованих нижче:

- *Message header error* – помилка в заголовку повідомлення. Вказує на проблему в заголовку повідомлення, таку, як неприйнятна довжина повідомлення, неприйнятне значення поля маркера або неприйнятний тип повідомлення;
- *Open message error* – помилка в відкривному повідомленні. Вказує на наявність такої проблеми, як незабезпечуваний номер версії, неприйнятний номер AS або адресу IP та незабезпечуваний код підтвердження;

- *Update message error* – помилка в повідомленні про коригування. Вказує на наявність проблеми в повідомленні про коригування. Прикладами таких проблем можуть бути неправильно сформований перелік атрибутів, помилка в переліку атрибутів і недійсний атрибут наступного пересилання;
- *Hold time expired* – "час утримування закінчився". Вказує на закінчення періоду часу утримування, після чого вузол BGP буде оголошено нечинним.

Повідомлення keepalive (*продовжуй діяти*). Повідомлення keepalive не містять додаткових полів, крім розміщених у заголовку BGP. Ці повідомлення відправляються досить часто для того, щоб перешкоджати закінченню часу утримування таймера.

BGP є протоколом, який орієнтується на вектор відстані. Вектор описується списком AS по 16 біт на AS. BGP відрізняється від RIP і OSPF тим, що використовує TCP як транспортний протокол. Дві системи, які застосовують BGP, зв'язуються один з одним і пересилають за допомогою TCP повні таблиці маршрутизації. Надалі обмін йде тільки в разі якихось змін.

Протокол BGP дає змогу реалізувати маршрутну політику, яку визначає адміністратором AS. Політика відображається у конфігураційних файлах BGP. Маршрутна політика – це не частина протоколу, вона визначає рішення. Наприклад, коли місця призначення можна досягти кількома шляхами, політика враховує безпеку, економічні інтереси та ін.

Протокол BGP версії 4 (BGPv4) є вдосконаленою версією (порівняно з попередніми). Ця версія дає змогу пересилати інформацію про маршрут в межах одного IP-пакету. BGPv4 успішно працює у будь-якій топології зв'язків між автономними системами, що відповідає сучасному стану Інтернету. Для налаштування зв'язку маршрутизаторів спочатку робиться спроба реалізувати вищий з протоколів (наприклад, BGPv4), якщо одна з них не підтримує цю версію, номер версії знижується.

11.5. Протоколи транспортного рівня в мережах TCP/IP

Протоколи транспортного рівня вирішують завдання *передавання даних між прикладними процесами*. У стеку TCP/IP це завдання вирішують *протокол призначених для користувача датаграм і протокол керування передаванням*. Ці протоколи, також як і протоколи прикладного рівня, налаштовують тільки на кінцевих вузлах; вони забезпечують інтерфейс із вищерозташованим прикладним рівнем, передаючи дані, які надходять на вхідний інтерфейс хосту, котрий відповідає застосованню.

Протокол UDP

Протокол датаграм кориристувачів (User Datagram Protocol, **UDP**) описано в документі RFC 768. Він є одним із двох протоколів, розташованих над протоколом IP, і надає прикладним програмам транспортні послуги. Кожне

повідомлення протоколу UDP називають *датаграмою користувача*.

Протокол UDP, також як і протокол IP, забезпечує негарантовану доставку датаграм одержувачеві та не підтримує налаштування з'єднань. Взаємодія між прикладними програмами та протоколом UDP здійснюється через протокольні порти.

Під терміном «**протокольний порт**» (на відміну від порту фізичного пристрою) розуміють *абстрактну точка входу* в конкретнеу прикладну програму (застосовання), що знаходиться на певному комп'ютері. Надалі (відповідно до контексту) ми будемо використовувати також термін «порт», маючи на увазі «протокольний порт».

У стеку протоколів TCP/IP протокольний порт – це механізм, який дає змогу одному хосту одночасно підтримувати декілька сеансів зв'язку з віддаленими хостами та програмами. Можна сказати, що протокольний порт визначає прикладний процес, який є одержувачем інформації. Коли робоча станція отримує з мережі пакет, у якому вказано її IP-адресу, вона направляє його до конкретної прикладної програми, використовуючи для цього визначений під час налаштування сеансу зв'язку унікальний номер порту для цієї програми.

Механізм, який використовує кожна прикладна програма для визначення портів, на яких вона працює, або портів, до яких необхідно здійснювати доступ, надає мережева операційна система. Більшість операційних систем забезпечує синхронний доступ до портів. Кожен порт ідентифікується цілим позитивним числом. Для зв'язку з портом одержувача

відправник повинен знати IP-адресу хосту та відповідний номер протокольного порту. Кожне повідомлення містить також номер порту відправника. Таким чином, кожна прикладна програма, яка отримує повідомлення, має можливість відповісти безпосередньо відправнику (програмі).

Датаграма містить дві частини: заголовок та область даних. Заголовок складається з чотирьох 16-бітових полів, які визначають порт відправника, порт одержувача, довжину повідомлення й контрольну суму. На рисунку 11.5 зображено формат полів у датаграмі протоколу UDP.

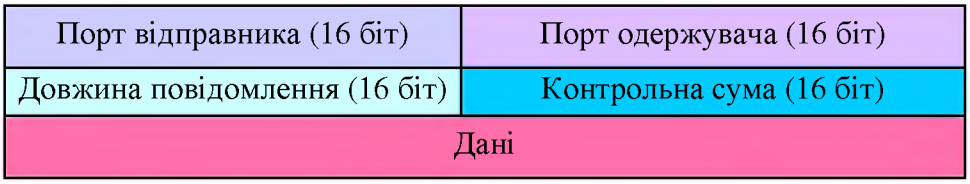

Рисунок 11.5. Формат полів у датаграми UDP

Поля «*Порт відправника*» і «*Порт одержувача*» містять 16-бітові номери портів, відповідно визначаючи прикладний процес на хості відправника та на хості одержувача. Поле «Порт відправника» може бути не використано, при цьому воно має містити нулі. Поле «*Довжина повідомлення*» містить інформацію про кількість байтів у датаграмі, при цьому враховується довжина заголовка UDP і даних.

Прикладні програми, що використовують UDP, повинні самі забезпечувати надійність передавання повідомлень. Обчислення контрольної суми датаграми UDP може проводитися, а може не проводиться. Значення «нуль» у полі «*Контрольна сума*» означає, що сума не обчислювалася.

Контрольна сума не обчислюється при роботі протоколу UDP у високонадійній локальній мережі. При роботі в ненадійній мережі тільки контрольна сума може вказати на достовірність і цілісність даних, які надійшли. Це пов'язано з тим, що протокол IP не обчислює контрольну суму поля даних у IP-датаграмах.

Для розрахунку контрольної суми UDP-датаграми необхідно мати додаткову інформацію. Для цієї мети до початку UDP-датаграми приписується псевдозаголовок, а до кінця датаграми додається байт з нулів для вирівнювання числа бітів повідомлення до кратного шістнадцяти. Після цього обчислюється контрольна сума отриманої датаграми. Кінцеві доповнення з нулів і псевдозаголовків не передаються разом з UDP-датаграмою.

Для обчислення контрольної суми отриманої UDP-датаграми спочатку зберігається нуль у полі «Контрольна сума», потім обчислюється 16-бітна сума, яка містить псевдозаголовок, заголовок самої датаграми й дані. Після отримання датаграми перевіряють контрольну суму, використовуючи IP-адресу призначення, отриману з заголовка IP-датаграми, яка містила UDP-датаграму. Якщо контрольні суми однакові, то датаграма дійсно надійшла до потрібного одержувача.

На рисунку 11.6 показаний формат псевдозаголовка. Псевдозаголовок має довжину 12 байтів. Поле «*Протокол*» містить код типу протоколу IP. Для перевірки контрольної суми одержувач повинен отримати ці поля з IP-заголовка, сформувати свій псевдозаголовок і обчислити контрольну суму.

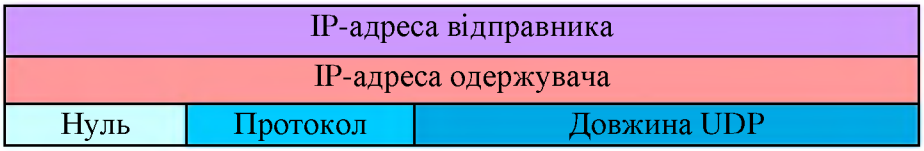

Рисунок 11.6 Формат псевдозаголовка

Дані протоколу UDP інкапсулюються в IP-датаграми (див.рис.11.7).

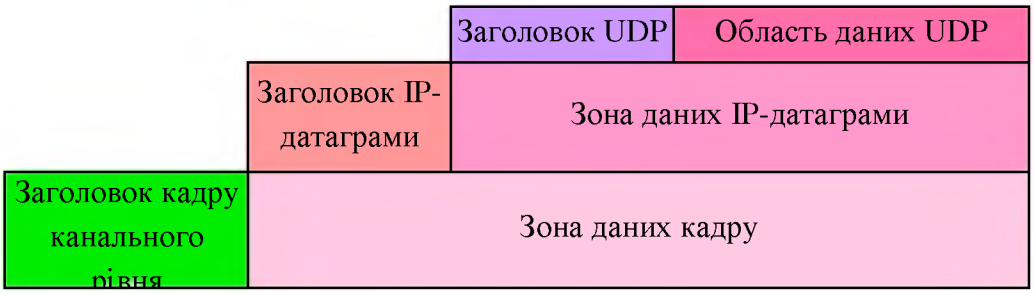

Рисунок 11.7. Два рівні інкапсуляції повідомлень UDP

Таким чином, IP-заголовок визначає хост, який є відправником та хост-одержувач, а UDP-заголовок – порти застосовань, які працюють на них.

Порт UDP надає протокольні порти, які використовують для розмежування декількох застосовань, що виконуються на одному комп'ютері. Крім того, протокол UDP може обслуговувати відразу декілька прикладних процесів, забезпечуючи приймання та передавання датаграм. Для цього в його програмному забезпеченні передбачено виконання процесів мультиплексування й демультиплексування. Ці процеси супроводжуються призначенням портів. Кожна прикладна програма повинна отримати від операційної

системи протокольний порт і пов'язаний з ним номер, який потім поміщається в поле «Порт відправника». Протокол UDP приймає UDP-датаграми, які приходять від протоколу IP, і демультиплексує їх по портах призначення. Рисунок 11.8 ілюструє приклад демультиплексування.

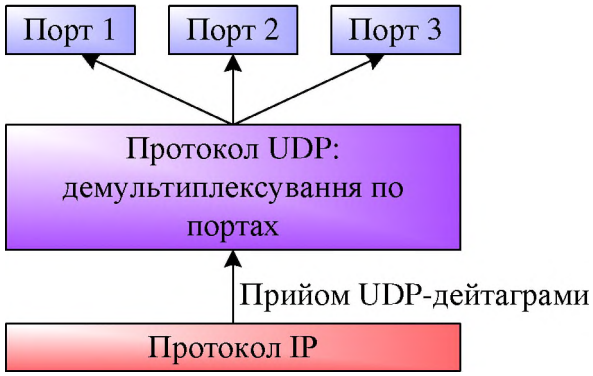

Рисунок 11.8. Приклад демультиплексування UDP-датаграми

Порт UDP найбільш наочно можна подати у вигляді черги. Операційна система створює внутрішню чергу повідомлень, які надходять. Якщо надійшло повідомлення з номером порту, якого немає серед використовуваних на прийомі портів, датаграма знищується та висилається **повідомлення протоколу керувальних повідомлень** у **Інтернеті** (Internet Control Message Protocol, **ICMP**) про помилку – «Порт є недосяжним».

Існує два підходи для призначення портів. Перший підхід використовує централізоване керування призначенням. Центральний орган IANA призначає номери портів, а потім публікує список призначень і контролює їх використання. У більшості випадків їх можуть використовувати тільки системні

процеси або застосування привілейованих користувачів. Діапазон номерів портів, призначених IANA, знаходиться в межах від 0 до 1023.

Другий підхід використовує динамічне призначення номерів портів, при якому мережеве забезпечення саме призначає порт, коли програма цього потребує. Ці номери портів не є загальновідомими, організація IANA їх не контролює, а використовують їх користувацькі процеси або застосування.

Числові значення цих портів знаходяться в межах від 1024 до 65535. Порти, числові значення яких знаходяться в межах від 1024 до 5000, називають тимчасовими. Хоча організація IANA не контролює використання цих портів, вона веде списки їх використання в інтересах спільнот Інтернету.

Для отримання інформації про поточне призначення портів необхідно надіслати відповідний запит. У стеку протоколів TCP/IP прийнято змішаний підхід, коли група портів призначається апріорі, але більшість інших можна використовувати вільно.

Протокол UDP використовують для передавання потокового відео, ігор реального часу, а також деяких інших типів даних, не так критичних до втрати пакетів, а критичних до затримок. Якщо застосування вимагає високої надійності, то використовують протокол TCP, який розглядатимемо далі.

Протокол TCP

Протокол керування передаванням (Transmission Control Protocol, **TCP**) описано в документі RFC 793. Його

використовують як *надійний транспортний засіб* для взаємодії розподілених прикладних процесів у TCP/IP мережах. Протокол TCP працює, як і протокол UDP, на транспортному рівні. Він забезпечує надійне транспортування даних між прикладними програмами шляхом налаштування логічного з'єднання між ними.

Для забезпечення надійного передавання даних налаштованими логічними з'єднаннями між парами прикладних програм протокол TCP повинен забезпечувати виконання таких функцій:

- передавати необхідні дані;
- підтримувати достовірність даних під час передавання;
- керувати потоком даних;
- розділяти канали зв'язку;
- обслуговувати налаштовані з'єднання;
- підтримувати встановлений пріоритет користувачів і відповідний рівень безпеки.

Сегмент є одиницею даних протоколу TCP. Вихідні від застосування дані буферизують засоби TCP. Для передавання на мережевий рівень з буферу «вирізають» певну безперервна частина даних, яку називають сегментом. Розмежування сегментів TCP здійснює протокол IP. Не всі сегменти, надіслані через з'єднання, будуть однакового розміру. Однак обидва учасники з'єднання повинні домовитися про максимальний розміру сегменту, який вони будуть використовувати. Цей розмір вибирають таким чином, щоб при пакуванні сегменту в IP-датаграму він містився туди цілком.

Адресатом інформації є модуль протоколу TCP на приймальному кінці. Цей модуль, у свою чергу, розміщує дані сегменту в буфер прикладної програми одержувача та сповіщає його про прибуття даних. З кожним модулем TCP пов'язано модуль протоколу IP, який забезпечує передавання локальною мережею. При цьому відбувається інкапсуляція сегменту TCP у датаграму протоколу IP. Ця датаграма, у свою чергу, поміщається в кадр канального рівня відповідного типу. Протокол IP здійснює фрагментацію та збирання сегментів TCP, необхідне для здійснення передавання та доставки їх через безліч мереж із різними технологіями канального рівня.

Передавання здійснюється надійно завдяки використанню підтверджень після отримання даних і механізму нумерації черг. Концептуально кожному байту даних присвоюють номер черги. Номер черги для першого байта даних у сегменті передають разом із цим сегментом і називають номером черги для сегмента. Оскільки кожен байт пронумеровано, то їх можна розпізнати. Механізм розпізнання байтів має накопичувальний характер, тобто розпізнання номера N означає, що всі байти з попередніми номерами (N - 1, N - 2, ...) вже отримано та розпізнано. Цей механізм дає змогу реєструвати появу дублів у разі повторного передавання. Нумерація байтів у межах сегменту здійснюється так, щоб перший байт даних відразу вслід за заголовком мав найменший номер, а наступні байти нумерувалися по зростаючій. Діапазон номерів лежить в межах від 0 до (232 - 1). Так як набір обмежено, то всі арифметичні операції з номерами черг повинні здійснюватися за модулем 232.

Номери черг не встигають пройти весь діапазон номерів у 232 значень, перш ніж пов'язані з ними дані з сегменту, який відправляють, отримають підтвердження від одержувача, а всі дублікати цього сегмента покинуть мережу. В іншому випадку двом сегментам можуть бути призначені однакові номери, що викличе проблему в одержувача. На швидкості 100 Мбіт/с один цикл використання всіх номерів складе 5,4 хв.

Сегменти також несуть номер підтвердження, який є номером наступного очікуваного байта даних, що передається у зворотному напрямку. Коли протокол TCP передає сегмент з даними, він поміщає його копію в чергу повторного передавання та запускає таймер. Після приходу підтвердження для цих даних відповідний сегмент видаляють з черги повторної передавання. Якщо підтвердження не приходить до закінчення часу, то сегмент посилають повторно. Пошкодження під час передавання фіксують за допомогою додавання до кожного сегменту, який передають, контрольної суми, перевірки її під час отримання та подальшої ліквідації дефектних сегментів.

Протокол TCP дає можливість одержувачеві керувати кількістю даних, які надсилає йому відправником. Це досягається посилкою разом з кожним підтвердженням так званого вікна. Вікно визначає кількість сегментів інформації, яку відправник може послати до отримання подальших вказівок. У вікні вказано діапазон номерів, наступних після номеру успішно прийнятого сегменту.

Правильність передавання кожного сегменту підтверджує квитанція одержувача. Для організації повторного передавання раніше перекручених даних відправник нумерує

сегменти для передавання. Для кожного сегменту відправник чекає від одержувача позитивну квитанцію, тобто службове повідомлення, яке повідомляє про те, що вихідний сегмент отримано, а дані в ньому є коректними. Час очікування квитанції є обмеженим. Відправляючи сегмент, відправник запускає таймер, і якщо після закінчення часу таймера квитанцію не отримано, то сегмент вважається загубленим.

Вибираючи величину часу очікування, так званого «тайм-ауту», чергової квитанції необхідно враховуватися швидкість і надійність фізичних каналів зв'язку, їх довжину і багато інших факторів. У протоколі TCP для кожного передавання фіксують час від моменту відправлення сегменту до приходу квитанції про його прийом (час обігу). Отримані значення часу обігу усереднюють з ваговими коефіцієнтами, які зростають від попереднього заміру до наступному. Як тайм-аут вибирають середній час обігу, помножений на деякий коефіцієнт. Варіюючи величину вікна, можна вплинути на завантаження мережі. Чим більшим є вікно, тим більшу порцію непідтверджених даних можна надіслати в мережу. Якщо прийомний буфер протоколу TCP переповнено, то він, відправляючи чергову квитанцію, вміщує в неї новий, зменшений розмір вікна. Якщо одержувач зовсім відмовляється від прийому, то в квитанції вказується вікно нульового розміру. Винятки є можливими в ситуації, коли надсилається повідомлення з позначкою «терміново». Порт зобов'язаний прийняти сегмент, навіть якщо для цього доведеться витіснити з буферу вже розміщені там дані.

У деяких реалізаціях протоколу TCP одержувач, якщо отримано спотворений сегмент, повинен відправити негативну

квитанцію. Існує два підходи до організації обміну позитивними й негативними квитанціями: з простоями та з організацією вікна.

Метод з простоями вимагає, щоб відправник очікував отримання квитанції від одержувача та тільки після цього посилав наступний сегмент. На рисунку 11.9 показаний приклад методу з простоями.

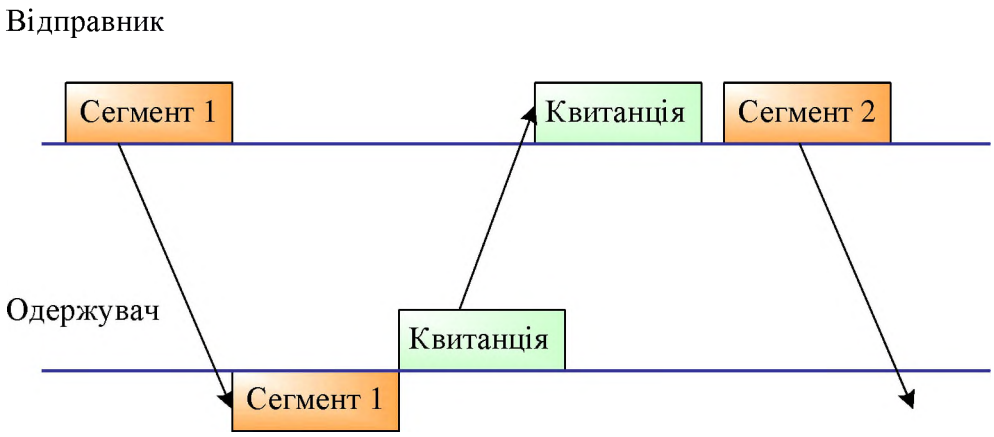

Рисунок 11.9. Приклад методу передавання сегментів з простоями

Даний метод має низьку швидкістю передавання даних. Особливо це стає помітним на низькошвидкісних каналах зв'язку.

Використовуючи **метод з організацією вікна**, названого ще **методом безперервного відправляння сегментів**, відправник може передати деяку кількість сегментів безперервно – без отримання на ці сегменти квитанції. Кількість сегментів, які дозволено передавати таким чином, вказується розміром вікна. На рисунку 11.10 наведено приклад реалізації цього методу для розміру вікна в N сегментів.

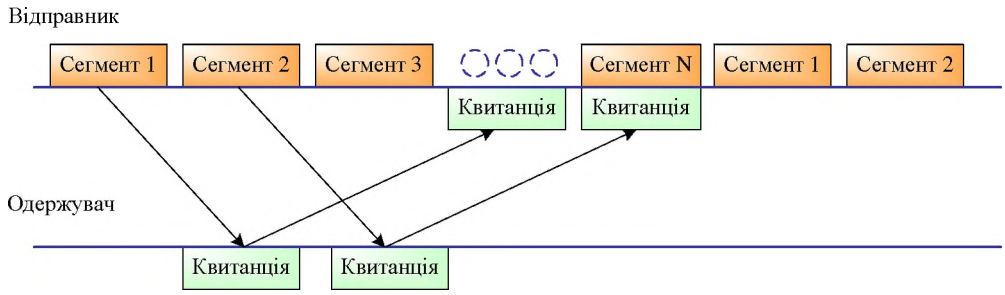

Рисунок 11.10. Метод безперервної відправки сегментів

Сегменти під час обмінювання нумерують циклічно – від 1 до N. Відправляючи сегмент 1, відправнику дозволено передати ще N - 1 сегментів до отримання квитанції на сегмент 1. Якщо ж за цей час квитанція на сегмент 1 так і не прийшла, то процес передавання припиняється, а після закінчення деякого часу сегмент 1 вважається загубленим і передається знову.

Якщо потік квитанцій надходить регулярно, в межах числа N сегментів, то швидкість обміну досягає максимально можливої величини для даного каналу зв'язку. Такий алгоритм називають **алгоритмом ковзного вікна**. Щоразу, отримуючи квитанції, вікно переміщається, захоплюючи нові сегменти, які дозволено передавати без підтвердження. Як квитанції одержувач сегмента відсилає відповідь, у якій міститься число, яке на одиницю перевищує максимальний номер байта в отриманому сегменті. Якщо розмір вікна дорівнює N, а остання квитанція містила значення K, то відправник може посилати нові сегменти доти, поки в черговий сегмент не потрапить байт з номером K + N. Цей сегмент виходить за межі вікна, що свідчить про необхідність призупинення передавання.

Для одночасного використання можливостей протоколу TCP кількома прикладними програмами на одному комп'ютері використовують набір адрес і протокольних портів. Зауважимо, що порти протоколу TCP відрізняються від портів протоколу UDP. Оскільки кожна програма протоколу TCP обирає ідентифікаторів портів незалежно, то вони не будуть унікальні. Унікальною буде сукупність ідентифікатора порту та його IP-адреси. Ця сукупність має назву «**сокет»**.

З'єднання між відправником і одержувачем визначаються двома сокетами на кінцях. Це з'єднання можна використовувати для передавання даних в обох напрямках, тобто воно є дуплексним.

Спрощено процес з'єднання можна подати такою послідовністю дій:

1) ініціатор з'єднання надсилає запит до протоколу TCP на відкриття порту для передавання;
2) після відкриття порту протокол TCP на стороні застосовання-ініціатора надсилає запит застосованню, з яким потрібно налаштувати з'єднання;
3) протокол TCP на приймальній стороні відкриває порт для прийому даних і відсилає квитанцію, яка підтверджує прийом запиту;
4) приймальня сторона відкриває порт для передавання і також передає запит до протилежної сторони;
5) застосовання-ініціатор відкриває порт для прийому та повертає квитанцію. З цього моменту з'єднання вважається налаштованим: починається обмін інформацією.

Існують кілька основних концепцій зв'язку портів із прикладними програмами, враховуючи будь-які реалізації протоколу TCP, та *загальновідомі сокети*. Для збереження всієї сукупності інформації щодо процесу створення сокетів є структура даних, яку називають **блоком керування передаванням** (Transmission Control Block, **TCB**).

Блок TCB формується для кожного з'єднання, є необхідним для підтримування з'єднань протоколу TCP та містить ряд змінних, якими є номери локального та віддаленого сокетів, прапори безпеки та пріоритети для даного з'єднання, покажчики буферів надсилання та отримання, покажчики поточного сегменту та черги повторного надсилання. Крім перерахованих вище змінних, блок має ряд змінних, які визначають черговість відправлення. Такими змінними є:

1. Відправлення:

- **SND.UNA** – надсилання не підтверджено;
- **SND.N**XT – надіслати наступний сегмент;
- **SND.WND** – відправити вікно;
- **SND.UP** – відправити строковий покажчик;
- **SND.WL1** – номер черги сегменту, використаний для оновлення останнього вікна;
- **SND.WL2** – номер підтвердження в сегменті, що використовується для оновлення останнього вікна;
- **ISS –** початковий номер черги відправлення.

2. Отримання:

- **RCV.NXT** – отримати наступний сегмент;
- **RCV.WND** – отримати вікно;
- **RCV.UP** – отримати строковий покажчик;
- **IRS** – початковий номер черги отримання.

Часто використовують змінні, які беруть своє значення з полів чергового сегменту. Такими є:

- **SEG.SEQ** – номер черги для сегменту;
- **SEG.ACK** – номер підтвердження для сегменту;
- **SEG.LEN** – довжина сегменту;
- **SEG.WND** – вікно для сегменту;
- **SEG.UP** – строковий покажчик для сегменту;
- **SEG.PRC** – пріоритет для сегменту.

Відправник даних за допомогою значення змінної SND.NXT відстежує наступний номер у черзі, що підлягає відправленню. Одержувач даних за допомогою змінної RCV.NXT відстежує наступний номер, прибуття якого він очікує. У змінну SND.UNA відправник даних поміщає значення найдавнішого номера, який було відправлено, але який ще не отримав підтвердження. Коли відправник створює та посилає якийсь сегмент, він збільшує значення змінної SND.NXT. Адресат, отримавши цей сегмент, збільшує значення змінної RCV.NXT і відправляє підтвердження. Після отримання підтвердження збільшується значення змінної SND.UNA. Різниця у значеннях цих змінних є *величиною*

затримування сегментів у мережі. Величину, на яку накладають зміну змінних, називають *довжиною поля даних* у сегменті.

З'єднання можна здійснювати активно, іноді відбувається пасивне очікування з'єднання ззовні. Програма, яка зробила запит на пасивне відкриття, може приймати запити на з'єднання від інших застосовань. Після приходу запиту на активне з'єднання протокол TCP інформує їх про налаштування з'єднання. Два застосовання, які зробили один одному одночасні запити на активне відкриття, отримають коректне з'єднання. Якщо на один і той самий локальний сокет зроблено кілька пасивних запитів на відкриття, які записуються в блоці TCB, і при цьому здійснюється активний запит на відкриття ззовні, то чужий активний сокет зв'язуватиметься з тим блоком TCB, у якому зазначено саме цей сокет.

Для кожного з'єднання існує номер у черзі відправлення та номер у черзі отримання. Початковий номер черги відправлення вибирає програма TCP, яка посилає дані в цій черзі, а початковий номер у черзі отримання з'ясовується під час налаштування з'єднання. У цей час обидва модулі протоколу TCP повинні синхронізувати один з одним первинні номери черг. Це проводиться шляхом обміну сегментами, які налаштовують з'єднання. Ці сегменти несуть прапор синхронізації SYN і вихідні номери для черг. Синхронізація вимагає, щоб кожна сторона посилала свій власний перший номер черги та отримувала підтвердження. Кожна сторона повинна отримати початковий номер черги напарника й надіслати підтвердження. Наприклад, для зв'язку сторони А зі стороною Б необхідно виконати такі дії:

1. Сторона А надсилає сегмент із SYN і своїм номером черги N стороні Б.
2. Сторона Б надсилає підтвердження – «ваш номер черги N» стороні А.
3. Сторона Б надсилає сегмент із SYN і своїм номером черги стороні А.
4. Сторона А відправляє підтвердження – «ваш номер черги К» стороні Б.

Кроки 2 і 3 можна об'єднати, тому такий обмін називають налаштуванням з'єднання з підтвердженням трьох шляхів.

Відміна з'єднання також передбачає обмінювання сегментами, які несуть керувальний прапор про скасування з'єднання.

У протоколі TCP існує так званий механізм «*проштовхування*», особливість якого в тому, що протокол TCP повинен передати все не відправлені раніше дані. Коли програма протоколу TCP на приймальній стороні виявляє ввімкнення механізму «проштовхування», вона не може отримувати нові дані доти, поки всі дані в її буфері не буде передано застосованню, яке їх очікує. Тобто вміст буферу одержувача передається користувачеві на оброблювання, навіть якщо він не був заповнений. Якщо надіслані дані заповнюють буфер користувача до того, як отримано команду до «проштовхування», користувачеві відправляють блок даних, відповідний розміру буферу.

Протокол TCP також має механізм для відправлення термінових даних. У цьому випадку протокол, не очікуючи

заповнення буфера до рівня розміру сегменту, негайно передає вказані дані в мережу. Про таких дані говорять, що їх передають *поза потоком*.

Протокол TCP використовує тип сервісу та опцію безпеки протоколу IP для забезпечення користувачам пріоритету та безпеки даних на кожному з'єднанні. Модулі протоколу TCP, які діють у багаторівневій системі безпеки, повинні адекватно оголошувати в сегментах, які надсилають, необхідну безпеку та пріоритет.

Заголовок TCP прямує за заголовком протоколу IP і доповнює його інформацією, специфічною для протоколу TCP. На рисунку 11.11 зображено формат заголовка протоколу TCP.

<table>
<tr><td colspan="3">Порт відправника (16 біт)</td><td>Порт одержувача (16 біт)</td></tr>
<tr><td colspan="4">Номер черги (32 біти)</td></tr>
<tr><td colspan="4">Номер підтвердження (32 біти)</td></tr>
<tr><td>Зсув даних (4 біти)</td><td>Резерв (6 біт)</td><td>Контрольні біти (6 біт)</td><td rowspan="2">Вікно (16 біт)</td></tr>
<tr><td colspan="3">Контрольна сума (16 біт)</td></tr>
<tr><td colspan="3">Опції (довжина змінна)</td><td>Вирівнювальне поле до 32 біт</td></tr>
</table>

Рисунок 11.11. Формат заголовка протоколу TCP

Поле «**Номер черги**» визначає номер черги для першого байта даних у цьому сегменті. Виняток становлять випадки, коли наявним є прапор синхронізації SYN. Поле «**Номер підтвердження**» містить наступний номер черги, який відправник даної датаграми бажає отримати в зворотному напрямку. Для цього необхідно встановити контрольний біт

підтвердження ACK. Номери підтвердження посилаються постійно, як тільки з'єднання вважається налаштованим.

Поле «**Зсув даних**» визначає кількість 32-бітних слів у TCP-заголовку та вказує на початок поля даних. Заголовок протоколу TCP завжди закінчується на 32-бітній межі слова, навіть якщо він містить опції. Поле «**Резерв**» має бути заповнено нулями. Поле «**Вікно**» містить оголошене значення розміру вікна в байтах. Поле «**Контрольна сума**» розраховується за сегментом, визначається 16-бітне доповнення суми всіх 16-бітових слів заголовка й даних. Якщо сегмент містить непарну кількість байтів, то його доповнюють нулями справа до утворення 16-бітного слова. При цьому вирівнювальний байт не передають разом з сегментом у мережі. Формат і значення поля «Контрольні біти» наведені у таблиці 11.4.

Таблиця 11.4.

Біти поля «Контрольні біти»					
1	**2**	**3**	**4**	**5**	**6**
URG	ACK	PSH	RST	SYN	FIN
Поле указателя срочности задействован	Поле «Номера подтверждения» задействовано	Включена функция проталки-вания	Перезагрузка данного соединения	Синхрониза-ция номеров очереди	Данных для передачи нет

Контрольна сума враховує також 96-бітний псевдозаголовок, який ставлять перед заголовком протоколу TCP. Псевдозаголовок містить адресу відправника, адреса одержувача, тип протоколу та довжину TCP-сегмента. Механізм псевдозаголовка забезпечує захист протоколу TCP від сегментів спотворення під час передавання.

Поле «**Покажчик терміновості**» визначає зміщення даного сегменту щодо номера черги. Цей покажчик повідомляє номер черги для байти, наступного після термінових даних. Поле використовують спільно з контрольним бітом URG.

Поле «**Опції**» має змінну довжину та може взагалі бути відсутнім. Воно розташовується в кінці заголовка протоколу TCP, його довжина кратна 8 бітам. Протокол TCP повинен бути готовий обробляти всі види опцій. Опції використовують для вирішення допоміжних завдань, наприклад для вибору максимального розміру сегменту. «**Компенсаційне поле**» може мати змінну довжину, бути фіктивним полем, яке використовують для доведення розміру заголовка до цілого числа 32-бітових слів.

З'єднання протоколу TCP переходять з одного стану в інший у відповідь на певні події: запити клієнта, прихід сегментів з прапорами SYN, ACK, RST, FIN і після закінчення заданого часу.

Проміжні стани з’єднань мають фіксовані значення та позначають:

- **LISTEN** – очікування запиту на з'єднання від чужих портів і програм TCP;
- **SYN-SENT** – очікування парного запиту на налаштування з'єднання (відправник запит вже зробив);
- **SYN-RECEIVED** – очікування підтвердження після того, як запит на з'єднання вже прийнято та відправлено;

- **ESTABLISHED** – стан відкритого з'єднання (прийняті дані можна надати користувачеві);
- **FIN-WAIT-1** – очікування запиту від чужої програми TCP або підтвердження раніше відправленого запиту на закриття з'єднання;
- **FIN-WAIT-2** – очікування запиту на закриття з'єднання від чужої програми TCP;
- **CLOSE-WAIT** – очікування запиту на закриття з'єднання від свого клієнта;
- **CLOSING** – очікування підтвердження запиту про закриття з'єднання від чужої програми TCP;
- **LAST-ACK** – очікування запиту на закриття з'єднання, раніше відправленого чужій програмі TCP;
- **TIME-WAIT** – часовий період, після якого можна бути впевненим, що чужа програма TCP отримала підтвердження свого запиту на закриття з'єднання;
- **CLOSED** – стан повного закриття з'єднання.

На рисунку 11.12 зображено діаграму зміни станів з'єднання.

Для розуміння механізму використання змінних під час налаштування з'єднань як приклад можна навести процедуру налаштування з'єднання з підтвердженням трьох шляхів (див. рис. 11.13). Кожен рядок рисунку пронумеровано. Стрілки → означають відправлення сегмента від модуля TCP сторони А в модуль TCP сторони Б.

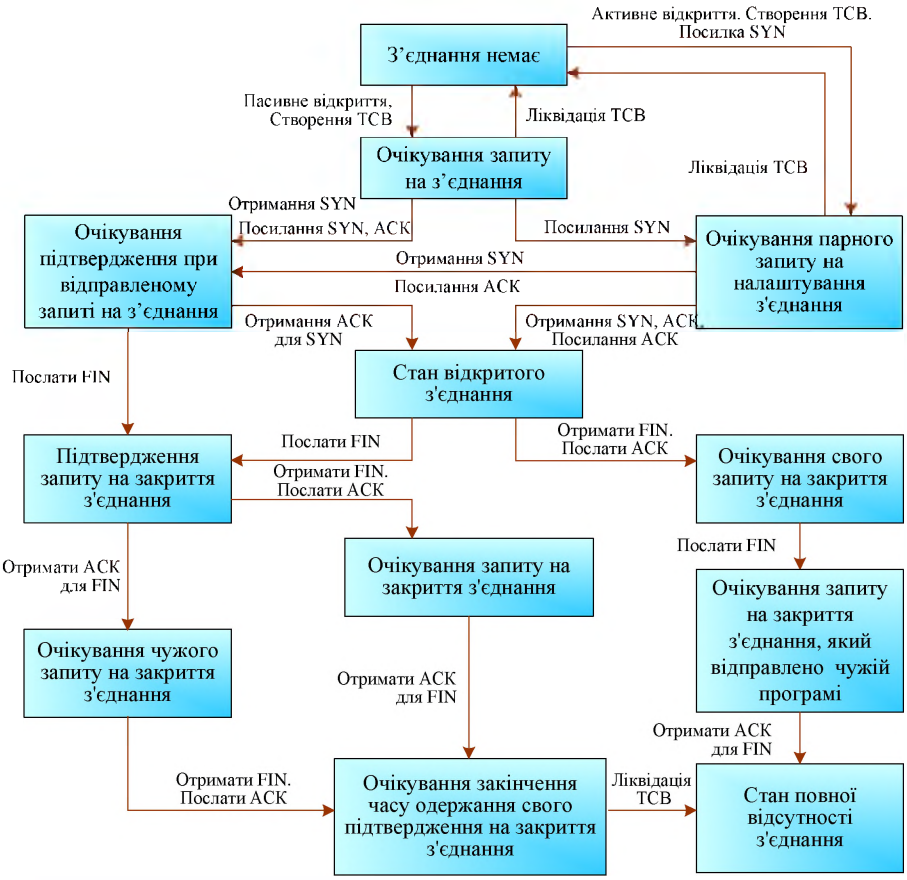

Рисунок 11.12. Діаграма зміни станів з'єднання

	Модуль TCP на стороні А	Модуль TCP на стороні Б
1.	CLOSED	LISTEN
2.	SYN-SENT ⟶ <SEQ=100><CTL=SYN>	⟶SYN-RECEIVED
3.	ESTABLISHED⟵ <SEQ=300><ACK=101><CTL=SYN, ACK>	⟶ SYN-RECEIVED
4.	ESTABLISHED ⟶ <SEQ=101><ACK=301><CTL=ACK>	⟶ ESTABLISHED
5.	ESTABLISHED ⟶ <SEQ=101><ACK=301><CTL=ACK><DATA>	⟶ ESTABLISHED

Рисунок 11.13. Процедура підтвердження трьох шляхів для синхронізації з'єднання

Стрілки ← показують зворотні процеси. Проміжний стан з'єднання відповідає моменту надсилання або отримання сегменту. Зміст сегменту наведено в скороченій формі, він є номером черги, прапором керування і полем АСК. Інші поля не показано.

Сторона А вказує, що вона буде використовувати номер черги 100. У відповідь сторона Б посилає свій номер черги 300 і каже, що чекає на отримання номера 101 (рядок 3). У рядку 5 модуль TCP сторони А передає деяку порцію даних.

На рисунку 11.14 відтворено нормальну процедуру закриття з'єднання.

	Модуль TCP на стороні А	**Модуль TCP на стороні Б**
1.	ESTABLISHED	ESTABLISHED
2.	FIN-WAIT-1→	→ CLOSE-WAIT
3.	FIN-WAIT-2← <SEQ=300><ACK=101><CTL=	← CLOSE-
4.	TIME-WAIT←	← LAST-ACK
5.	TIME-WAIT→ <SEQ=101><ACK=301><CTL=	→ CLOSED
6.	CLOSED	

Рисунок 11.14. Нормальна процедура закриття з'єднання

Протокол TCP розглядали, зважаючи на його використання як транспортного механізму обмінювання маршрутною інформацією в протоколі політики маршрутизації BGP. Протокол TCP знаходиться в постійному розвитку. Один із останніх кроків у розвитку протоколу регламентовано у документі RFC-1323. Ця модернізація протоколу адаптує його до дуже високих швидкостей передавання (до терабіт в секунду).

Контрольні питання

1. У чому полягає відмінність понять «IP-мережа» та «TCP/IP-мережа»?

2. Яке призначення протоколу IP і його місце в стеку TCP/IP?

3. Що таке IP-адреса, чим відрізняються IP-адреси для різних класів мереж?

4. Що таке маска підмережі? Якщо маска підмережі дорівнює 255.255.240.0, скільком дорівнює максимальна кількість хостів у логічній мережі?

5. Що розуміють під методом CIDR?

6. Якими є загальні, приватні, динамічні та статичні IP-адреси, чим вони відрізняються від доменних імен?

7. Охарактеризуйте відмінність форматів пакетів протоколу IP версії 4 і 6-ї версії.

8. Яке призначення протоколу дозволу адрес ARP?

9. Як влаштовано глобальний Інтернет?

10. Для чого призначено таблиці маршрутизації? Охарактеризуйте їх специфіку.

11. Охарактеризуйте однокрокові алгоритми різних класів?

12. Назвіть найбільш поширений протокол на базі дистанційно-векторного алгоритму, на базі алгоритму стану зв'язків?

13. Охарактеризуйте протокол BGP, формат його пакету.

14. Які повідомлення передбачено в документі RFC 1163 для протоколу BGP?

15. Яке призначення протоколів транспортного рівня в стеку TCP/IP?

16. Що називають «протокольним портом»?

17. Охарактеризуйте протокол UDP.

18.Що забезпечує функція мультиплексування (демультиплексування) в протоколі UDP?

19. Як призначають номери портів у протоколі UDP?

20. У чому полягає відмінність протоколу TCP від протоколу UDP?

21. Як формується одиниця даних протоколу TCP – сегмент?

22. Як забезпечується надійність передавання даних у протоколі TCP?

23. Як працює алгоритм «ковзного вікна»?

24. Що називають «сокетом»?

25. Що таке заголовок протоколу TCP? Охарактеризуйте призначення його полів.

26. Як відбувається налаштування логічного з'єднання?