

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Організація комп'ютерних мереж

Підручник

*Затверджено Вченою радою КПІ ім. Ігоря Сікорського як
підручник для студентів, які навчаються за
спеціальностями 121 «Інженерія програмного
забезпечення» та 122 «Комп'ютерні науки»*

Київ
КПІ ім. Ігоря Сікорського
2018

Організація комп'ютерних мереж [Електронний ресурс] : підручник: для студ. спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; Ю. А. Тарнавський, І. М. Кузьменко. – Електронні текстові дані (1 файл: 45,7 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с.

*Гриф надано Вченою радою КПІ ім. Ігоря Сікорського
(протокол № 10 від 12.11.2018 р.)*

Електронне мережеве навчальне видання

ОРГАНІЗАЦІЯ КОМП'ЮТЕРНИХ МЕРЕЖ

Підручник

Автори: *Тарнавський Юрій Адамович, канд. фіз.-мат. наук, доц.
Кузьменко Ігор Миколайович, канд. техн. наук, доц.*

Відповідальний
редактор *Гагарін О. О., канд. техн. наук, доц.*

Рецензенти: *Рач В. А., д.т.н., проф., директор
Навчально-наукового інституту
інформаційних та комунікаційних
технологій Університету економіки та права
«КРОК»
Бондарчук Ю. В., к.ф.-м.н., доцент
кафедри системного аналізу та теорії
прийняття рішень, Київського
національного університету ім. Тараса
Шевченка*

Викладено теоретичний матеріал, описано основні компоненти мережі та вимоги до мереж, наведено матеріал для самостійної роботи студентів та їх контролю.

Для студентів спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки», які вивчають дисципліни «Організація комп'ютерних мереж», «Розподілена обробка даних», «Архітектура комп'ютерів». Видання також може бути корисним для викладачів та спеціалістів, які працюють в даній області.

©КПІ ім. Ігоря Сікорського, 2018
©Ю. А. Тарнавський, І. М. Кузьменко

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП	8
Частина I. Теоретичні основи організації комп'ютерних мереж	
1. ОСНОВНІ ПОНЯТТЯ ТА ХАРАКТЕРИСТИКИ МЕРЕЖІ	10
1.1. Поняття комунікаційної та інформаційної мереж.....	10
1.2. Основні поняття мережевих технологій	12
1.3. Розвиток комп'ютерних мереж.....	14
1.4. Основні характеристики комп'ютерних мереж.....	17
1.5. Класифікація комп'ютерних мереж	18
Контрольні запитання до розділу	20
2. АРХІТЕКТУРА МЕРЕЖ	21
2.1. Поняття архітектури мережі і основні види архітектур	21
2.2. Архітектура «термінал-головний комп'ютер»	22
2.3. Архітектура «клієнт-сервер».....	25
2.4. Однорангова архітектура.....	29
2.5. Архітектура «комп'ютер-мережа».....	31
2.6. Архітектура інтелектуальної мережі.....	32
Контрольні запитання до розділу	34
3. ВЗАЄМОДІЯ РІВНІВ ЕТАЛОННОЇ МОДЕЛІ OSI.....	35
3.1. Поняття відкритої системи.....	35
3.2. Багаторівневий підхід до організації мережевої взаємодії.....	36
3.3. Модель ISO	40
3.4. Протокольна технологія	44
Контрольні запитання до розділу	45
4. ВЕРХНІ РІВНІ МОДЕЛІ OSI.....	46
4.1. Прикладний рівень.....	46
4.2. Рівень представлення даних.....	48
4.3. Сеансовий рівень.....	51
Контрольні запитання до розділу	55
5. НИЖНІ РІВНІ МОДЕЛІ OSI.....	56
5.1. Транспортний рівень.....	56
5.2. Мережевий Рівень	58
5.3. Канальний рівень	61
5.4. Фізичний рівень.....	63

5.5. Мережозалежні протоколи	64
Контрольні запитання до розділу	66
6. СІМЕЙСТВО СТАНДАРТІВ IEEE 802	67
6.1. Структура сімейства	67
6.1.1. Підрівень LLC.....	69
6.1.2. Підрівень MAC	71
6.2. Сімейство стандартів IEEE 802.....	72
Контрольні запитання до розділу	74
7. ПРОТОКОЛИ І СТЕКИ ПРОТОКОЛІВ	75
7.1. Стеки комунікаційних протоколів.....	75
7.2. Стек протоколів OSI	76
7.2.1. Електронна пошта X.400	77
7.2.2. Служба розподілених каталогів X.500	80
7.3. Стек протоколів TCP/IP	83
7.4. Стек протоколів IPX/SPX	85
Контрольні запитання до розділу	87
8. АРХІТЕКТУРА СТЕКА ПРОТОКОЛІВ MICROSOFT TCP/IP.....	88
8.1. Стандарти по TCP/IP.....	88
8.2. Архітектура Microsoft TCP/IP	89
8.3. Специфікація NDI	91
8.4. Мережеві протоколи TCP/IP	92
8.5. Діагностичні утиліти в Microsoft TCP/IP	93
Контрольні запитання до розділу	95
Частина II. Логічна організація комп'ютерних мереж	
9. АДРЕСАЦІЯ В IP-МЕРЕЖАХ	96
9.1. Адресний простір і види адрес.....	96
9.1. Локальні адреси.....	98
9.2. Мережеві адреси.....	99
9.2.1. Адресація на основі класів	100
9.2.2. Маски адрес	102
9.2.3. Загальні і приватні адреси	103
9.3. Символьні адреси	105
Контрольні запитання до розділу	108
10. ТОПОЛОГІЯ ЛОКАЛЬНОЇ МЕРЕЖІ.....	109
10.1. Поняття топології мережі	109

10.2. Топологія «шина».....	111
10.3. Топологія «зірка»	113
10.4. Топологія «кільце».....	114
10.5. Змішані топології	116
Контрольні запитання до розділу	117
11. МЕТОДИ ДОСТУПУ.....	118
11.1. Загальна характеристика методів доступу.....	118
11.2. Метод доступу CSMA/CD	120
11.2. Метод доступу CSMA/CA	125
11.3. Метод доступу TPMA.....	126
11.4. Метод доступу DPP.....	128
Тестові завдання.....	130
Частина III. Апаратне і програмне забезпечення комп'ютерних мереж	
12. ОСНОВНІ КОМПОНЕНТИ ЛОКАЛЬНОЇ МЕРЕЖІ.....	131
12.1. Мережеві адаптери.....	131
12.2. Повторювачі.....	133
12.3. Концентратори.....	134
12.4. Мости.....	138
12.5. Комутатори	140
12.6. Маршрутизатори	142
Контрольні запитання до розділу	143
13. МЕРЕЖЕВІ ОПЕРАЦІЙНІ СИСТЕМИ.....	144
13.1. Поняття мережевої ОС	144
13.2. Функціональні компоненти мережевої ОС.....	145
13.3. Мережеві служби і мережеві сервіси	147
13.4. Однорангові і серверні мережеві ОС.....	150
13.5. Огляд відомих мережевих ОС.....	153
13.5.1. ОС Unix	153
13.5.2. ОС NetWare.....	155
13.5.3. ОС Windows	156
Контрольні запитання до розділу	158
14. КАНАЛИ І ЛІНІЇ ЗВ'ЯЗКУ. КАБЕЛЬНІ СИСТЕМИ.....	159
14.1. Поняття каналу зв'язку	159
14.2. Види ліній зв'язку	160
14.3. Кабельні системи.....	162

14.3.1. Вита пара.....	162
14.3.2. Волоконно-оптичні кабелі.....	165
14.3.3. Коаксіальні кабелі	167
14.4. Структурована кабельна мережа	168
Контрольні запитання до розділу	171
15. ХАРАКТЕРИСТИКИ ЛІНІЙ ЗВ'ЯЗКУ	172
15.1. Типи характеристик ліній зв'язку.....	172
15.2. Амплітудно-частотна характеристика.....	172
15.3. Пропускна здатність	174
15.4. Перешкодостійкість	177
15.5. Достовірність передачі даних	178
15.6. Формула Шеннона	178
Контрольні запитання до розділу	180
16. КАБЕЛЬНІ СИСТЕМИ ETHERNET	181
16.1. Типи Ethernet	181
16.2. Ethernet типу 10BASE5	183
16.3. Ethernet типу 10BASE2	185
16.4. Ethernet типу 10BASE-T	186
16.5. Ethernet типу 10BASE-FL	187
16.6. Ethernet типу 100BASE-TX	188
16.7. Ethernet типу 100BASE-T4	189
16.8. Ethernet типу 100BASE-FX.....	190
Контрольні запитання до розділу	190
Частина IV. Практичні завдання з організації комп'ютерних мереж	
17. ВАРІАНТИ ПРАКТИЧНИХ ЗАВДАНЬ.....	191
17.1. Використання особливостей анімації при створенні проекту мережі	191
17.2. Розробка нового проекту в середовищі NetCracker	196
17.3. Розробка багаторівневого проекту мережі	205
17.4. Налаштування та використання бази пристроїв NetCracker	213
17.5. Моделювання мережі в Cisco Packet Tracer.....	222
17.6. Передача повідомлень між клієнтом та сервером на базі TCP-протоколу	232
17.7. Передача повідомлень між клієнтом та сервером на базі UDP-протоколу	247
Контрольні запитання до розділу	253
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	257

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЕОМ – електронна обчислювальна машина,
ІОМ – інформаційно-обчислювальна мережа,
АТС – автоматична телефонна станція
ПК – персональний комп'ютер
АС – абонентська система,
LAN – локальна комп'ютерна мережа,
WAN – глобальна мережа,
URL – уніфікований локатор ресурсів (адреса ресурсу),
HTTP – протокол передачі гіпертексту та інших типів даних,
HTML – мова розмітки гіпертексту,
W3C – Консорціум Всесвітньої мережі (павутини),
MAN – міська мережа (мережа мегаполісів),
SNA – системна мережева архітектура,
FER – комунікаційний процесор (контролер зв'язку),
CC – кластерний контролер,
DOS – дискова операційна система,
UNIX – сімейство операційних систем,
SSP – пункт комутації послуг,
SCP – пункт управління послугами,
RFC – документ, технічні специфікації для всесвітньої мережі,
OSI – базова еталонна модель взаємодії відкритих систем,
PDU – узагальнена назва фрагменту даних на різних рівнях OSI,
MAC – унікальний ідентифікатор обладнання для комп'ютерних мереж,
CRC – цикловий надлишковий код,
IEEE – міжнародна організація інженерів електротехніки, електроніки та радіоелектронної промисловості.

ВСТУП

Підручник ґрунтується на матеріалах лекцій дисципліни «Організація комп'ютерних мереж» спеціальності «Програмна інженерія», які викладалися у Національному технічному університеті «КПІ ім. Ігоря Сікорського». Це – одна з найважливіших частин підготовки сучасних фахівців у галузі інформаційних технологій, оскільки програми з функціонування мереж є невід'ємною складовою сучасних операційних систем. Необхідність розроблення таких програм пов'язана з упровадженням локальних та глобальних мереж у різних галузях господарювання, наприклад, для організації адміністративного керування або автоматизації бухгалтерських та банківських систем.

Матеріал підручника корелює з курсами «Основи програмування та алгоритмічні мови», «Системне програмування і операційні системи» й студентам бажано ознайомитися з ними.

У підручнику описано принцип дії апаратного забезпечення, що необхідно для організації функціонування локальних мереж (за інструкцією користувача вивчають пакет Net Cracker 4.1). На основі протоколів TCP/IP розглянуто засоби створення прикладного програмного забезпечення. Висвітлено питання з теорії передачі та теорії кодування інформації, основ побудови, програмування та використання локальних комп'ютерних мереж. Викладено теоретичні принципи, на основі яких побудовані комп'ютерні мережі й подано практичні рекомендації використання програмних пакетів, які забезпечують функціонування локальних обчислювальних мереж, описано методи проектування програмного забезпечення для обміну інформацією між комп'ютерами, які входять до складу локальної мережі.

Підручник складається з розділів:

- основи теорії передачі інформації;
- семирівнева еталонна модель OSI та приклади конкретних стеків

протоколів, які забезпечують взаємодію між програмами в мережі;

- склад та принципи побудови мережеских оболонок та мережеских операційних систем;

- засоби синхронізації взаємодіючих процесів;

- топології мереж та відповідні апаратні засоби для побудови мережі;

- методи побудови програмного забезпечення, яке функціонує в мережевому середовищі, за технологією «клієнт-сервер»;

- описано мережескі операційні системи та конфігурування технічних засобів;

- методи налагодження мережевого з'єднання та керування доступом до мережеских ресурсів.

ЧАСТИНА І. ТЕОРЕТИЧНІ ОСНОВИ ОРГАНІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ

1. ОСНОВНІ ПОНЯТТЯ ТА ХАРАКТЕРИСТИКИ МЕРЕЖІ

1.1. Поняття комунікаційної та інформаційної мереж

Комунікаційна мережа – система, що складається з вузлів (пунктів) і ліній передачі (зв'язків, з'єднань, комунікацій), в якій вузли відіграють функції генерації, перетворення, збереження і споживання продукту, а лінії передачі забезпечують передачу продукту між пунктами.

Як продукт можуть виступати інформація, енергія, речовина та інше. Відповідно розрізняють інформаційні, енергетичні, речовинні та інші мережі.

У складі комунікаційної мережі, що схематично зображена на рис. 1.1, розрізняють:

- а) кінцеві, або термінальні вузли, (телефони, ЕОМ, принтери, тощо);
- б) комунікаційні вузли (АТС, мультиплексори, демультиплексори, маршрутизатори та ін.).

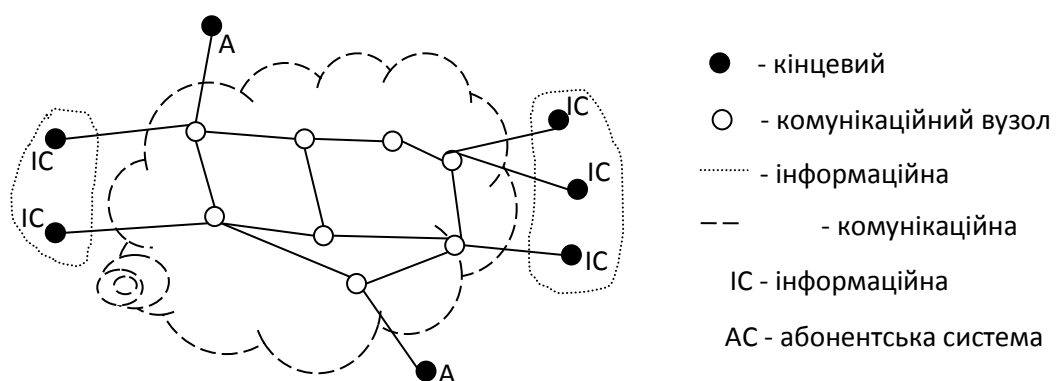


Рис. 1.1. Схематичне зображення комунікаційної мережі

Кінцеві вузли створюють і споживають продукт.

Комунікаційні вузли здійснюють:

- прийом, проміжне збереження і передачу;
- керують напрямком передачі здійснюючи маршрутизацію;
- контролюють перевантаженість вузлів і правильність передачі.

Інформаційно-обчислювальна мережа (ІОМ) – комунікаційна мережа, в якій продуктом генерування, переробки, збереження і споживання є інформація в електронному вигляді.

Як кінцеві вузли ІОМ можуть виступати комп'ютери та їх периферійне обладнання (принтери, плотери та ін.), обчислювальне, вимірювальне і виконуюче обладнання автоматичних і автоматизованих систем. Їх також називають абонентськими вузлами або абонентськими системами.

Як комутаційні вузли ІОМ можуть виступати маршрутизатори, комутатори, мости, повторювачі.

Як правило, в інформаційно-обчислювальних мережах можна виділити групи кінцевих вузлів, між якими здійснюється найбільш інтенсивний обмін інформацією (локалізація трафіку). Такі групи складають інформаційні підмережі, а їх вузли називають інформаційними системами. Решта вузлів, які не входять до інформаційних підмереж, утворюють комунікаційну підмережу.

Інформаційна підмережа виконує функцію збереження інформації і складається з інформаційних систем. Під інформаційною системою тут слід розуміти систему, що є джерелом або споживачем інформації.

Комунікаційна підмережа виконує функції передачі інформації, а також функції, пов'язані з перетворенням інформації. В загальному випадку в структурі комунікаційної підмережі присутні наступні компоненти, що показані на рис. 1.2:

- Мережа доступу (Access Network) – забезпечує концентрацію потоків від обладнання користувачів (телефон, ПК, телевізор).
- Магістральна мережа (Backbone або Core Network) – об'єднує мережі доступу, забезпечуючи транзит трафіку між ними по високошвидкісним

каналам зв'язку.

– Інформаційні ресурси, або центри управління сервісами (Data Centers або Service Control Point) – інформаційні ресурси, що використовують для обслуговування користувачів (абонентів), наприклад, в телефонній мережі – довідкові служби, служби екстреного виклику.

На базі однієї комунікаційної підмережі може бути побудована група інформаційних мереж.

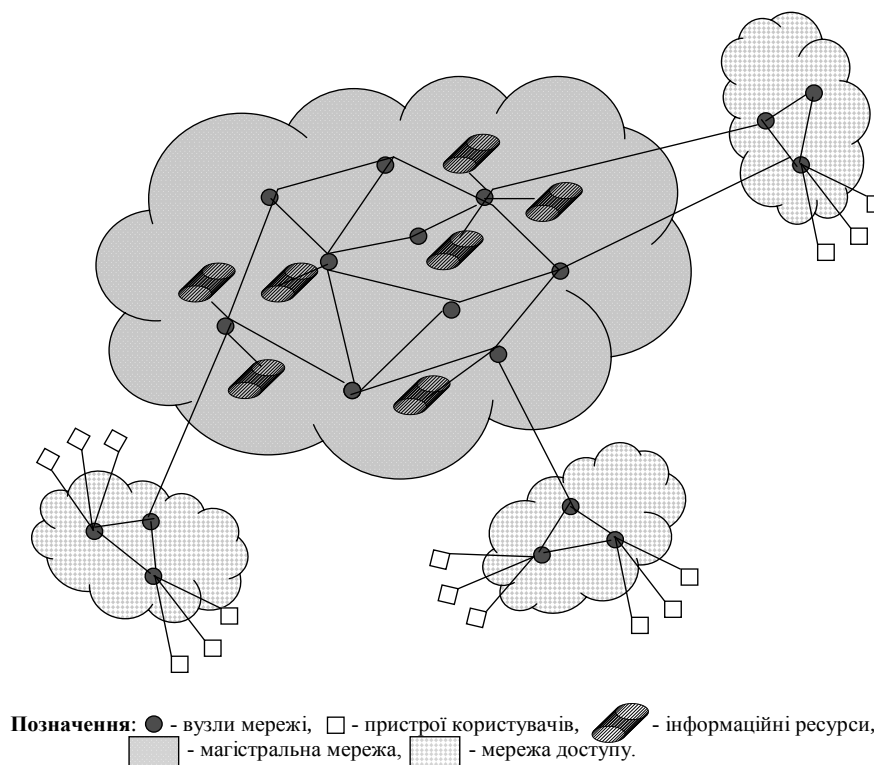


Рис. 1.2. Структура комунікаційної підмережі

1.2. Основні поняття мережевих технологій

Комп'ютерна мережа – це інформаційно-обчислювальна мережа, що призначена для обміну і розподіленої обробки інформації; вона складається з взаємодіючих абонентських систем (АС), об'єднаних за допомогою комунікаційної підмережі. Схематичне зображення комп'ютерної мережі показано на рис. 1.3.

Абонентська система – це сукупність ЕОМ, програмного забезпечення, периферійного обладнання та засобів зв'язку з комунікаційною підмережею,

якою забезпечується виконання прикладних процесів.

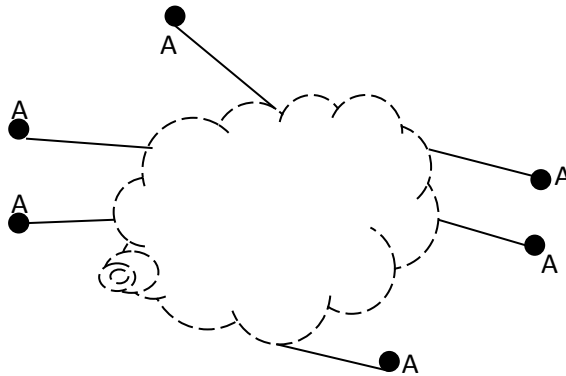


Рис. 1.3. Схематичне зображення комп'ютерної мережі

Комунікаційна підмережа або телекомунікаційна система – сукупність фізичного середовища передачі інформації, апаратних і програмних засобів, що забезпечують взаємодію абонентських систем.

Прикладний процес – процедури введення, обробки і видачі інформації, що виконуються в інтересах користувача, і описуються прикладними програмами.

Схему класифікації прикладних процесів в комп'ютерній мережі зображено на рис. 1.4.



Рис. 1.4. Класифікації прикладних процесів в комп'ютерній мережі

До спеціальних прикладних процесів належать:

- процеси керування роботою мережі,

- процеси діагностики роботи мережі,
- процеси забезпечення безпечної роботи в мережі та ін.

До програмних прикладних процесів відносяться такі, що керуються однією або групою пов'язаних програм.

Людино-машинні прикладні процеси реалізуються через взаємодію людини з терміналом.

1.3. Розвиток комп'ютерних мереж

У 60–90 рр. XX ст. розвиток комп'ютерів був основою зміни економіки, оскільки електронні обчислювальні машини типу «Днепр», «Мир» обробляли дані на перфокартах. Програма передавалась на обрахунок, як набір перфокарт, які приносили та вставлялися в порядку черги в ЕОМ, що працювала цілодобово. Аналогічно приходили і забирали результати, оскільки віддаленого спілкування з ЕОМ не було.

Це було незручно і тому з'явилися обчислювальні мережі з однією обчислювальною станцією – багатотермінальні системи (рис. 1.5). Однак, існуючі ЕОМ мали вільний машинний час і проблема полягала в передачі даних від сторонньої організації на обрахунок ЕОМ (mainframe).

У широкому розумінні, термін «мейнфрейм» вживається для позначення великої універсальної ЕОМ – високопродуктивного комп'ютера, призначеного для організації централізованих сховищ даних великої ємності і виконання інтенсивних обчислювальних робіт.

На початку 1990-х років на ринку мейнфреймів розпочалася криза, пік якої прийшовся на 1993 рік – за деякими прогнозами саме в цьому році передбачалось виключити останній інтерфейс. Проте згодом інтерес до мейнфреймів знову почав зростати. Практика показала, що централізована обробка даних на основі мейнфреймів здатна забезпечити вирішення більшості задач побудови інформаційних систем масштабу підприємства і при цьому є більш простою і дешевою, ніж розподілена обробка.

Нині ринок мейнфреймів стабілізувався, а обсяги продаж щорічно зростають.

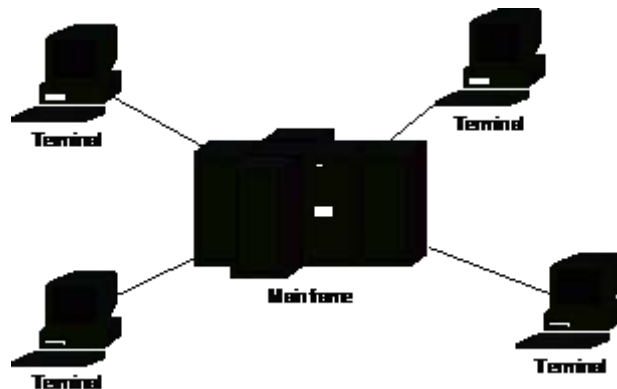


Рис. 1.5. Схематичне зображення багатотермінальної системи

У кінці 60-х – на початку 70-х рр. з'явилися перші глобальні мережі, оскільки існуючі ЕОМ мали вільний машинний час. Тому перші комп'ютерні мережі будувалися між ЕОМ на базі існуючих телефонних мереж для передачі даних між комп'ютерами різних організацій, використовуючи існуючі телекомунікаційні – телефонні мережі, для передачі інформації між ЕОМ, з маленькою швидкістю (рис. 1.6).

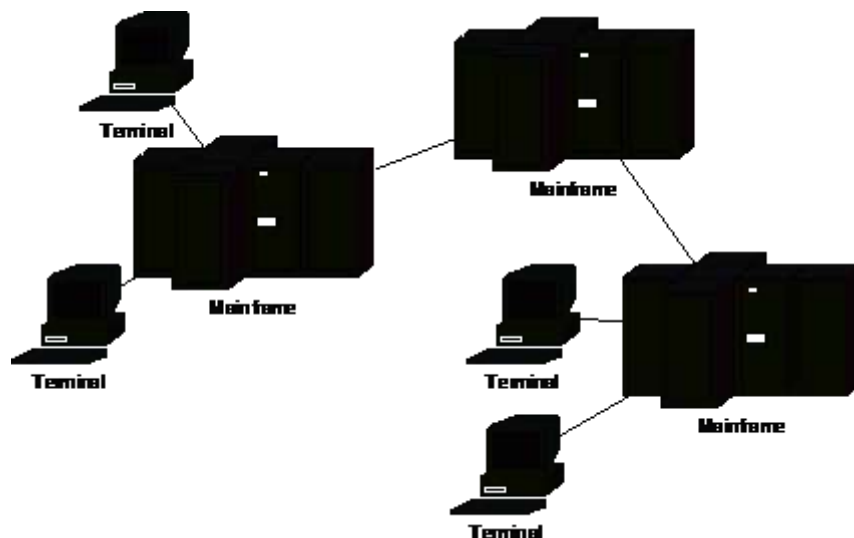


Рис. 1.6. Схематичне зображення глобальної мережі

Для віддаленого підключення ЕОМ в такій системі виникла потреба у використанні телефонних мереж, а перші протоколи передачі даних були на основі телефонних комунікацій. Як приклад, в 1969 р. Міністерство оборони США створює ARPANET (Advanced Research Project Agency Network) використовуючи цифрові лінії зв'язку компанії AT&T.

На початку 70-х років зі створенням перших мініЕОМ (на основі великих інтегральних схем – ВІС), виникла потреба в об'єднанні багатотермінальних систем у межах підприємства. З'явилися локальні обчислювальні мережі, що схематично зображені на рис. 1.7.

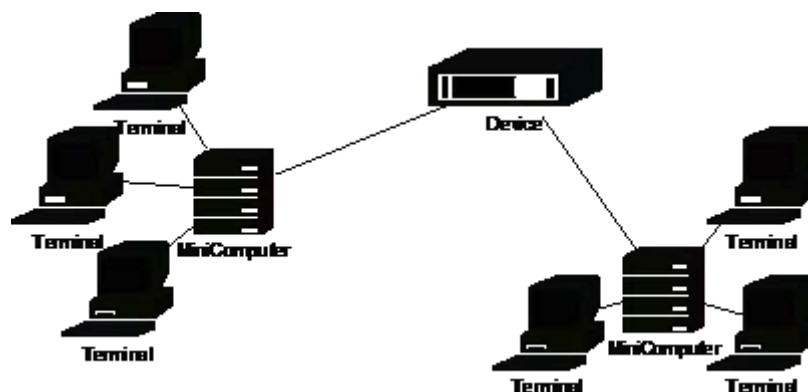


Рис. 1.7. Схематичне зображення локальної обчислювальної мережі

У середині 80-х років з'явилися ПК і стандартні технології побудови (Ethernet, TokenRing та інші) комп'ютерних мереж на їх основі з використанням стандартного комунікаційного обладнання, як показано на рис. 1.8.

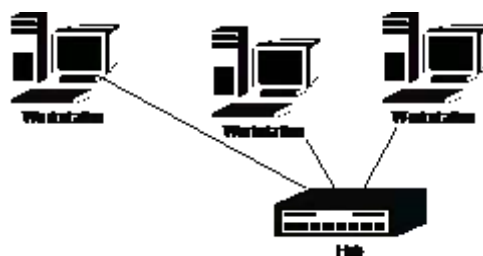


Рис. 1.8. Приклад мережі з використанням комунікаційного обладнання

В топології мережі стали використовувати віддалене з'єднання комп'ютерів і автоматичний режим обміну даними (файлами, електронною поштою).

Наприкінці 80-х, в 90-ті рр. розпочинається комерційне використання Internet, з'являється WWW. Серед LAN технологій з'явився лідер Ethernet, відбувається розвиток стандартів:

- оновлення мережі передачі даних,
- URL, HTTP, HTML для CERN (Тім Бернерс Лі) (1989),
- доопрацювання стандартів (1991-1995,)
- початок роботи W3C (1994).

За останнє десятиріччя відбувається зближення комп'ютерних мереж. Стираються відмінності LAN і WAN (Wide Area Network), Internet - технології переходять з WAN в LAN:

- Ethernet 10Gb замінює інші протоколи зв'язку,
- телекомунікаційні мережі з'єднуються з комп'ютерними мережами,
- оптоволоконний бум,
- швидкість передачі даних значно росте та перестає бути вузьким місцем.

Основні переваги використання комп'ютерних мереж такі: спільне використання інформаційних і обчислювальних ресурсів; удосконалення комунікацій; свобода у територіальному розташуванні; підвищення оперативності та якості рішень, що приймаються.

1.4. Основні характеристики комп'ютерних мереж

Основними характеристиками комп'ютерної мережі є наступні:

- мережева топологія – відображає просторове розташування мережевих вузлів та каналів зв'язку, яким визначається здатність мережевих

компонентів приймати і передавати дані.

- Мережеві протоколи – виражають формальний опис формату повідомлень і правил, за якими здійснюється обмін даними між вузлами мережі.
- Мережеві інтерфейси – апаратні технічні засоби з'єднання функціональних вузлів.
- Мережеві технічні засоби – пристрої, що забезпечують з'єднання абонентських систем в комп'ютерну мережу.
- Мережеве програмне забезпечення – програмне забезпечення, що призначене для управління роботою комп'ютерної мережі і забезпечення інтерфейсу користувача.

1.5. Класифікація комп'ютерних мереж

Класифікацію комп'ютерних мереж за низкою ознак зображено нижче, на рис. 1.9.



Рис. 1.9. Класифікація комп'ютерних мереж

Найпоширенішою є класифікація комп'ютерних мереж за

територіальною ознакою. Відповідно з нею розрізняють такі мережі.

- Локальні мережі (LAN) – мережі з просторовою протяжністю 1-2 км, в яких використовуються високоякісні лінії зв'язку зі швидкостями передачі даних ~100 Мбіт/с.

- Глобальні мережі (WAN) – територіально розподілені на великих площах мережі, що використовують вже існуючі лінії зв'язку з невисокою якістю передачі даних (телефонні і телеграфні зі швидкостями передачі даних порядку десятків Кбіт/с), або нові, спеціально прокладені лінії зв'язку та потребують складного комунікаційного обладнання для прийому-передачі даних.

- Міські мережі (MAN) – середні за протяжністю мережі, що призначені для обслуговування крупного міста (мегаполіса) і використовують цифрові магістральні лінії зв'язку (оптоволоконні) зі швидкостями передачі даних від 45 Мбіт/с.

Основні відмінності між LAN і WAN:

- протяжність.
- якість ліній зв'язку.
- швидкість передачі даних.
- складність обладнання та методів передачі і обробки даних.

Сучасною тенденцією є зближення LAN і WAN за рахунок:

- покращення якості передачі даних (за рахунок використання оптоволоконних ліній зв'язку),
- виникнення MAN,
- зростання захищеності інформації,
- виникнення intranet-технологій,
- можливість інтерактивної роботи навіть в WAN.

Контрольні запитання до розділу

1. В чому схожість та відмінність між комунікаційною та інформаційно-обчислювальною мережею?
2. Дайте визначення комп'ютерної мережі, комунікаційної підмережі та прикладних процесів.
3. В чому відмінність між Internet та Ethernet? Переваги комп'ютерних мереж.
4. Навести класифікацію та надати відмінності між локальними та глобальними мережами.
5. Навести основні характеристики комп'ютерних мереж.

2. АРХІТЕКТУРА МЕРЕЖ

2.1. Поняття архітектури мережі і основні види архітектур

Архітектура (від лат. architectura) – мистецтво проектування і будівництва.

Архітектура системи відбиває склад і взаємозв'язок компонентів системи, тобто визначає технологію її функціонування.

Архітектура комп'ютерної мережі – це концепція її побудови, яка визначає:

- основні елементи мережі;
- топологію мережі і функції кожного її елементу;
- фізичну і логічну організацію взаємодії елементів мережі.

За формою представлення комп'ютерних мереж розрізняють фізичну та логічну архітектуру.

Фізична архітектура – форма представлення комп'ютерної мережі у вигляді взаємодіючих апаратних засобів. Приклад фізичної архітектури комп'ютерної мережі зображено на рис. 2.1.

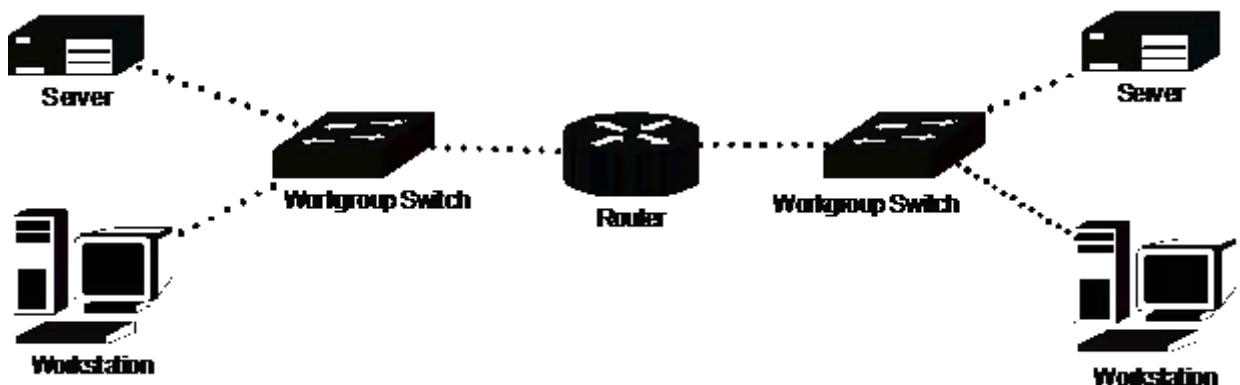


Рис. 2.1. Приклад фізичної архітектури комп'ютерної мережі

Логічна архітектура – форма представлення комп'ютерної мережі у вигляді взаємопов'язаних елементів (функцій). Приклад логічної архітектури комп'ютерної мережі зображено на рис. 2.2. Логічна архітектура відбиває

цілісну технологію комп'ютерної мережі і може бути деталізованою через рівні фізичної архітектури.

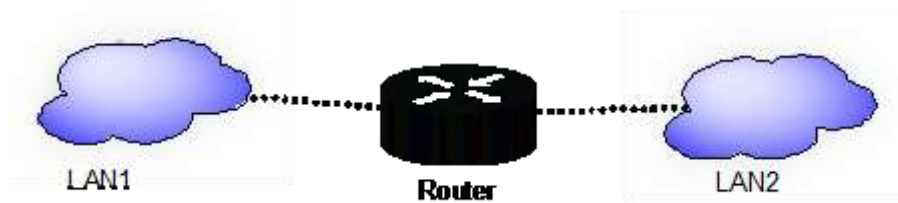


Рис. 2.2. Приклад логічної архітектури комп'ютерної мережі

У комп'ютерних мережах розрізняють п'ять архітектурних шаблонів:

1. Архітектура «термінал-головний комп'ютер».
2. Архітектура «клієнт-сервер».
3. Однорангова архітектура.
4. Архітектура «комп'ютер-мережа».
5. Архітектура інтелектуальної мережі.

2.2. Архітектура «термінал-головний комп'ютер»

Архітектура «термінал-головний комп'ютер» (terminal-host computer architecture) – це така концепція комп'ютерної мережі, коли вся обробка даних здійснюється одним, або групою головних комп'ютерів.

Така архітектура передбачає три основних типи обладнання, що показано на рис. 2.3:

- а) Головний комп'ютер (host computer) – здійснює управління мережею, збереження і обробку даних.
- б) Термінали (terminal) – забезпечують передачу головному комп'ютеру команд для організації сеансів роботи, введення даних і отримання результатів.
- в) Мультиплексори (multiplexor) – забезпечують «об'єднання» потоків даних від терміналів в спільний вихідний потік. Отже, мультиплексор –

це комбінаційний пристрій, який забезпечує передачу даних, що надходять з кількох входів на один вихід.

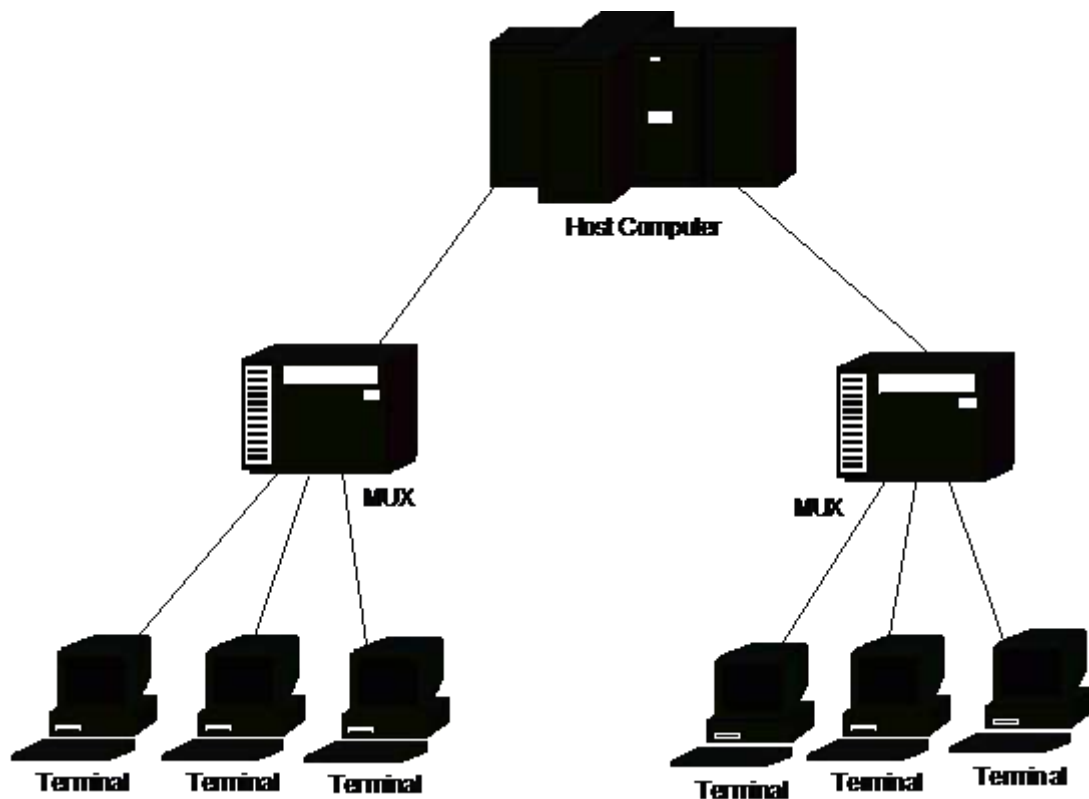


Рис. 2.3. Архітектура «термінал-головний комп'ютер»

Класичним прикладом архітектури «термінал-головний комп'ютер» є системна мережева архітектура – System Network Architecture (SNA) – запатентована компанією IBM мережева архітектура, що забезпечує підключення систем локальних мереж до мейнфреймів і мінікомп'ютерів IBM. Під мейнфреймом розуміється комп'ютер з архітектурою IBM/370, а під мінікомп'ютером – AS/400. В SNA використовують чотири основних типи обладнання:

- Мейнфрейми (Mainframe) – великі універсальні ЕОМ.
- Периферійне обладнання – термінали(Terminal) і принтери (Printer).
- Кластерні контролери (Cluster Controller, CC) – комунікаційні пристрої, що виконують мультиплексування.
- Комунікаційні процесори (Front End Processor, FEP) або контролери

зв'язку – комунікаційні пристрої, що розташовуються між лініями зв'язку і мейнфреймом і використовуються для його «розвантаження» – звільнення від виконання комунікаційних задач (контролю і усунення помилок передачі даних, кодування повідомлень, обслуговування ліній зв'язку та інше).

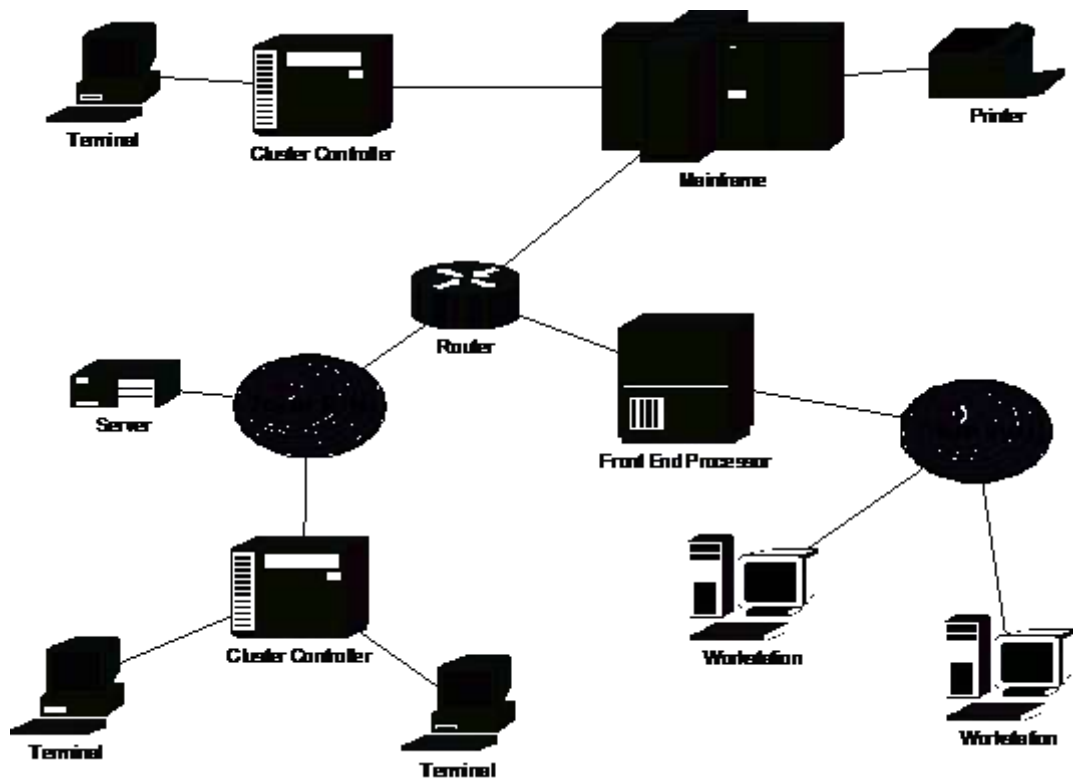


Рис. 2.4. Приклад реалізації SNA

SNA вирішила також проблему з підключенням до мейнфреймів і мінікомп'ютерів персональних комп'ютерів, що працювали під управлінням різних операційних систем (DOS, UNIX, Macintosh, Windows), як показано на рис. 2.5.

У 2007 р. році спеціалісти Computer World віднесли мережевий протокол SNA до розряду «мертвих». Однак, мережі на базі SNA ще можна зустріти в банках, страхових компаніях та інших фінансових установах.

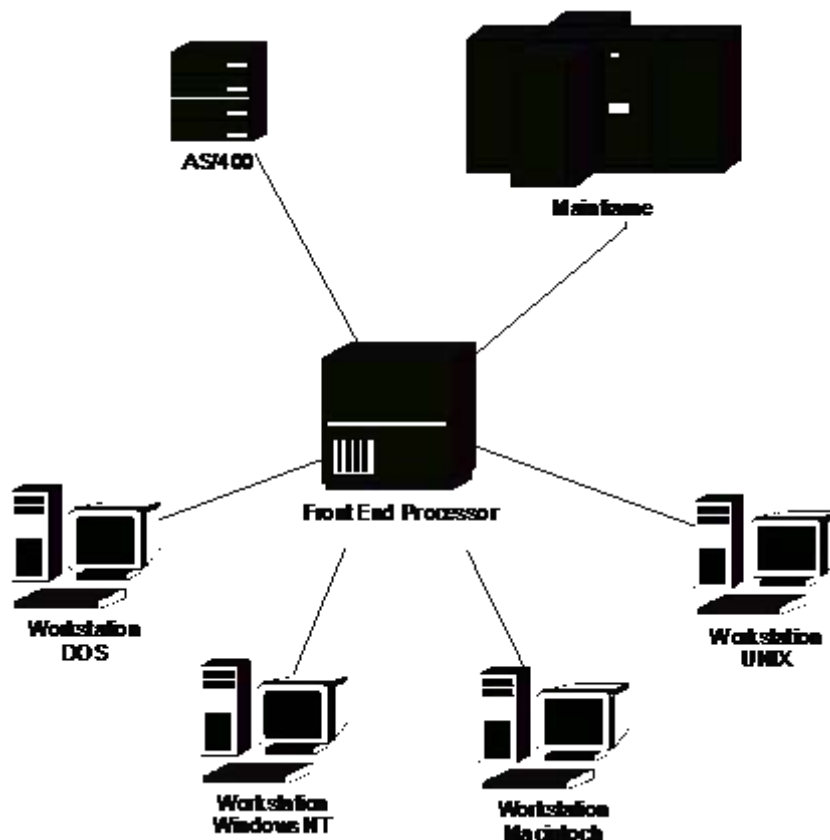


Рис. 2.5. Архітектура SNA з підключенням до мейнфрейму персональних комп'ютерів з різними операційними системами

2.3. Архітектура «клієнт-сервер»

Архітектура «клієнт-сервер» (client-server architecture) – це концепція комп'ютерної мережі, в якій основна частина ресурсів зосереджена на серверах, що обслуговують своїх клієнтів.

Зв'язок між комп'ютерами в мережі даної архітектури відбувається за рахунок відправки/прийому спеціальних повідомлень, що передаються через мережеві адаптери і лінії зв'язку. На рис. 2.6 показано, що за допомогою таких повідомлень один комп'ютер (PC-1) може надсилати іншому (PC-2) запити на доступ до його локальних ресурсів – даних на диску, периферійних пристроїв (принтерів, модемів і т.п.).

Для забезпечення можливості обміну повідомленнями операційні системи комп'ютерів (OS) доповнюються відповідними програмними модулями – клієнтами і серверами.

На тих комп'ютерах мережі, ресурси яких повинні бути доступні іншим користувачам, додаються модулі, які постійно знаходяться в режимі очікування запитів від користувачів; такі модулі отримали назву програмних серверів, оскільки їх основним завданням є обслуговування (serve) запитів користувачів.

На тих комп'ютерах мережі, які здійснюють доступ до ресурсів інших комп'ютерів, додаються програмні модулі, які можуть формувати запити і передавати їх по мережі, такі модулі отримали назву програмних клієнтів.

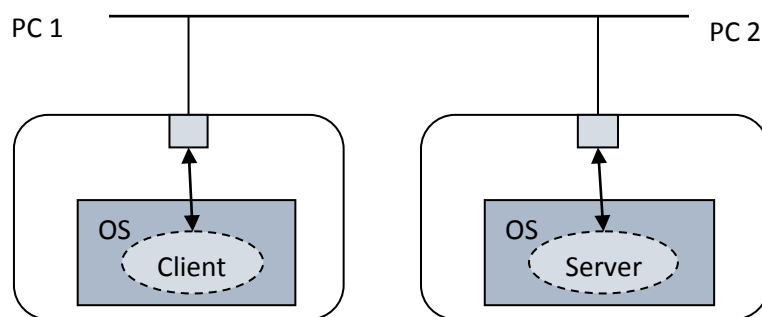


Рис. 2.6. Шаблон архітектури «клієнт-сервер»

Тобто, сервер (Server) – програмний прикладний процес, що забезпечує виконання сервісної функції.

Клієнт (Client) – програмний або людино-машинний прикладний процес, що викликає сервісну функцію.

Сервісна функція – комплекс прикладних програм, у відповідності з яким виконуються різноманітні прикладні процеси.

Пара модулів «клієнт»-«сервер», що забезпечує сумісний доступ користувачів до певного типу ресурсів, називають мережева службою.

Терміни «клієнт» і «сервер» використовуються не тільки для позначення програмних модулів, а й для позначення місця, що має комп'ютер в мережі. Якщо комп'ютер надає свої ресурси у спільне використання, він називається сервером, а якщо споживає ресурси інших –

клієнтом.

У цьому разі:

Сервер (Server) – спеціалізований комп'ютер, що надає сервіс іншим комп'ютерам мережі за їх запитами.

Клієнт (Client) – комп'ютер, який використовує ресурси сервера і забезпечує користувача інтерфейсом для роботи.

Сервіс – процес обслуговування клієнта.

Інтерфейс користувача – процедури взаємодії користувача з мережею.

У шаблоні архітектури «клієнт-сервер» комп'ютери клієнтів називають робочими станціями. Сервери спеціально оптимізуються для швидкої обробки запитів клієнтів до розподілених ресурсів і для управління захистом файлів.

За функціональним призначенням розрізняють (рис. 2.7) кілька типів серверів: файлові, друкування, додатків, поштовий, комунікаційний, баз даних. Розглянемо їх детальніше.

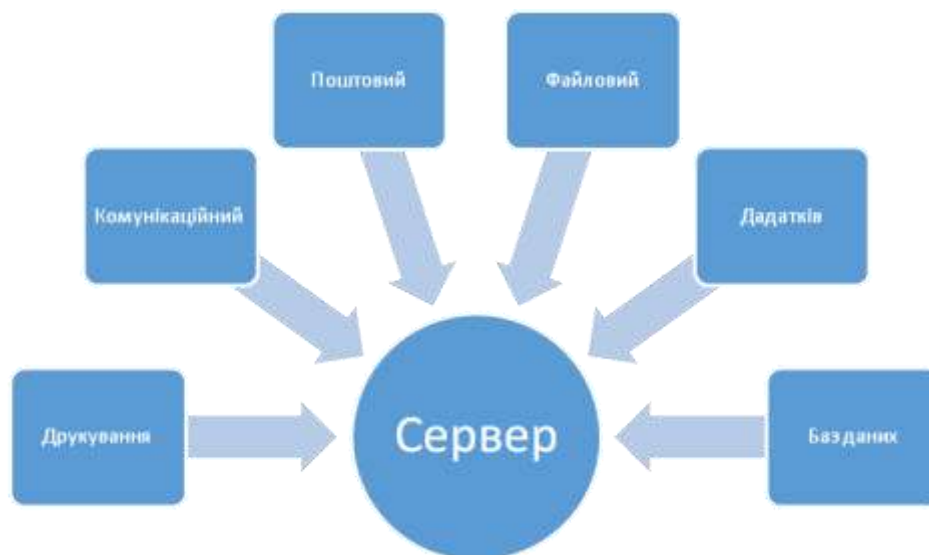


Рис. 2.7. Функціональне призначення сервера

Сервер друкування (принт-сервер) – комп'ютер, що забезпечує доступ до централізовано поділюваного принтера, створює чергу друку і виконує

управління принтером.

Комунікаційний, або сервер віддаленого доступу (Access Server) – надає можливість робочим станціям поділяти модем, або вузол зв'язку, з великою ЕОМ, що забезпечує доступ до мережі з віддаленого місця, обладнаного модемом. Часто комунікаційний сервер суміщає і функції сервера додатків.

Поштовий сервер – комп'ютер, що призначений для передачі електронних повідомлень між користувачами мережі.

Файловий сервер – комп'ютер, який виконує функції управління локальною мережею, доступом користувачів до файлів, що спільно використовуються.

Сервер додатків – комп'ютер, що виконує прикладну задачу, запуск якої здійснюється користувачем зі свого терміналу.

Сервер бази даних – комп'ютер, що забезпечує вибірку необхідних даних з бази даних і пересилку через мережу лише даних, що запитані клієнтом.

В архітектурі «клієнт-сервер» окремий (виділений) сервер забезпечує також централізований захист мережі завдяки перевірці облікових записів користувачів. Зокрема, Windows NT використовує систему доменних імен для управління користувачами, групами і машинами. Перед тим, як користувач отримає доступ до мережевих ресурсів, він має пройти процедуру авторизації, тобто – повідомити своє реєстраційне ім'я і пароль контролеру домену – серверу, який дозволяє доступ тільки у випадку допустимої комбінації імені і паролю.

Розглянемо переваги і недоліки архітектури «клієнт-сервер». Переваги:

- дозволяє організовувати мережі з великою кількістю робочих станцій.
- Спрощує мережеве адміністрування завдяки можливості централізованого управління обліковими записами.
- Забезпечує ефективний доступ до мережевих ресурсів (без

використання паролів доступу до ресурсів).

Недоліки архітектури «клієнт-сервер»:

- критична по відношенню до працездатності сервера,
- вимагає кваліфікованого персоналу для адміністрування мережі,
- підвищення вартості мережі через використання потужних серверів.

2.4. Однорангова архітектура

Однорангова архітектура (peer-to-peer architecture) – архітектурний шаблон комп'ютерної мережі, що базується на рівнозначності комп'ютерів в мережі. Тобто – кожний вузол (peer) виступає як в ролі клієнта, так і сервера. Відповідно – жоден комп'ютер не має ні вищого пріоритету на доступ, ні підвищеної відповідальності за надання ресурсів у спільне використання. Однорангові мережі називають також робочими групами.

Кожний користувач у такій мережі є одночасно і адміністратором мережі, оскільки, через використання паролів, він керує доступом до ресурсів свого комп'ютера.

Однією з областей застосування однорангової архітектури мереж є обмін файлами. Користувачі файлообмінних мереж розміщують свої файли в так називаних «розшарених» (англ. share - поділяти) каталогах, зміст яких доступний для завантаження іншим користувачам. В якості прикладу такої однорангової мережі можна навести файлообмінну мережу Gnutella2. Проте, в більшості архітектура файлообмінних мереж є не одноранговою, а гібридною, в якій сервери використовуються для координації роботи, пошуку, або надання інформації про комп'ютери і їх статуси. Прикладами таких гібридних файлообмінних мереж є EDonkey, BitTorrent.

Особливим видом однорангової архітектури мереж є бездротова ad hoc мережа, що показана на рис. 2.8. В такій мережі реалізується децентралізоване управління через використання Bluetooth, або Wi-Fi.

Розглянемо переваги та недоліки однорангової архітектури. Переваги

наступні:

- простота в установленні та налаштуванні мережі,
- невисока вартість і простота експлуатації мережі,
- незалежність комп'ютерів (від сервера),
- простота в управлінні ресурсами (кожний користувач управляє доступом до ресурсів власного комп'ютера),
- відсутність необхідності в персоналі для адміністрування мережі.

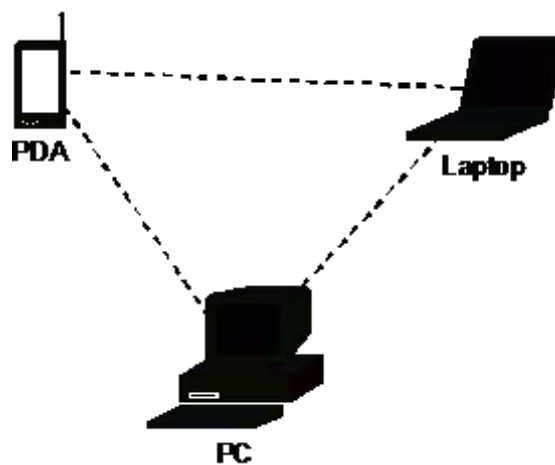


Рис. 2.8. Приклад шаблону однорангової архітектури мережі з використанням Wi-Fi

Недоліки однорангової архітектури:

- невелика кількість (близько 10) комп'ютерів в мережі,
- необхідність використання великої кількості паролів, якими забезпечується доступ до ресурсів мережі,
- зменшення продуктивності тих комп'ютерів, ресурси яких інтенсивно використовуються,
- відсутність централізованих можливостей для пошуку і управління даними.

З порівняння однорангової та архітектури «клієнт-сервер» для комп'ютерної мережі встановлено, що однорангову архітектуру слід застосовувати у випадку, якщо:

- кількість користувачів не перевищує 10;
- всі комп'ютери знаходяться недалеко один від одного;
- фінансові витрати на обслуговування мережі необхідно мінімізувати,
- відсутня необхідність використання спеціалізованих серверів (баз даних, додатків та ін.),
- немає необхідності в централізованому адмініструванні.

Архітектуру «клієнт-сервер» слід застосовувати, якщо:

- кількість користувачів перевищує 10,
- комп'ютери розподілені на значній території,
- необхідне забезпечити безпечну роботу (через централізоване управління ресурсами),
- присутня необхідність використання спеціалізованих серверів,
- необхідно поділяти ресурси на рівні груп користувачів.

2.5. Архітектура «комп'ютер-мережа»

Архітектура «комп'ютер-мережа» (computer-network architecture) – концепція комп'ютерної мережі, в якій програмне забезпечення надається користувачу як Інтернет-сервіс: користувач отримує доступ до даних, але не може управляти операційною системою і програмним забезпеченням, з яким він працює. До даного шаблону архітектури, що показаний на рис. 2.9, вживається також термін Cloud computing, що перекладається як «обчислення в хмарі», або як «хмарні обчислення». Ідеологія Cloud computing започаткована в 2007 р. та широко розвивається. завдяки розвитку каналів зв'язку.

Концепція Cloud computing дає користувачам недосяжні раніше можливості: обмежені в ресурсах компанії можуть дозволити собі власні бізнес-додатки і поштові сервери, реально маючи тільки доступ до Інтернет.

Прикладом Cloud computing є компанія Google, яка надає користувачам

необмежений дисковий простір для збереження електронної пошти (Google Mail) та стандартні офісні додатки в режимі on-line (Google Apps).

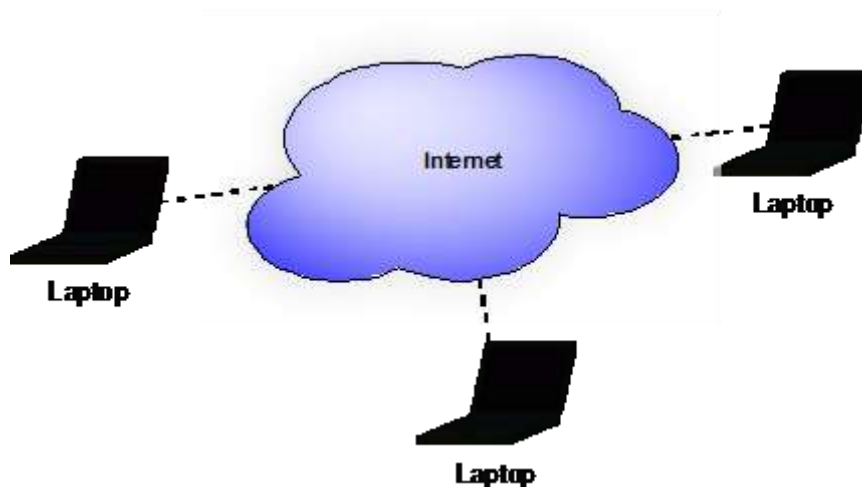


Рис. 2.9. Приклад шаблону архітектури «комп'ютер-мережа»

2.6. Архітектура інтелектуальної мережі

Поняття інтелектуальної мережі (Intelligent Network – IN) пов'язане з наданням користувачам комутованої телекомунікаційної мережі розширеного і постійно розширюваного набору послуг. Головна ідея полягає у виокремленні процесу традиційної комутації викликів від процесу введення нових послуг. Для цього потрібні певні інтерфейси між комутаторами мережі і «інтелектуальною надбудовою», що показана на рис. 2.10. Модернізація послуг в цьому випадку виконується лише шляхом модернізації програмного забезпечення «інтелектуальної надбудови», що дозволяє швидко впроваджувати на існуючих мережах будь-які послуги незалежно від виробника комунікаційного обладнання.

Відповідно до такої архітектури, для надання послуг впроваджують пункти комутації послуг (SSP – Service Switching Point) та пункти управління послугами (SCP – Service Control Point).

У пункті управління послугами SCP міститься основний «інтелект» у вигляді певних алгоритмів надання послуг і наборів баз даних.

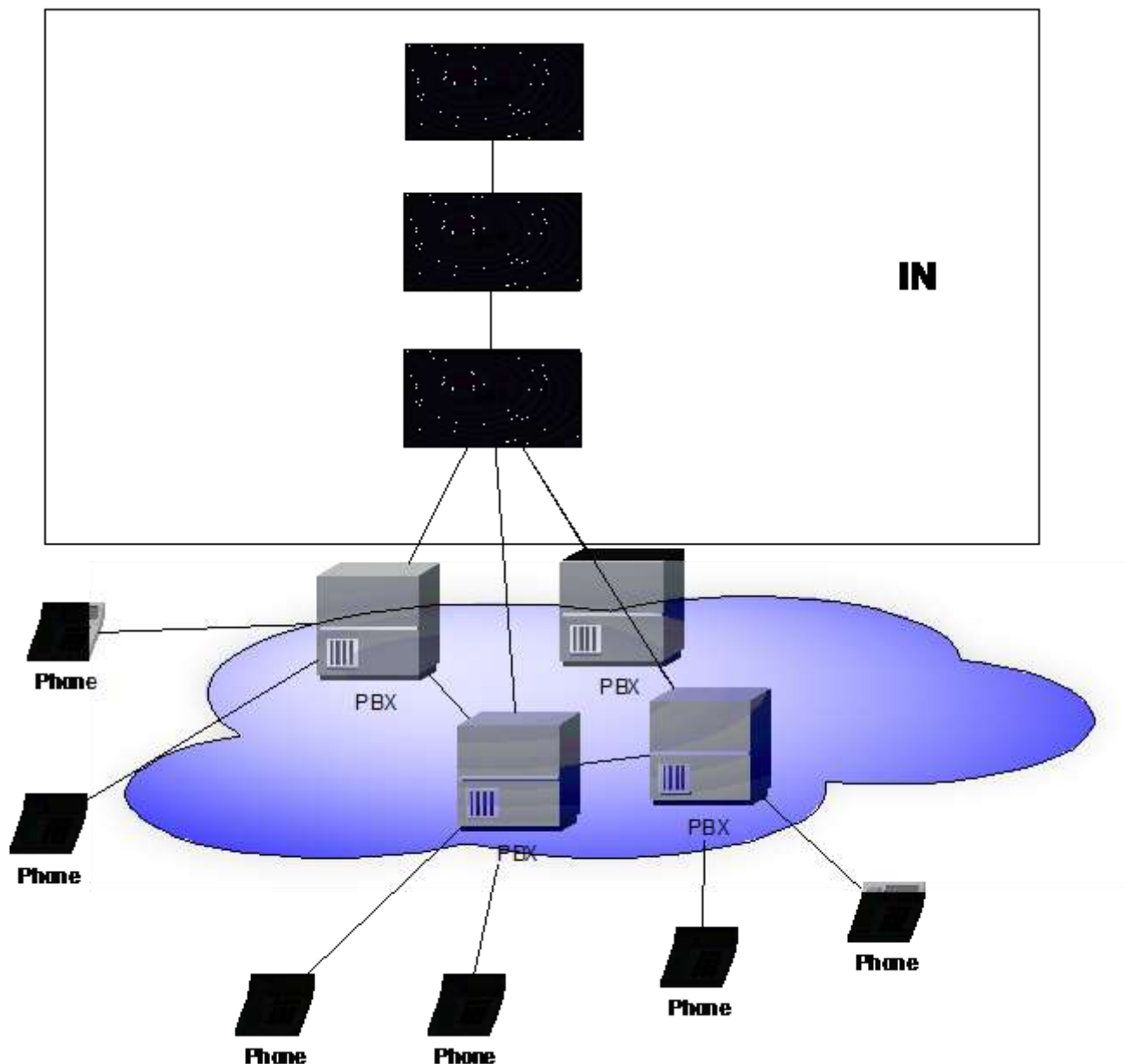


Рис. 2.10. Приклад архітектури інтелектуальної мережі

Пункт комутації послуг SSP розпізнає виклики до IN, що надходять з комп'ютерної мережі та направляє їх в SCP для оброблення, виступаючи в ролі інтерфейсу до «інтелектуальної надбудови».

Система управління послугами SMS (Service Management System) забезпечує введення нових послуг, корекцію старих, містить відомості про абонентів, оригінали програм.

Набір основних послуг інтелектуальної мережі:

- послуги безкоштовного виклику (послуга «800»),
- послуги з додатковою оплатою (наприклад, за консультації),
- послуги створення корпоративних мереж,
- послуги перенаправлення викликів та ін.

ІN підтримує телефонні мережі загального призначення, мережі радіотелефонного зв'язку, мережі передачі даних та інші.

Контрольні запитання до розділу

1. Види архітектури в мережі.
2. Переваги та недоліки основних архітектурних шаблонів мережі.
3. Порівняти однорангову та клієнт-серверну архітектури.
4. Порівняти хмарну та однорангову архітектури, їх переваги та недоліки.
5. Порівняти клієнт-серверну та хмарну архітектури.
6. Переваги та недоліки інтелектуальної мережі.

3. ВЗАЄМОДІЯ РІВНІВ ЕТАЛОННОЇ МОДЕЛІ OSI

3.1. Поняття відкритої системи

В широкому розумінні слова відкрита система – це система, що побудована у відповідності з відкритими специфікаціями. Де під специфікаціями розуміється формальний опис апаратних, або програмних компонентів, способів їх функціонування, взаємодії з іншими компонентами, умов експлуатації і особливих характеристик.

Під відкритими специфікаціями розуміються опубліковані, загальнодоступні специфікації, що прийняті зацікавленими сторонами в якості стандартів.

Використання відкритих специфікацій дозволяє:

- розробляти апаратні і програмні засоби для розширення і модифікації існуючих систем за допомогою третіх осіб;
- створювати комплексні системи з продуктів різних виробників.

Практично, повна відкритість систем часто виявляється недосяжною. Як правило, визначенню відкритих відповідає лише частина специфікацій, що підтримуються через зовнішні інтерфейси. Чим більше відкритих специфікацій використано для розробки, тим система є відкритішою.

Особливого значення відкриті специфікації набувають для побудови комп'ютерних мереж, що складаються з різноманітного комп'ютерного і комунікаційного обладнання і тому проблема сумісності для них є однією з найбільш гострих.

Дотримання принципів відкритості при побудові мереж надає наступні переваги:

- можливість використання апаратних і програмних засобів від незалежних виробників,
- можливість удосконалення окремих елементів мережі без зміни інших,

- можливість легкого об'єднання мереж,
- простота у освоєнні та обслуговуванні.

Яскравим прикладом відкритої системи є Інтернет. Сама назва стандартів, що забезпечують функціонування Інтернет – Request For Comments (RFC), тобто «запит на коментарі» – вказує на їх відкритий характер. В результаті Інтернет змогла об'єднати в собі найрізноманітніше обладнання і програмне забезпечення великої кількості мереж всього світу.

3.2. Багаторівневий підхід до організації мережевої взаємодії

Для вирішення складних задач, до яких належить і задача організації взаємодії між пристроями мережі, використовується універсальний прийом – декомпозиція.

Процедура декомпозиції показана на рис. 3.1 та включає:

- розбиття однієї складної задачі на кілька більш простих під задач – модулів,
- визначення функцій кожного модуля,
- визначення інтерфейсів для забезпечення взаємодії між модулями.

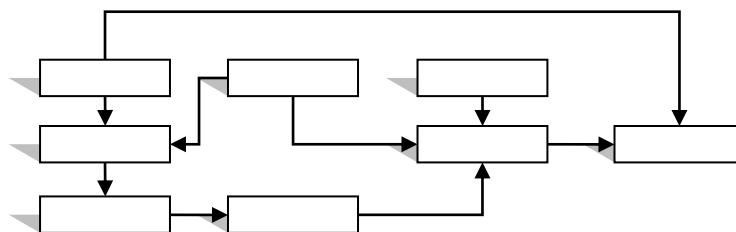


Рис. 3.1. Приклад декомпозиції складної задачі

Приклад декомпозиції задачі передачі повідомлення дано на рис. 3.2.

При декомпозиції задач використовують багаторівневий підхід:

1. Вся множина модулів поділяється на ієрархії рівнів: від вищих до нижчих.
2. Модулі певного рівня вибираються так, що для виконання своїх задач вони можуть звертатись з запитами лише сусідніх модулів нижчого рівня.

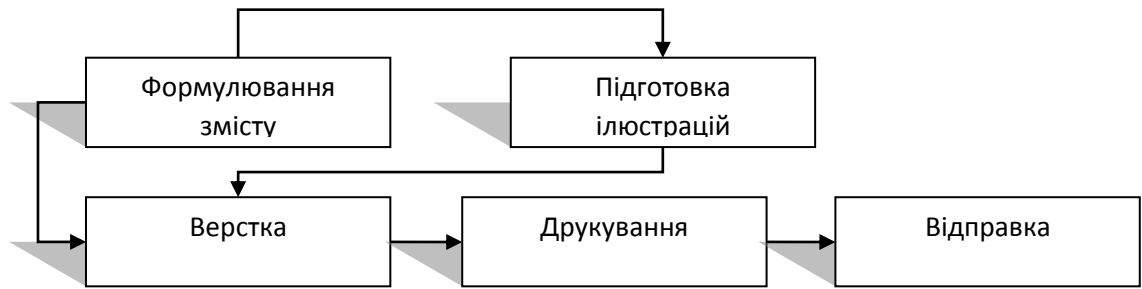


Рис. 3.2. Декомпозиція задачі передачі повідомлення

3. Результати роботи модулів певного рівня можуть передаватись лише сусіднім модулям верхнього рівня через інтерфейс взаємодії, показаний на рис. 3.3.

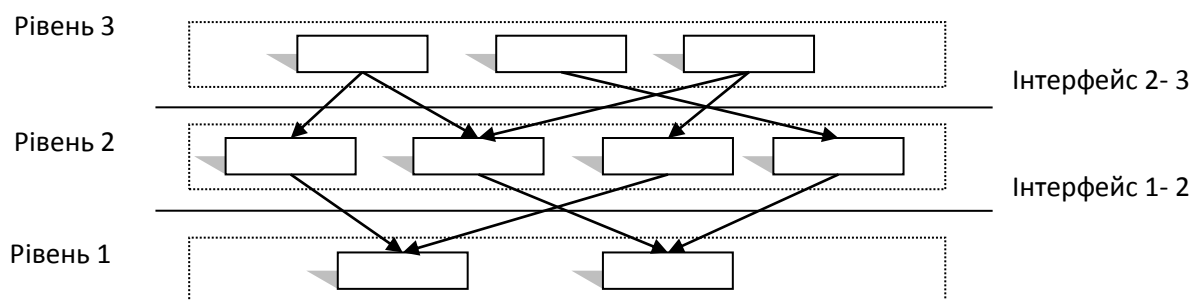


Рис. 3.3. Приклад декомпозиції складної задачі

Приклад 3-рівневої декомпозиції задачі передачі повідомлення наведено на рис. 3.4:

Засоби взаємодії в мережі також можна представити в вигляді ієрархічно організованої множини модулів, як, наприклад, показано на рис. 3.5.

Проте, взаємодія в мережі базується на участі пари взаємодіючих вузлів зв'язку. Тому потрібна організація узгодженої роботи модулів різних вузлів зв'язку, як показано на рис. 3.6.

Як видно з рис. 3.6, кожен модуль, або рівень, одного вузла зв'язку має взаємодіяти з таким же модулем, або рівнем, іншого вузла зв'язку так, ніби вони зв'язані безпосередньо. Такий зв'язок називається логічним, або

віртуальним. Одиницю даних, що передається між модулями одного рівня називають пакетом (packet).

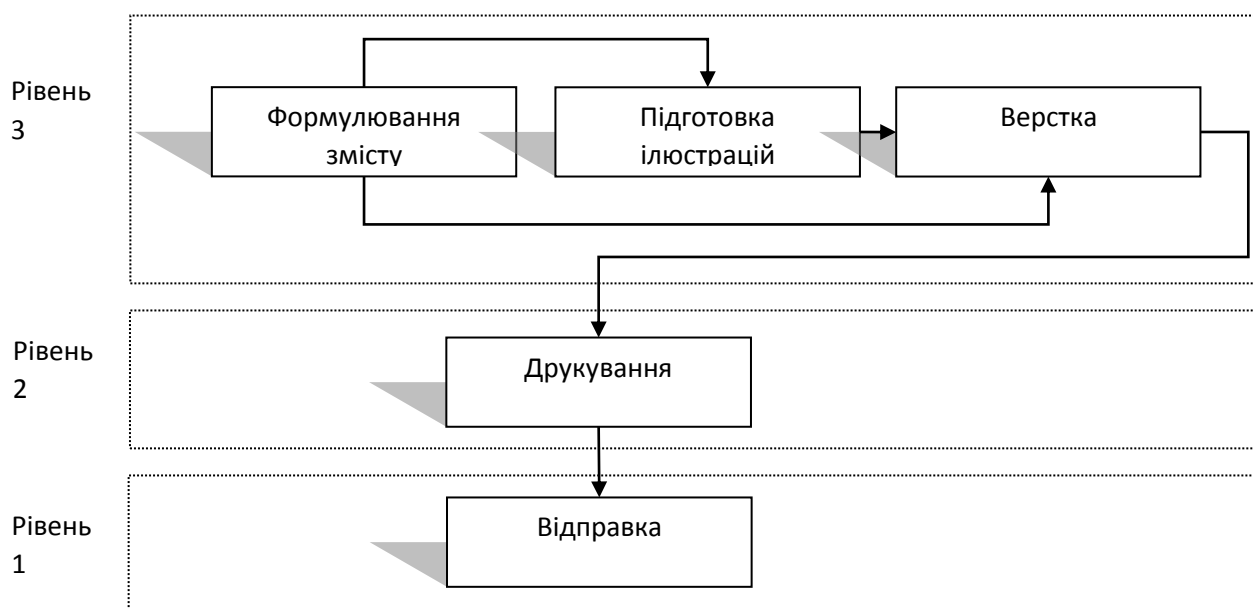


Рис. 3.4. Декомпозиція задачі передачі повідомлення

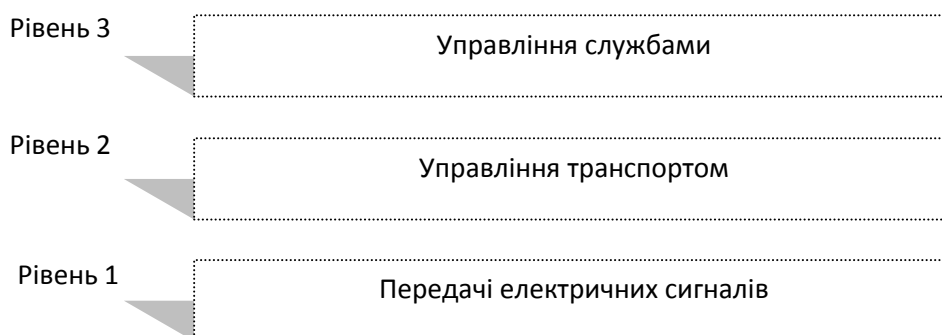


Рис. 3.5. Декомпозиція задачі взаємодії в мережі

У дійсності, при відправці інформації, дані проходять послідовно через всі нижчі модулі кожного з вузлів зв'язку. На кожному з них до пакета даних додається службова інформація, необхідна для виконання задач на даному рівні. Ця службова інформація міститься перед даними (заголовок) та після них (трейлер, хвіст), обрамляючи пакет, що надходить з вищого рівня. Такий

процес послідовної упаковки даних для передачі з одного рівня на інший, показаний на рис. 3.7, називається інкапсуляцією даних (data encapsulation).

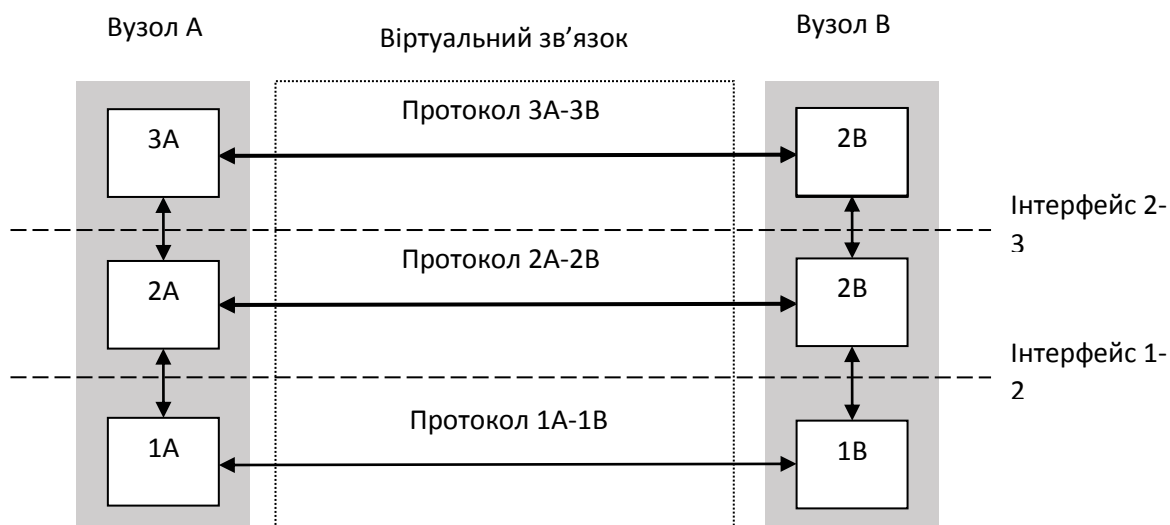


Рис. 3.6. Декомпозиція взаємодії пари вузлів зв'язку



Рис. 3.7. Інкапсуляція даних

На приймаючій стороні пакет проходить через усі рівні в оберненому порядку. При цьому на кожному рівні зчитується, а потім видаляється, службова інформація, що була додана на цьому рівні відправляючою стороною, після чого пакет передається на наступний рівень. Такий процес оберненої розпаковки даних називається декапсуляцією даних (data decapsulation).

Процедура взаємодії вузлів може бути описана у вигляді набору правил

взаємодії кожної пари відповідних рівнів обох сторін (горизонтальні зв'язки).

Формалізовані правила, які визначають послідовність і формат повідомлень для обміну між мережевими компонентами одного рівня, що належать різним вузлам, називають протоколом.

Модулі, що реалізують протоколи сусідніх рівнів і належать одному вузлу, також взаємодіють між собою у відповідності з визначеними правилами (вертикальні зв'язки). Ці правила називають інтерфейсом. Інтерфейс визначає набір операцій (сервісів), що надаються даним рівнем сусідньому рівню.

Ієрархічно організований набір протоколів, достатній для організації взаємодії вузлів у мережі, називається стеком комунікаційних протоколів.

Комунікаційні протоколи можуть бути реалізовані як програмно, так і апаратно. Найчастіше протоколи нижніх рівнів реалізуються комбінацією програмних і апаратних засобів, а протоколи верхніх рівнів – лише програмними.

3.3. Модель ISO

З метою допомоги виробникам в стандартизації програмного і апаратного забезпечення комп'ютерних мереж, Міжнародна організація з стандартизації (International Standard Organization – ISO) у 80-х рр. XX ст. разом з іншими розробила еталонну модель взаємодії відкритих систем - Open System Interconnection (OSI) Reference Model, що показана на рис. 3.8.

Модель OSI збудована на основі великого досвіду, набутого в 70-х рр. при створенні глобальних комп'ютерних мереж. Модель будувалася так, щоб розділити функції стеків протоколів і забезпечити можливість їх розробки незалежними організаціями, тобто щоб процес розробки протоколів став більш раціональним.

Модель OSI, визначає сім рівнів взаємодії систем і функції, які має виконувати кожний рівень.

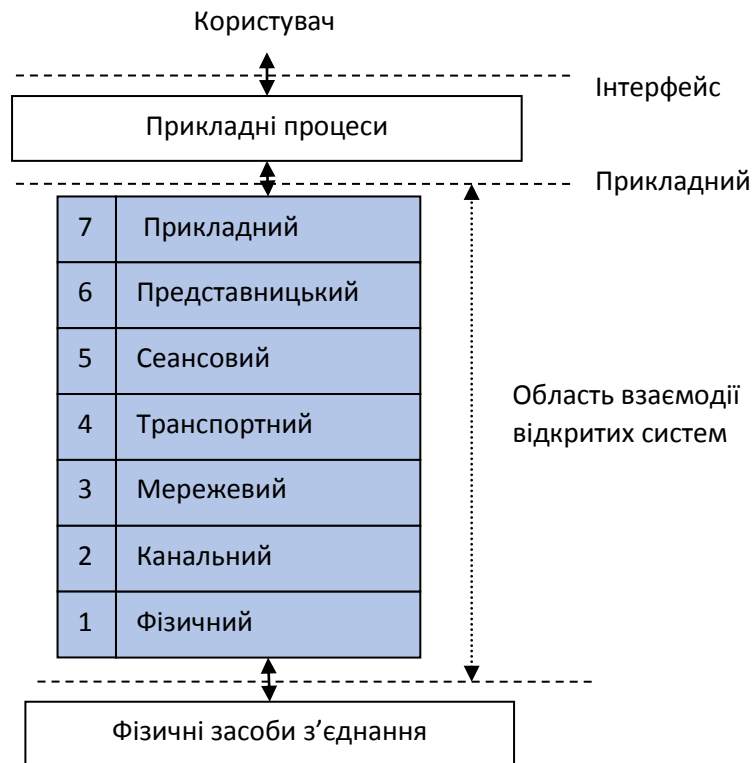


Рис. 3.8. Модель OSI

Модель OSI описує тільки системні засоби взаємодії (які реалізуються операційною системою, системними утилітами, системними апаратними засобами) і не включає засоби взаємодії з прикладними процесами користувачів. Свої власні протоколи взаємодії прикладні процеси реалізують, звертаючись до системних засобів. Тому необхідно розрізняти рівень взаємодії додатків і прикладний рівень. В табл. 3.1 показано призначення кожного з рівнів моделі та приклади протоколів.

Наступним питанням є опис блоків даних у протоколах – PDUs (Protocol Data Units). Термін пакет (packet), зокрема, визначає блок даних, що передається через мережеве середовище, хоча він також застосовується і для опису даних на будь-якій стадії процесу.

На фізичному рівні відбувається передача бітів (bits). На канальному рівні інформація сформована в кадри (frames). На мережевому рівні ще не сформовані кадри носять назву дейтаграми (datagrams).

Таблиця 3.1. Призначення рівнів моделі OSI

№	Рівень	Призначення	Приклади
7	Прикладний (Application)	Забезпечує послуги, що надаються безпосередньо прикладним програмам	SMTP, HTTP, FTP і т.п.
6	Представлення (Presentation)	Забезпечує кодування і перетворення даних. Цей же рівень здійснює шифрування і стиснення даних.	Стандарти кодування (GIF, JEPEG, TIFF MPEG і т.п.)
5	Сеансовий (Session)	Забезпечує проведення сеансів зв'язку (тобто установку, підтримку і переривання зв'язку). Цей же рівень розпізнає логічні імена абонентів, контролює надані їм права доступу.	Remote Procedure Call, Session Control Protocol (SCP)
4	Транспортний (Transport)	Забезпечує доставку даних від одного вузла до іншого без помилок і втрат, а також в необхідній послідовності, через їх розбивку на пакети і нумерування пакетів. Доставка пакетів можлива як з встановленням з'єднання (віртуального каналу), так і без нього.	TCP, UDP
3	Мережевий (Network)	Забезпечує логічну структурування мережі і маршрутизацію пакетів між підмережами. Цей же рівень здійснює перетворення мережевих адрес в фізичні (наприклад, IP-адрес в MAC-адреси).	IP
2	Канальний (Data Link)	Забезпечує надійну передачу даних в рамках підмережі з тим чи іншим каналом зв'язку (шляхом формування низькорівневих кадрів для даного виду підмережі)	Ethernet, Token Ring, FDDI, Frame Relay, PPP (Point-to-Point Protocol)
1	Фізичний (Physical)	Забезпечує умови прийому-передачі по фізичному каналу зв'язку шляхом визначення вимог до його фізичних, механічних, електричних та інших характеристик (рівні напруги, частота, опір і т.п.)	LAN категорії 3, LAN категорії 5, V.90

На транспортному рівні інформація розбивається на сегменти (segments). Відповідно, на прикладному рівні інформація розглядається як повідомлення (messages). Протоколи представницького і сеансового рівня не формують своїх заголовків і тому до даних у них застосовують термін повідомлення.

З урахуванням цього модель взаємодії відкритих систем набуває вигляду, показаного на рис. 3.9.

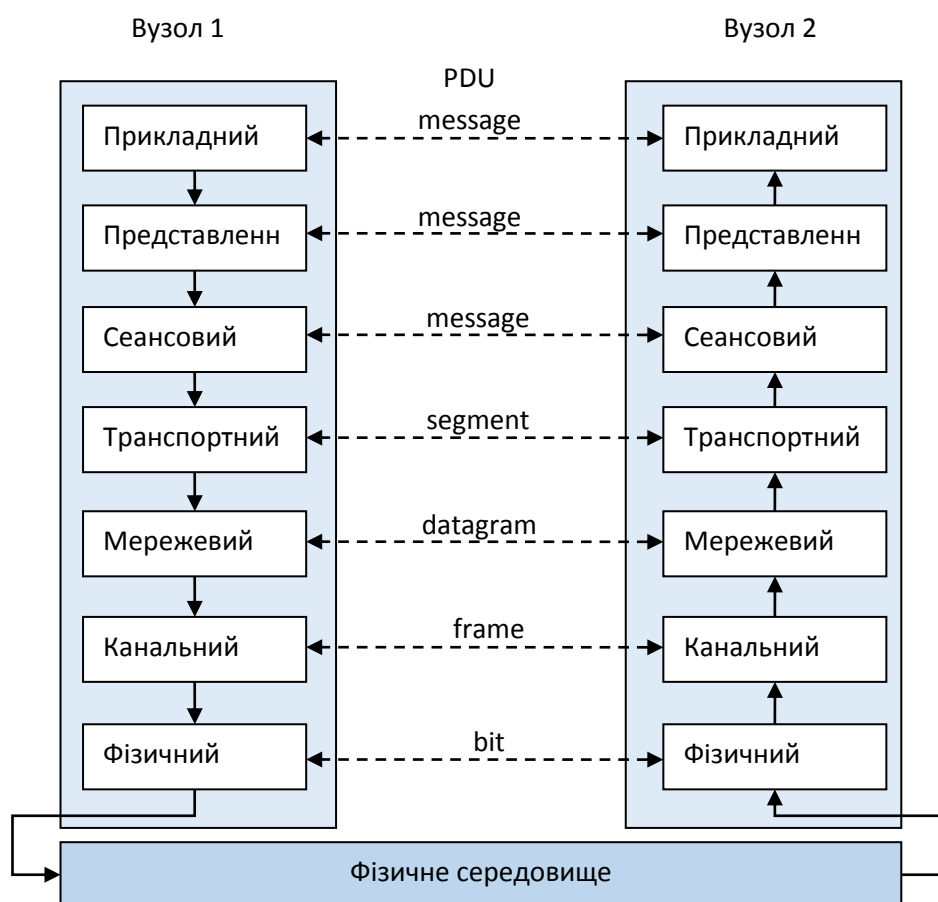


Рис. 3.9. Приклад декомпозиції складної задачі

На рис. 3.10 показано, що чотири нижніх рівні моделі OSI забезпечують виключно транспортні функції, які реалізуються за допомогою відповідних апаратних засобів (мережева карта, повторювач, концентратор, комутатор, маршрутизатор). Їх прийнято називати мережевим транспортом. А три верхніх рівні є виключно програмною надбудовою над мережевим

транспортом і їх задача полягає у наданні послуг мережі прикладним додаткам.



Рис. 3.10. Приклад декомпозиції складної задачі

3.4. Протокольна технологія

Розвиток мережевих технологій привів до появи нової області обчислювальної техніки – протокольної технології. Ця технологія ґрунтується на формальних методах опису протоколів, що дозволяє виконувати аналіз (верифікацію) описів, а також автоматизувати процес трансляції цих описів безпосередньо в машинну реалізацію.

Формальні методи опису протоколів поділяються на дві групи:

- автоматні методи – розглядають об’єкт як «автомат»,
- методи послідовностей – розглядають об’єкт як «чорний ящик».

В якості представника першої групи може бути наведена мова ESTELLE (Extended State Transition Language), другої групи – мова LOTOS (Language of Temporal Ordering Specification). Ці мови були розроблені Міжнародною організацією з стандартизації (ISO) та є базовими засобами опису міжнародних стандартів протоколів.

Контрольні запитання до розділу

1. Поняття відкритої системи та її переваги.
2. Аналіз та синтез задачі передачі повідомлення.
3. Інкапсуляція даних, пакети даних.
4. Описати всі рівні моделі взаємодії відкритих систем, призначення та приклади.
5. Поняття протоколів та їх призначення.

4. ВЕРХНІ РІВНІ МОДЕЛІ OSI

4.1. Прикладний рівень

Основна задача прикладного рівня (Application Layer) – організація взаємодії OSI з прикладними програмами користувача, що лежать за межами OSI, як показано на рис. 4.1.

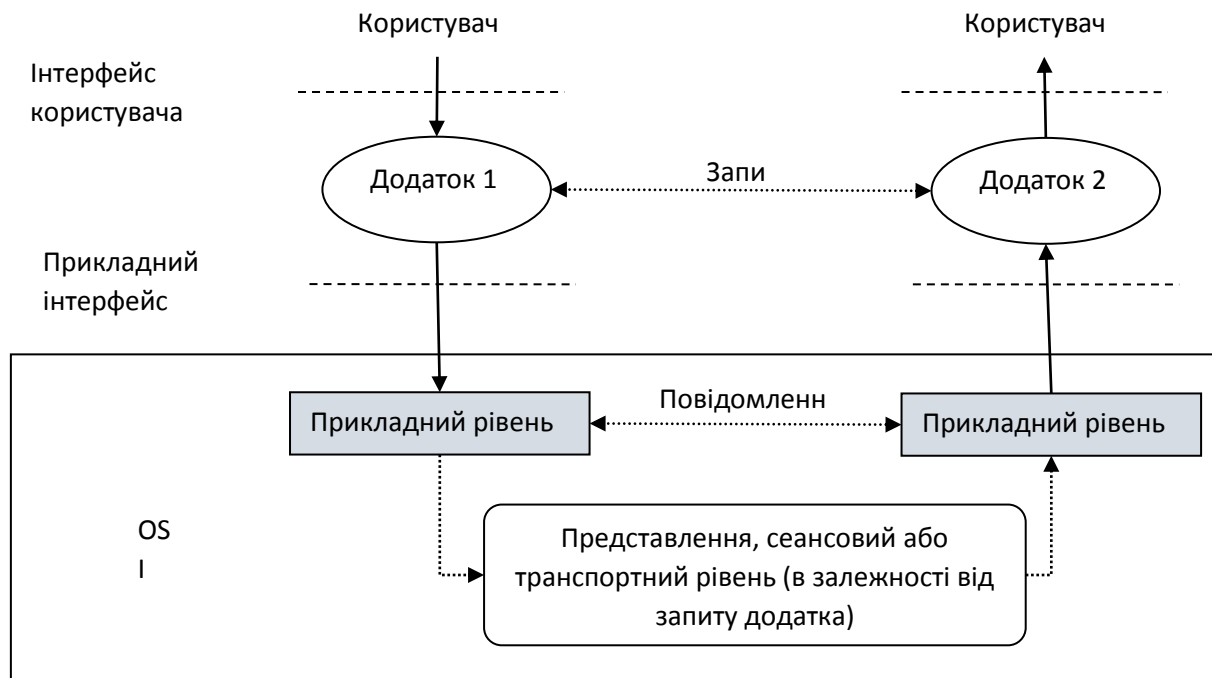


Рис. 4.1. Прикладний рівень між 1 та 2 вузлами зв'язку

Прикладний інтерфейс визначає набір операцій (примітивів), які надаються користувачу для формування запитів на отримання послуги (сервісу). Наприклад, в межах шаблону мережі “клієнт-сервер”, набір цих примітивів може бути таким: LISTEN (очікування), CONNECT (з'єднання), RECEIVE (прийом), SEND (відправка), DISCONNECT (розрив з'єднання).

Якщо набір протоколів входить до складу операційної системи (як це

зазвичай і буває), то примітиви є системними викликами. Вони приводять до виникнення системних переривань, в результаті чого управління комп'ютером передається операційній системі, яка і пересилає пакети.

В загальному випадку процеси прикладного рівня не є синонімами самих додатків. Наприклад, при використанні текстового процесора для перегляду файлу на сервері, сам процесор не має процесу для доступу до файлу і тому мусить відправити відповідний запит до своєї операційної системи.

Аналогічно і інші додатки можуть звертатися до протоколів, що використовують специфічні типи запитів у мережі. До них належать:

- SMTP (Simple Mail Transfer Protocol) – простий протокол пересилання пошти,
- POP3 (Post Office Protocol) – поштовий протокол доставки,
- HTTP (Hypertext Transfer Protocol) – протокол передачі гіпертексту,
- NNTP (Network News Transfer Protocol) – протокол передачі новин у мережі,
- SNMP (Simple Network Management Protocol) – простий протокол управління мережею.

Проте деякі додатки розроблені спеціально для доступу до мережевих ресурсів. Наприклад, спеціалізований додаток FTP-клієнта невіддільний від відповідного протоколу прикладного рівня – FTP (File Transfer Protocol). Інші протоколи, які нерозривно зв'язані з додатками, що їх використовують:

- DHCP (Dynamic Host Configuration Protocol) – протокол динамічної конфігурації хостів,
- TFTP (Trivial File Transfer Protocol) – найпростіший протокол передачі файлів,
- DNS (Domain Name System) – система доменних імен.

4.2. Рівень представлення даних

Рівень представлення даних (Presentation Layer) має лише одну функцію – перекладу. Він отримує примітиви від прикладного рівня і передає їх дублікати далі, на сеансовий рівень. Основні функції примітивів не змінюються в ході проходження ними рівня представлення, хоча можуть бути «перекладені» в термінах іншого синтаксису. Адже додаток звертається з запитом до мережевих ресурсів, використовуючи свій «рідний» синтаксис, проте синтаксис вузла зв'язку, що приймає запит, може бути іншим. Наприклад, з'єднання персонального комп'ютера з мейнфреймом може вимагати перекодування повідомлення (з ASCII в EBCDIC). Вузли зв'язку також можуть застосовувати шифрування та/або стиснення даних.

Обробка повідомлень на рівні представлення даних визначається специфікацією ASN.1 (Abstract Syntax Notation One). Відповідно до неї кожний з комп'ютерів має підтримувати абстрактний синтаксис, який є «рідним» для його додатку, і синтаксис передачі, який використовується для доставки повідомлень через мережу. «Переклад» здійснюється в два етапи:

- спочатку представницький рівень на боці відправника переводить дані з формату абстрактного синтаксису в формат синтаксису передачі,
- потім представницький рівень на боці одержувача переводить дані з формату синтаксису передачі в формат абстрактного синтаксису.

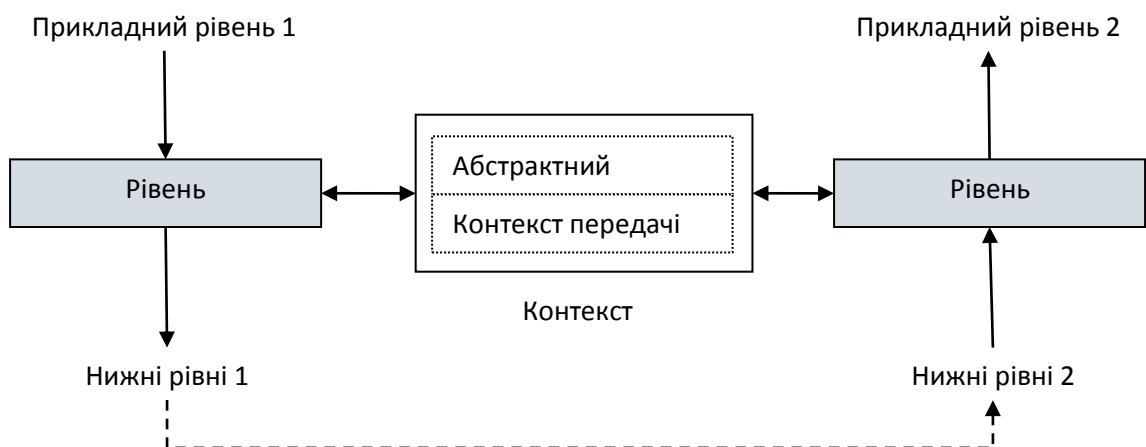


Рис. 4.2. Обробка повідомлень між 1 та 2 вузлами зв'язку на рівні представлення

Вибір синтаксису передачі для кожного абстрактного синтаксису ґрунтується на домовленості, яка досягається після встановлення з'єднання рівня представлення по моделі OSI між вузлами зв'язку.

Процес узгодження синтаксису починається з застосування примітиву P-CONNECT для передачі контексту представлення. Викликається відправка повідомлення P-CONNECT, яке може містити кілька контекстів передачі для кожного абстрактного контексту, як показано на рис. 4.3.

Отримавши повідомлення P-CONNECT, одержувач передає контекст процесам прикладного рівня, які вирішують, який контекст передачі буде використовуватись. В ньому для кожного абстрактного контексту вказується один вибраний контекст передачі, або помилка. Після отримання цього списку відправником він стає набором оговорених контекстів.

Після завершення процесу узгодження вузли зв'язку можуть запропонувати новий контекст представлення і додати (або видалити) його до набору оговорених контекстів.

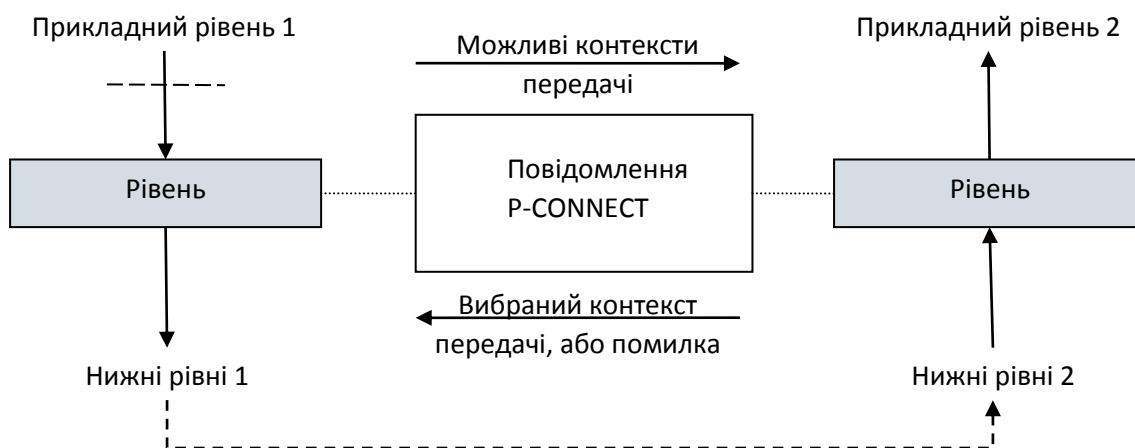


Рис. 4.3. Процес узгодження синтаксису між 1 та 2 вузлами зв'язку

Прикладом протоколу рівня представлення є протокол захищених сокетів - SSL (Secure Socket Layer), розроблений компанією Netscape 1995 р.

для захисту комерційної інформації в WWW від несанкціонованого доступу.

Протокол SSL отримав велику популярність і широке розповсюдження, хоча і не став офіційним стандартом Інтернет. Він інтегрований у більшість браузерів та веб-серверів і використовує асиметричну криптосистему з відкритим ключем. Підключення по SSL завжди ініціюється клієнтом через виклик URL, що починається з https:// (замість http://).

Протокол SSL забезпечує:

- аутентифікацію на базі цифрових сертифікатів,
- безпечний обмін даними по встановленому мережевому з'єднанню.

Цифрові сертифікати дозволяють:

- встановити особу власника,
- зробити доступним відкритий ключ власника.

Цифровий сертифікат випускається перевіреною уповноваженою організацією – центром сертифікатів (Certificate Authority) і видається на обмежений час.

Протокол SSL підтримує два типи аутентифікації:

- з боку клієнта – дозволяє клієнту перевірити істинність сервера,
- з боку сервера - дозволяє серверу перевірити особу користувача.

При аутентифікації з боку клієнта процес зв'язку складається з наступних етапів, що показані на рис. 4.4:

1. Клієнт ініціює запит до сервера на встановлення безпечного з'єднання.
2. У відповідь на запит, сервер передає клієнту свій цифровий сертифікат з відкритим ключем.
3. Сервер генерує і пересилає клієнту ключ сесії, зашифрований за допомогою його закритого ключа.
4. Клієнт розшифровує ключ сесії, використовуючи для цього отриманий раніше відкритий ключ сервера, і розпочинається сесія безпечного зв'язку на основі симетричного криптографічного алгоритму.

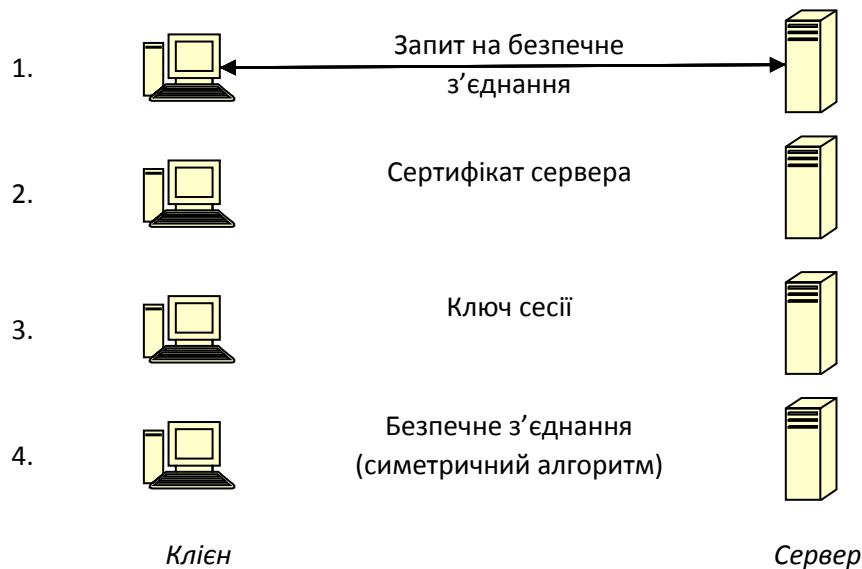


Рис. 4.4. SSL аутентифікація з боку клієнта

При аутентифікації з боку сервера процес зв'язку складається з таких етапів:

1. Клієнт ініціює запит до сервера на встановлення безпечного з'єднання.
2. Сервер запитує у клієнта цифровий сертифікат.
3. Клієнт відправляє серверу свій цифровий сертифікат з відкритим ключем.
4. Сервер відправляє клієнту свій цифровий сертифікат з відкритим ключем.

Розпочинається сесія безпечного зв'язку на основі асиметричного криптографічного алгоритму з використанням відкритих і закритих ключів клієнта та сервера.

4.3. Сеансовий рівень

Головні функції сеансового рівня (Session Layer) пов'язані з обміном повідомленнями між двома кінцевими вузлами зв'язку [2]. Цей обмін називається діалогом (dialog). Два найбільш важливих сервіси цього рівня –

управління діалогом і розділення діалогу.

Управління діалогом (dialog control) – це сервіс, що дозволяє двом вузлам зв'язку розпочати діалог, обмінятись повідомленнями, а потім закінчити діалог з упевненістю, що кожен вузол зв'язку отримав призначені для неї дані.

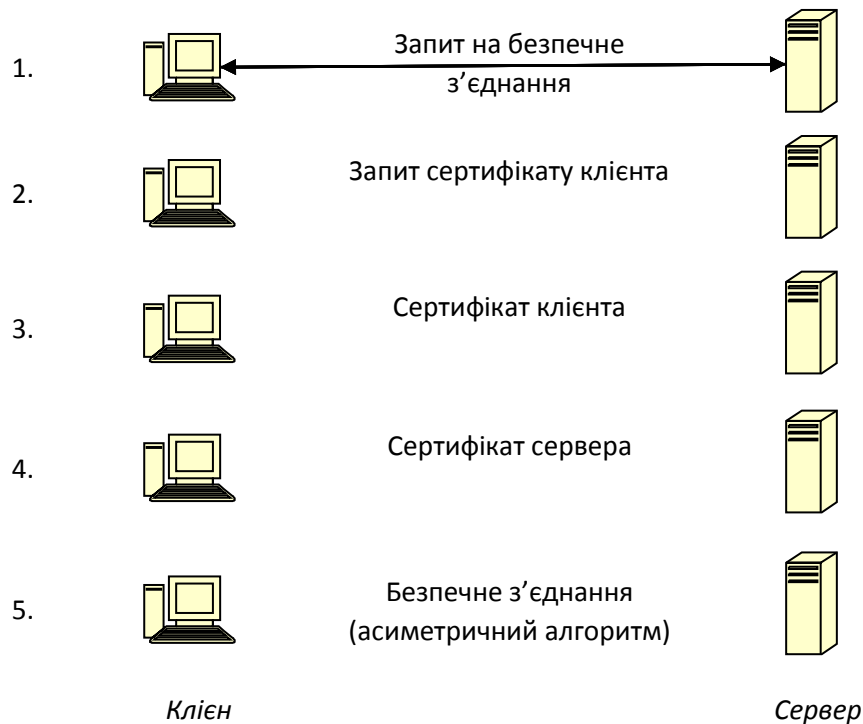


Рис. 4.5. SSL аутентифікація з боку сервера

Управління діалогом передбачає вибір одного з двох режимів управління обміном повідомленнями:

- напівдуплексний (half duplex) – взаємодіючі вузли зв'язку передають і приймають дані по черзі,
- дуплексний (full duplex) – взаємодіючі вузли зв'язку передають і приймають дані одночасно.

Вибраний режим не підлягає зміні протягом сеансу. Зміна режиму можлива тільки після розірвання з'єднання і встановлення його заново.

В напівдуплексному режимі в кожний момент часу тільки один вузол

зв'язку може передавати повідомлення. Право передачі визначається за допомогою маркера даних (data token). Тільки отримавши маркер даних вузол зв'язку може розпочати передачу.

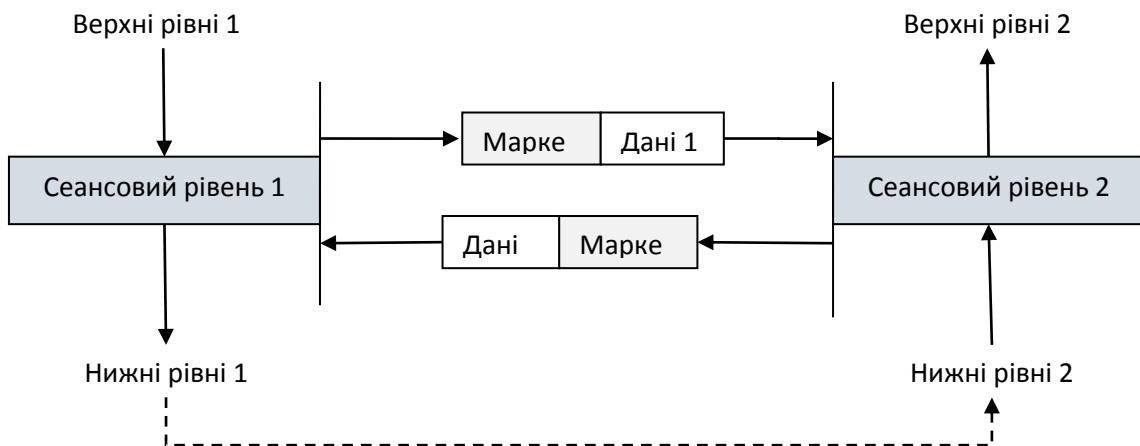


Рис. 4.6. Напівдуплексний режим обміну даними між 1 та 2 вузлами зв'язку

При дуплексному обміні маркер даних не використовується тому обидва вузла даних можуть передавати повідомлення одночасно. Тому використання дуплексного режиму значно ускладнює процес обміну.

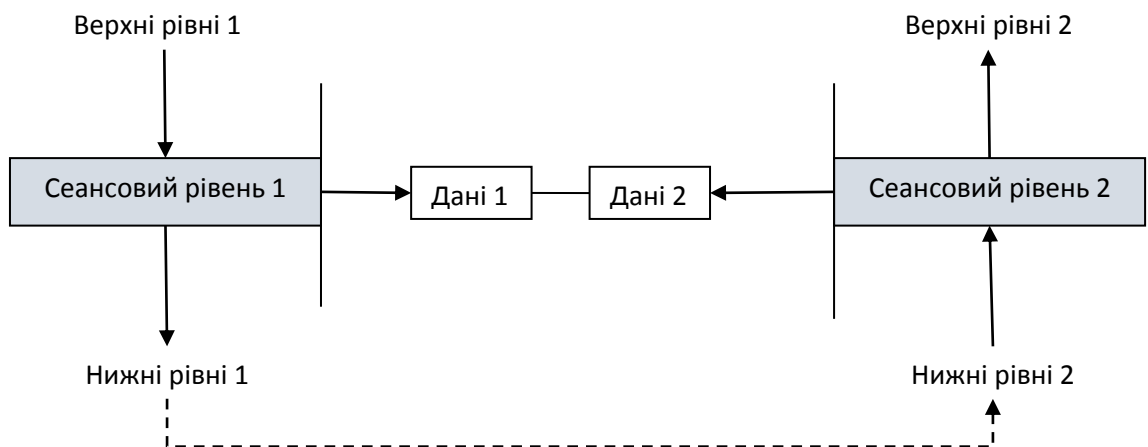


Рис. 4.7. Дуплексний режим обміну даними між 1 та 2 вузлами зв'язку

Використання маркера даних, зокрема, забезпечує механізм упорядкованого завершення (orderly termination). Згідно з цим механізмом:

1. Один з вузлів зв'язку сигналізує про намір розірвати з'єднання і передає маркер.

2. Інший вузол зв'язку, отримавши маркер, пересилає всі дані, що залишились в буфері, і підтверджує прийом запиту на роз'єднання.

3. Отримавши підтвердження (а значить – всі дані), перший вузол зв'язку розриває з'єднання.

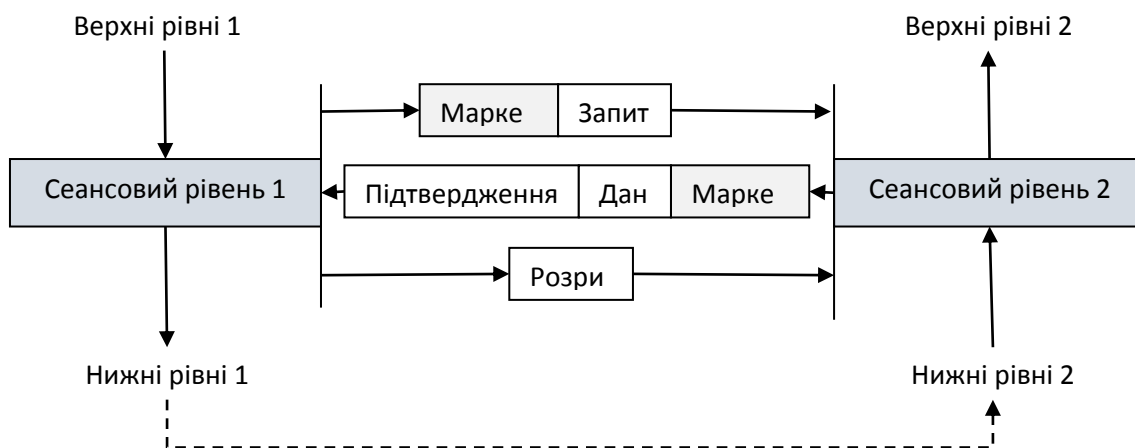


Рис. 4.8. Механізм упорядкованого завершення між 1 та 2 вузлами зв'язку

В управлінні діалогом передбачений також механізм погодженого роз'єднання (negotiated release). Він надає можливість одному вузлу зв'язку відмовити іншому в роз'єднанні. Ця можливість використовується у випадку колізії, коли обидва вузли зв'язку зроблять запит на роз'єднання (в дуплексному режимі). Для упередження подібних колізій використовується маркер роз'єднання (release token).

Розділення діалогу (dialog separation) – це сервіс, що дозволяє зафіксувати в один і той самий момент часу інформацію про стан обох вузлів зв'язку.

Сервіс базується на вставці в потік даних спеціальних вказівників – точок контролю (ТК) (checkpoints).

Контрольні точки створюються прикладними додатками, щоб зберегти свій поточний стан на диску у випадку системного збою. Виконати узгоджену дію на двох різних комп'ютерах в точно визначений момент часу майже неможливо. Розділення діалогу дозволяє зафіксувати стан обох вузлів

зв'язку одночасно.

В напівдуплексному режимі процес створення контрольних точок відносно простий і показаний на рис. 4.9:

- один вузол зв'язку створює точку контролю і відправляє повідомлення про це,
- інший вузол зв'язку, отримавши це повідомлення, створює свою точку контролю, будучи впевненим, що стан першого при цьому не змінився.

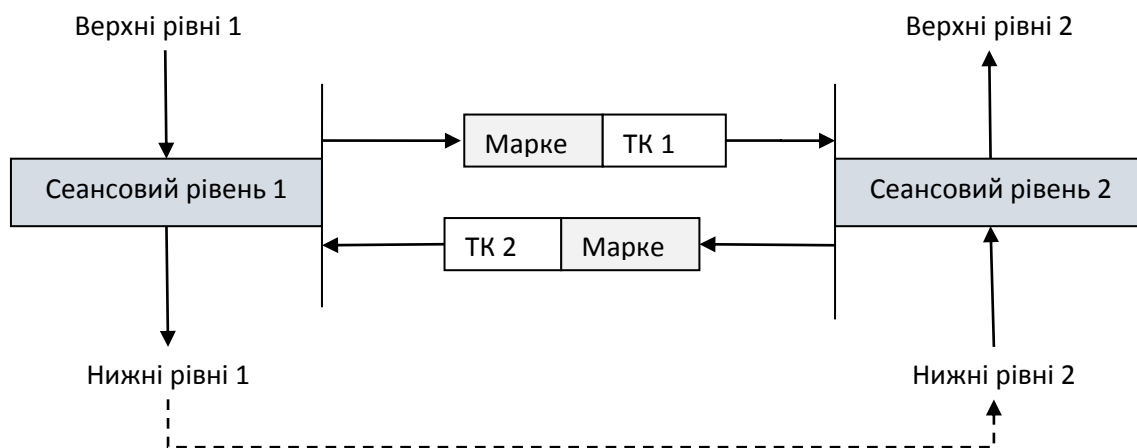


Рис. 4.9. Процес створення контрольних точок в напівдуплексному режимі

Такий процес називається простою синхронізацією (minor synchronization). В дуплексному режимі також можливе застосування простої синхронізації. Для цього використовується спеціальний маркер простої синхронізації.

Контрольні запитання до розділу

1. Верхні рівні моделі OSI
2. Прикладний рівень, протоколи прикладного рівня.
3. Рівень представлення даних, характеристика протоколу SSL
4. Сеансовий рівень, дуплексний та напівдуплексний режими обміну даними на сеансовому рівні.
5. Поняття маркера даних, розділення діалогу та точок контролю.

5. НИЖНІ РІВНІ МОДЕЛІ OSI

5.1. Транспортний рівень

Протоколи транспортного рівня (Transport Layer) забезпечують надійну передачу даних для протоколів більш високих рівнів і прикладних додатків. Транспортний рівень забезпечує наскрізну передачу даних між абонентами мережі.

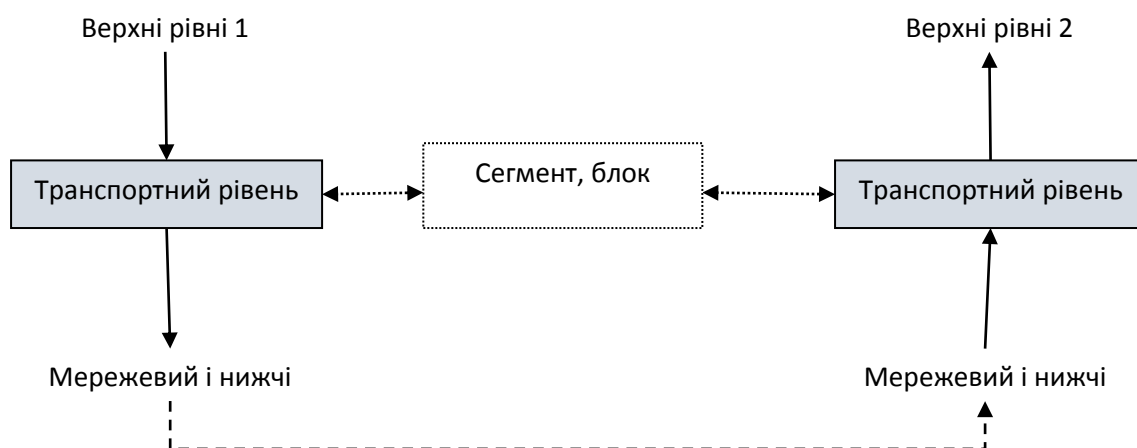


Рис. 5.1. Передача даних блоками на транспортному рівні

Локальна мережа (з надійними лініями зв'язку) може обійтись методами відновлення втрачених даних більш низьких рівнів, не витрачаючи обчислювальні ресурси на реалізацію складних методів корекції помилок на транспортному рівні. Однак, будь-яка ненадійна лінія (глобальна мережа) потребує контролю за помилками саме з боку протоколів транспортного рівня.

В моделі OSI теоретично передбачені п'ять класів протоколів транспортного рівня (TP0, TP1, ..., TP4). Проте більшість стеків протоколів, таких як TCP/IP, мають у своєму складі лише два протоколи (відповідають класам TP0, TP4), які забезпечують:

- надання послуг без встановлення з'єднання (connectionless

protocol),

- надання послуг із встановленням з'єднання (connection oriented).

Протокол без встановлення з'єднання просто упаковує дані і відправляє їх за адресом призначення без перевірки того, що відправник доступний, і очікування підтверджень про отримання даних.

У стеку протоколів TCP/IP протоколом без встановлення з'єднання є UDP (User Datagram Protocol). Він не гарантує доставку даних, які в цьому випадку називають дейтаграмами. Типова транзакція складається з запиту і відповіді, яка виконує функцію підтвердження прийому. Приклад – транзакції системи доменних імен DNS (Domain Name System).

Протокол із встановленням з'єднання спочатку встановлює логічний зв'язок між системою-відправником і системою-одержувачем. Після передачі даних відправником одержувач повинен підтвердити їх надходження. Відсутність підтвердження розглядається відправником як сигнал для повторної передачі сегменту.

Використання протоколу з встановленням з'єднання гарантує успішну доставку даних. Ціна цього – додатковий трафік, що викликається повідомленнями про установку з'єднання, відправку підтверджень і роз'єднання.

У стеку протоколів TCP/IP протоколом з встановленням з'єднання є TCP (Transmission Control Protocol). Він надає повний спектр послуг з доставки, але ціна цього – більш високі затрати обчислювальних ресурсів.

Протоколи з встановленням з'єднання забезпечують виконання таких функцій:

- сегментацію. При передачі великої кількості даних вони розбиваються на сегменти, які нумеруються. Нумерація дозволяє приймаючій стороні відновити порядок сегментів, а також повідомити, який сегмент був ушкодженим;
- управління потоком (flow control). Ця функція надає можливість

приймаючій стороні вказати відправнику на необхідність зменшення швидкості передачі даних у випадках загрози перевантаженості системи і втрати даних. Наприклад, в TCP заголовок сегмента містить поле Window, в якому вказується допустима швидкість передачі;

- виявлення помилок. Забезпечує можливість виправлення помилок передачі даних через запит у відправника повторної передачі ушкоджених сегментів (зі зміненою контрольною сумою).

Як правило, функції транспортного рівня повністю реалізуються програмними засобами – на відміну від трьох нижніх рівнів, в реалізації яких важливе місце займають технічні засоби.

5.2. Мережевий Рівень

Мережевий рівень (Network Layer) служить для побудови єдиної транспортної системи, основу якої можуть складати різноманітні мережі, робота яких ґрунтується на відмінних принципах передачі даних.

Мережевий рівень відповідає за доставку даних між мережами.

В термінах мережевого рівня моделі OSI:

мережа – це сукупність комп'ютерів, що об'єднані між собою у відповідності з однією із стандартних топологій (шина, кільце, зірка) на основі одного певного протоколу канального рівня, визначеного для даної топології.

Мережевий рівень визначає два типи комп'ютерів, що можуть бути задіяні при пересилці пакетів:

- кінцеві системи – комп'ютер-відправник, або комп'ютер-одержувач дейтаграм.
- проміжні системи – маршрутизатори, або комутатори, що з'єднують дві і більше мереж і перенаправляють дейтаграми по шляху до місця призначення.

В кінцевих системах всі сім рівнів протоколів беруть участь у

створенні і отриманні пакетів.

Проміжні системи обробляють пакети та передають їх лише до мережевого рівня, як показано на рис. 5.2.

Протокол мережевого рівня зчитує адресу системи-одержувача і за нею визначає наступний пункт передачі. Якщо пунктом призначення є робоча станція локальної мережі, то проміжна система переправляє дейтаграму безпосередньо робочій станції. Якщо пункт призначення розташовується у віддаленій мережі, то проміжна система звертається до таблиці маршрутизації, щоб визначити маршрутизатор, який забезпечить проходження дейтаграм до місця призначення за найбільш ефективним маршрутом.

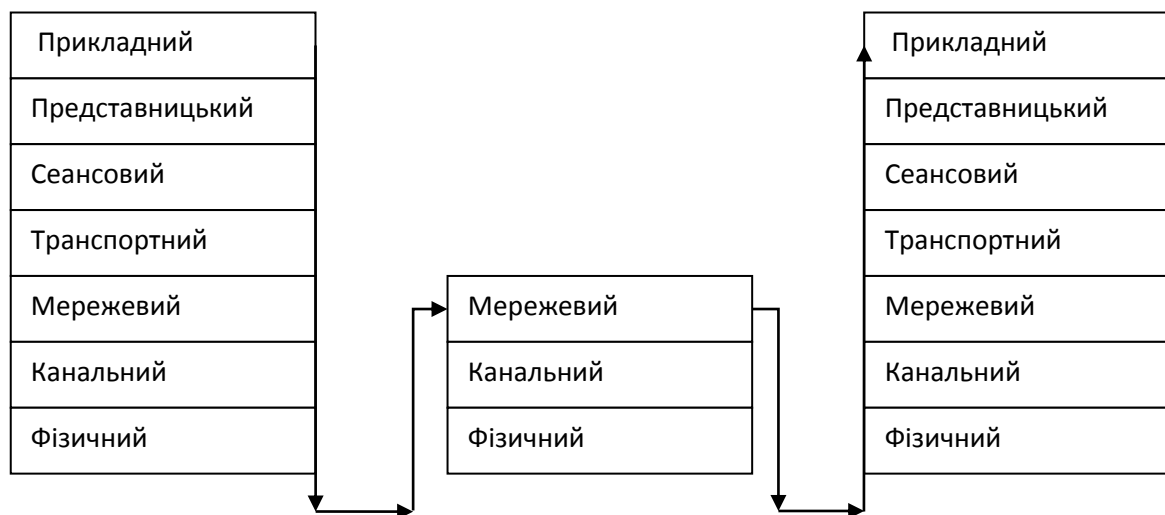


Рис. 5.2. Передача даних пакетами на мережевому рівні

Збір і збереження в таблиці маршрутів відомостей про можливі маршрути – окремий процес мережевого рівня. Він може здійснюватися вручну – адміністратором, або автоматично – спеціалізованими протоколами обміну між маршрутизаторами інформацією про можливі маршрути.

Прикладом протоколу мережевого рівня є протокол IP (Internet Protocol) з стеку протоколів TCP/IP. Приклад передачі пакетів за протоколом

IP на мережевому рівні показаний на рис. 5.3.

До мережевого рівня належить також протокол IPX стеку IPX/SPX.

Протоколи IP і IPX – це протоколи, що маршрутизуються (Router Protocols), тобто займаються доставкою інформації в мережі. До цього ж рівня відносяться специфічні протоколи, за допомогою яких маршрутизатори управляють трафіком, наприклад – протоколи маршрутизації (Routing Protocols). До них, наприклад, належать:

- RIP (Routing Information Protocol) – протокол обміну маршрутною інформацією.
- OSPF (Open Shortest Path First) – протокол маршрутизації з вибором найкращого маршруту.

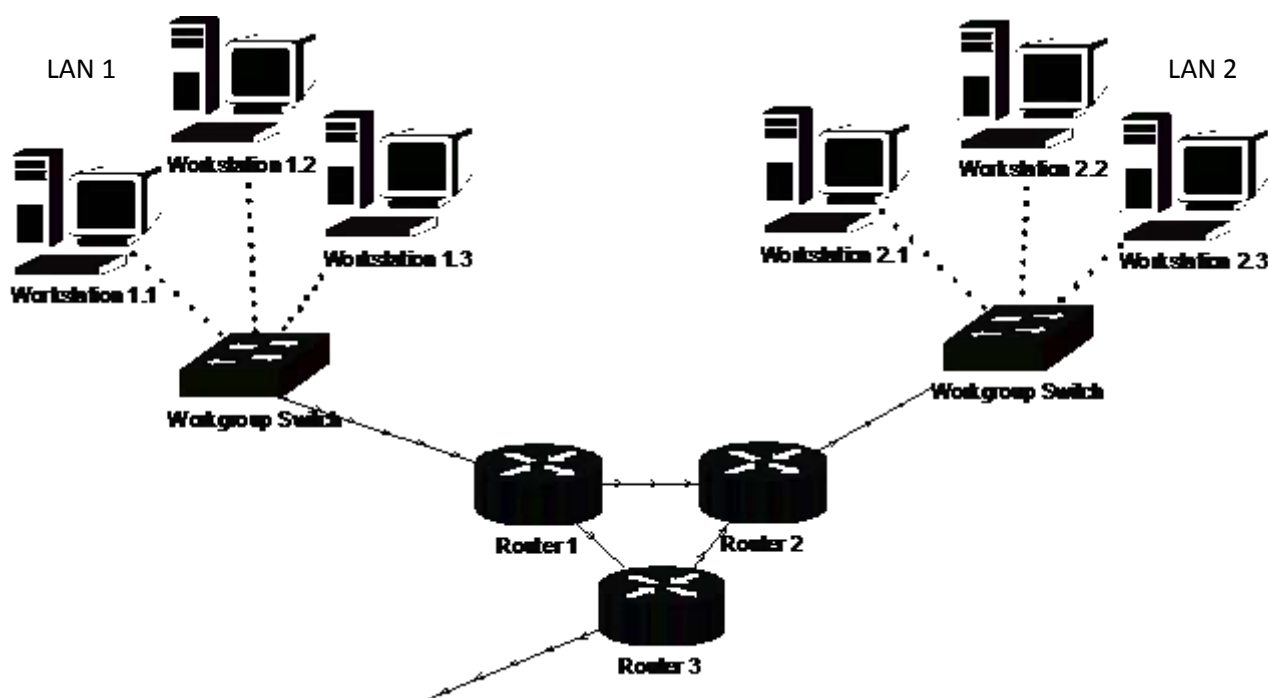


Рис. 5.3. Передача пакетів за протоколом IP на мережевому рівні

Слід зауважити, що більшість мережевого рівня є протоколами без встановлення з'єднання (наприклад, IP і IPX). В цьому випадку сервіси, орієнтовані на установку з'єднання, перекладаються на транспортний рівень.

Проте існують і протоколи мережевого рівня з встановленням з'єднання (наприклад, X.25).

5.3. Канальний рівень

Канальний рівень (Data Link Layer) відповідає за управління лінією передачі (Data Link) кадрів, як показано на рис. 5.4 [20, 22].

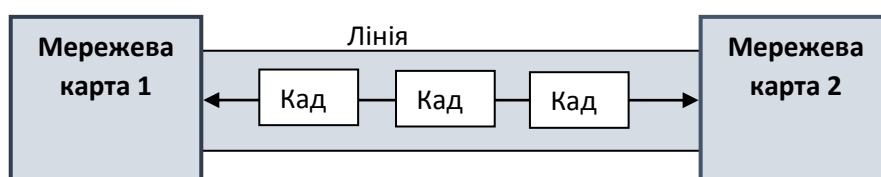


Рис. 5.4. Передача даних покадрово на канальному рівні

Одним з основних завдань канального рівня є здійснення контролю за доступом до спільно використовуваного середовища передачі даних – лінії передачі даних.

Для того, щоб мережа могла функціонувати ефективно, кожна станція, що підключена до лінії передачі, повинна мати можливість регулярно передавати дані. Це одна з причин чому дані при передачі трансформуються в кадри (frames). Адже якщо станції будуть передавати дані безперервним потоком, вони монополізуватимуть певний канал у мережі.

Найчастіше застосовуються два основних методи контролю доступу до середовища:

- доступ з контролем несучої і виявленням колізій (Carrier Sense Multiple Access with Collision Detection – CSMA/CD). Метод полягає у тому, що кожна станція прослуховує лінію передачі і передає дані тільки в тому випадку, якщо лінія вільна. Якщо ж кілька станцій починають передачу одночасно, виникає колізія. Кожна зі станцій вміє виявляти колізії і здійснювати повторну передачу,

– доступ з передачею маркера (Token Passing Multiple Access – TPMA). Метод ґрунтується на передачі від однієї станції до іншої спеціального кадру, який називається маркером (token). Тільки система, що захопила маркер, може здійснювати передачу, що виключає виникнення колізій.

На каналний рівень покладається також задача адресації кадрів з використанням апаратних (MAC) адрес. Такі адреси найчастіше «зашиваються» в мікросхему пам'яті мережевого адаптера. В мережах Ethernet, Token Ring використовуються MAC-адреси довжиною в 6 байт, перші три з яких визначають виробника, а останні три призначаються пристрою виробником.

Апаратні адреси відправника і одержувача розміщуються в заголовку протоколу. Всі комп'ютери, підключені до одного сегменту мережі, отримують кожен кадр, але лише станція, адреса якої співпадає з адресом одержувача, зчитує кадр в буфер пам'яті і обробляє його. Всі інші станції відкидають кадр без обробки.

Крім розглянутих функцій управління доступом і адресації кадрів каналний рівень здійснює також функцію виправлення помилок передачі даних. Для цього станція, що відправляє кадр, обчислює значення циклічного надлишкового коду (Cyclical Redundancy Check – CRC) для всього кадру і включає його в поле контрольної послідовності кадра FCS (Frame Check Sequence). Один з можливих способів обрахунку CRC – знайти залишок від ділення повідомлення на деякий відомий дільник. Коли кадр досягає місця призначення, одержувач виконує точно такі ж обчислення і порівнює результат із значенням поля FCS. Якщо значення не співпадають, кадр вважається пошкодженим і «відкидається». Одержувач не здійснює будь-яких дій, щоб передача пошкоджених кадрів повторилась. Ці дії виконують протоколи вищих рівнів моделі OSI.

5.4. Фізичний рівень

Фізичний рівень (Physical Layer) забезпечує передачу бітів даних фізичними лініям зв'язку, що показано на рис. 5.5.

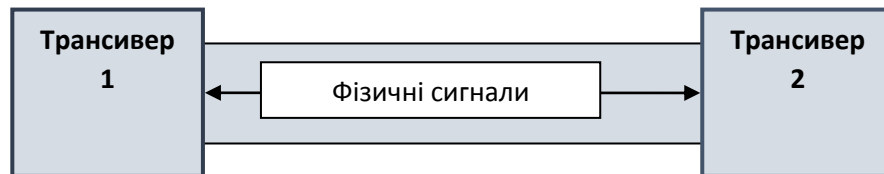


Рис. 5.5. Передача даних побітово на фізичному рівні

Специфікації фізичного рівня визначають:

- фізичні характеристики середовища передачі даних (полоса пропускання, затухання, хвильовий опір, час затримки і т.п).
- фізичні характеристики сигналів (амплітуда напруги, тривалість бітового інтервалу, форма і т.п.).
- умови використання ліній зв'язку (тип кабелю, обмеження на довжини сегментів, правила кодування бітів, стандарти роз'ємів і т.п.).

Серед мережевого обладнання фізичного рівня виділяється такий важливий компонент, як трансивер (transceiver), який, як правило, розміщується на платі мережевого адаптера. Трансивер відповідає за передачу і прийом сигналів з мережевого середовища. Трансивер об'єднує в собі трансмітер і ресивер, передач і приймач. Трансивер перетворює нулі і одиниці в напругу, світлові імпульси, радіохвилі. Сигнали, що виробляються трансивером, можуть бути як аналоговими, так і цифровими. Більшість мереж передачі даних використовують цифрові сигнали, але деякі безпроводні технології використовують аналогову радіоапаратуру.

Всі стандартні мідні і оптоволоконні середовища передачі даних використовують різні форми цифрових сигналів. Спосіб кодування сигналів визначається конкретним протоколом канального рівня. В мережі за

технологією Ethernet (для витної пари, коаксіального і оптоволоконного кабелів та ін.), наприклад, використовують манчестерську систему кодування. В ній двійкові (логічні) величини визначаються, виходячи з напрямку зміни полярності напруги: перехід, що здійснюється всередині бітового інтервалу, від додатного значення до від'ємного відповідає нулю, а від від'ємного до додатного – одиниці, рис. 5.6.

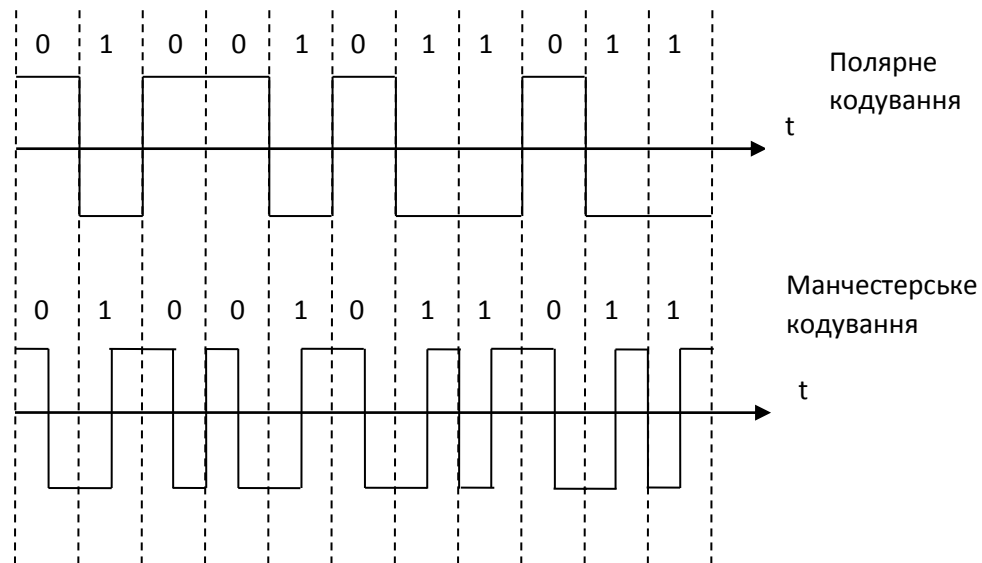


Рис. 5.6. Полярне та Манчестерське кодування побітових сигналів

На відміну від простішої полярної системи кодування, в якій підтримка протягом бітового інтервалу додатного значення кодує нуль, а від'ємного – одиницю, манчестерський спосіб кодування є таким, що само синхронізується (self-timing). Це є перевагою манчестерського способу кодування, оскільки від не залежить від помилок, що можуть бути при визначенні часу бітового інтервалу в полярній системі кодування.

5.5. Мережозалежні протоколи

Функції всіх рівнів моделі OSI можуть бути поділені на дві групи:

1. Мережозалежні – які залежать від технічної реалізації мережі.
2. Програмно-орієнтовані, або мережонезалежні – орієнтовані на

роботу з додатками.

Три нижніх рівні є мережозалежними – протоколи цих рівнів тісно пов’язані з технологічною реалізацією мережі, використовуваною апаратурою.

Три верхніх рівні орієнтовані на роботу з додатками. На протоколи цих рівнів не впливають ніякі зміни в технічній реалізації мережі.

Транспортний рівень є проміжним: він приховує деталі функціонування нижніх рівнів від верхніх. Це дозволяє розробляти прикладні додатки, що не залежать від технічних засобів транспортування даних.

Взаємодія станцій здійснюється через комунікаційні пристрої, які використовують відповідні протоколи нижніх рівнів, як показано на рис. 5.7.

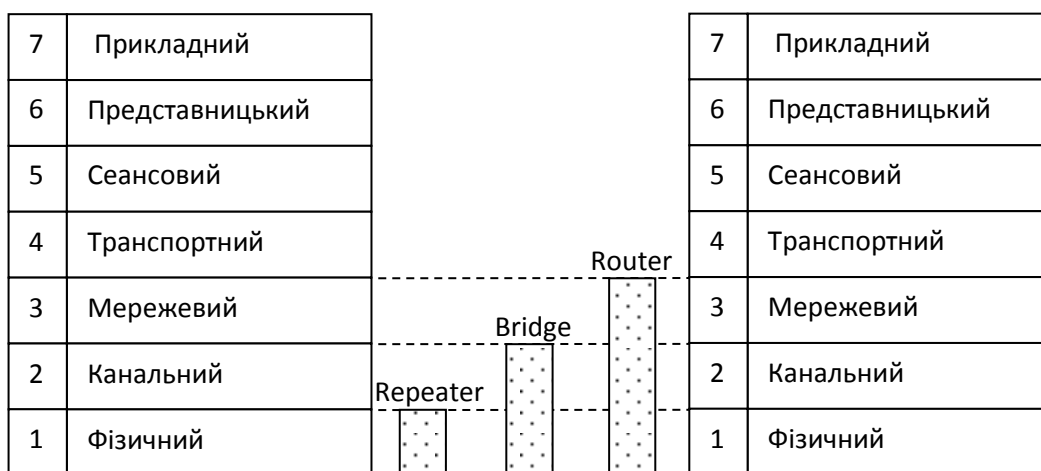


Рис. 5.7. Робота мережевих пристроїв в рамках моделі OSI

В залежності від типу, мережевий пристрій може працювати:

- тільки на фізичному рівні (повторювач – Repeater),
- на фізичному і канальному рівнях (міст – Bridge),
- на фізичному, канальному і мережевому рівнях (маршрутизатор – Router).

1. Нижні рівні моделі OSI.
2. Транспортний рівень, протоколи без та з встановленням з'єднання.
3. Мережевий рівень при пересилці пакетів, протоколи мережевого рівня.
4. Канальний рівень, методи контролю доступу до мережі, контрольна сума.
5. Фізичний рівень моделі, компоненти фізичного рівня, способи кодування сигналів на цьому рівні.
6. Рівні роботи мережевих пристроїв.

6. СІМЕЙСТВО СТАНДАРТІВ IEEE 802

6.1. Структура сімейства

Сімейство стандартів IEEE 802 містить рекомендації щодо проектування локальних мереж.

Робота над стандартами розпочалась в 1980 році в IEEE (Institute of Electrical and Electronics Engineers) після створення при ньому комітету зі стандартизації локальних мереж.

Перші стандарти IEEE 802 створювались на основі поширених технологій цілого ряду фірм – Ethernet (розробники – Digital, Intel, Xerox), ArcNet (розробник – Datapoint Corporation), Token Ring (розробник – IBM). Наступні стандарти вже з початку розроблялись не однією, а групою зацікавлених компаній і затверджувалися комітетом зі стандартизації. Наприклад, стандарти Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN.

Специфіку локальних мереж відображають нижні рівні моделі OSI, тоді як верхні в значній мірі мають спільні риси як для локальних, так і для глобальних мереж. Цим пояснюється те, що стандарти сімейства IEEE 802 охоплюють лише два нижніх рівні моделі OSI – фізичний і каналний.

Сімейство стандартів IEEE 802 має досить чітку структуру, що показана на рис. 6.1. В окрему групу 802.1 винесені стандарти, що мають спільний для всіх технологій характер. Серед них важливе практичне значення мають стандарти міжмережевої взаємодії (internetworking) – на побудову більш складних мереж на основі базових мережевих технологій. Сюди входять наступні стандарти:

- 802.1d – описує логіку роботи прозорого моста.
- 802.1h – описуються стандарти базової роботи транслуючого моста (може без маршрутизатора об'єднувати Ethernet і FDDI, Ethernet і Token Ring і т.п.).

- 802.1Q – описує спосіб побудови віртуальних локальних мереж (VLAN).

В інших групах описуються стандарти базових технологій локальних мереж. В них специфіка локальних мереж знайшла своє відображення у розділенні канального рівня на два підрівні:

- LLC (Logical Link Control) – підрівень логічної передачі даних,
- MAC (Media Access Control) – підрівень управління доступом до середовища.

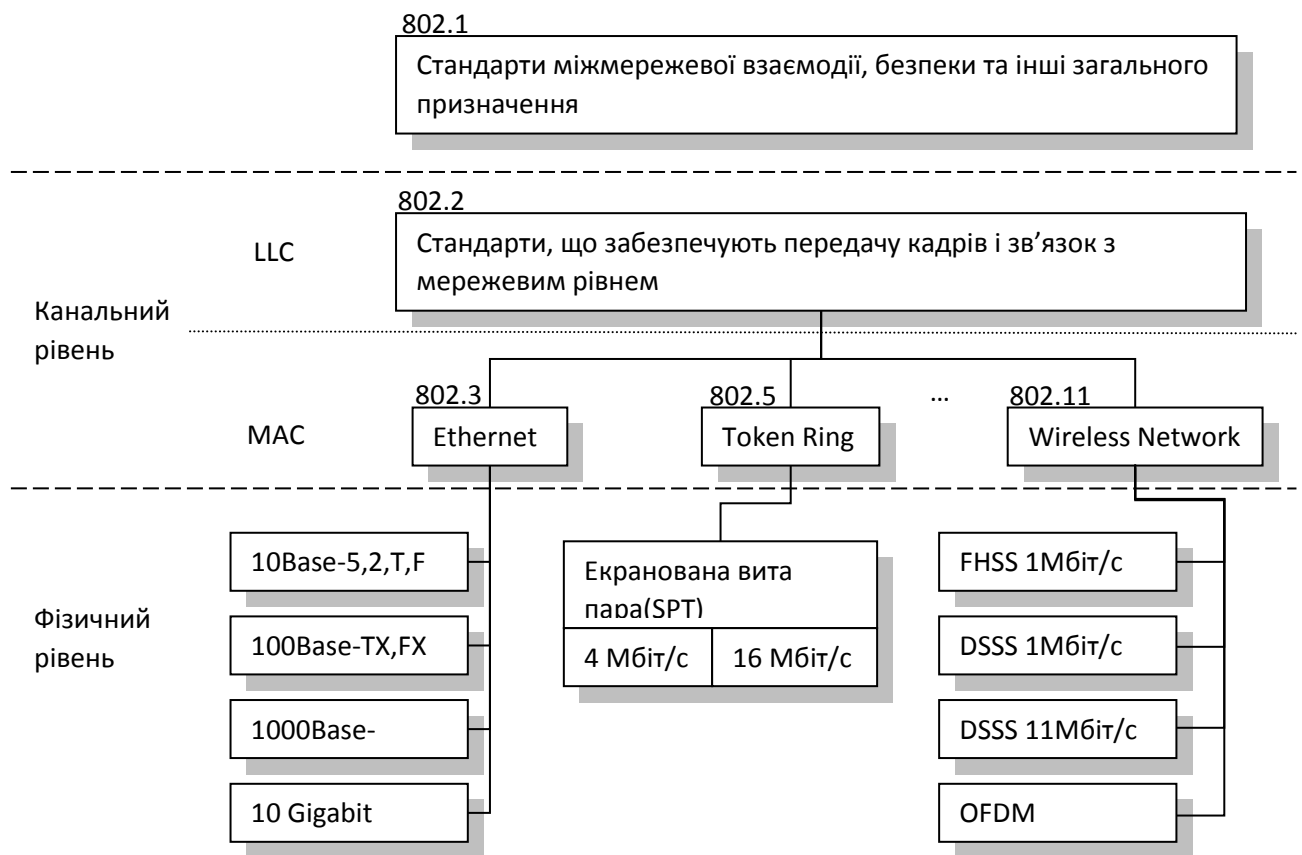


Рис. 6.1. Структура стандартів IEE 802

Функції підрівня LLC, як правило, реалізуються програмно: відповідним модулем операційної системи, а функції підрівня MAC реалізуються програмно-апаратно: мережевим адаптером і його драйвером.

Протоколи підрівнів LLC і MAC взаємно незалежні: кожний протокол

підрівня MAC може застосовуватись з будь-яким протоколом підрівня LLC, і навпаки.

Опис кожної технології поділений на дві частини: опис підрівня MAC і опис фізичного рівня. Практично у кожній технології єдиному протоколу підрівня MAC відповідають кілька протоколів фізичного рівня (це ж справедливо і для ArcNet, FDDI, 100VG-AnyLAN).

6.1.1. Підрівень LLC

Підрівень LLC виконує дві основні функції:

1. Надає інтерфейс до сусіднього мережевого рівня.
2. Забезпечує доставку кадрів з заданим ступенем надійності.

Інтерфейсними функціями LLC є:

- перетворення логічних адрес в фізичні і навпаки (наприклад, IP-адрес в MAC-адреси).
- мультиплексування і демультимплексування (передача даних від кількох протоколів мережевого рівня єдиному протоколу рівня MAC і навпаки, як показано на рис. 6.2):

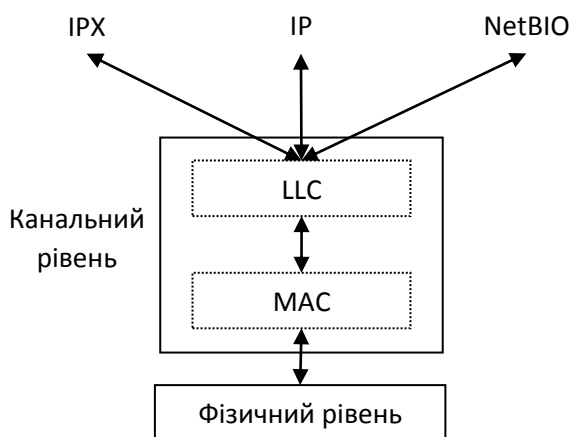


Рис. 6.2. Інтерфейсні функції підрівня LLC

Функція доставки кадрів з заданим ступенем надійності – наступна функція підрівня LLC. Нею передбачається підтримка кількох режимів

роботи, які відрізняються наявністю, або відсутністю, процедур відновлення кадрів у випадку їх пошкодження/втрати:

- LLC1 – підрежим без встановлення з'єднання і без підтвердження. Надає можливість здійснювати передачу даних з мінімумом витрат.
- LLC2 – підрежим з встановленням з'єднання і підтвердженням. Надає можливість встановити логічне з'єднання і виконувати процедури відновлення кадрів і їх упорядкування в рамках встановленого з'єднання.
- LLC3 – режим без встановлення з'єднання, проте з підтвердженням. Передбачений для випадків, коли часові витрати установки логічного з'єднання є недопустимими, а підтвердження про коректність прийому необхідне (наприклад, в системах реального часу, що керують промисловими об'єктами).

Використання одного з трьох підрежимів LLC залежить від стратегії розробників конкретного стеку протоколів. Найчастіше використовується підрежим LLC1, оскільки канали локальних мереж забезпечують високу якість передачі даних (без ушкодження і втрати кадрів). В цьому випадку, використання більш надійного режиму LLC2 приводить до невиправданої надлишковості, яка уповільнює загальну пропускну здатність стеку комунікаційних протоколів. Наприклад, в таких популярних стеках протоколів, як TCP/IP та IPX/SPX, підрівень LLC завжди працює в режимі LLC1.

Проте стек NetBIOS/SMB від Microsoft/IBM, використовує режим LLC2 – при функціонуванні NetBIOS/SMB в режимі з відновленням даних, і режим LLC1 - при функціонуванні NetBIOS/SMB в режимі передачі дейтаграм. Підрежим LLC2 використовується також стеком SNA в тому випадку, коли мейнфрейми або мінікомп'ютери IBM взаємодіють через мережі Token Ring. Підрежим LLC2 використовується в Hewlet-Packard у випадку, підключення принтерів до Ethernet безпосередньо через вбудовані

мережеві адаптери.

6.1.2. Підрівень MAC

Основними функціями підрівня MAC є:

1. Забезпечення доступу до поділюваного середовища передачі даних.
2. Передача кадрів між кінцевими вузлами.

Як показано на рис. 6.3, основними методами доступу до поділюваного середовища передачі даних є методи випадкового доступу. Вони ґрунтуються на тому, що вузол, який має кадр для передачі, пробує його відправити без будь-якої узгодженості з іншими вузлами. Такі методи не гарантують, що вузол отримає доступ до поділюваного середовища передачі даних протягом визначеного часу.

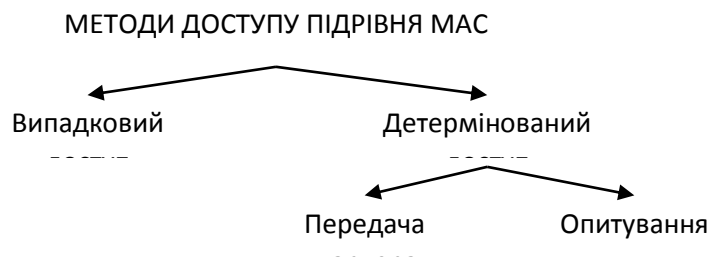


Рис. 6.2. Методи доступу до поділюваного середовища на підрівні MAC

Альтернативними до них є методи детермінованого доступу, в яких максимальний час очікування доступу до поділюваного середовища передачі даних є фіксованим.

Алгоритми детермінованого доступу використовують два механізми, що вказані нижче.

- Механізм передачі маркера – кожний вузол, який отримав маркер, має право на використання поділюваного середовища протягом фіксованого проміжку часу.
- Механізм опитування – спеціальний вузол (арбітр) періодично опитує інші вузли і, зібравши заявки, вирішує кому надати доступ до

поділюваного середовища.

Передача кадрів здійснюється підрівнем MAC в кілька етапів, які не залежать від використовуваного методу доступу:

- формування кадру. На цьому етапі здійснюється заповнення полів кадру на основі інформації, отриманої від протоколу вищого рівня – адреса відправника і одержувача, дані користувача, ознака протоколу верхнього рівня). Після формування кадру підрівень MAC обраховує контрольну суму і заносить її у відповідне поле.
- Передача кадру. Підрівень MAC передає кадр на фізичний рівень, який побітово передає всі поля кадру в середовище передачі даних.
- Прийом кадру. Підрівень MAC перевіряє адресу призначення кадру, що надійшов, і якщо співпадає з власною, обробляє його: перевіряє контрольну суму і відправляє верхньому рівню, або відкидає.

6.2. Сімейство стандартів IEEE 802

Сімейство IEEE 802 на даний час включає 24 основних стандарти:

802.1 – Internetworking. Визначає механізми управління маршрутизацією та мережевими пристроями.

802.2 – Logical Link Control. Визначає механізм передачі кадрів з заданою надійністю.

802.3 – Ethernet Carrier Sense Multiple Access with Collision Detection. Визначає метод доступу з прослуховуванням несучої і виявленням колізій, що використовується в Ethernet.

802.4 – Token Bus Lan. Визначає метод доступу до шини з використанням маркера. Прототипом була технологія ArcNet компанії Datapoint Corporation, виробництво апаратури для якої було згорнуто в 90-х р. Сам стандарт розформовано.

802.5 – Token Ring Lan. Визначає метод доступу до кільця з передачею маркера. Прототип – Token Ring. Стандарт не активний, не використовується.

802.6 – Metropolitan Area Network. Наводяться рекомендації з побудови регіональних мереж. Стандарт розформовано.

802.7 – Broadband Technical Advisory Group. Описує рекомендації з застосування широкополосних мережевих технологій (носії, інтерфейси, обладнання). Стандарт розформовано.

802.8 – Fiber Technical Advisory Group. Містить рекомендації з побудови оптоволоконних мереж. Прототип – FDDI. Стандарт розформовано.

802.9 – Integrated Voice and Data Network. Містить рекомендації з побудови гібридних мереж, в яких об'єднується голосовий і цифровий трафік. Стандарт розформовано.

802.10 – Network Security. Містить рекомендації з забезпечення безпеки обміну даними, управління мережами. Стандарт розформовано.

802.11 – Wireless Network. Описує рекомендації з використання безпроводних мереж.

802.12 – Demand Priority Access LAN. Описує рекомендації з використання мереж 100VG-AnyLAN. Стандарт розформовано.

802.14 – Стандарти роботи з кабельним модемом. Стандарт розформовано.

802.15 – Wireless PAN. Стандарти персональної мережі (Personal Area Networks), навколо людини. Зокрема Bluetooth, взаємодія Bluetooth і Wi-Fi, змішані мережі.

802.16 – Бездротова міська мережа. WiMAX-сертифікація, як альтернатива виділеним телефонним лініям і DSL

802.17 – Еластичне кільце пакетів. Ґрунтується на кільцевій топології і складається з вузлів пакетної комутації з'єднаних з сусідніми вузлами в кільце однією парою оптичного волокна.

802.18 – Радіорегулювання.

802.19 – Співіснування мереж.

802.20 – Мобільний широкомовний бездротовий доступ. Бездротова

технологія широкосмугової передачі також відома як iBurst (або HC-SDMA, High Capacity Spatial Division Multiple Access). Розроблялася компанією ArrayComm і оптимізувала використання своєї смуги пропускання частот за допомогою чутливих антен. Компанія Kyocera виробник пристроїв iBurst

802.21 – Media Independent Handoff. Стандарт підтримує алгоритми, що дозволяють безперебійну передачу даних між мережами одного типу, та передачу даних між різними типами мереж, – медіа-незалежну передачу даних, або вертикальну передачу даних.

802.22 – Місцеві бездротові мережі. Стандарт бездротових регіональних мереж, що описує дворівневу архітектуру (рівень РНУ і рівень MAC) з багатоточковим (point-to-multipoint) з'єднанням.

802.23 Робоча група надзвичайних сервісів Новий стандарт, 2010 р.

802.24 Smart Grid TAG – інтеграція влади, комунікації та інформаційних технології для покращення електроенергетичної інфраструктури, що обслуговує кінцевих користувачів. Новий стандарт, 2012 р.

Контрольні запитання до розділу

1. Стандарти роботи мережі.
2. Підрівень логічної передачі даних, його функції.
3. Підрівень управління доступом до середовища, методи доступу.
4. Діючі та розформовані стандарти.
5. Відмінність стандартів 802.11 та 802.3. Їх переваги та недоліки.

7. ПРОТОКОЛИ І СТЕКИ ПРОТОКОЛІВ

7.1. Стеки комунікаційних протоколів

Стек комунікаційних протоколів – це ієрархічно організована сукупність протоколів, що вирішують задачу взаємодії абонентських станцій.

Існує велика кількість стеків комунікаційних протоколів. Найбільш популярними з них є наступні, наведені в таблиці 7.1.

Табл.7.1. Стеки комунікаційних протоколів

Відкриті	TCP/IP
	IPX/SPX
	NetBIOS/SMB
	OSI
Фірмові	DECnet (Digital Equipment)
	AppleTalk/AFP (Apple)
	SNA (IBM)

Ряд стеків носять відкритий характер і підтримуються багатьма сторонніми розробниками програмного і апаратного забезпечення мереж. Інші ж через їх закритість використовуються майже виключно в продуктах окремих виробників.

Проте всі ці стеки, за виключенням SNA, на нижніх рівнях (фізичному і канальному) використовують добре стандартизовані протоколи базових технологій (Ethernet, Token Ring, FDDI), що дозволяє використовувати одну й ту ж апаратуру при застосуванні різних стеків протоколів.

Слід зазначити, що рівні багатьох стеків не відповідають рекомендованій моделлю OSI. Зокрема, функції сеансового рівня та рівня представлення часто об'єднуються з прикладним рівнем. Причиною такої невідповідності є те, що модель OSI стала результатом узагальнення роботи

реально використовуваних стеків протоколів, а не навпаки.

7.2. Стек протоколів OSI

Стек протоколів OSI підтримується урядом США: за його програмою (GOSIP) всі мережі в державних установах після 1990 року повинні або безпосередньо підтримувати стек протоколів OSI, або забезпечувати можливість переходу на цей стек в майбутньому. Більшість організацій поки тільки планують перехід на стек протоколів OSI.

Про підтримку стеку протоколів OSI заявили ряд провідних виробників апаратного і програмного забезпечення, включаючи DEC, Hewlett-Packard, IBM, SUN. Одним з найкрупніших виробників, що підтримує стек протоколів OSI, є компанія AT&T – її мережа Stargroup повністю базується на ньому.

Стек протоколів OSI відрізняється своєю складністю і вимагає великої обчислювальної потужності.

На відміну від інших стеків протоколів, стек OSI повністю відповідає моделі OSI, як показано в таблиці 7.2.

Табл.7.2. Стек комунікаційних протоколів моделі OSI

Рівні моделі OSI	Протоколи стеку OSI				
7	X.400	X.500	FTAM	VT	JTM
6		Протокол представлення OSI			
5	Сеансовий протокол OSI				
4	Транспортний протокол OSI (класи 0-4)				
3	Мережеві протоколи з встановленням з'єднання (ES-IS) і без нього (IS-IS)				
2	Ethernet	Token Ring	FDDI	ISDN	X.25
1					HDLC LAP-B

На фізичному рівні стеків протоколів OSI підтримує Ethernet, Token Ring, FDDI. На мережевому рівні підтримуються протоколи з встановленням з'єднання і без встановлення з'єднання. На транспортному рівні визначені п'ять класів транспортного сервісу (від найнижчого 0 до найвищого 4), які відрізняються ступенем стійкості до помилок.

Протоколи стеку OSI мережевого, транспортного і сеансового рівнів не набули широкого розповсюдження. Проте протоколи прикладного рівня можуть працювати над транспортом TCP/IP (див. RFC 1006), чим пояснюється їх більша розповсюдженість.

З найбільш перспективних протоколів прикладного рівня стеку OSI є протоколи:

- X.400 – електронної пошти; призначений для побудови глобальної системи обміну повідомленнями.
- X.500 – служби розподілених каталогів; призначений для побудови глобальної довідкової служби.
- VT – віртуального терміналу; призначений для забезпечення сумісності протоколів емуляції терміналів.
- FTAM – передачі, доступу і управління файлами; містить засоби маніпулювання файлами і доступу до вмісту файлів.
- JTM – пересилки і управління роботами; призначений для підтримки пакетної обробки даних.

7.2.1. Електронна пошта X.400

З множини систем електронної пошти лише два типи завоювали всесвітнє визнання – пошта Інтернет і пошта X.400. На відміну від пошти Інтернет пошта X.400 з самого початку розроблялася з комерційною метою, а тому виключає можливість втрати або перекручення інформації.

Системи X.400 призначені, перш за все, для захищеного і надійного документообігу. В них гарантується не лише сама доставка, але і строк доставки, можливе отримання підтверджень про доставку, або прочитання

повідомлень. Тому вони використовуються там, де є вимоги до достовірності, надійності і захищеності інформації – в державних, військових, банківських структурах.

Про підтримку даного стандарту оголосили ряд виробників апаратного і програмного забезпечення – IBM, AT&T, Sun та інші. Компанією Nexor в США розроблений стандарт для військових систем обробки повідомлень ACP-123, який ґрунтується на рекомендаціях X.400.

В якості прикладу можна навести також мережу R400. Це – багатофункціональна поштова система, яка дозволяє обмінюватись повідомленнями не тільки в рамках поштової мережі X.400, а також по факсу, телексу. В рамках R400 реалізовані шлюзи з поштовими мережами Novell MHS, REMART і RELCOM.

Система обробки повідомлень (СОП) на основі X.400, що показана на рис. 7.1, складається з компонентів, наведених нижче.

- Агент користувача (АК) – прикладний процес, що забезпечує користувачу інтерфейс з системою управління повідомленнями.
- Агент передачі повідомлень (АПП) – прикладний процес, що перенаправляє повідомлення іншим АПП або АК.
- Сховище повідомлень (СП) – для збереження доставлених повідомлень.
- Система передачі повідомлень (СПП) – забезпечує транспортування повідомлень від АК відправника до АК одержувача.
- Модуль доступу фізичної доставки (МДФД) – забезпечує зв'язок з системою фізичної доставки, тобто системою, яка доставляє фізичні повідомлення (поштова служба).
- Модуль доступу (МД) – забезпечує зв'язок з іншими системами обміну даними (наприклад, телекс, інші системи електронної пошти).

В силу архітектурних особливостей X.400 для гарантованої установки з'єднання між двома АПП потрібна ручна настройка значного числа параметрів. Тому динамічна маршрутизація в системах X.400 не можлива.

Проте у випадку використання каталогу організації (X.500) інформація про маршрути може задаватись автоматично.

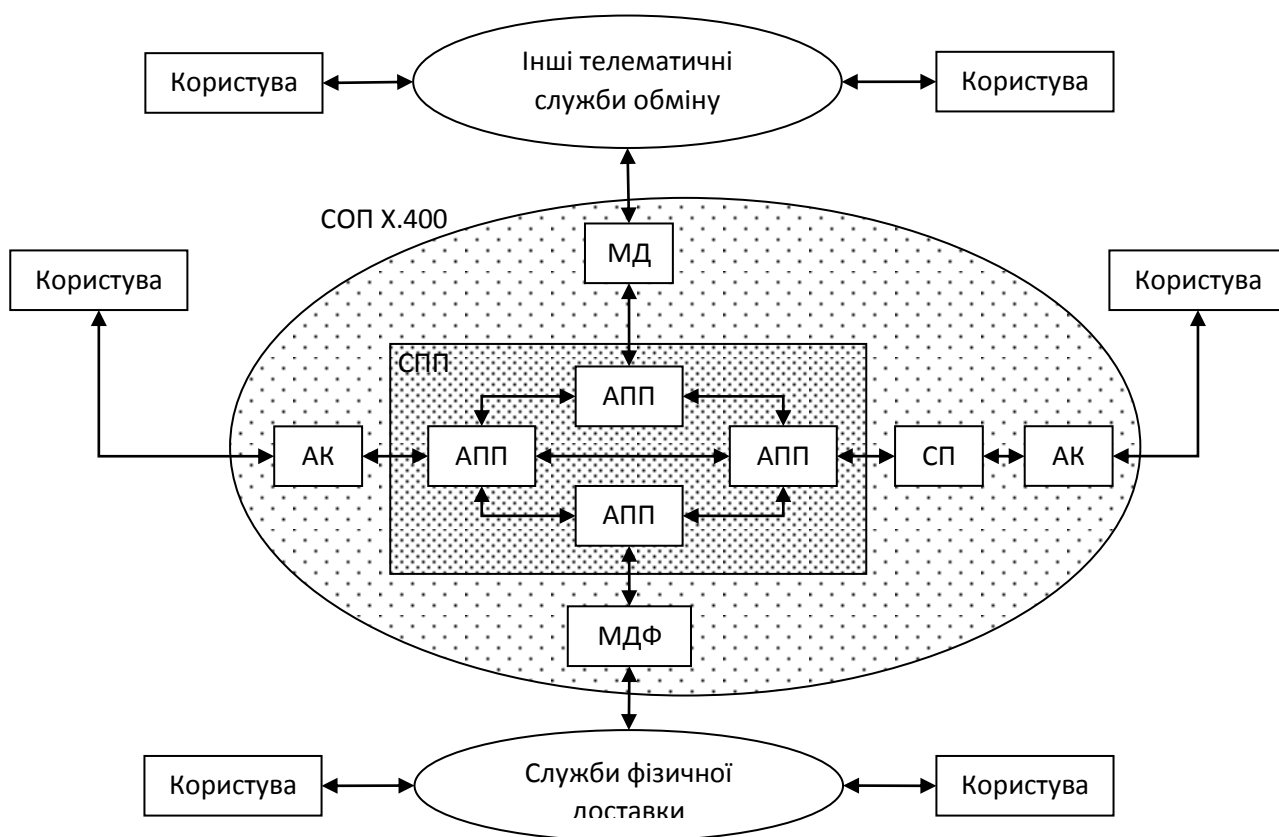


Рис. 7.1. Система обробки повідомлень

Для опису формату повідомлення в рекомендаціях X.400 прийнята звична парадигма конверта (envelope) і вмісту (content) традиційних поштових систем. Конверт містить вичерпну інформацію про те, куди і кому треба доставити повідомлення, обернену адресу і помітку про терміновість доставки. На основі цієї інформації виконується маршрутизація повідомлення і його передача з можливим проміжним збереженням (store and forward). Конверт може мати спеціальну помітку про необхідність установки на ньому електронного «штампу» (trace information) кожним АПП, через який проходить повідомлення.

Однак, широкого розповсюдження за межами державних, військових і банківських установ протоколи X.400 не набули через складність реалізації,

високу вартість впровадження та експлуатації, відсутність відкритого доступу до стандартів.

7.2.2. Служба розподілених каталогів X.500

X.500 – серія стандартів для служби розподіленого каталогу мережі. Спочатку протоколи X.500 планувалось використовувати для збереження довідкової інформації, необхідної для передачі електронної пошти у відповідності з вимогами протоколів X.400. Зараз каталоги X.500 можуть надавати централізовану інформацію про всі поіменовані об'єкти мережі – ресурси, додатки і користувачів. Так, каталог X.500 може містити телефонні номери, електронні адреси і іншу інформацію, що стосується, наприклад, ідентифікації і авторизації користувачів. Ці дані можуть надаватись як іншим додаткам, так і кінцевим користувачам.

Як стандарт розподіленого каталогу X.500 визначає стратегію розділення даних, їх реплікацію, кешування, синхронізацію і поширення.

На простому рівні дані просто розділяються між кількома серверами.

Наступний рівень – реплікація часто запитуваної інформації. Вона будується на механізмі головного/підлеглого (master/slave) серверів, коли один сервер займається тим, що розсилає репліки своєї інформації підлеглим серверам.

Кешування – інший спосіб реплікації, при якому сервер зберігає у себе копію найбільш часто запитувану інформацію, яка офіційно зберігається на іншому сервері. Кешування легко організувати, але це не гарантує актуальності отриманої з кешу інформації.

Синхронізація частково знімає ці проблеми, використовуючи оновлення інформації за розкладом, що дозволяє забезпечити компроміс між актуальністю даних і мережевим трафіком.

Розподілення – найпростіший спосіб синхронізації, при якому інформація рухається в одну сторону і приймаючий сервер не запитується про необхідність оновлення.

Інформація розподіленого каталогу в X.500 фізично розташовується на різних серверах, які називаються DSAs (Directory System Agents) – системні агенти каталогу. Клієнтів каталогу називають DUAs (Directory User Agents) – агенти користувача каталогу.

Для виконання пошуку інформації в каталозі можуть використовуватись три операції:

- зчеплення (chaining),
- відсилка (referral),
- багатоадресна розсилка (multicasting).

При виконанні зчеплення DSA намагається знайти інформацію у себе, але має можливість передати запит іншому DSA для отримання більш повної інформації. Коли інформація від усіх DSA буде тримана першим DSA, вона передається ним користувачу.

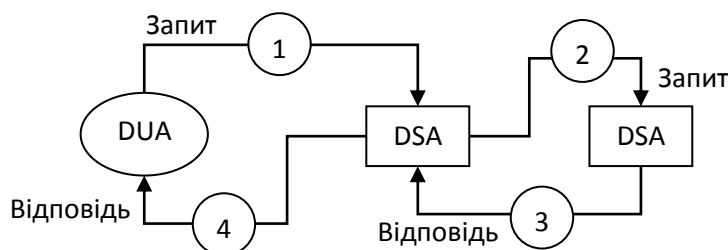


Рис. 7.2. Пошук інформації в каталозі X.500 на основі зчеплення

Відсилкою називається операція, при якій DUA керує запитом до DSA: замість передачі запиту іншому DSA повертається «порада», до кого DUA має звернутись за відповіддю

Багатоадресна розсилка нагадує зчеплення з тією різницею, що запит DUA передається першим DSA зразу кільком іншим, а результат видається користувачу одночасно всіма DSA.

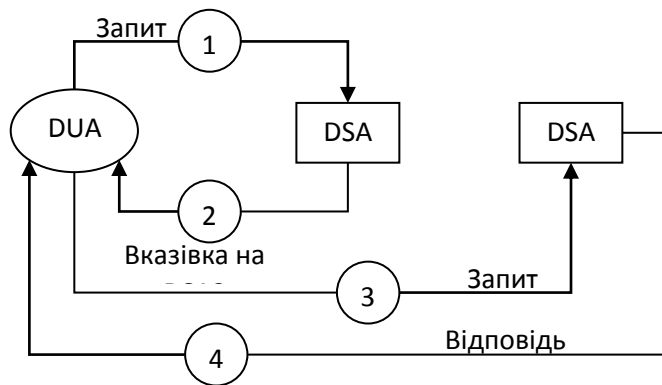


Рис. 7.3. Пошук інформації в каталозі X.500 на основі відсилки

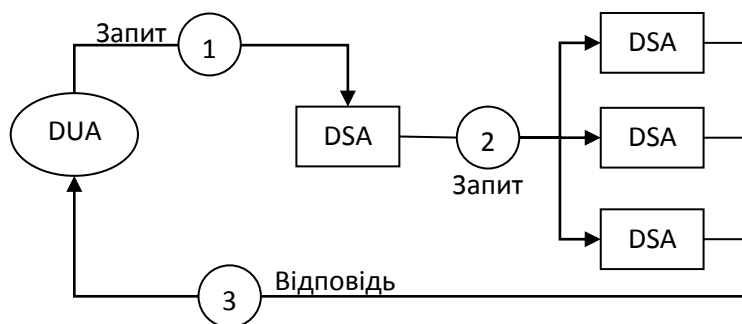


Рис. 7.4. Пошук інформації в каталозі X.500 на основі розсилки

Операція для виконання пошуку обирається користувачем в залежності від того, хоче він перекласти обробку на сервер (зчеплення, розсилка), чи на клієнта (відсилка). Для публічних операторів значно вигідніше використання зчеплення, оскільки при цьому у користувача існує одна точка входу, що полегшує виставлення рахунку.

Взаємодія DUA і DSA здійснюється за протоколом DAP (Directory Access Protocol), а взаємодія між DSA під час обробки клієнтського запиту – по протоколу DSP (Directory System Protocol).

Через складність DAP було запропоновано кілька його модифікацій, які забезпечують більш простий і швидкий доступ до каталогу. Однією з таких модифікацій став протокол LDAP (Lightweight DAP) – «полегшений»

протокол доступу до каталогів, який використовує стек TCP/IP і дозволяє виконувати операції аутентифікації, пошуку, порівняння, додавання, редагування або видалення даних. Підтримка LDAP вбудована в багато сучасних програмних продуктів, наприклад, Microsoft Outlook Express, Internet Explorer (v.4 і вище), Microsoft Exchange.

7.3. Стек протоколів TCP/IP

Стек комунікаційних протоколів TCP/IP (називають також стеком Інтернет або стеком DoD.), розроблений з ініціативи Міністерства оборони США (Department of Defense, DoD), є на сьогодні одним з найбільш популярних.

Значний внесок в розвиток стеку зроблено університетом Берклі, що реалізував протоколи TCP/IP в своїй операційній системі UNIX. Широке розповсюдження UNIX, пов'язане з становленням і розвитком всесвітньої інформаційної мережі Інтернет, стало причиною популярності стеку протоколів TCP/IP.

Структура стеку протоколів TCP/IP показана в таблиці 7.3. Найнижчий рівень стеку TCP/IP – рівень мережевих інтерфейсів – відповідає фізичному і канальному рівням моделі OSI. Він підтримує всі базові мережеві технології – Ethernet, Token Ring, FDDI, ISDN, x.25. Розроблена також спеціальна специфікація, що визначає використання технології АТМ на цьому рівні.

Наступний рівень стеку TCP/IP – рівень міжмережевої взаємодії – відповідає мережевому рівню моделі OSI. В якості основного протоколу цього рівня використовується протокол IP (Internet Protocol), який є дейтаграмним. До цього ж рівня відносяться і протоколи, що пов'язані з обробкою таблиць маршрутизації – RIP (Routing Internet Protocol) і OSFP (Open Shortest Path First), а також протокол ICMP (Internet Control Message Protocol) для обміну інформацією про помилки між відправником і одержувачем даних.

Табл.7.3. Стек комунікаційних протоколів моделі TCP/IP

Рівні моделі OSI	Протоколи стеку TCP/IP				Рівні моделі TCP/IP	
7	HTTP	FTP	SMTP	Telnet	I	Прикладний
6						
5	TCP	UDP	II	Транспортний		
4						
3	IP	RIP	OSPF	ICMP	III	Міжмережевої взаємодії
2	Ethernet	Token Ring	FDDI	ISDN	IV	Мережевих інтерфейсів
1						

Вищий рівень стеку TCP/IP – транспортний рівень – відповідає транспортному і сеансовому рівням моделі OSI. На цьому рівні функціонують два протоколи:

- TCP (Transmission Control Protocol) – забезпечує надійний зв’язок з встановленням з’єднання,
- UDP (User Datagram Protocol) – забезпечує ненадійний зв’язок без встановлення з’єднання.

Найвищий рівень стеку TCP/IP – прикладний рівень – відповідає рівням представлення і прикладному рівням моделі OSI. На цьому рівні працюють такі широко застосовувані протоколи, як:

- HTTP (HyperText Transfer Protocol) – забезпечує передачу гіпертекстових документів.
- FTP (File Transfer Protocol) – реалізує віддалений доступ до файлів.
- TFTP (Trivial FTP) – реалізує тільки передачу файлу, використовуючи в якості транспорту протокол UDP.
- SMTP (Simple Mail Transfer Protocol) – реалізує пересилку електронної пошти.

- POP (Post Office Protocol) – забезпечує доставку електронної пошти з поштового відділення.
- Telnet – забезпечує передачу потоків байтів між прикладним процесом і терміналом.

7.4. Стек протоколів IPX/SPX

Стек комунікаційних протоколів IPX/SPX розроблено компанією Novell для її пропрієтарної мережевої операційної системи NetWare. В 1980-90-х р. мережі, на основі цього стеку, були широко розповсюдженими через популярність NetWare. Особливості стеку протоколів IPX/SPX обумовлені орієнтацією на роботу в локальних мережах невеликих розмірів, що складаються з малопотужних персональних комп'ютерів.

Компанія Novell і в наш час продовжує працювати над розвитком стеку IPX/SPX, але він все більше поступається за популярністю стеку протоколів TCP/IP. Структура стеку протоколів IPX/SPX показана в таблиці 7.4.

Табл. 7.4. Стек комунікаційних протоколів моделі OSI

Рівні моделі OSI	Протоколи стеку IPX/SPX		
7	SAP	NCP	
6			
5			
4	SPX		
3	IPX	RIP	NLSP
2	Ethernet	Token Ring	FDDI
1			

На фізичному і канальному рівнях в мережах Novell використовуються базові мережеві технології – Ethernet, Token Ring, FDDI та інші.

На мережевому рівні використовується протокол IPX (Internetwork Packet Exchange), який підтримує тільки дейтаграмний спосіб обміну повідомленнями. Сервери NetWare використовують на мережевому рівні протоколи маршрутизації RIP (Routing Information Protocol), або NLSP (Netware Link Services Protocol).

Оскільки IPX є протоколом без встановлення з'єднання, для підтвердження правильності доставки даних він покладається на протоколи верхніх рівнів. Наприклад, якщо клієнт надсилає запит до сервера, використовуючи протокол NCP, то відповідь сервера є підтвердженням отримання запиту.

Для транспортного передбачений свій протокол – SPX (Sequenced Packet Protocol), який надає надійний зв'язок з встановленням з'єднання, управлінням потоком і визначення порядку слідування пакетів. Проте NetWare використовує його не часто, головним чином при зверненні до серверів друку і виконання резервного копіювання. В більшості типових процедур доступу до файлів томів NetWare застосовує протокол NCP, яким генерується основна частина трафіку. Завдяки такому підходу забезпечується висока продуктивність мереж Novell.

На верхніх рівнях (сеансовому, представлення і прикладному) працюють протоколи NCP і SAP.

Протокол NCP (Netware Core Protocol) використовується клієнтськими системами NetWare для запитів файлів, розташованих на томах серверів NetWare, і відправки завдань для друку. Сервери NetWare застосовують NCP для передачі запитаних файлів назад клієнтам.

Протокол SAP (Service Advertising Protocol) використовується серверами NetWare і маршрутизаторами для оголошень про свої сервісні послуги. Клієнт NetWare перед тим, як звернутись до сервера NetWare, повинен дізнатись про його існування з повідомлень SAP. Ці повідомлення містять ім'я сервера, його адресу і опис послуги, що надається. Інші системи

при отриманні повідомлення SAP створюють для кожного сервера тимчасовий запис у своїй базі даних ресурсів мережі (NDS).

Повідомлення SAP розсилаються серверами NetWare кожні 60 секунд, що дозволяє мережевим пристроям постійно корегувати дані про те, які сервісні послуги існують на даний час в мережі. Проте вони завантажують мережу, тому однією з основних задач маршрутизаторів стає фільтрація трафіку SAP-пакетів.

Контрольні запитання до розділу

1. Поняття стеку комунікаційних протоколів, відкриті за закриті стеки.
2. Стек протоколів OSI, протоколи стеку.
3. Електронна пошта, її компоненти.
4. Служба каталогів X.500, пошук інформації на основі розсилки.
5. Стек протоколів TCP/IP та IPX/SPX

8. АРХІТЕКТУРА СТЕКА ПРОТОКОЛІВ MICROSOFT TCP/IP

8.1. Стандарти по TCP/IP

Стек комунікаційних протоколів TCP/IP був розроблений в 1969 році Агенцією перспективних дослідницьких проектів (Advanced Research Project Agency – ARPA) Міністерства оборони (Department of Deffence - DoD) США для експериментальної мережі, відомої як ARPANET (ARPA Network). Мета розробки полягала в забезпеченні високошвидкісних комунікаційних з'єднань між окремими комп'ютерними мережами.

Міністерства оборони США забезпечило доступність стеку TCP/IP, реалізувавши його протоколи в операційній системі UNIX. Розповсюдженість протоколів TCP/IP забезпечила можливість об'єднання з ARPANET мереж університетів, дослідницьких організацій та іншими, що в підсумку привело до виникнення глобальної інформаційної системи.

Тривалий час координаційну роль в Інтернет виконувало Міністерство оборони США. Лише в 1993 р. обслуговування цивільних користувачів Інтернет було передане в Національний науковий фонд (National Science Foundation – NSF) США і на сьогодні виконується двома агенціями:

- InterNIC Registration Services – служба реєстрації, яка координує іменування і адресацію комп'ютерів в Інтернет (належить компанії Network Solution Inc).
- InterNIC Directory Database Services – служба каталогів і баз даних, яка служить депозитарієм стандартів Інтернет і інших інформаційних документів (належить компанії AT&T).

Розробка нових стандартів Інтернет координується Радою з архітектури Інтернет (Internet Architecture Board – IAB). В 1992 р. була сформована Асоціація Інтернет (Internet Society – IS), до якої увійшла IAB, що показано на рис. 8.1.

IAB контролює роботу двох груп:

- робочої групи технологій Інтернет (Internet Engineering Task Force, IETF), яка розробляє нові протоколи.
- управляючої групи технологій Інтернет (Internet Engineering Steering Group, IESG), яка здійснює керівництво і контроль за діяльністю IETF.

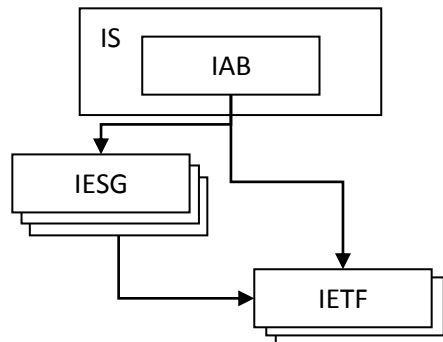


Рис. 8.1. Структура Асоціації Інтернет

Стандарти по TCP/IP публікуються у вигляді документів RFC (Request for Comments). При публікації RFC-документу привласнюється унікальний номер. Зміст вихідного RFC-документу ніколи не оновлюється. Якщо виникає необхідність у змінах, публікується новий RFC-документ під іншим номером. Всі документи доступні для безкоштовного завантаження з депозитаріїв в Інтернет.

8.2. Архітектура Microsoft TCP/IP

Компанія Microsoft прийняла протоколи TCP/IP в якості стратегічного транспорту для побудови корпоративних мереж на платформі Windows і спрощення їх інтеграції з іншими великими корпоративними, державними і загальнодоступними мережами. В зв'язку з цим на основі існуючих RFC-документів компанія розробила власну програмну реалізацію цих протоколів, яка отримала назву Microsoft TCP/IP.

Microsoft TCP/IP – це високопродуктивна реалізація стандартного

промислового набору протоколів TCP/IP, що використовуються в операційних системах Windows (починаючи з Windows 95).

Саме завдяки Microsoft TCP/IP операційна система Windows забезпечує роботу в Інтернет. Мережева архітектура Windows показана на рис. 8.2.

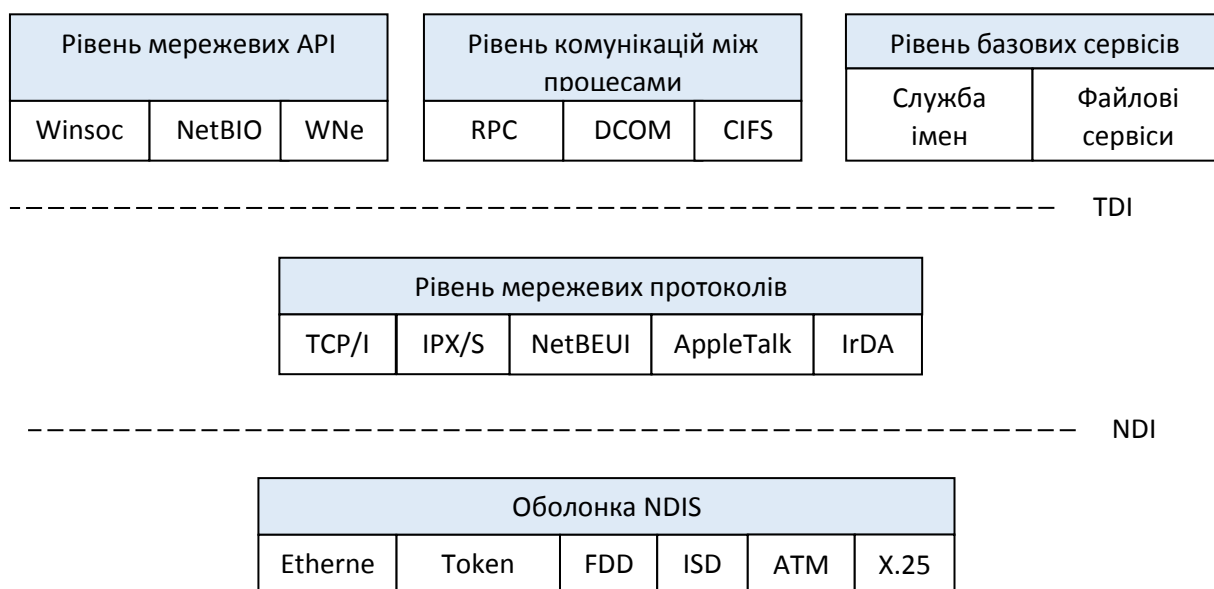


Рис. 8.2. Мережева архітектура Windows

Мережескі компоненти Windows розподілені по кількох рівнях, кожний з яких відповідає за виконання специфічних задач.

Мережескі протоколи Microsoft взаємодіють з драйверами мережеских плат за специфікацією NDIS (Network Driver Interface Specification). NDI діє як посередник між мережескими адаптерами і мережескими протоколами. NDI забезпечує підтримку як традиційних мережеских середовищ, які не вимагають логічних з'єднань – Ethernet, Token Ring, FDDI, так і таких, які вимагають логічних з'єднань – ISDN, ATM.

Рівень мережеских протоколів включає TCP/IP, IPX/SPX, NetBEUI, AppleTalk, IrDA, NWLink, ATM і DLC. Крім того, при установці Microsoft SNA Server стають доступними протоколи SNA.

Стандартний інтерфейс для взаємодії мережеских протоколів з їх

клієнтами визначається специфікацією TDI (Transport Driver Interface). В якості клієнтів мережесих протоколів можуть виступати мережесі API, мережесі редіректори, прикладні додатки.

Специфікація TDI, як і специфікація NDI, є відкритою і тому їх можна отримати в Microsoft.

Рівень мережесих API надає стандартні API для мережесих додатків і служб: Winsock, NetBIOS, TAPI (Telephony API), MAPI (Messaging API), WNet API та інші.

Рівень комунікацій між процесами підтримує роботу в шаблоні архітектури «клієнт-сервер» і розподілену обробку даних. Він включає такі сервіси, як RPC(Remote Procedure Call), DCOM (Distributed Component Object Model), іменовані канали (named pipes), поштові ящики (mailslots) і CIFS (Common Internet File System).

Рівень базових мережесих сервісів надає базові мережесі сервіси мережесим додаткам, до яких відносяться управління мережесими адресами, служба імен, файлові сервіси та інші.

8.3. Специфікація NDI

NDIS – це специфікація архітектури мережесих драйверів, що дозволяє мережесим протоколам взаємодіяти з мережесим адаптером, або з іншими апаратними приладами. Драйвери мережесих адаптерів для роботи в Windows мають бути написані у відповідності до цієї специфікації.

В Windows NDI реалізований в файлі Ndis.sys, який також називається оболонкою NDIS.

Оболонка NDIS – це код, що «покриває» всі NDIS-сумісні драйвери пристроїв.

Оболонка NDIS представляє собою набір підпрограм, що спрощують розробку NDIS-драйверів.

В реальності протоколи взаємодіють не з мережесими адаптерами, а з

оболонкою NDIS. Мережеві адаптери стають зовсім незалежними від мережевих протоколів.

В NDIS 5.0 (входить до Windows 2000) включена нова функціональність, яка забезпечує підтримку наступним технологіям:

- Wake-On-LAN – технологія управління електроспоживанням. NDIS передбачає зниження рівня електроспоживання мережевих плат за запитами системи. Такий запит може бути заданим користувачем (при переході в «сплячий» режим), або самою системою (при тривалому простої). На сьогодні Microsoft TCP/IP – єдиний стек протоколів, який підтримує управління електроживленням в мережі.
- Media Sense – технологія, що дозволяє мережевому адаптеру повідомляти, чи підключений він фізично до мережі. Протоколи і додатки здатні розпізнати і діяти відповідно – наприклад, відобразити значок про стан підключення.
- Network Plug and Play – програмно-апаратна підтримка, що дозволяє комп'ютерній системі розпізнати зміни в апаратній конфігурації і адаптуватись під них за мінімальної участі користувача, або без нього.
- TCP/IP Task Offload – апаратне прискорення виконання деяких операцій, пов'язаних з TCP/IP (наприклад, підрахунок контрольних сум для TCP і UDP). Це зменшує навантаження на центральний процесор і дозволяє йому ефективніше виконувати інші задачі, що може збільшити пропускну здатність з'єднання з мережею. Транспортний драйвер видає спеціальний запит щоб з'ясувати, чи реалізована в мережевому адаптері відповідна апаратна підтримка, і якщо це так – транспортний драйвер може вимагати від мережевого адаптера надання відповідних сервісів.

8.4. Мережеві протоколи TCP/IP

В Microsoft TCP/IP внесені ряд удосконалень:

1. Підтримка великих вікон TCP. Розмір вікна визначає максимальне число

пакетів, які може відсилати відправник без отримання підтверджень про доставку. Великий розмір вікна збільшує продуктивність протоколу при передачі великих обсягів даних.

2. Вибіркове підтвердження (selective acknowledgments). Дозволяє одержувачу підтверджувати і запитувати від відправника тільки ті пакети, які втрачені/пошкоджені в ході доставки. В попередніх реалізаціях TCP якщо до одержувача не надходив хоча б один пакет, відправнику доводилось повторно передавати не тільки його, а й усі наступні.
3. Оцінка RRT (Round Trip Time) – час повної передачі групи пакетів з підтвердженнями. Збільшення точності оцінки RRT дозволяє оптимізувати час очікування перед повторною передачею.
4. IP-безпека. З метою забезпечення криптографічного захисту контролю за доступом, аутентифікації джерела даних, конфіденційності трафіку і т.п. можливе використання спеціального протоколу IPSec. Оскільки IPSec працює на IP-рівні, його сервіси доступні протоколам більш високих рівнів, а значить, і відповідним додаткам. IPSec може захистити трафік між хостами, між шлюзами або між хостами і шлюзами. Необхідні для цього служби настраюються через вибрану політику IP-безпеки – локальної або групової з використанням Active Directory. Слід зауважити, що використання IPSec призводить до зменшення продуктивності мережі.
5. Генерація якості обслуговування (Generic Quality of Service - GQoS). Забезпечує розгортання додатків, що пересилають дані по IP-мережам в режимі реального часу, забезпечуючи прийнятний рівень затримки, полосу пропускання і нерівномірність передачі. Можна сказати, що GQoS надає можливість використовувати переваги АТМ в TCP/IP-середовищах.
- 6.

8.5. Діагностичні утиліти в Microsoft TCP/IP

До складу Microsoft TCP/IP входять ряд утиліт для виявлення і усунення проблем в TCP/IP-мережах, що показані в Таблиці 8.1.

Таблиця 8.1. Утиліти Microsoft TCP/IP

Утиліта	Опис
Hostname	Виводить хост-ім'я локального комп'ютера (не передбачає будь-яких параметрів і ключів)
Ipconfig	Виводить звіт про конфігурацію IP-інтерфейса.
Ping	Перевіряє доступність віддаленої системи. Утиліта відправляє на IP-адресу одержувача відлуння-запит ICMP. Якщо хост з вказаною адресою відповідає, можна спробувати використати замість IP-адреси хост-ім'я. Утиліта спочатку намагається перетворити хост-ім'я в IP-адресу через DNS-сервер, потім через WINS-сервер і нарешті за допомогою локального широко віщання. Якщо перевірка за адресою закінчилась успішно, а за ім'ям – ні, це свідчить про проблему не в мережевому з'єднанні, а в розпізнаванні імен. Використання ключа –а дозволяє перетворювати IP-адреси в хост-імена.
Tracert	Дозволяє прослідкувати маршрут до віддаленої системи. Утиліта відправляє відлуння-запити ICMP, послідовно збільшуючи значення поля TTL на 1, як показано на рис. 8.3. Обробка запиту маршрутизатором полягає у зменшенні значення поля TTL на 1 і відправці спеціального повідомлення Time Exceeded у випадку коли TTL=0. Після кожної успішної доставки утиліта відправляє пакет на один перехід далі.
Pathping	Дозволяє прослідкувати маршрут до віддаленої системи і оцінити втрати пакетів на кожному маршрутизаторі. Утиліта суміщає в собі функціональність утиліт Ping і Tracert та надає додаткову інформацію про ступінь перевантаженості каналу.
Netstat	Відображає статистику по протоколам і TCP/IP -з'єднанням. Найчастіше використовується з такими ключами: -а – для виведення інформації по всім з'єднанням; -е – для виведення статистики по інтерфейсу; -s – для виведення статистики по TCP, IP, ICMP і UDP для локального хосту.
Arp	Для перегляду кеша ARP і виявлення в ньому некоректних записів. Якщо двом хостам не вдається з'єднатись по команді Ping, слід виконати на кожному з них команду Arp –а, щоб перевірити правильність MAC-адрес в кешах ARP. З'ясувати коректну MAC-адресу можна за допомогою Ipconfig.

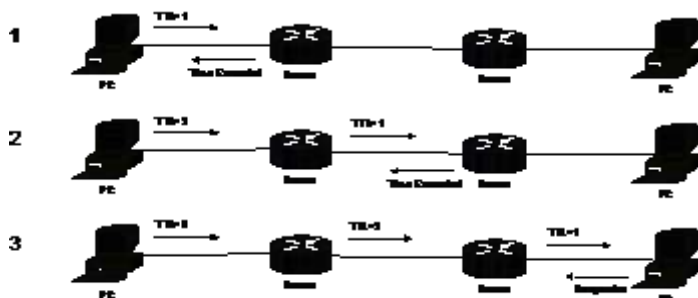


Рис. 8.3. Приклад роботи утиліти Tracert

Контрольні запитання до розділу

1. Стандарти стеку протоколів TCP/IP, архітектура TCP/IP.
2. Специфікація архітектури мережесих драйверів для Windows.
3. Мережесі протоколи TCP/IP.
4. Утилїти TCP/IP, їїх призначення.
5. Відмінність між утилїтами tracert та arp, між ping та pathping.

ЧАСТИНА II. ЛОГІЧНА ОРГАНІЗАЦІЯ КОМП'ЮТЕРНИХ МЕРЕЖ

9. АДРЕСАЦІЯ В IP-МЕРЕЖАХ

9.1. Адресний простір і види адрес

При об'єднанні в мережу трьох або більше вузлів виникає необхідність в ідентифікації кожного вузла, точніше – їх мережевих інтерфейсів, на основі певної схеми адресації.

Адреси можуть використовуватись для ідентифікації не тільки окремих інтерфейсів, але й їх груп. За допомогою таких групових адрес дані можуть адресуватись відразу кільком вузлам. Окремим випадком групових адрес є широкомовні адреси – направлені за ними дані доставляються всім вузлам мережі.

Існують різні схеми адресації.

Множина всіх адрес, які є доступними в даній схемі адресації, називається адресним простором (address space).

Адресний простір може мати плоску (лінійну) або ієрархічну організацію.

У випадку плоского адресного простору, як показано на рис. 9.1, множина адрес є не структурованою. Прикладом лінійно організованого простору адрес є множина телефонних номерів, наданих абонентам оператором телефонного зв'язку.

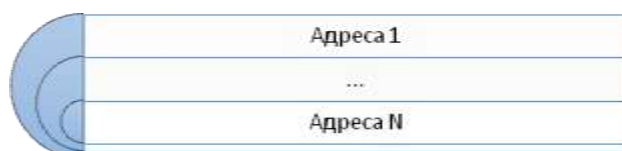


Рис. 9.1. Плоский адресний простір

У випадку ієрархічної організації простору він складається з вкладених одна до одної груп. На рис. 9.2 показано приклад адресного простору дворівневої ієрархії.

Ієрархічна організація адресного простору є більш раціональною у порівнянні з лінійною: вона дозволяє на початку передачі даних користуватись лише старшою складовою адреси, далі – іншими складовими, а в кінці – молодшою складовою адреси.

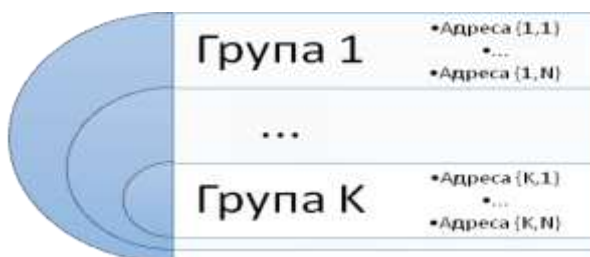


Рис. 9.2. Приклад адресного простору дворівневої ієрархії.

Типовим прикладом ієрархічно організованих адрес є звичайна поштова адреса, в якій послідовно уточняється місцезнаходження адресата.

Мережевий інтерфейс може мати одночасно кілька різних адрес в залежності від використовуваних схем адресації. Для перетворення адрес з одного виду до іншого використовуються спеціальні допоміжні протоколи, які називаються протоколами розрізнення адрес.

Розрізнення адрес може здійснюватися як централізованими, так і розподіленими засобами телекомунікацій.

У випадку централізованого підходу: таблиці відповідності адрес, наприклад символьних і числових, зберігаються на окремо виділених комп'ютерах мережі (серверах імен); інші комп'ютери мережі звертаються до них для того, щоб за символьною адресою встановити числову.

У випадку розподіленого підходу кожен комп'ютер мережі розрізняє адреси самостійно. Наприклад, щоб розрізнити числову адресу, комп'ютер

відправляє широкомовне повідомлення з проханням розпізнати цю адресу. Той комп'ютер, числова адреса якого співпадає з запитуваною, надсилає у відповідь відповідну адресу іншого виду, наприклад апаратну.

Перевагою розподіленого підходу є те, що він не вимагає виділення спеціального комп'ютера і формування на ньому таблиці відповідності адрес. Недоліком розподіленого підходу є те, що в ньому використовуються широкомовні повідомлення, які обробляються всіма вузлами мережі і тому перевантажують мережу. Тому розподілений підхід може використовуватись тільки в невеликих мережах. В крупних мережах доводиться використовувати централізований підхід.

У стеку протоколів TCP/IP використовуються три типи адрес:

- локальні (апаратні) адреси – використовуються на каналному рівні для доставки даних в мережах із заданою базовою технологією;
- мережеві (IP) адреси – використовуються на мережевому рівні для пересилки даних між мережами;
- символічні адреси – використовуються на прикладному рівні з метою забезпечення комфортної роботи з мережевими ресурсами.

9.1. Локальні адреси

Локальна, або апаратна адреса – це адреса каналного рівня, яка використовується засобами базової мережевої технології для доставки даних в межах підмережі складеної мережі (інтермережі).

Якщо підмережею інтермережі є локальна мережа, то локальна адреса – це MAC-адреса. Проте до складу інтермережі можуть входити також підмережі, що використовують технології глобальних мереж (X.25, АТМ та інші). В цьому випадку локальними адресами будуть відповідні адреси цих технологій.

MAC-адреса – унікальний ідентифікатор мережевого інтерфейсу в локальній мережі, що зафіксований в його апаратурі.

MAC-адреса представляє собою двійкове число довжиною 48 біт, яке для простоти сприйняття часто записують у вигляді 12 цифр в 16-й системі числення, розділених дефісами, наприклад, 00-C0-DF-11-47-9F. В такому записі перші шість цифр ідентифікують виробника мережевого пристрою, а інші шість – сам пристрій. Простір MAC-адрес є плоским.

Мережевий інтерфейс приймає і обробляє тільки кадри з MAC-адресою, яка співпадає з його власною. Всі інші кадри просто відкидаються.

На сусідньому з канальним мережевим рівнем використовується інший тип адрес – мережеві адреси. Для розрізнення мережевих адрес в локальні протокол розрішення адреси – ARP (Address Resolution Protocol).

В локальних мережах, що використовують спільно поділюване середовище передачі даних (наприклад, Ethernet, Token Ring та інші), ARP використовує широкомовні кадри канального рівня. В глобальних мережах (X.25, Frame Relay), як правило, широкомовний доступ не підтримується.

Вузол, якому необхідно розрізнити мережеву адресу як фізичну, формує кадр запиту (ARP Request) і розсилає його широкомовно. Всі вузли локальної мережі отримують ARP-запит і порівнюють вказану в ньому мережеву адресу зі своєю власною. У випадку їх збігу вузол формує ARP-відповідь (ARP Replay), вказавши в ній свою MAC-адресу, і направлено відправляє його вузлу, від якого надійшов ARP-запит.

З метою зменшення кількості широкомовних кадрів ARP-запитів стек протоколів Microsoft TCP/IP (як і інші) містить підтримку кеша ARP – таблиці зі списком IP-адрес, що вже були розрізнені в MAC-адреси. Кеш ARP перевіряється перед відправкою ARP-запиту. Кожний інтерфейс має власний кеш ARP, переглянути який дозволяє команда `arp` з ключем `-a`.

9.2. Мережеві адреси

Мережева адреса, або IP-адреса – це логічна адреса мережевого рівня, яка використовується в якості ідентифікатора вузла складеної мережі

(інтермережі).

Протокол Ipv4 виділяє на IP-адресу двійкове число довжиною 8 байтів (32 біти). Відповідно, протокол Ipv6 виділяє 16 байтів (128 бітів) і це найсуттєвіша різниця між IPv4 та IPv6.

Двійкова нотація Ip-адреси не зручна для використання. Тому IP-адреса записується в десятковій, або шістнадцятковій, системі числення. Наприклад, 32-біти для протоколу IPv4 розбиваються на чотири октети (по 8 бітів), кожен з яких перетворюється в десяткове число (в діапазоні 0-255) і відділяється від інших крапкою, як показано в Таблиці 9.1.

Таблиця 9.1. Переведення двійкової нотації Ip-адреси в десяткову

Двійкова нотація IP-адреси :	100000000000010100000001000011110			
Виокремлені октети в 2-й системі числення	10000000	00001010	00000010	00011110
Десяткові значення октетів:	128	10	2	30
Десяткова нотація IP-адреси:	128.10.2.30			

IP-адреси є ієрархічними: вони складаються з двох частин, перша з яких ідентифікує мережу з даним вузлом, а друга – власне вузол. Яка частина IP-адреси є ідентифікатором мережі, яка вузла може визначатись двома шляхами:

- на основі класів,
- на основі масок.

9.2.1. Адресація на основі класів

В адресації на основі класів ідентифікатор мережі визначається значеннями перших бітів адреси. У відповідності з ними виділяються п'ять класів, що показані в таблиці 9.2.

Якщо адреса починається з 0, то вона відноситься до класу A і під номер мережі відводиться перший октет.

Таблиця 9.2. П'ять класів адрес, відповідно до протоколу IPv4

	1-й октет	2-й октет	3-й октет	4-й октет
Клас А	0 Номер мережі	Номер вузла		
Клас В	10 Номер мережі		Номер вузла	
Клас С	110 Номер мережі			Номер вузла
Клас D	1110 Широкомовні адреси			
Клас Е	11110 Зарезервовані адреси			

Якщо адреса починається з 10, то вона відноситься до класу В і під номер мережі відводиться два перших октети.

Якщо адреса починається з 110, то вона відноситься до класу С і під номер мережі відводиться три перших октети.

Якщо адреса починається з 1110, то вона належить до класу D і визначає особливу широкомовну адресу (multicast): пакети з такою адресою направляються одночасно всім вузлам з цією адресою.

Якщо адреса починається з 11110, то вона належить до класу Е, який зарезервований для майбутніх застосувань.

Стек протоколів Microsoft TCP/IP підтримує адреси класів А, В і С.

Серед адрес класів А, В і С існує набір спеціально зарезервованих адрес, які не можуть використовуватись для ідентифікації вузлів:

- в номері мережі, або вузла, не можна встановлювати всі біти в 1.
- в номері мережі, або вузла, не можна встановлювати всі біти в 0.
- номер мережі не може починатися з 127.

Такі адреси мають спеціальне застосування. Так, IP-адреси, що починаються з 127 використовуються для тестування програмного забезпечення і взаємодії мережевих процесів в рамках окремого вузла. Наприклад, пакет з адресою 127.0.0.1 не передаватиметься в мережу, а повертатиметься верхнім рівням протоколів.

Характеристики адрес класів А, В і С, які можна використовувати для ідентифікації вузлів, можна звести в Таблицю 9.3.

Таблиця 9.3. Характеристика адрес за класом, відповідно до протоколу IPv4

Клас	Діапазон значень першого октету	Максимальна кількість мереж	Максимальна кількість вузлів в мережі
А	1-126	126	16 777 214
В	128-191	16 382	65 534
С	192-223	2 097 150	254

9.2.2. Маски адрес

Адресацію на основі класів не можна вважати раціональною, оскільки при її використанні виявляється фіксованою кількість вузлів в мережі. Тому на даний час широко використовується інший, більш гнучкий, спосіб визначення границі між номерами мережі і вузла – на основі маски підмережі.

Згідно з визначенням, даним в RFC 950:

Маска адреси, або маска підмережі – це 32-бітне число, що використовується для розрізнення номера мережі і номера вузла в IP-адресі, біти якої задаються так:

- всі біти, що відповідають номеру мережі, встановлюються в 1.
- всі біти, що відповідають номеру хосту, встановлюються в 0.

Іншими словами, маска адреси – це двійкове число, яке використовується в парі з IP-адресою і має таку ж довжину. Маска адреси містить неперервну послідовність одиниць в тих розрядах, які позначають номер мережі, і неперервну послідовність нулів в тих розрядах, які позначають номер вузла.

Маска «маскує» частину IP-адреси, яка використовується для отримання номера мережі.

Маски адрес для стандартних класів показано в Таблиці 9.4.

Таблиця 9.4. Маски адрес для класів

Клас адрес	Біти маски	Маска в 10-й нотації	Маска в 16-й нотації	Маска у вигляді префіксу
A	11111111 00000000 00000000 00000000	255.0.0.0	FF.00.00.00	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	FF.FF.00.00	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	FF.FF.FF.00	/24

Крім двійкової нотації для представлення масок також використовуються:

- точково-десятькова нотація – використовує перетворення 32-бітного значення маски в точково-десятькову форму,
- шістнадцятикова нотація – значення кожного октету 32-бітного значення маски записується у вигляді 16-кового числа,
- у вигляді префіксу адреси – вказується число бітів номера мережі в 32-бітному значенні маски; це число записується у вигляді /<кількість бітів> і називається префіксом адреси (мережі).

Приклад розбиття адрес класу C на підмережі з використанням маски показано в Таблиці 9.5.

9.2.3. Загальні і приватні адреси

При роботі в Інтернет використовуються два типи IP-адрес: загальні (public) і приватні (private).

Загальні адреси призначаються комітетом InterNIC і складаються з ідентифікаторів мереж, глобально унікальних в Інтернет. При цьому маршрутизатори Інтернету забезпечують доставку трафіка, що направляється на ці адреси, за призначенням. Цей трафік доступний в Інтернет.

В приватних мережах, які не планується під'єднати до Інтернет, можуть використовуватись будь-які адреси, в тому числі і такі, які призначаються InterNIC. Проте, якщо в майбутньому така мережа

підключиться до Інтернет, може виявитись, що в ній використовуються адреси, які вже виділені InterNIC іншим організаціям. Такі дубльовані адреси називаються недопустимими (illegal addresses). Інтернет-з'єднання з цих адрес неможливі.

Таблиця 9.5. Використання маски для розбиття на підмережі

Необхідна кількість підмереж	Число бітів для ідентифікації підмережі	Маска підмережі	Кількість вузлів в підмережі
1-2	1	255.255.255.128	126
3-4	2	255. 255.255.192	62
5-8	3	255. 255.255.224	30
9-16	4	255. 255.255.240	14
17-32	5	255. 255.255.248	6
33-64	6	255. 255.255.252	2

Як правило, далеко не всі вузли потребують прямого з'єднання з Інтернет. Основній масі організацій потрібні невеликі діапазони адрес лише для вузлів, напряду підключених до Інтернет – маршрутизаторів, проксі-серверів, брандмауерів, або трансляторів мережевих адрес (network address translator, NAT).

Вузлам, що не потребують прямого доступу до Інтернет, повинні бути призначені IP-адреси, які ніколи не виділяються як загальні. Спеціально для цього зарезервована частина адресного простору IP, яка отримала назву приватного адресного простору (private address space). IP-адреси з цього простору ніколи не виділяються як загальні і називаються приватними адресами (private addresses).

Приватний адресний простір визначається трьома адресними блоками (RFC 1918):

- 10.0.0.0/8 – ідентифікатор мережі класу А, що допускає IP-адреси в діапазоні від 10.0.0.1 до 10.255.255.254.
- 172.16.0.0/12 – ідентифікатор мережі класу В, що допускає IP-адреси в діапазоні від 172.16.0.1 до 172.31.255.254.

– 192.168.0.0/16 – ідентифікатор мережі класу C, що допускає IP-адреси в діапазоні від 192.168.0.1 до 192.168.255.254.

Маршрутизатори Інтернет не підтримують маршрутизацію за приватними адресами і тому вони є недоступними з Інтернет.

Інтернет-трафік від вузла з приватною адресою повинен надходити шлюзу прикладного рівня (наприклад, проксі-серверу), або транслятору мережевих адрес, який перед відправкою трафіку в Інтернет перетворює приватні адреси в допустимі загальні адреси.

9.3. Символьні адреси

Інтернет-додатки для з'єднання з адресатом можуть використовувати IP-адреси. Проте для забезпечення більш комфортної роботи користувачів передбачена також можливість використання символьних, так званих хост-імен.

Хост-ім'я (host name) – це символьний псевдонім, що призначається комп'ютеру для його ідентифікації в якості TCP/IP-вузла (хоста).

Довжина хост-імені може досягати 255 знаків і містити алфавітно-цифрові символи, дефіси, точки. Одному хосту можна призначити кілька хост-імен.

Ім'я комп'ютера в Windows може не співпадати з хост-ім'ям.

Найбільш поширеними формами хост-імен в Microsoft TCP/IP є:

- зрозуміле ім'я (nickname) – псевдонім, що призначається користувачем.
- доменне ім'я (domain name) – структуроване ім'я, що формується згідно домовленостям, прийнятим в Інтернет.

Зрозумілі імена утворюють плоский адресний простір, який не забезпечує унікальність імен в межах великої мережі.

Для ефективно організації іменування комп'ютерів в великих мережах ефективним є застосування ієрархічних адрес, якими є доменні імена.

Саме для побудови гнучкої і масштабованої схеми іменування, InterNIC створив ієрархічну систему доменних -DNS.

DNS (Domain Name System) – це централізована служба, що ґрунтується на розподіленій базі відповідностей «доменне ім'я – IP-адреса».

Доменна система імен має ієрархічну деревоподібну структуру, що допускає використання в імені довільної кількості складових частин, як показано на рис. 9.3.

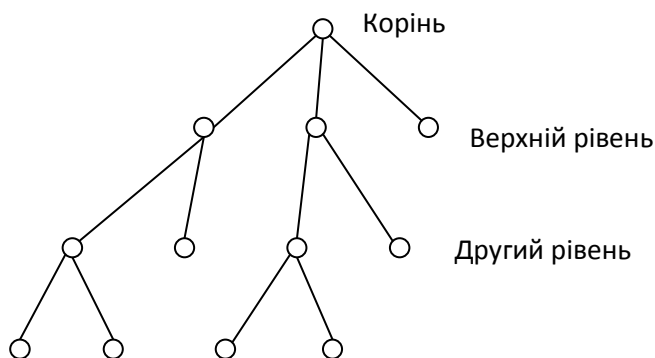


Рис. 9.3. Структура доменної системи імен

Доменне ім'я формується з ланцюжка імен в напрямку від хосту до кореня. Складові частини імені відділяються одна від іншої крапками.

Розділення імені на частини дозволяє поділити адміністративну відповідальність за призначення унікальних імен між різними організаціями в межах свого рівня ієрархії.

Кореневий домен управляється центром InterNIC. Домени верхнього рівня призначаються для кожної країни, а також на організаційній основі. Для позначення країн використовуються двобуквені аббревіатури, а для типів організацій – com (комерційні), edu (освітні), gov (державні), net (мережеві), agr (утворюють ARPANET), org (інші організації).

Кореневий домен утворюють 16 серверів, які називаються серверами кореневої зони (root zone servers). Вони містять копії одних і тих самих

файлів і отримали назви `a.root_server.net`, `b.root_server.net` та ін. Перший з них – `a.root_server.net` – виступає в ролі первинного сервера імен Інтернет. Всі інші є по відношенню до нього вторинними.

На серверах кореневої зони розміщується інформація про сервери імен, що обслуговують домени верхнього рівня.

Кореневі сервери DNS видають тільки вказівники на сервери DNS наступного верхнього рівня, тобто є рекурсивними. Всі інші DNS-сервери (крім серверів кореневої зони), що встановлюються в доменах нижніх рівнів, називаються локальними серверами імен і саме вони відповідають за розрізнення імен.

Локальний сервер імен має адресу кореневого сервера і формує запит, послідовно запитуючи у серверів DNS вказівники на рівні ієрархічного дерева, поки не добереться до кінця цього ланцюжка і не отримає потрібну IP-адресу. При цьому локальний сервер буде повторювати процедуру запитування інформації про інші сервери імен до тих пір, поки не отримає відповіді. Тому локальні сервери імен є ітераційними.

Приклад пошуку доменного імені `computer.division.firm.com` показано на рис. 9.4.

Іншим способом розрізнення хост-імен в IP-адреси є використання локальної бази даних, яка в Windows зберігається в файлі `hosts` з каталогу `\%System-Root%\system32\drivers\etc`.

Перевагою використання файлу `hosts` в тому, що користувач може створювати в ньому потрібні записи, даючи часто використовуваним ресурсам зрозумілі імена. Проте використання файлу `hosts` є неефективним при великій кількості записів. Microsoft TCP/IP дозволяє розрізняти хост-імена як через локальний файл `hosts`, так і через DNS-сервер:

- TCP/IP спочатку перевіряє, чи міститься дане хост-ім'я в файлі `hosts`.
- якщо в файлі `Hosts` його не має, формується запит до DNS-сервера.

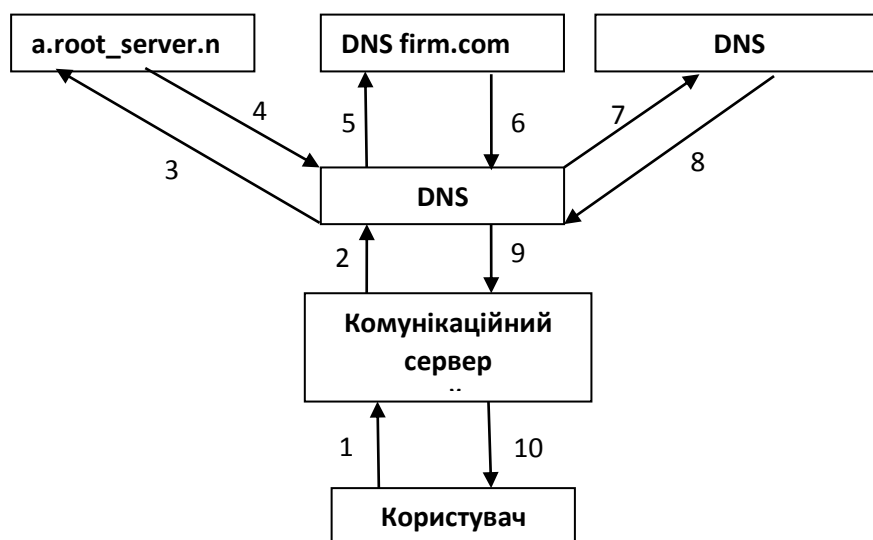


Рис. 9.4. Пошук доменного імені computer.division.firm.com

Контрольні запитання до розділу

1. Види адресних просторів, адреси в стеку TCP/IP.
2. Реалізація локальних адрес та запитів до неї.
3. Мережева адреса, нотація мережевих адрес.
4. Класи мережевих адрес, їх відповідність маскам адрес.
5. Маски IP адрес, розбиття на підмережі.
6. Поняття загальної та приватної адреси.
7. Символьні адреси, хост та доменна система адрес, доменне ім'я комп'ютера.
8. Локальний та кореневий сервери, хост-каталог.

10. ТОПОЛОГІЯ ЛОКАЛЬНОЇ МЕРЕЖІ

10.1. Поняття топології мережі

Топологія визначає спосіб з'єднання вузлів мережі (комп'ютерів і комунікаційного обладнання) або їх груп фізичними лініями зв'язків.

Поняття топології відноситься, перш за все, до локальних мереж, в яких структуру зв'язків можна прослідкувати; у глобальних мережах структура зв'язків прихована від користувачів і може змінюватись з часом.

Під топологією локальної мережі розуміється конфігурація графа, вершинами якого є її вузли (комп'ютери та інше комунікаційне обладнання), а ребрами – фізичні лінії зв'язку.

У разі об'єднання двох-трьох вузлів мережі проблем з лініями зв'язку не виникає, як показано на рис. 10.1. Інша справа, якщо таких вузлів мережі буде, наприклад, п'ять. У цьому разі кількість ліній зв'язку буде вже 10, що ускладнює мережу. Тому розрізняють два види топологій:

- повнозв'язні топології – передбачають наявність окремої лінії зв'язку для будь-якої пари вузлів.
- неповнозв'язні топології – одержуються з повнозв'язної топології шляхом видалення окремих ліній зв'язку.

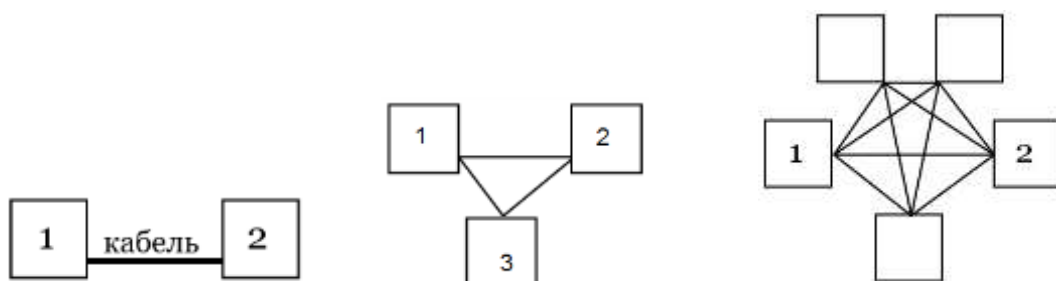


Рис. 10.1. Приклади повнозв'язної топології мережі

Якщо в мережі налічується N вузлів, то для утворення повнозв'язної

топології необхідно $N(N-1)/2$ ліній зв'язку. Тобто, така конфігурація виявляється громіздкою при збільшенні кількості вузлів і тому не набула поширення.

Спосіб, який дозволяє зменшити кількість використовуваних ліній зв'язку, полягає у тому, щоб надати окремим вузлам мережі можливість пересилати повідомлення «транзитом». Такими вузлами можуть виступати як універсальні комп'ютери, так і спеціалізовані пристрої (мости, комутатори та ін.). В цьому випадку повідомлення від одного вузла до іншого можуть передаватись через «ланцюжок» транзитних вузлів, а отже кількість безпосередніх, прямих з'єднань зменшується. Такого виду конфігурація є неповнозв'язаною, оскільки кожний комп'ютер безпосередньо не сполучається з усіма іншими. Саме неповнозв'язні конфігурації набули широкого поширення і стали основними в технологіях локальних мереж.

При об'єднанні більше ніж двох комп'ютерів, з ростом кількості вузлів мережі стрімко зростає кількість варіантів з'єднань. Це відбувається за рахунок неповнозв'язних конфігурацій, як показано на рис. 10.2.

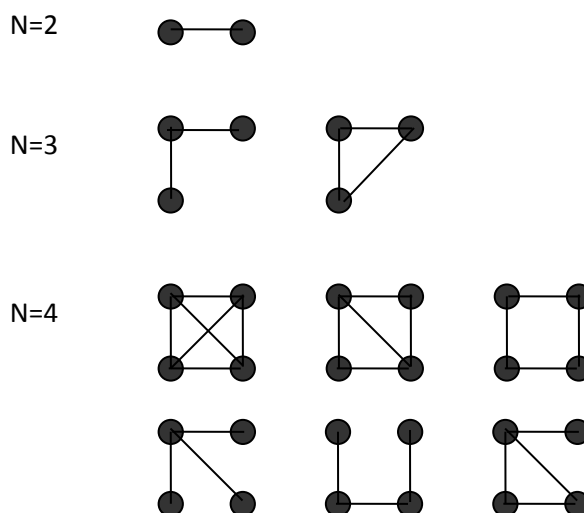


Рис. 10.2. Зростання кількості варіантів з'єднань при об'єднанні N комп'ютерів

Виникає проблема вибору конфігурації фізичних зв'язків або фізичної топології мережі.

Існують три базові топології локальних мереж:

- шина (Bus) – всі комп'ютери підключаються до однієї спільної лінії зв'язку;
- зірка (Star) – до одного центрального комп'ютера приєднується решта периферійних комп'ютерів, причому кожен з них використовує окрему лінію зв'язку;
- кільце (Ring) – комп'ютери послідовно об'єднуються в «кільце»: кожен з них отримує дані лише від попереднього і передає дані лише наступному.

Базові топології можуть комбінуватись між собою у різних варіантах, утворюючи інші, більш складні, змішані топології.

10.2. Топологія «шина»

В топології «шина» всі вузли мають рівноправний доступ до лінії зв'язку, яка використовується ними по черзі, як показано нижче – на рис. 10.3.

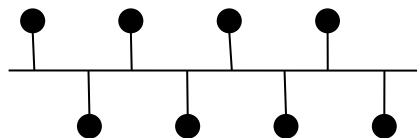


Рис. 10.3. Топологія «шина»

Для доступу до лінії зв'язку не використовується спеціальний централізований пристрій, як в інших топологіях. Відсутність такого пристрою підвищує надійність системи. Але в цьому випадку функції управління доступом до лінії зв'язку мають покладатися на мережеве обладнання, що підвищує вимоги до нього.

Важливою особливістю топології «шина» є те, що вона нечутлива по

відношенню до відмови одного з вузлів (всі інші можуть продовжувати обмін інформацією), але дуже чутлива до пошкоджень в кабельній системі і відмов мережевого обладнання. Так може здатися, що при обриві кабелю утворюються дві цілком працездатні шини. Проте це не так: особливості розповсюдження електричних сигналів по довгим лініям зв'язку вимагають включення на кінцях шини спеціальних погоджуючи пристроїв – термінаторів. Без них сигнал відображається від кінця лінії і спотворюється так, що зв'язок по мережі стає неможливим. Тобто у разі розриву, або пошкодження кабелю порушується узгодження лінії зв'язку і припиняється обмін інформацією навіть між тими вузлами, які залишилися сполученими між собою. Коротке замикання в будь-якій точці кабелю шини також виводить з ладу всю мережу.

Інша особливість топології «шина» пов'язана з ослабленням сигналів при їх проходженні лініями зв'язку, що накладає жорсткі обмеження на сумарну довжину ліній зв'язку. Певним чином збільшити її можна через використання спеціальних відновників сигналів (повторювачів), проте цей спосіб також має свої обмеження, які пов'язані з кінцевою швидкістю поширення сигналів лініями зв'язку.

Слід зауважити, що топологія «шина» вимагає використання мінімальної кількості кабелю у порівнянні з іншими топологіями. А додавання нових абонентів в шину досить просте і можливе навіть під час роботи мережі.

Таким чином основні переваги і недоліки топології «шина» шини полягають у такому.

Переваги:

- простота побудови і невисока вартість,
- нечутливість по відношенню до відмов окремих вузлів.

Недоліки:

- чутливість до пошкоджень кабельної системи,

- жорсткі обмеження на сумарну довжину ліній зв'язку,
- складність мережевого обладнання.

10.3. Топологія «зірка»

В топології «зірка» один з вузлів є центральним (найчастіше в якості центрального вузла використовуються повторювач або комутатор) до якого під'єднуються інші периферійні вузли [1]. Обмін інформацією між периферійними вузлами здійснюється виключно через центральний вузол, тому він виявляється найбільш завантаженим, як показано на рис. 10.4.

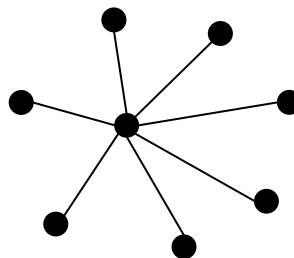


Рис. 10.4. Топологія «зірка»

Внаслідок повністю централізованого управління конфлікти в топології «зірка» в принципі неможливі.

Топологія «зірка» є стійкою по відношенню до відмов периферійних вузлів, проте будь-яка відмова центрального вузла робить мережу повністю непрацездатною. У зв'язку з цим необхідно вживати спеціальних заходів щодо підвищення надійності центрального комп'ютера і його мережевого обладнання.

Обрив кабелю або коротке замикання в ньому порушує обмін лише з одним вузлом, а вся решта може нормально продовжувати роботу.

Те, що кожна лінія зв'язку використовується лише двома абонентами – центральним і одним з периферійних вузлів – дозволяє суттєво спростити мережеве обладнання.

Суттєвим недоліком топології «зірка» є жорстка обмеженість кількості

периферійних комп'ютерів: їх число, як правило, не може перебільшувати 8-16. У цих межах підключення нових комп'ютерів здійснюється досить просто, але за ними воно просто неможливе. Проте допускається можливість нарощування мережі за рахунок підключення замість периферійного ще одного центрального вузла.

Ще одним недоліком топології «зірка» є значно більші, ніж в інших топологіях, витрати кабелю, що впливає на вартість всієї мережі.

Таким чином основні переваги і недоліки топології «зірка» шини полягають у такому.

Переваги:

- принципова неможливість виникнення конфліктів між вузлами,
- стійкість до пошкодження кабельної системи,
- нечутливість по відношенню до відмови окремих вузлів.

Недоліки:

- жорстке обмеження кількості периферійних вузлів,
- висока вартість.

10.4. Топологія «кільце»

В топології «кільце», що показана на рис. 10.5, кожний вузол сполучається лініями зв'язку лише з сусідніми: від одного одержує інформацію, а іншому передає.

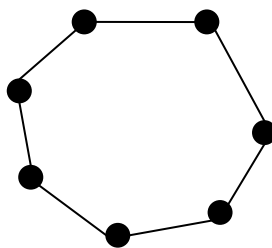


Рис. 10.5. Топологія «кільце»

Кільцева топологія має високу стійкість до перевантажень, оскільки в ній відсутні конфлікти, пов'язані з одночасним захопленням лінії зв'язку кількома комп'ютерами (як у випадку шини), а також відсутній центральний абонент, який може бути переобтяжений великими потоками інформації (як у випадку зірки). Тому ця топологія забезпечує упевнену роботу з великими потоками інформації по мережі.

Важливою особливістю топології «кільце» є те, що кожен вузол підсилює сигнал, що надходить до нього, тому розміри кільцевих мереж можуть досягати значних розмірів (на практиці – десятки кілометрів). Кільце в цьому відношенні істотно перевершує будь-які інші топології.

Як і у разі шини, максимальна кількість вузлів в кільці може бути досить великою (~1000). Підключення нових комп'ютерів до кільця виконується досить просто, хоча і вимагає обов'язкової зупинки роботи мережі на цей час.

Сигнал у кільці проходить послідовно через всі комп'ютери мережі, тому вихід з ладу одного з них (або ж його мережевого обладнання) порушує роботу мережі в цілому і в цьому полягає основний недолік кільця. Обрив або коротке замикання в будь-якому з кабелів кільця теж унеможливило б роботу всієї мережі. Кільцева топологія є найуразливішою до пошкоджень кабелю, тому у разі її використання часто передбачається прокладка двох (або більш) паралельних ліній зв'язку, одна з яких знаходиться в резерві.

Таким чином основні переваги і недоліки топології «зірка» шини полягають у такому.

Переваги:

- висока стійкість до перевантажень,
- велика кількість вузлів,
- значні розміри мережі.

Недоліки:

- критичність по відношенню до відмови окремих вузлів,
- вразливість до пошкодження кабельної системи,
- висока вартість.

10.5. Змішані топології

Базові топології є основою для створення інших більш складних конфігурацій.

Так у результаті комбінації кількох мереж з топологією «зірка» утворюється топологія «дерево» (tree), показана на рис. 10.6. При цьому в ієрархічному порядку об'єднуються лише центральні вузли.

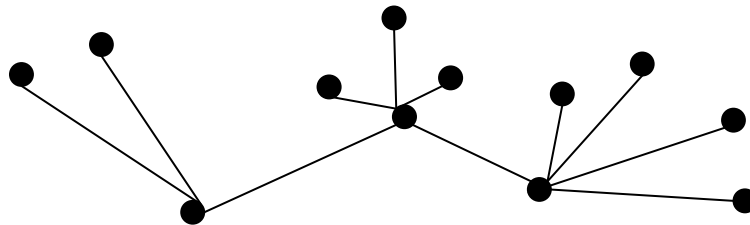


Рис. 10.6. Топологія «дерево»

Досить поширеною є зірково-шинна (Star-Bus) топологія, показана на рис. 10.7, в якій підмережі з топологією «шина» підключаються до спільного центрального вузла.

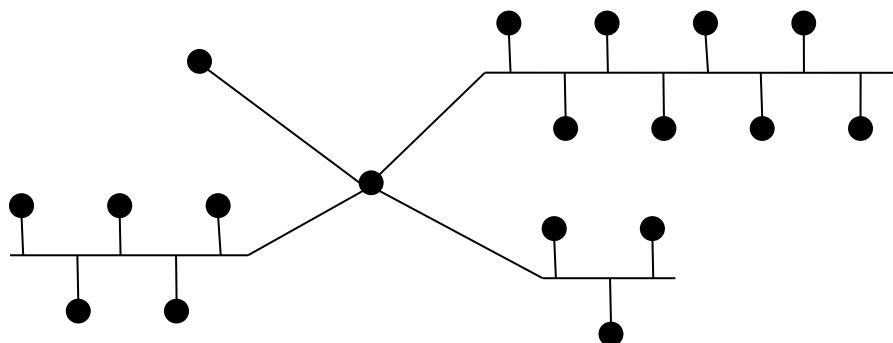


Рис. 10.7. Топологія «зірка-шина»

Центральні вузли зірки можуть об'єднуватись між собою, утворюючи

так називану магістральну або опорну шину. Таким чином вдається комбінувати переваги шинної і зіркової топологій, а також легко нарощувати мережу.

У разі об'єднання центральних вузлів зірок у кільце утворюється зірково-кільцева (Star-Ring) топологія, що показано на рис. 10.8.

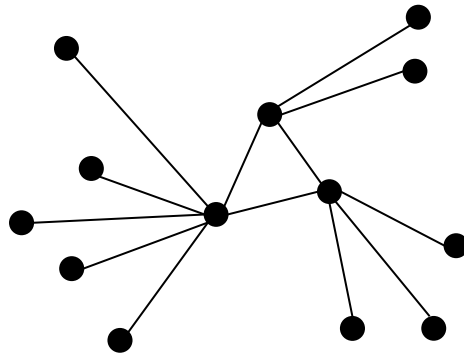


Рис. 10.8. Топологія «зірка-кільце»

Вона дозволяє комбінувати переваги відповідно зіркової і кільцевої топологій, наприклад, для збільшення розмірів мережі.

Контрольні запитання до розділу

1. Поняття топології мережі, базові топології локальних мереж.
2. Переваги та недоліки топології «шина».
3. Переваги та недоліки топології «зірка».
4. Переваги та недоліки топології «кільце».
5. Топологія «зірка-шина», топологія «зірка-кільце», їх переваги та недоліки.

11. МЕТОДИ ДОСТУПУ

11.1. Загальна характеристика методів доступу

Локальна комп'ютерна мережа об'єднує між собою абонентські станції за допомогою спільно використовуваного середовища передачі даних. При цьому одночасна передача кількома станціями своїх кадрів є неможливою, оскільки веде до перекручення і втрати обох кадрів – виникає колізія (конфлікт). Виникає необхідність у визначенні методів (правил), за якими станції отримують доступ до поділюваного середовища і, відповідно, права на передачу. Тому в кожній мережі застосовується той або інший метод доступу, що упереджує виникнення конфліктів між станціями.

Метод доступу (метод арбітражу або метод управління обміном) – це спосіб визначення того, яка з абонентських станцій зможе наступною використовувати поділюване середовище для передачі даних.

Метод доступу визначає технологію спільного використання середовища множиною вузлів мережі.

Метод доступу – одна з найважливіших характеристик мережі. Від його ефективності залежить швидкість обміну інформацією в мережі, її навантажувальна здатність, час реакції мережі на зовнішні події та інше.

Тип методу доступу багато в чому визначається особливостями топології мережі, проте не пов'язаний з нею жорстко. Основні методи доступу показано на рис. 11.1. Методи доступу поділяються на:

- централізовані – управління доступом зосереджене в одному вузлі;
- децентралізовані – виділений центр управління відсутній.

Перевагою централізованих методів є відсутність конфліктів, а їх недолік – нестійкість до відмов центрального вузла. Перевагою децентралізованих методів є висока стійкість до відмов, а недолік – виникнення конфліктів, які необхідно усувати.

Децентралізовані методи доступу у свою чергу поділяються на:

- детерміновані – передбачають порядок надання доступу до середовища передачі за визначеними правилами; виникнення конфліктів при цьому майже виключається;
- випадкові – передбачають довільний (випадковий) порядок доступу до середовища передачі; виникнення конфліктів при цьому є очікуваним і вимагаються способи їх розрізнення.



Рис. 11.1. Основні методи доступу

Випадкові методи гірше працюють при великій інтенсивності обміну в мережі і не гарантують абоненту час доступу. Час доступу – інтервал між виникненням необхідності передавати і отриманням можливості передати кадр. Проте вони є більш стійкими до відмов мережевого обладнання і більш ефективно використовують мережу при малій інтенсивності обміну. Допустимим вважається навантаження не вище 30-40%. При більшому навантаженні різко зростає кількість колізій, що приводить до різкого падіння продуктивності мережі – настає колапс, або крах мережі.

11.2. Метод доступу CSMA/CD

Метод доступу CSMA/CD (Carrier Sense Multiple Access with Collision Detection) або метод множинного доступу з прослуховуванням несучої і розрізненням колізій, що використовується в технології Ethernet, є одним з найбільш поширених децентралізованих методів.

Метод полягає у наступному. Кожна абонентська станція прослуховує канал, щоб з'ясувати, чи є він вільний в даний момент. Якщо середовище вільне, то станція, яка має кадр для передачі, починає його передавати. У разі виявлення сигналу, що свідчить про вже розпочату передачу іншою станцією, передача свого кадру припиняється на деякий інтервал часу, після якого спроба отримати доступ до каналу повторюється.

Після завершення передачі кадру станція має витримати паузу в 9,6 нс – міжкадровий інтервал (inter packet gap).

Конкуренція між станціями за захоплення середовища передачі (каналу), як зображено на рис. 11.2, може привести до виникнення колізій, викликаних тим, що:

- дві станції виявляють, що середовище передачі вивільнилось, і одночасно розпочинають передачу своїх кадрів;
- дві станції розпочинають передачу майже одночасно - протягом інтервалу, що менший за час затримки сигналу в лінії передачі.

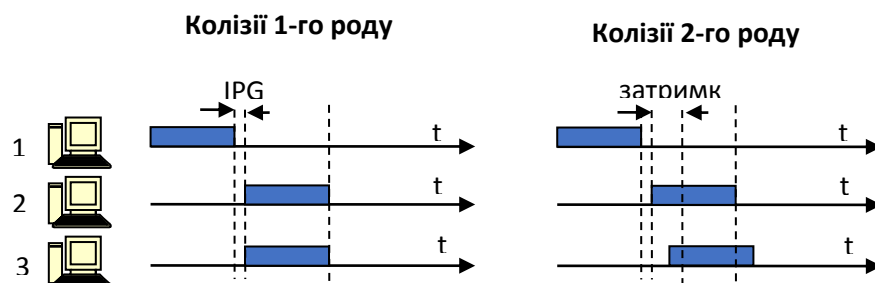


Рис. 11.2. Виникнення колізії в алгоритмі CSMA/CD

При виявленні колізії станції переривають передачу своїх кадрів і переходять в режим очікування. Час очікування у кожній станції вибирається випадковим чином і може складати від 0 до 52,4 мс. По закінченню часу очікування здійснюється спроба відновлення перерваної передачі. Тобто метод доступу CSMA/CD є випадковим. Приклад роботи алгоритму CSMA/CD показано на рис. 11.3.

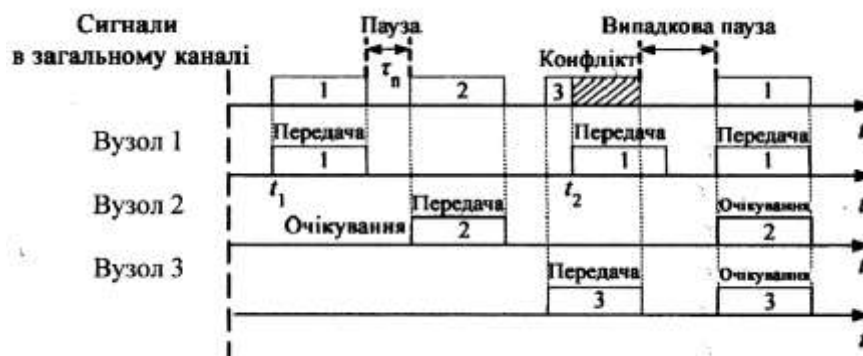


Рис. 11.3. Приклад роботи алгоритму протоколу CSMA/CD

Вузол 1 першим просканував загальний канал і не виявив CS, або CD. Вузол 1 передає свій пакет у загальний канал, що був вільний. Його отримують всі інші вузли (ПК). Вузол 2 теж хоче передати, але чекає на звільнення загального каналу плюс пауза, протягом якої може прийти CD. Потім Вузол 2 передає свій пакет у загальний канал. Аналогічно, Вузол 3 отримує доступ до загального каналу та передає свій пакет. Проте, в цей момент можливо Вузол 1 не виявив, що загальний канал – зайнятий і почав передачу пакета (заштриховано на рис. 11.3. В цьому разі інформація у загальній шині буде змінена та не співпадатиме з контрольною сумою. У цьому разі всі Вузли зупиняють передачу на термін, що визначається часом отримання 32 bits на швидкості 10 Mbit/sec за максимальної кількості повторень 16.

$$32 \cdot \frac{16}{10^7} = 51,2 \text{ мкс}$$

Далі Вузол-відправник, після паузи, повторює передачу пакету.

Якщо 16 повторень не дали результат - дається access denied (т. зв. усічений двійковий алгоритм відстрочки). Для зменшення колізій при проектуванні мережі встановлюється коефіцієнт завантаження мережі до 30%. Колізія призводить до затримки у роботі мережі, тому її варто відразу виявляти.

Мережеві адаптери могли б розпізнавати колізії на основі порівняння даних, що передаються абонентом, з даними на лінії зв'язку. Але таке просте порівняння можливе лише у випадку найпростішого полярного кодування (код NRZ). Застосування іншого виду кодування вимагає свого підходу до розпізнавання колізій. Зокрема при використанні коду Манчестер-II (манчестерський код, в якому один з двох рівнів сигналу нульовий), використовується та обставина, що за відсутності колізій сигнал на лінії завжди має постійну складову, яка дорівнює половині його розмаху (так званий сигнал за несучою частотою – Carrier Sense), а у випадку колізій це правило виконуватись не буде, оскільки цей сигнал про колізію (32 bit jam signal) відправляється в загальний канал станцією, що виявила не співпадіння контрольної суми у пакета. Тобто, встановлено колізію – Collision Detection (CD). Саме за виходом рівня постійної складової за встановлені межі кожний мережевий адаптер і визначає наявність колізій в мережі, оскільки за умови колективного доступу всі мережеві адаптери відразу отримають дані, які передає один з них – Multiply Access (MA)

Для випадкових методів управління доступом (всіх) важливим є питання про те, якою має бути мінімальна тривалість кадру, щоб колізію (другого роду) могли визначити всі абоненти. Її можна оцінити на основі наступного розрахунку.

Розглянемо мережу діаметру L , в якій сигнали поширюються по кабелю зі швидкістю V . Нехай абонент 1 завершив передачу, а у абонентів 2 і 3 виникла необхідність передачі даних. Абонент 2 розпочне передачу зразу після вивільнення середовища передачі. А абонент 3 дізнається про цю подію

лише через час L/V . Кадр від абонента 3 дійде до абонента 2 ще через час L/V після початку передачі. До цього часу передача абонента 2 ні в якому разі не повинна завершитись, інакше він не виявить колізію. Тому одержуємо, що мінімально допустима тривалість кадру в мережі має складати $2 L/V$. Цей час називається подвоєним, або круговим, часом затримки сигналу в мережі, або PDV (Path Delay Value), як показано на рис. 11.4.

$PDV=2 L/V$ можна розглядати як міру одночасності подій в мережі. Наприклад, маємо для оптоволоконної мережі, що працює за технологією Ethernet, довжина мережі не може бути більша, ніж найвіддаленіший ПК, який може вчасно отримати повідомлення про колізію та повторити відправку останнього пакета.

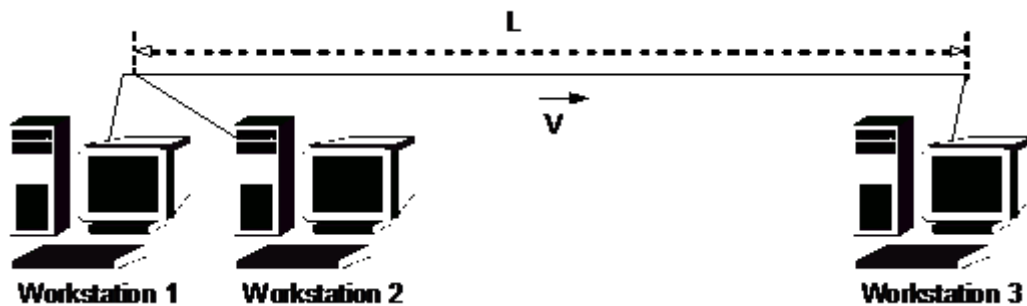


Рис. 11.4. Обрахунок кругового часу затримки сигналу в мережі

Якщо мінімальний розмір пакета Ethernet складає $64 + 8$ (пreamбула) = 72 bytes = 576 bits. А сигнал рухається мережею зі швидкістю 0,6-0,9 від швидкості світла, то час отримання повідомлення про колізію за пропускну здатності мережі 10 Mbit/s складає $576 \cdot 10^{-7} = 57,6$ мкс (так званий Round Time Trip)

А відстань на яку може сягати Ethernet, забезпечуючи захист від колізії визначається як

$$0,6 \cdot 300 \cdot 10^3 \cdot \frac{57,6}{2} = 5,2 \text{ км.}$$

0,6 – коефіцієнт, що вказує на зменшення швидкості передачі,

порівняно з швидкістю світла;

$300 \cdot 10^3$ – швидкість світла, км/с;

$57/2$ – час проходження сигналу про колізію в одному напрямку.

У стандарті вказується, що відстань між станціями з оптоволоконним кабелем не перевищує 2 км, оскільки повторювачі та ін. мережеве обладнання також збільшують час отримання повідомлення про колізію.

З вищенаведеного розрахунку витікає наступне, що вказує на недоліки CDMA/CD:

1) якщо зменшити кількість даних в пакеті, то зменшиться відстань на яку може сягати Ethernet, забезпечуючи захист від колізії. Так званий діаметр домена колізії.

2) Якщо в мережі збільшити пропускну здатність мережі до 100 Mbit/s, то діаметр домена колізії зменшиться до 520 м.

Наприклад, топологія мережі, що зображена на рис. 11.5, містить 6 ПК, три концентратора і один комутатор. Відповідно, мережа містить три домени колізій, тобто три сегменти мережі абоненти яких (ПК) конкурують між собою за передачу сигналів у загальний канал.

Також використання комутаційного обладнання підвищує надійність роботи мережі. У випадку обриву в загальній шині всі ПК не працюватимуть з мережею. Проте, якщо обрив відбудеться у підмережі, що обмежена концентратором, то інші частини мережі – працюватимуть.

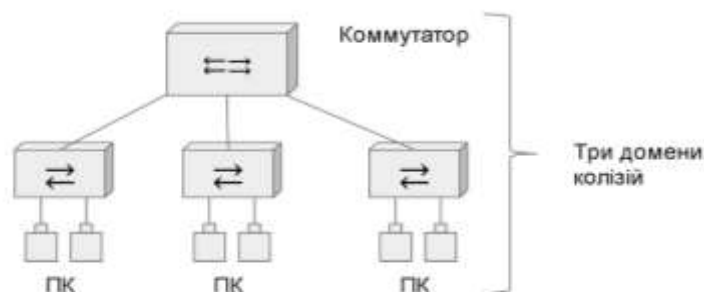


Рис. 11.5. Топологія мережі, що містить три домени колізії

11.2. Метод доступу CSMA/CA

Для уникнення колізій при передачі інформації в безпроводних мережах використовується алгоритм протоколу CSMA/CA із запобіганням колізії.

Оскільки радіопередавачі в основному працюють в напівдуплексному режимі даними, обмін даними відбувається послідовно: один передає дані, а інший – приймає. Окрім цього сигнал передавача може бути набагато сильніший за сигнал приймача. В цьому разі сигнал приймача і сигнал про колізію можуть бути нівельовані зовнішніми шумами.

Тому використовується алгоритм запобігання колізії. Цей алгоритм дещо складніший, ніж алгоритм Collision Detection, оскільки мережа на фізичному рівні є безпроводною.

Алгоритм протоколу CSMA/CA схожий з протоколом це CSMA/CD, оскільки канал прослуховується перед початком відправки пакету. Однак, є наступні відмінності:

- якщо станція-відправник готова переслати кадри, то вона починає період мовчання випадкової довжини;
- якщо період мовчання закінчився і канал не зайнятий, то станція-відправник відсилає кадр;
- якщо кадр дійшов до станції-отримувача, та відсилає станції-відправнику коротке підтвердження (якщо кадр не дійшов, підтвердження немає);
- якщо станція-відправник не отримала підтвердження, то вона подвоює період мовчання та повторює спробу відправки, нарощуючи довжину паузи, поки кадр успішно не буде відправлений, або не буде досягнуто максимальну кількість повторів.

Приклад роботи алгоритму протоколу CSMA/CA показано на рис. 11.6. Як видно з рис. 11.6, станція-відправник I відправляє кадр першою. В цей момент часу станція-відправник II та станція-відправник III знаходяться в

режимі випадкового очікування, оскільки вони знають що канал до станції-отримувача – зайнятий. Після отримання підтвердження станцією-відправником I, станції переходять в режим мовчання. Станції II та III не відправляють свої кадри, оскільки це призвело б до колізії, а починають випадкове очікування. Станція III має короткий період очікування тому вона відправляє дані першою, а станція II після випадкового очікування потрапляє в період мовчання, оскільки станція II отримала підтвердження. Після завершення мовчання станція II завершує випадкове очікування та лише потім відправляє кадр.

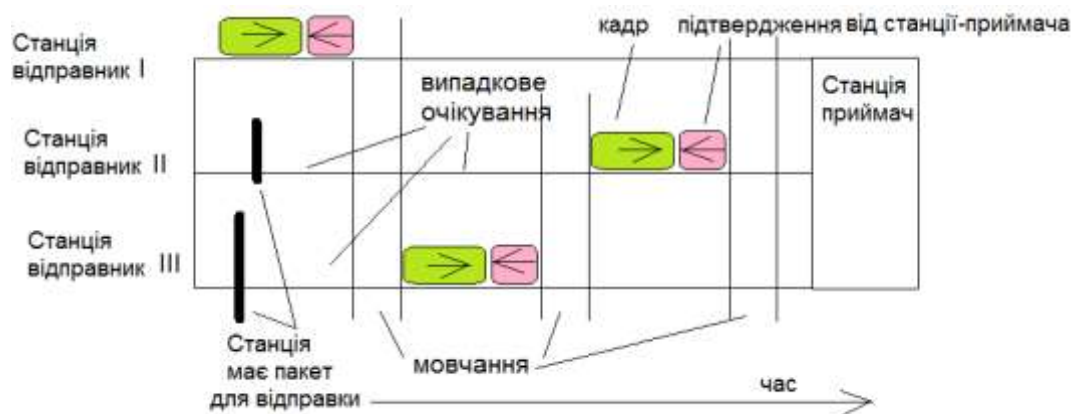


Рис. 11.6. Приклад роботи алгоритму протоколу CSMA/CA

11.3. Метод доступу TPMA

Метод доступу TPMA (Token Passing Multiple Access) – метод з передачею маркера, або маркерний метод. Застосовується в мережах з кільцевою топологією та відноситься до децентралізованих детермінованих методів.

Суть методу полягає у наступному. Кільце розглядається як спільний поділюваний ресурс, доступ до якого здійснюється у відповідності з алгоритмом передачі станціям права на використання кільця за допомогою кадру, який називається маркером.

Маркер – унікальний кадр даних, що постійно циркулює між вузлами

кільцевої мережі і визначає право на доступ станції до кільця.

Мережа з маркерним доступом контролюється так названим активним монітором – абонентською станцією, яка вибирається при ініціалізації мережі. Основними функціями активного монітора є:

- генерація маркера;
- контроль за наявністю маркера: якщо після генерації маркер не повертається активному монітору протягом певного часу, то генерується і запускається в кільце новий маркер.

Алгоритм роботи маркерного метода доступу показано на рис. 11.7 та є наступним:

1. Абонент 1, що має дані для відправки, скажімо, абоненту 3, має дочекатись надходження вільного маркера (ВМ).

2. Отримавши маркер, абонент 1 помічає маркер як зайнятий (ЗМ), додає до нього свій кадр (К) і відправляє цю посилку ЗМ+К наступному в кільці абоненту. Інші абоненти, отримавши посилку ЗМ+К, перевіряють, чи не співпадає адреса призначення з їх адресою, та якщо ні – передають посилку далі по кільцю.

3. Абонент 3, якому адресована посилка, приймає кадр, встановлює в маркері біт підтвердження прийому (ПМ) і передає посилку ПМ+К далі по кільцю.

4. Посилка ПМ+К доставляється абоненту 1, пройшовши через все кільце. Абонент 1 помічає маркер як вільний, видаляє з посилки свій кадр і передає вільний маркер ВМ далі по кільцю.

Головна перевага метода ТРМА перед CSMA/CD полягає в тому, що в ньому гарантується час доступу абонентів до середовища передачі T : $T=(N-1)t_k$, де N – число абонентів в мережі, t_k – час проходження кадру по кільцю. Маркерний метод доступу значно більш ефективніший, ніж випадкові методи, при великій інтенсивності обміну в мережі – при завантаженості понад 30-40%. Він забезпечує можливість роботи з великим навантаженням,

яке теоретично може досягати 100%.

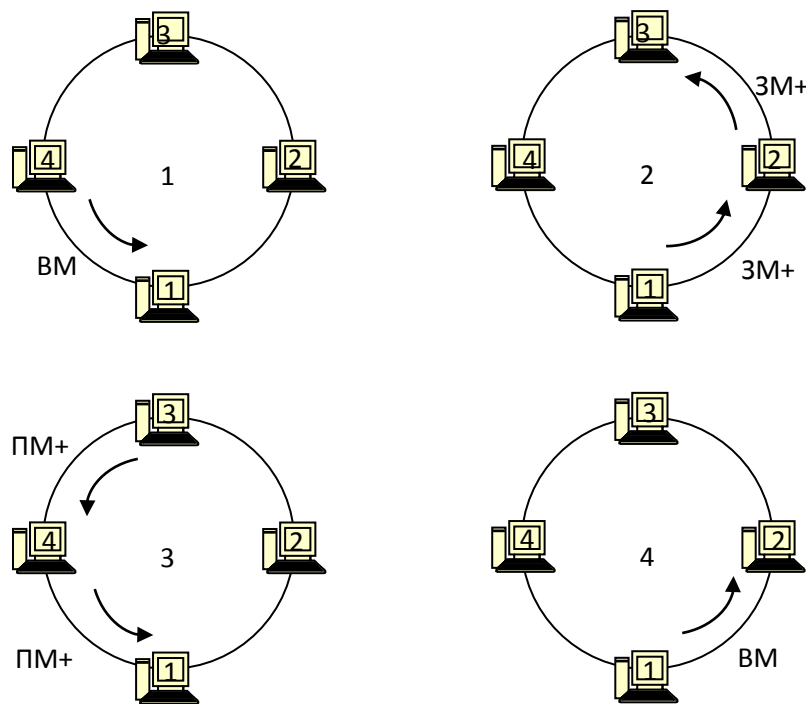


Рис. 11.7. Алгоритм маркерного метода доступу

11.4. Метод доступу DPP

Метод доступу DPP (Demand Priority Protocol) – метод пріоритетних запитів, або метод пріоритетного доступу на вимогу застосовується в мережах з зірковою топологією і відноситься до централізованих сховища.

Сутність методу полягає у передачі центральному вузлу функції арбітра мережі, який визначає порядок доступу до поділюваного середовища, як показано на рис. 11.8. В якості центрального вузла може виступати комп'ютер (центральный абонент), або концентратор. Концентратор – багатопортовий повторювач: сигнали, що надходять на один з портів, дублюються на всі інші порти.

В методі пріоритетного доступу на вимогу центральный вузол циклічно опитує свої порти. Якщо периферійному вузлу необхідно передати дані, він передає на порт центрального вузла спеціальний сигнал та повідомляє

пріоритет кадру, який збирається передати.

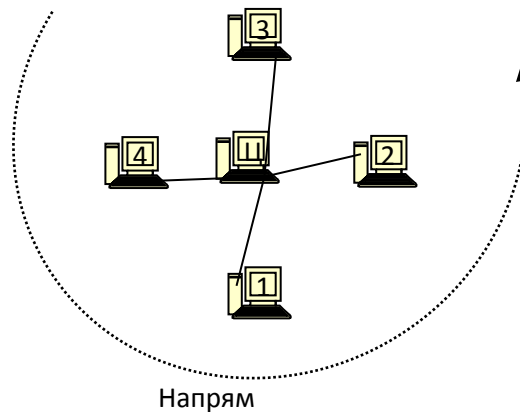


Рис. 11.8. Алгоритм метода пріоритетного доступу на вимогу

Високий пріоритет відповідає даним, що чутливі до часових затримок (аудіо, відео, дані прикладних додатків, що працюють в режимі реального часу). Дані, для яких фактор часу менш вагомий, отримують низький пріоритет. Крім того, враховується частота отримання периферійними вузлами доступу до середовища передачі: якщо вузол протягом тривалого часу не отримував дозволу на передачу, пріоритет його кадрів зростає.

Якщо середовище передачі вільне, центральний вузол дозволяє передачу і, отримавши від периферійного вузла кадр, пересилає його за адресою призначення. Якщо середовище передачі є зайнятим, то заявка на передачу даних ставиться в чергу і обробляється відповідно до порядку надходження та пріоритетів кадрів. Пріоритети кадрів абонентів можуть визначатись за їх фізичним розташуванням: в кожний момент часу найвищий пріоритет отримує наступний за розташуванням абонент.

Метод гарантує, що жодному абоненту не доведеться чекати своєї черги надто довго. Максимальний час доступу для будь-якого абонента буде дорівнювати сумарному часу передачі кадрів всіх абонентів мережі, крім даного.

Ніяких колізій в даному методі бути не може, оскільки рішення про

надання доступу приймається центральним вузлом (якому ні з ким конфліктувати). Якщо всі абоненти дуже активні, то всі вони передаватимуть по черзі, а центральний вузол має бути надзвичайно надійним.

Швидкість управління в методі є невисокою, адже навіть коли передає дані лише один абонент, після передачі кожного кадру він має очікувати, поки центральний вузол не завершить опитування всіх інших абонентів.

Тестові завдання

Встановити кількість доменів колізії для топології мережі, що показана нижче.

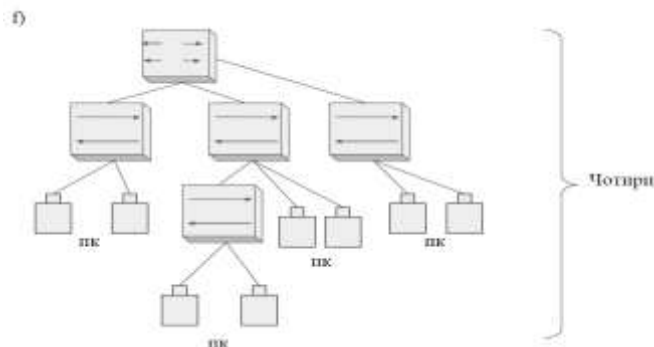


Рис. 11.9. Варіант завдання 1

Наприклад: Скільки доменів колізії?

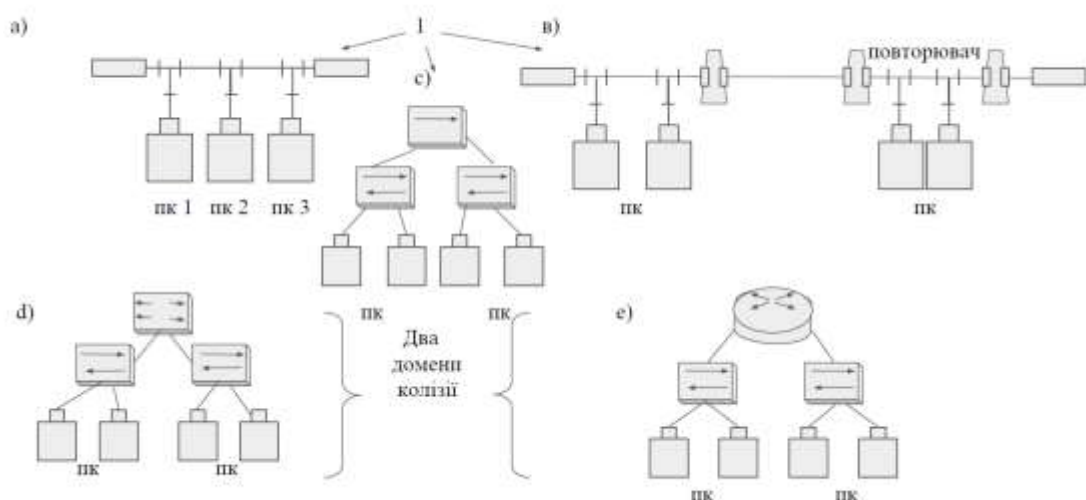


Рис. 11.10. Варіант завдання 2

Вищевказані завдання виконуються самостійно, відповідно до варіанту.

ЧАСТИНА III. АПАРАТНЕ І ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ МЕРЕЖ

12. ОСНОВНІ КОМПОНЕНТИ ЛОКАЛЬНОЇ МЕРЕЖІ

12.1. Мережеві адаптери

Для забезпечення зв'язку персонального комп'ютера (ПК) з іншими пристроями використовується мережевий адаптер.

Мережевий адаптер (Network Interface Card, NIC) – це периферійний пристрій, що безпосередньо взаємодіє з середовищем передачі даних.

Мережевий адаптер виконує функції фізичного і канального рівнів моделі OSI.

Основні функції мережевих адаптерів:

- реалізація методу доступу до середовища передачі даних,
- кодування і декодування даних,
- впізнавання кадрів, що приймаються,
- буферизація даних,
- гальванічна розв'язка комп'ютера і кабелю мережі.

В залежності від технології побудови мережі адаптери поділяються на адаптери Ethernet, Token Ring, FDDI і т.д. Мережеві адаптери виконуються у вигляді окремої плати, що вставляється в системні слоти розширення системної шини комп'ютера.

Мережеві адаптери Ethernet і Fast Ethernet можуть з'єднуватись з комп'ютером через один з нижчеописаних стандартних інтерфейсів.

- Шина ISA (Industry Standart Architecture). Забезпечує обмін 8- і 16-розрядними даними з невисокою пропускною здатністю (до 16 Мбайт/с). Вимагає конфігурування адаптерів (вручну або за допомогою програми установки). Вважається морально застарілою.

- Шина PCI (Peripheral Component Interconnect). Забезпечує обмін 32- і 64-розрядними даними і відрізняється високою пропускнуою здатністю (до 264 Мбайт/с). Підтримує технологію Plug-and-Play. Практично витіснила шину ISA і стала основною.
- Шина PC CARD (вона ж PCMCIA). Застосовується лише в ноутбуках. Більшість ноутбуків оснащують вбудованим мережевим адаптером, який з'єднується з шиною PCI. Передбачає просте підключення міні-плат розширення і високу швидкість обміну даними з ними.

Часто мережеві адаптери вбудовують в системні плати, що спрощує процес підключення комп'ютера до мережі.

Мережевий адаптер розраховується на один тип середовища передачі (наприклад – вита пара), але може підтримувати і кілька середовищ (наприклад – тонкий/товстий коаксіальний кабель). Для цього на платі адаптера встановлюють відповідні роз'єми. Найбільш універсальними є так звані адаптери «Combo», які мають повний набір роз'ємів (BNC, RJ-45 і AUI).

Головним завданням мережевого адаптеру є прийом та передача даних. Ця функція поділяється між адаптером і його драйвером. В одних моделях адаптерів (як правило, встановлюються на клієнтські машини) більша частина роботи з даними передається драйверу адаптера – сам адаптер спрощується і здешевлюється, але зростає навантаження центрального процесора. В інших, більш складних моделях, які зазвичай встановлюють на серверні машини, використовується власний процесор, який самостійно виконує основну обчислювальну роботу. На рис. 12.1 показана топологія мережі, що об'єднує три ПК з NIC через концентратор (Hub), робота якого буде розглянута нижче. Принцип підключення мережевого адаптера до концентратора полягає в наступному. Кожен з мережевих адаптерів NIC ПК має два виходи для передачі та прийому сигналу з використанням кабелю, наприклад – виті пари. Передавач сигналу від ПК, що розміщений на NIC,

носить назву трансмітера і позначається Tx, а приймач сигналу носить назву ресивера і позначається Rx. Відповідно трансмітер Tx від ПК1 з'єднується кабелем з приймачем Концентратора Rx, а трансмітер Концентратора Tx з'єднується з ресивером Rx ПК1, що розміщений на NIC. Передача сигналів між ПК1-ПК3 в Концентраторі зображена стрілками.

Слід зауважити, що швидкість обміну даними по мережі є інтегральною характеристикою, що визначається не лише адаптером, а й рядом інших факторів – швидкодією процесора, диску, об'ємом оперативної пам'яті, рівнем перешкод та завантаженості лінії, програмним забезпеченням та іншим.

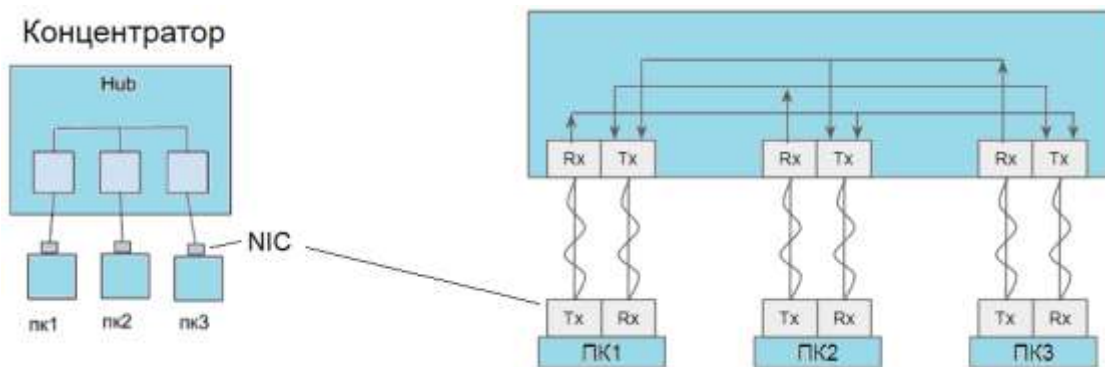


Рис. 12.1. Зображення концентратора

Тому вибір швидкого адаптера далеко не завжди гарантує помітний виграш в швидкості обміну. Отримати реальні кількісні показники продуктивності можна лише в результаті тестування всієї мережі за допомогою тестових програм (Netbench, nGrinder) і порівняння їх результатів.

12.2. Повторювачі

Повторювач, або репітер (англ. repeater) – комунікаційний пристрій, що відновлює послаблені сигнали (їх амплітуду і форму) і ретранслює їх в інший сегмент мережі з метою збільшення радіусу мережі.

Повторювачі не здійснюють ніякої інформаційної обробки сигналів, що

надходять до них.

Повторювачі використовували в мережах Ethernet на основі коаксіального кабелю, як показано на рис. 12.2. Повторювачі об'єднували в спільну шину кілька відрізків кабелю, до яких підключались абонентські станції.

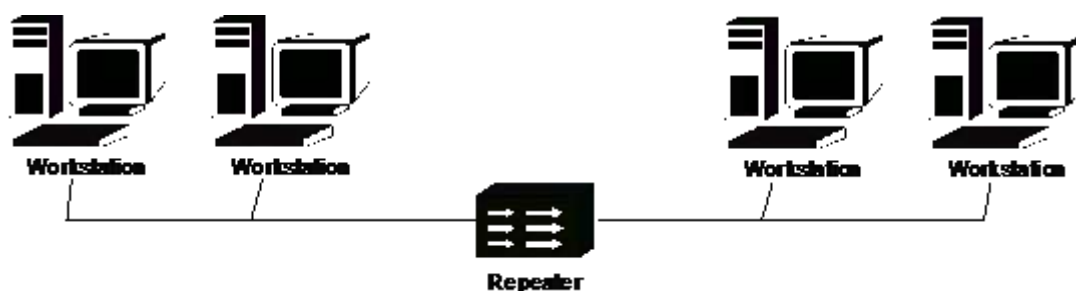


Рис. 12.2. Мережа на базі повторювача

Повторювачі працюють на фізичному рівні моделі OSI. Кілька повторювачів (репітерів) можуть конструктивно об'єднуватися в спільному корпусі, утворюючи репітерний концентратор, або концентратор. Перевага такого об'єднання полягає в тому, що всі точки виявляються зібраними в одному місці, що спрощує реконфігурацію мережі, контроль і пошук несправності.

12.3. Концентратори

Концентратор, або хаб (від англ. hub) – спеціальний багатопортовий пристрій, основна функція якого полягає у повторі кадру з одного з портів на інші. Іноді концентратори називають також репітерними концентраторами, щоб відрізнити їх від комутуючих концентраторів – комутаторів. Тому, репітери можна розглядати як двопортові концентратори. На рис. 12.3 показано зразок локальної мережі на базі концентратора.

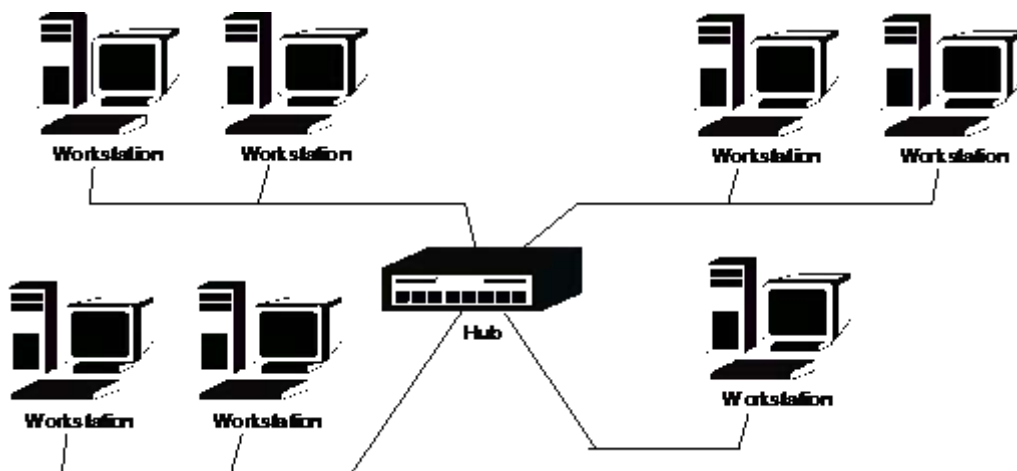


Рис. 12.3. Мережа на базі концентратора

Концентратори виконують ті ж функції, що й репітери, та працюють на фізичному рівні моделі OSI. Для кожної мережевої технології (Ethernet, Token Ring, FDDI і т.д) використовуються свої концентратори, призначені для роботи саме з нею.

За конструкцією розрізняють концентратори наступних видів:

- концентратори з фіксованою кількістю портів – виконуються у вигляді окремого корпусу з фіксованою кількістю портів (наприклад, на 8, 16, 24 порти).
- Стекові, або нарощувані концентратори – виконуються у вигляді окремого корпусу з фіксованою кількістю портів, які мають спеціальні порти для об'єднання їх внутрішніх шин між собою, як показано на рис. 12.4. Внаслідок цього швидкість взаємодії між об'єднаними концентраторами є значно вищою, ніж при з'єднанні через звичайний порт.
- Модульні концентратори на основі шасі – мають спільне шасі з внутрішньою шиною, до якої підключаються модулі з фіксованою кількістю портів. Структуру модульного концентратора на основі шасі показано на рис. 12.5.

Модульні концентратори на основі шасі дозволяють шляхом перекомутації зв'язків будувати складні конфігурації мереж, але мають

високу вартість. Застосування таких концентраторів вважається економічно виправданим лише у випадку необхідності підтримки великої кількості портів (близько 100).

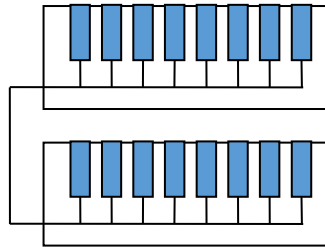


Рис. 12.4. Структура нарощуваного концентратора

Для мереж Ethernet стандарт IEEE 802.3 визначає два класи концентраторів:

- концентратори II класу повторюють сигнали, що надходять з одного сегмента, і передають їх в інші сегменти без будь-якого перетворення.
- Концентратори I класу перетворюють сигнали, що надходять з одного сегмента, в цифрову форму перед їх передачею в інші сегменти.

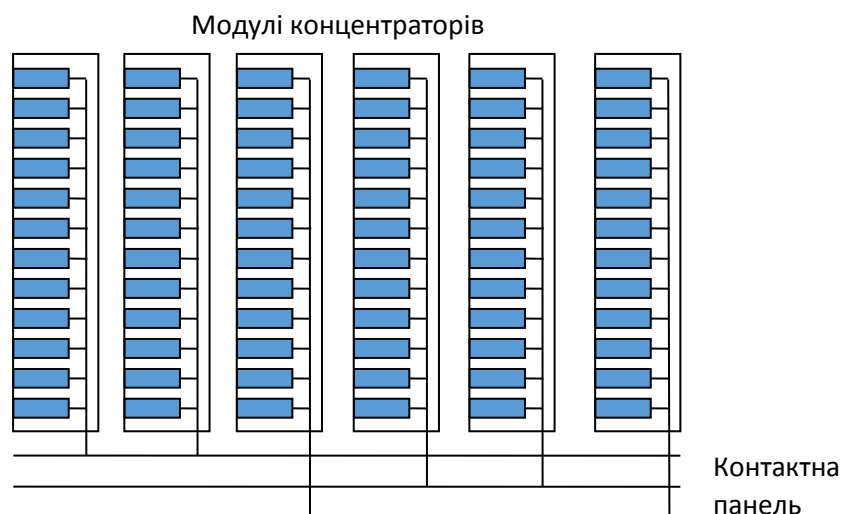


Рис. 12.5. Структуру модульного концентратора

Концентратори II класу не можуть перетворювати методи кодування

сигналів, тому до них можна підключати лише сегменти, що використовують однакову систему сигналів, наприклад, 100BASE-TX і 100BASE-FX, або однотипні, але не 100BASE-TX і 100BASE-T4.

Для з'єднання концентраторів II класу між собою використовується спеціальний порт розширення (UpLink port). Кожний концентратор підключається через цей порт до одного зі звичайних портів іншого концентратора, як показано на рис. 12.6.

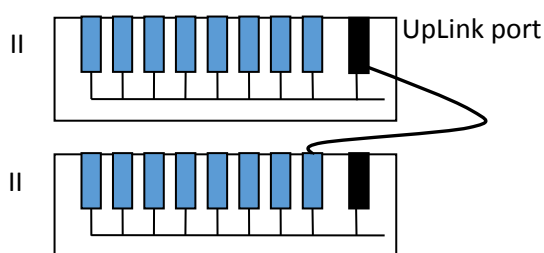


Рис. 12.6. З'єднання концентраторів II класу через UpLink port

Затримка сигналів в концентраторах II класу менше, ніж в концентраторах класу I, і це дозволяє використовувати довші кабелі. Проте забезпечення малої затримки накладає обмеження на нарощуваність і на кількість портів (не більше 24) таких концентраторів, вони є складнішими у виробництві. Тому концентратори II класу поступово витісняються концентраторами I класу.

На відміну від концентраторів II класу, концентратори I класу здатні перетворювати коди, тому до них можна підключати сегменти різних типів, наприклад, 100BASE-TX, 100BASE-TX і 100BASE-FX. Проте, процес перетворення вимагає часу, тому такі концентратори є повільніші та мають більшу затримку сигналів.

Концентратори I класу мають ширші можливості для нарощування, і тому саме вони входять до складу концентраторів на базі шасі.

Завдяки внутрішнім цифровим шинам сигналів концентратори I класу

допускають віддалене управління з управляючої станції: контроль за навантаженням мережі, станом портів, інтенсивністю помилок та ін. Такий концентратор, що допускає віддалене управління, називається інтелектуальним хабом (intelligent hub). При цьому для обміну даними з управляючою станцією використовується спеціально розроблений протокол SNMP (Simple Network Management Protocol – простий мережевий протокол керування).

Протокол SNMP (RFC 1067, RFC 1098, RFC 1157) відноситься до прикладного рівня моделі OSI і може працювати з протоколами IP і IPX. Протокол збирає інформацію про мережу та управляє пристроями мережі.

Збір інформації передбачає зберігання об'єктів даних про пристрої мережі в текстових файлах у форматі ASN.1, які називаються MIB (Management Information Base – база керуючої інформації). Існує ряд фірмових і стандартних форматів MIB для мережевих адаптерів, концентраторів, мостів і мережі в цілому. Наприклад, MIB концентратора може містити дані про кількість кадрів, отриманих кожним з портів.

Для управління пристроєм його контролер має виконувати програму агента SNMP, яка збирає дані про пристрій і керує його об'єктами даних з баз керуючої інформації MIB.

Керуюча станція, яку називають також NMS (Network Management Station – станція керування мережею) – це один з комп'ютерів, підключених до мережі, на якому запущений спеціальний пакет прикладних програм для відображення стану мережевих пристроїв і управління ними.

12.4. Мости

При досягнення деякого граничного значення числа вузлів мережі, тривалість затримок перед отриманням ними доступу до поділюваного середовища зростає і пропускна здатність мережі стрімко починає падати. Для вирішення цієї проблеми мережа розбивається на кілька сегментів, що

об'єднуються за допомогою мостів.

Міст (англ. bridge) – комунікаційний пристрій, призначений для об'єднання мереж з різними стандартами обміну (наприклад, Ethernet і Token Ring), або кількох сегментів однієї мережі (наприклад, Ethernet).

Міст ретранслює кадри з однієї мережі до іншої, або з одного сегмента до іншого (як повторювач), але аналізує адресу їх призначення. Тобто, кадр транслюється в іншу мережу, або сегмент, лише в тому випадку, коли в цій мережі, або сегменті, знаходиться адресат. В результаті – за допомогою мостів мережа поділяється на кілька підмереж, якими розподіляється комп'ютерний трафік, внаслідок цього зменшується завантаженість середовища передачі даних. Приклад архітектури мережі на базі моста показано на рис. 12.7.

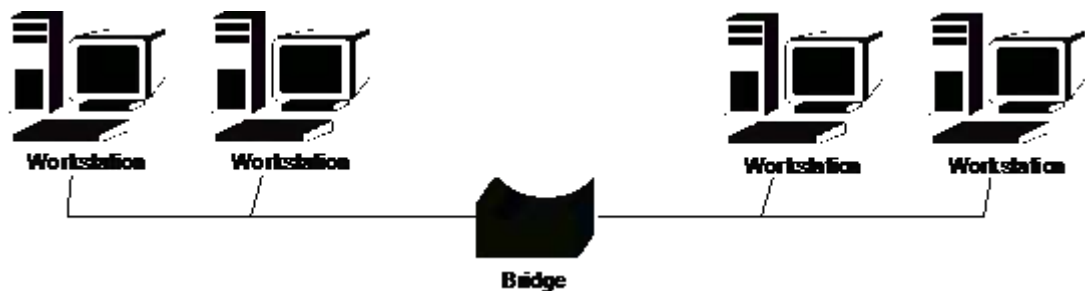


Рис. 12.7. Архітектура мережі на базі моста

В основі роботи мостів лежить принцип прозорості. Він означає, що мережеві адаптери не здійснюють будь-яких додаткових зусиль для пересилки своїх кадрів, вони «не бачать» міст. Досягається така прозорість за рахунок того, що міст будує особливу адресну таблицю, на основі якої і приймає рішення про необхідність ретрансляції кадрів.

Механізм реалізації принципу прозорості моста полягає у наступному. Міст приймає всі кадри, що передаються по мережі, і записує їх до свого буферу, з якого вони надходять на подальшу обробку. Обробка кадрів здійснюється послідовно по мірі їх надходження. При цьому аналізуються

адреса відправника і адреса одержувача. Якщо вони:

- містяться в адресній таблиці і належать різним сегментам – здійснюється ретрансляція кадру.
- Містяться в адресній таблиці і знаходяться в одному сегменті – кадр видаляється з буферу і нікуди не ретранслюється.
- Не містяться в адресній таблиці – кадр ретранслюється до всіх сегментів, крім того, з якого він надійшов, а незнайомі адреси додаються до адресної таблиці.

Таким чином мости «самонавчаються», дізнаючись про розташування вузлів в сегментах. Потім міст передає кадри лише в місце призначення, зменшуючи загальний обсяг даних, що передаються через мережу.

Мости працюють на каналному рівні моделі OSI, включаючи його верхній підрівень LLC (для зв'язування різнорідних мереж). Як правило, мости підтримують не більше 4 портів.

На даний час мости витісняють комутатори, які є більш функціональними.

12.5. Комутатори

Комутатор (комутуючий концентратор) або свіч (від англ. switch) – багатопортовий комунікаційний пристрій, який дозволяє об'єднувати кілька сегментів в одну мережу, забезпечуючи її високу продуктивність і пропускну здатність.

Комутатор може розглядатись як дуже швидкісний міст. Він дозволяє розділити мережу на кілька підмереж для збільшення допустимого радіусу мережі і зниження навантаження в її частинах.

На відміну від мостів, комутатори здійснюють не послідовну, а паралельну обробку кадрів, просуваючи їх одночасно між усіма парами своїх портів.

Ще однією відмінністю комутаторів, на відміну від мостів, є те, що

вони не приймають, а в реальному часі розпізнають адресу приймача і перенаправляють кадри з одної частини мережі до іншої. При цьому ніякої обробки кадрів не виконується, тому комутатори практично не зменшують швидкість обміну в мережі. Проте вони не можуть перетворювати формат кадрів.

Комутатори не ретранслюють колізії, на відміну від концентраторів. Також комутатори ведуть таблицю комутації, що ставить у відповідність порт комутатора та адресу відправника. Таблиця комутації формується в асоціативній пам'яті комутатора коли приходить пакет, що містить MAC - адресу відправника. В початковий момент таблиця комутації порожня, а розсилка пакета дублюється всім ПК в мережі. Якщо комутатор знає з таблиці адресу, то відсилає пакет за адресом і це скорочує час доставки пакетів.

Логічна структура комутатора показана на рис. 12.8. Вона містить перехресну матрицю (crossbar matrix), у точках перетину якої можуть встановлюватись зв'язки на час передачі кадру. В результаті кадр, що надходить з будь-якого сегмента, може бути переданий в будь-який інший.

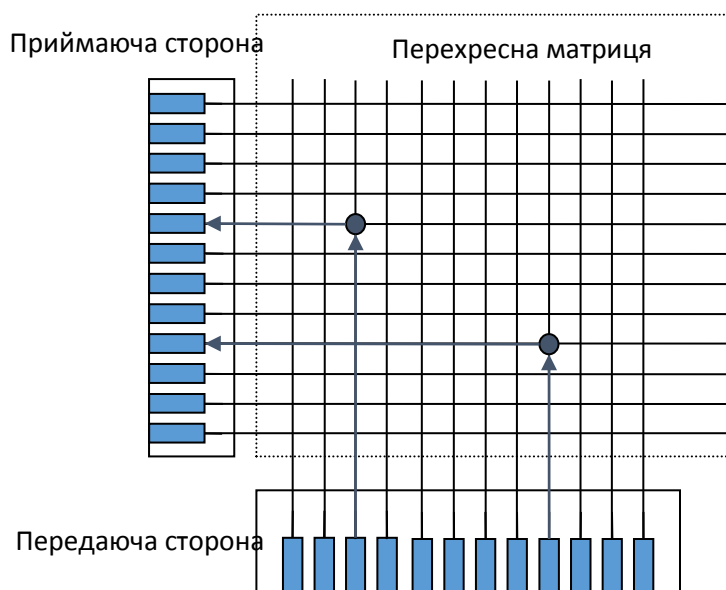


Рис. 12.8. Логічна структура комутатора

Найчастіше комутатори випускають на 6, 8, 12, 16 і 24 порти. При розбиванні мережі на частини за допомогою комутатора рекомендується дотримуватись правила «80/20»: для ефективної роботи комутатор необхідно, щоб 80% всіх передач здійснювалось в межах однієї частини (сегмента), і 20% всіх передач має здійснюватися між різними частинами (проходили через комутатор).

На практиці це правило найчастіше зводиться до того, що сервер і працюючі з ним станції (клієнти) розташовуються в одному сегменті.

12.6. Маршрутизатори

Маршрутизатор (англ. Router) - мережевий пристрій, що пересилає пакети даних між комп'ютерними мережами. Маршрутизатори виконують функції керування трафіком в Інтернеті. Пакет даних, як правило, пересилається з одного маршрутизатора на інший маршрутизатор через мережі, які складають мережу Інтернет, доки він не досягне свого кінцевого вузла. Схематичне зображення маршрутизатора показано на рис. 12.9.

Маршрутизатор підключається до двох, або більше, ліній зв'язку з різних мереж. Коли пакет даних надходить на маршрутизатор, він зчитує інформацію про IP-адресу в пакеті, щоб визначити кінцевий пункт призначення. Потім, використовуючи інформацію у таблиці маршрутизації, або політику маршрутизації, направляє пакет до наступної мережі.



Рис. 12.9. Зображення маршрутизатора

Найбільш знайомий тип маршрутизаторів – це домашні та невеликі

офісні маршрутизатори, які передають IP-пакети між домашніми комп'ютерами та Інтернетом. Прикладом маршрутизатора може стати власний кабель, або DSL маршрутизатор, який підключається до Інтернету через постачальника послуг Інтернету (ISP). Більш складні маршрутизатори, наприклад корпоративні, підключають великі мережі, або мережі Інтернет до потужних маршрутизаторів, які передають дані на високій швидкості до волоконно-оптичних магістралей. Хоча маршрутизатори, як правило, призначені для апаратних пристроїв, також існують маршрутизатори на базі програмного забезпечення.

Контрольні запитання до розділу

1. Основні компоненти локальної мережі, мережевий адаптер.
2. Повторювач, мережа на базі повторювача.
3. Концентратори, види та класи концентраторів.
4. Мости, принцип прозорості моста.
5. Комутатори, їх відмінність від концентраторів. Маршрутизатори.

13. МЕРЕЖЕВІ ОПЕРАЦІЙНІ СИСТЕМИ

13.1. Поняття мережевої ОС

Мережева операційна система – це операційна система (ОС) окремого комп'ютера, яка містить засоби для роботи користувача в комп'ютерній мережі.

Основними функціями мережевих ОС є:

- керування каталогами і файлами – полягає в забезпеченні доступу до даних, що фізично розташовані в різних вузлах мережі.
- Керування ресурсами – полягає в забезпеченні запитів на надання ресурсів, що надані у спільне використання.
- Комунікаційні функції – забезпечують адресацію, буферизацію, маршрутизацію даних.
- Захист від несанкціонованого доступу – полягає в забезпеченні визначених умов доступу до ресурсів (з окремих станцій, у визначений час, визначену кількість разів і т.д.).
- Забезпечення відмовостійкості – полягає в забезпеченні працездатності системи при виникненні дестабілізуючих факторів (через використання автономних джерел живлення, дублювання даних на дискових накопичувачах і т.д.).
- Керування мережею – полягає у використанні відповідних протоколів керування, які забезпечують збір даних про параметри функціонування мережі, фіксують виникнення аномальних значень параметрів, здійснюють антивірусний захист і т.д.

Найпоширенішими на сьогодні є три основні мережеві ОС – UNIX, Windows NT і Novell Netware.

13.2. Функціональні компоненти мережевої ОС

В структурі мережевої ОС розрізняють наступні компоненти, показані на рис. 13.1

- Засоби керування локальними ресурсами – реалізують всі функції ОС автономного комп'ютера (інтерфейс користувача, розподіл оперативної пам'яті, керування зовнішньою пам'яттю та інше).

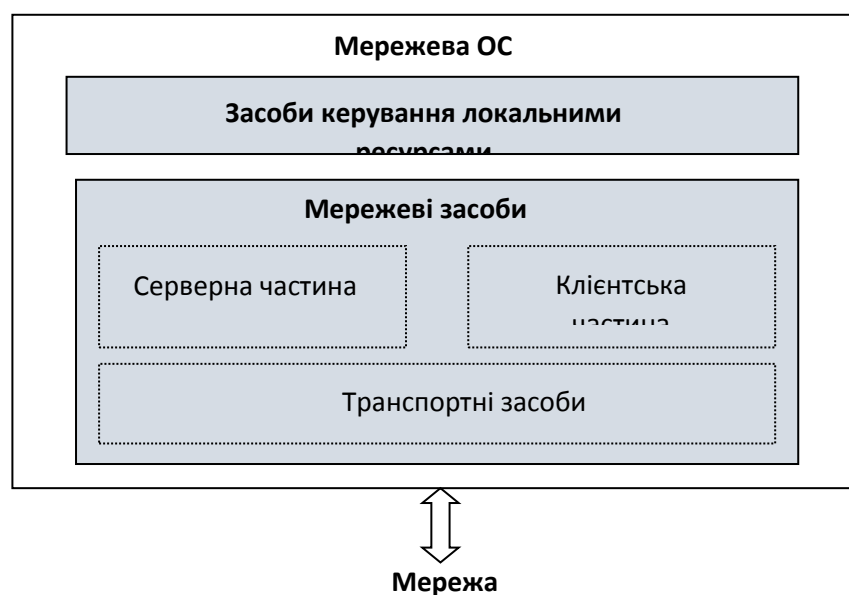


Рис. 13.1. Компоненти мережевої ОС

- Мережеві засоби – реалізують функції для роботи в мережі.
- Серверна частина мережевої ОС – мережеві засоби для надання локальних ресурсів і послуг у спільне використання.
- Клієнтська частина мережевої ОС – мережеві засоби для формування запитів на доступ до віддалених ресурсів і послуг.
- Транспортні засоби мережевої ОС – мережеві засоби, що забезпечують передачу і отримання повідомлень з комунікаційної системи.

В залежності від функцій, що покладаються на комп'ютер, в його ОС може бути відсутньою або клієнтська, або серверна частина.

Взаємодія компонентів мережевої ОС відбувається за схемою, що

наведена на рис. 13.2. Тут на комп'ютері-клієнті відсутня серверна частина ОС, а на комп'ютері-сервері – клієнтська частина ОС. Окремо показаний редіректор - компонент клієнтської частини, що перехоплює запити від додатків і переадресовує їх:

- до локальної ОС, якщо запитаний ресурс даного комп'ютера, або
- в мережу, якщо запитаний віддалений ресурс.

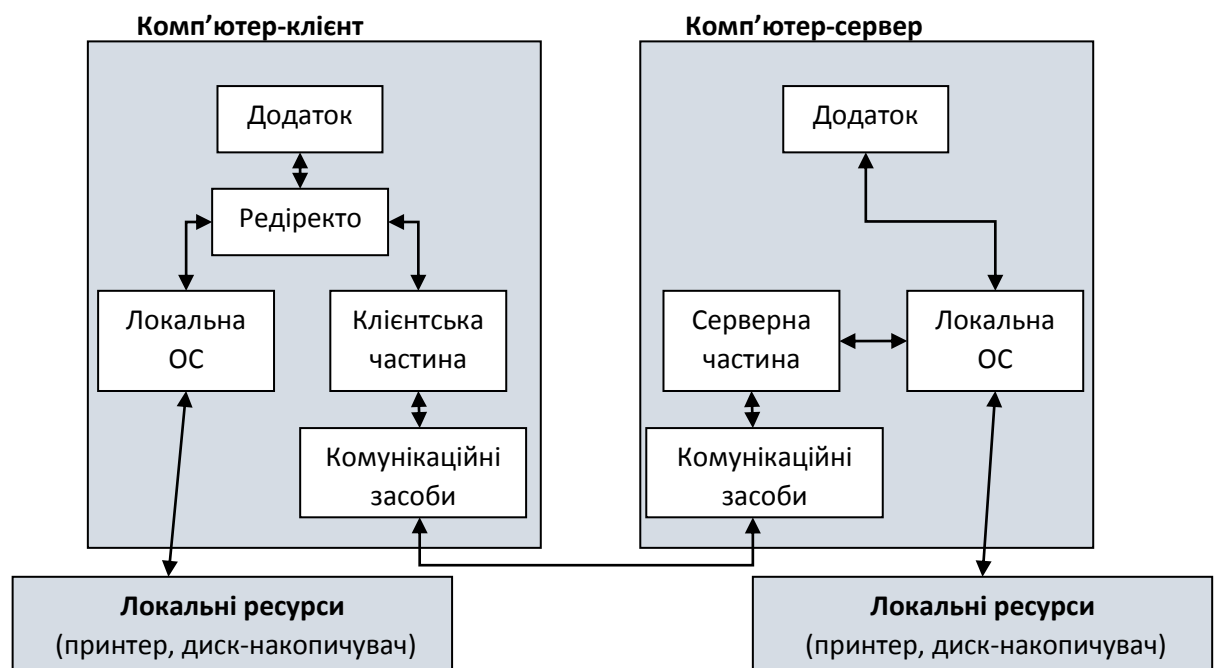


Рис. 13.2. Взаємодія компонентів мережевої ОС

Зручність використання редіректора полягає в тому, що прикладним додаткам не треба піклуватись про те, з локальними чи віддаленими ресурсами вони працюють – клієнтська частина ОС сама розпізнає і перенаправляє запити на віддалену станцію.

Припустимо, прикладному додатку на комп'ютері-клієнті необхідно звернутись до деякого ресурсу (наприклад, завантажити, або роздрукувати файл). Для цього він формує запит, що у випадку звернення до віддаленого ресурсу переадресовується редіректором до клієнтської частини ОС.

Клієнтська частина ОС на комп'ютері-клієнті не може отримати

безпосередній доступ до ресурсів іншого комп'ютера. Тому вона направляє до серверної частини ОС на комп'ютері-сервері повідомлення виконати необхідні дії. Такі повідомлення можуть містити не лише команди на виконання дій, але й самі дані (наприклад, файл).

При цьому клієнтська частина ОС перетворює запит з локальної форми (що прийнята в локальній частині ОС на комп'ютері-клієнті) в мережеву (що відповідає вимогам серверної частини ОС на комп'ютері-сервері) і передає його транспортній системі, яка відповідає за доставку повідомлень серверу.

Транспортні засоби ОС здійснюють керування передачею повідомлень між клієнтською та серверною частинами ОС через комунікаційну систему. Вони, зокрема, забезпечують виконання таких функцій, як формування повідомлень, розбиття повідомлень на пакети, організацію надійної доставки пакетів та інше.

Правила взаємодії комп'ютерів при обміні повідомленнями через мережу визначаються в комунікаційних протоколах. Комунікаційні протоколи передають повідомлення клієнтських і серверних частин ОС через мережу, не втручаючись в їх зміст.

На комп'ютері-сервері, що містить поділювані ресурси, має працювати серверна частина ОС. Вона постійно очікує запити з мережі, приймає їх, перетворює в локальну форму і передає на виконання своїй локальній ОС. Після отримання результату серверна частина ОС звертається до транспортної системи і направляє відповідь на комп'ютер-клієнт.

Клієнтська частина ОС на комп'ютері-клієнті перетворює результат у локальний формат і адресує його тому прикладному додатку, який видав запит.

13.3. Мережеві служби і мережеві сервіси

Мережева служба – сукупність серверної і клієнтської частин ОС, які надають доступ до певного типу ресурсу, або сервісу, через мережу.

Мережевий сервіс – набір послуг, який надає користувачу мережева служба. Наприклад, служба друку забезпечує доступ до принтерів і надає сервіс друкування. Найбільш важливими для користувачів є файлова служба і сервіс друкування.

Мережева служба може бути представлена в ОС обома (серверною і клієнтською) частинами, або лише однією з них.

Слід зазначити, що для надання мережевою службою деякого мережевого сервісу використовуються ресурси (процесорний час, дисковий простір і т.п.) не лише сервера, а й клієнта – він теж може витрачати частину своїх ресурсів на підтримку роботи.

Як правило, взаємодія між клієнтською і серверною частинами ОС стандартизується, так що один тип сервера може використовуватись різними типами клієнтів. Єдиною умовою для цього є те, що клієнти і сервери мають підтримувати спільний набір комунікаційних протоколів.

На практиці склались кілька підходів до побудови мережевих ОС, що відрізняються глибиною інтеграції мережевих служб в ОС:

- створення мережевої оболонки над вже існуючою локальною ОС,
- інтеграція мережевих служб в локальну ОС,
- створення окремих програмних продуктів для реалізації мережевих служб.

Перший підхід, що показано на рис. 13.3, використовувався на ранніх етапах побудови мережевих ОС. Мережеві ОС тоді представляли собою локальну ОС з надбудованою на нею мережевою оболонкою для виконання основних мережевих сервісів. Одна мережева оболонка може призначатись для роботи з різними операційними системами.

Мережеві оболонки часто поділяють на:

- клієнтські – містять клієнтські частини мережевих служб;
- серверні – містять серверні частини мережевих служб, як мінімум.

Типовим прикладом мережевої оболонки є програмне забезпечення мереж для ОС NetWare де в якості локальної використовується ОС MS-DOS.

Серверна оболонка містить серверні компоненти двох основних служб – файлової і служби друку. Клієнтська оболонка складається з клієнтських компонентів цих служб, а також компоненти, що підтримує інтерфейс користувача.

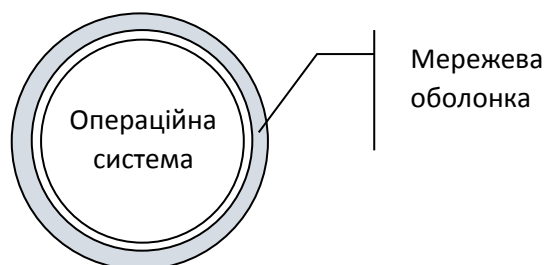


Рис. 13.3. Структура мережевої оболонки над існуючою локальною ОС

Проте згодом розробниками мережесистем більш ефективним був визнаний другий підхід, при якому ОС з самого початку проектується спеціально для роботи в мережі. В цьому випадку всі внутрішні механізми ОС можуть бути оптимізованими для виконання мережесистемних функцій, як показано на рис. 13.4. Наприклад, за рахунок інтегрованих мережесистемних засобів в ОС Windows NT вдалось забезпечити більш високі показники продуктивності і захищеності інформації, ніж при використанні мережесистемних оболонок (наприклад, оболонки LAN Manager для OS/2).

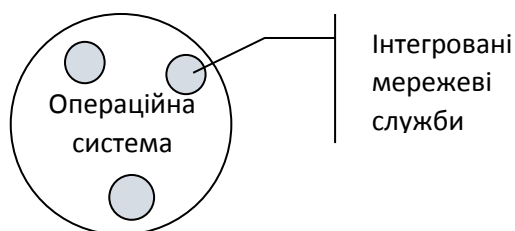


Рис. 13.4. Структура ОС з інтегрованими мережесистемними засобами

Третій спосіб реалізації мережесистемної служби (у вигляді окремого мережесистемного додатку, як показано на рис. 13.5) дозволяє стороннім

розробникам долучитись до розробки мережеских служб і забезпечити їх більш широку функціональність. Наприклад, сервер віддаленого керування WinFrame від компанії Citrix доповнює можливості вбудованого сервера віддаленого доступу RAS (Remote Access Server), вбудованого у Windows NT.

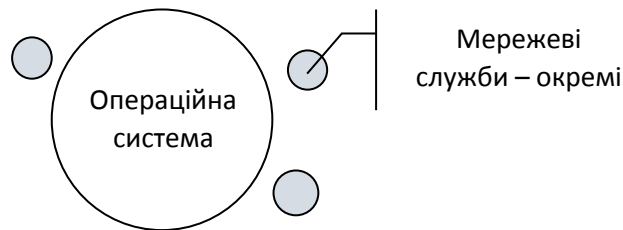


Рис. 13.5. Структура ОС з окремими мережескими додатками

13.4. Однорангові і серверні мережескі ОС

В залежності від компонентного складу мережеских ОС, що використовуються вузлами комп'ютерної мережі, можливою є одна з трьох можливих схем її побудови:

- однорангова мережа,
- мережа з виділеними серверами,
- гібридна мережа.

В однорангових мережах на комп'ютерах встановлюється така ОС, яка надає всім вузлам мережі потенційно рівні можливості. Очевидно, що ОС в одноранговій мережі повинна включати як серверні, так і клієнтські частини мережеских служб, як показано на рис. 13.6.

Проте, можливе виникнення функціональної несиметричності:

- на окремих комп'ютерах серверні частини не активізуються і вони виконують роль лише клієнтів,
за окремими комп'ютерами закріплюються лише функції з обслуговування запитів і вони виконують роль лише серверів.

Зміна ролі комп'ютера в одноранговій мережі досягається за рахунок того, що функції серверної, або клієнтської частини не використовуються.

Однорангові мережі прості в організації і експлуатації. За такою схемою організують невеликі мережі з кількістю комп'ютерів 10-20 шт., оскільки в цьому випадку користувачам нескладно узгодити домовленості про використання спільних ресурсів і необхідності в централізованих засобах адміністрування не виникає.

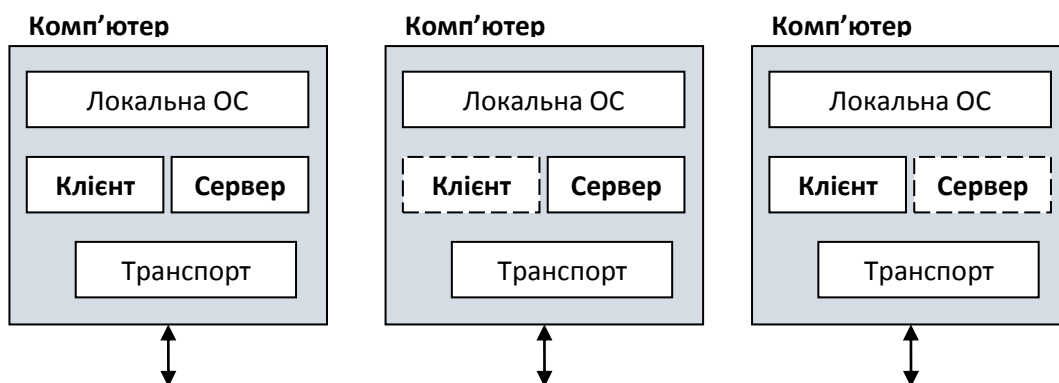


Рис. 13.6. Структура ОС для однорангової комп'ютерної мережі

Прикладами однорангових ОС можуть бути Windows for Workgroups, Windows 95/98, Windows NT Workstation.

В мережі з виділеними серверами, що показана на рис. 13.7, використовуються спеціальні варіанти мережових ОС, що оптимізовані для роботи в ролі серверів. Такі ОС називаються серверними ОС, на відміну від клієнтських ОС, що встановлюють на комп'ютери користувачів.

Спеціалізація ОС для роботи в якості сервера є основним способом підвищення ефективності серверних операцій. Чим менше функцій виконує ОС, тим більш ефективно їх можна реалізувати. Тому для оптимізації виконання серверних операцій розробники змушені відмовлятися від деяких інших функцій, іноді аж до повного їх відкидання.

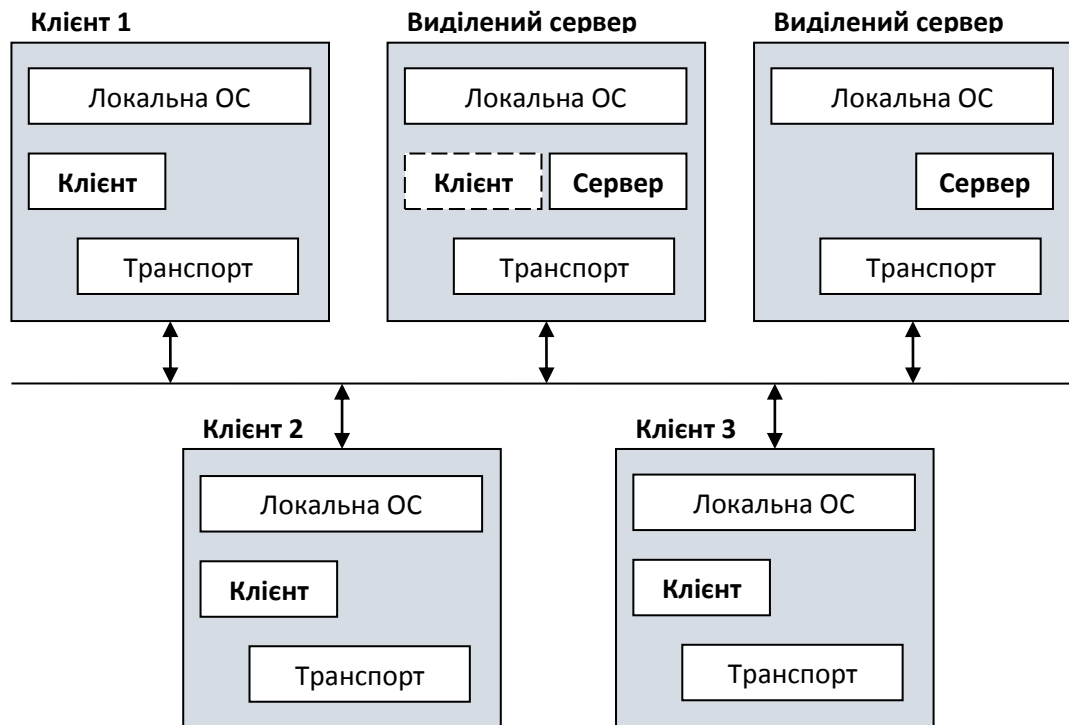


Рис. 13.7. Структура ОС для мережі з виділеними серверами

Одним з яскравих прикладів такого підходу є серверна ОС NetWare. З метою оптимізації роботи файлової служби і служби друку розробники виключили з системи інші елементи, що є важливими для універсальної ОС: графічний інтерфейс користувача, засоби захисту додатків мультипрограми режиму один від одного, механізм віртуальної пам'яті тощо. Це дозволило досягти унікальної швидкості файлового доступу і друку та вивело ОС NetWare в лідери на довгий час. Проте вузька спеціалізація серверної ОС є одночасно і її слабкою стороною: відсутність графічного інтерфейсу і засобів захисту додатків не дозволяє використовувати серверну ОС NetWare для виконання додатків, а отже потребує включення до мережі інших серверних ОС для виконання функцій, відмінних від файлового сервісу і сервісу друку.

Розробники сучасних серверних ОС відмовляються від внесення функціональних обмежень і включають до складу серверних всі компоненти, що необхідні для їх використання як в якості універсального сервера, так і в

якості клієнтської ОС.

Поєднати переваги мереж з виділеними серверами і однорангових мереж дозволяє використання гібридної мережі, що показана на рис. 13.8.

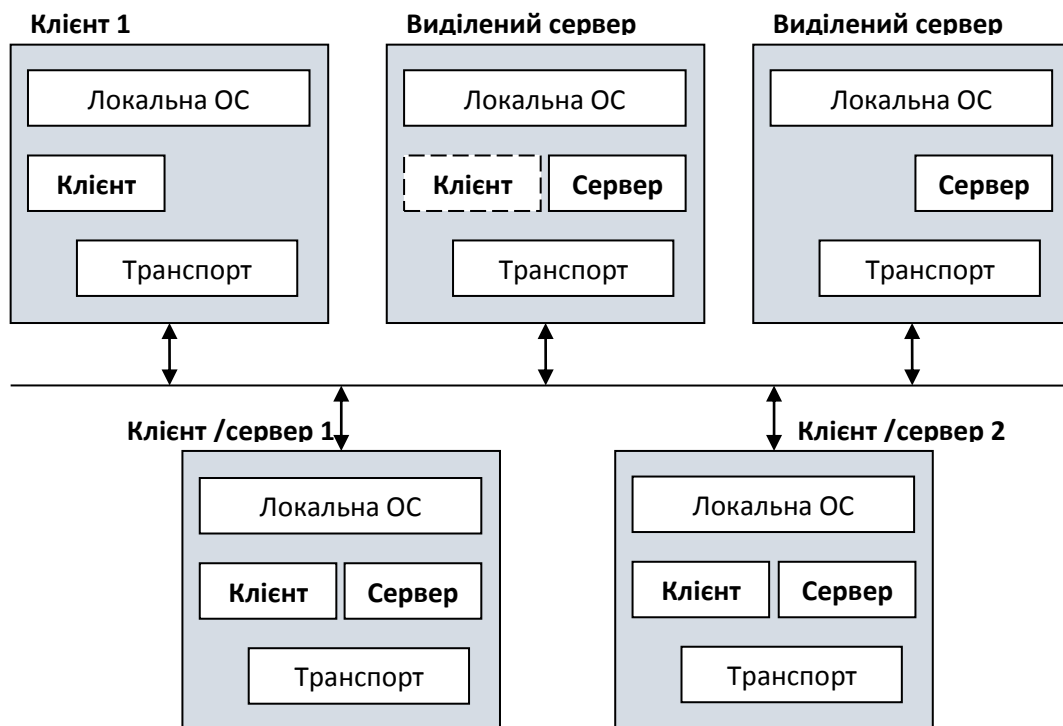


Рис. 13.8. Структура ОС для гібридної мережі

Часто випускають два варіанти однієї ОС: один для використання в якості серверної ОС, а другий – в якості клієнтської ОС. Вони відрізняються набором мережевих служб і утиліт, а також параметрами конфігурації, окремі з яких встановлюються за замовчуванням і не підлягають зміні. Так, ОС Windows NT випускається у двох варіантах: Windows NT Workstation в якості клієнтської ОС і Windows NT Server в якості серверної ОС.

13.5. Огляд відомих мережевих ОС

13.5.1. ОС Unix

Ця ОС є пропонує оптимальні рішення для роботи з Internet,

включаючи доступ до ресурсів Web, Telnet, FTP, баз даних і т.п. Оскільки UNIX створювалась спеціально для обробки великих обсягів даних і повної інтеграції з мережевим середовищем, то вона майже завжди переважає за швидкістю будь-яку іншу комбінацію апаратного і програмного забезпечення.

ОС UNIX розроблена в 1969 р. в AT&T Bell Labs. Протягом наступних 10 років розвиток UNIX відбувався, в основному, в Bell Labs. Відповідні початкові версії називалися "Version n" (Vn). Згодом до ОС UNIX долучились інші розробники.

Сьогодні для платформи Intel x86 доступні кілька версій UNIX:

- FreeBSD – добре зарекомендувала себе як система для побудови інтранет- і Інтернет-серверів. Вона надає достатньо надійні мережеві служби і ефективне керування пам'яттю. Крім своєї стабільності, FreeBSD популярна і завдяки своїй ліцензії: вона дозволяє використовувати код не лише у вільному програмному забезпеченні, а й у пропрієтарному.
- NetBSD – вільно поширювана, захищена, крос-платформна операційна система. Підтримуються 53 апаратні платформи. Перша офіційна версія NetBSD - 0.8 - була випущена в 1993 р., а поточна версія - NetBSD 8.0 - випущена в 2018 р. В кінці 1995 року від NetBSD відгалузилися проект OpenBSD.
- OpenBSD – основною відмінністю від інших вільних ОС (таких, як NetBSD, FreeBSD) є початкова орієнтованість проекту на створення найбільш безпечної, вільної та ліцензійно чистої з існуючих операційних систем. Нові версії (релізи) OpenBSD виходять кожні півроку.
- Linux – на відміну від більшості ОС, Linux не має єдиної «офіційної» комплектації. Ця ОС поставляється у великій кількості так званих дистрибутивів, в яких програми GNU з'єднуються з ядром Linux та іншими програмами. Найбільш відомими дистрибутивами Linux є Ubuntu, Debian, Red Hat, Fedora, Mandriva, SuSE, Gentoo, Slackware, Archlinux.

- Solaris – ОС, що розробляється Oracle Corporation (раніше Sun Microsystems). Незважаючи на те, що історично Solaris - операційна система з закритим вихідним кодом, більша частина її програмного коду відкрита і опублікована. Починаючи з 2005 р., пропонується для завантаження загальнодоступна (в бінарному вигляді, тобто із закритим вихідним кодом) некомерційна версія Solaris.

13.5.2. ОС NetWare

NetWare – мережева ОС, створена компанією Novell. Займала суттєву долю ринку (65-75%) на початку 90-х рр.

Основна ідея, що покладена в основу NetWare, полягає у наступному. Виділені сервери підключаються до мережі і надають у спільне використання свій дисковий простір (томи) і під'єднані до них принтери. На клієнтах запускаються резидентні програми, які дозволяють користувачам зареєструватись в мережі і отримати доступ до томів та виконати друкування на мережевих принтерах.

Для серверної частини ОС NetWare фірма Novell розробила спеціалізовану ОС, оптимізовану для виконання файлових операцій і друку на мережевих принтерах. Розплатою за її високу продуктивність стало те, що виділені сервери не можуть використовуватись в якості робочих станцій.

В якості клієнтських ОС Novell випускає дві власні ОС – Novell DOS 7 і UnixWare. Для популярних платформ інших виробників Novell випускає мережеві оболонки з клієнтськими функціями по відношенню до серверів NetWare.

В NetWare використовується протокол передачі пакетів NCP (NetWare Core Protocol – протокол ядра NetWare), який дозволяє клієнтам передавати запити до серверів NetWare і отримувати від них відповіді. Початково NCP був прив'язаний до протоколів IPX/SPX.

Перший програмний продукт NetWare вийшов у 1983 р. Наступними версіями були:

- NetWare 2.x – відсутність модульної структури робило складним керування системою.
- NetWare 3.x – за рахунок реалізації модульності з'явилась можливість додавання нових функцій в систему. NetWare v3.11 стала першою мережевою ОС, яка забезпечувала доступ до мережевих ресурсів з робочих станцій DOS, Windows, OS/2, UNIX и Macintosh.
- NetWare 4.x – розроблена спеціально для побудови мереж масштабу підприємства. Нововведенням стала служба каталогів NetWare Directory Services (NDS), яка забезпечувала прозорий доступ до всіх мережевих ресурсів багатосерверної мережі.
- NetWare 4.11 (intraNetWare) – з цією версією постачався перший повністю 32-розрядний клієнт для робочих станцій Microsoft Windows і графічна утиліта адміністрування NetWare Administrator (NWADMIN, або NWADMN32).
- NetWare 5.x – Novell визнала роль Інтернет і зробила для NCP підтримку стеку протоколів TCP/IP, а не IPX/SPX.
- NetWare 6.x – додано і удосконалено ряд компонентів системи. Остання версія – NetWare 6.5.

13.5.3. ОС Windows

Компанія Microsoft запропонувала кілька версій мережевої ОС Windows:

- Microsoft Windows NT – існувала у двох версіях: Windows NT Server встановлювалась на комп'ютерах-серверах мережі, а Windows NT Workstation встановлювалась на комп'ютерах-клієнтах мережі. Windows NT 4.0 поєднала в собі покращену інтеграцію з Internet і корпоративними мережами, підвищену продуктивність, відмінну сумісність з іншими ОС компанії Microsoft.
- Microsoft Windows 2000 Server стала наступним поколінням ОС Windows і існувала у таких версіях:

- Windows 2000 Server - для серверів робочих груп.
- Windows 2000 Advanced Server - для більш надійних серверів відділів.
- Windows 2000 Datacenter Server - для найбільш відповідальних систем обробки даних.
- Windows Server 2003 – стала результатом розвитку Windows 2000 Server і існувала у таких версіях:
 - Windows Server 2003 Standard Edition – для підприємств малого бізнесу и окремих підрозділів організації.
 - Windows Server 2003 Enterprise Edition – для підприємств середнього бізнесу.
 - Windows Server 2003 Datacenter Edition – для вирішення задач, що вимагають високого рівня масштабованості і надійності.
 - Windows Server 2003 Web Edition – для Web-серверів.
- Microsoft Windows Server 2008 — ОС нового покоління, в основу якої покладена Windows Server 2003. Призначена для забезпечення користувачів найбільш продуктивною платформою. При спільному використанні клієнтських комп'ютерів Windows Vista и серверів під Windows Server 2008 значно підвищується продуктивність і надійність мережі.
- Windows Server 2012 R2 - серверно-орієнтована система, випущена 2013 р. Мінімальні вимоги: процесор 1,4 ГГц, пам'ять 512 МВ, диск 32 ГВ.
- Windows Server 2016 – в систему додано цілий ряд нових можливостей: механізм оновлення ОС хостів кластера без його зупинки (Cluster Operating System Rolling Upgrade), синхронна реплікація сховищ на рівні блоків з підтримкою географічно розподілених кластерів, віртуальний мережевий контролер (software-defined networking stack) для одночасного управління

фізичними і віртуальними мережами, новий формат файлів конфігурації віртуальних машин (.VMCX і .VMRS), з більш високим ступенем захисту від збоїв на рівні сховища та ряд інших.

Контрольні запитання до розділу

1. Функції мережевих операційних систем.
2. Компоненти мережевої операційної системи, взаємодія компонентів.
3. Поняття мережевої служби та мережевого сервісу, види мережевих оболонок.
4. Схема побудови мережевої операційної системи.
5. Переваги та недоліки сучасних мережевих систем.
6. Порівняння NetWare та Windows, Linux.

14. КАНАЛИ І ЛІНІЇ ЗВ'ЯЗКУ. КАБЕЛЬНІ СИСТЕМИ

14.1. Поняття каналу зв'язку

Канал зв'язку – сукупність ліній зв'язку і пристроїв, що забезпечують передачу сигналів від передавача до приймача.

Лінія зв'язку – фізичне середовище передачі даних, по якому здійснюється передача сигналів. Іноді терміни «лінія зв'язку» і «канал зв'язку» використовуються як тотожні.

У складі пристроїв каналу зв'язку розрізняють:

- апаратуру передачі даних – забезпечує передачу і прийом сигналів;
- проміжне обладнання – виконує дві функції: підсилює сигнали і забезпечує постійну комутацію між абонентами.

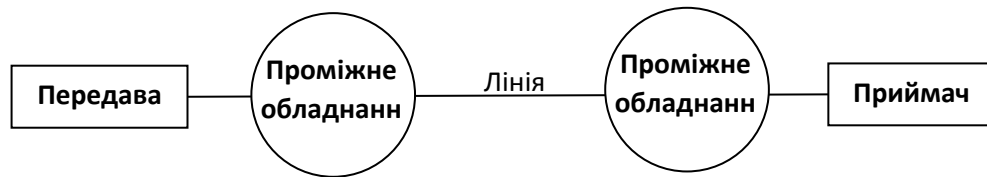
Таким чином, канал зв'язку утворюють:

- лінія зв'язку,
- апаратуру передачі даних (мережеві карти, модеми),
- проміжне обладнання (повторювачі, концентратори та інше).

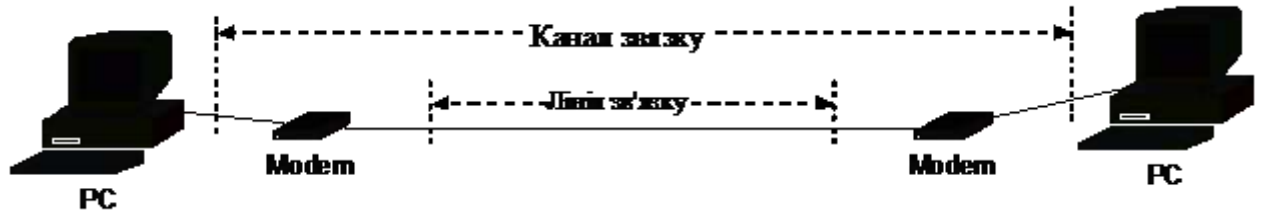
Схематично структуру каналу зв'язку та приклад простого каналу зв'язку зображено на рис. 14.1 а) та б) , відповідно.

В комп'ютерних мережах використовують:

- провідні канали зв'язку – будуються на базі провідних і кабельних ліній зв'язку (телефонні і телеграфні проводи, мідні коаксіальні кабелі, мідні виті пари, волоконно-оптичні кабелі, хвильоводи),
- безпроводні канали зв'язку – будуються на базі радіорелейних ліній зв'язку, ліній зв'язку транкової радіомережі, супутникового зв'язку, радіозв'язку надвисокої частоти (НВЧ) і мікрохвильового діапазону, оптичного зв'язку інфрачервоного і видимого діапазону випромінювання.



а)



б)

Рис. 14.1. Структура каналу зв'язку (а) та приклад простого каналу зв'язку (б)

14.2. Види ліній зв'язку

Для передачі комп'ютерних даних використовуються такі види ліній зв'язку, показані на рис. 14.2.

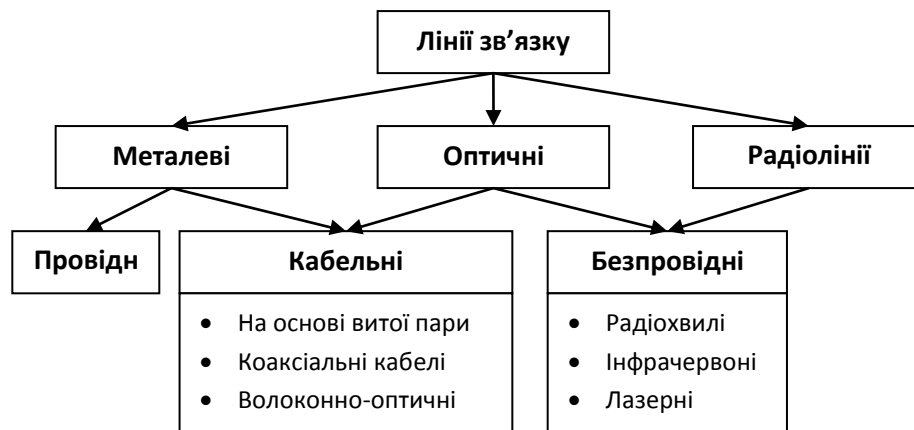


Рис. 14.2. Види ліній зв'язку

Провідні (повітряні) лінії – утворюють проводи без ізолюючих обплетень, що підвішуються до стовпів. Такими лініями традиційно передаються телефонні, або телеграфні сигнали, або при відсутності інших

можливостей, можуть використовуватися також для передачі даних. Також розвиваються технології, які дозволяють використовувати для передачі даних лінії електропостачання.

Кабельні лінії – складаються з мідних провідників, захищених кількома шарами ізоляції. В комп'ютерних мережах застосування знайшли три типи кабельних ліній:

- на основі витих пар мідних провідників,
- на основі оптоволоконних (волоконно-оптичних) кабелів,
- на основі коаксіальних кабелів.

Радіолінії (радіоканали) використовують для передачі даних радіосигнали. Швидкість передачі даних може досягати десятків Гбіт/с. Недоліком радіоліній є висока вартість передаючих і приймаючих пристроїв, низька захищеність, секретність і надійність зв'язку. Радіолінії використовують, якщо необхідно підтримувати зв'язок з рухомим об'єктом, або уникнути витрат, пов'язаних з укладкою кабелю.

В радіоканалах сигнали передаються шляхом модуляції високочастотної несучої частоти. Діапазони коротких, середніх і довгих хвиль (КХ, СХ, ДХ) забезпечують дальній зв'язок при невисокій швидкості передачі; при цьому використовується відбивання радіохвиль від іоносфери Землі. В діапазоні вище 4 ГГц радіохвилі не відбиваються від іоносфери Землі, тому радіохвилі більш високих частот – діапазонів ультракоротких хвиль (УКХ) і надвисоких хвиль (НВХ) – використовуються в супутникових і радіорелейних каналах зв'язку і є більш швидкими.

Інфрачервоні канали використовують для передачі даних радіохвилі в інфрачервоному діапазоні випромінювання. Гранична швидкість передачі даних по інфрачервоному каналу – 5-10 Мбіт/с. Як і у випадку радіоканалу, використовують відносно дорогі приймачі та передавачі, секретність не забезпечується. Головна перевага по відношенню до радіоліній – нечутливість до електромагнітних перешкод та відсутність необхідності

отримання дозволу на установку і експлуатацію у зв'язку з малою потужністю випромінювання (до 50 МВт). Погано працюють в умовах сильної запиленості повітря.

Лазерні канали утворюють за допомогою передавача і приймача лазерного випромінювання. Через високу вартість використовуються в спеціальних випадках: при відсутності можливості провідного з'єднання (наприклад, водна перешкода), для створення резервних каналів і т.д.

14.3. Кабельні системи

Кабельна система – фізичне середовище передачі даних, що побудоване на базі кабельних ліній зв'язку.

Історично саме кабельні системи стали використовуватись першими для цілей телекомунікації.

Основними типами кабелів, які використовуються при побудові комп'ютерних мереж, є:

- вита пара.
- волоконно-оптичні кабелі.
- коаксіальні кабелі.

14.3.1. Вита пара

Сьогоднішні кабельні системи ґрунтуються на використанні так називаної витої пари.

Вита пара (Twisted Pair) – кабель, що складається з двох скручених проводів у спільній ізоляції, як показано на рис. 14.3.



Рис. 14.3. Схематичне зображення витої пари

Суть витої пари полягає в рівномірному розподіленні зовнішніх шумів. Припустимо, між двома паралельними провідниками не витої пари різниця в

напрузі складає 2В. За умови впливу зовнішнього шуму, один з них може виявитися в зоні електро-магнітного поля, що додатково генерує 9В, а інший в зоні поля, що додатково генерує 5В. Тоді різниця складе не 2В, а 6В, як показано на рис. 14.4.

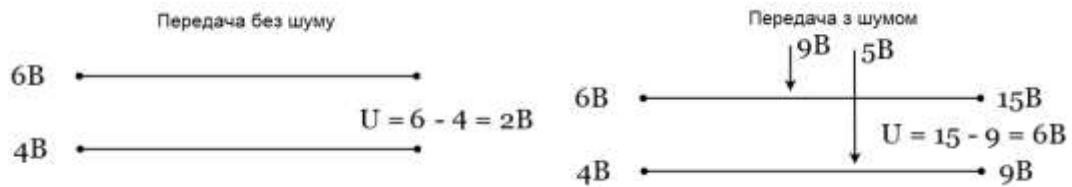


Рис. 14.4. Шуми у не витій парі

Якщо провідники скручено, як у витій парі, то зовнішнє джерело шуму може рівномірно генерувати в них напругу, що практично не змінить різницю напруги, як показано на рис. 14.5.

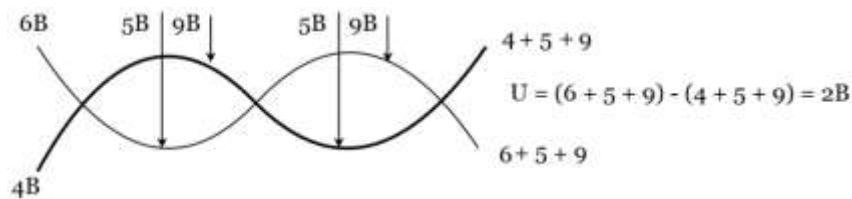


Рис. 14.5. Шуми у витій парі

Розрізняють два види витих пар:

- неекранована вита пара, або UTP (Unshielded Twisted Pair), – скручені проводи без додаткового екранування,
- екранована вита пара, або STP (Shielded Twisted Pair), – скручені проводи поміщуються в екрановану оплітку.

Кабель UTP відрізняється не високою вартістю і зручністю укладки, оскільки має високу гнучкість. Через не високу захищеність від шумів, лінії на основі UTP досить короткі – до 100 м. Досягнута швидкість передачі даних до 40 Гбіт/с. Стандарт EIA/TIA 568 визначає категорії кабелів на

основі неекранованої пари, які, серед інших, наведені в таблиці 14.1.

Табл. 14.1. Категорії кабелів

Категорія	Призначення і характеристики	Швидкість передачі
1	Звичайний телефонний кабель, в якому пари проводів не виті.	20 Кбіт/с
2	Кабель з витих пар для передачі даних у полосі частот до 1 МГц. Зараз використовується рідко.	4 Мбіт/с
3	Широко використовується для передачі даних і голосу в полосі частот до 16 МГц. Містить 9 витків на метр. Зараз найбільш поширений.	10 Мбіт/с
4	Використовується для передачі даних в полосі частот до 20 МГц. Призначався для роботи в мережах по стандарту IEEE 802.5 (Token Ring Lan). Зараз використовується рідко.	16 Мбіт/с
5	Розрахований на передачу даних в полосі частот до 100 МГц. Містить не менше 27 витків на метр. На сьогодні найбільш досконалий кабель, що рекомендується для використання в сучасних високошвидкісних мережах (типу Fast Ethernet).	100 Мбіт/с
5e	Найбільш поширений в комп'ютерних мережах. Переваги – в меншій собівартості та товщині.	100 Мбіт/с (2 пари) 1000 Мбіт/с (4 пари)
6	Неекранований кабель (UTP) складається з 4 пар провідників, здатний передавати дані на відстань до 55 м	10 Гбіт/с
6A	Складається з 4 пар провідників передає дані на відстань до 100 метрів. Кабель має або спільний екран (F / UTP), або екрани навколо кожної пари (U / FTP).	10 Гбіт/с
7	Кабель цієї категорії має загальний екран і екрани навколо кожної пари (F / FTP, або S / FTP).	10 Гбіт/с
7A	Кабель цієї категорії має загальний екран і екрани навколо кожної пари (F / FTP, або S / FTP). Полоса частот – до 1000 МГц	10 Гбіт/с
8/8.1	Повністю сумісний з кабелем кат. 6A. Швидкість передачі даних до 40 Гбіт/с при використанні стандартних конекторів 8P8C. Кабель цієї категорії має або загальний екран, або екрани навколо кожної пари (F / UTP або U / FTP). Полоса частот – 1600....2000 МГц	40 Гбіт/с
8.2	Сумісний з кабелем кат. 7A. Швидкість передачі даних до 40 Гбіт/с при використанні стандартних конекторів 8P8C або GG45 / ARJ45 і TERA. Кабель має загальний екран і екрани навколо кожної пари (F / FTP або S / FTP). Полоса частот – 1600....2000 МГц	40 Гбіт/с

Всі кабелі UTP, незалежно від категорії, випускаються в 4-парному виконанні, як показано на рис. 14.4.

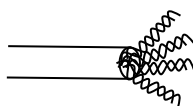


Рис. 14.4. Схематичне зображення UTP кабеля

Для з'єднання з обладнанням використовуються 8-контактні вилки і розетки RJ-45.

Як видно з табл.14.1, кабелі категорій 7 і 8.2 мають полоси пропускання 1000 і до 2000 МГц, відповідно. Кабелі категорії 6 можуть бути як неекранованими, так і екранованими. Кабелі категорії 7 обов'язково екрануються. Такі кабелі значно дорожчі і за вартістю наближаються до оптоволоконних. Вони використовуються у високошвидкісних мережах на відрізках довших, ніж кабелі 5 категорії.

Кабель STP добре захищає сигнали, що передаються від зовнішніх перешкод. Проте наявність екрану, який вимагає якісного заземлення, підвищує вартість ліній зв'язку і ускладнює прокладку. Основним стандартом на STP-кабелі є фірмовий стандарт IBM, у відповідності з яким визначаються дев'ять типів (не категорій) кабелів: Type 1, Type 2,..., Type 9 . Основним є Type 1, електричні параметри якого приблизно відповідають параметрам UTP категорії 5, але хвильовий опір більший (150 Ом проти 100 Ом).

14.3.2. Волоконно-оптичні кабелі

Оптоволоконний кабель складається з одного, або кількох, світловодів (оптичних волокон), що розміщуються у спільній захисній оболонці. Кожний світловод складається з центрального провідника (серцевини), який має високий показник переломлення світла, і скляної оболонки, яка має низький показник переломлення світла.

Основними конструкціями оптичних волокон є, як показано на рис. 14.5:

- одномодовий кабель;
- багатомодовий кабель.



Рис. 14.5. Конструкції оптичних волокон

В одномодових кабелях застосовуються серцевини з дуже малим діаметром, який є одного порядку з довжиною хвилі – близько 8...9 мкм. Розповсюдженими є одномодові кабелі 9/125 мкм, де 9 мкм – діаметр серцевини – світловоду, а 125 мкм – діаметр скляної оболонки. В такому кабелі існує лише одна мода. Виробництво світловодів такого малого діаметру є складним технологічним процесом і тому вартість одномодових кабелів є високою. Але перевагою таких кабелів є передача даних на великі відстані (до сотень кілометрів) з високими швидкостями (десятки Гбіт/с).

У багатомодових кабелях використовується ширші серцевини, що робить їх дешевшими. Розповсюдженими є багатомодові кабелі 50/125 мкм. Але в них може поширюватися кілька мод, що призводить до дисперсії імпульсу передачі, інтерференції променів, і в цілому веде до погіршення характеристик кабелю. Тому багатомодові кабелі використовуються в основному при передачі даних на невеликі відстані (до 2000 м) з невисокими швидкостями (до 1 Гбіт/с).

В якості джерела світла використовуються напівпровідникові лазери, або світлодіоди, з довжинами хвиль 850, 1300, або 1550 нм, що відповідають

«вікнам прозорості» оптоволокна.

В цілому, перевагами оптоволоконного кабелю є:

- висока перешкодозахищеність,
- невисоке затухання сигналів,
- висока швидкість розповсюдження сигналів.

Основними недоліками оптоволоконного кабелю є наступні:

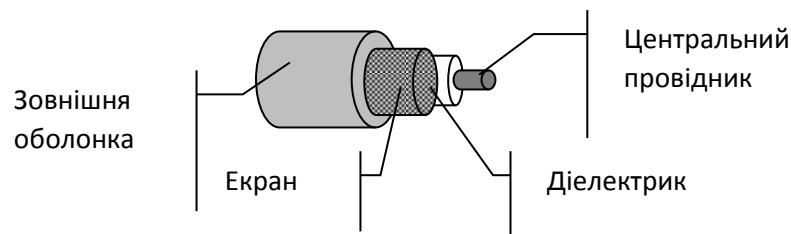
- складність монтажу (мала гнучкість кабелю, оптична точність установки).

Через складність монтажу оптоволоконний кабель продається у вигляді нарізаних кусків з встановленими роз'ємами.

- чутливість до іонізуючих випромінювань, що зменшують прозорість,
- чутливість до температурних перепадів, що призводять до утворення тріщин.

14.3.3. Коаксіальні кабелі

Коаксіальний кабель представляє собою електричний кабель, що складається з центрального мідного провідника і металевої оплітки (екрану), розділених шаром діелектрика і поміщених у зовнішню ізолюючу оболонку.



Металева оплітка відіграє подвійну роль – служить для передачі даних та захищає центральний провід від зовнішніх шумів.

Для побудови комп'ютерних мереж використовують:

- товстий коаксіальний кабель – з діаметром центрального провідника 2,17 мм і зовнішнім діаметром близько 1 см.
- тонкий коаксіальний кабель – з діаметром центрального провідника 0,89 мм і зовнішнім діаметром близько 0,5 см.

Хвильовий опір обох кабелів однаковий і складає близько 50 Ом.

Раніше коаксіальні кабелі використовувались досить широко, але на даний час не використовуються через широке розповсюдження витвої пари і оптоволоконна.

14.4. Структурована кабельна мережа

Відомо, що вартість кабельної системи визначається не вартістю кабелю, а вартістю робіт з його прокладання.

В 90-х роках отримав розвиток новий вид організації промислового зв'язку, який ґрунтується на виготовленні, поставці, монтажі, сертифікації повністю комплектних, сумісних з усім мережевим обладнанням систем проводки і з'єднань для будівель і споруд – структуровані кабельні системи.

Структурована кабельна система (СКС) – це набір комунікаційних елементів (кабелів, роз'ємів, кросових панелей, шаф, конекторів), а також методика їх використання, яка дозволяє створювати швидко розширювані структури зав'язків в комп'ютерних мережах. СКС є конструктором, за допомогою якого проектувальник будує потрібну конфігурацію. Загальне зображення СКС показано на рис. 14.6.

Основні переваги СКС:

- гнучкість – забезпечується простота керування переміщенням мережевої апаратури в споруді і між спорудами,
- універсальність – забезпечуються передача комп'ютерних даних, голосу, відео,
- стійкість – гарантується висока надійність і захищеність системи на багато років (найчастіше – на 25 років).

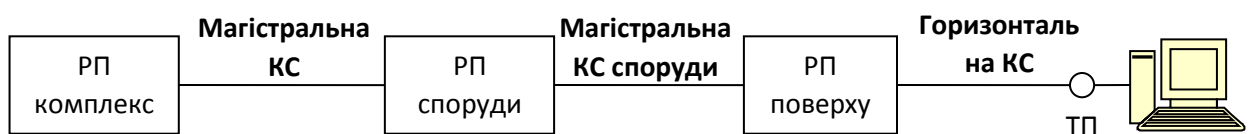


Рис. 14.6. Структурована кабельна система

Основною перешкодою широкого впровадження СКС є їх висока вартість.

Розробка СКС здійснюється у відповідності з стандартами СКС – документами, що детально описують і регламентують процес створення кабельних з'єднань локальних мереж.

Стандарти СКС періодично переглядаються (приблизно раз на п'ять років) в зв'язку з розвитком апаратних засобів комп'ютерних мереж.

На сьогодні існують три стандарти СКС, які відрізняються між собою деталями:

- EIA/TIA-568 «A Commercial Telecommunications Wiring Standard» – американський стандарт.
- CENELEC EN50173 «Performance Requirements of Generic Cabling Schemes» – європейський стандарт.
- ISO/IEC 11801 «Information Technology – Generic Cabling for Customer Premises Cabling» – міжнародний стандарт.

За своїм змістом стандарти СКС поділяються на три групи:

- стандарти проектування,
- стандарти монтажу,
- стандарти адміністрування.

У відповідності зі стандартами проектування, СКС включає три підсистеми:

- магістральна кабельна система (КС) комплексу – з'єднує розподільчі пункти комплексів з розподільчими пунктами споруд,
- магістральна КС споруди (вертикальна КС) – з'єднує розподільчі пункти поверхів з розподільчими пунктами споруди,
- горизонтальна КС – з'єднує розподільчі пункти поверхів з точками підключення (розетками) користувачів.

Розподільчі пункти (РП) представляють собою кросові шафи, що забезпечують можливість створення мережі базової топології – шини, кільця, або зірки.

У відповідності зі стандартами монтажу, рекомендується використовувати лише виту (кабель категорії 5 і вище) і волоконно-оптичний кабель. До того ж – чим вищий рівень підсистеми, тим більша перевага надається оптоволокну.

На основі стандартів СКС визначаються площі будівель (споруд), де можуть розташовуватись робочі місця користувачів. На рис. 14.7 показано архітектуру СКС у комплексі з кількох споруд.

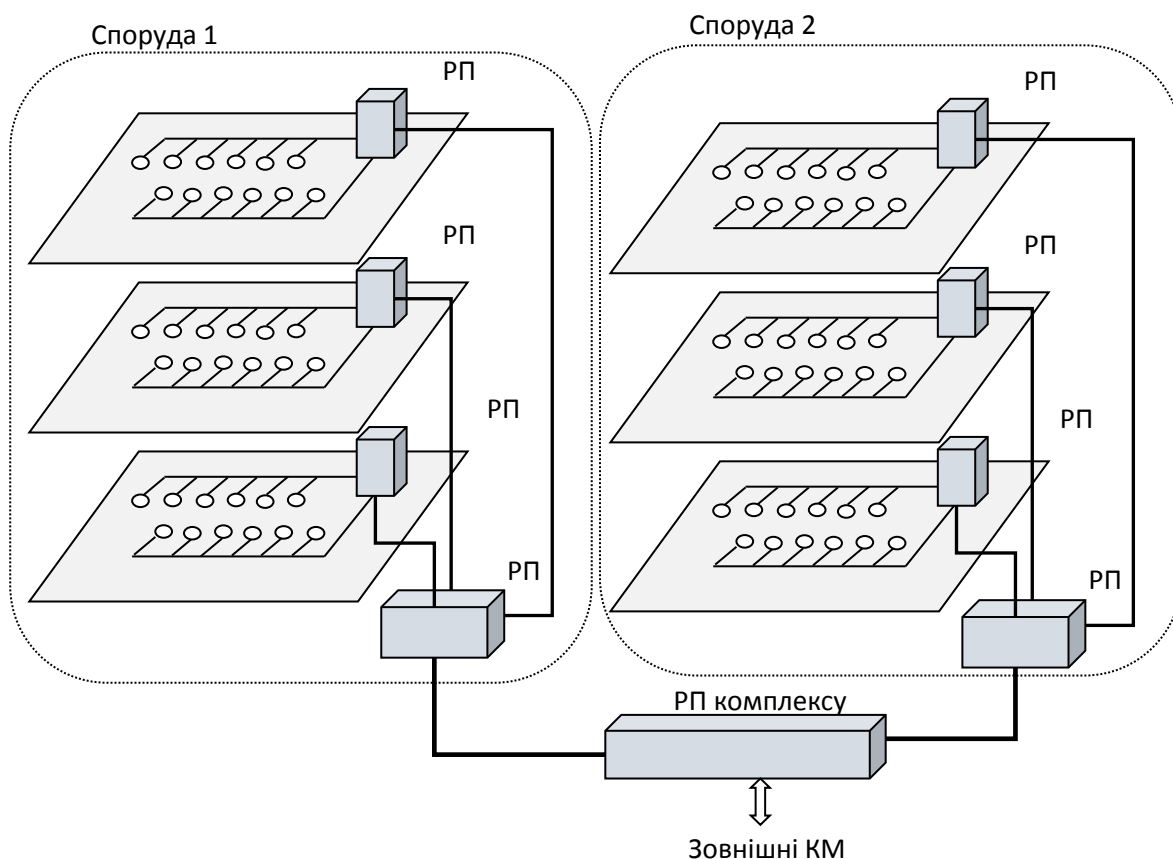


Рис. 14.7. Архітектура СКС для комплексу з кількох споруд

Як показано на рис. 14.7, кожне робоче місце обладнується парою розеток: одна – для підключення телефону, інша – для підключення

комп'ютера. Кожна розетка для підключення комп'ютера має містити як мінімум два гнізда: для підключення витої пари та волоконно-оптичного кабелю. Кожна розетка з'єднується «горизонтальними» кабелями з розподільчими пунктами, які рекомендується створювати на кожному поверсі будівлі. Довжина «горизонтальних» кабелів не повинна перевищувати 90 м (10 м залишається для підключення через трансиверний кабель). В якості «вертикальних» кабелів для з'єднання розподільчих пунктів поверхів з розподільчими будівлі може використовуватись як вита пара, так і волоконно-оптичний кабель. В якості магістральних ліній для з'єднання розподільчих пунктів будівель з розподільчим пунктом комплексу рекомендується використовувати волоконно-оптичний кабель.

Контрольні запитання до розділу

1. Поняття каналу зв'язку, лінії зв'язку.
2. Основні види лінії зв'язку, типи кабельних ліній.
3. Типи кабелів для мережі, категорії кабелів.
4. Будова, переваги та недоліки волоконно-оптичного кабелю.
5. Будова та види коаксіальних кабелів.
6. Архітектура структурованих кабельних мереж.

15. ХАРАКТЕРИСТИКИ ЛІНІЙ ЗВ'ЯЗКУ

15.1. Типи характеристик ліній зв'язку

Для визначення характеристик лінії зв'язку використовують аналіз її реакції на деякі еталонні впливи. Найчастіше в якості таких еталонних впливів використовують синусоїдальні сигнали різної частоти, що виступають у ролі зовнішніх шумів.

До основних характеристик ліній зв'язку відносяться наступні фізичні параметри.

1. Амплітудно-частотна характеристика – дозволяє визначити форму вихідного сигналу для будь-якого вхідного.
2. Пропускна здатність – це показник кількості одиниць інформації, яку лінія зв'язку може передати за певний проміжок часу. Характеризує максимально можливу швидкість передачі даних по лінії зв'язку.
3. Перешкодостійкість – визначає здатність лінії зв'язку зменшувати рівень перешкод, що створюються в зовнішньому середовищі та на внутрішніх провідниках.
4. Достовірність передачі даних – характеризує ймовірність пошкодження кожного біту даних, що передаються.

Увага розробників приділяється в першу чергу таким характеристикам, як пропускна здатність і достовірність передачі, оскільки вони прямо впливають на продуктивність і надійність створюваної мережі.

15.2. Амплітудно-частотна характеристика

Амплітудно-частотна характеристика – це залежність коефіцієнта передачі $A_{\text{вих}}/A_{\text{вх}}$ від частоти; тут $A_{\text{вих}}$ і $A_{\text{вх}}$ – амплітуди гармонічних складових сигналу відповідно на виході і вході лінії зв'язку.

Для визначення форми вихідного сигналу необхідно знайти спектр

вихідного сигналу, перетворити амплітуди складових гармоніки у відповідності з амплітудно-частотною характеристикою, а потім знайти форму вихідного сигналу, склавши перетворені гармоніки.

На практиці замість амплітудно-частотної характеристики використовуються інші, що є похідними від неї – полоса пропускання і затухання.

Полоса пропускання – діапазон частот, для якого коефіцієнт передачі $A_{\text{вих}}/A_{\text{вх}}$ перевищує 0,5. Ця характеристика визначає діапазон частот, які передаються по лінії зв'язку без значних спотворень.

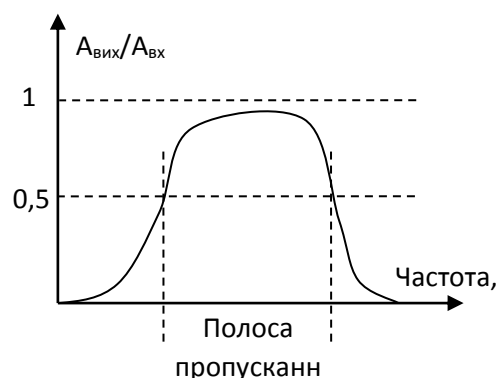


Рис. 15.1. Полоса пропускання як амплітудно-частотна характеристика

Для прикладу наведено значення полоси пропускання окремих середовищ:

- телефонна пара – 3100 Гц,
- вита пара – понад 100 МГц,
- волоконно-оптичний кабель – до 10 ТГц.

Затухання – обчислюється за формулою $Z = 10 \times \log_{10} \frac{P_{\text{вих}}}{P_{\text{вх}}}$ і вимірюється в децибелах (дБ); тут $P_{\text{вих}}$ і $P_{\text{вх}}$ – потужність сигналу відповідно на виході і вході лінії зв'язку. Ця характеристика визначає відносне зменшення потужності сигналу визначеної частоти. Оскільки $P_{\text{вих}} < P_{\text{вх}}$ завжди, то $Z < 0$, однак на практиці часто оперують абсолютними

значеннями Z . Оцінка значень затухання сигналу наведена в таблиці 15.1.

Табл. 15.1. Затухання сигналу залежно від відношення потужності

$P_{\text{вих}} / P_{\text{вх}}$	0,1	0,01	0,001	0,0001
Z , дБ	-10	-20	-30	-40

Наприклад, на частоті 100 МГц при довжині 100 м практично:

- для витої пари категорії 3 затухання $Z=-11,5$ дБ.
- для витої пари категорії 5 затухання $Z=-23,6$ дБ.

Тобто, як видно з таблиці 15.1, для витої пари категорії 5 за частоти сигналу 100 МГц від початкового сигналу через 100 м залишиться близько 0,01 його потужності (або 1%).

15.3. Пропускна здатність

Пропускна здатність лінії зв'язку вимірюється в бітах за секунду (біт/с) і залежить від амплітудно-частотної характеристики та спектру сигналів, що передаються: якщо спектральні складові сигналу виходять за полосу пропускання, сигнал зашумлюється та не уловлюється. На рис. 15.2 показано вплив полоси пропускання на пропускну здатність лінії зв'язку. На рис. 15.2 а показано накладання спектральних складових сигналу на полосу пропускання, що забезпечує високу пропускну здатність (показано на рис 15.2 а – справа). На 15.2 б показано вихід спектральних складових сигналу за полосу пропускання, що призводить до зашумлення сигналу (рис 15.2 б – справа).

Пропускна здатність вимірюється в біт/с, а не в байт/с, оскільки дані в мережах передаються послідовно, а не паралельно. Відповідно, похідні одиниці утворюються за допомогою множника 1000 (а не $2^{10}=1024$) : 1 Кбіт/с=1 000 біт/с, 1 Мбіт/с= 1 000 Кбіт/с , 1 Гбіт/с= 1 000 Кбіт/с і т.д.

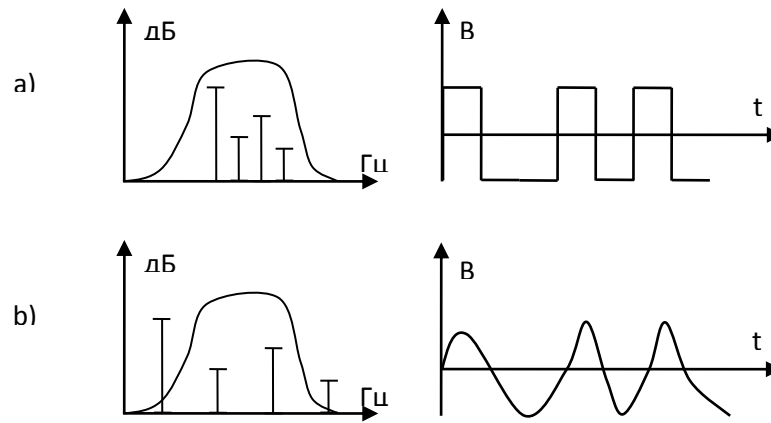


Рис. 15.2. Вплив полоси пропускання на пропускну здатність лінії зв'язку

Приклад 1. Припустимо, пропускну здатність лінії зв'язку дорівнює 100 Мбіт/с. Обрахуємо скільки байт/с може передаватись по цій лінії: $100 \text{ Мбіт/с} = 100\,000\,000 \text{ біт/с} = 100\,000\,000 : 8 \text{ байт/с} = 12\,500\,000 \text{ байт/с} = 12\,500\,000 : 1024 : 1024 \text{ Мбайт/с} = 11,9 \text{ Мбайт/с}$.

Вибір способу представлення дискретних даних у вигляді сигналів, що подаються лінією зв'язку, називається фізичним, або лінійним кодуванням. Від вибраного способу кодування залежить спектр, а отже і пропускну здатність лінії зв'язку.

Приклад 2. При використанні манчестерського кодування на один бітовий інтервал доводиться одна зміна рівня сигналу. Тому, для забезпечення пропускну здатності 10 Мбіт/с потрібна частота модуляції сигналів 10 МГц, а тривалість передачі одного біта інформації дорівнює $1 \text{ с} / 10^7 \text{ Гц} = 10^{-7} = 100 \text{ нс}$.

В лініях глобального зв'язку замість передачі електричного струму, що змінюється при зміні значення біта, використовується сигнал синусоїдальної форми, оскільки такі сигнали поширюються з мінімальним зашумленнями, перекрученнями. Такий синусоїдальний сигнал, параметри якого змінюються при зміні значення біта, називається несучою частотою.

Швидкість роботи апаратних засобів прийнято вимірювати в спеціальних одиницях вимірювання – бодах. Бод (baud) – кількість змін

інформаційного параметру несучої за секунду. Швидкість передачі в бодах - це кількість змін сигналу зв'язку) за одну секунду.

Якщо сигнал (напруга, частота, або фаза) змінюється один раз для кожного біта даних, то один біт / с дорівнює одному боду. Наприклад, модем з швидкодією 300 бод змінює свій стан 300 раз в секунду.

Для кодування інформації в лініях зв'язку використовують зміну одного параметру синусоїдального сигналу – амплітуду, частоту, або фазу. У відповідності з цим розрізняють наступні види модуляції, що показані на рис. 15.3:

- амплітудна модуляція – передбачає зміну амплітуди вихідного сигналу у відповідності зі зміною інформаційного сигналу,
- частотна модуляція – передбачає зміну частоти вихідного сигналу у відповідності зі зміною інформаційного сигналу,
- фазова модуляція – передбачає зміну фази вихідного сигналу у відповідності зі зміною інформаційного сигналу.

Методи амплітудної і частотної модуляції потребують для передачі одного біту не менше однієї зміни несучої амплітуди, або, відповідно, частоти.

Основна перевага фазової модуляції – це можливість кодування в кожній зміні більше одного бітового значення. Так, якщо передавач використовує для зсуву фази T бітів, то приймач може вивільнити всі T бітів, вимірявши величину зсуву. Оскільки кожен зсув кодує T бітів, то максимальна швидкість передачі даних визначається за формулою:

$$V=2R \times \log_2 2^T = 2RT,$$

де R – швидкість в бодах.

Приклад 3. Нехай модем, що працює на швидкості 2400 бод, передає інформацію, використовуючи амплітудно-частотну модуляцію. Зокрема, чотири стани фази (0° , 90° , 180° , 270°) і два значення амплітуди. В цьому випадку $T=3$, оскільки всього варіантів кодування інформації $2^3=8$. Тому

модем передаватиме дані зі швидкістю $2 \times 2400 \times \log_2 2^3 = 14\,400$ біт/с.

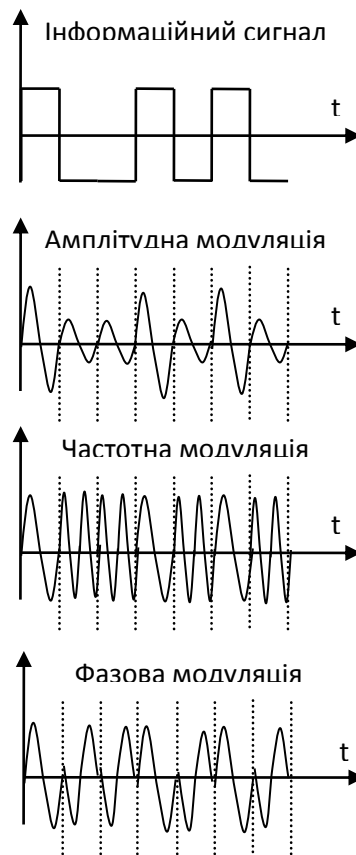


Рис. 15.3. Види модуляції сигналу в лінії зв'язку

15.4. Перешкодостійкість

У відповідності до видів лінії зв'язку, або за типом середовища передачі, найкращу перешкодостійкість мають волоконно-оптичні, гіршу – металеві, а найнижчу – радіолінії, як показано на рис. 15.4.



Рис. 15.4. Приклади перешкодостійкості

По відношенню до кабелю з витих пар перешкодостійкість визначається показником перехресних наведень на ближньому кінці – NEXT (Near End Cross Talk).

NEXT характеризує перешкодостійкість кабелю до внутрішніх джерел перешкод, які виникають внаслідок впливу електричних наведень однієї витой пари на іншу, і розраховується за формулою:

$$\boxed{NEXT = 10 \times \log_{10}(P_{\text{вих}}/P_{\text{нав}})},$$

де $P_{\text{вих}}$ – потужність вихідного сигналу, $P_{\text{нав}}$ – потужність наведеного сигналу.

Чим менше NEXT, тим кращим є кабель. Наприклад, для витой пари категорії 5 на частоті 100 МГц значення NEXT не перевищує -27 дБ.

Останнім часом став застосовуватись модифікований показник PS NEXT (Power Sum NEXT) – відображає сумарну потужність перехресних наведень від усіх витих пар кабелю.

15.5. Достовірність передачі даних

Достовірність передачі називають також інтенсивністю бітових помилок – (Bit Error Rate).

Наприклад, якщо значення BER дорівнює 10^{-4} , або 1/10000, це означає, що з 10 000 біт не достовірним є один.

В кабельних лініях зв'язку BER складає від 10^{-4} до 10^{-6} , в оптоволоконних – близько 10^{-9} .

15.6. Формула Шеннона

Чим більше невідповідність між полосою пропускання і шириною спектра інформаційних сигналів, тим ймовірніші помилки у розпізнаванні даних приймаючою стороною, а значить швидкість передачі стає нижчою.

Зв'язок між полосою пропускання F і її максимально можливою пропускною здатністю $V_{\max}$ безвідносно до прийнятого фізичного способу кодування встановлює формула Шеннона:

$$V = F \times \log_2(1 + P_c/P_{\text{ш}}),$$

де V – пропускна здатність лінії зв'язку в біт/с, F – ширина полоси пропускання в Гц, P_c – потужність сигналу, $P_{\text{ш}}$ – потужність шуму.

Формула Шеннона показує:

- Найбільш ефективний спосіб збільшення пропускної здатності V полягає у збільшенні полоси пропускання F .
- Менш продуктивний збільшення пропускної здатності V можливий також за рахунок збільшення відношення $P_c/P_{\text{ш}}$.

Приклад 4. Аналогова телефонна лінія має полосу пропускання $F = 3\,400 - 300 = 3\,100$ (Гц), а відношення сигнал/шум $P_c/P_{\text{ш}} \sim -35$ дБ. Тоді, оскільки $P_c/P_{\text{ш}} = 3162$, пропускна здатність телефонної лінії $V_{\text{тел}} = 3100 \times \log_2(1 + 3162) = 3100 \times 11,6 = 36\,044$ (біт/с). На практиці досягається дещо нижче значення – 33 600 біт/с.

Для дискретного каналу зв'язку за умови відсутності бітових помилок пропускна здатність визначається формулою Найквіста:

$$V = 2F \times \log_2 N,$$

де V – пропускна здатність лінії зв'язку в біт/с, F – ширина полоси пропускання в Гц, N – кількість можливих станів інформаційного параметру.

Зокрема, якщо сигнал має лише два відмінні стани, то пропускна здатність $V = 2F$. Але при $N > 2$ пропускна здатність каналу може бути вищою.

При врахуванні бітових помилок формула Найквіста приймає такий вид:

$$V = 2F \times [\log_2 N + P_{\text{пом}} \times \log_2(P_{\text{пом}}/(N-1)) + (1 - P_{\text{пом}}) \times \log_2(1 - P_{\text{пом}})],$$

де V – пропускна здатність лінії зв'язку в біт/с, F – ширина полоси пропускання в Гц, $P_{\text{пом}}$ – відношення числа біт, отриманих з помилками, до

загального числа переданих біт за час спостереження.

Зокрема, для $N=2$:

1. При $R_{\text{пом}} = 0$ пропускна здатність $V = 2F$.
2. При $R_{\text{пом}} = 1/2$ пропускна здатність $V = 0$.
3. При $R_{\text{пом}} = 1$ пропускна здатність $V = 2F$.

Інтерпретація отриманих даних полягає у наступному. При $R_{\text{пом}} = 0$ пропускна здатність досягає свого максимального значення $V = 2F$. При $R_{\text{пом}} = 1/2$ пропускна здатність $V = 0$, тобто прийняті дані не містять корисної інформації, оскільки кожний з прийнятих бітів може виявитись хибним. При $R_{\text{пом}} = 1$ знову $V = 2F$, оскільки кожний біт з високою ймовірністю інвертується і доля корисної інформації знову зростає.

Контрольні запитання до розділу

1. Параметри ліній зв'язку.
2. Амплітудно-частотна характеристика, полоса пропускання, затухання.
3. Пропускна здатність лінії зв'язку, амплітудна і частотна модуляції.
4. Перешкодостійкість лінії зв'язку, достовірність передачі.
5. Обрахунок бітових помилок за формулою Найквіста.

16. КАБЕЛЬНІ СИСТЕМИ ETHERNET

16.1. Типи Ethernet

Мережі Ethernet є на сьогодні найбільш поширеними. Вони не вирізняються винятковими характеристиками або оптимальними алгоритмами роботи, а за рядом параметрів поступаються іншим технологіям. Проте завдяки потужній підтримці, високому рівню стандартизації, великим обсягам випуску обладнання мережі Ethernet значно відрізняється від інших.

Класична мережа Ethernet у відповідності з стандартом IEEE 802.3 має такі основні характеристики:

- топологія – шина,
- середовище передачі – коаксіальний кабель,
- метод доступу – CSMA/CD,
- передача вузькополосна (без модуляції),
- пропускна здатність – 10 Мбіт/с.

Для мереж Ethernet з пропускною здатністю 10 Мбіт/с стандарт визначає чотири типи середовищ передачі даних:

- 10BASE5 (товстий коаксіальний кабель).
- 10BASE2 (тонкий коаксіальний кабель).
- 10BASE-T (вита пара).
- 10BASE-FL (оптоволоконний кабель).

В цих позначеннях перша цифра позначає пропускну здатність в Мбіт/с, слово BASE означає передачу в основній полосі частот (без модуляції високочастотного сигналу), а останній елемент позначає допустиму довжину сегмента в сотнях метрів (5 – 500 м, 2 – 185 м) або тип лінії зв'язку (Т – вита пара, F – оптоволоконний кабель).

В 1995 році з'явився стандарт на більш швидку технологію Fast

Ethernet (IEEE 802.3u) з пропускною здатністю 100 Мбіт/с. В якості середовища передачі використовується вита пара або оптоволоконний кабель. Використання топології шини стандартом не передбачено; в якості базової використовується топологія зірки.

Для мереж Fast Ethernet з пропускною здатністю 100 Мбіт/с стандарт визначає три основні типи середовищ передачі даних:

- 100BASE-T4 (чотири витих пари, напівдуплексний режим передачі).
- 100BASE-TX (дві витих пари).
- 100BASE-FX (оптоволоконний кабель).

В цих позначеннях, аналогічно, перша цифра позначає пропускну здатність в Мбіт/с, BASE означає передачу в основній полосі частот, а останній елемент позначає тип лінії зв'язку (Т – вита пара, F – оптоволоконний кабель).

Типи 100BASE-T4 і 100BASE-TX часто об'єднують в один тип під ім'ям 100BASE-T, а типи 100BASE-TX і 100BASE-FX – в тип 100BASE-X.

В 1998 році прийнятий також стандарт на ще більш швидкісну технологію Gigabit Ethernet (IEEE 802.3z) з пропускною здатністю 1000 Мбіт/с. В мережах Gigabit Ethernet зберігається все той же метод доступу CSMA/CD, що добре зарекомендував себе в попередніх мережах Ethernet, використовуються ті ж формати кадрів і розміри, тобто ніякого перетворення протоколів в місцях з'єднання з сегментами Ethernet і Fast Ethernet не потрібно.

Для мереж Gigabit Ethernet визначені чотири типи середовищ:

- 1000BASE-T (чотири неекранованих витих пари кат. 5е, довжиною до 100 м).
- 1000BASE-CX (екранована вита пара довжиною до 25 м, практично не зустрічається).
- 1000BASE-SX (багатомодовий оптоволоконний кабель довжиною до 500 м для світла з довжиною хвилі 850 нм).

- 1000BASE-LX (багатомодовий довжиною до 500 м і одномодовий довжиною до 2000 м оптоволоконний кабель для світла з довжиною хвилі 1300 нм).

В мережах Gigabit Ethernet використовується власний метод кодування 8B/10B (восьми бітам даних, які необхідно передати, ставиться у відповідність 10 біт, що передаються по мережі), який також є самосинхронізованим, але не вимагає подвоєння полоси пропускання, як у випадку коду Манчестер-II.

40GbE та 100GbE. В період 2007-2010 рр. розроблено стандарти IEEE P802.3ba (-2010), що встановлюють швидкість передачі даних 40 та 100 Гбіт/с за умови спільного використання кількох ліній зв'язку на 10, або 25 Гбіт/с.

Робота на відстанях 100 і 150 м зі швидкостями 40 і 100 Гбіт/с по оптичному кабелю (стандарти 40GBASE-SR4 і 100GBASE-SR10) забезпечена використанням хвиль близько 850 нм, подібно до стандарту 10GBASE-SR.

Робота на відстанях 10 і 40 км забезпечується використанням чотирьох різних довжин хвиль (близько 1310 нм) і використанням оптичних елементів зі швидкістю передачі даних 25 Гбіт/с (для 100GBASE-LR4 і 100GBASE-ER4) і 10 Гбіт/с (для 40GBASE-LR4).

16.2. Ethernet типу 10BASE5

Товстий коаксіальний кабель представляє собою 50 Ом кабель діаметром 1 см. Цей кабель є дорогим, проте він має хорошу перешкодостійкість, мале затухання і високу механічну міцність. Такий кабель використовувався в перших мережах Ethernet і майже вийшов з ужитку через складність монтажу і високу вартість.

За стандартом до одного сегменту довжиною до 500 м не повинно підключатися більше 100 абонентів, причому відстань між точками підключення не повинна бути меншою 2,5 м. Тому часто на оболонку кабелю

наносять мітки через кожні 2,5 м.

На обох кінцях товстого кабелю мають бути встановлені 50 Ом термінатори, один з яких заземляється.

Товстий кабель не підводиться безпосередньо до кожного комп'ютера через складність монтажу. Для приєднання мережевих адаптерів до товстого кабелю використовуються трансивери.

Трансивер, або MAU (Medium Attachment Unit) встановлюється безпосередньо на товстому кабелі (за допомогою спеціального обладнання – «вампірів») і з'єднується з адаптером трансиверним кабелем.

Трансиверний кабель представляє собою гнучкий кабель діаметром 1 см, що містить чотири екрановані виті пари. Довжина трансиверного кабелю може досягати 50 м, що забезпечує свободу переміщення комп'ютерів. На кінцях трансиверного кабелю встановлюються 15-контактні AUI-роз'єми типу «вилка». Ethernet на основі товстого кабелю показано на рис. 16.1

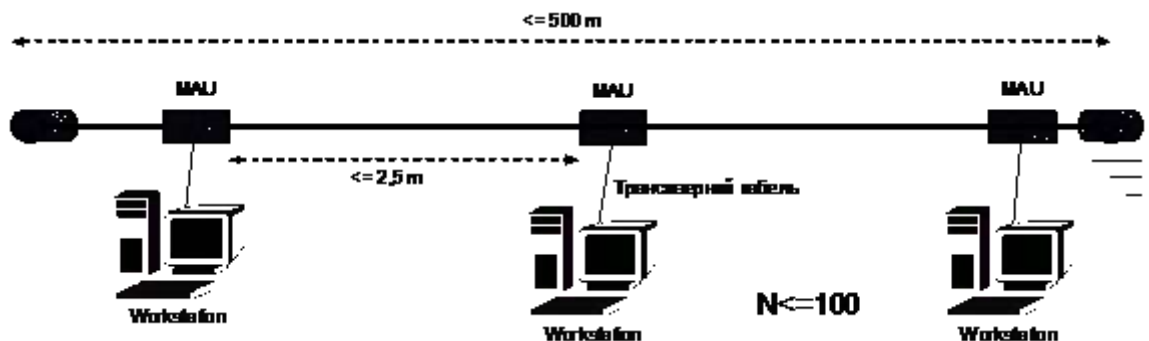


Рис. 16.1. Ethernet типу 10BASE5 на основі товстого кабелю

Максимальна кількість сегментів при реалізації мережі лише на товстому кабелі не повинна перевищувати п'яти, тобто загальна довжина мережі не може перевищувати 2,5 км. Для цього необхідно використати чотири репітери. Таким чином загальна кількість комп'ютерів, під'єднаних до товстого кабелю, не може перевищувати п'ятисот.

16.3. Ethernet типу 10BASE2

Тонкий коаксіальний кабель має вдвічі менший діаметр (біля 5 мм) і значно дешевший (приблизно в три рази) по відношенню до товстого кабелю і мають такий самий хвильовий опір 50 Ом. Вища гнучкість кабелю забезпечила більшу зручність монтажу, тому свого часу мережі на основі тонкого кабелю набули значного поширення, але сьогодні також майже вийшли з ужитку.

Апаратура для роботи з тонким кабелем простіша, ніж для роботи з товстим кабелем. Між кожною парою абонентів прокладається окремий кусок кабелю з двома роз'ємами типу BNC (байонетними) на кінцях. Роз'єми встановлюються за допомогою спеціального інструменту – обтискача.

До плати мережевого адаптера з BNC-роз'ємом приєднується байонетний T-конектор, який з'єднує плату з двома кусками кабелю, як показано на рис. 16.2.

На кінцевих комп'ютерах мають бути встановлені 50 Ом термінатори, один з яких заземляється.

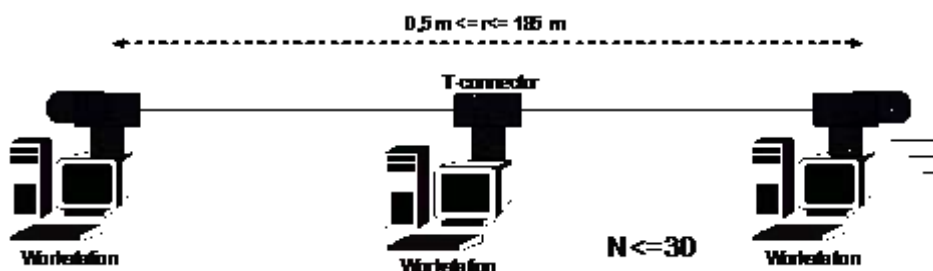


Рис. 16.2. Ethernet типу 10BASE2 на основі тонкого кабелю

За стандартом кількість сегментів мережі на тонкому кабелі не може перевищувати п'яти, а максимальна кількість абонентів (разом з репітерами) на одному сегменті не може перевищувати 30. Тому максимальна довжина мережі на тонкому кабелі не може перевищувати $185 \times 5 = 925$ м і об'єднувати більше $30 \times 5 = 150$ комп'ютерів.

16.4. Ethernet типу 10BASE-T

В мережах типу 10BASE-T передача сигналів здійснюється двома витими парами, одна з яких використовується для передачі, а інша – для прийому. Кабелем, що містить такі виті пари, кожний з абонентів приєднується до концентратора (хабу), реалізуючи топологію пасивної зірки, як показано на рис. 16.3. Концентратор забезпечує метод доступу CSMA/CD.

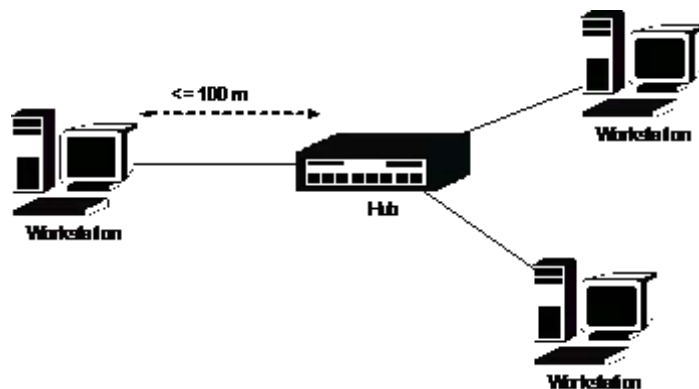


Рис. 16.3. Ethernet типу 10BASE-T

Стандарт передбачає використання неекранованої витієї пари категорії 3, або більш якісної – категорії 5 і вище. Довжина кабелю між мережевим адаптером і концентратором не повинна перевищувати 100 м.

Кабелі приєднуються до адаптера і концентратора 8-контактними роз'ємами RJ-45, в яких використовуються лише чотири контакти, відповідно до Таблиці 16.1.

Як видно з рис. 16.4, застосовуються два види з'єднання проводів кабелю:

- прямий кабель (direct cable) – з'єднуються однакові контакти обох роз'ємів,
- перехресний кабель (crossover cable) – передаючі контакти одного роз'єму з'єднуються з приймаючими контактами іншого, і навпаки.

Таблиця 16.1. Контакти роз'єму RJ-45

Контакт	Призначення	Колір
1	TX+	Білий/помаранчевий
2	TX-	Помаранчевий/білий
3	RX+	Білий/зелений
4	Не використовується	
5	Не використовується	
6	RX-	Зелений/білий
7	Не використовується	
8	Не використовується	

Для з'єднання мережевих адаптерів комп'ютерів з концентратором використовується прямий кабель. Прямим має бути також кабель, що з'єднує порт розширення UpLink одного концентратора із звичайним портом іншого. Але якщо концентратори з'єднуються через звичайні порти, кабель має бути перехресним. Перехресний кабель використовується також для об'єднання в мережу двох комп'ютерів без використання концентратора.

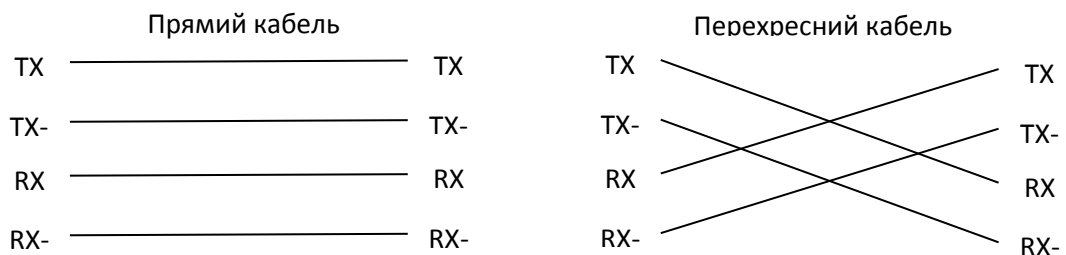


Рис. 16.4. Види з'єднання проводів кабелю

16.5. Ethernet типу 10BASE-FL

Застосування в Ethernet оптоволокна дозволило:

- значно збільшити допустиму довжину сегменту,
- суттєво підвищити перешкодостійкість.

Передача даних відбувається двома оптоволоконними кабелями в протилежних напрямках, як в 10BASE-T. Іноді використовуються двохпровідні оптоволоконні кабелі, що містять два оптоволокна в спільній оболонці. Використовується багатоходовий кабель із довжиною хвилі світла 850 нм.

Сама вартість оптоволоконного кабелю є близькою до вартості тонкого кабелю, проте апаратура виявляється значно дорожчою оскільки використовують дорогі оптоволоконні трансивери.

Оптоволоконний трансивер або FOMAU (Fiber Optic MAU) здійснює перетворення електричного сигналу в оптичний і навпаки. Для з'єднання трансивера з адаптером застосовується стандартний AUI-кабель (той же, що в 10BASE5), але його довжина не повинна перевищувати 25 м. Довжина оптоволоконного кабелю, що з'єднує трансивери і концентратор, може досягати 2 км, як показано на рис. 16.5.

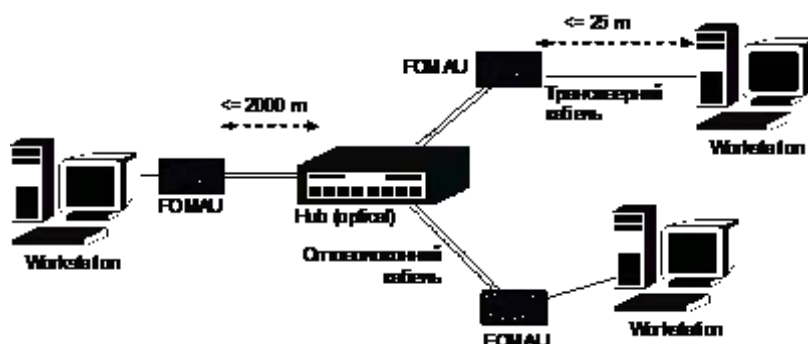


Рис. 16.5. Ethernet типу 10BASE-FL

16.6. Ethernet типу 100BASE-TX

Схема об'єднання комп'ютерів в мережу 100BASE-TX аналогічна 10BASE-T, але є відмінності:

- концентратор має бути розрахований на підключення сегментів 100BASE-TX.
- застосовується кабель з неекранованими витими парами категорії 5,

– мережеві адаптери мають бути Fast Ethernet,

Для під'єднання кабелів також використовуються роз'єми RJ-45, з 8 контактів яких використовуються чотири.

Довжина кабелю не може перевищувати 100 м, але стандарт рекомендує робити 10% запас і обмежуватись довжиною 90 м.

Як і в 10BASE-T, можуть використовуватись прямий і перехресний кабелі для того ж призначення.

16.7. Ethernet типу 100BASE-T4

Основна відмінність 100BASE-T4 від 100BASE-TX полягає в тому, що передача здійснюється не двома, а чотирма неекранованими витими парами. При цьому кабель може бути не лише категорії 5, а і категорій 3 і 4, але пропускна здатність залишається 100 Мбіт/с.

Схема об'єднання комп'ютерів в мережу не відрізняється від 100BASE-TX. Рекомендована довжина кабелю між адаптером і концентратором також 90 м.

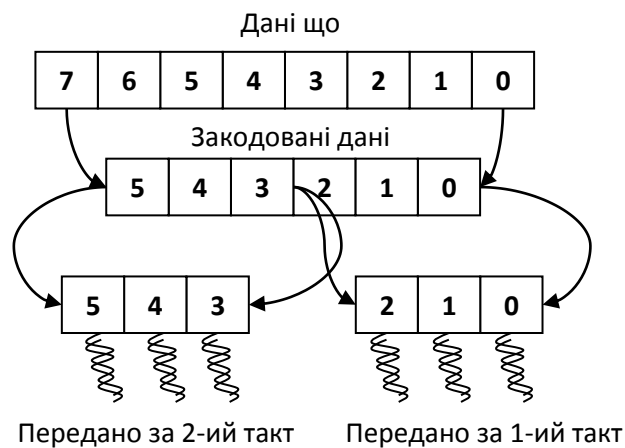


Рис. 16.6. Передача даних за кодування – 8В/6Т

Для реалізації передачі даних зі швидкістю 100 Мбіт/с кабелем з вузькою полосою пропускання (категорії 3) використовується оригінальний метод кодування – 8В/6Т. Ідея такого кодування полягає у тому, що 8 біт, які

треба передати, перетворюються в 6 тернарних (трирівневих) сигналів, які передаються за два такти трьома витими парами. При шести розрядному тризначному коді кількість можливих станів $3^6=729$, що більше $2^8=256$. В результаті, кожна вита пара передає дані зі швидкістю 25 Мбіт/с. Четверта вита пара в передачі даних участі не приймає, а використовується для виявлення колізій, як показано на рис. 16.6.

16.8. Ethernet типу 100BASE-FX

Апаратура 100BASE-FX дуже близька до 10BASE-FL. Тут також використовується пасивна зірка з використанням різнонаправлених оптоволоконних кабелів. Максимальна довжина кабелю між адаптером і концентратором складає 412 м. Може застосовуватись багатомодовий, або одномодовий кабель з довжиною хвилі світла 1,35 мкм.

100BASE-FX використовує той же метод кодування 4B5B та рядок NRZI, як і в 100BASE-TX.

Контрольні запитання до розділу

1. Типи Ethernet, їх середовища передачі.
2. Ethernet на товстому кабелі, трансивер, термінатор.
3. Ethernet на тонкому кабелі, переваги та недоліки.
4. Ethernet на витій парі, метод доступу з визначенням колізії.
5. Ethernet з оптоволоконним кабелем, інші технології Ethernet.

ЧАСТИНА IV. ПРАКТИЧНІ ЗАВДАННЯ З ОРГАНІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ

17. ВАРІАНТИ ПРАКТИЧНИХ ЗАВДАНЬ

17.1. Використання особливостей анімації при створенні проекту мережі

Мета роботи: на основі NetCracker та Cisco Packet Tracer вивчити можливості моделювання та навчитися створювати топологію мережі. Матеріал теоретичних відомостей для підрозділів 17.1-17.5. базується на літературі [25-26].

Теоретичні відомості

1. Запустіть NetCracker.
2. Відкрийте файл Router.net.
3. Розмістіть робоче вікно так, щоб збільшити місце для роботи з проектом (Рис. 17.1).
4. Для запуску анімації на панелі задач (Рис. 17.2) виберіть кнопку Start  або з меню Control виберіть команду Start і зачекайте коли почнеться рух пакетів у робочій області.

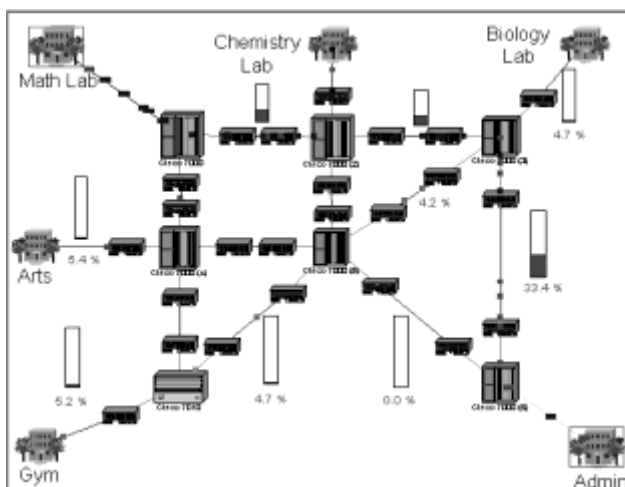


Рис. 17.1 Робоче вікно з анімацією



Рис. 17.2. Панель керування

5. Для зміни параметрів режиму моделювання натисніть на кнопку Animation Setup, що показано на рис. 17.3.

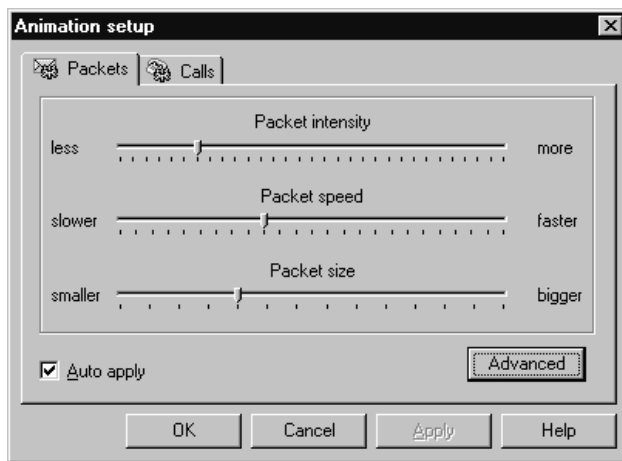


Рис. 17.3. Встановлення режимів моделювання

6. Скористайтесь лівою кнопкою миші для зміни розміру та швидкості пакетів. Натисніть OK для збереження параметрів і закриття вікна.

7. Відкрийте нижчий, другий рівень мережі, двічі клацнувши по будові, поміченій як Math Lab (Математична лабораторія). Отримаємо робочу область, показану на рис. 17.4.

Перегляньте, скільки інформаційних потоків пропускає підмережа математичної лабораторії, яке комутаційне устаткування задіяно в підмережі. Поверніться у верхній рівень проекту, закривши вікно математична лабораторія.

8. Відкрийте підмережу Admin і перегляньте, скільки інформаційних потоків пропускає підмережа Admin, яке комутаційне устаткування задіяно в підмережі. Закрийте вікно Admin кнопкою Close.

9. Збільшіть масштаб зображення проекту (інструмент ) і розмістіть

робочу область таким чином, щоб з'єднання (лінк) між Cisco 7000 (3) та Cisco 7000 (6) знаходилося у центрі робочої області.

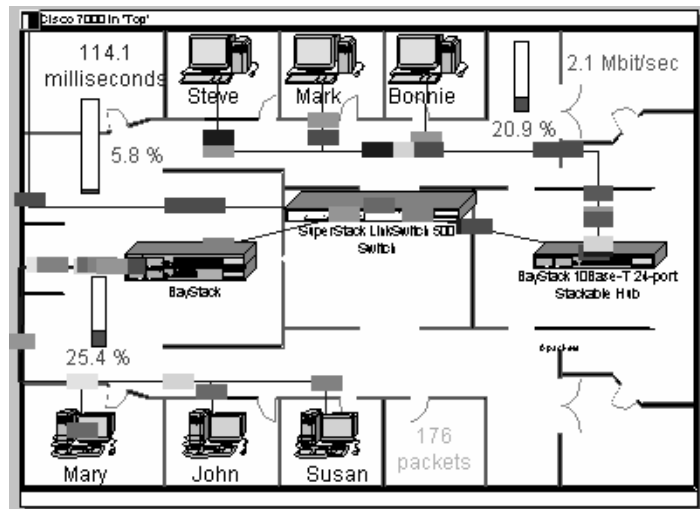


Рис. 17.4. Робоча область Math Lab

10. Щоб фізично розірвати зв'язок, на панелі Modes натисніть на кнопку Break/Restore. Потім помістіть курсор над з'єднанням двох маршрутизаторів Cisco і натисніть на з'єднання. З'явиться червона блискавка, яка сигналізує про те, що з'єднання обірвалося і трафік зупинився. Трафік змінюється згідно з роутинговим протоколом.

11. Тепер відновіть зв'язок: розмістіть курсор над перерваним зв'язком і клацніть лівою кнопкою миші. Курсор прийме форму гайкового ключа, указуючи, що ви знаходитесь в режимі Restore. При приміщенні курсора поверх перерваного зв'язку трафік відновлюється, спалахи червоного кольору зникають.

12. Вимкніть режим Break/Restore, натиснувши на інструментальній панелі Sites Modes кнопку Standard mode .

13. Для зупинки анімації натисніть кнопку Pause  на панелі інструментів.

14. Щоб отримати інформацію про пакет, помістіть курсор над ним. Викличте правою клавішою миші локальне меню, що показане на рис. 17.5,

та виберіть команду Say Info для інформації про пакет.

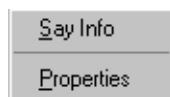


Рис. 17.5. Локальне меню пакетів

15. Встановіть курсор над пакетом та викличте з локального меню команду Properties. Відкриється діалогове вікно Properties, що показано на рис. 17.6.

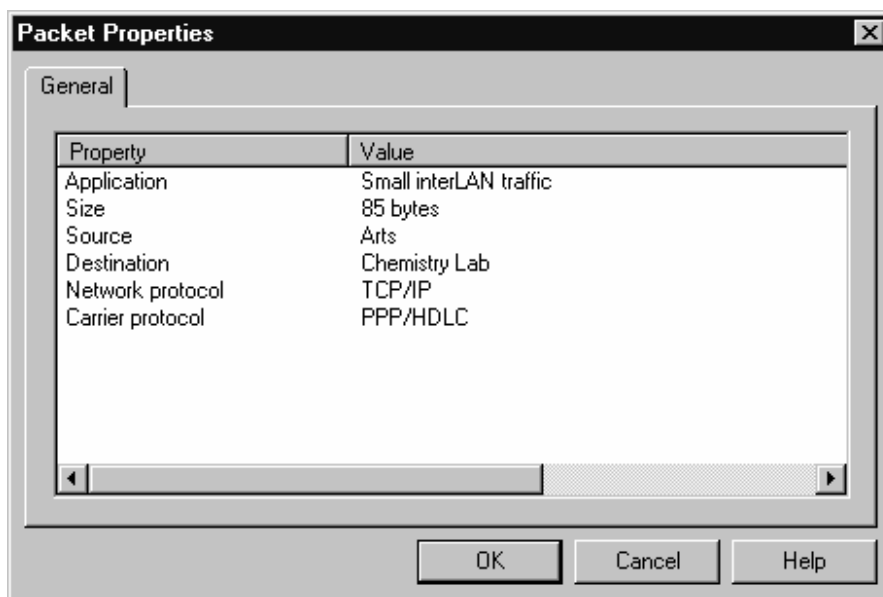


Рис. 17.6. Вікно діалогу Властивості Пакету

У вікно виводиться інформація, що стосується назви програми (Application), розміру пакета (Size), адреси відправника (Source), адреси призначення (Destination), типу протоколу мережі (Network protocol) та протоколу канального рівня (Carrier protocol). Закрийте діалогове вікно, натиснувши OK або Enter.

16. Створіть вигин у лінії зв'язку:

- a) переключіть в стан паузи, натиснувши кнопку Pause,
- b) утримуйте кнопку CTRL на клавіатурі, двічі клацніть ліву кнопку миші прямо на зв'язку. На зв'язку з'явиться чорний квадрат,

с) натисніть і утримайте клавішу миші на чорному квадраті і перетягніть його до нового положення, потім відпустите ліву кнопку миші.

Зв'язок згинається в місці, що вказано і дані слідуєть навколо вигину зв'язку.

17. Переіменуйте об'єкт GYM:

а) клацніть правою кнопкою на будівлі під назвою GYM, щоб звернутися до локального меню, і виберіть команду Properties,

б) надрукуйте Кафе в полі імені, натисніть ОК і закрийте діалог властивостей.

18. Виведіть звіт про діалоги і затримку передачі інформації в мережі: Tools - Reports - Wisard - Statistical - Data flows - Application Statistic.

19. Для закриття проекту спочатку зупиніть анімацію, а потім закрийте файл з меню File. Коли буде виведене повідомлення про те, чи зберігати зміни, натисніть No.

Тестові завдання

Відкрийте файл проекту NetCracker, вказаний у варіанті індивідуального завдання (Табл. 17.1), ознайомтеся з структурою його верхнього рівня і підрівнів, запустіть анімацію і змодельуйте ситуацію виведення окремого приладу, при необхідності налаштуйте параметри анімації, сформууйте стандартні звіти:

Табл. 17.1. Варіанти тестових завдань

Варіант	Файл
1	Hier.net
2	Techno.net
3	Tutor.net
4	Hier.net
5	Techno.net

- про використання обладнання (Device Utilization),
- про статистику роботи сегментів WAN (WAN Segments Statistics),
- про статистику роботи сегментів LAN (LAN Segments Statistics).

Збережіть структурну схему пропонованої мережі і сформовані звіти для включення в загальний протокол роботи.

Вимоги до протоколу роботи

Протокол має містити

1. Назву, тему та мету роботи.
2. Розділ 1 «Короткий опис, конфігурація і склад устаткування заданої мережі».
3. Розділ 2 «Короткий опис, конфігурація і склад устаткування підмереж».
4. Розділ 3 «Характеристики інформаційних пакетів».
5. Розділ 4 «Звіти про роботу мережі»:
 - про використання обладнання (Device Utilization).
 - про статистику роботи сегментів WAN (WAN Segments Statistics).
 - про статистику роботи сегментів LAN (LAN Segments Statistics).
6. Висновки (з відповідями на контрольні питання).

17.2. Розробка нового проекту в середовищі NetCracker

Мета роботи: створювати проекти NetCracker, вибирати комунікаційні пристрої, з'єднувати їх цифровими каналами, генерувати анотований звіт для покращення презентації проекту.

Теоретичні відомості

1. Запустіть NetCracker.
2. У меню File виберіть команду New. Якщо проект відкрито (.NET) у робочому вікні, то зберегти цей проект перед відкриттям іншого проекту. Не зберігайте жодного з проектів-прикладів NetCracker. Відкрийте вікно проекту до максимальних розмірів.
3. В браузері пристроїв виберіть Switches. Переконайтесь, що для

браузера пристроїв вибраний вид представлення Types (в списку Hierarchy). В браузері пристроїв послідовно розкрийте групи Switches, Workgroup, Ethernet і виберіть каталог Bay Networks для відображення Bay Networks switches в області зображень пристроїв.

4. Для розміщення комутатора у вікні проекту виконайте наступні кроки:

a. Виділіть модель BayStack 28104/ADV Fast Ethernet Switch в області зображень та перетягніть її у вікно проекту.

b. Збільшіть зображення пристрою, пересуваючи маркери розміру.

c. Збільшіть зображення назви пристрою. Для цього викличте контекстне меню та виберіть в ньому пункт Properties. З'явиться діалог Title Properties, в якому змініть значення розміру з 16 на 26 та натисніть ОК, або Enter, для закриття діалогу.

d. Відтягніть поле з назвою пристрою від зображення пристрою і збільшіть його розміри за допомогою маркерів встановлення розміру.

5. Додайте дві робочі станції в проект. Для цього:

a. В браузері пристроїв знайдіть і розгорніть групу LAN workstations.

b. Виберіть підгрупу Workstations, а в ній каталог Digital Equipment.

У полі зображень відобразяться робочі станції, виготовлені Digital Equipment Corporation.

c. Виділіть робочу станцію Alpha Station 200 4/166 та перетягніть її у вікно проекту. Змініть її розмір та розмір шрифту в назві.

d. Виберіть каталог PC у групі LAN workstations. Тепер скористайтесь смугою прокручування в браузері та виділіть каталог IBM.

Виділіть Aptiva C Series в області зображень та перетягніть в робоче поле, змініть розмір пристрою і кегль шрифту на 26пт та розмір назви. Зображення виглядатиме як на рис. 17.7.

6. Розмістіть LAN adapter на кожній з двох робочих станцій.

a. В браузері пристроїв розгорніть групу LAN adapters і підгрупу

Ethernet.

b. Виберіть каталог 3COM Corp.

c. В області зображень відшукайте Fast EtherLink 10/100 PCI карту, виділіть її, перетягніть на Alpha Station 200 4/166 та відпустіть кнопку миші після зміни курсора на знак плюс. Курсор змінює своє зображення на знак плюс, якщо карта підходить для робочої станції. Якщо форма курсору не змінюється, значить дана карта не сумісна з робочою станцією.

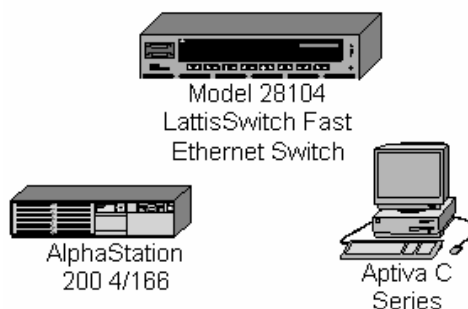


Рис. 17.7. Структура простої мережі

d. Виділіть Fast EtherLink 10/100 PCI карту знову, перетягніть у вікно проекту та вставте у робочу станцію Aptiva C Series, щоб курсор змінився на знак плюс та відпустіть клавішу мишки.

Для розпізнання пристроїв, які є сумісними з виділеними, виконайте наступне:

1. Виділіть пристрій.
2. Виберіть команду Find Compatible з меню Object або натисніть кнопку Compatibles на панелі інструментів Database.
3. Відкрийте список LAN adapters, в ньому - Ethernet список, потім виділіть каталог продавців.
4. Виберіть сумісний пристрій в області зображень та замініть ним попередньо вибраний пристрій.
7. Приєднайте робочі станції до комутатора.
 - a. На панелі Modes натисніть на кнопку Link devices (з'єднання

пристроїв) .

б. Помістіть курсор на Alpha-станцію та натисніть на зображення пристрою, потім перемістіть курсор на комутатор та натисніть на його зображення.

З'явиться вікно діалогу Link Assistant, що показано на рис. 17.8.

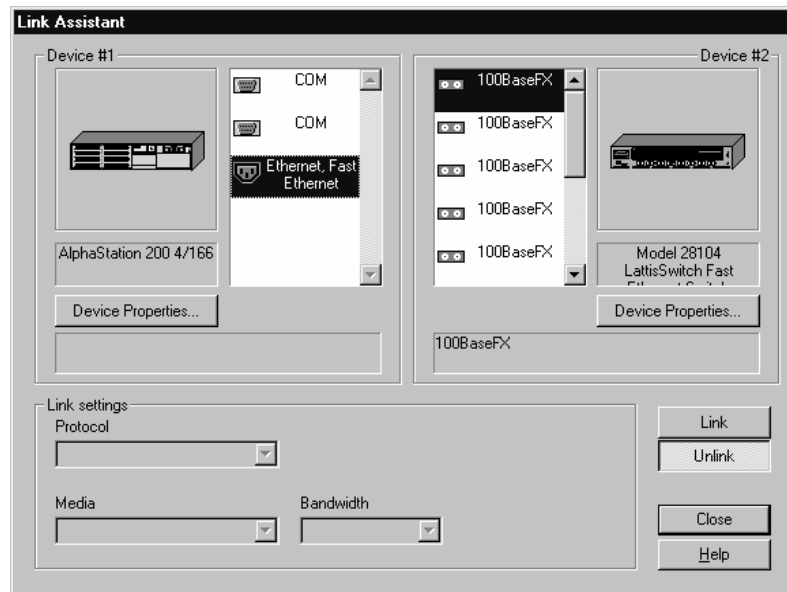


Рис. 17.8. Вікно діалогу Link Assistant

с. Натисніть кнопки Link і далі Close для створення з'єднання та закриття діалогу.

д. Скористайтесь методом Quick Link для з'єднання робочої станції IBM з комутатором, натиснувши клавішу Shift, лівою клавішою мишки натисніть на комутаторі, потім – на робочій станції IBM (при цьому режим Link Mode з виділеною кнопкою Link devices – активний).

Діалог Link Assistant не з'являється. При цьому зв'язок між другою робочою станцією та комутатором встановлено.

8. Перевірте тип ліній зв'язку. Ви помітите, що колір з'єднання - жовтий.

а. Для перевірки типу лінії зв'язку вам потрібно звернутися до діалогу Legends, який викликається з меню View командою Legends. Жовтий колір

вказує, що тип зв'язку – волоконно-оптичний.

b. Закрийте діалог Legends, натиснувши кнопку Close.

9. Визначте профіль трафіку до робочої станції.

a. Натисніть на кнопку Set Traffics 

b. Лівою клавішою мишки натисніть на робочій станції Alpha, потім – на робочій станції IBM.

Відкриється вікно діалогу Профілю (рис. 17.9).

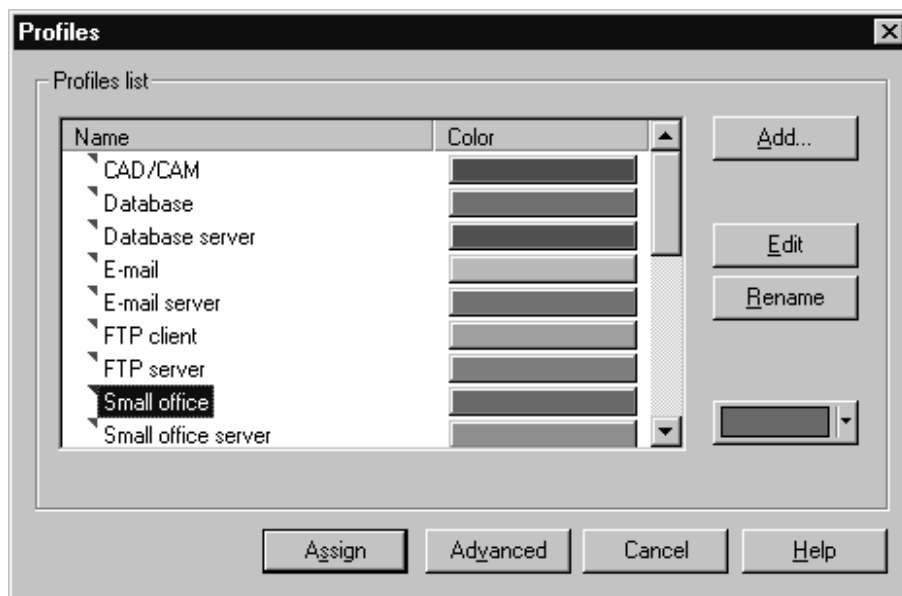


Рис. 17.9. Вікно діалогу Profiles

c. Для визначення трафіку типу Small Office між двома робочими станціями виберіть з переліку профілів тип Small office. Виберіть колір трафіку з запропонованої кольорової палітри. З'являється повідомлення Do you want to update? (“Бажаєте оновити?”). Натисніть Yes для позначення усього трафіку цього типу певним кольором. Натисніть No для виділення тільки цього трафіку обраним кольором, при цьому колір інших трафіків цього типу змінено не буде.

d. Натисніть кнопку Assign для визначення трафіку та закрийте діалог.

e. Повторіть кроки 9b-d, але цього разу виділивши першою робочу станцію IBM, а потім робочу станцію Alpha.

10. Перевірте встановлення трафіку між двома робочими станціями, запустивши анімацію. На панелі Control натисніть кнопку Start. З'явиться трафік між робочими станціями, який буде проходити через комутатор.

11. Змініть інтенсивність руху пакетів.

a. Натисніть на кнопку Animation Setup  щоб відкрити вікно діалогу Animation Setup.

b. Перейдіть на вкладку Packet intensity і перетягніть повзунок вліво, потім вправо на 4 пункти.

c. Натисніть OK для закриття вікна діалогу.

Через декілька секунд інтенсивність руху пакетів буде змінено.

12. Змініть швидкість руху пакетів.

a. Натисніть на кнопку Animation Setup  щоб відкрити вікно діалогу Animation Setup.

b. Перейдіть на вкладку Packet speed та перемістіть бігунок на найбільше значення.

c. Натисніть OK для закриття вікна діалогу.

Через декілька секунд швидкість руху пакетів буде змінено.

13. Зміна розміру пакетів.

a. Натисніть на кнопку Animation Setup щоб відкрити вікно діалогу Animation Setup.

b. Виберіть вкладку Packet size та перемістіть бігунок на найбільше значення.

c. Натисніть OK для закриття вікна діалогу.

14. Перегляньте усі пристрої, використані у проекті мережі, скориставшись вкладкою Recently used області зображень.

15. Розмістіть карту у вигляді фону.

a. Натисніть правою кнопкою мишки будь-де у вікні проекту (але не на зображенні пристрою і не на з'єднанні) для виклику локального меню, в якому виберіть команду Site Setup. Відкриється вікно діалогу Site Setup.

b. Перейдіть на вкладку Background, потім відмітьте пункт Map.

b. Скористайтесь кнопкою Browse для відкриття вікна діалогу Browse, знайдіть та виберіть бажаний файл карти, натисніть кнопку Open. Ім'я файлу з'явиться у полі Selected map file. Натисніть OK для прийняття внесених вами змін та закриття вікна діалогу.

Вікно діалогу виглядає як на рис. 17.10. Окрім карт, які поставляються з продуктом, можна використовувати власні карти.

d. Натисніть OK для прийняття змін та закриття діалогу.

16. Зробіть задній план кольоровим.

З меню Sites виберіть команду Site Setup.

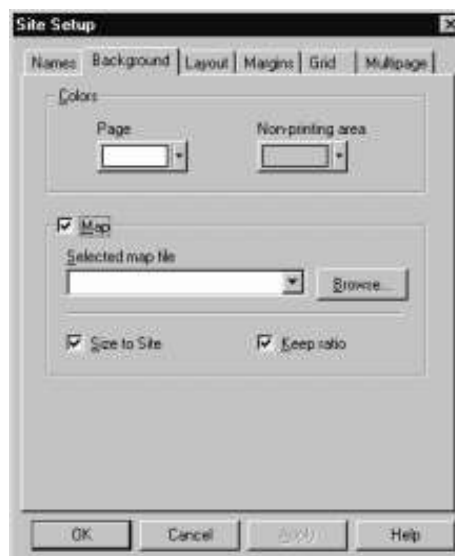


Рис. 17.10. Вікно діалогу Site Setup

Відкриється вікно діалогу Site Setup.

a. Перейдіть на вкладку Background, потім зніміть виділення з пункту Map.

Поле Selected map file стане неактивним.

b. Натисніть на полі Page. Відобразиться кольорова палітра, виберіть колір та відпустіть кнопку мишки.

c. Натисніть на полі Non-printing area для перегляду кольорової палітри

та виберіть інший колір, відпустіть кнопку мишки.

d. Натисніть ОК для внесення змін та закриття діалогу.

Новий фон та колір будуть відображені. Також можете додати колір фону у вікно, в якому розміщена карта.

17. Перегляньте профайли трафіку. Виберіть команду Data flow з меню Global. Відкриється вікно діалогу Data Flow (рис. 17.11). Два профайли трафіків, які створено, відображені в панелі Flows in the model (Потоки моделі). Натисніть Close щоб закрити вікно діалогу.

18. В меню File виконайте команду Save. Оскільки ви раніше не зберігали цей файл, то відкриється вікно діалогу команди Save As.

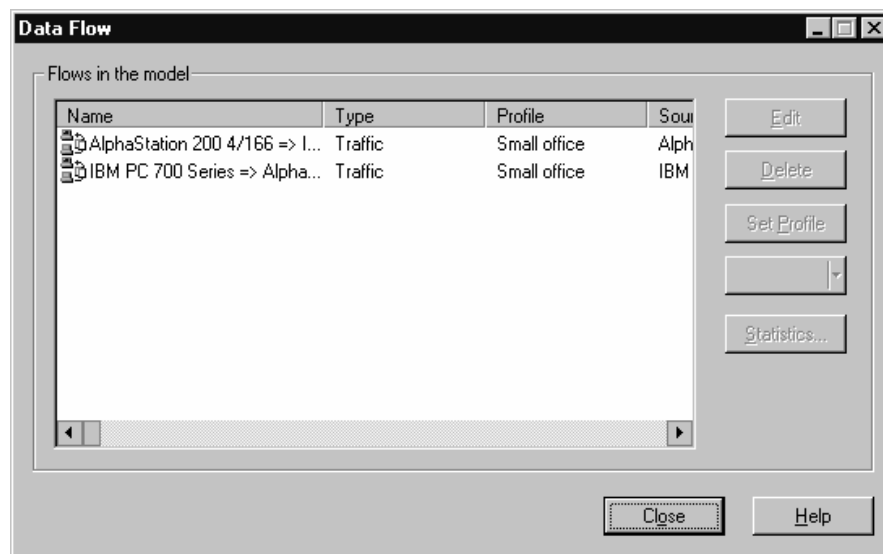


Рис. 17.11. Вікно діалогу Data Flow

19. За замовчанням файлу дається ім'я Net1.net, воно написано у полі Name. Перейменуйте файл. Розширення .NET автоматично буде додано до імені файла.

20. Зупиніть анімацію проекту кнопкою Stop, потім з меню File виконайте команду Close. Для закриття проекту спочатку зупиніть анімацію, а потім закрийте файл з меню File. На повідомленні про те, чи зберігати зміни, натисніть No.

Тестові завдання

Створіть проект мережі з топологією і складом обладнання, показаними на рис. 17.12.

Задайте трафік з профілями згідно варіанту завдання за таблицею 17.2.

Перед установкою трафіку станція-сервер встановіть необхідне серверне програмне забезпечення, вибравши його в групі Network and enterprise software- Server software браузеру пристроїв.

Запустіть анімацію і переконайтесь в працездатності моделі.

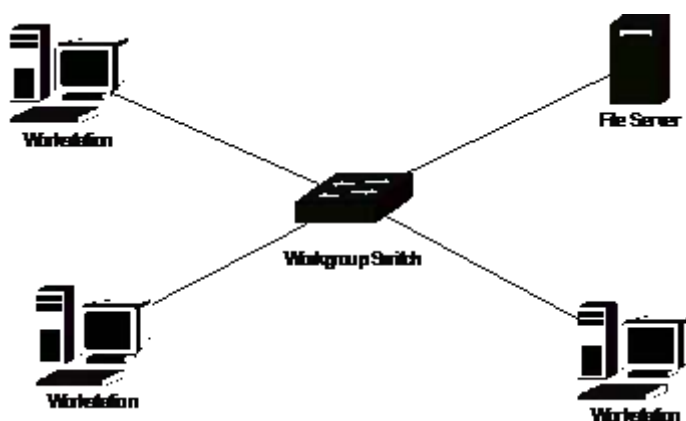


Рис. 17.12. Проект мережі з топологією «зірка»

Таблиця 17.2. Варіанти тестового завдання

Варіант	Трафік станція-станція	Трафік станція-сервер
1	Peer-to-peer	SQL server's client
2	Peer-to-peer	FTP client
3	Peer-to-peer	HTTP client
4	Peer-to-peer	E-Mail (SMTP)
5	Peer-to-peer	E-Mail (POP)

Сформуйте стандартні звіти:

- використання обладнання (Device Utilization),
- про статистику роботи додатків (Application Statistics),
- про статистику роботи мережевих пристроїв (Network Devices

Statistics).

Збережіть структурну схему пропонованої мережі і сформовані звіти для включення в протокол роботи.

17.3. Розробка багаторівневого проекту мережі

Мета роботи: ознайомитись з методами створення багаторівневих мережеских проектів, моделювання архітектури клієнт/сервер, відображення підсумків моделювання.

Теоретичні відомості

1. Відкрийте файл Tutor.net, який знаходиться в підкаталозі Samles.
2. Переключіться в браузері програми на вкладку Project Hierarchy.

Проекти демонструються як багаторівнева ієрархічна структура. Для проектів з одним рівнем буде відображено тільки верхній (Top) рівень. Кожен рівень має символи розкриття (знак +), або згортання (знак -) ієрархічної структури. Кожному рівню в Project Hierarchy відповідає окреме вікно сайту.

Кожен елемент ієрархії на вкладці Project Hierarchy відображається в робочій області у вигляді окремого вікна після подвійного кліка.

Вкладені рівні у вікні проекту обводяться червоною рамкою.

3. Виконайте подвійне клацання по об'єкту-контейнеру Building (рис. 17.13).
4. Тепер відобразіть головне вікно проекту Top, вибравши з меню Window команду Top.
5. Для відображення обох вікон проекту виберіть з меню Window команду Cascade.
6. Поміняйте вікна місцями, змініть їхні розміри за допомогою команди Zoom.



Building

Рис. 17.13. Об'єкт-контейнер

Залиште головне вікно проекту поточним. Робоче поле може виглядати як на рис. 3.2.

Тепер закрийте вікно Top натиснувши кнопку Close.

7. Щоб відкрити знову вікно Top, у вікні Building двічі натисніть на позначці з'єднання (connector icon). Вікно Top відкриється знову.

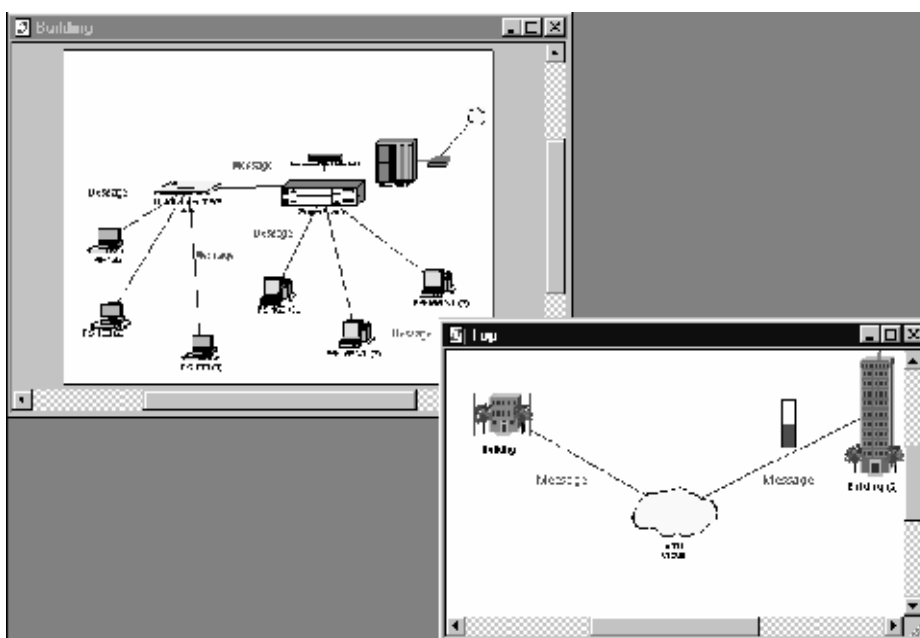


Рис. 17.14. Багаторівневий проект

8. Перейменуйте вікно сайту (проекту).
 - a. Спочатку зробіть вікно Top поточним, натиснувши на ньому.
 - b. Відкрийте діалог Setup вибравши з меню Sites команду Site Setup.
 - c. Виберіть вкладку Names. Підсвітіть ім'я (Top) в полі Site name, потім надрукуйте "The MacNally Corporation".
 - d. Натисніть OK для запам'ятовування змін та закрийте вікно діалогу.

е. Аналогічно перейменуйте вікно Building на “The MacNally Building”.

Нові назви The MacNally Corporation та the MacNally Building з’являться в заголовках вікон, у Project Hierarchy та в командах меню Window.

9. Анотуйте проект, використовуючи інструментальну палітру (drawing tools).

а. Зробіть поточним вікном MacNally Building та виділіть знак з’єднання.

б. В панелі Modes натисніть кнопку Draw.

с. Виберіть кнопку рисування ліній (Line) на панелі рисування (Drawing toolbar).

Скористайтесь Line для рисування стрілки, яка вказує на верхній правий кутку вікна сайту. Поверніться в стандартний режим (Standard mode) натиснувши на кнопці з рисунком стандартного курсору.

д. Змініть колір стрілки:

виділіть лінію,

з меню Object, з підменю Styles виберіть команду Draw color,

виберіть колір та натисніть ОК,

повторіть попередні дії для кожної ділянки стрілки.

е. Для помітки значка з’єднання, виконайте наступні дії:

з панелі Modes виберіть кнопку Draw,

з панелі Drawing (рис. 17.15) виберіть інструмент Text,

виділіть прямокутник, в якому буде розміщено текст.



Рис. 17.15. Панель Drawing

ф. Надрукуйте “Link to MacNally Corporation” та натисніть Enter.

г. Поверніться в режим Standard, натиснувши в панелі інструментів

кнопку стандартного режиму (Standard mode).

10. Виділіть канал зв'язку пристроїв, використавши команду Trace.

a. Запустіть анімацію, натиснувши кнопку Start.

b. На панелі Modes натисніть кнопку Trace . Потім у вікні MacNally Building клацніть на робочій станції P5-166 XL (3) і на робочій станції P5-133 XL (3).

Канал зв'язку між цими робочими станціями буде підсвічено червоним кольором.

11. Виділіть канал зв'язку між об'єктами, що знаходяться в різних вікнах проекту.

a. Якщо режим Trace все ще ввімкнутий, натисніть у вікні MacNally Building робочу станцію P5-133 XL (3).

b. Потім натисніть на Building (2) у вікні MacNally Corporation.

Канал зв'язку між цими двома об'єктами буде підсвічено червоним кольором.

c. Поверніть проект до стандартного режиму, натиснувши кнопку Standard mode.

Якщо натиснути на кнопці Trace, підсвічення червоним кольором зніметься, але режим Trace залишиться.

12. Зупиніть анімацію кнопкою Stop.

13. Закрийте поточний проект без збереження, вибравши з меню File команду Close.

14. Відкрийте новий проект командою New, меню File.

В робочій області з'явиться нове вікно.

15. У браузері пристроїв натисніть на вкладку Devices. Виберіть список Buildings, campuses and LAN workgroups.

В області зображень (Image) з'являться об'єкти Buildings, campuses and LAN workgroup.

16. Виділіть один з об'єктів Building в області зображень та перетягніть

його у вікно Top.

17. Перетворіть об'єкт Building на контейнерний об'єкт, скориставшись командою Expand.

Виділіть об'єкт Building у вікні проекту, потім правою кнопкою мишки відкрийте локальне меню та виберіть команду Expand, або виберіть команду Expand з меню Object.

Зображення Building у вікні Top тепер обведено червоною рамкою, що означає, що це – контейнерний об'єкт.

Щоб побачити ієрархічну структуру, в Browser натисніть вкладку Project Hierarchy.

18. Закінчіть проект, побудувавши в будівлі Building найпростішу мережу з архітектурою клієнт/сервер. Краще використовувати типові, вже сконфігуровані пристрої (Generic Devices), які включені в базу пристроїв NetCracker Professional. Для цього натисніть вкладку Devices в браузері об'єктів, потім відкрийте список LAN workstation та виберіть каталог типових пристроїв Generic Devices. Типові робочі станції відображаються в області зображень Image (рис. 17.16).



Рис. 17.16. Приклади зображень робочих станцій

а. У області зображень Image виділіть та перетягніть у вікно Building робочу станцію Ethernet, яка вже має карту LAN-адаптера.

б. З меню Edit виконайте команду Duplicate.

в. В браузері Device відкрийте список концентраторів (Switches), в ньому виберіть Workgroup, потім Ethernet та натисніть на каталозі типових пристроїв.

d. У полі Image виділіть типовий Ethernet-комутатор (рис. 17.17) та перетягніть його у вікно Building.

е. Натисніть на кнопки Link devices .

f. Натисніть на робочій станції та тягніть з'єднання до концентратора. Відпустіть кнопку мишки.

З'явилася вікно діалогу Link Assistant, в якому натисніть кнопку Link, потім Close.



Ethernet Switch

Рис. 17.17. Приклад зображення комутатора

g. Повторіть ці дії і для другої робочої станції.

h. Зробіть вікно Top поточним.

i. Перейдіть в стандартний режим роботи, потім виділіть список Buildings, campuses and LAN workgroups у браузері Device.

У полі Image з'явилися зображення Buildings, campuses and LAN workgroup device (рис. 17.18)

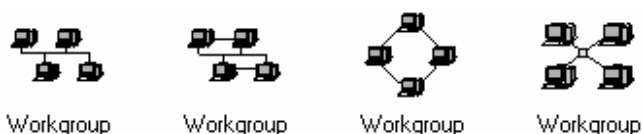


Рис.17.18. Приклади типових робочих груп

j. Виділіть та перетягніть зображення типової робочої групи у вікно Top

k. У вікні Top з'єднайте робочу групу з Building (в панелі Modes натисніть на кнопку Link, потім натисніть мишкою на робочій групі, потім на будівлі). Пунктирна лінія вказує, що це з'єднання ще не завершено.

l. Перейдіть у стандартний режим та двічі натисніть на зображенні

Building у вікні Top. Відкриється та стане поточним вікно Building.

m. З панелі Modes виділіть команду Link devices. У вікні Building для завершення з'єднання натисніть на зображенні з'єднання (Connector icon), потім на концентраторі.

Відкриється діалог Link Assistant. Зображення з'єднання (Connector icon) зазвичай розміщено в куточку вікна. Якщо потрібно, скористайтесь збільшувальною кнопкою для знаходження зображення з'єднання.

n. Виберіть Ethernet port в панелі Switch Port Configuration, натисніть кнопку Link, потім Close. Діалог Link Assistant закриється та встановиться з'єднання між Building та Top.

19. Зробіть одну з робочих станцій сервером.

a. В браузері пристроїв Device спустіться до списку “Network and enterprise software”. В ньому виберіть “Server software”, після чого в області зображень Image з'являться доступні типи серверів.

b. Перетягніть E-mail сервер на зображення робочої станції. Курсор має змінитися на стрілку зі знаком плюс, що буде означати, що ви можете встановити це програмне забезпечення.

20. Призначте трафік клієнт-сервер.

a. В інструментальній панелі Modes натисніть кнопку Set Traffic.

b. У вікні Building натисніть на робочій станції без серверного програмного забезпечення, потім у цьому ж вікні натисніть на робочій станції з серверним програмним забезпеченням.

Відкриється вікно діалогу Profiles.

c. Виберіть тип трафіку E-mail та натисніть кнопку Assign.

21. Призначте трафік група-станція.

a. У вікні Top натисніть на зображенні робочої групи, потім перейдіть у вікно Building та натисніть на робочій станції без серверного програмного забезпечення. Відкриється вікно діалогу Profiles.

b. Виберіть тип трафіку Small office та натисніть кнопку Assign.

с. Для запуску анімації натисніть кнопку Start на інструментальній панелі Control.

д. Для зупинки анімації натисніть кнопку Stop.

Тестові завдання

В якості індивідуального завдання додайте у вікно Building об'єкт Room (з групи Buildings, campuses and LAN workgroups браузера) і перетворіть його на контейнер. У вікні контейнера Room побудуйте проект мережі у відповідності з варіантом завдання попередньої роботи. З'єднайте робочу групу у вікні Top з комутатором у вікні Room і задайте для трафіку між робочою групою і робочими станціями тип Small office. Запустіть анімацію і переконайтесь в працездатності моделі.

1. Сформууйте стандартні звіти:

- про використання обладнання (Device Summary),
- про статистику роботи додатків (Aplication Statistics),
- про статистику роботи модулів (Module Statistical Items).

2. Збережіть створений файл проекту NetCracker, скориставшись командою Save з меню File. Закрийте проект, вибравши команду Close з меню File. Закрийте NetCracker, вибравши команду Exit з меню File.

Вимоги до протоколу роботи

Протокол має містити:

1. Назву, тему та мету роботи.

2. Розділ 1 «Короткий опис, конфігурація і склад устаткування заданої мережі».

3. Розділ 2 «Опис мережевого трафіку»

4. Розділ 3 «Модифікація фонового зображення».

5. Розділ 4 «Звіти про роботу мережі»:

- про використання обладнання (Device Utilization),
- про статистику роботи додатків (Aplication Statistics),
- про статистику роботи мережевих пристроїв (Network Devices

Statistics).

6. Висновки (з відповідями на контрольні питання).

17.4. Налаштування та використання бази пристроїв NetCracker

Мета роботи: знайомство з функціями системи налаштування пристроїв Device Factory та використовувати можливості комбінованого пошуку (Compatible Search) в базі даних.

Теоретичні відомості

1. В NetCracker відкрийте проект Router.net з підкаталогу Samples.
2. Перейдіть на вкладку Project Hierarchy.
3. У браузері вікна Project Hierarchy виберіть вікно Math Lab.
4. Виділіть робочу станцію Steve.
5. Викличте Device Factory Wizard одним зі способів:

- натисніть на кнопку Device Factory ,
- виберіть з меню Database команду Device Factory,
- з меню Object виберіть команду Add to Database: Via Factory.

Можна скористатися будь-яким з вказаних способів для виклику Device Factory. Коли створюється новий пристрій на базі існуючого, то є деякі відмінності у використанні команд Device Factory з меню Database та Add to Database: Via Factory з меню Object. Кнопка Device Factory та команда Device Factory з меню Database звертаються до пристроїв, які розміщені в полі Image, а команда Add to Database: Via Factory звертається до пристроїв, виділених в вікні проекту (Project pane). Відкриється вікно діалогу Device Factory Wizard (рис. 17.19).

У вікні Device Factory Wizard є варіанти:

- створити з нуля (Create from scratch),
- створити з робочої станції Steve (Create from Steve).

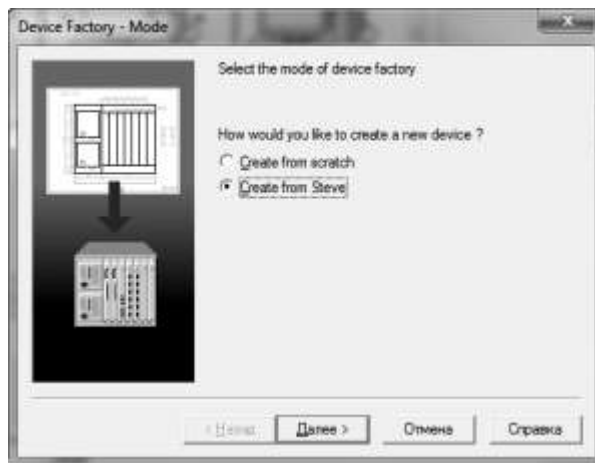


Рис. 17.19. Вікно Device Factory-Mode.

6. Виберіть варіант “Create from Steve” та натисніть кнопку Next.

Тип пристрою, який ви вибрали, підсвічено і ви можете впевнитися, що пристрої саме цього типу ви вибрали для модифікації (рис. 17.20).

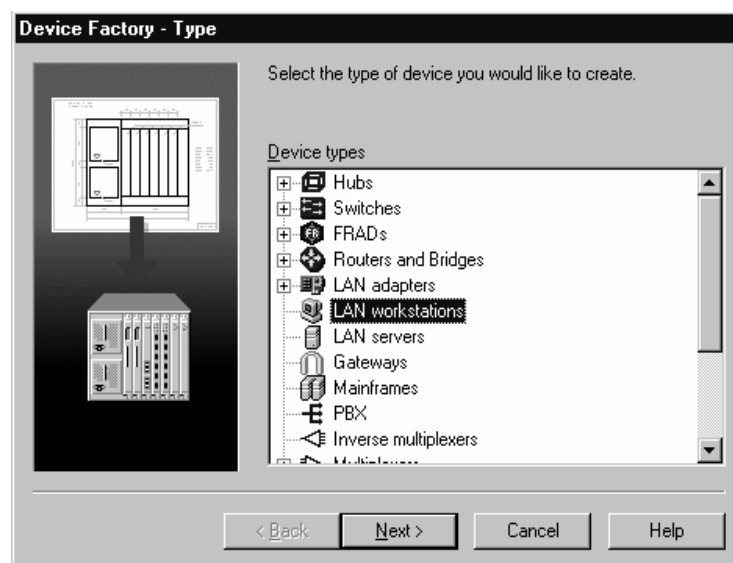


Рис. 17.20. Вікно Device Factory Type Screen.

7. Натисніть кнопку Next. Відкриється вікно Device Factory Computer (рис. 17.21).

8. Змініть кількість слотів на 4. Тобто, скільки комп’ютер має місць для установки плат адаптерів та внутрішніх модемів.

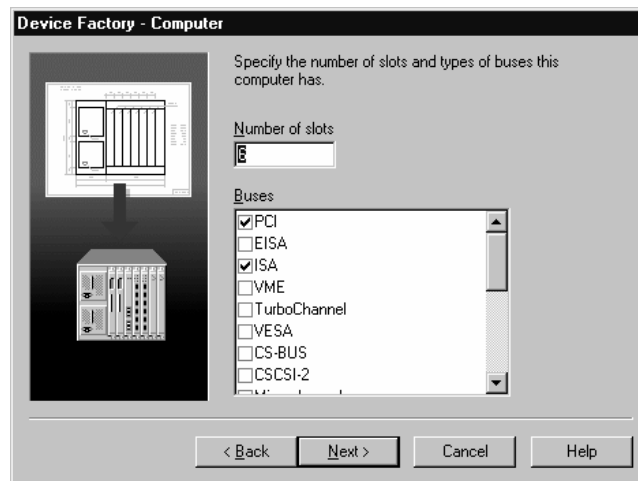


Рис. 17.21. Вікно Device Factory Computer

9. В секції Buses відмітьте типи шин VESA, PCI та ISA. Новий створений пристрій буде приймати тільки ті адаптери та внутрішні модеми, які мають один з вибраних типів шин.

10. Натисніть кнопку Next. Відкриється вікно Device Factory Name (рис. 17.22).

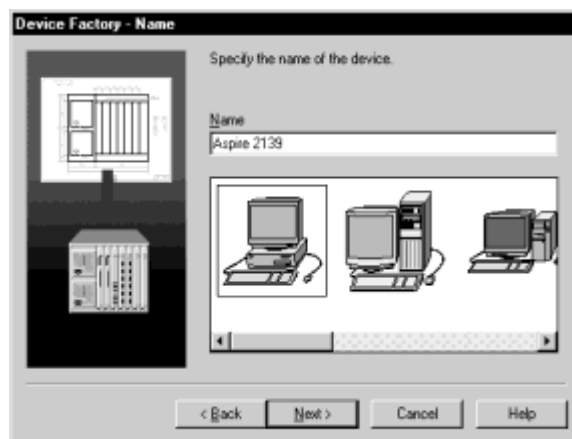


Рис. 17.22. Вікно Device Factory Name

11. Надрукуйте ім'я "Development Group Workstation" та натисніть кнопку Next. Відкриється вікно Device Factory Port Groups (рис. 17.23).

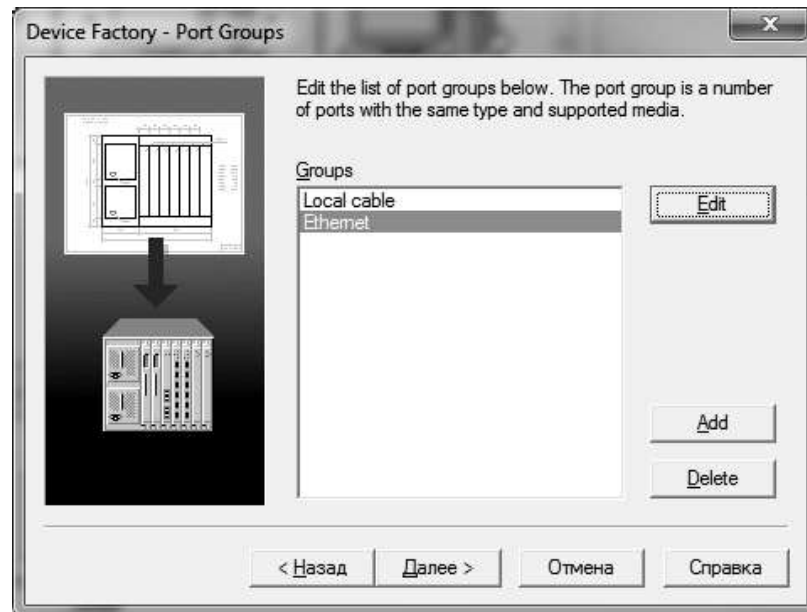


Рис. 17.23. Вікно Device Factory-Port Groups

12. Додайте групу порту Ethernet, натиснувши кнопку Add. Відкриється вікно Port Factory Number (рис. 17.24).

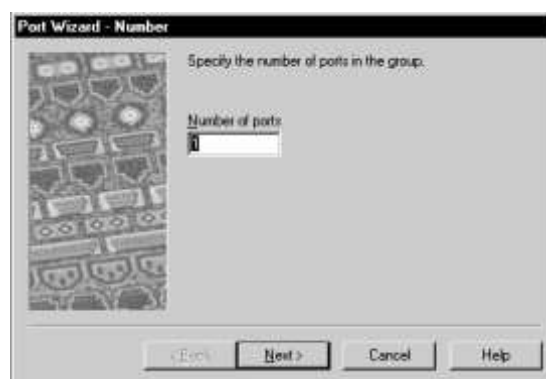


Рис. 17.24. Вікно Factory Number

Якщо ввести кількість портів більшу, ніж це можливо для цього типу пристрою, з'явиться повідомлення про помилку.

13. Змініть кількість портів у групі на 2 та натисніть кнопку Next. З'явиться вікно Port Factory Link Type (рис. 17.25). Як мінімум один тип порту має бути виділений. Якщо нічого не виділено, з'явиться повідомлення про помилку.

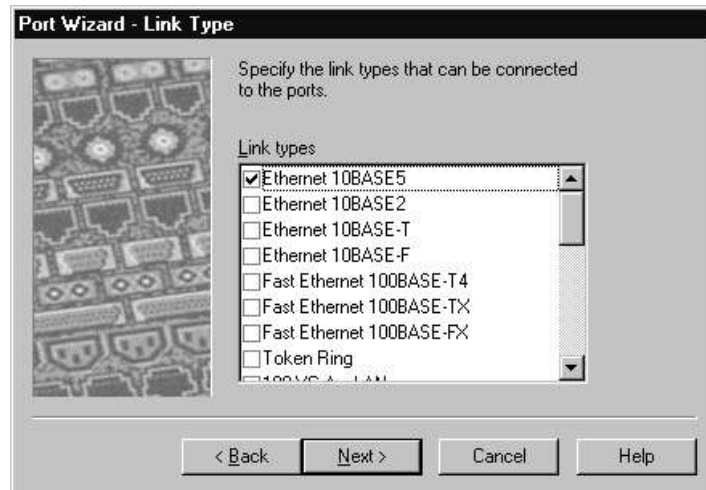


Рис. 17.25. Вікно Port Factory Link Type

14. Виділіть поля Ethernet 10BASE2 та Ethernet 10BASE-T та натисніть кнопку Next. Відкриється вікно Port Factory wizard Media (рис. 17.26).

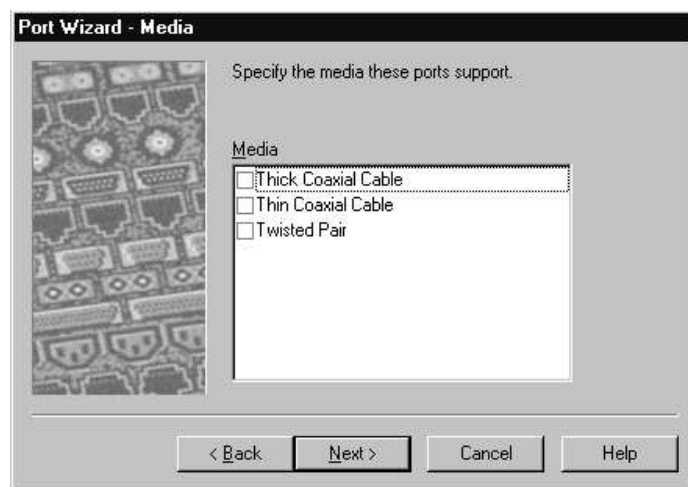


Рис. 17.26. Вікно Port Factory Media

Усі порти в групі All the ports in a group only except media of the type checked.

Як мінімум один тип медіа має бути виділений. Якщо нічого не виділено, з'явиться повідомлення про помилку.

15. Виділіть типи медіа тонкий коаксіальний кабель (Thin Coaxial Cable) та виту пару (Twisted Pair) і натисніть кнопку Next.

16. Натисніть кнопку Next на сторінці Device Factory Port Groups та натисніть Finish для збереження тільки що створеного пристрою в базі даних користувачів.

17. В меню File виберіть команду Close. Не зберігайте змін у проекті Router.net project.

18. Відкрийте новий проект та відобразіть браузер бази даних (якщо він не відображається), виконавши команду Database Browser з меню View.

19. Для відображення пристроїв з бази даних користувача, в якій вже є один тільки що створений пристрій:

- з випадаючого списку Hierarchy (панель Database) виберіть значення User,

- в області перегляду Image перейдіть на вкладку Devices.

20. Виберіть у області перегляду Image робочу станцію, яку ви тільки що створили - “Development Group Workstation” - та перетягніть її в робочу область проекту.

21. Тепер знайдіть пристрої, які сумісні з “Development Group Workstation”. Для цього в панелі Database натисніть кнопку Compatible, або з меню Object виберіть команду Find Compatible. Браузер автоматично переключиться в режим Compatible Device Browser та відобразить сумісні пристрої. Браузер відображає тільки ті пристрої, які сумісні з виділеним пристроєм (рис. 17.27).

22. Знайдіть в базі даних сумісну ATM-карту:

- a. Виберіть Types у полі Database Hierarchy,

- b. Розкрийте список LAN adapters, у ньому відкрийте список ATM,

- c. Відкрийте каталог Interphase.

23. Виділіть 5525 PCI ATM адаптер та перетягніть його на нову робочу станцію. Вигляд курсору зміниться на символ +, що означає сумісність карти.

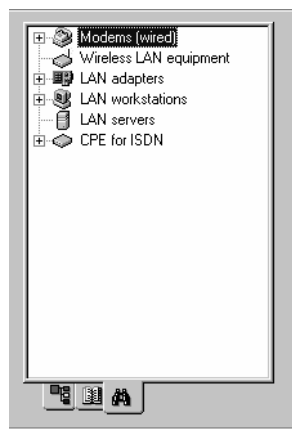


Рис. 17.27. Результати Compatible Search

24. Для копіювання робочої станції з картою адаптера, виберіть з меню Edit команду Replicate. Відкриється вікно діалогу Replicate (рис. 17.28).

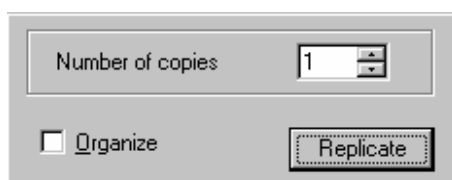


Рис. 17.28. Діалог Replicate

- а. Для створення 10 копій, вкажіть число 10 у полі Number of copies,
- б. Для впорядкування нових копій в геометричному порядку, відмітьте поле Organize.

25. Натисніть кнопку Replicate. Відкриється вікно Organize (рис. 17.29).

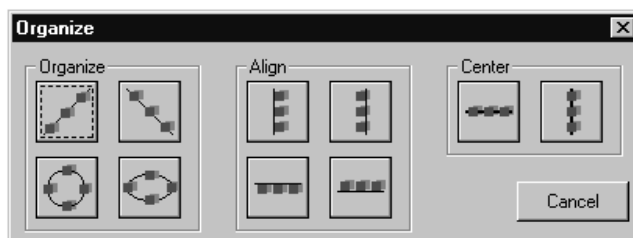


Рис. 17.29. Вікно діалогу Organize

Можна організувати будь-яку групу об'єктів, виділивши цю групу та виконавши команду Organize з меню Object.

26. Виберіть зразок кола. Після цього вікно діалогу автоматично закривається. Десять скопійованих об'єктів (робочі станції з картами) розміщуються у формі кола.

27. Для пошуку пристроїв в базі даних за іншими критеріями скористайтесь пошуком по базі даних (Database search). Для початку пошуку натисніть кнопку Find з меню Database Відкриється вікно діалогу Find (рис. 17.30).

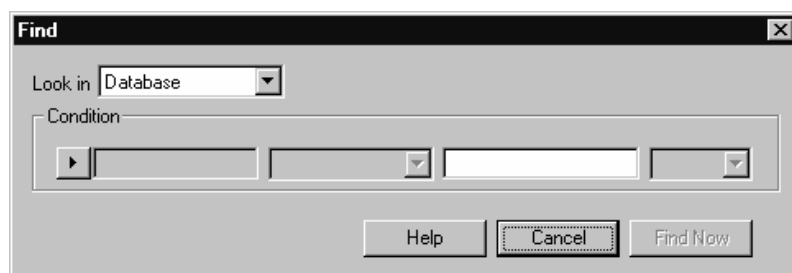


Рис. 17.30. Вікно діалогу Find

29. Натисніть на кнопку Condition та виберіть модель (Model). В наступному полі зі списком виберіть Includes. В третьому списку напишіть 7000 та натисніть кнопку Find Now. Браузер автоматично переключиться в режим Search Device Browser та відобразить результати пошуку.

30. Закрийте проект Router.net без збереження.

Тестові завдання

В новому проекті у відповідності з варіантом завдання (Табл. 17.3) створіть і додайте до Device Factory комплект компонентів обладнання для побудови комп'ютерної мережі, що складається з мережевого адаптера, персонального комп'ютера з цим мережевим адаптером і комутатора.

Використайте створені компоненти для побудови 2-рівневого проекту комп'ютерної мережі, верхній рівень якої відображає дві будівлі, а на нижньому рівні – їх локальні мережі ЛОМ1 і ЛОМ2.

Задайте трафік з профілем Small Office Peer-to-peer між комп'ютерами однієї мережі і трафік з профілем Small InterLAN traffic між комп'ютерами

різних мереж.

Запустіть анімацію і переконайтесь в працездатності моделі.

Сформууйте стандартні звіти:

- про використання обладнання (Device Utilization),
- про компонентний склад мережі (Network Summary),
- про статистику роботи додатків (Application Statistics).

Таблиця 17.3. Варіанти тестових завдань

Варіант	Технологія Ethernet	Шина	Кількість портів комутатора	Кількість комп'ютерів	
				ЛОМ1	ЛОМ2
1	10Base5	ISA	8	7	10
2	10Base2	EISA	14	15	8
3	10Base-T	VESA	12	14	14
4	100Base-T4	PCI	16	20	6
5	100Base-TX	PCI	10	17	14

Збережіть структурну схему пропонованої мережі і сформовані звіти для включення в звіт про виконання роботи. Закрийте NetCracker.

Вимоги до протоколу роботи

Протокол має містити:

1. Назву, тему та мету роботи.
2. Розділ 1 «Створення і додавання в базу пристроїв NetCracker нових компонентів»
3. Розділ 2 «Короткий опис, конфігурація і склад устаткування заданої мережі»
4. Розділ 3 «Опис мережевого трафіку»
5. Розділ 4 «Звіти про роботу мережі»:
 - про використання обладнання (Device Utilization).
 - про компонентний склад мережі (Network Summary).

- про статистику роботи додатків (Application Statistics).
6. Висновки (з відповідями на контрольні питання).

17.5. Моделювання мережі в Cisco Packet Tracer

Мета роботи: знайомство з багатофункціональною програмою моделювання мереж Cisco Packet Tracer та створення моделі мережі

Теоретичні відомості

Багатофункціональна програма моделювання мереж Cisco Packet Tracer допомагає створювати мережеві топології з використанням маршрутизаторів і комутаторів Cisco, робочих станцій і мережевих з'єднань типу Ethernet, Serial, ISDN, Frame Relay. Дана програма моделювання є корисною як для навчання, так і для роботи. Наприклад, для роботи з налаштування мережі на етапі її створення, або під час усунення несправностей. Загальний вигляд програми Cisco Packet Tracer показано на рис. 17.31.

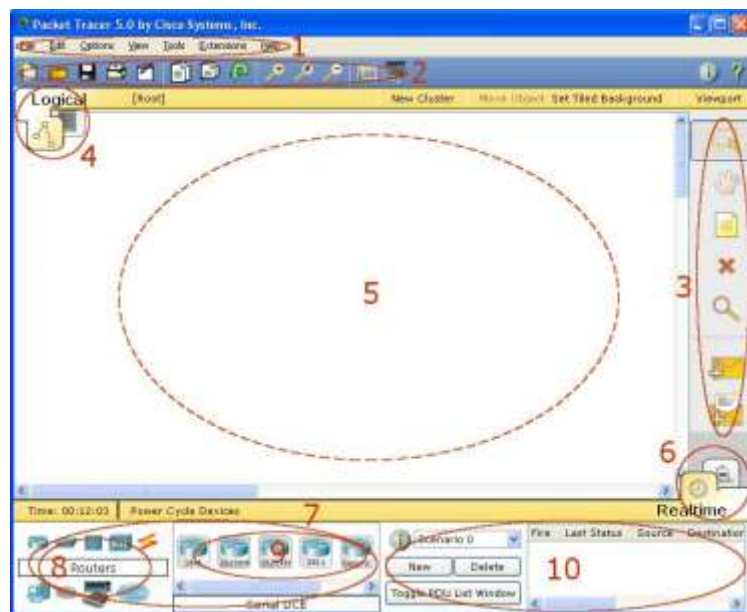


Рис. 17.31. Загальний вигляд програми Cisco Packet Tracer

Робоча область Packet Tracer складається з наступних елементів:

1. *Menu Bar* – панель, що містить меню File, Edit, Options, View, Tools, Extensions, Help.

2. *Main Tool Bar* – панель містить графічні зображення ярликів для доступу до команд меню File, Edit, View і Tools, а також кнопку Network Information.

3. *Common Tools Bar* – панель, яка забезпечує доступ до основних інструментів програми: Select, Move Layout, Place Note, Delete, Inspect, Add Simple PDU і Add Complex PDU.

4. *Logical/Physical Workspace and Navigation Bar* – панель, яка дає можливість перемикати робочу область: фізичну або логічну, а також дозволяє переміщатися за рівнями кластера.

5. *Workspace* – область для побудови мережі та її моделювання, виведення інформації і статистики.

6. *Realtime/Simulation Bar* – панель, за допомогою закладок якої включають Realtime/Simulation режим. Панель має кнопки для Power Cycle Devices, Play Control і перемикач Event List в режимі Simulation.

7. *Network Component Box* – область, де обирають пристрої та зв'язки для розміщення їх на робочому просторі. Містить також області Device-Type Selection і Device-Specific Selection.

8. *Device-Type Selection Box* – область містить доступні типи пристроїв і зв'язків в Packet Tracer. Область Device-Specific Selection вказується, залежно від обраного пристрою

9. *Device-Specific Selection Box* – область використовується для вибору пристроїв і з'єднань, необхідних для побудови в робочому просторі мережі.

10. *User Created Packet Window* – вікно, що керує пакетами, створеними в мережі під час моделювання.

Для створення топології необхідно обрати пристрій з панелі Network Component, а потім з панелі Device-Type Selection обрати тип пристрою. Після цього – натиснути ліву кнопку миші в полі робочої області програми (Workspace). Також можна перемістити пристрій безпосередньо з області Device-Type Selection – обирається модель пристрою за замовчуванням.

Для швидкого створення декількох екземплярів одного пристрою потрібно, утримуючи кнопку Ctrl, натиснути на пристрій в області Device-Specific Selection і відпустити кнопку Ctrl. Після цього можна кілька разів натиснути на робочій області для додавання копії пристрою.

У Packet Tracer представлені наступні типи пристроїв:

- маршрутизатори;
- комутатори та мости;
- хаби і повторювачі;
- кінцеві пристрої – ПК, сервери, принтери, IP-телефони;
- бездротові пристрої – точки доступу та бездротові маршрутизатори;
- решта пристроїв – хмара, DSL-модем і кабельний модем.

Для прикладу, додамо необхідні типи пристроїв у робочу область програми, як показано на рис. 17.32.

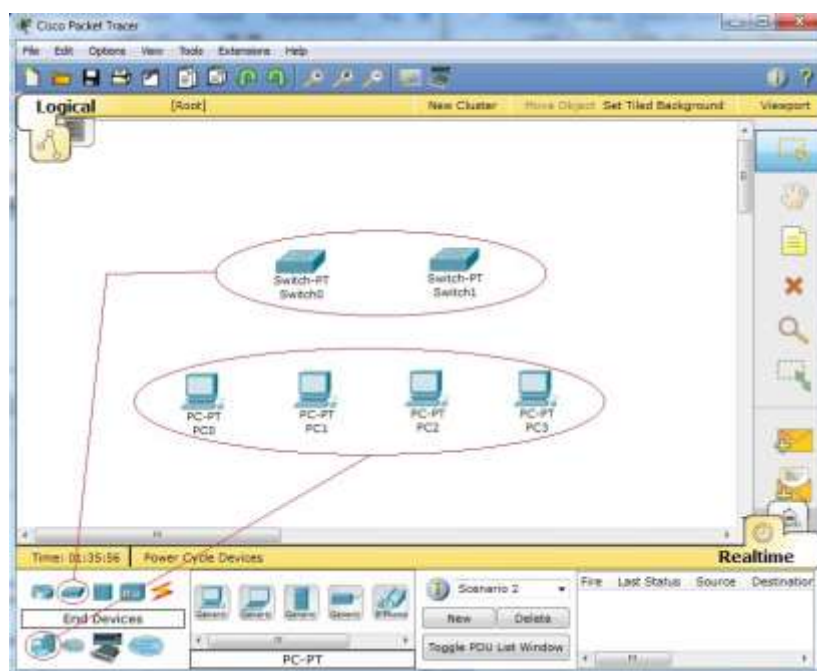


Рис. 17.32. Додавання пристроїв

При додаванні пристрою, користувач може відкрити діалогове вікно пристрою (для цього на пристрої клікають лівою кнопкою миші).

Діалогове вікно властивостей має дві вкладки:

Physical – містить графічний інтерфейс пристрою і дозволяє моделювати роботу з ним на фізичному рівні;

Config - містить всі необхідні параметри для настройки пристрою і має зручний для цього інтерфейс.

У вкладці *Config* користувач може присвоїти ім'я пристрою та встановити його параметри.

Також, в залежності від пристрою, властивості можуть мати додаткову вкладку для керування роботою обраного елемента: *Desktop* (якщо обрано кінцевий пристрій), або *CLI* (якщо обрано маршрутизатор). Видалення пристрою з робочою області відбувається кнопкою *Delete (Del)*.

З'єднаємо додані пристрої за допомогою зв'язків. Для цього оберемо вкладку *Connections* з панелі *Network Component Box*, що показана на рис. 17.33. На вкладці зображено всі можливі типи з'єднань між пристроями. Оберемо відповідний тип кабелю. Показчик миші зміниться на курсор "Connection" – у вигляді роз'єму. Натиснемо на перший пристрій та оберемо інтерфейс, з яким потрібно виконати з'єднання, а потім натиснемо на другий пристрій та виконаємо аналогічну операцію. Пристрої можна також з'єднати за допомогою *Automatically Choose Connection Type* (автоматично з'єднує елементи в мережі). Після натискання на кожному з пристроїв, які потрібно з'єднати, між пристроями з'явиться кабельне з'єднання, а індикатори на кожному кінці покажуть статус з'єднання (для інтерфейсів, що мають індикатори).



Рис. 17.33. Вкладка *Connections* з можливими з'єднаннями

Cisco Packet Tracer підтримує ряд мережевих з'єднань, що показані в табл. 17.4. Кожен тип кабелю може бути з'єднаний лише з певними типами

інтерфейсів.

Після створення мережі її топологію потрібно зберегти, обравши пункт меню File -> Save або іконку Save на панелі Main Tool Bar. Файл збереженої топології має розширення *.pkt.

Packet Tracer дає можливість моделювати роботу з інтерфейсом командного рядка (ІКР) операційної системи IOS, встановленої на всіх комутаторах і маршрутизаторах компанії Cisco.

Таблиця 17.4. Мережеві з'єднання в Packet Tracer

Тип кабелю	Опис
 Console	Консольне з'єднання, що може бути між ПК і маршрутизаторами, або комутаторами. Повинні бути виконані деякі вимоги для роботи консольного сеансу з ПК: швидкість з'єднання з обох сторін – однакова, має бути 7 біт даних (або 8 біт) для обох сторін, контроль парності – однаковий, має бути 1 або 2 ступенів біта (але вони не обов'язково повинні бути однаковими), потік даних може бути довільний для обох сторін.
 Copper Straight-through	Цей тип кабелю є стандартним середовищем передачі Ethernet для з'єднання пристроїв, які функціонують на різних рівнях OSI. Має бути з'єднаний з наступними типами портів: мідний 10 Мбіт/с (Ethernet), мідний 100 Мбіт/с (Fast Ethernet) і мідний 1000 Мбіт/с (Gigabit Ethernet).
 Copper Cross-over	Цей тип кабелю є середовищем передачі Ethernet для з'єднання пристроїв, які функціонують на одному рівні моделі OSI. Він може бути з'єднаний з наступними типами портів: мідний 10 Мбіт/с (Ethernet), мідний 100 Мбіт/с (Fast Ethernet) і мідний 1000 Мбіт/с (Gigabit Ethernet).

Тип кабелю	Опис
 Fiber	Оптоволоконне середовище використовується для з'єднання між оптичними портами (100 Мбіт/с або 1000 Мбіт/с).
 Phone	З'єднання через телефонну лінію може бути здійснено між пристроями, що мають модемні порти. Стандартне уявлення модемного з'єднання - це кінцевий пристрій (наприклад, ПК), що додзвонюється в мережеву хмару.
 Coaxial	Коаксіальний кабель використовується для з'єднання між коаксіальними портами, такими як кабельний модем, з'єднаний з хмарою Packet Tracer.
 Serial DCE and DTE	З'єднання через послідовні порти, які використовуються для зв'язків WAN. Для настройки таких з'єднань необхідно установити синхронізацію на стороні DCE-пристрою. Синхронізація DTE виконується за вибором. Сторону DCE можна визначити за іконкою «годинник» поруч з портом. При виборі типу з'єднання Serial DCE перший пристрій, до якого застосовується з'єднання, стає DCE-пристроєм, а другий, автоматично, – стороною DTE. Можливо також зворотне розташування сторін, якщо обраний тип з'єднання Serial DTE.

Підключившись до пристрою, ми можемо працювати з ним як з консоллю реального пристрою. Packet Tracer забезпечує підтримку практично всіх команд, доступних на реальних пристроях.

Підключення до ІКР комутаторів, або маршрутизаторів, можна зробити, натиснувши на пристрій і перейшовши через вікно властивостей до вкладки CLI.

Для відтворення роботи командного рядка на кінцевому пристрої

необхідно у властивостях обрати вкладку Desktop та натиснути на ярлик Command Prompt.

Робота з файлами в Packet Tracer

Packet Tracer має можливість зберігати в текстових файлах конфігурацію ряду пристроїв, зокрема маршрутизаторів та комутаторів. Для цього відкрити властивості пристрою і у вкладці Config натиснути кнопку "Export ..." для експорту конфігурації Startup Config, або Running Config. Отримаємо діалогове вікно для збереження необхідної конфігурації в файл, який буде мати розширення *.txt. Текст файлу з конфігурацією пристрою running-config.txt (ім'я за замовчуванням) аналогічний тексту інформації отриманої з використанням команди show running-config в IOS- пристроях.

Конфігурація кожного пристрою зберігається в окремому текстовому файлі. Користувач може змінювати конфігурації в збереженому файлі за допомогою текстових редакторів. Для завантаження до пристрою збережених, або відредагованих налаштувань потрібно у вкладці Config натиснути кнопку "Load ..." для завантаження необхідної конфігурації Startup Config, або кнопку "Merge ..." для завантаження конфігурації Running Config.

Приклад виконання

Розмістимо в робочій області два комутатора Switch-PT. За замовчуванням – Switch0 і Switch1 та додамо до них чотири комп'ютери з іменами PC0, PC1, PC2, PC3. З'єднаємо пристрою в мережу Ethernet, як показано на рис. 17.34. Збережемо створену топологію, натиснувши кнопку Save (в меню File).

Відкриємо властивості пристрою PC0, клікнувши по його зображенню. Перейдемо до вкладки Desktop і промодельюємо його роботу, натиснувши Comand Prompt.

Список команд отримаємо після вказання «?» і натискання Enter. Для конфігурації комп'ютера скористаємося командою ipconfig, яку запустимо в командному рядку:

ipconfig 192.168.1.2 255.255.255.0

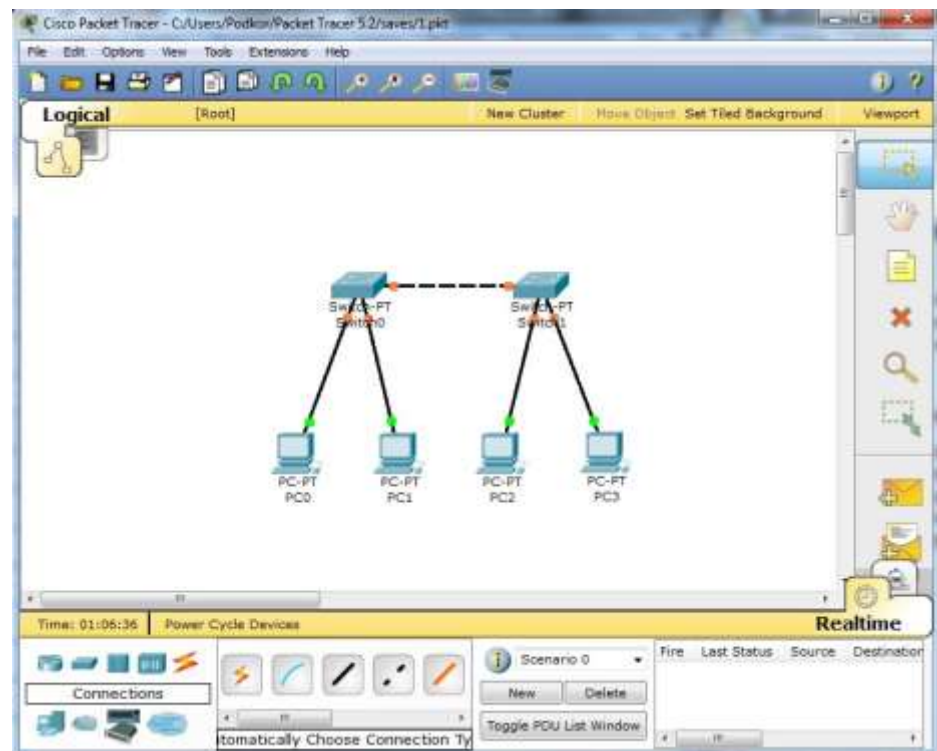


Рис. 17.34. Експериментальна модель мережі

IP-адресу і маску мережі також можна вводити в графічному інтерфейсі пристрою, як показано на рис. 17.35.

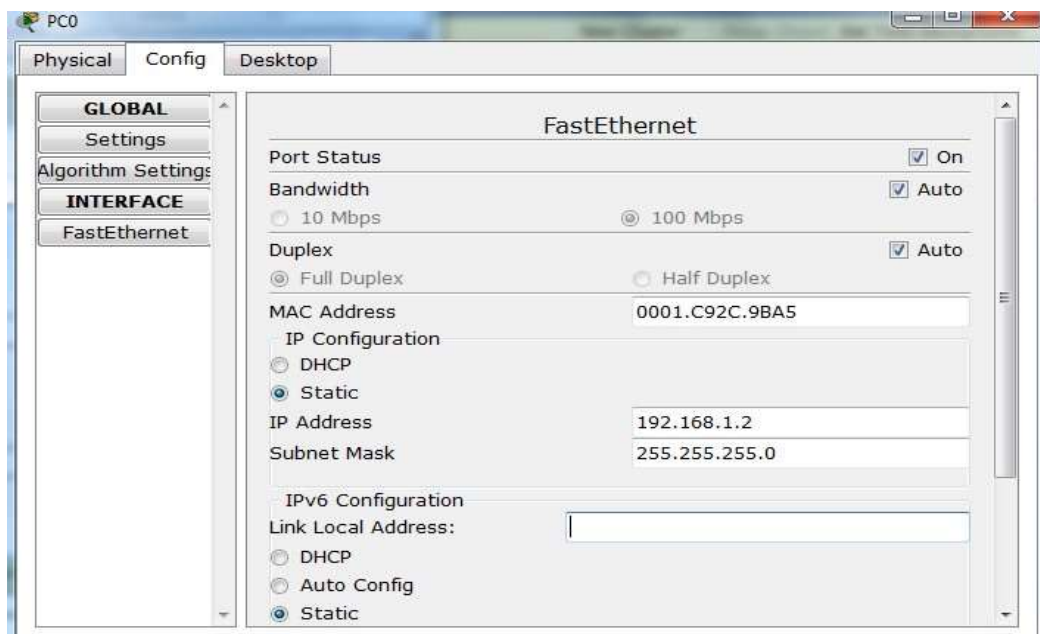


Рис. 17.35. Налаштування пристрою

Поле DEFAULT GATEWAY – адреса шлюзу. В даному випадку не заповнюється, оскільки дана топологія мережі не потребує маршрутизації. Налаштуємо кожен з пристроїв, відповідно до табл. 17.5. У кожного з комп'ютерів переглянемо призначені адреси командою `ipconfig` без параметрів.

Таблиця 17.5 Налаштування комп'ютерів підмережі

Пристрій	IP-address	Subnet mask
PC0	192.168.1.2	255.255.255.0
PC1	192.168.1.3	255.255.255.0
PC2	192.168.1.4	255.255.255.0
PC3	192.168.1.5	255.255.255.0

У Packet Tracer передбачено режим моделювання з описанням роботи утиліти `ping`. Перейдемо в даний режим, натиснувши на однойменний значок в нижньому лівому кутку робочої області, або комбінацію клавіш `Shift+S`. При цьому відкриється панель моделювання, показана на рис. 17.36, на якій відобразатиметься виконання `ping`-процесу.

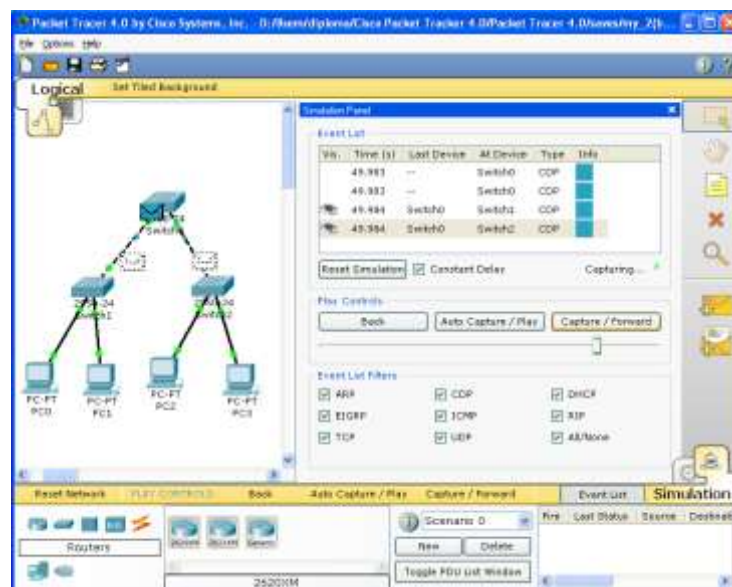


Рис. 17.36. Панель моделювання

Запустимо ping-процес та перейдемо до вкладки «Панель моделювання». На топології спроектованої мережі спостерігатимемо рух пакетів (PDU).

Кнопка «Автоматично» моделює весь ping-процес, а «Покроково» відображає його поетапно. Моделювання припиняється при завершенні ping-процесу, або при закритті вікна «Редагування» відповідного пристрою.

Якщо мережа працює, команда ping пересилатиме пакети на будь-який комп'ютер, як показано на рис. 17.37.

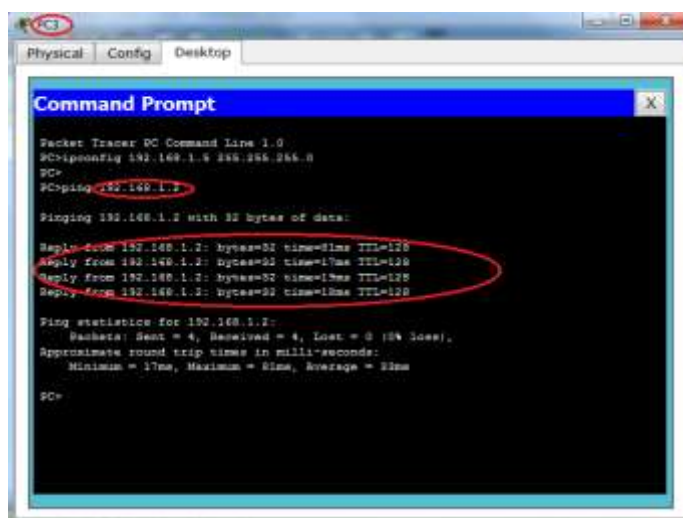


Рис. 17.37. Виконання команди ping в командному рядку

Режим моделювання «Simulation» дозволяє відслідковувати протоколи передачі даних та контролювати рівень моделі OSI, на якому даний протокол задіяний. Інформація про пакет та його структуру відображається натисканням правої кнопки миші на кольоровий квадрат в графі «Інформація», як показано на рис. 17.38.

Тестові завдання

Створіть топологію мережі, яка показана вище – на рис. 17.36. Кількість комп'ютерів визначається як 4+номер варіанту.

Відповідно до табл. 17.5, призначте всім комп'ютерам IP-адреси з масками.

Запустіть утиліту ping і перевірте стабільність з'єднання всіх об'єктів мережі за протоколом IP.

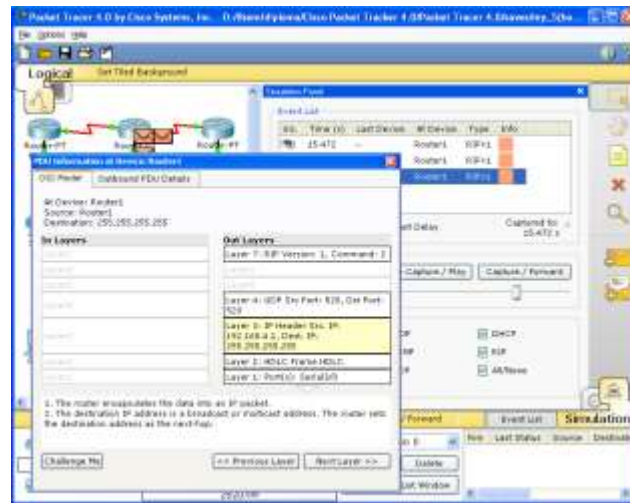


Рис. 17.38. Аналіз моделі OSI в Cisco Packet Tracer.

Переключіться в «Режим моделювання» і виконайте команду Ping з одного комп'ютера на інший. Поясніть обмін даними по протоколу ICMP між пристроями та роль протоколу ARP при цьому.

17.6. Передача повідомлень між клієнтом та сервером на базі TCP-протоколу

Мета роботи: реалізувати в середовищі C++ на базі функцій бібліотеки WinSock клієнт-серверний додаток для обміну повідомленнями в рамках протоколу TCP для ОС Windows.

Теоретичні відомості

WinSock або Windows socket – це інтерфейс прикладного програмування (API), створений для реалізації додатків в мережі на основі протоколу TCP/IP. Для роботи використовується WSOCK32.DLL. Ця бібліотека знаходиться в теці \System32 системного каталогу Windows.

Існують дві версії WinSock:

- WinSock 1.1 - підтримує тільки протокол TCP/IP;
- WinSock 2.0 - дозволяє створювати незалежні від транспортних протоколів додатки, що працюють з TCP/IP (Transmission Control

Protocol/Internet Protocol), IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange), NETBEUI (NetBios Extended User Interface).

- WinSock виділяє функції трьох типів, необхідних для створення додатку, які наведено в таблицях 17.6-17.8:
- функції Берклі (блокуючі, що зупиняють роботу програми до свого завершення, і неблокуючі, що виконуються паралельно з програмою).
- Ініціалізації і деініціалізації бібліотеки.
- Інформаційні (отримання інформації про найменування доменів, служби, протоколи Internet).

Таблиця 17.6. Функції Берклі (блокуючі)

Назва функції	Призначення
Accept	Створює новий сокет і підключає його до віддаленого комп'ютера
Closesocket	Закриває одну з сторін з'єднання
Connect	Ініціалізує з'єднання з боку вказаного сокета
Recv	Приймає дані від підключеного сокета
Recvfrom	Приймає дані від підключеного або непідключеного сокета
Send	Посилає дані підключеному сокету
Sendto	Посилає дані підключеному або непідключеному сокету

Таблиця 17.7. Функції Берклі (неблокуючі)

Назва функції	Призначення
Bind	Зв'язує віртуальний сокет з фізичним
Inetaddr	Конвертує рядок адреси в значення, яке можна використовувати в структурі in_addr
Ioctlsocket	Управляє параметрами сокета
Listen	Переводить сокет в режим прослуховування порта.
Socket	Створює точку з'єднання

Таблиця 17.8. Функції ініціалізації і деініціалізації бібліотеки WinSock

Назва функції	Призначення
WSACleanup	Завершує роботу з WinSock DLL
WSAGetLastError	Отримує інформацію про останню помилку
WSASetLastError	Встановлює повернення після помилки
WSAStartup	Ініціалізує WinSock DLL

Схема взаємодії функцій WinSock

Для реалізації поставленого завдання необхідно створити клієнтський і серверний додаток. Вони різні по організації, але є загальні дії, необхідні як для клієнтської, так і для серверної частин. Схема взаємодії функцій WinSock показана на Рис. 1. У блоці 1 (рис. 7.1) відображені загальні дії сервера і клієнта, в блоках 2 і 3, відповідно, дії сервера і клієнта, а в блоці 4 – їх взаємодія.

Блок 1 (Загальні дії)

У роботі реалізовано програмний продукт у вигляді клієнт-серверного додатку. Сервер буде працювати послідовно з встановленням логічного з'єднання на базі протоколу TCP.

Узагальнений алгоритм роботи послідовного сервера зі встановленням логічного з'єднання на базі протоколу TCP показано на рис. 17.39 та полягає в наступному:

1. Сервер створює сокет, що містить IP адресу та порт .
2. Сервер переводить створений сокет в режим прослуховування для приймання послідовних запитів від клієнтів.
3. Сервер приймає запит від певного клієнта на встановлення з'єднання, надсилає підтвердження клієнту та отримує підтвердження з'єднання від клієнта.
4. Сервер отримує запити від клієнта, опрацьовує їх, та за необхідності

надсилає відповіді клієнту.

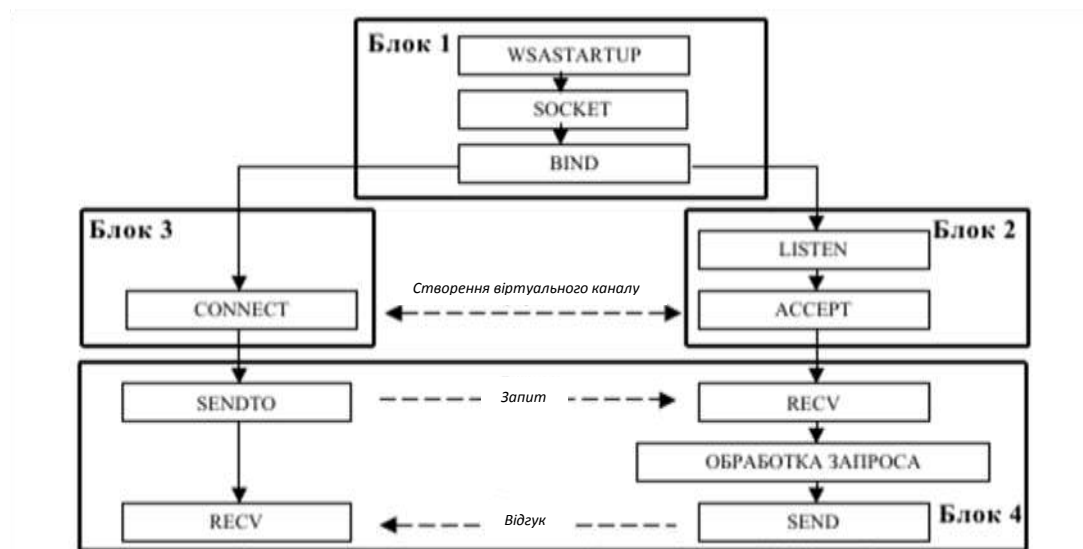


Рис. 17.39. Схема взаємодії функцій WinSock

- Після завершення обміну даними із конкретним клієнтом з'єднання закривається. А сам сервер повертається до третього пункту і очікує подальших запитів від клієнтів.

Для використання функцій WinSock 2.x, необхідно в початковий текст програми включити директиву `#include <winsock2.h>` і підготувати її до роботи функцією `WSAStartup`. При вдалому завантаженні бібліотеки, потрібно створити сокет, використовуючи функцію `Socket` і асоціювати сокет з адресною структурою `sockaddr_in`, яка містить інформацію про протокол з'єднання, IP-адреса і порт ПК.

Блок 2 (Реалізація серверної частини)

Після створення серверного сокета, він прослуховує порт функцією `Listen`, тобто перевіряє його на предмет запиту від клієнта. Після надходження запиту, сервер обробляє його функцією `Accept`.

Блок 3 (Реалізація клієнтської частини)

Після створення клієнтського сокета, він посилає запит на підключення до сервера, використовуючи функцію `Connect` і вказавши в якості одного з

параметрів IP-адресу сервера. Ця функція є блокуючою, тобто виконання програми припиниться до тих пір, поки не прийде відповідь від сервера. При позитивній відповіді сокет підключається до сервера і може з ним взаємодіяти.

Блок 4 (Реалізація обміну даними)

Процес взаємодії між клієнтом і сервером зводиться до відправки і прийому повідомлень. Для відправки повідомлення використовується функція Send, або Sendto. Їх відмінність полягає в тому, що для функції Send необхідне з'єднання (Connect, Accept), для Sendto воно необов'язкове. Для прийому повідомлень застосовують функції Recv або Recvfrom. Для функції Recvfrom з'єднання (Connect, Accept) також необов'язкове.

Бібліотека функцій WinSock

ІНІЦІАЛІЗАЦІЯ WINSOCK

Функція WSStartup ініціалізує бібліотеку WinSock. Вона завжди розташовується першою на початку роботи з WinSock. Її опис:

```
int WSStartup (WORD wVersionRequested, LPWSADATA lpWSADATA).
```

Перший параметр – це версія, що використовується. Молодший байт основна версія, старший байт розширення версії. Якщо ініціалізація відбулася, то повернеться нульове значення. Ініціалізація полягає в зіставленні номера версії і реально існуючої DLL в системі.

Другий параметр - це вказівник на структуру WSADATA, в яку повернуться параметри ініціалізації:

```
typedef struct WSADATA {  
WORD wVersion;  
WORD wHighVersion;  
char szDescription[WSADESCRIPTION_LEN+1];  
char szSystemStatus[WSASYS_STATUS_LEN+1];  
unsigned short iMaxSockets;  
unsigned short iMaxUdpDg;
```

```
char FAR *lpVendorInfo;  
}
```

Оператор `WSAGetLastError` видає код помилки при негативному результаті.

Оператор `WSACleanup()` завершує звернення до функцій WinSock. При вдалому виконанні повернеться нуль.

Приклад ініціалізації бібліотеки WinSock:

```
#include <stdio.h>  
#include <winsock2.h>  
#include <windows.h>  
  
int main()  
{  
    //Ініціалізація бібліотеки сокетів  
    WSADATA MyWSAData;  
    int ErrWSAData;  
    ErrWSAData = WSAStartup((0x0202), (WSADATA *) &MyWSAData);  
    if (ErrWSAData != 0)  
    {  
        printf("Error WSAStartup %d\n", WSAGetLastError());  
        return -1;  
    }  
    printf("WSAStartup OK!");  
}
```

СТВОРЕННЯ І ВИДАЛЕННЯ СОКЕТА

Функція створення сокета має наступний вигляд:

`SOCKET socket(int af, int type, int protocol)`, де

`af` – характеризує набір протоколів, в рамках якого взаємодіятимуть клієнт і сервер (це може бути TCP/IP, UDP, IPX і т. д.). Для протоколу TCP/IP параметр `af` повинен бути рівний `AF_INET`, що відповідає формату адреси,

прийнятому в Internet.

`type` – визначає тип комунікацій (`SOCK_STREAM`, і `SOCK_DGRAM`). Якщо параметр рівний `SOCK_STREAM`, то сокет буде використаний для передачі даних через канал зв'язку з використанням протоколу TCP/IP. Якщо ж використовується `SOCK_DGRAM`, то передача даних виконуватиметься без створення каналів зв'язку через датаграмний протокол UDP.

`protocol` – задає код конкретного протоколу з вказаного набору (заданого `af`), який буде реалізований в даному з'єднанні. Протоколи позначаються символьними константами з префіксом `IPPROTO_` (наприклад, `IPPROTO_IP` або `IPPROTO_UDP`). Допускається значення `protocol=0` (протокол не вказаний), в цьому випадку використовується значення за умовчанням для даного виду з'єднань.

Параметр, що повертається, є дескриптором сокета.

Якщо операція `socket` завершилася успішно, вихідний параметр рівний дескриптору сокета, інакше - `INVALID_SOCKET` (-1). За допомогою оператора `WSAGetLastError` можна одержати код помилки, що прояснює причину негативного результату.

Знищує сокет функція `closesocket(SOCKET s)`, де `s` – змінна типу `SOCKET`, одержана в результаті виклику функції `socket`.

Приклад створення сокета:

```
SOCKET MySocket;
MySocket = socket(AF_INET, SOCK_STREAM, IPPROTO_IP);
if (MySocket < 0)
{
    printf("Error WSAStartup %d\n", WSAGetLastError());
    WSACleanup();
    return -1;
}
```

ПРИВ'ЯЗКА АДРЕСИ ДО СОКЕТА

Перш, ніж сервер зможе використовувати сокет, він повинен зв'язати його з локальною адресою, що складається з IP-адреси і номеру порту. Якщо сервер має декілька IP-адрес, то сокет може бути зв'язаний зразу з усіма - для цього як IP-адресу слід вказати константу `INADDR_ANY` рівною нулеві, або конкретну IP-адресу.

Сервер включає в себе певний порт, тому необхідно його прописувати (зв'язувати) вручну. Клієнт теж повинен зв'язувати сокет з конкретною адресою перед його використанням, проте за нього це робить функція `Connect`, асоціюючи сокет з одним з портів, випадково вибраних з діапазону 1024-5000.

Функція зв'язування сокета з фізичною адресою має вид:

```
int bind(SOCKET s, const struct sockaddr FAR* name, int namelen) де
```

`s` – цілочисельний код дескриптора;

`name` – містить три величини: IP-адреса, код протокольного набору, номер порту, який визначає характер додатку;

`namlen` – визначає довжину другого параметра.

Структура адресної інформації має вигляд:

```
struct sockaddr_in{  
short sin_family; /* Вказується протокол*/  
unsigned short sin_port; /* Вказується порт*/  
struct in_addr sin_addr; /* Вказується IP-адреса*/  
char sin_zero[8];  
}
```

У серверній частині додатку IP-адресу можна зробити рівним `INADDR_ANY` (або `=0`), оскільки серверу не обов'язково знати свою IP-адресу. При коректному виконанні функція `bind` повертає код 0, інакше `SOCKET_ERROR=-1`.

Приклад прив'язки адреси до сокета:

```

//Зв'язування сокета
sockaddr_in LocalAddr;
int ErrBind;
LocalAddr.sin_family = AF_INET;
LocalAddr.sin_port = htons(1234); //Функція забезпечує мережевий
порядок
LocalAddr.sin_addr.s_addr = INADDR_ANY;
ErrBind = bind(MySocket (sockaddr *) &LocalAddr, sizeof(LocalAddr));
if (ErrBind != 0)
{
    printf("Error bind %d\n", WSAGetLastError());
    closesocket(MySocket);
    WSACleanup();
    return -1;
}

```

Очікування і обробка запитів на підключення клієнта

Для підключення сокета, його необхідно перевести в стан очікування функцією Listen:

```
int listen(SOCKET s, int backlog), де
```

backlog - задає максимальний розмір черги для запитів, тобто скільки запитів може бути прийняте на обслуговування без втрат. Очікуючий сокет надсилає кожному відправнику повідомлення-відгук, яке підтверджує отримання запиту на з'єднання.

Запити з черги, сформованої функцією Listen, обробляються функцією Асепт, що встановлює зв'язок з сокетом клієнта:

```
int Асепт(SOCKET s, struct sockaddr FAR *addr, int FAR* addrlen), де
s - дескриптор сокета, який прослуховує з'єднання (той, що і в Listen);
addr - вказівник на структуру, яка містить адресу;
addrlen - вказівник на довжину адреси addr.
```

При виникненні помилки повертається код INVALID_SOCKET.

Приклад використання функцій Listen і Accept:

//Переведення сервера в режим очікування запитів

```
int ErrListen;
```

```
ErrListen=listen(MySocket,5); //Довжина черги 5
```

```
if (ErrListen != 0)
```

```
{
```

```
printf("Error listen %d\n", WSAGetLastError());
```

```
closesocket(MySocket);
```

```
WSACleanup();
```

```
return -1;
```

```
}
```

 //Вибір клієнта з черги

```
SOCKET AcceptSocket;
```

```
sockaddr_in AcceptAddr;
```

```
int SizeAcceptAddr=sizeof(AcceptAddr);
```

```
AcceptSocket=accept(MySocket, (sockaddr *) &AcceptAddr,
```

```
&SizeAcceptAddr);
```

```
if (AcceptSocket !=INVALID_SOCKET)
```

```
{
```

```
printf("Error accept %d\n", WSAGetLastError());
```

```
return -1;
```

```
}
```

ПІДКЛЮЧЕННЯ ДО СЕРВЕРА

Клієнт для з'єднання з сервером використовує функцію Connect:

```
int connect(SOCKET s, const struct sockaddr FAR* name, int namelen),
```

де

s – дескриптор сокета;

name – ідентифікатор адреси місця призначення (показчик на

структуру даних);

namelen – довжина цієї адреси.

Таким чином, функція Connect повідомляє IP-адресу і номер порту віддаленої машини. Якщо адресне поле структури name містить нулі, Connect поверне помилку WSAEADDRNOTAVAIL (або SOCKET_ERROR=-1).

Приклад організації підключення до сервера:

```
//Підключення до сервера
#define PORT 1234
#define SERVERADDR "127.0.0.1"
SOCKET ClientSocket ;
sockaddr_in ServerAddr;
int ErrConnect;
ServerAddr.sin_family=AF_INET;
ServerAddr.sin_port = htons(PORT);
ServerAddr.sin_addr.s_addr = inetaddr(SERVERADDR);
ErrConnect = connect(ClientSocket, (sockaddr *) &ServerAddr,
sizeof(ServerAddr));
If (ErrConnect != 0)
{
printf("Error connect %d\n", WSAGetLastError());
return -1;
}
```

Відправка і прийом повідомлень

Для відправки повідомлень використовується функція Send, або Sendto:

int Send (SOCKET s, const char FAR* buf, int len, int flags), де

s – дескриптор сокета на віддаленій машині;

buf – вказівник на масив символів, що підлягають пересилці;

len – розмір другого параметра;

flgas – служить для цілей діагностики і управління передачею даних.

Рекомендується порівнювати його нулю.

Приклад розсилки повідомлень оператором Send:

//Пересилка даних у разі установки з'єднання

```
char Buff [255];
```

```
int ErrSend;
```

```
ErrSend = send(AcceptSocket, &Buff, sizeof(Buff), 0);
```

```
if (ErrSend == SOCKET_ERROR)
```

```
{
```

```
printf("Error send %d\n", WSAGetLastError());
```

```
closesocket(MySocket);
```

```
WSACleanup();
```

```
return -1;
```

```
}
```

Функція sendto служить для пересилки даних без установки з'єднання, тобто відправка даних йде без підтвердження отримання даних:

int Sendto (SOCKET s, const char FAR* buf, int len, int flags, const struct sockaddr FAR* to, int tolen), де

s - дескриптор відправника;

buf - вказівник на масив даних, призначених для пересилки;

len - розмір другого параметра;

flags - служить для цілей діагностики і управління передачею даних;

to - адресна структура сокета віддаленої машини;

tolen - розмір адресної структури сокета віддаленої машини.

Приклад відправки повідомлень функцією Sendto:

//Пересилка даних без установки з'єднання

```
char Buff [255];
```

```
int ErrSendTo;
```

```
SOCKET MySocket;
```

```
sockaddr_in SendToAddr;
```

```
ErrSendTo = sendto(MySocket, &Buf, sizeof(Buf), 0, &SendToAddr,
sizeof(SendToAddr));
```

```
if (ErrSendTo = SOCKET_ERROR)
{
printf("Error sendto %d\n", WSAGetLastError());
closesocket(MySocket);
WSACleanup();
return -1;
}
```

Для прийому повідомлення застосовується функція Recv або Recvfrom. Функцію Recvfrom так само, як і Sendto, можна застосовувати, не з'єднуючись з відправником.

```
int recv (SOCKET s, char FAR* buf, int len, int flags), де
s - дескриптор сокета одержувача;
buf - вказівник на масив одержаних даних;
len - розмір другого параметра;
flags - служить для цілей діагностики і управління передачею даних.
```

Приклад прийому повідомлень функцією Kescv:

```
//Отримання даних у разі установки з'єднання
```

```
char Buff [255];
SOCKET MySocket
int ErrResv;
ErrResv = resv(MySocket, &Buff, sizeof(Buff), 0);
if (ErrResv = SOCKET_ERROR)
{
printf("Error send %d\n", WSAGetLastError());
closesocket(MySocket);
WSACleanup();
return -1;
```

```
}
```

int Recvfrom (SOCKET s, const char FAR* buf, int len, int flags, const struct sockaddr FAR* to, int tolen), де

s – дескриптор одержувача;

buf – покажчик на масив одержаних даних;

len – розмір другого параметра;

flags – служить для цілей діагностики і управління передачею даних;

to – адресна структура сокета віддаленої машини;

tolen – розмір адресної структури сокета віддаленої машини.

Приклад прийом повідомлень, використовуючи функцію Recvfrom:

```
// Отримання даних без установки з'єднання
```

```
char Buff [255];
```

```
SOCKET MySocket;
```

```
int ErrRecvFrom;
```

```
sockaddr_in RecvFromAddr;
```

```
ErrSendTo = recvfrom (MySocket, &Buf, sizeof(Buf), 0, & RecvFromAddr,  
sizeof(RecvFromAddr));
```

```
If (ErrRecvFrom = SOCKET_ERROR)
```

```
{
```

```
printf("Error sendto %d\n", WSAGetLastError());
```

```
closesocket(MySocket);
```

```
WSACleanup();
```

```
return -1;
```

```
}
```

Тестові завдання

Створити програмний продукт, що працює як клієнт-серверний додаток з установкою логічного з'єднання за протоколом TCP. Завдання реалізується відповідно до варіанту, що наведений нижче.

1. Клієнт відправляє на сервер два цілі числа. Сервер перемножує ці

числа, додає номер варіанта та відсилає відповідь клієнту.

2. Клієнт відправляє на сервер два числа. Сервер підносить перше число до степені, що рівна другому числу, і відправляє відповідь клієнту.

3. Клієнт відправляє на сервер слово. Сервер рахує кількість букв і відправляє число клієнту.

4. Клієнт відправляє на сервер слово. Сервер видаляє всі голосні та відправляє відповідь клієнту.

5. Клієнт відправляє на сервер слово. Сервер видаляє всі приголосні та відправляє відповідь клієнту.

6. Клієнт відправляє на сервер два слова. Сервер їх порівнює і відправляє букви що зустрічаються в обох словах клієнту.

7. Клієнт відправляє на сервер два числа і отримує від сервера True, якщо друге число ділиться на перше без залишку (або False – в іншому разі).

8. Клієнт відправляє на сервер слово, що містить пробіли. Сервер видаляє пробіли і повертає слово клієнту без пробілів.

9. Клієнт відправляє на сервер слово з букв і цифр. Сервер видаляє букви і повертає слово клієнту.

10. Клієнт відправляє на сервер слово із букв і цифр. Сервер видаляє цифри і повертає клієнту слово.

11. Клієнт відправляє на сервер слово. Сервер читає слово справа наліво і так повертає його клієнту.

12. Клієнт надсилає серверу число в двійковій системі від 0 до 1111. Сервер конвертує число в десяткову систему та повертає клієнту.

13. Клієнт відправляє на сервер число в шістнадцятковій системі числення від 0 до FF. Сервер конвертує число в десяткову систему та повертає клієнту.

14. Клієнт відправляє на сервер час в годинах і хвилинах. Сервер отримує і визначає час, що пройшов від початку нового тисячоліття, в днях, годинах, хвилинах – та повертає клієнту.

15. Клієнт надсилає на сервер строку, що містить зірочки. Сервер зірочки видаляє і повертає строку клієнту.

16. Клієнту надсилає на сервер строку написаною латиницею. Сервер транслітерує її в кирилицю і відправляє клієнту.

17. Клієнт надсилає на сервер число, написане арабськими цифрами. Сервер записує число в римські цифри та відправляє клієнту.

18. Клієнт надсилає на сервер число, написане римськими цифрами. Сервер записує число арабськими цифрами та відправляє клієнту.

17.7. Передача повідомлень між клієнтом та сервером на базі UDP-протоколу

Мета роботи: реалізувати в середовищі Python на базі бібліотеки `socket` клієнт-серверний додаток для обміну повідомленнями в рамках протоколу UDP для ОС Windows.

Інтерфейс сокетів

Інтерфейс відноситься до транспортного рівня стеку протоколів TCP/IP. Крім протоколів, що використовують для взаємодії різних хостів, використовують також інтерфейс.

Загалом, інтерфейс транспортного рівня дозволяє писати програми для мережі.

Інтерфейс, що називається інтерфейсом сокетів, вперше з'явилися в Операційній системі Berkeley Unix в 1982 р. Сокети, що характеризуються IP-адресою та портом, можна розглядати як файл спеціального типу. Дані, що заносяться в цей файл, автоматично передаються по мережі на інший комп'ютер, що читає їх. Тобто, опрацювання даних відбувається подібно до роботи з текстовим файлом. А вся абстракція – взаємодії з мережею – прихована від програміста, що дуже зручно. Тому сокети використовують в операційних системах Unix, Linux, Windows, а також підтримуються мовами програмування. Тобто, сокети стали стандартом взаємодії програм з

транспортним рівнем стеку протоколів TCP/IP.

Розглянемо операції сокетів Берклі, оскільки багато сучасних сокетів мають аналогічні операції.

Операції з сокетами Берклі діляться на чотири типи: перший тип – це створення сокетів (методи Socket, Bind, Listen); другий – установка з'єднання (методи Connect, Accept); третій – передача даних (методи Send, Receive) і четвертий – закриття з'єднання (метод Close).

Сокети Берклі використовують модель «клієнт-сервер» де «сервер» – це програма, що постійно працює на комп'ютері з відомою IP-адресою і портом.

Розглянемо роботу з сокетами Берклі у межах моделі «клієнт-сервер», тобто створимо Socket на сервері і організуємо прийом запитів на з'єднання від клієнтів. Метод Socket створює об'єкт Socket (в найпростішому випадку це файл), далі метод Bind поєднує сокет з певною IP-адресою та портом. Після цього метод Listen вказує, що сервер готовий приймати з'єднання по мережі та створює чергу для з'єднань. Якщо сервер отримає більше запитів на з'єднання, ніж вказано у Listen, то нові запити будуть відкидатися.

Далі сервер викликає метод Accept і переходить в режим очікування, тобто сервер очікує на з'єднання від клієнта.

У свою чергу клієнт спочатку викликає метод Socket для створення сокета з довільною IP адресою та портом, що призначаються операційною системою і це дозволяє не викликати метод Bind на клієнті. Далі викликається метод Connect, що вказує IP-адресу і порт сервера з якими встановлюється з'єднання та яким відправляється запит.

Метод Connect встановлює з'єднання лише у випадку роботи лише з TCP-протоколом. У випадку використання UDP-протоколу в сокеті (SOCK_DGRAM) цей виклик зберігає тільки інформацію сервера і порту для відправки повідомлень та їх отримання. Тобто, виклик можна і не використовувати, а застосувати для роботи з сокетами API, що передбачають явне вказування інформації про сервер (sendto/recvfrom).

Надалі, щоб інші клієнти могли з'єднатися з сервером, використовуючи ту ж IP-адресу та порт, створюється копія сокета. Тому наступне з'єднання встановлюватиметься не з початковим сокетом, а з його копією. І наступні дані передаватимуться вже через копію сокета. Клієнт готує дані та передає їх методом Send через мережу серверу, а сервер за допомогою методу Receive читає їх. Сервер і клієнт можуть обмінюватися даними протягом кількох сеансів. Після цього обміну клієнт викликає метод Close, що розриває з'єднання.

Сокети використовуються для програм, що взаємодіють між собою за моделлю "клієнт-сервер" на транспортному рівні (взаємодія на прикладному рівні, так звані «web-socket» у межах даної роботи не розглядаються). Таким чином, протоколи транспортного рівня приховані від програміста, а вся взаємодія по мережі реалізується через інтерфейс сокетів.

Сокети дейтаграм, що використовують UDP-протокол, є засобом ненадійного обміну пакетами даних. Ненадійність пов'язана з відсутністю підтвердження їх доставки за призначенням. Тому один і той же пакет даних може доставлятися декілька разів, що є їх недоліком. Разом з тим, сокети дейтаграм використовуються для відправки даних, що містять окремі пакети, або записи. Сокети дейтаграм також використовуються для розсилки пакета за кількома адресами одночасно.

Програма-сервер, код та опис

На прикладі нижче, мовою Python 3.6, показано обмін даними у моделі «клієнт-сервер» з використанням сокетів. Розглянемо роботу програми-сервера.

Програма-сервер спочатку створює сокет. При його створенні, як константи, обираються протоколи мережевого рівня. У прикладі вказано IP-протокол мережевого рівня та TCP-протокол транспортного рівня. Створений сокет методом Bind прив'язується до IP-адреси і порту. Викликається метод Listen, що говорить про готовність до прийому даних, і що черга з'єднання

буде з одного елемента. Далі викликається метод `Accept` і програма переходить в режим очікування запитів на з'єднання від клієнтів. Повернення з методу `Accept` відбувається лише після встановлення з'єднання з клієнтом. Далі методом `Receive` в циклі сервер по 1024В читає дані та відправляє ті ж дані назад клієнту. Вихід з циклу відбувається по закінченню даних. Далі з'єднання закривається.

Для довідки: типи протоколів, що найчастіше використовуються в сокетах. На мережевому рівні це протоколи IPv4 та IPv6 – позначаються `socket.AF_INET` та `socket.AF_INET6`, відповідно. На транспортному рівні це протоколи TCP та UDP – позначаються `socket.SOCK_STREAM` та `socket.SOCK_DGRAM`, відповідно (перший передає потік бітів, а другий – дейтаграми).

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind(('127.0.0.1',12345))
s.listen(1)
conn,addr = s.accept()
while True:
    data = conn.recv(1024)
    if not data: break
    conn.sendall(data)
conn.close()
```

Програма-клієнт, код

Розглянемо роботу програми-клієнта. Програма-клієнт спочатку створює Socket з використанням протоколу IP на мережевому рівні та UDP-протоколу на транспортному. Метод `Bind` не викликається для з'єднання з IP-адресою і портом, оскільки це робиться бібліотеками Python і Windows автоматично. Далі методом `Connect` відправляється запит на установку

з'єднання з сервером за вказано IP-адресою та портом сервера. Після встановлення з'єднання серверу побайтово відправляється повідомлення 'Hello world' та приймаються від нього ті ж дані за методом Recv, закриваємо з'єднання (метод Close) та друкуємо дані.

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.connect(('127.0.0.1',12345))
s.sendall(b 'Hello world')
data = s.recv(1024)
s.close ()
print ('Received data frm server:' , repr (data))
```

Тестові завдання

Створити програмний продукт у вигляді додатку, що реалізує клієнт-серверну архітектуру. Сервер працює з запитами клієнтів послідовно, без встановлення логічного з'єднання, тобто – використовуючи протокол UDP.

Завдання реалізується відповідно до варіанту, що наведений нижче.

1. Клієнт відправляє в консольному режимі з клавіатури слово завершуючи відправку натисканням «Enter». Сервер визначає довжину слова і повертає у вигляді числа клієнту.
2. Клієнт відправляє в консольному режимі з клавіатури слово завершуючи відправку натисканням «Enter». Сервер видаляє перший та останній символ і повертає слово клієнту.
3. Клієнт відправляє в консольному режимі з клавіатури слово завершуючи відправку натисканням «Enter». Сервер, отримавши слово, читає його справа наліво і відправляє клієнту.
4. Клієнт відправляє в консольному режимі з клавіатури слово, що

містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, видаляє цифри і відправляє слово клієнту.

5. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, замінює всі поділіся цифри на символ * і відправляє слово клієнту.

6. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає чи його довжина кратна чотирьом. У цьому разі видаляються всі букви і відправляє слово клієнту.

7. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає його довжину. Якщо довжина кратна десяти, то рахує кількість дужок і відправляє число клієнту.

8. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає його довжину. Якщо довжина кратна двом то початок слова міняється місцями з закінченням і нове слово відправляється клієнту.

9. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає його довжину і видаляє всі символи латинського алфавіту. Нове слово і кількість видалених символів сервером відправляються клієнту.

10. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає його довжину. Якщо довжина перевищує 10 символів, видаляє всі символи написані кирилицею. Нове слово і кількість видалених символів сервер повертає клієнту.

11. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, замінює парні та непарні склади і відправляє нове слово клієнту.

12. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає його довжину. Якщо вона перевищує 10 символів, то слово скорочується та відправляється клієнту.

13. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає його довжину. Якщо довжина перевищує 11 символів, то в слові 1 та 3 символи замінюються зірочками. Змінена слово відправляється клієнту.

14. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, визначає кількість цифр. Якщо кількість парна, відправляє клієнту цю кількість цифр як число.

15. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, замінює в слові кожен непарний символ на зірочку і відправляє нове слово та кількість зірочок клієнту.

16. Клієнт відправляє в консольному режимі з клавіатури слово, що містить букви і цифри, завершуючи відправку натисканням «Enter». Сервер, отримавши слово, розбиває його на два рівні (якщо парне слово) по довжині і відправляє їх клієнту.

Контрольні запитання до розділу

1. Яке призначення функції автоаналізу пакету NetCracker?

2. Як при використанні функції автоаналізу здійснюється «об'єднання» пристроїв?
3. Як при використанні функції автоаналізу здійснюється «поділення» пристрою, що має декілька IP-адрес, на два пристрої?
4. Як при використанні функції автоаналізу здійснюється ручне додавання пристроїв?
5. Яке призначення мітки пристроїв у вигляді жовтої стрілки при виконанні автоаналізу?
6. Яке призначення інструменту IP Planner в NetCracker?
7. Який порядок використання IP Planner в NetCracker?
8. Як і для чого встановлюються статистичні індикатори?
9. Як здійснюється налаштування властивостей статистичних індикаторів?
10. Де відображається системний час?
11. Яке призначення блоку Utilization Graph?
12. Які можливості для упорядкування вторинних вікон передбачені в NetCracker?
13. Як імітується зростання інтенсивності запитів від клієнтів?
14. Яке призначення стандартного звіту Network Devices Statistics?
15. Що означають поняття «сумісність» і «несумісність» пристроїв мережі?
16. Як здійснюється пошук сумісного обладнання в NetCracker?
17. Які типи пристроїв використовуються при побудові ЛОМ?
18. Як здійснюється з'єднання приладів мережі в проекті NetCracker?
19. Як задається трафік між робочими станціями мережі в NetCracker?
20. Які параметри інформаційного пакету можуть бути змінені в NetCracker?
21. Як додати до схеми мережі фон в NetCracker?
22. Які можливості надає NetCracker для роботи з рівнями ієрархічної

- структури проекту?
23. Як перейменувати вікно проекту NetCracker?
 24. Які інструментальні засоби надає NetCracker для анотування проекту?
 25. Як відобразити канал зв'язку від одного пристрою до іншого в NetCracker?
 26. Як створюється новий рівень ієрархії в проекті NetCracker?
 27. Яке призначення папки Generic Devices в браузері пристроїв NetCracker?
 28. Який порядок створення мережі типу «клієнт/сервер»?
 29. Які відомості про роботу мережі відображаються в звіті Module Statistical Items?
 30. Яке призначення Device Factory?
 31. Які типи шини введення/виведення передбачені для пристроїв при додаванні в базу пристроїв NetCracker? Дайте коротку характеристику основних з них.
 32. Чим визначається те, в якій з базових мережевих технологій (Ethernet, Token Ring, FDDI) може використовуватись пристрій, що додається в базу пристроїв NetCracker?
 33. Як в базі пристроїв NetCracker переглянути ті, що були додані користувачем?
 34. Як і за якими критеріями може здійснюватись пошук в базі пристроїв NetCracker?
 35. Як додати в проект NetCracker групу однотипних об'єктів? Як задається порядок їх розташування?
 36. Як здійснюється управління анімацією в NetCracker?
 37. Як задаються параметри анімації в NetCracker?
 38. Як переглянути підрівень проекту NetCracker?
 39. Як розірвати (відновити) лінію зв'язку проекту NetCracker?
 40. Які відомості про параметри інформаційних пакетів і як можуть бути

отримані?

41. Як здійснюється вигин лінії зв'язку в проекті NetCracker?
42. Як в проекті мережі здійснюється перейменування об'єктів?
43. Чи підтримує WinSock 1.1 роботу з протоколом UDP?
44. Яка версія WinSock дозволяє працювати з IPX/SPX?
45. Опишіть загальні дії сервера і клієнта.
46. Як класифікуються функції WinSock?
47. Чи вірно твердження, що функція assert є неблокуючою?
48. Як реалізується ініціалізація WinSock?
49. Що таке сокет і як він створюється?
50. Для чого призначена функція bind?
51. Які функції і як використовуються для відправки і отримання повідомлень?
52. Який зв'язок між датаграмою та UDP-протоколом?
53. У чому відмінності між UDP та TCP-протоколом?
54. Які мережеві додатки використовують UDP-протокол? Наведіть приклади.
55. Для чого необхідні IP адреси в локальній мережі?
56. Що таке сокет та які існують типи сокетів?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Таненбаум Э. Компьютерные сети : моногр. / Э. Таненбаум / пер. с англ. – 4-е изд. – СПб. : Питер, 2003. – 992 с.
2. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: моногр. / В. Г. Олифер, Н. А. Олифер. – СПб. : Питер, 2001. – 672 с.
3. Блейхут Р. Теория и практика кодов, контролирующих ошибки : моногр. / Р. Блейхут / пер. с англ. – М. : Мир, 1986. – 576 с.
4. Уилсон Эд. Мониторинг и анализ сетей. Методы выявления неисправностей : моногр. / Эд. Уилсон / пер. с англ. – М. : Лори, 2002. – 368 с.
5. Фролов А. В., Фролов Г. В. Локальные сети персональных компьютеров. Использование протоколов IPX, SPX, NETBIOS : моногр. / А. В. Фролов, Г. В. Фролов /– М. : Диалог-МИФИ, 1995. – 160 с.
6. Линдберг К. Руководство администратора Novell NetWare 5 для профессионалов : моногр. / К. Линдберг / пер. с англ. – СПб. : Питер, 2000. – 496 с.
7. Вишневский А. В. Служба каталога Windows 2000 : учеб. курс / А. В. Вишневский / – СПб. : Питер, 2001. – 464 с.
8. Крол Эд. Все об Internet : моногр. / Эд. Крол / пер. с англ. – Киев : Изд. группа BHV, 1996. – 524 с.
9. Стивенс У. Р. Протоколы TCP/IP: практическое рук. / У. Р. Стивенс / пер. с англ. – СПб. : «Невский диалект» – «БХВ-Петербург», 2003. – 672 с.
10. Альбитц П., Ли К. DNS и BIND : моногр. / П. Альбитц, К. Ли / пер. с англ. – СПб. : Символ-Плюс, 2002. – 696 с.
11. Спейнаур С., Куэрсиа В. Справочник Web-мастера / С. Спейнаур, В. Куэрсиа / пер. с англ. – Киев : Изд. группа BHV, 1997. – 368 с.
12. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи / Дж Кларк, Дж Кейн / пер. с англ. – М. : Радио и

связь, 1987. – 392 с.

13. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки : моногр. / У. Питерсон, Э. Уэлдон / пер. с англ. – М. : Мир, 1976. – 600 с.

14. Фролов А. В., Фролов Г. В. Локальные сети персональных компьютеров. Монтаж сети, установка программного обеспечения : учеб. / А. В. Фролов, Г. В. Фролов – М. : Диалог-МИФИ, 1995. – 176 с.

15. Фролов А. В., Фролов Г. В. Локальные сети персональных компьютеров. Работа с сервером Novell NetWare : учеб. / А. В. Фролов, Г. В. – М. : Диалог-МИФИ, 1993. – 168 с.

16. Кулаков Ю. А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование : моногр. / Ю. А. Кулаков, С. В. Омелянский – Киев : Юниор, 1999. – 544 с.

17. Гаскин Д. Администрирование Novell NetWare 6.0/6.5: моногр. / Д. Гаскин / пер. с англ. – СПб. : «БХВ-СПб», 2003. – 1056 с.

18. Шапиро Д., Бойс Д. Windows 2000 Server. Библия пользователя: моногр. / Д. Шапиро, Д. Бойс / пер. с англ. – СПб. : Диалектика, 2002. – 912 с.

19. Харт Д. М. Системное программирование в среде Win32 : моногр. / Д. М. Харт / пер. с англ. – М. : Изд. дом «Вильямс», 2001. – 464 с.

20. Петерсен Р. Linux: руководство по операционной системе : моногр. / Р. Петерсен / пер. с англ. – Киев. : Изд. группа BHV, 1997. – 688 с.

21. Дунаев С. UNIX-сервер: настройка, конфигурирование, работа в операционной среде, Internet-возможности : моногр. / С. Дунаев / М. : Диалог-МИФИ, 1998. 310 с.

22. Хант К. TCP/IP. Сетевое администрирование : монографія / К. Хант / пер. с англ. – СПб. : Символ-Плюс, 2004. – 816 с.

23. Сліпченко В. Г. Локальні комп'ютерні мережі. Проектування, використання та програмування: навч. посіб. / В. Г. Сліпченко, В. І. Гайдаржи, В. А. Лабжинський. – Київ: ІВЦ «Політехніка», 2002. – 184 с.

25. Net Cracker 4.1. User Manual. Нормативні матеріали:

[Электронный ресурс]. – Режим доступа : <http://soft-landia.ru/netcracker.html>. –
Назва з екрана.

26. Cisco Packet Tracer. Лабораторная работа 5: [Электронный ресурс].– Режим доступа: <https://studfiles.net/donntu/145/folder:11411/#5682479>
– Назва з екрана.