

ЗМІСТ

ЗМІСТ.....	1
ВСТУП.....	4
Модуль 1.....	4
1. Теоретичні відомості.....	4
1.1. Еволюція обчислювальних мереж.....	4
1.1.1. Поява перших обчислювальних машин.....	5
1.1.2. Мультипрограмування	5
1.1.3. Глобальні обчислювальні мережі	7
1.1.4. Роль персональних комп'ютерів в еволюції комп'ютерних мереж	10
1.1.5. Еволюція мережних операційних систем	10
1.2. Характеристики і стандарти мереж передачі та розподілу даних.....	13
1.2.1. Вимоги до мереж	13
1.2.2. Модульність і стандартизація	18
1.2.3. Джерела стандартів	19
1.2.4. Стандарти Internet.....	20
1.2.5. Поява стандартних технологій локальних мереж.....	22
1.2.6. Модель OSI.....	22
1.2.7. Багаторівневий підхід. Декомпозиція задач мережної взаємодії.....	23
1.2.8. Загальна характеристика моделі OSI.....	24
1.2.10. Стандартизація мереж. Поняття "Відкрита система"	32
1.2.11. Протокол. Інтерфейс. Стек протоколів	33
1.3. Основні задачі побудови мереж	37
1.3.1. Зв'язок комп'ютера з периферійними пристроями	37
1.3.2. Зв'язок двох комп'ютерів.....	39
1.3.3. Клієнт, редиректор і сервер.....	40
1.3.4. Фізична передача даних по лініях зв'язку.....	41
1.3.5. Задачі зв'язку декількох комп'ютерів. Топологія фізичних зв'язків.....	42
1.3.6. Адресація вузлів мережі.....	48
1.4. Комутація і мультиплексування	51
1.4.1. Узагальнена задача комутації.....	51
1.4.2. Мультиплексування і демультиплексування.....	55
1.4.3. Різні підходи до виконання комутації.....	58
1.4.4. Порівняння комутації.....	63
1.4.5. Пропускна спроможність мереж з комутацією пакетів.....	64
1.4.6. Ethernet – приклад стандартної технології комутації пакетів.....	66
1.4.7. Дейтаграмна передача.....	68
1.4.8. Віртуальні канали в мережах з комутацією пакетів.....	69
1.5. Практичні відомості.....	70
1.6. Контрольні запитання.....	72
1.7. Задачі і вправи.....	74
1.4. Список літератури для модуля 1.....	75
Модуль 2.....	75
2. Теоретичні відомості.....	75
2.1. Прикладне мережне програмне забезпечення	75
2.1.1. Перспективи розвитку розподіленої обробки даних	75
2.1.2. Загальна характеристика стану ринку СКБД	75
2.1.3. Засоби розробки прикладних програм	76
2.2. Системи керування базами даних	77
2.2.1. Технологія файл-сервер і клієнт-сервер	77

2.2.2. Розподілені бази даних.....	79
2.2.3. Технологія аналітичної обробки даних OLAP та багатовимірні сховища даних.....	81
2.2.4. Об'єктно-орієнтовані БД.....	82
2.2.5. Мова побудови запитів SQL.....	83
2.3. База даних як модель предметної області.....	83
2.4. Моделі організації даних.....	84
2.4.1. Ієрархічна модель.....	84
2.4.2. Мережна модель.....	85
2.4.3. Багатовимірна модель даних.....	86
2.5. Реляційна модель даних.....	86
2.5.1. Концепція реляційної моделі даних.....	86
2.5.2. Реляційна алгебра і реляційне счислення. Операції над відношеннями.....	87
2.6. Проектування реляційних баз даних.....	89
2.6.1. Процес нормалізації відношень.....	89
2.6.2. Перша нормальна форма.....	89
2.6.3. Типи функціональних залежностей.....	89
2.6.4. Друга нормальна форма.....	90
2.6.5. Третя нормальна форма та подальша нормалізація.....	91
2.7. Семантичне моделювання даних, ER-діаграми.....	92
2.8. Засоби проектування даних.....	93
2.9. Загальна характеристика баз знань та експертних систем.....	93
2.10. Аналіз мереж методами теорії масового обслуговування.....	95
2.10.1. Математичні моделі систем масового обслуговування.....	97
2.10.2. Основні співвідношення теорії масового обслуговування.....	99
2.11. Функціональні ролі комп'ютерів в мережі.....	102
2.11.1. Багатошарова модель мережі.....	102
2.11.2. Однорангові мережі.....	105
2.11.3. Мережі з виділеним сервером.....	106
2.11.4. Гібридна мережа.....	107
2.11.5. Мережні служби і операційна система.....	108
2.12. Структуризація мереж.....	109
2.12.1. Причини структуризації транспортної інфраструктури мереж.....	109
2.12.2. Фізична структуризація мережі.....	109
2.12.3. Логічна структуризація мережі.....	111
2.13. Конвергенція комп'ютерних і телекомунікаційних мереж.....	115
2.13.1. Загальна структура телекомунікаційної мережі.....	116
2.13.2. Мережі операторів зв'язку.....	117
2.13.3. Інфраструктура.....	120
2.13.4. Переваги і проблеми використання мереж.....	124
2.14. Практичні відомості.....	125
2.14.1. Приклад розрахунку затримки передачі даних.....	125
2.14.2. Розрахунок коректності конфігурації мережі Ethernet.....	126
2.14.3. Розрахунок коректності конфігурації мережі Fast Ethernet.....	129
2.14.4. Обмеження на кількість повторювачів.....	129
2.14.5. Проектування логічної структури бази даних.....	130
2.15. Задачі і вправи.....	138
2.14. Список літератури для модуля 2.....	139
3. Методичні вказівки для курсового проектування по дисципліні «Комп'ютерні системи і мережі».....	140
3.1. Цілі і задачі курсового проектування.....	140
3.2. Тематика і зміст курсових проектів.....	140
3.3. Завдання по курсовому проектуванню.....	140
3.4. Правила оформлення пояснювальної записки.....	141

3.5. Правила оформлення графічного матеріалу.....	141
3.6. Методика курсового проектування.....	141
3.6.1. Загальні теоретичні відомості про обчислювальні мережі.....	141
3.6.2. Проектування локальної обчислювальної мережі.....	143
3.6.3. Проектування апаратного і програмного забезпечення для використання глобальних обчислювальних мереж.....	149
3.6.4. Розрахунок економічної ефективності від упровадження обчислювальної мережі.....	156
3.6.5. Розрахунок суми витрат на розробку, впровадження і експлуатацію обчислювальної мережі.....	157
3.6.6. Вибір методики розрахунку економічної ефективності.....	158
3.6.7. Порядок захисту.....	159

ВСТУП

Навчальний посібник призначений для вивчення дисципліни спеціальності 8.091501 “Комп’ютерні системи та мережі”. Посібник підготовлений для роботи за кредитно-модульною системою і складається з двох взаємопов’язаних модулів, зміст яких перекликається і тісно пов’язаний один з одним.

У модулі 1 розглянуті еволюційні аспекти комп’ютерних мереж, їх різновиди, структура та типові характеристики. Модуль 1 призначений для вивчення загальних питань побудови комп’ютерних мереж та мереж передачі даних і, таким чином, практично забезпечує вивчення першого розділу дисципліни “Комп’ютерні системи та мережі”, “Мережі ЕОМ” та інших того ж профілю.

Модуль 2 базується на загальнотеоретичних основах побудови мереж передачі даних, тобто матеріалах, які розглянуті у модулі 1. У модулі 2 розглянуті прикладні аспекти функціонування мереж передачі даних, забезпечення роботи мережних прикладних програм та гарантованого надання послуг з потрібною якістю користувачам мережі. Матеріал модуля 2 може використовуватися при вивченні прикладних, експлуатаційних та техніко-економічних дисциплін даного профілю.

Контрольні запитання, задачі і вправи, приклади побудовані так, щоб студент, який вивчає цю спеціальність, отримав чіткі уявлення як про загальні закономірності розвитку мереж передачі даних, їх стан та перспективи, так і про взаємозв’язок окремих елементів та вузлів, їх взаємний вплив одне на одного і на загальну ефективність системи в цілому. Даються методичні вказівки по курсовому проектуванню з дисциплін, назви яких наведені вище.

Модуль 1

1. Теоретичні відомості

1.1. Еволюція обчислювальних мереж

Мережі передачі даних, звані також обчислювальними або комп’ютерними мережами, є результатом еволюції двох найважливіших науково-технічних галузей сучасної цивілізації – комп’ютерних і телекомунікаційних технологій (рис. 1.1):

1. З одного боку, мережі передачі даних є окремим випадком розподілених обчислювальних систем, в яких група комп’ютерів погоджено виконує набір взаємозв’язаних завдань, обмінюючись даними в автоматичному режимі.

2. З іншого боку, комп’ютерні мережі можуть розглядатися як засіб передачі інформації на великі відстані, для чого в них застосовуються методи кодування і мультиплексування даних, що отримали розвиток в різних телекомунікаційних системах.

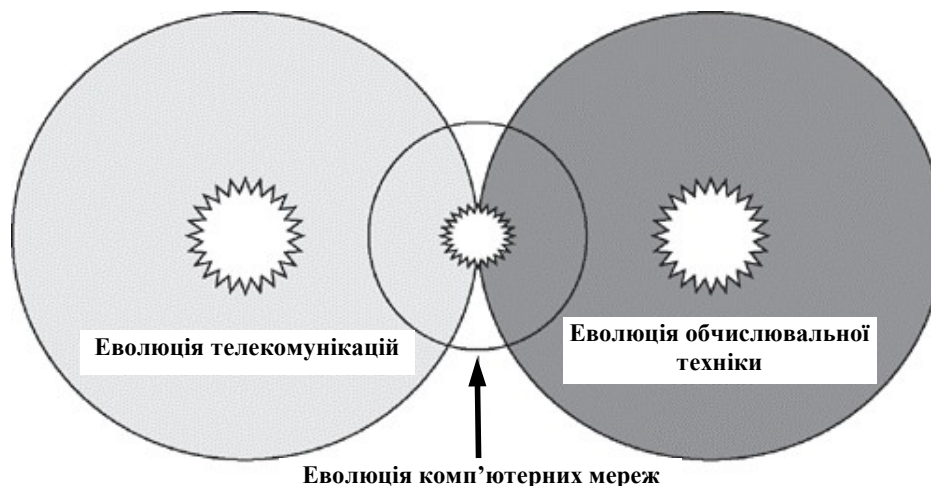


Рис. 1.1. Еволюція комп’ютерних мереж на стику обчислювальної техніки і телекомунікаційних технологій.

Отже:

– комп’ютерна мережа – це набір комп’ютерів, зв’язаних комунікаційною системою і забезпечених відповідним програмним забезпеченням, яке надає користувачам мережі доступ до ресурсів цього набору комп’ютерів;

– мережу можуть утворювати комп’ютери різних типів – невеликі мікропроцесори, робочі станції, міні-комп’ютери, персональні комп’ютери або суперкомп’ютери;

– передачу повідомлень між будь-якою парою комп’ютерів мережі забезпечує комунікаційна система, яка може включати кабелі, повторювачі, комутатори, маршрутизатори і інші пристрої;

– комп'ютерна мережа дозволяє користувачеві працювати з своїм комп'ютером як з автономним, і додає до цього можливість доступу до інформаційних і апаратних ресурсів інших комп'ютерів мережі.

1.1.1. Поява перших обчислювальних машин

Ідея комп'ютера була запропонована англійським математиком Чарльзом Бебіджем (Charles Babbage) в середині дев'ятнадцятого століття. Проте його механічна "аналітична машина" по-справжньому так і не запрацювала. Справжнє народження цифрових обчислювальних машин відбулося незабаром після закінчення другої світової війни. В середині 40-х були створені перші лампові обчислювальні пристрої. Для цього періоду характерні наступні тенденції:

- комп'ютер був скоріше предметом дослідження, а не інструментом для вирішення яких-небудь практичних завдань з інших областей;
- одна і та ж група людей брала участь і в проектуванні, і в експлуатації, і в програмуванні обчислювальної машини;
- програмування здійснювалося виключно на машинній мові;
- не було ніякого системного програмного забезпечення, окрім бібліотек математичних і службових підпрограм;
- операційні системи ще не з'явилися, всі завдання організації обчислювального процесу вирішувалися вручну кожним програмістом з пульта управління.

З середини 50-х років почався наступний період в розвитку обчислювальної техніки, пов'язаний з появою нової технічної бази – напівпровідникових елементів. У цей період:

- виросла швидкодія процесорів, збільшилися об'єми оперативної і зовнішньої пам'яті;
- комп'ютери стали надійнішими;
- з'явилися перші алгоритмічні мови, і, таким чином, до бібліотек математичних і службових підпрограм додався новий тип системного програмного забезпечення – транслятори;
- були розроблені перші системні програми, що управляли, – монітори, які автоматизували всю послідовність дій оператора по організації обчислювального процесу.

Програмні монітори – перші операційні системи

Програмні монітори це – прообраз сучасних операційних систем, вони стали першими системними програмами, призначеними не для обробки даних, а для управління обчислювальним процесом. В ході реалізації моніторів була розроблена формалізована мова управління завданнями, за допомогою якої програміст повідомляв систему і оператора, які дії і в якій послідовності він хотів би виконати на обчислювальній машині. Типовий набір директив зазвичай включав ознаку початку окремої роботи, виклик транслятора, виклик завантажувача, ознаки початку і кінця початкових даних.

Оператор складав пакет завдань, які надалі без його участі послідовно запускалися на виконання монітором. Крім того, монітор був здатний самостійно обробляти найбільш поширені аварійні ситуації, що виникали при роботі призначених для користувача програм, такі як відсутність початкових даних, переповнювання регістрів, ділення на нуль, звернення до неіснуючої області пам'яті і т.д.

1.1.2. Мультипрограмування

Наступний важливий період розвитку операційних систем відноситься до 1965-1975 років. В цей час в технічній базі обчислювальних машин відбувся перехід від окремих напівпровідникових елементів типу транзисторів до інтегральних мікросхем, що відкрило шлях до появи наступного покоління комп'ютерів, представником якого є, наприклад, *IBM/360*.

У цей період були реалізовані практично всі основні механізми, властиві сучасним ОС: мультипрограмування, мультипроцесування, підтримка багатотермінального, розрахованого на багато користувачів режиму, віртуальна пам'ять, файлові системи, розмежування доступу і мережна робота. У ці роки починається розквіт системного програмування. З напрямку прикладної математики, що представляє інтерес для вузького круга фахівців, системне програмування перетворюється на галузь індустрії, яка робить безпосередній вплив на практичну діяльність мільйонів людей.

В умовах різких збільшених можливостей комп'ютера, пов'язаних з обробкою і зберіганням даних, виконання тільки однієї програми в кожен момент часу виявилось край неефективним. Почалися розробки в області мультипрограмування. Мультипрограмування – спосіб організації обчислювального процесу, при якому в пам'яті комп'ютера знаходиться одночасно декілька програм, що поперемінно виконуються на одному процесорі. Мультипрограмування було реалізоване в двох варіантах:

- пакетна обробка;
- розділення часу.

Системи пакетної обробки призначалися для вирішення завдань в основному обчислювального характеру, які не вимагають швидкого отримання результатів. Головною метою і критерієм ефективності систем пакетної обробки є максимальна пропускна спроможність, тобто рішення максимального числа завдань в одиницю часу.

Для досягнення цієї мети в системах пакетної обробки використовується наступна схема функціонування (рис. 1.2): на початку роботи формується пакет завдань, кожне завдання містить вимогу до системних

ресурсів; з цього пакету завдань формується мультипрограмний набір, тобто безліч одночасно виконуваних завдань. Для одночасного виконання вибираються завдання, що пред'являють до ресурсів різні вимоги так, щоб забезпечувалося збалансоване завантаження всіх пристроїв обчислювальної машини. Наприклад, в мультипрограмному наборі бажана присутність і обчислювальних завдань, і завдань з інтенсивним введенням-виводом. Таким чином, вибір нового завдання з пакету завдань залежить від внутрішньої ситуації, що складається в системі, тобто вибирається "вигідне" завдання. Отже, в обчислювальних системах, що працюють під управлінням пакетних ОС, неможливо гарантувати виконання того або іншого завдання протягом певного періоду часу.

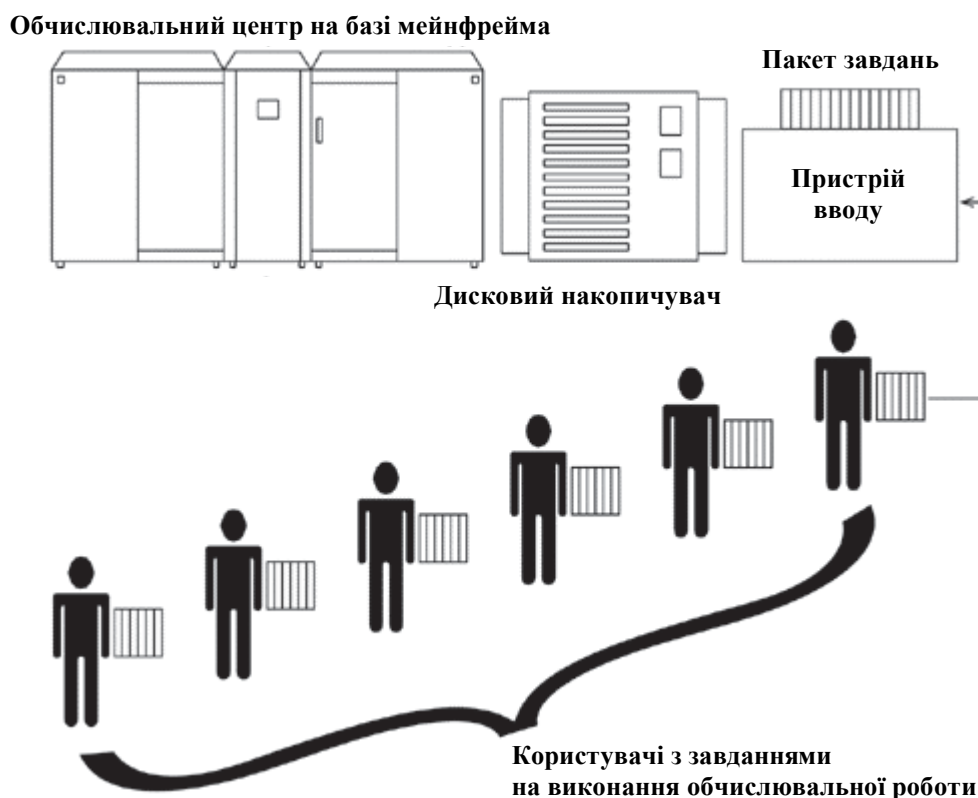


Рис. 1.2. Централізований характер обчислень в системах пакетної обробки.

У системах пакетної обробки перемикання процесора з одного завдання на інше відбувається за ініціативою найактивнішого завдання, наприклад, коли воно "відмовляється" від процесора тому, що необхідно виконати операцію введення-виводу. Тому існує висока вірогідність того, що одне завдання може надовго зайняти процесор, і виконання інтерактивних завдань стане неможливим. Взаємодія користувача з обчислювальною машиною, на якій встановлена система пакетної обробки, зводиться до того, що користувач приносить завдання, віддає його диспетчерові-операторові, а в кінці дня після виконання всього пакету завдань отримує результат. Очевидно, що такий порядок підвищує ефективність функціонування апаратури, але знижує ефективність роботи користувача.

У системах розділення часу користувачам (або одному користувачеві) надається можливість інтерактивної роботи відразу з декількома застосуваннями. Для цього кожне застосування повинне регулярно взаємодіяти з користувачем. Зрозуміло, що в пакетних системах можливості діалогу користувача обмежені.

У системах розділення часу ця проблема вирішується за рахунок того, що ОС примусово періодично припиняє виконання завдань, не чекаючи, коли вони самі звільнять процесор. Всім завданням поперемінно виділяється квант процесорного часу, таким чином, користувачі, що запустили програми на виконання, дістають можливість підтримувати з ними діалог.

Системи розділення часу покликані виправити основний недолік систем пакетної обробки – ізоляцію користувача-програміста від процесу виконання завдань. Кожному користувачеві в цьому випадку надається термінал, з якого він може вести діалог з своєю програмою. Оскільки в системах розділення часу кожному завданню виділяється тільки квант процесорного часу, жодне завдання не займає процесор надовго, і час

відповіді виявляється прийнятним. Якщо квант невеликий, то у всіх користувачів, що одночасно працюють на одній і тій же машині, складається враження, що кожен з них використовує машину одноособово.

Ясно, що системи розділення часу володіють меншою пропускною спроможністю, чим системи пакетної обробки, оскільки на виконання приймається кожне запущене користувачем завдання, а не те, яке "вигідне" системі. Крім того, продуктивність системи знижується через додаткове витрачання обчислювальної потужності на частіше перемикання процесора із завдання на завдання. Це цілком відповідає тому, що критерієм ефективності систем розділення часу є не максимальна пропускна спроможність, а зручність і ефективність роботи користувача. Разом з тим, мультипрограмне виконання інтерактивних завдань підвищує пропускну спроможність комп'ютера (хай і не в такому ступені, як пакетні системи). Апаратура завантажується краще, оскільки поки одне завдання чекає повідомлення користувача, інші завдання можуть оброблятися процесором.

Багатотермінальні системи як прообраз мережі

Багатотермінальний режим використовувався не тільки в системах розділення часу, але і в системах пакетної обробки. При цьому не тільки оператор, але і всі користувачі діставали можливість формувати свої завдання і управляти їх виконанням з свого терміналу. Такі операційні системи отримали назву систем віддаленого введення завдань.

Термінальні комплекси могли розташовуватися на великій відстані від процесорних стійок, з'єднуючись з ними за допомогою різних глобальних зв'язків – модемних з'єднань телефонних мереж або виділених каналів. Для підтримки віддаленої роботи терміналів в операційних системах з'явилися спеціальні програмні модулі, що реалізують різні (в той час, як правило, нестандартні) протоколи зв'язку. Такі обчислювальні системи з віддаленими терміналами зберігаючи централізований характер обробки даних, якоюсь мірою були прообразом сучасних комп'ютерних мереж (рис. 1.3), а відповідне системне програмне забезпечення – прообразом мережних операційних систем.

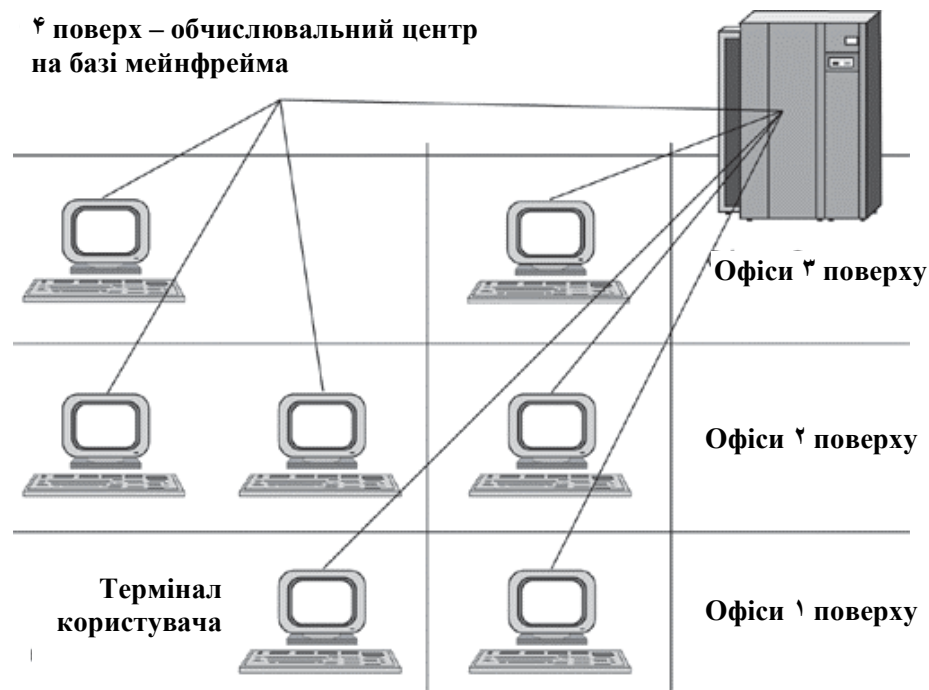


Рис. 1.3. Багатотермінальна система – прообраз обчислювальної мережі.

Багатотермінальні централізовані системи вже мали всі зовнішні ознаки локальних обчислювальних мереж, проте по суті ними не були, оскільки зберігали суть централізованої обробки даних автономно працюючого комп'ютера.

Дійсно, рядовий користувач роботу за терміналом мейнфрейма сприймав приблизно так само, як зараз сприймає роботу за підключеним до мережі персональним комп'ютером. Користувач міг дістати доступ до загальних файлів і периферійних пристроїв, при цьому у нього створювалася повна ілюзія одноосібного володіння комп'ютером, оскільки він міг запустити потрібну йому програму у будь-який момент і майже відразу ж отримати результат.

1.1.3. Глобальні обчислювальні мережі

Хоча теоретичні роботи із створення концепцій мережної взаємодії велися майже з моменту появи обчислювальних машин, значущі практичні результати по об'єднанню комп'ютерів в мережі були отримані лише в кінці 60-х, коли за допомогою глобальних зв'язків і техніки комутації пакетів вдалося реалізувати

взаємодію машин класу мейнфреймів і суперкомп'ютерів (рис. 1.4). Ці дорогі комп'ютери зберігали унікальні дані і програми, обмін якими дозволив підвищити ефективність їх використання.



Рис. 1.4. Об'єднання віддалених СУПЕР-ЕОМ глобальними зв'язками.

Але ще до реалізації зв'язків "комп'ютер-комп'ютер", було вирішено простіше завдання – організація зв'язку "віддалений термінал-комп'ютер". Термінали, що знаходяться від комп'ютера на відстані багатьох сотень, а то і тисяч кілометрів, з'єднувалися з комп'ютерами через телефонні мережі за допомогою модемів. Такі мережі дозволяли численним користувачам діставати віддалений доступ до ресурсів декількох могутніх комп'ютерів класу СУПЕР-ЕОМ, що розділялися. І тільки потім були розроблені засоби обміну даними між комп'ютерами в автоматичному режимі. На основі цього механізму в перших мережах були реалізовані служби обміну файлами, синхронізації баз даних, електронної пошти та інші, які стали тепер традиційними – мережні служби.

У 1969 році міністерство оборони США ініціювало роботи по об'єднанню в загальну мережу суперкомп'ютерів оборонних і науково-дослідних центрів. Ця мережа, що отримала назву *ARPANET*, послужила відправною точкою для створення першої і найвідомішої нині глобальної мережі, – *Internet*. Мережа *ARPANET* об'єднувала комп'ютери різних типів, що працювали під управлінням різних ОС з додатковими модулями, які реалізують комунікаційні протоколи, загальні для всіх комп'ютерів мережі. Такі ОС можна вважати першими мережними операційними системами.

Мережні ОС на відміну від багатотермінальних дозволяли не тільки розосередити користувачів, але і організувати розподілене зберігання і обробку даних між декількома комп'ютерами, зв'язаними електричними зв'язками. Будь-яка мережна операційна система, з одного боку, виконує всі функції локальної операційної системи, а з іншого боку, володіє деякими додатковими засобами, що дозволяють їй взаємодіяти по мережі з операційними системами інших комп'ютерів. Програмні модулі, що реалізують мережні функції, з'являлися в операційних системах поступово, у міру розвитку мережних технологій, апаратної бази комп'ютерів і виникнення нових завдань, що вимагають мережної обробки.

У 1974 році компанія *IBM* оголосила про створення власної мережної архітектури для своїх мейнфреймів, що отримала назву *SNA* (System Network Architecture, системна мережна архітектура). В цей же час в Європі активно велися роботи із створення і стандартизації мереж *X.25*.

Таким чином, хронологічно першими з'явилися глобальні мережі *WAN* (*Wide Area Network*), тобто мережі, які об'єднують територіально розосереджені комп'ютери, можливо, знаходяться в різних містах і країнах. Саме при побудові глобальних мереж було вперше запропоновано і відпрацьовано багато основних ідей і концепцій сучасних обчислювальних мереж, таких, наприклад, як багаторівнева побудова комунікаційних протоколів, технологія комутації пакетів і маршрутизація пакетів в складених мережах.

Спадщина телефонних мереж

Глобальні комп'ютерні мережі дуже багато чого успадкували від інших, набагато старіших і глобальніших мереж – телефонних. Головним результатом створення перших глобальних комп'ютерних мереж була відмова від принципу комутації каналів, який впродовж багатьох десятиріч успішно використовувався в телефонних мережах.

Складений канал з постійною швидкістю, що виділяється на весь час сеансу зв'язку, не може ефективно використовуватися для передачі пульсуючого трафіку комп'ютерних даних, у якого періоди інтенсивного обміну чергуються з тривалими паузами. Експерименти і математичне моделювання показали, що пульсуючий і в значній мірі слабо чутливий до затримок комп'ютерний трафік набагато ефективніше передається по мережах, що використовують принцип комутації пакетів, коли дані розділяються на невеликі порції, які самостійно переміщуються по мережі за рахунок вбудовування адреси кінцевого вузла в заголовок пакету.

Оскільки прокладка високоякісних ліній зв'язку на великі відстані обходиться дуже дорого, спочатку в глобальних мережах часто використовувалися вже існуючі канали зв'язку, принципово призначені зовсім

для інших цілей. Наприклад, протягом багатьох років *глобальні мережі* будувалися на основі телефонних каналів тональної частоти, здатних в кожен момент часу вести передачу тільки однієї розмови в аналоговій формі. Оскільки швидкість передачі дискретних комп'ютерних даних по таких каналах була дуже низькою (десятки кілобіт в секунду), набір послуг, що надаються, в глобальних мережах такого типу зазвичай обмежувався передачею файлів, переважно у фоновому режимі, і електронною поштою.

Крім низької швидкості такі канали мають і інший недолік – вони вносять значні спотворення до переданих сигналів. Тому протоколи глобальних мереж, побудованих з використанням каналів зв'язку низької якості, відрізняються складними процедурами контролю і відновлення даних. Типовим прикладом таких мереж є мережі *X.25*, розроблені ще в початку 70-х, коли низькошвидкісні аналогові канали, що орендуються у телефонних компаній, були переважним типом каналів, що сполучають комп'ютери і комутатори глобальної обчислювальної мережі.

Розвиток технології глобальних комп'ютерних мереж багато в чому визначався прогресом телефонних мереж. З кінця 60-х років в телефонних мережах все частіше стала застосовуватися передача голосу в цифровій формі, що привело до появи високошвидкісних цифрових каналів, що сполучають АТС і що дозволяють одночасно передавати десятки і сотні розмов. Була розроблена спеціальна технологія плезіохронної цифрової ієрархії (*Plesiochronous Digital Hierarchy, PDH*), призначена для створення так званих первинних, або опорних мереж. Такі мережі не надають послуг кінцевим користувачам, вони є фундаментом, на якому будуються швидкісні цифрові канали "точка-точка", які сполучають устаткування іншої мережі, яка вже працює на кінцевого користувача.

Спочатку технологія PDH, що підтримує швидкості до 140 Мбіт/с, була внутрішньою технологією телефонних компаній. Проте з часом ці компанії стали здавати частину своїх каналів PDH в оренду підприємствам, які використовували їх для створення власних телефонних і глобальних комп'ютерних мереж.

Технологія синхронної цифрової ієрархії (*Synchronous Digital Hierarchy, SDH*), яка з'явилася в кінці 80-х років, розширила діапазон швидкостей цифрових каналів до 10 Гбіт/с, а технологія спектрального мультиплексування *DWDM* (*Dense Wave Division Multiplexing*) – до сотень гигабіт і навіть декілька терабіт в секунду. Сьогодні глобальні мережі за різноманітністю і якістю послуг, що надаються, наздогнали локальні мережі, які довгий час лідирували в цьому відношенні, хоч і з'явилися значно пізніше.

Міні-комп'ютери – передвісники локальних мереж

На початку 70-х років відбулася важлива подія, що безпосередньо вплинула на еволюцію комп'ютерних мереж. В результаті технологічного прориву в області виробництва комп'ютерних компонентів з'явилися великі інтегральні схеми (ВІС). Їх порівняно невисока вартість і багаті функціональні можливості привели до створення міні-комп'ютерів, які стали реальними конкурентами мейнфреймів.

З цього часу десяток міні-комп'ютерів, маючи ту ж вартість, що і мейнфрейм, виконував деякі завдання (як правило, що добре розпаралелюються) швидше. Навіть невеликі підрозділи підприємств дістали можливість мати власні комп'ютери. До середини 70-х років стали широко використовуватися міні-комп'ютери *PDP-11, Nova, HP*.

За допомогою міні-комп'ютерів здійснювалося управління технологічним устаткуванням і виконувалися інші завдання рівня відділу підприємства. Таким чином, з'явилася концепція розподілу комп'ютерних ресурсів по всьому підприємству. Проте при цьому всі комп'ютери однієї організації як і раніше продовжували працювати автономно (рис. 1.5).

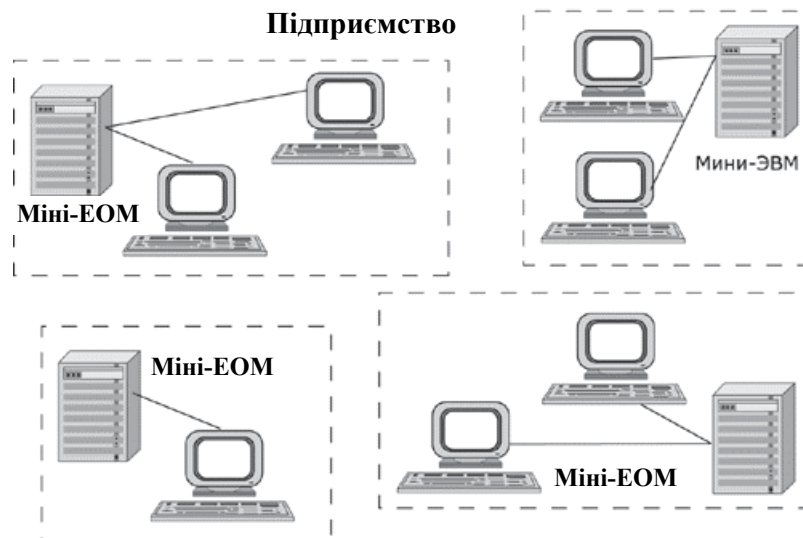


Рис. 1.5. Автономне використання декількох міні-комп'ютерів на одному підприємстві.

Архітектура міні-комп'ютерів була значно спрощена в порівнянні з мейнфреймами, що знайшло віддзеркалення і в їх операційних системах. Багато мультипрограмних функцій, розрахованих на користувачів ОС мейнфреймів, було усічено з урахуванням обмеженості ресурсів міні-комп'ютерів. Операційні системи міні-комп'ютерів часто стали робити спеціалізованими, наприклад, тільки для управління в реальному часі або тільки для підтримки режиму розділення часу (*RSX-11M* для тих же комп'ютерів). Ці операційні системи не завжди були розрахованими на багато користувачів, що у багатьох випадках виправдовувалося невисокою вартістю машин. Важливою віхою в історії міні-комп'ютерів і взагалі в історії операційних систем стало створення ОС *Unix*.

1.1.4. Роль персональних комп'ютерів в еволюції комп'ютерних мереж

Початок 80-х років пов'язаний з ще однією знаменною для історії мереж подією – появою персональних комп'ютерів. Ці пристрої стали ідеальними елементами для побудови мереж: з одного боку, вони були достатньо могутніми для роботи мережного програмного забезпечення, а з іншого – явно потребували об'єднання обчислювальної потужності для вирішення складних завдань, а також розділення дорогих периферійних пристроїв і дискових масивів. Тому персональні комп'ютери стали активно використовуватися в локальних мережах, причому не тільки як клієнтські комп'ютери, але і як центри зберігання і обробки даних, тобто як мережні сервери, потіснивши з цих ролей міні-комп'ютери і мейнфрейми.

З погляду архітектури персональні комп'ютери нічим не відрізнялися від міні-комп'ютерів типу *PDP-11*, але їх вартість була істотно нижча. Якщо з появою міні-комп'ютера можливість мати власну обчислювальну машину отримали відділи підприємств або університети, то створення персонального комп'ютера дало таку можливість окремішній людині.

Створення персональних комп'ютерів послужило могутнім каталізатором для бурхливого зростання локальних мереж, оскільки з'явилася відмінна матеріальна основа у вигляді десятків і сотень машин, які належать одному підприємству і розташовані в межах однієї будівлі. Комп'ютери стали використовуватися не тільки фахівцями. В результаті підтримка мережних функцій стала для ОС персональних комп'ютерів необхідною умовою.

Нові можливості користувачів локальних мереж

Розробники локальних мереж привнесли в організацію роботи користувачів багато нового. Так, стало набагато простіше, ніж в глобальних мережах, діставати доступ до мережних ресурсів – в локальній мережі користувачеві не доводиться запам'ятовувати складні ідентифікатори ресурсів, що розділяються. Для цих цілей система надає список ресурсів в зручній для сприйняття формі, наприклад, у вигляді деревовидної графічної структури ("дерева" ресурсів). Ще один прийом, який дозволяє оптимізувати роботу в локальній мережі, полягає в тому, що після з'єднання з віддаленим ресурсом, користувач дістає можливість звертатися до нього за допомогою тих же команд, які він застосовував при роботі з локальними ресурсами. Наслідком (і в той же час рушійною силою) такого прогресу стала поява величезної кількості непрофесійних користувачів, звільнених від необхідності вивчати спеціальні (і достатньо складні) команди для мережної роботи.

Може виникнути питання – чому всі ці переваги користувачі отримали тільки з появою локальних мереж? Головним чином, це пов'язано з використанням в локальних мережах якісних кабельних ліній зв'язку, на яких навіть мережні адаптери першого покоління забезпечували швидкість передачі даних до 10 Мбіт/с. При невеликій протяжності, властивій локальним мережам, вартість таких ліній зв'язку була цілком прийнятною. Тому економне витрачання пропускної спроможності каналів, одне з основних завдань, покладених на технології перших глобальних мереж, ніколи не виходило на перший план при розробці протоколів локальних мереж. У таких умовах основним механізмом прозорого доступу до ресурсів локальних мереж стали періодичні широкомовні оголошення серверів про свої ресурси і послуги. На підставі таких оголошень клієнтські комп'ютери складали списки наявних в мережі ресурсів і надавали їх користувачеві.

1.1.5. Еволюція мережних операційних систем

Проте і "дружній" інтерфейс, і мережні функції з'явилися у операційних системах персональних комп'ютерів не відразу. Перша версія найбільш популярної операційної системи раннього етапу розвитку персональних комп'ютерів – *MSDOS* компанії *Microsoft* – не надавала таких можливостей. Брак функцій для *MSDOS* і подібних нею ОС компенсувався зовнішніми програмами, що надавали користувачеві зручний графічний інтерфейс (наприклад, *Norton Commander*) або засоби тонкого управління дисками (наприклад, *PC Tools*). Найбільший вплив на розвиток програмного забезпечення для персональних комп'ютерів надало операційне середовище *Windows* компанії *Microsoft*, яке була надбудовою над *MSDOS*.

Разом з версією *MSDOS 3.1* в 1984 році компанія *Microsoft* випустила продукт *Microsoft Networks*, який зазвичай називають *MSNET*. Деякі концепції, закладені в *MSNET*, такі як введення в структуру базових мережних компонентів – редиректора і мережного сервера, успішно перейшли в пізніші мережні продукти *Microsoft*: *LAN Manager*, *Windows for Workgroups*, а потім і в *Windows NT*.

Інший шлях вибрали розробники *Novell*. Вони спочатку зробили ставку на створення операційної системи з вбудованими мережними функціями і добилися на цьому шляху великих успіхів. Мережні

операційні системи *NetWare* виробництва *Novell* на довгий час стали еталоном продуктивності, надійності і захищеності для локальних мереж.

У 1987 р. в результаті спільних зусиль *Microsoft* і *IBM* з'явилася перша багатозадачна операційна система для персональних комп'ютерів з процесором *Intel 80286*, що повною мірою використовує можливості захищеного режиму – *OS/2*.

Мережні розробки компаній *Microsoft* і *IBM* привели до появи *NETBios* – дуже популярного транспортного протоколу і одночасно інтерфейсу прикладного програмування для локальних мереж, що знайшло застосування практично у всіх мережних операційних системах для персональних комп'ютерів. Цей протокол і сьогодні застосовується для створення невеликих локальних мереж.

Не дуже вдала доля *OS/2* не дозволила системам *LAN Manager* і *LAN Server* захопити помітну частку ринку, але принципи роботи цих мережних систем багато в чому знайшли віддзеркалення в успішнішою *Microsoft Windows NT*, що містить вбудовані мережні компоненти (деякі з них мають приставку LM – від *LAN Manager*).

На персональні комп'ютери встановлювалися спеціально для них розроблені операційні системи, подібні *MSDOS*, *NetWare* і *OS/2*, а також адаптувалися ті, що існують ОС. Поява процесорів *Intel 80286* і особливо *80386* з підтримкою мультипрограмування дозволило перенести на платформу персональних комп'ютерів ОС *Unix*. Найбільш відомою системою цього типу була версія *Unix* компанії *Santa Cruz Operation (SCO Unix)*.

У 90-і роки практично всі операційні системи, що займають помітне місце на ринку, стали мережними. Мережні функції сьогодні вбудовуються в ядро ОС і є його невід'ємною частиною. Операційні системи отримали засоби для роботи зі всіма основними технологіями локальних (*Ethernet*, *Fast Ethernet*, *Gigabit Ethernet*, *Token Ring*, *FDDI*, *ATM*) і глобальних (*X.25*, *Frame relay*, *ISDN*, *ATM*) мереж, а також засоби для створення складених мереж (*IP*, *IPX*, *AppleTalk*, *RIP*, *OSPF*, *NLSP*). У операційних системах використовуються засоби мультиплексування декількох стеків протоколів, що дозволяє комп'ютерам підтримувати мережну роботу з різнорідними клієнтами і серверами. З'явилися спеціалізовані ОС, призначені виключно для виконання комунікаційних завдань. Наприклад, мережна операційна система *IOS*, що працює в маршрутизаторах, організовує в мультипрограму режимі виконання набору програм, кожна з яких реалізує один з комунікаційних протоколів.

У другій половині 90-х років всі виробники операційних систем різко підсилили підтримку засобів роботи з *Internet* (окрім виробників *Unix*-систем, в яких ця підтримка завжди була істотною). Окрім самого стека *TCP/IP* в комплект постачання почали включати утиліти, що реалізують такі популярні сервіси *Internet* як *telnet*, *ftp*, *DNS* і *Web*. Вплив *Internet* виявився і в тому, що комп'ютер перетворився з обчислювального пристрою в засіб комунікацій з розвиненими обчислювальними можливостями.

На сучасному етапі розвитку операційних систем на передній план вийшли засоби забезпечення безпеки. Це обумовлено збільшеною цінністю інформації, що обробляється комп'ютерами, а також підвищеним рівнем ризику, пов'язаного з передачею даних по мережах, особливо по загальнодоступних, таким як *Internet*. Багато операційних систем володіють сьогодні розвиненими засобами захисту інформації, заснованими на шифруванні даних, аутентифікації і авторизації.

Сучасним операційним системам властива багатоплатформенність, тобто здатність працювати на комп'ютерах різного типу. Багато операційних систем мають спеціальні версії для підтримки кластерної архітектури, що забезпечує високу продуктивність і відмовостійкість. Виключення поки складає ОС *NetWare*, всі версії якої розроблені для платформи *Intel*, а реалізація функцій *NetWare* у вигляді оболонки для інших ОС, наприклад, *NetWare for AIX*, успіху не мала.

Останніми роками отримала подальший розвиток тенденція підвищення зручності роботи з комп'ютером. Ефективність роботи користувача стає основним чинником, що визначає ефективність обчислювальної системи в цілому. Зусилля людини не повинні витрачатися на настройку параметрів обчислювального процесу, як це відбувалося в ОС попередніх поколінь. Наприклад, в системах пакетної обробки для мейнфреймів кожен користувач повинен був за допомогою мови управління завданнями визначити велику кількість параметрів, що відносяться до організації обчислювальних процесів в комп'ютері. Так, для системи *OS/360* мова управління завданнями *JCL* передбачала можливість визначення користувачем більше 40 параметрів, серед яких були пріоритет завдання, вимоги до основної пам'яті, граничний час виконання завдання, перелік використовуваних пристроїв введення-виводу і режими їх роботи.

Сучасна операційна система бере на себе вибір параметрів операційного середовища, за допомогою різних адаптивних алгоритмів. Наприклад, тайм-аути в комунікаційних протоколах часто визначаються залежно від умов роботи мережі. Розподіл оперативної пам'яті між процесами здійснюється автоматично за допомогою механізмів віртуальної пам'яті залежно від активності цих процесів і інформації про частоту використання ними тієї або іншої сторінки. Миттєві пріоритети процесів визначаються динамічно залежно від передісторії, що включає, наприклад, час знаходження процесу в черзі, відсоток використання виділеного кванта часу, інтенсивність введення-виводу і т.п. Навіть в процесі установки більшість ОС пропонують режим вибору параметрів за умовчанням, який гарантує хай не оптимальну, але завжди прийнятну якість роботи систем.

Постійно підвищується зручність інтерактивної роботи з комп'ютером шляхом включення в операційну систему розвинених графічних інтерфейсів, що використовують разом з графікою звук і відео. Це особливо

важливо для перетворення комп'ютера на термінал нової загальнодоступної мережі, якою поступово стає *Internet*, оскільки для масового користувача термінал повинен бути по простоті використання подібний до телефонного апарату. Призначений для користувача інтерфейс операційної системи стає все більш інтелектуальним, він направляє дії людини в типових ситуаціях і виконує багато завдань автоматично.

Рівень зручності в роботі з ресурсами, які сьогодні операційні системи ізольованих комп'ютерів надають користувачам, адміністраторам і розробникам програм, для *мережних операційних систем* є тільки принадою перспективою. Поки ж користувачі і адміністратори мережі витрачають значний час на спроби з'ясувати, де знаходиться той або інший ресурс, розробники мережних програм докладають багато зусиль для визначення місцеположення даних і програмних модулів в мережі. Операційні системи майбутнього повинні забезпечити високий рівень прозорості мережних ресурсів, узявши на себе завдання організації розподілених обчислень, перетворивши мережу на віртуальний комп'ютер. Саме такий сенс вкладають в лаконічне гасло "Мережа – це комп'ютер" фахівці компанії Sun, але, щоб запровадити гасло в життя, розробникам операційних систем належить пройти ще довгий шлях.

У табл. 1.1 наведено приблизну хронологію розвитку комп'ютерних мереж.

Таблиця 1.1

Хронологічна послідовність найважливіших подій в історії розвитку комп'ютерних мереж	
Перші лампові комп'ютери	Початок 40-х
Перші комп'ютери на напівпровідникових схемах (транзисторах)	Середина 50-х
Перші комп'ютери на інтегральних схемах. Перші мультипрограмні ОС	Середина 60-х
Перші глобальні зв'язки комп'ютерів	Кінець 60-х
Початок передач по телефонних мережах голосу в цифровій формі	Кінець 60-х
Поява великих інтегральних схем. Перші міні-комп'ютери	Почало 70-х
Перші нестандартні локальні мережі	Почало 70-х
Створення мережної архітектури <i>IBM SNA</i>	1974
Створення технології X.25	1974
Поява персональних комп'ютерів	Почало 80-х
Створення <i>Internet</i> в сучасному вигляді. Установка на всіх вузлах стеку <i>TCP/IP</i>	Почало 80-х
Поява стандартних технологій локальних мереж	<i>Ethernet</i> - 1980 <i>Token Ring</i> - 1985 <i>FDDI</i> - 1989
Початок комерційного використання <i>Internet</i>	Кінець 80-х
Розробка <i>Web</i>	1991

1.2. Характеристики і стандарти мереж передачі та розподілу даних

1.2.1. Вимоги до мереж

Відповідність стандартам – це тільки одна з багатьох вимог, що пред'являються до сучасних мереж. У цьому розділі розглядаються деякі інші, не менш важливі. Найзагальніше побажання, яке можна висловити відносно роботи мережі, – це виконання мережею того набору послуг, для надання яких вона призначена: наприклад, надання доступу до файлових архівів або сторінок публічних Web-сайтів Internet, обмін електронною поштою в межах підприємства або в глобальних масштабах, інтерактивний обмін голосовими повідомленнями IP-телефонії і т.п.

Решта всіх вимог – продуктивність, надійність, сумісність, керованість, захищеність, розширюваність і масштабованість пов'язані з якістю виконання цього основного завдання. І хоча всі перераховані вище вимоги вельми важливі, часто поняття "якість обслуговування" *QoS* комп'ютерної мережі трактується вужче: у нього включаються тільки дві найважливіші характеристики мережі – продуктивність і надійність.

Продуктивність

Потенційно висока продуктивність – це одна з основних переваг розподілених систем, до яких відносяться комп'ютерні мережі. Ця властивість забезпечується принциповим, але, на жаль, не завжди можливим розподілом робіт, що практично реалізовується, між декількома комп'ютерами мережі.

Основні характеристики продуктивності мережі:

- час реакції;

швидкість передачі трафіку;

- пропускна спроможність;
- затримка передачі і варіація затримки передачі.

Час реакції мережі є інтегральною характеристикою продуктивності мережі з погляду користувача. Саме цю характеристику має на увазі користувач, коли говорить: "Сьогодні мережа працює поволі". У загальному випадку час реакції визначається як інтервал між виникненням запиту користувача до якої-небудь мережної служби і отриманням відповіді на нього. Очевидно, що значення цього показника залежить від типу служби, до якої звертається користувач, від того, який користувач і до якого сервера звертається, а також від поточного стану елементів мережі – завантаженості сегментів, комутаторів і маршрутизаторів, через які проходить запит, завантаженості сервера і т.п. Тому має сенс використовувати також і середньозважену оцінку *часу реакції* мережі, усереднюючи цей показник по користувачах, серверах і часі дня (від якого в значній мірі залежить завантаження мережі).

Час реакції мережі зазвичай складається з декількох складових. У загальному випадку в нього входить:

- час підготовки запитів на клієнтському комп'ютері;
- час передачі запитів між клієнтом і сервером через сегменти мережі і проміжне комунікаційне устаткування;
- час обробки запитів на сервері;
- час передачі відповідей від сервера клієнтові і час обробки отримуваних від сервера відповідей на клієнтському комп'ютері.

Очевидно, що розкладання часу реакції на складові користувача не цікавить – йому важливий кінцевий результат. Проте для мережного фахівця дуже важливо виділити із загального часу реакції складові, відповідні етапам власне мережної обробки даних, – передачу даних від клієнта до сервера через сегменти мережі і комунікаційне устаткування. Знання мережних складових часу реакції дозволяє оцінити продуктивність окремих елементів мережі, виявити вузькі місця і при необхідності виконати модернізацію мережі для підвищення її загальної продуктивності.

Продуктивність мережі може характеризуватися також швидкістю передачі трафіку. Швидкість передачі трафіку може бути миттєвою, максимальною і середньою:

- середня швидкість обчислюється шляхом ділення загального об'єму переданих даних на час їх передачі, причому вибирається достатньо тривалий проміжок часу – година, день або тиждень;
- миттєва швидкість відрізняється від середньої тим, що для усереднювання вибирається дуже маленький проміжок часу – наприклад, 10 мс або 1 з;
- максимальна швидкість – це найбільша швидкість, зафіксована протягом періоду спостереження.

Найчастіше при проектуванні, настройці і оптимізації мережі використовуються такі показники, як середня і максимальна швидкість. Середня швидкість, з якою окремих елементів або мережа в цілому обробляє трафік, дозволяє оцінити роботу мережі впродовж тривалого часу, протягом якого через закон великих чисел піки і спади інтенсивності трафіку компенсують один одного. Максимальна швидкість дозволяє оцінити, як мережа справлятиметься з піковими навантаженнями, характерними для особливих періодів роботи, наприклад, в уранішній час, коли співробітники підприємства майже одночасно реєструються в мережі і звертаються до файлів, що розділяються, і баз даних. Зазвичай при визначенні швидкісних характеристик деякого сегменту або пристрою в переданих даних не виділяється трафік якогось певного користувача, програми або комп'ютера – підраховується загальний об'єм переданої інформації. Проте, для точнішої оцінки якості обслуговування така деталізація бажана, і останнім часом системи управління мережами все частіше дозволяють її виконувати.

Пропускна спроможність – це максимально можлива швидкість обробки трафіку, визначена стандартом технології, на якій побудована мережа. Пропускна спроможність відображає максимально можливий об'єм даних, що передається мережею або її частиною в одиницю часу. Пропускна спроможність вже не є, подібно до часу реакції або швидкості проходження даних по мережі, призначеною для користувача характеристикою, оскільки вона говорить про швидкість виконання внутрішніх операцій мережі – передачі пакетів даних між вузлами мережі через різні комунікаційні пристрої. Зате вона безпосередньо характеризує якість виконання основної функції мережі – транспортування повідомлень – і тому частіше використовується при аналізі продуктивності мережі, чим час реакції або швидкість.

Пропускна спроможність вимірюється або в бітах в секунду, або в пакетах в секунду. В останньому випадку треба враховувати, чи є довжина пакетів постійною. При оцінці пропускної спроможності мереж, де циркулюють пакети (кадри, повідомлення) зі змінною довжиною, беруть деякий середньостатистичний параметр довжини пакетів. Це, як правило, математичне очікування довжини пакетів як випадкової величини.

Пропускна спроможність мережі залежить як від характеристик фізичного середовища передачі (мідний кабель, оптичне волокно, вита пара), так і від прийнятого способу передачі даних (технологія Ethernet, Fast Ethernet, ATM). Пропускна спроможність часто використовується як характеристика не стільки мережі, скільки власне технології, на якій побудована мережа. Важливість цієї характеристики для мережної технології показує, зокрема, і те, що її значення іноді стає частиною назви, наприклад, 10 Мбіт/с Ethernet, 100 Мбіт/с Ethernet.

На відміну від часу реакції або швидкості передачі трафіку пропускна спроможність не залежить від завантаженості мережі і має постійне значення, яке визначається використовуваними в мережі технологіями. На різних ділянках гетерогенної мережі, де використовується декілька різних технологій, пропускна спроможність може бути різною. Для аналізу і настройки мережі дуже корисно знати дані про пропускну спроможність окремих її елементів. Важливо відзначити, що через послідовний характер передачі даних різними елементами мережі загальна пропускна спроможність будь-якого складеного шляху в мережі буде рівна мінімальній з пропускних спроможностей елементів, що становлять маршрут. Для підвищення пропускної спроможності складеного шляху необхідно в першу чергу звернути увагу на найповільніші елементи. Іноді корисно оперувати загальною пропускну спроможністю мережі, яка визначається як середня кількість інформації, переданої між всіма вузлами мережі за одиницю часу. Цей показник характеризує якість мережі в цілому, не диференціюючи його по окремих сегментах або пристроях.

Затримка передачі визначається як затримка між моментом надходження даних на вхід якого-небудь мережного пристрою або частини мережі і моментом появи їх на виході цього пристрою. Цей параметр продуктивності по сенсу близький до часу реакції мережі, але відрізняється тим, що завжди характеризує тільки мережні етапи обробки даних, без затримок обробки кінцевими вузлами мережі.

Звичайно якість мережі характеризують величинами максимальної затримки передачі і варіацією затримки. Не всі типи трафіку чутливі до затримок передачі, в усякому разі, до тих величин затримок, які характерні для комп'ютерних мереж, – зазвичай затримки не перевищують сотень мілісекунд, рідше – декількох секунд. Такого порядку затримки пакетів, що породжуються файловою службою, службою електронної пошти або службою друку, мало впливають на якість цих служб з погляду користувача мережі. З іншого боку, такі ж затримки пакетів, що переносять голосові або відеодані, можуть призводити до значного зниження якості що надається користувачеві інформації – виникненню ефекту "луни", неможливості розібрати деякі слова, вібрації зображення і т.п.

Всі вказані характеристики продуктивності мережі достатньо незалежні. Тоді як пропускна спроможність мережі є постійною величиною, швидкість передачі трафіку може варіюватися залежно від завантаження мережі, не перевищуючи, звичайно, межі, що встановлюється пропускну спроможністю. Так в односегментній мережі 10 Мбіт/с *Ethernet* комп'ютери можуть обмінюватися даними з швидкостями 2 Мбіт/с і 4 Мбіт/с, але ніколи – 12 Мбіт/с.

Пропускна спроможність і затримки передачі також є незалежними параметрами, так що мережа може володіти, наприклад, високою пропускну спроможністю, але вносити значні затримки при передачі кожного пакету. Приклад такої ситуації дає канал зв'язку, утворений геостационарним супутником. Пропускна спроможність цього каналу може бути вельми високою, наприклад, 2 Мбіт/с, тоді як затримка передачі завжди складає не меншого 0,24 з, що визначається швидкістю розповсюдження електричного сигналу (близько 300000 км/с) і довжиною каналу (72000 км.).

Надійність і безпека

Одна з первинних цілей створення розподілених систем, до яких відносяться і обчислювальні мережі, полягала в досягненні більшої надійності в порівнянні з окремими обчислювальними машинами. Важливо розрізняти декілька аспектів надійності. Для порівняно простих технічних пристроїв використовуються такі показники надійності, як:

- середній час напрацювання на відмову;
- вірогідність відмови;
- інтенсивність відмов.

Проте ці *показники* придатні для оцінки надійності простих елементів і пристроїв, які можуть знаходитися тільки в двох станах, – працездатному або непрацездатному. Складні системи, що складаються з багатьох елементів, окрім станів працездатності і непрацездатності, можуть мати і інші проміжні стани, які ці характеристики не враховують. Для оцінки надійності складних систем застосовується інший набір характеристик:

- готовність або коефіцієнт готовності;
- збереження даних;
- узгодженість (несуперечність) даних;
- вірогідність доставки даних;
- безпека;
- відмовостійкість.

Готовність або коефіцієнт готовності (availability) означає період часу, протягом якого система може використовуватися. Готовність може бути підвищена шляхом введення надмірності в структуру системи: ключові елементи системи повинні існувати в декількох екземплярах, щоб при відмові одного з них функціонування системи забезпечували інші. Щоб комп'ютерну систему можна було вважати високонадійною, вона повинна як мінімум володіти високою готовністю, але цього недостатньо. Необхідно забезпечити збереження даних і захист їх від спотворень. Крім того, повинна підтримуватися узгодженість (несуперечність) даних, наприклад, якщо для підвищення надійності на декількох файлових серверах зберігається декілька копій даних, то потрібно постійно забезпечувати їх ідентичність.

Оскільки мережа працює на основі механізму передачі пакетів між кінцевими вузлами, однією з характеристик надійності є вірогідність доставки пакету вузлу призначення без спотворень. Разом з цією характеристикою можуть використовуватися і інші показники: вірогідність втрати пакету (по будь-якій з причин – через переповнювання буфера маршрутизатора, неспівпадання контрольної суми, відсутності працездатного шляху до вузла призначення і т. д.), вірогідність спотворення окремого біта переданих даних, співвідношення кількості втрачених і доставлених пакетів.

Іншим аспектом загальної надійності є *безпека* (security), тобто здатність системи захистити дані від несанкціонованого доступу. У розподіленій системі це зробити набагато складніше, ніж в централізованій. У мережах повідомлення передаються по лініях зв'язку, що часто проходять через загальнодоступні приміщення, в яких можуть бути встановлені засоби прослуховування ліній. Іншим вразливим місцем можуть стати залишені без нагляду персональні комп'ютери. Крім того, завжди є потенційна загроза злову захисту мережі від неавторизованих користувачів, якщо мережа має виходи в глобальні загальнодоступні мережі.

Ще однією характеристикою надійності є *відмовостійкість* (fault tolerance). У мережах під відмовостійкістю розуміється здатність системи приховати від користувача відмову окремих її елементів. Наприклад, якщо копії таблиці бази даних зберігаються одночасно на декількох файлових серверах, користувачі можуть просто не відмітити відмови однієї з них. У відмовостійкій системі вихід з ладу одного з її елементів призводить до деякого зниження якості її роботи (деградації), а не до повного останову. Так, при відмові одного з файлових серверів в попередньому прикладі збільшується тільки час доступу до бази даних через зменшення ступеня розпаралелювання запитів, але в цілому система продовжуватиме виконувати свої функції.

Розширюваність і масштабованість

У табл. 1.2 наведені основні відмінності розширюваності та масштабованості.

Таблиця 1.2.

Розширюваність (extensibility)	Масштабованість (scalability)
Можливість порівняно легкого додавання окремих елементів мережі	Легкість розширення системи може забезпечуватися в деяких вельми обмежених межах
Можливість додавання (необов'язково легкого) елементів мережі	Масштабованість означає, що нарощувати мережу можна в дуже широких межах, при збереженні споживчих властивостей мережі

Розширюваність (extensibility) означає можливість порівняно легкого додавання окремих елементів мережі (користувачів, комп'ютерів, застосувань, служб), нарощування довжини сегментів мережі і заміни існуючої апаратури могутнішою. При цьому принципово важливо, що легкість розширення системи іноді може забезпечуватися у вельми обмежених межах. Наприклад, локальна мережа *Ethernet*, побудована на основі одного сегменту товстого коаксіального кабелю, володіє хорошою розширюваністю в тому сенсі, що дозволяє без зусиль підключати нові станції. Проте, така мережа має обмеження на число станцій – воно не повинне перевищувати 30-40. Хоча мережа допускає фізичне підключення до сегменту і більшого числа станцій (до 100), але при цьому найчастіше різко знижується продуктивність мережі. Наявність такого обмеження і є ознакою поганої масштабованості системи при хорошій розширюваності.

Масштабованість (scalability) означає, що мережа дозволяє нарощувати кількість вузлів і протяжність зв'язків в дуже широких межах, при цьому продуктивність мережі не погіршується. Для забезпечення масштабованості мережі доводиться застосовувати додаткове комунікаційне устаткування і спеціальним чином структурувати мережу. Наприклад, хорошою масштабованістю володіє багатосегментна мережа, що

побудована з використанням комутаторів і маршрутизаторів і має ієрархічну структуру зв'язків. Така мережа може включати декілька тисяч комп'ютерів і при цьому забезпечувати кожному користувачеві мережі потрібна якість обслуговування.

Прозорість

Прозорість (transparency) мережі досягається у тому випадку, коли мережа представляється користувачам не як безліч окремих комп'ютерів, зв'язаних між собою складною системою кабелів, а як єдина традиційна обчислювальна машина з системою розділення часу. Відоме гасло компанії Sun Microsystems "Мережа – це комп'ютер" говорить саме про таку прозору мережу.

Прозорість може бути досягнута на двох різних рівнях – на рівні користувача і на рівні програміста. На рівні користувача прозорість означає, що для роботи з віддаленими ресурсами він використовує ті ж команди і звичні процедури, що і для роботи з локальними ресурсами. На програмному рівні прозорість полягає в тому, що застосуванню для доступу до віддалених ресурсів потрібні ті ж виклики, що і для доступу до локальних ресурсів. Прозорості на рівні користувача досягти простіше, оскільки всі особливості процедур, пов'язані з розподілим характером системи, ховаються від користувача програмістом, який створює прикладну програму. Прозорість на рівні прикладної програми вимагає заховання всіх деталей розподілу ресурсів засобами мережної операційної системи. Прозорість – властивість мережі приховувати від користувача деталі свого внутрішнього устрою, що спрощує роботу в мережі.

Мережа повинна приховувати всі особливості операційних систем і відмінності в типах комп'ютерів. Користувач комп'ютера *Macintosh* повинен мати можливість звертатися до ресурсів, підтримуваних *UNIX*-системою, а користувач *UNIX* розділяти інформацію з користувачами *Windows 95*. Переважна більшість користувачів нічого не хоче знати про внутрішні формати файлів або про синтаксис команд *UNIX*. Користувач терміналу *IBM 3270* повинен мати можливість обмінюватися повідомленнями з користувачами мережі персональних комп'ютерів без необхідності вникати в секрети адрес, які важко запам'ятовуються.

Концепція прозорості може бути застосована до різних аспектів мережі. Наприклад, прозорість розташування означає, що від користувача не потрібно знати місцезнаходження програмних і апаратних ресурсів, таких як процесори, принтери, файли і бази даних. Ім'я ресурсу не повинне включати інформацію про місце його розташування, тому імена типу `machine1:prog.c` або `\ftp_serv\pub` прозорими не є. Аналогічно, прозорість переміщення означає, що ресурси можуть вільно переміщатися з одного комп'ютера в інший без зміни імен. Ще одним з можливих аспектів прозорості є прозорість паралелізму, яка полягає в тому, що процес розпаралелювання обчислень відбувається автоматично, без участі програміста, при цьому система сама розподіляє паралельні гілки програми по процесорах і комп'ютерах мережі. В даний час не можна сказати, що прозорість повною мірою властива багатьом обчислювальним мережам. Це скоріше мета, до якої прагнуть розробники сучасних мереж.

Підтримка різних видів трафіку

Комп'ютерні мережі спочатку призначалися для сумісного доступу до ресурсів комп'ютерів: файлів, принтерів і т.п. Трафік, який створюється цими традиційними службами комп'ютерних мереж, має свої особливості і істотно відрізняється від трафіку повідомлень в телефонних мережах або, наприклад, в мережах кабельного телебачення. Проте в 90-і роки в комп'ютерні мережі проник трафік мультимедійних даних, що представляють в цифровій формі мову і відеозображення. Комп'ютерні мережі стали використовуватися для організації відеоконференцій, навчання на основі відеофільмів і т.п. Природно, що для динамічної передачі мультимедійного трафіку потрібні інші алгоритми і протоколи, і, відповідно, інше устаткування. Хоча частка мультимедійного трафіку поки невелика, він вже почав проникати як в глобальні, так і в локальні мережі, і цей процес, очевидно, активно продовжуватиметься.

Головною особливістю трафіку, що утворюється при динамічній передачі голосу або зображення, є наявність жорстких вимог до синхронності переданих повідомлень. Для якісного відтворення безперервних процесів, якими є звукові коливання або зміни інтенсивності світла у відеозображенні, необхідне отримання зміряних і закодованих амплітуд сигналів з тією ж частотою, з якою вони були зміряні на стороні, що передавала. При запізнюванні повідомлень спостерігатимуться спотворення.

В той же час трафік комп'ютерних даних характеризується край нерівномірною інтенсивністю надходження повідомлень в мережу за відсутності жорстких вимог до синхронності доставки цих повідомлень. Наприклад, доступ користувача, що працює з текстом на віддаленому диску, породжує випадковий потік повідомлень між віддаленим і локальним комп'ютерами, залежний від дій користувача, причому затримки при доставці в деяких (достатньо широких з комп'ютерної точки зору) межах мало впливають на якість обслуговування користувача мережі. Всі алгоритми комп'ютерного зв'язку, відповідні протоколи і комунікаційне устаткування були розраховані саме на такий "пульсуючий" характер трафіку, тому необхідність передавати мультимедійний трафік вимагає внесення принципових змін, як в протоколи, так і в устаткування. Сьогодні практично всі нові протоколи в тому або іншому ступені надають підтримку мультимедійного трафіку.

Поєднання в одній мережі традиційного комп'ютерного і мультимедійного трафіку представляє особливу складність. Передача виключно мультимедійного трафіку комп'ютерною мережею хоч і пов'язана з певними складнощами, але доставляє менше клопоту. А ось співіснування двох типів трафіку з протилежними вимогами до якості обслуговування є набагато складнішим завданням. Зазвичай протоколи і устаткування комп'ютерних мереж відносять мультимедійний трафік до факультативного, тому якість його

обслуговування залишає бажати кращого. Сьогодні витрачаються великі зусилля по створенню мереж, які не ущемляють інтереси одного з типів трафіку. Найбільш близькі до цієї мети мережі на основі технології *ATM*, розробники якої спочатку враховували випадок співіснування різних типів трафіку в одній мережі.

Керованість

В ідеалі засоби управління мережами є системою, що здійснює спостереження, контроль і управління кожним елементом мережі, від простих до найскладніших пристроїв, при цьому така система розглядає мережу як єдине ціле, а не як розрізнений набір окремих пристроїв. Під керованістю мережі розуміють можливість централізовано контролювати стан основних елементів мережі, виявляти і вирішувати проблеми, що виникають при роботі мережі, виконувати аналіз продуктивності і планувати розвиток мережі.

Хороша система управління спостерігає за мережею і, виявивши проблему, активізує певну дію, виправляє ситуацію і повідомляє адміністратора про те, що відбулося і які кроки зроблені. Одночасно з цим система управління повинна накопичувати дані, на підставі яких можна планувати розвиток мережі. Нарешті, система управління повинна бути незалежною від виробника і володіти зручним інтерфейсом, що дозволяє виконувати всі дії з однієї консолі.

Вирішуючи тактичні задачі, адміністратори і технічний персонал стикаються з щоденними проблемами забезпечення працездатності мережі. Ці завдання вимагають швидкого рішення, обслуговуючий мережу персонал повинен оперативно реагувати на повідомлення про несправності, що поступають від користувачів або автоматичних засобів управління мережею. Поступово стають помітні загальні проблеми *продуктивності*, конфігурації мережі, обробки збоїв і безпеки даних, що вимагають стратегічного підходу, тобто планування мережі. Планування, окрім цього, включає прогноз змін вимог користувачів до мережі, питання застосування нових програм, нових мережних технологій і т.п.

Необхідність в системі управління особливо яскраво виявляється у великих мережах: корпоративних або глобальних. Без системи управління в таких мережах потрібна присутність кваліфікованих фахівців з експлуатації в кожній будівлі кожного міста, де встановлено устаткування мережі, що у результаті приводить до необхідності утримання величезного штату обслуговуючого персоналу.

В даний час в області систем управління мережами багато невирішених проблем. Явно недостатньо дійсно зручних, компактних і багатопрокольних засобів управління мережею. Більшість існуючих засобів зовсім не управляють мережею, а всього лише здійснюють спостереження за її роботою. Вони стежать за мережею, але не виконують активних дій, якщо з мережею щось відбулося або може відбутися. Мало систем, що масштабуються, здатних обслуговувати як мережі масштабу відділу, так і мережі масштабу підприємства, дуже багато систем управляють тільки окремими елементами мережі і не аналізують здатність мережі виконувати якісну передачу даних між кінцевими користувачами.

Сумісність

Сумісність або інтегрованість означає, що мережа може включати різноманітне програмне і апаратне забезпечення, тобто в ній можуть співіснувати різні операційні системи, що підтримують різні стеки комунікаційних протоколів, і працювати апаратні засоби і програми від різних виробників. Мережа, що складається з різнотипних елементів, називається неоднорідною або гетерогенною, а якщо гетерогенна мережа працює без проблем, то вона є інтегрованою. Основний шлях побудови інтегрованих мереж – використання модулів, виконаних відповідно до відкритих стандартів і специфікацій.

Якість обслуговування

Якість обслуговування *QoS* визначає кількісні оцінки вірогідності того, що мережа передаватиме певний потік даних між двома вузлами відповідно до потреб застосування або користувача. Наприклад, при передачі голосового трафіку через мережу під якістю обслуговування найчастіше розуміють гарантії того, що голосові пакети доставлятимуться мережею із затримкою не більш N мс, при цьому *варіація затримки* не перевищить M мс, і ці характеристики стануть витримуватися мережею з вірогідністю 0,95 на певному тимчасовому інтервалі. Тобто програмі, яка передає голосовий трафік, важливо, щоб мережа гарантувала дотримання саме цього приведеного вище набору характеристик якості обслуговування. Файловому сервісу потрібні гарантії середньої смуги пропускання і розширення її на невеликих інтервалах часу до деякого максимального рівня для швидкої передачі пульсацій.

В ідеалі мережа повинна гарантувати особливі параметри якості обслуговування, сформульовані для кожної окремої програми. Проте механізми *QoS*, що із зрозумілих причин розробляються, і вже існуючі обмежуються рішенням більш простих завдань – гарантуванням якихось усереднених вимог, заданих для основних типів програм. Найчастіше параметри, що фігурують в різноманітних визначеннях якості обслуговування, регламентують наступні показники роботи мережі:

- пропускна спроможність;
- затримки передачі пакетів;
- рівень втрат і спотворень пакетів.

Якість обслуговування гарантується для деякого потоку даних. Нагадаємо, що потік даних – це послідовність пакетів, що мають деякі загальні ознаки, наприклад, адреса вузла-джерела, інформація, що ідентифікує тип застосування (номер порту *TCP/UDP*) і т.п. До потоків застосовуються такі поняття, як агрегація і диференціювання. Так, потік даних від одного комп'ютера може бути представлений як

сукупність потоків від різних програм, а потоки від комп'ютерів одного підприємства агреговані в один потік даних абонента деякого провайдера послуг.

Механізми підтримки якості обслуговування самі по собі не створюють пропускну спроможність. Мережа не може дати більш того, що має. Отже фактична пропускну спроможність каналів зв'язку і транзитного комунікаційного устаткування – це ресурси мережі, що є відправною точкою для роботи механізмів *QoS*. Механізми *QoS* тільки управляють розподілом наявної пропускну спроможності відповідно до вимог програм і настройок мережі. Найочевидніший спосіб перерозподілу пропускну спроможності мережі полягає в управлінні чергами пакетів.

Оскільки дані, якими обмінюються два кінцеві вузли, проходять через деяку кількість проміжних мережних пристроїв, таких як концентратори, комутатори і маршрутизатори, то підтримка *QoS* вимагає взаємодії всіх мережних елементів на шляху трафіку, тобто "з кінця в кінець" ("end-to-end", "e2e"). Будь-які гарантії *QoS* настільки відповідають дійсності, наскільки їх забезпечує найбільш "слабкий" елемент в ланцюжку між відправником і одержувачем. Тому потрібно чітко розуміти, що підтримка *QoS* тільки в одному мережному пристрої, нехай навіть і магістральному, може лише вельми трохи поліпшити якість обслуговування або ж зовсім не вплинути на параметри *QoS*.

Реалізація в комп'ютерних мережах механізмів підтримки *QoS* є порівняно новою тенденцією. Довгий час комп'ютерні мережі існували без таких механізмів, і це пояснюється в основному двома причинами. По-перше, більшість програм, що виконуються в мережі, були "невимогливими", тобто для таких програм затримки пакетів або відхилення середньої пропускну спроможності в достатньо широкому діапазоні не приводили до значної втрати функціональності. Прикладами "невимогливих" програм є найбільш поширені в мережах 80-х років програми електронної пошти або віддаленого копіювання файлів.

По-друге, сама пропускну спроможність 10-мегабітних мереж *Ethernet* у багатьох випадках не була дефіцитом. Так, сегмент *Ethernet*, що розділяється, до якого було підключено 10-20 комп'ютерів, які зрідка копіюють невеликі текстові файли, об'єм яких не перевищує декілька сотень кілобайт, дозволяв трафіку кожної пари взаємодіючих комп'ютерів пересуватися по мережі так швидко, як потрібно тим програмам, які породили цей трафік.

В результаті більшість мереж працювали з тією якістю транспортного обслуговування, яка забезпечувала потреби програм. Правда, ніяких гарантій щодо контролю затримок пакетів або пропускну спроможності, з якою пакети передаються між вузлами в певних межах, ці мережі не давали. Більш того, при тимчасових перевантаженнях мережі, коли значна частина комп'ютерів одночасно починала передавати дані з максимальною швидкістю, затримки і пропускну спроможність ставали такими, що робота програмного забезпечення давала збій – йшла дуже поволі, з розривами сесій і т.п.

Існує два основні підходи до забезпечення якості роботи мережі. Перший полягає в тому, що мережа гарантує користувачеві дотримання деякої числової величини показника якості обслуговування. Наприклад, мережі *Frame relay* і *ATM* можуть гарантувати користувачеві заданий рівень пропускну спроможності. При другому підході (*best effort*) мережа старається по можливості якісніше обслужити користувача, але нічого при цьому не гарантує.

Транспортний сервіс, який надавали такі мережі, отримав назву "*Best effort*", тобто сервіс "з максимальними зусиллями" (або "по можливості"). Мережа прагне обробити трафік, що поступає, щонайшвидше, але при цьому ніяких гарантій щодо результату не дає. Прикладами можуть служити більшість технологій, розроблених в 80-і роки: *Ethernet*, *Token Ring*, *IP*, *X.25*. Сервіс "з максимальними зусиллями" заснований на деякому справедливому алгоритмі обробки черг, що виникають при перевантаженнях мережі, коли протягом деякого часу швидкість надходження пакетів в мережу перевищує швидкість просування цих пакетів. У простому випадку алгоритм обробки черги розглядає пакети всіх потоків як рівноправні і просуває їх в порядку надходження (*First In - First Out, FIFO*). У тому випадку, коли черга стає дуже великою (не вміщується в буфері), проблема вирішується простим відкиданням нових пакетів, що поступають.

Очевидно, що сервіс "з максимальними зусиллями" забезпечує прийнятну якість обслуговування тільки в тих випадках, коли продуктивність мережі набагато перевищує середні потреби, тобто є надмірною. У такій мережі пропускну спроможність достатня навіть для підтримки трафіку пікових періодів навантаження. Також очевидно, що таке рішення не економічне – принаймні, по відношенню до пропускну спроможностей сьогоденних технологій і інфраструктур, особливо для глобальних мереж. Проте, побудова мереж з надмірною пропускну спроможністю, будучи найпростішим способом забезпечення потрібного рівня якості обслуговування, іноді застосовується на практиці. Наприклад, деякі провайдери послуг мереж *TCP/IP* надають гарантію якісного обслуговування, постійно підтримуючи певний рівень перевищення пропускну спроможності своїх магістралей в порівнянні з потребами клієнтів.

В умовах, коли багато механізмів підтримки якості обслуговування тільки розробляються, використання для цих цілей надмірної пропускну спроможності часто виявляється єдиною можливим, хоч і тимчасовим рішенням.

1.2.2. Модульність і стандартизація

Модульність – це одна з невід'ємних властивостей обчислювальних мереж. Модульність виявляється не тільки в багаторівневому представленні комунікаційних протоколів в кінцевих вузлах мережі, хоча це,

безумовно, важлива і принципова особливість мережної архітектури. Мережа складається з величезного числа різних модулів – комп'ютерів, мережних адаптерів, мостів, маршрутизаторів, модемів, операційних систем і модулів програм.

Різноманітні вимоги, що пред'являються підприємствами до комп'ютерних мереж, привели до появи численних і різноманітних пристроїв і програм для побудови мережі. Ці продукти відрізняються не тільки основними функціями (наприклад, функціями, які виконують повторювачі, мости або програмні ридиректорами), але і численними допоміжними функціями, що надають користувачам або адміністраторам додаткові зручності, такі як автоматизована конфігурація параметрів пристрою, автоматичне виявлення і усунення деяких несправностей, можливість програмної зміни зв'язків в мережі і т.п. Різноманітність збільшується також тому, що багато пристроїв і програми відрізняються поєднаннями тих або інших основних і додаткових функцій – існують, наприклад, пристрої, які об'єднують в собі основні можливості комутаторів і маршрутизаторів, до яких додається ще і набір деяких додаткових функцій, характерний тільки для даного продукту.

В результаті не існує компанії, яка змогла б забезпечити виробництво повного набору всіх типів і підтипів устаткування і програмного забезпечення, необхідного для побудови мережі. Але, оскільки всі компоненти мережі повинні працювати погоджено, потрібно було приймати численні стандарти, які, якщо не у всіх, то хоч би в більшості випадків, гарантували б сумісність устаткування і програм різних фірм-виробників. Таким чином, поняття "модульність" і "стандартизація" в мережах нерозривно зв'язані, і модульний підхід тільки тоді дає переваги, коли він супроводжується відповідністю стандартам.

В результаті відкритий характер стандартів і специфікацій важливий не тільки для комунікаційних протоколів, але і для всіх численних функцій різноманітних пристроїв і програм, що випускаються для побудови мережі. Потрібно відзначити, що більшість стандартів, які приймаються сьогодні, носить відкритий характер. Час закритих систем, точні специфікації на які були відомі тільки фірмі-виробникові, пройшов. Стало очевидним, що можливість взаємодії з продуктами конкурентів не знижує, а навпаки, підвищує цінність виробу, оскільки його можна застосувати в більшій кількості працюючих мереж, побудованих на основі продуктів різних виробників. Тому навіть компанії, які раніше випускали вельми закриті системи, такі як *IBM*, *Novell* або *Microsoft*, сьогодні беруть активну участь в розробці відкритих стандартів і застосовують їх в своїх продуктах.

Сьогодні в секторі мережного устаткування і програмного забезпечення з питання сумісності продуктів різних виробників склалася наступна ситуація. Практично всі продукти, як програмні, так і апаратні, сумісні по функціях і властивостях, які були реалізовані вже достатньо давно, і відповідні стандарти розроблені і прийняті, принаймні, 3-4 року назад. В той же час дуже часто принципово нові пристрої, протоколи і властивості виявляються несумісними навіть у провідних виробників. Така картина характерна не тільки для тих пристроїв або функцій, стандарти на які ще не встигли прийняти (це природно), але і для пристроїв, стандарти на які існують вже декілька років. Сумісність досягається тільки після того, як всі виробники реалізують відповідний стандарт в своїх виробках, причому однаково.

1.2.3. Джерела стандартів

Роботи по стандартизації обчислювальних мереж ведуться великою кількістю організацій. Залежно від статусу організацій розрізняють наступні види стандартів:

- стандарти окремих фірм (наприклад, стек протоколів *DECnet* компанії *Digital Equipment* або графічний інтерфейс *OPEN LOOK* для *Unix*-систем компанії *Sun*);

- стандарти спеціальних комітетів і об'єднань, які створюються декількома фірмами, наприклад, стандарти технології *ATM*, що розробляються спеціально створеним об'єднанням *ATM Forum*, що налічує близько 100 колективних учасників, або стандартів союзу *Fast Ethernet Alliance* по розробці стандартів 100 Мбіт *Ethernet*;

- національні стандарти, наприклад, стандарт *FDDI*, один з численних стандартів, розроблених Американським національним інститутом стандартів (*ANSI*), або стандарти безпеки для операційних систем, розроблені Національним центром комп'ютерної безпеки (*NCSC*) Міністерства оборони США;

- міжнародні стандарти, наприклад, модель і стек комунікаційних протоколів Міжнародної організації по стандартизації (*ISO*), численні стандарти Міжнародного союзу електрозв'язку (*ITU*), зокрема стандарти на мережі з комутацією пакетів *X.25*, мережі *Frame relay*, *ISDN*, модеми і багато інших.

Деякі стандарти, безперервно розвиваючись, можуть переходити з однієї категорії в іншу. Зокрема, фірмові стандарти на продукцію, що набула широкого поширення, зазвичай стають міжнародними стандартами де-факто, оскільки вимушують виробників з різних країн слідувати фірмовим стандартам, щоб забезпечити сумісність своїх виробів з цими популярними продуктами. Наприклад, через феноменальний успіх персонального комп'ютера компанії *IBM* фірмовий стандарт на архітектуру *IBM PC* став міжнародним стандартом де-факто.

Більш того, зважаючи на широке розповсюдження, деякі фірмові стандарти стають основою для національних і міжнародних стандартів де-юре. Наприклад, стандарт *Ethernet* спочатку розроблений компаніями *Digital Equipment*, *Intel* і *Xerox*, через деякий час і в декілька зміненому вигляді був прийнятий

як національний стандарт *IEEE 802.3*, а потім організація *ISO* затвердила його як міжнародний стандарт *ISO 8802.3*.

Далі приводяться короткі відомості про організації, які найактивніше і найуспішніше займаються розробкою стандартів в області обчислювальних мереж.

Міжнародна організація по стандартизації (International Organization for Standardization, *ISO*, часто звана також International Standards Organization) є асоціацією провідних національних організацій різних країн по стандартизації. Головним досягненням *ISO* стала модель взаємодії відкритих систем *OSI*, яка в даний час є концептуальною основою стандартизації в області обчислювальних мереж. Відповідно до моделі *OSI* цією організацією був розроблений стандартний стек комунікаційних протоколів *OSI*.

Міжнародний союз електрозв'язку (International Telecommunications Union, *ITU*) – організація, яка в даний час є спеціалізованим органом Організації Об'єднаних Націй. Найбільш значну роль в стандартизації обчислювальних мереж грає Міжнародний консультативний комітет, що постійно діє в рамках цієї організації, з телефонії і телеграфії (МККТТ) (Consultative Committee on International Telegraphy and Telephony, *CCITT*). В результаті проведеної в 1993 році реорганізації *ITU CCITT* дещо змінив напрям своєї діяльності і змінив назву – тепер він називається сектором телекомунікаційної стандартизації *ITU* (*ITU Telecommunication Standardization Sector, ITU-T*). Основу діяльності *ITU-T* складає розробка міжнародних стандартів в області телефонії, телематических служб (електронної пошти, зв'язку факсиміле, телетексту, телексу і т. д.), передачі даних, аудио- і відеосигналів. За роки своєї діяльності *ITU-T* випустив величезну кількість рекомендацій-стандартів. Свою роботу *ITU-T* будує на вивченні досвіду різних організацій, а також на результатах власних досліджень. Раз на чотири роки видаються праці *ITU-T* у вигляді так званої "Книги", яка насправді є цілим набором звичайних книг, згрупованих у випуски, які, у свою чергу, об'єднуються в томи. Кожен том і випуск містять логічно взаємозв'язані рекомендації. Наприклад, том III Синьої Книги містить рекомендації для цифрових мереж з інтеграцією послуг (*ISDN*), а весь том VIII (за винятком випуску VIII.1, який містить рекомендації серії V для передачі даних по телефонній мережі) присвячений рекомендаціям серії X: X.25 для мереж з комутацією пакетів, X.400 для систем електронної пошти, X.500 для глобальної довідкової служби і багатьом іншим.

Інститут інженерів по електротехніці і радіоелектроніці (Institute of Electrical and Electronics Engineers, *IEEE*) – національна організація США, що визначає мережні стандарти. У 1981 році робоча група 802 цього інституту сформулювала основні вимоги, яким повинні задовольняти локальні обчислювальні мережі. Група 802 визначила безліч стандартів, з них найвідомішими є стандарти 802.1, 802.2, 802.3 і 802.5, які описують загальні поняття, які використовуються в області локальних мереж, а також стандарти на два нижні рівні мереж *Ethernet* і *Token Ring*.

Європейська асоціація виробників комп'ютерів (European Computer Manufacturers Association, *ECMA*) – некомерційна організація, що активно співробітничав з *ITU-T* і *ISO*, займається розробкою стандартів і технічних оглядів, що відносяться до комп'ютерної і комунікаційної технологій. Вона відома своїм стандартом *ECMA-101*, який використовується при передачі відформатованих текстів і графічних зображень із збереженням оригінального формату.

Асоціація виробників комп'ютерів і оргтехніки (Computer and Business Equipment Manufacturers Association, *CBEMA*) – організація американських виробників апаратного забезпечення, аналогічна європейській асоціації *EKMA*. Вона бере участь в розробці стандартів на обробку інформації і відповідне устаткування.

Асоціація електронної промисловості (Electronic Industries Association, *EIA*) – промислово-торгова група виробників електронного і мережного устаткування, є національною комерційною асоціацією США, проявляє значну активність в розробці стандартів для проводів, конекторів і інших мережних компонентів. *RS232C* – найбільш відомий стандарт *EIA*.

Міністерство оборони США (Department of Defense, *DoD*) має численні підрозділи, що займаються створенням стандартів для комп'ютерних систем. Однією з найвідоміших розробок *DoD* є стек транспортних протоколів *TCP/IP*.

Американський національний інститут стандартів (American National Standards Institute, *ANSI*) представляє США в Міжнародній організації по стандартизації *ISO*. Комітети *ANSI* займаються розробкою стандартів в різних областях обчислювальної техніки. Так, комітет *ANSI X3T9.5* спільно з компанією *IBM* здійснює стандартизацію локальних мереж великих ЕОМ (архітектура мереж *SNA*). Відомий стандарт *FDDI* також є результатом діяльності цього комітету *ANSI*. В області мікрокомп'ютерів *ANSI* розробляє стандарти на мови програмування, інтерфейс *SCSI*. *ANSI* розробив рекомендації по переносимості для мов *C*, *FORTRAN*, *COBOL*.

1.2.4. Стандарти *Internet*

Особливу роль у виробленні міжнародних відкритих стандартів грають стандарти *Internet*. Зважаючи на популярність *Internet*, яка постійно росте, ці стандарти стають міжнародними стандартами "де-факто", і багато з них набувають згодом статус офіційних міжнародних стандартів за рахунок затвердження однієї з вищеперелічених організацій, зокрема *ISO* і *ITU-T*. Існує декілька організаційних підрозділів, які відповідають за розвиток *Internet* і, зокрема, за стандартизацію засобів *Internet* (рис. 1.6).

Основним з них є *Internet Society (ISOC)* – професійне співтовариство, яке займається загальними питаннями еволюції і зростання *Internet* як глобальної комунікаційної інфраструктури. Під управлінням *ISOC* працює *Internet Architecture Board (IAB)* – організація, у веденні якої знаходиться технічний контроль і координація робіт для *Internet*. *IAB* координує напрям досліджень і нових розробок для стека *TCP/IP* і є кінцевою інстанцією при визначенні нових стандартів *Internet*.

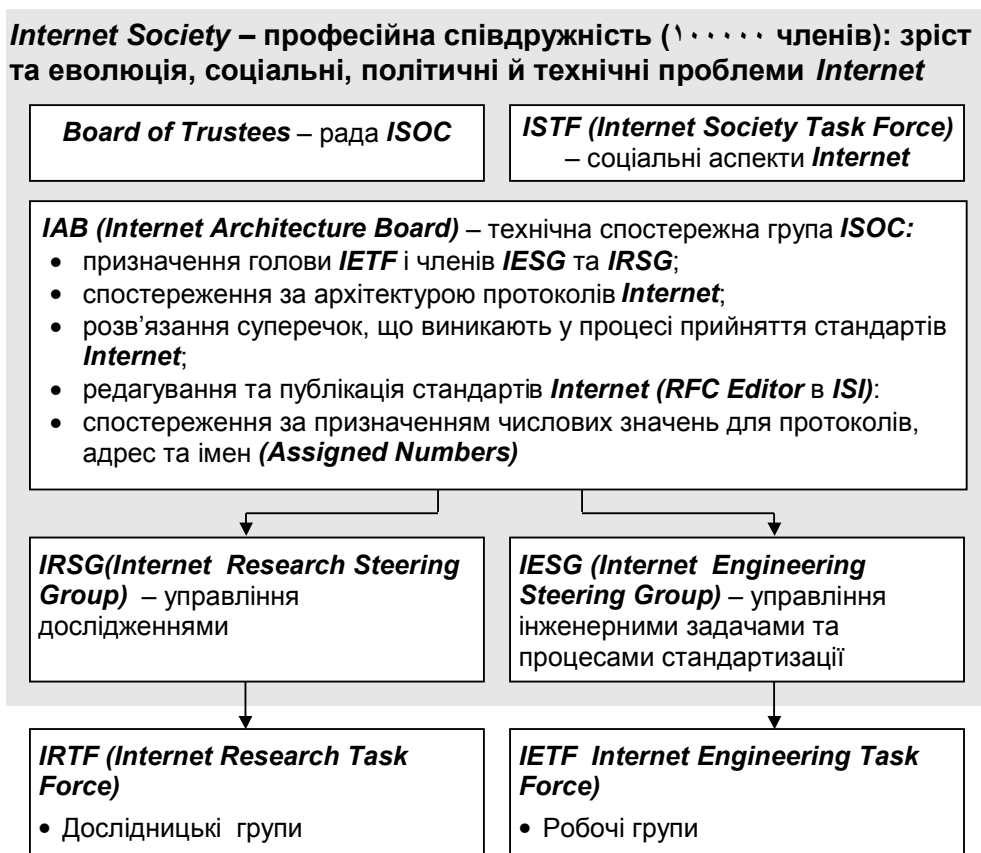


Рис. 1.6. Стандартизація *Internet*.

До *IAB* входить дві основні групи: *Internet Engineering Task Force (IETF)* і *Internet Research Task Force (IRTF)*. *IETF* – це інженерна група, яка займається вирішенням найбільш актуальних технічних проблем *Internet*. Саме *IETF* визначає специфікації, які потім стають стандартами *Internet*. У свою чергу, *IRTF* координує довгострокові дослідницькі проекти по протоколах *TCP/IP*.

У будь-якій організації, яка займається стандартизацією, процес вироблення і ухвалення стандарту складається з ряду обов'язкових етапів, які, власне, і складають процедуру стандартизації. Розглянемо цю процедуру на прикладі розробки стандартів *Internet*. На рис. 1.7 показана схема проходження стандарту через всі етапи:

1. Спочатку в *IETF* представляється так званий робочий проект (draft) у вигляді, доступному для коментарів (на малюнку даний етап позначений enter). Він публікується в *Internet*, після чого широкий круг зацікавлених осіб включається в обговорення цього документа, до нього вносяться виправлення, і, нарешті, настає момент, коли можна зафіксувати зміст документа. На даному етапі проекту привласнюється номер RFC (можливий і інший варіант розвитку подій – після обговорення робочий проект відкидається і віддаляється з *Internet*).

2. Після привласнення номера проекту набуває статус пропонованого стандарту (на малюнку proposed). Протягом 6 місяців цей пропонований стандарт проходить перевірку практикою, в результаті до нього вносяться зміни.

3. Якщо результати практичних досліджень свідчать про ефективність пропонованого стандарту, то йому, зі всіма внесеними змінами, привласнюється статус проекту стандарту (на малюнку draft std). Потім в перебіг як мінімум 4-х місяців проходять його подальші випробування "на міцність", при цьому створюється принаймні дві програмні реалізації.

Якщо під час перебування в ранзі проекту стандарту до документа не було внесено ніяких виправлень, йому може бути привласнений статус офіційного стандарту *Internet* (на рисунку standard).

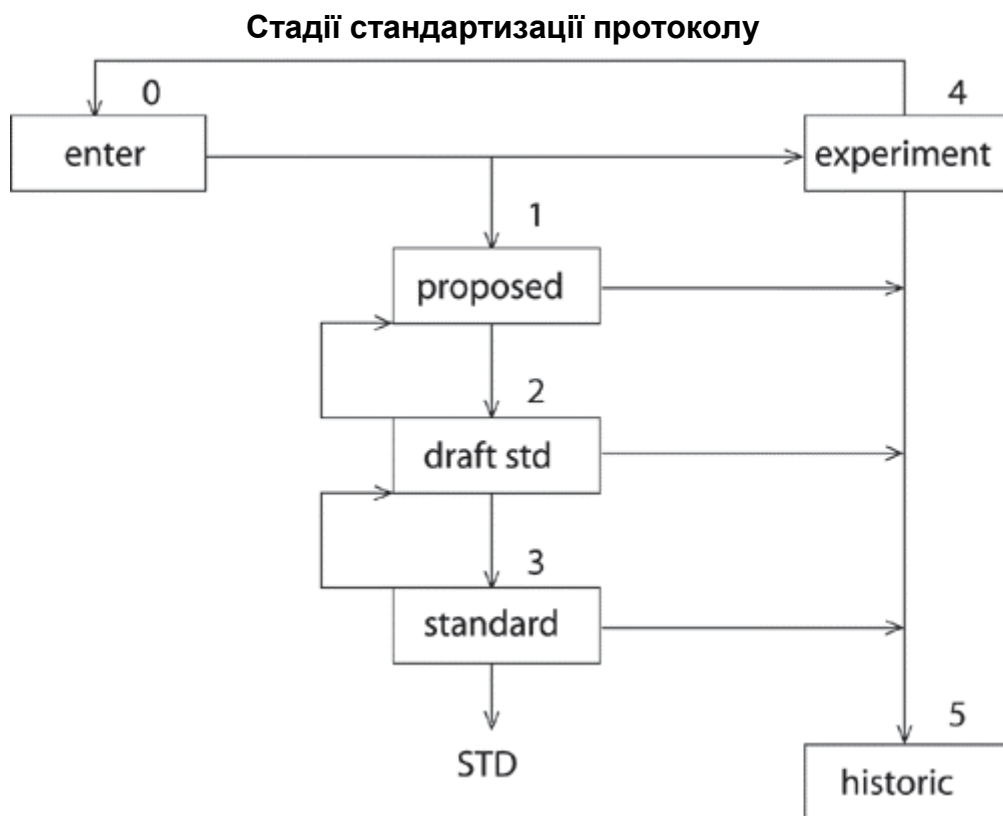


Рис. 1.7. Стадії стандартизації протоколу *Internet*.

Слід відмітити, що всі стандарти *Internet* носять назву *RFC* з відповідним порядковим номером, але далеко не всі *RFC* є *стандартами Internet* – часто ці документи є коментарі до якого-небудь стандарту або просто описи деякої проблеми *Internet*. Список затверджених офіційних *стандартів Internet* публікується у вигляді документа *RFC* і доступний в *Internet*, наприклад, за адресою <http://www.internic.net/>

1.2.5. Поява стандартних технологій локальних мереж

В середині 80-х років положення справ в локальних мережах стало мінятися. Затвердилися стандартні технології об'єднання комп'ютерів в мережу – *Ethernet*, *Arcnet*, *Token Ring*, *Token Bus*, декілька пізніше – *FDDI*. Всі стандартні технології локальних мереж спиралися на той же принцип комутації, який був з успіхом випробуваний і довів свої переваги при передачі трафіку даних в глобальних комп'ютерних мережах – принцип комутації пакетів.

Стандартні мережні технології зробили завдання побудови локальної мережі майже тривіальною. Для створення мережі досить було придбати мережні адаптери відповідного стандарту, наприклад, *Ethernet*, стандартний кабель, приєднати адаптери до кабелю стандартними роз'ємами і встановити на комп'ютер одну з популярних мережних операційних систем, наприклад, *Novell NetWare*. Після цього мережа починала працювати, і подальше приєднання кожного нового комп'ютера не викликало жодних проблем – природно, якщо на ній був встановлений мережний адаптер тієї ж технології.

У 80-і роки були прийняті основні стандарти на комунікаційні технології для локальних мереж: у 1980 році – *Ethernet*, в 1985 – *Token Ring*, в кінці 80-х – *FDDI*. Це дозволило забезпечити сумісність мережних операційних систем на нижніх рівнях, а також стандартизувати інтерфейс ОС з драйверами мережних адаптерів.

Кінець 90-х виявив явного лідера серед технологій локальних мереж – сімейство *Ethernet*, в яке увійшли класична технологія *Ethernet* 10 Мбіт/с, а також *Fast Ethernet* 100 Мбіт/с і *Gigabit Ethernet* 1000 Мбіт/с. Прості алгоритми роботи зумовили низьку вартість устаткування *Ethernet*. Широкий діапазон ієрархії швидкостей дозволяє раціонально будувати локальну мережу, застосовуючи ту технологію, яка найбільшою мірою відповідає завданням підприємства і потребам користувачів. Важливо також, що всі технології *Ethernet* дуже близькі один одному за принципами роботи, що спрощує обслуговування і інтеграцію побудованих на їх основі мереж.

1.2.6. Модель OSI

Теза про користь стандартизації є справедливою для всіх галузей, в комп'ютерних мережах набуває особливого значення. Суть мережі – це з'єднання різного устаткування, а значить, проблема сумісності є однією з найбільш гострих. Без дотримання всіма виробниками загальноприйнятих правил розробки

устаткування прогрес в справі "будівництва" мереж був би неможливий. Тому весь розвиток комп'ютерної галузі, кінець кінцем, відображений в стандартах – будь-яка нова технологія тільки тоді набуває "законного" статусу, коли її зміст закріплюється у відповідному стандарті.

У комп'ютерних мережах ідеологічною основою стандартизації є багаторівневий підхід до розробки засобів мережної взаємодії. Саме на основі цього підходу була створена стандартна семирівнева модель взаємодії відкритих систем, що стала свого роду універсальною мовою мережних фахівців.

1.2.7. Багаторівневий підхід. Декомпозиція задач мережної взаємодії

Організація взаємодії між пристроями мережі є складним завданням. Як відомо, для вирішення складних задач використовується універсальний прийом – декомпозиція, тобто розбиття однієї задачі на декілька задач-модулів (рис. 1.8). Декомпозиція полягає в чіткому визначенні функцій кожного модуля, а також порядку їх взаємодії (інтерфейсів). В результаті досягається логічне спрощення задачі, а, крім того, з'являється можливість модифікації окремих модулів без зміни решті частини системи.

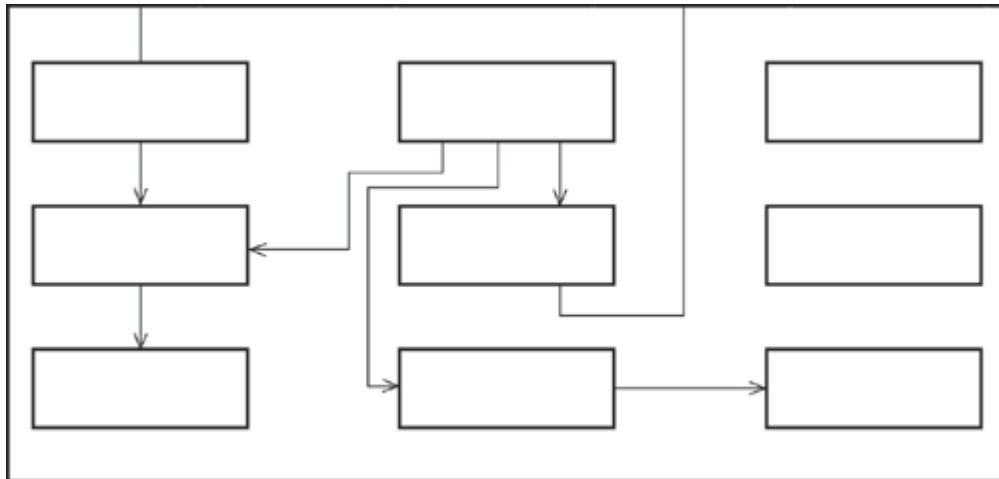


Рис. 1.8. Приклад декомпозиції задачі.

При декомпозиції часто використовують багаторівневий підхід. Він полягає в наступному:

- всю множину модулів, в яких вирішуються окремі задачі, розбивають на групи і упорядковують по рівнях, які створюють ієрархію;
- відповідно до принципу ієрархії для кожного проміжного рівня можна вказати сусідні, вище розміщені і нижче розміщені рівні, які безпосередньо примикають до нього (рис. 1.9);
- група модулів, які складають кожен рівень, повинна бути сформована так, щоб всі модулі цієї групи для виконання своїх завдань зверталися із запитом тільки до модулів сусіднього нижче розміщеного рівня;
- з іншого боку, результати роботи всіх модулів, віднесених до деякого рівня, можуть бути передані тільки модулям сусіднього вище розміщеного рівня.

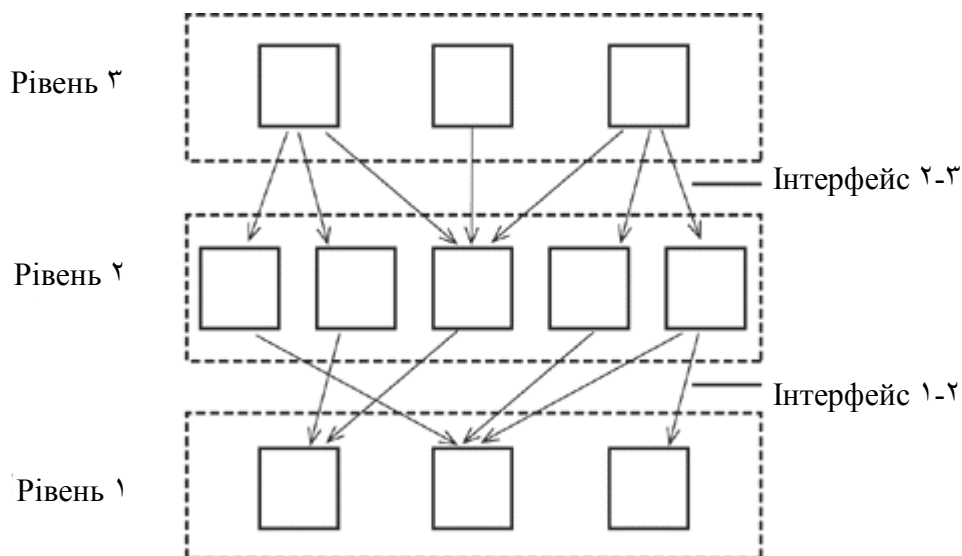


Рис. 1.9. Багаторівневий підхід – створення ієрархії завдань.

Така ієрархічна декомпозиція завдання припускає чітке визначення функції кожного рівня і інтерфейсів між рівнями. Інтерфейс визначає набір функцій, які нижче розміщений рівень надає вище розміщеному. В результаті ієрархічної декомпозиції досягається відносна незалежність рівнів, а значить, можливість їх автономної розробки і модифікації.

Засоби рішення задачі організації мережної взаємодії, звичайно, теж можуть бути представлені у вигляді ієрархічно організованої безлічі модулів. Наприклад, модулям нижнього рівня можна доручити питання, пов'язані з надійною передачею інформації між двома сусідніми вузлами, а модулям наступного, вищого, рівня – транспортування повідомлень в межах всієї мережі. Очевидно, що останнє завдання – організація зв'язку двох будь-яких, не обов'язково сусідніх, вузлів – є більш загальною і тому її можна вирішити за допомогою багатократних звернень до нижче розміщеного рівня. Так, скріплення вузлів А і Б (рис. 1.10) може бути зведено до послідовного скріплення пар проміжних суміжних вузлів. Таким чином, модулі вище розміщеного рівня при рішенні своїх задач розглядають засоби нижнього рівня як інструмент.

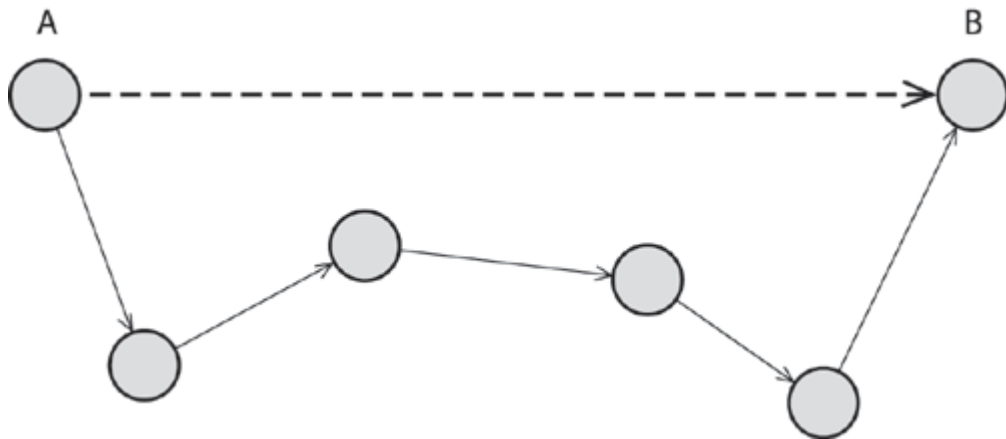


Рис. 1.10. Декомпозиція задачі скріплення довільної пари вузлів на більш приватні завдання скріплення пар сусідніх вузлів.

1.2.8. Загальна характеристика моделі OSI

З того, що протокол є угода, прийнята двома взаємодіючими об'єктами, в даному випадку двома комп'ютерами, які працюють в мережі, зовсім не витікає, що він обов'язково є стандартним. Але на практиці при реалізації мереж зазвичай використовуються стандартні протоколи. Це можуть бути фірмові, національні або міжнародні стандарти.

На початку 80-х років ряд міжнародних організацій по стандартизації – *ISO*, *ITU-T* і деякі інші розробили модель, яка зіграла значну роль в розвитку мереж. Ця модель називається моделлю *ISO/OSI*. Модель взаємодії відкритих систем (Open System Interconnection, *OSI*) визначає різні рівні взаємодії систем в мережах з комутацією пакетів, дає їм стандартні імена і вказує, які функції повинен виконувати кожен рівень. Модель *OSI* була розроблена на підставі великого досвіду, отриманого при створенні комп'ютерних мереж, в основному глобальних, в 70-і роки. Повний опис цієї моделі займає більше 1000 сторінок тексту.

У моделі *OSI* (рис. 1.11) засоби взаємодії діляться на сім рівнів: прикладний, представницький, сеансовий, транспортний, мережний, каналний і фізичний. Кожен рівень має справу з певним аспектом взаємодії мережних пристроїв.

формату. Звичайне повідомлення складається із заголовка і поля даних. Заголовок містить службову інформацію, яку необхідно передати через мережу прикладному рівню машини-адресата, щоб повідомити його, яку роботу треба виконати. У нашому випадку заголовок, очевидно, повинен містити інформацію про місцезнаходження файлу і про тип операції, яку необхідно виконати. Поле даних повідомлення може бути порожнім або містити які-небудь дані, наприклад ті, які необхідно записати у віддалений файл. Але для того, щоб доставити цю інформацію за призначенням, належить вирішити ще багато завдань, відповідальність за які несуть рівні, що лежать нижче.

Після формування повідомлення прикладний рівень направляє його вниз по стеку представницькому рівню. Протокол представницького рівня на підставі інформації, отриманої із заголовка прикладного рівня, виконує необхідні дії і додає до повідомлення власну службову інформацію – заголовок представницького рівня, в якому містяться вказівки для протоколу представницького рівня машини-адресата. Отримане в результаті повідомлення передається вниз сеансовому рівню, який у свою чергу додає свій заголовок, і т.д. (Деякі протоколи поміщають службову інформацію не тільки на початку повідомлення у вигляді заголовка, але і в кінці, у вигляді так званого "кінцевика".) Нарешті, повідомлення досягає нижнього, фізичного рівня, який, власне, і передає його по лінії зв'язку машині-адресатові. До цього моменту повідомлення "обростає" заголовками всіх рівнів (рис. 1.12).

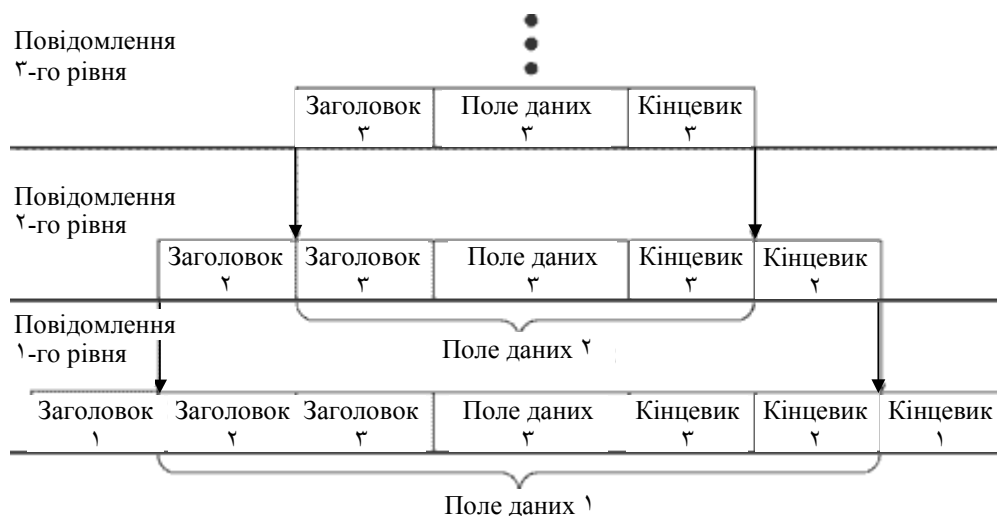


Рис. 1.12. Вкладеність повідомлень різних рівнів.

Коли повідомлення по мережі поступає на машину-адресат, воно приймається її фізичним рівнем і послідовно переміщується вгору з рівня на рівень. Кожен рівень аналізує і обробляє заголовок свого рівня, виконуючи відповідні даному рівню функції, а потім видаляє цей заголовок і передає повідомлення вище розміщеного рівня.

Разом з терміном повідомлення (message) існують і інші терміни, які вживаються мережними фахівцями для позначення одиниць даних в процедурах обміну. У стандартах *ISO* для позначення одиниць даних, з якими мають справу протоколи різних рівнів, використовується загальна назва – протокольний блок даних (Protocol Data Unit, PDU). Для позначення блоків даних певних рівнів часто використовуються спеціальні назви: кадр (frame), пакет (packet), дейтаграма (datagram), сегмент (segment).

Фізичний рівень

Фізичний рівень (Physical layer) має справу з передачею бітів по фізичних каналах зв'язку, таким, як коаксіальний кабель, вита пара, оптоволоконний кабель або цифровий територіальний канал. До цього рівня мають відношення характеристики фізичних середовищ передачі даних, такі як смуга пропускання, перешкодостійкість, хвильовий опір та інші. На цьому ж рівні визначаються характеристики електричних сигналів, що передають дискретну інформацію, таку як крутизна фронтів імпульсів, рівні напруги або струму переданого сигналу, тип кодування, швидкість передачі сигналів. Крім того, тут стандартизуються типи роз'ємів і призначення кожного контакту.

Фізичний рівень реалізується апаратно, і на цьому рівні вирішуються такі задачі:

- передача бітів по фізичних каналах;
- формування електричних сигналів;
- кодування інформації;
- синхронізація;
- модуляція.

Функції фізичного рівня реалізуються у всіх пристроях, підключених до мережі. З боку комп'ютера функції фізичного рівня виконуються мережним адаптером або послідовним портом. Прикладом протоколу фізичного рівня може служити специфікація 10Base-T технології *Ethernet*, яка визначає як кабель

використовує неекрановану виту пару категорії 3 з хвильовим опором 100 Ом, роз'єм RJ-45, максимальну довжину *фізичного сегменту* 100 метрів, манчестерський код для представлення даних в кабелі, а також деякі інші характеристики середовища і *електричних сигналів*.

Канальний рівень

На фізичному рівні просто пересилаються біти. При цьому не враховується, що в тих мережах, в яких лінії зв'язку використовуються (розділяються) поперемінно декількома парами взаємодіючих комп'ютерів, фізичне середовище передачі може бути зайняте. Тому одним із завдань канального рівня (*Data Link layer*) є перевірка доступності середовища передачі. Інше завдання канального рівня – реалізація механізмів виявлення і корекції помилок. Для цього на канальному рівні біти групуються в набори, звані кадрами (*frames*). Канальний рівень забезпечує коректність передачі кожного кадру поміщаючи спеціальну послідовність біт в початок і кінець кожного кадру, для його виділення, а також обчислює контрольну суму, обробляючи всі байти кадру певним способом, і додає контрольну суму до кадру. Коли кадр приходить по мережі, одержувач знову обчислює контрольну суму отриманих даних і порівнює результат з контрольною сумою з кадру. Якщо вони співпадають, кадр вважається правильним і приймається. Якщо ж контрольні суми не співпадають, то фіксується помилка. Канальний рівень може не тільки виявляти помилки, але і виправляти їх за рахунок повторної передачі пошкоджених кадрів. Необхідно відзначити, що функція виправлення помилок для канального рівня не є обов'язковою, тому в деяких протоколах цього рівня вона відсутня, наприклад, в *Ethernet* і *Frame relay*.

Функції канального рівня реалізуються програмно-апаратно. Основна задача – доставка пакетів з надійністю не нижче заданої. Для цього необхідно вирішувати такі підзадачі:

- організація каналів передачі між двома сусідніми станціями в мережі з довільною топологією;
- організація каналів передачі між будь-якими станціями в мережі з типовою топологією;
- перевірка доступності середовища, що розділяється;
- виділення *кадрів* з потоку даних, що поступають по мережі;
- формування *кадрів* при відправці даних;
- підрахунок і перевірка контрольної суми.

У протоколах канального рівня, які використовуються в локальних мережах, закладена певна структура зв'язків між комп'ютерами і способи їх адресації. Хоча канальний рівень і забезпечує доставку кадру між будь-якими двома вузлами локальної мережі, він це робить тільки в мережі з певною топологією зв'язків, саме тією топологією, для якої він був розроблений. До таких типових топологій, які підтримуються протоколами канального рівня локальних мереж, відносяться "загальна шина", "кільце" і "зірка", а також структури, отримані з них за допомогою мостів і комутаторів. Прикладами протоколів канального рівня є протоколи *Ethernet*, *Token Ring*, *FDDI*, *10VG-AnyLAN*.

У локальних мережах протоколи *канального рівня* використовуються комп'ютерами, *мостами*, *комутаторами* і *маршрутизаторами*. У комп'ютерах функції *канального рівня* реалізуються спільними зусиллями *мережних адаптерів* і їх *драйверів*.

У глобальних мережах, які рідко володіють регулярною топологією, *канальний рівень* часто забезпечує обмін повідомленнями тільки між двома сусідніми комп'ютерами, сполученими індивідуальною лінією зв'язку. Прикладами протоколів "точка-точка" (як часто називають такі протоколи) можуть служити широко поширені протоколи *PPP* і *LAP-B*. У таких випадках для доставки повідомлень між кінцевими вузлами через всю мережу використовуються засоби мережного рівня. Саме так організовані мережі X.25. Іноді в глобальних мережах функції канального рівня в чистому вигляді виділити важко, оскільки в одному і тому ж протоколі вони об'єднуються з функціями мережного рівня. Прикладами такого підходу можуть служити протоколи технологій *ATM* і *Frame relay*.

В цілому канальний рівень – це могутній набір функцій по пересилці повідомлень між вузлами мережі. В деяких випадках протоколи канального рівня виявляються самодостатніми транспортними засобами, і тоді поверх них можуть безпосередньо працювати без залучення засобів мережного і транспортного рівнів протоколи прикладного рівня або програми. Наприклад, існує реалізація протоколу управління мережею *SNMP* безпосередньо поверх *Ethernet*, хоча стандартно цей протокол працює поверх мережного протоколу *IP* і транспортного протоколу *UDP*. Природно, що застосування такої реалізації буде обмеженим – вона не підходить для складених мереж різних технологій, наприклад, *Ethernet* і X.25, і навіть для такої мережі, в якій у всіх сегментах застосовується *Ethernet*, але між сегментами існують петлюваті зв'язки. А ось в двохсегментній мережі *Ethernet*, об'єднаній мостом, реалізація *SNMP* над канальним рівнем буде цілком працездатна.

Проте, для забезпечення якісного транспортування повідомлень в мережах будь-яких топологій і технологій функції канального рівня виявляється недостатньо, тому в моделі OSI рішення цієї задачі покладається на два наступні рівні – мережний і транспортний.

Канальний рівень забезпечує передачу пакетів даних, верхніх рівнів, що поступають від протоколів, вузлу призначення, адреса якого також указує протокол верхнього рівня. Протоколи канального рівня оформляють передані ним пакети в кадри власного формату, поміщаючи вказану адресу призначення в одне з полів такого кадру, а також супроводжуючи кадр контрольною сумою. Протокол канального рівня має локальне значення, він призначений для доставки кадрів даних, як правило, в межах мереж з простою

топологією зв'язків і однотипною або близькою технологією, наприклад, в односегментних мережах *Ethernet* або ж в багатосегментних мережах *Ethernet* і *Token Ring* ієрархічній топології, розділених тільки мостами і комутаторами. У всіх цих конфігураціях адрес призначення має локальне значення для даної мережі і не змінюється при проходженні кадру від вузла-джерела до вузла-призначення. Можливість передавати дані між локальними мережами різних технологій пов'язана з тим, що в цих технологіях використовуються адреси однакового формату, до того ж виробники мережних адаптерів забезпечують унікальність адреси незалежно від технології.

Іншою областю дії протоколів каналного рівня є зв'язки типу "точка-точка" глобальних мереж, коли протокол каналного рівня є відповідальним за доставку кадру безпосередньому сусідові. Адреса в цьому випадку не має принципового значення, а на перший план виходить здатність протоколу відновлювати спотворені і загублені кадри, оскільки погана якість територіальних каналів, особливо комутованих телефонних, часто вимагає виконання подібних дій. Якщо ж перераховані вище умови не дотримуються, наприклад, зв'язки між сегментами *Ethernet* мають петльовидну структуру, або об'єднані мережі використовують різні способи *адресації*, як в мережах *Ethernet* і *X.25*, то протокол каналного рівня не може поодиноці справитися із завданням передачі кадру між вузлами і вимагає допомоги протоколу мережного рівня.

Мережний рівень

Мережний рівень (*Network layer*) служить для утворення єдиної транспортної системи, яка об'єднує декілька мереж, причому ці мережі можуть використовувати різні принципи передачі повідомлень між кінцевими вузлами і володіти довільною структурою зв'язків. Функції мережного рівня достатньо різноманітні. Розглянемо їх на прикладі об'єднання локальних мереж.

Протоколи каналного рівня локальних мереж забезпечують доставку даних між будь-якими вузлами тільки в мережі з відповідною типовою топологією, наприклад, топологією ієрархічної зірки. Це жорстке обмеження, яке не дозволяє будувати мережі з розвиненою структурою, наприклад, мережі, які об'єднують декілька мереж підприємства в єдину мережу, або високонадійні мережі, в яких існують надмірні зв'язки між вузлами. Можна було б ускладнювати протоколи каналного рівня для підтримки надлишкових петльовидних зв'язків, але принцип розділення обов'язків між рівнями приводить до іншого рішення. Щоб, з одного боку, зберегти простоту процедур передачі даних для типових топологій, а з іншої – допустити використання довільних топологій, вводиться додатковий мережний рівень.

На мережному рівні сам термін "мережа" наділяють специфічним значенням. В даному випадку під мережею розуміється сукупність комп'ютерів, сполучених між собою відповідно до однієї із стандартних типових топологій, яка використовує для передачі даних один з протоколів каналного рівня, визначений для цієї топології.

В середині мережі доставка даних забезпечується відповідним каналним рівнем, а ось доставкою даних між мережами займається мережний рівень, який і підтримує можливість правильного вибору маршруту передачі повідомлення навіть у тому випадку, коли структура зв'язків між мережами має характер, відмінний від прийнятого в протоколах каналного рівня.

Мережі з'єднуються між собою спеціальними пристроями, званими маршрутизаторами. **Маршрутизатор** – це пристрій, який збирає інформацію про топологію міжмережних з'єднань і пересилає пакети мережного рівня в мережу призначення. Щоб передати повідомлення від відправника, який знаходиться в одній мережі, одержувачеві, що знаходиться в іншій мережі, потрібно зробити деяку кількість транзитних передач між мережами, або **хопів** (від слова *hop* – стрибок), кожного разу вибираючи відповідний маршрут. Таким чином, **маршрут** є послідовність маршрутизаторів, через які проходить пакет.

Задача мережного рівня – доставка пакету:

- між будь-якими двома вузлами мережі з довільною топологією;
- між будь-якими двома мережами в складеній мережі;
- у мережі – сукупності комп'ютерів, що використовують для обміну даними єдину мережну технологію;
- по маршруту-послідовності проходження пакетом маршрутизаторів в складеній мережі.

На рис. 1.13 показано чотири мережі, які зв'язані трьома маршрутизаторами. Між вузлами А і В даній мережі пролягає два маршрути: перший – через маршрутизатори 1 і 3, а другий – через маршрутизатори 1, 2 і 3.

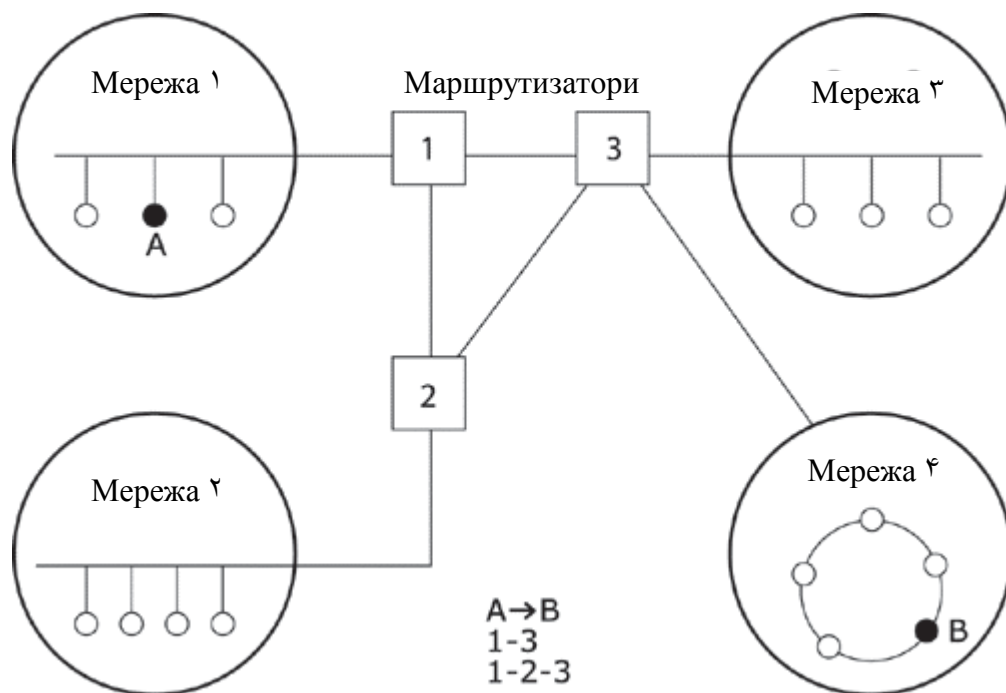


Рис. 1.13. Приклад складеної мережі.

Проблема вибору якнайкращого шляху називається маршрутизацією, і її рішення є одним з головних завдань мережного рівня. Ця проблема ускладнюється тим, що найкоротший шлях – не завжди найкращий. Часто критерієм при виборі маршруту є час передачі даних; він залежить від пропускної спроможності каналів зв'язку і інтенсивності трафіку, яка може з часом змінюватися. Деякі алгоритми маршрутизації намагаються пристосуватися до зміни навантаження, тоді як інші ухвалюють рішення на основі середніх показників за тривалий час. Вибір маршруту може здійснюватися і по інших критеріях, таких як надійність передачі.

У загальному випадку функції мережного рівня є ширшими, ніж функції передачі повідомлень по зв'язках з нестандартною структурою, які розглядалися на прикладі об'єднання декількох локальних мереж. Мережний рівень також вирішує задачі узгодження різних технологій, спрощення адресації в крупних мережах і створення надійних і гнучких бар'єрів на шляху небажаного трафіку між мережами.

Повідомлення мережного рівня прийнято називати пакетами (packet). При організації доставки пакетів на мережному рівні використовується поняття "номер мережі". В цьому випадку адреса одержувача складається із старшої частини – номеру мережі і молодшої – номеру вузла в цій мережі. Всі вузли однієї мережі повинні мати одну і ту ж старшу частину адреси, тому терміну "мережа" на мережному рівні можна дати і інше, формальніше, визначення: мережа – це сукупність вузлів, мережну адресу яких містить один і той же номер мережі.

На мережному рівні визначається два види протоколів. Перший вид – мережні протоколи (routed protocols) реалізують просування пакетів через мережу. Саме ці протоколи зазвичай мають на увазі, коли говорять про протоколи мережного рівня. Проте часто до мережного рівня відносять і інший вид протоколів, званих протоколами обміну маршрутною інформацією або просто протоколами маршрутизації (routing protocols). За допомогою цих протоколів маршрутизатори збирають інформацію про топологію міжмережних з'єднань. Протоколи мережного рівня реалізуються програмними модулями операційної системи, а також програмними і апаратними засобами маршрутизаторів.

На мережному рівні працюють протоколи ще одного типу, які відповідають за відображення адреси вузла, використовуваного на мережному рівні, в локальну адресу мережі. Такі протоколи часто називають протоколами дозволу адреси – *Address Resolution Protocol, ARP*. Іноді їх відносять не до мережного рівня, а до каналного, хоча тонкості класифікації не змінюють суті. Прикладами протоколів мережного рівня є протокол міжмережної взаємодії *IP* стека *TCP/IP* і протокол міжмережного обміну пакетами *IPX* стека *Novell*.

Транспортний рівень

На шляху від відправника до одержувача пакети можуть бути спотворені або загублені. Хоча деякі програми мають власні засоби обробки помилок, існують і такі, які вважають за краще відразу мати справу з надійним з'єднанням. Транспортний рівень (Transport layer) забезпечує програмам або верхнім рівням стека, прикладному і сеансовому, передачу даних з тим ступенем надійності, яка їм потрібна. Модель *OSI* визначає п'ять класів сервісу, які надаються транспортним рівнем. Ці види сервісу відрізняються якістю послуг, що надаються: терміновістю, можливістю відновлення перерваного зв'язку, наявністю засобів

мультиплексування декількох з'єднань між різними прикладними протоколами через загальний транспортний протокол, а головне – здібністю до виявлення і виправлення помилок передачі, таких як спотворення, втрата і дублювання пакетів.

Вибір класу сервісу транспортного рівня визначається, з одного боку, тим, в якому ступені завдання забезпечення надійності вирішується самими програмами і протоколами вищих, ніж транспортний, рівнів, а з іншого боку, залежить від того, наскільки надійною є система транспортування даних в мережі, яка забезпечується рівнями, розташованими нижче транспортного, – мережним, канальним і фізичним. Так, наприклад, якщо якість каналів передачі зв'язку дуже висока, і вірогідність наявності помилок, не виявлених протоколами нижчих рівнів, невелика, варто скористатися одним з полегшених сервісів транспортного рівня, необтяжених численними перевірками і різними прийомами підвищення надійності. Якщо ж транспортні засоби нижніх рівнів спочатку дуже ненадійні, то доцільно звернутися до найбільш розвиненого сервісу транспортного рівня, який працює, використовуючи максимум засобів для виявлення і усунення помилок, за допомогою попереднього встановлення логічного з'єднання, відстежування доставки повідомлень по контрольних сумах і циклічній нумерації пакетів, встановлення тайм-аутів доставки і т.п.

На транспортному рівні вирішуються задачі забезпечення доставки інформації з необхідною якістю між будь-якими вузлами мережі, в тому числі:

- розбиття повідомлення сеансового рівня на пакети, їх нумерація;
- буферизація пакетів, що приймаються;
- впорядкування пакетів, що прибувають;
- адресація прикладних процесів;
- управління потоком.

Як правило, всі протоколи, починаючи з транспортного рівня і вище, реалізуються програмними засобами кінцевих вузлів мережі – компонентами їх мережних операційних систем. Як приклад транспортних протоколів можна привести протоколи TCP і UDP *стека TCP/IP* і протокол SPX *стека Novell*.

Протоколи чотирьох нижніх рівнів узагальнено називають мережним транспортом або транспортною підсистемою, оскільки вони повністю вирішують задачу транспортування повідомлень із заданим рівнем якості в *складених мережах* з довільною топологією і різними технологіями. Інші три верхні рівні вирішують завдання надання прикладних сервісів на підставі наявної транспортної підсистеми.

Сеансовий рівень

Сеансовий рівень (Session layer) забезпечує управління діалогом: фіксує, яка із сторін є активною зараз, надає засоби синхронізації. Останні дозволяють вставляти контрольні точки в довгі передачі, щоб у разі відмови можна було повернутися назад до останньої контрольної точки, а не починати все спочатку. На практиці небагато програм використовують сеансовий рівень, і він рідко реалізується у вигляді окремих протоколів, хоча функції цього рівня часто об'єднують з функціями прикладного рівня і реалізують в одному протоколі.

Сеансовий рівень – це рівень управління діалогом об'єктів прикладного рівня, він вирішує такі задачі:

- встановлення способу обміну повідомленнями (дуплексний або напівдуплексний);
- синхронізація обміну повідомленнями;
- організація "контрольних точок" діалогу.

Представницький рівень

Представницький рівень (*Presentation layer*) має справу з формою представлення переданої по мережі інформації, не міняючи при цьому її змісту. За рахунок *представницького рівня* інформація, що передається прикладним рівнем однієї системи, завжди зрозуміла прикладному рівню іншої системи. За допомогою засобів даного рівня протоколи прикладних рівнів можуть подолати синтаксичні відмінності в представленні даних або ж відмінності в кодах символів, наприклад, в кодах ASCII і EBCDIC. На цьому рівні може виконуватися шифрування і дешифровка даних, завдяки яким секретність обміну даними забезпечується відразу для всіх прикладних служб. Прикладом такого протоколу є протокол Secure Socket Layer (SSL), який забезпечує секретний обмін повідомленнями для протоколів прикладного рівня стека *TCP/IP*.

На *представницькому рівні* узгоджується представлення (синтаксис) даних при взаємодії двох прикладних процесів:

- перетворення даних із зовнішнього формату у внутрішній;
- шифрування і розшифровка даних.

Прикладний рівень

Прикладний рівень (Application layer) – це набір різноманітних протоколів, за допомогою яких користувачі мережі дістають доступ до ресурсів, що розділяються, таким як файли, принтери або гіпертекстові Web-сторінки, а також організують спільну роботу, наприклад, за допомогою протоколу електронної пошти. Одиниця даних, якою оперує прикладний рівень, зазвичай називається повідомленням (message).

Прикладний рівень – набір всіх мережних сервісів, які надає система кінцевому користувачеві:

- ідентифікація, перевірка прав доступу;
- принт- і файл-сервіс, пошта, віддалений доступ.

Існує дуже багато різних служб прикладного рівня. Приведемо як приклад хоч би декілька найбільш поширених реалізацій файлових служб: NCP в операційній системі *Novell NetWare*, SMB в *Microsoft Windows NT*, NFS, FTP і TFTP, що входять в стек TCP/IP.

Мережно-залежні і мережно-незалежні рівні

Функції всіх рівнів моделі OSI можуть бути віднесені до однієї з двох груп: або до функцій, залежних від конкретної технічної реалізації мережі, або до функцій, орієнтованих на роботу із програмним забезпеченням.

Три нижні рівні – фізичний, каналний і мережний є мережно-залежні, тобто протоколи цих рівнів тісно пов'язані з технічною реалізацією мережі і комунікаційним устаткуванням, використовується. Наприклад, перехід на устаткування FDDI означає повну зміну протоколів фізичного і каналного рівнів у всіх вузлах мережі.

Три верхні рівні – прикладний, представницький і сеансовий орієнтовані на програмне забезпечення і мало залежать від технічних особливостей побудови мережі. На протоколи цих рівнів не впливають які б то не було зміни в топології мережі, заміна устаткування або перехід на іншу мережну технологію. Так, перехід від *Ethernet* до високошвидкісної технології *100VG-AnyLAN* не зажадає ніяких змін в програмних засобах, що реалізують функції прикладного, представницького і сеансового рівнів.

Транспортний рівень є проміжним, він приховує всі деталі функціонування нижніх рівнів від верхніх. Це дозволяє розробляти програми, незалежні від технічних засобів безпосереднього транспортування повідомлень.

На рис. 1.14 показані рівні моделі OSI, на яких працюють різні елементи мережі. Комп'ютер зі встановленою на ній мережною ОС взаємодіє з іншим комп'ютером за допомогою протоколів всіх семи рівнів. Цю взаємодію комп'ютери здійснюють опосередковано, через різні комунікаційні пристрої: концентратори, модеми, мости, комутатори, маршрутизатори, мультиплексори.

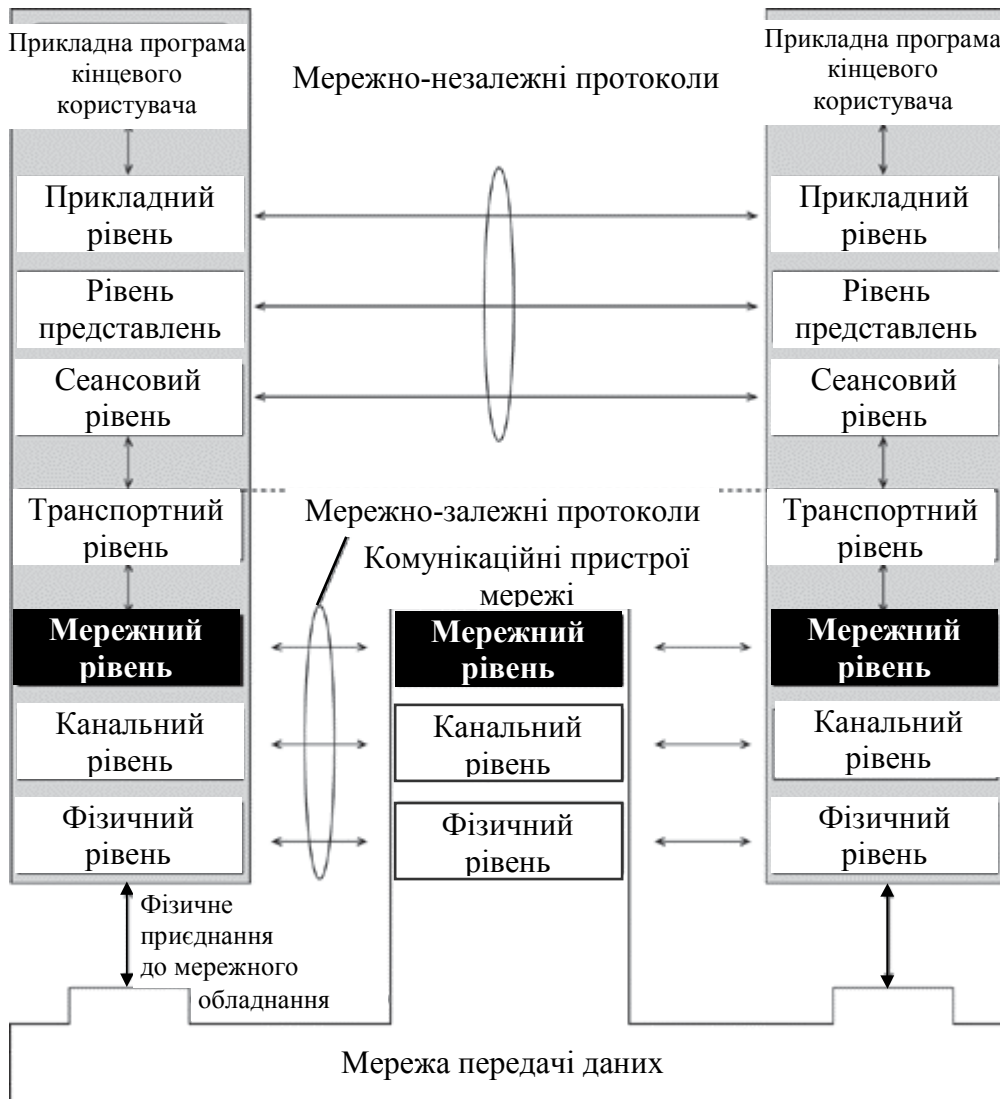


Рис. 1.14. Мережно-залежні і мережно-незалежні рівні моделі OSI.

Залежно від типу комунікаційний пристрій може працювати або тільки на фізичному рівні (повторювач), або на фізичному і каналному (міст), або на фізичному, каналному і мережному, іноді захоплюючи і транспортний рівень (маршрутизатор). На рис. 1.15 показана відповідність функцій різних комунікаційних пристроїв рівням моделі OSI.

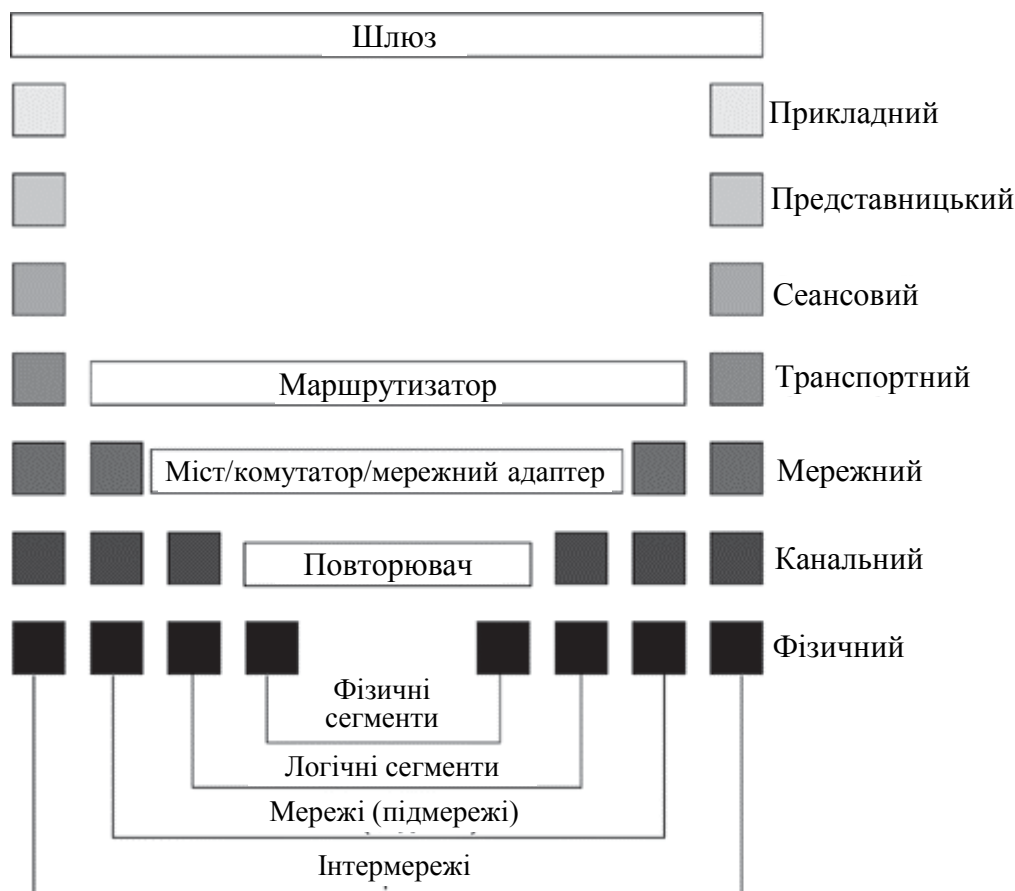


Рис. 1.15. Відповідність функцій різних пристроїв мережі рівням моделі OSI.

Модель OSI представляє хоч і дуже важливу, але тільки одну з багатьох моделей комунікацій. Ці моделі і пов'язані з ними стеки протоколів можуть відрізнятися кількістю рівнів, їх функціями, форматами повідомлень, службами, підтримуваними на верхніх рівнях, і іншими параметрами.

1.2.10. Стандартизація мереж. Поняття "Відкрита система"

Модель OSI, як випливає з її назви (Open System Interconnection), описує взаємозв'язки відкритих систем. У широкому значенні відкритою системою може бути названа будь-яка система (комп'ютер, обчислювальна мережа, ОС, програмний пакет, інші апаратні і програмні продукти), побудована відповідно до відкритих специфікацій. Нагадаємо, що під терміном "специфікація" (у обчислювальній техніці) розуміють формалізований опис апаратних або програмних компонентів, способів їх функціонування, взаємодії з іншими компонентами, умов експлуатації, обмежень і особливих характеристик. Зрозуміло, що не всяка специфікація є стандартом.

Під відкритими специфікаціями розуміються опубліковані, загальнодоступні специфікації, відповідні стандартам і прийняті в результаті досягнення згоди після всестороннього обговорення всіма зацікавленими сторонами. Використання при розробці систем відкритих специфікацій дозволяє третім сторонам розробляти для цих систем різні апаратні або програмні засоби розширення і модифікації, а також створювати програмно-апаратні комплекси з продуктів різних виробників.

Для реальних систем повна відвертість є недосяжним ідеалом. Як правило, навіть в системах, званих відкритими, цьому визначенню відповідають лише деякі частини, що підтримують зовнішні інтерфейси. Наприклад, відкритість сімейства операційних систем Unix полягає, крім усього іншого, в наявності стандартизованого програмного інтерфейсу між ядром і прикладними програмами, який дозволяє легко переносити програми з середовища однієї версії Unix в середу іншої версії. Ще одним прикладом часткової відкритості є застосування в достатньо закритій операційній системі Novell NetWare відкритого інтерфейсу Open Driver Interface (ODI) для включення в систему драйверів мережних адаптерів виробництва незалежних компаній. Чим більше відкритих специфікацій використано при розробці системи, тим більше відкритою вона є.

Модель *OSI* стосується тільки одного аспекту відкритості, а саме відкритості засобів взаємодії пристроїв, зв'язаних в обчислювальну мережу. Тут під відкритою системою розуміється мережний пристрій, готовий взаємодіяти з іншими мережними пристроями з використанням стандартних правил, що визначають формат, зміст і значення повідомлень, що приймаються і відправляються. Якщо дві мережі побудовано з дотриманням принципів відкритості, то це дає наступні переваги:

- можливість побудови мережі з апаратних і програмних засобів різних виробників, що дотримуються одного і того ж стандарту;
- можливість безболісної заміни одних компонентів мережі іншими, що дозволяє мережі розвиватися з мінімальними витратами;
- можливість легкого сполучення однієї мережі з іншою;
- простота освоєння і обслуговування мережі.

Яскравим прикладом відкритої системи є мережа *Internet*. Ця мережа розвивалася в повній відповідності з вимогами, що пред'являються до відкритих систем. У розробці її стандартів брали участь тисячі фахівців-користувачів з різних університетів, наукових організацій і фірм-виробників обчислювальної апаратури і програмного забезпечення, що працюють в різних країнах. Сама назва стандартів, що визначають роботу *Internet*, Request For Comments (*RFC*, що можна перекласти як "запит на коментарі") говорить про відкритий характер стандартів, що приймаються. В результаті мережа *Internet* об'єднала в собі різноманітне устаткування і програмне забезпечення величезної кількості мереж, розкиданих по всьому світу.

1.2.11. Протокол. Інтерфейс. Стек протоколів

Багаторівневе представлення засобів мережної взаємодії має свою специфіку, пов'язану з тим, що в процесі обміну повідомленнями беруть участь дві сторони, тобто в даному випадку необхідно організувати узгоджену роботу двох "ієрархій", що працюють на різних комп'ютерах. Обидва учасники мережного обміну повинні прийняти безліч угод. Наприклад, вони повинні погоджувати рівні і форму електричних сигналів, спосіб визначення довжини повідомлень, домовитися про методи контролю достовірності і т.п. Іншими словами, угоди повинні бути прийняті для всіх рівнів, починаючи від найнижчого – рівня передачі бітів – до найвищого, такого, що реалізовує сервіс для користувачів мережі.

На рис. 1.16 показана модель взаємодії двох вузлів. З кожного боку засіб взаємодії представлений чотирма рівнями. Процедура взаємодії цих двох вузлів може бути описана у вигляді набору правил взаємодії кожної пари відповідних рівнів обох сторін, що беруть участь.

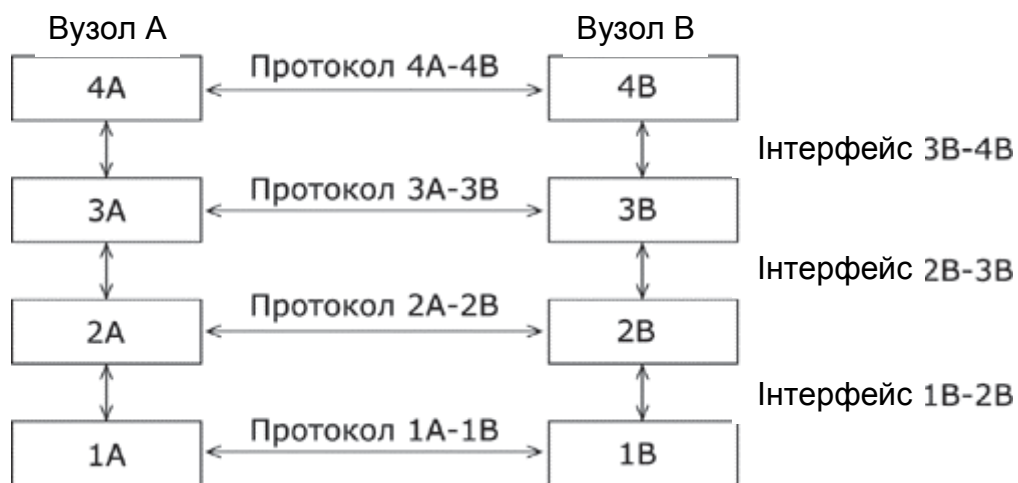


Рис. 1.16. Взаємодія двох вузлів.

Формалізовані правила, визначальна послідовність і формат повідомлень, якими обмінюються мережні компоненти одного рівня, але в різних вузлах, називаються протоколом.

Модулі, які реалізують протоколи сусідніх рівнів і знаходяться в одному вузлі, також взаємодіють один з одним відповідно до чітко певних правил за допомогою стандартизованих форматів повідомлень. Ці правила прийнято називати інтерфейсом. Інтерфейс – визначає послідовність і формат повідомлень, якими обмінюються мережні компоненти сусідніх рівнів в одному вузлі. Інтерфейс визначає набір послуг, що надається даним рівнем сусідньому рівню.

По суті, протокол і інтерфейс виражають одне і те ж поняття, але традиційно в мережах за ними закріплені різні області дії: протоколи визначають правила взаємодії модулів одного рівня в різних вузлах, а інтерфейси – модулів сусідніх рівнів в одному вузлі. Засоби кожного рівня повинні відпрацьовувати, по-перше, власний протокол, а по-друге, інтерфейси з сусідніми рівнями.

Ієрархічно організований набір протоколів, достатній для організації взаємодії вузлів в мережі, називається стеком комунікаційних протоколів. Комунікаційні протоколи можуть бути реалізовані як

програмно, так і апаратно. Протоколи нижніх рівнів часто реалізуються комбінацією програмних і апаратних засобів, а протоколи верхніх рівнів – як правило, чисто програмними засобами.

Програмний модуль, що реалізує деякий протокол, часто скорочено також називають протоколом. При цьому співвідношення між протоколом як формально певною процедурою і протоколом-програмним модулем, що реалізує цю процедуру, аналогічне співвідношенню між алгоритмом рішення деякої задачі і програмою, яка вирішує це завдання.

Зрозуміло, що один і той же алгоритм може бути запрограмований з різним ступенем ефективності. Так само і протокол може мати декілька програмних реалізацій. Саме тому при порівнянні протоколів слід враховувати не тільки логіку їх роботи, але і якість програмних рішень. Більш того, на ефективність взаємодії пристроїв в мережі впливає якість всієї сукупності протоколів, зокрема, наскільки раціонально розподілені функції між протоколами різних рівнів і наскільки добре визначені інтерфейси між ними.

Протоколи реалізуються не тільки комп'ютерами, але і іншими мережними пристроями – концентраторами, мостами, комутаторами, маршрутизаторами і т.д. Дійсно, в загальному випадку зв'язок комп'ютерів в мережі здійснюється не безпосередньо, а через різні комунікаційні пристрої. Залежно від типу пристрою в нім повинні бути вбудовані засоби, що реалізують той або інший набір протоколів.

Стандартні стеки комунікаційних протоколів

Найважливішим напрямом стандартизації в області обчислювальних мереж є стандартизація комунікаційних протоколів. В даний час в мережах використовується велика кількість стеків комунікаційних протоколів. Найбільш популярні наступні стеки:

- TCP/IP;
- IPX/SPX;
- NetBIOS/SMB;
- DECnet;
- SNA;
- OSI.

Всі ці стеки, окрім SNA на нижніх рівнях – фізичному і каналному, використовують одні і ті ж добре стандартизовані протоколи *Ethernet*, *Token Ring*, *FDDI* і ряд інших, які дозволяють задіяти у всіх мережах одну і ту ж апаратуру. Зате на верхніх рівнях всі стеки працюють по своїх протоколах. Ці протоколи часто не відповідають рекомендованому моделлю *OSI* розбиттю на рівні. Зокрема, функції сеансового і представницького рівня, як правило, об'єднані з прикладним рівнем. Така невідповідність пов'язана з тим, що модель *OSI* з'явилася як результат узагальнення тих, що вже існують і реально використовуваних стеків, а не навпаки.

Стек OSI

Слід чітко розрізнити модель *OSI* і стек *OSI*. Якщо модель *OSI* є концептуальною схемою взаємодії відкритих систем, то стек *OSI* є набором цілком конкретних специфікацій протоколів.

На відміну від інших стеків протоколів, стек *OSI* повністю відповідає моделі *OSI*, він включає специфікації протоколів для всіх семи рівнів взаємодії, визначених в цій моделі. На нижніх рівнях стек *OSI* підтримує *Ethernet*, *Token Ring*, *FDDI*, протоколи глобальних мереж, X.25 і *ISDN*, тобто використовує розроблені поза стеком протоколи нижніх рівнів, як і всі інші стеки. Протоколи мережного, транспортного і сеансового рівнів стека *OSI* специфіковані і реалізовані різними виробниками, але поширені поки мало. Найбільш популярними протоколами стека *OSI* є прикладні протоколи. До них відносяться: протокол передачі файлів FTAM, протокол емуляції терміналу VTP, протоколи довідкової служби X.500, електронної пошти X.400 і ряд інших.

Протоколи стека *OSI* відрізняються складністю і неоднозначністю специфікацій. Ці властивості стали результатом загальної політики розробників стека, що прагнули врахувати в своїх протоколах всі випадки і всі існуючі технології. До цього потрібно ще додати і наслідки великої кількості політичних компромісів, неминучих при ухваленні міжнародних стандартів з такого злободенного питання, як побудова відкритих обчислювальних мереж.

Через свою складність протоколи *OSI* вимагають великих витрат обчислювальної потужності центрального процесора, що робить їх найбільш відповідними для могутніх машин, а не для мереж персональних комп'ютерів.

Стек *OSI* – незалежний від виробників міжнародний стандарт. Його підтримує уряд США в своїй програмі *GOSIP*, відповідно до якої всі комп'ютерні мережі, що встановлюються в урядових установах США після 1990 року, повинні або безпосередньо підтримувати стек *OSI*, або забезпечувати засоби для переходу на цей стек в майбутньому. Проте, стек *OSI* популярніший в Європі, чим в США, оскільки в Європі залишилися менше старих мереж, що працюють по власних протоколах. Більшість організацій поки тільки планують перехід до стека *OSI*, і дуже небагато приступили до створення пілотних проектів. З тих, що працюють в цьому напрямі, можна назвати Військово-морське відомство США і мережу NFSNET. Одним з найбільших виробників, підтримуючих *OSI*, є компанія AT&T, її мережа Stargroup повністю базується на цьому стеку.

Стек TCP/IP

Стек *TCP/IP* був розроблений за ініціативою Міністерства оборони США більше 20 років тому назад для зв'язку експериментальної мережі ARPAnet з іншими мережами як набір загальних протоколів для різноманітного обчислювального середовища. Великий внесок у розвиток стека *TCP/IP*, який отримав свою назву від популярних протоколів *IP* і *TCP*, внесли фахівці з університету Берклі, які реалізували протоколи стека у версії ОС UNIX. Популярність цієї операційної системи привела до широкого розповсюдження протоколів *TCP*, *IP* і інших протоколів стека. Сьогодні цей стек використовується для зв'язку комп'ютерів усесвітньої інформаційної мережі *Internet*, а також у величезній кількості корпоративних мереж.

Стек *TCP/IP* на нижньому рівні підтримує всі популярні стандарти фізичного і каналного рівнів: для локальних мереж – це *Ethernet*, *Token Ring*, *FDDI*, для глобальних – протоколи роботи на аналогових комутованих і виділених лініях *SLIP*, *PPP*, протоколи територіальних мереж *X.25* і *ISDN*.

Основними протоколами стека, що дали йому назву, є протоколи *IP* і *TCP*. Ці протоколи в термінології моделі *OSI* відносяться до мережного і транспортного рівнів, відповідно. *IP* забезпечує просування пакету по складеній мережі, а *TCP* гарантує надійність його доставки.

За довгі роки використання в мережах різних країн і організацій стек *TCP/IP* увібрав в себе велику кількість протоколів прикладного рівня. До них відносяться такі популярні протоколи, як протокол пересилки файлів *FTP*, протокол емуляції терміналу *telnet*, поштовий протокол *SMTP*, який використовується в електронній пошті мережі *Internet*, гіпертекстові сервіси служби *WWW* і багато інших. Сьогодні стек *TCP/IP* є одним з найпоширеніших стеків транспортних протоколів обчислювальних мереж. Дійсно, тільки в мережі *Internet* об'єднано близько 10 мільйонів комп'ютерів по всьому світу, які взаємодіють один з одним за допомогою стека протоколів *TCP/IP*.

Стрімке зростання популярності *Internet* привело і до змін в розстановці сил в світі комунікаційних протоколів – протоколи *TCP/IP*, на яких побудований *Internet*, стали швидко тіснити безперечного лідера минулих років – стек *IPX/SPX* компанії *Novell*. Сьогодні в світі загальна кількість комп'ютерів, на яких встановлений стек *TCP/IP*, перевищила кількість комп'ютерів, на яких працює стек *IPX/SPX*, і це говорить про зміну відношення адміністраторів локальних мереж до протоколів, які використовуються на настільних комп'ютерах, оскільки саме на них раніше майже скрізь працювали протоколи компанії *Novell*, необхідні для доступу до файлових серверів *NetWare*. Процес просування стека *TCP/IP* на лідируючі позиції в будь-яких типах мереж продовжується, і зараз в комплекті постачання будь-якої промислової операційної системи обов'язково є програмна реалізація цього стека.

Хоча протоколи *TCP/IP* нерозривно пов'язані з *Internet*, і кожний з багатомільйонної армії комп'ютерів *Internet* працює на основі цього стека, існує велика кількість локальних, корпоративних і територіальних мереж, що безпосередньо не є частинами *Internet*, в яких також використовуються протоколи *TCP/IP*. Щоб відрізнити ці мережі від *Internet*, їх називають мережами *TCP/IP* або просто *IP*-мережами.

Оскільки стек *TCP/IP* спочатку створювався для глобальної мережі *Internet*, він має багато особливостей, які забезпечують йому перевагу перед іншими протоколами, коли мова заходить про побудову мереж, що включають глобальні зв'язки. Зокрема, дуже корисною властивістю, завдяки якій цей протокол може застосовуватися у великих мережах, є його здатність фрагментувати пакети. Дійсно, складена мережа часто складається з мереж, побудованих на абсолютно різних принципах. У кожній з цих мереж може бути встановлена власна величина максимальної довжини одиниці переданих даних (кадру). У такому разі при переході з однієї мережі, що має велику максимальну довжину, в іншу, з меншою максимальною довжиною, може виникнути необхідність розділення переданого кадру на декілька частин. Протокол *IP* стека *TCP/IP* ефективно вирішує цю задачу.

Іншою особливістю технології *TCP/IP* є гнучка система адресації, що дозволяє простіше в порівнянні з іншими протоколами аналогічного призначення включати в інтермережу (об'єднану або складену мережу) мережі інших технологій. Ця властивість також сприяє застосуванню стека *TCP/IP* для побудови великих гетерогенних мереж.

У стеку *TCP/IP* дуже економно використовуються можливості ширококомовних розсилок. Ця властивість просто необхідна при роботі на повільних каналах зв'язку, характерних для територіальних мереж.

Проте платою за переваги тут виявляються високі вимоги до ресурсів і складність адміністрування *IP*-мереж. Для реалізації могутніх функціональних можливостей протоколів стека *TCP/IP* потрібні великі обчислювальні витрати. Гнучка система адресації і відмова від ширококомовних розсилок приводять до наявності в *IP*-мережі різних централізованих служб типу *DNS*, *DHCP* і т.п. Кожна з цих служб спрощує адміністрування мережі і конфігурацію устаткування, але в той же час сама вимагає пильної уваги з боку адміністраторів. Можна приводити і інші доводи за і проти, проте факт залишається фактом - сьогодні *TCP/IP* найпопулярніший стек протоколів, широко використовуваний як в глобальних, так і в локальних мережах.

Стек *IPX/SPX*

Цей стек є оригінальним стеком протоколів фірми *Novell*, розробленим для мережної операційної системи *NetWare* ще в початку 80-х років. Протоколи мережного і сеансового рівнів *Internetwork Packet Exchange (IPX)* і *Sequenced Packet Exchange (SPX)*, які дали назву стеку, є прямою адаптацією протоколів *XNS* фірми *Xerox*, поширених в набагато меншому ступені, чим стек *IPX/SPX*.

Популярність стека *IPX/SPX* безпосередньо пов'язана з операційною системою *Novell NetWare*, яка довгий час зберігала світове лідерство по числу встановлених систем, хоча останнім часом її популярність набагато знизилася, і по темпах зростання вона помітно відстає від *Microsoft Windows NT*.

Багато особливостей стека *IPX/SPX* обумовлено орієнтацією ранніх версій ОС *NetWare* (до версії 4.0) на роботу в локальних мережах невеликих розмірів, що складаються з персональних комп'ютерів з скромними ресурсами. Зрозуміло, що для таких комп'ютерів компанії *Novell* потрібні були протоколи, на реалізацію яких була б потрібна мінімальна кількість оперативної пам'яті (обмеженою в *IBM*-сумісних комп'ютерах під управлінням *MSDOS* об'ємом 640 Кбайт) і які швидко працювали б на процесорах невеликої обчислювальної потужності. В результаті протоколи стека *IPX/SPX* до недавнього часу добре працювали в локальних мережах і не дуже у великих корпоративних мережах, оскільки вони дуже перенавантажували протоколами цього стека (наприклад, для встановлення зв'язку між клієнтами і серверами). Ця обставина, а також той факт, що стек *IPX/SPX* є власністю фірми *Novell*, і на його реалізацію потрібно отримувати ліцензію (тобто відкриті специфікації не підтримувалися), довгий час обмежували його поле діяльності тільки мережами *NetWare*. Проте з моменту випуску версії *NetWare 4.0* фахівці *Novell* внесли і продовжують вносити до протоколів серйозні зміни, направлені на їх адаптацію для роботи в корпоративних мережах. Зараз стек *IPX/SPX* реалізовано не тільки в *NetWare*, але і в декількох інших популярних мережних ОС, наприклад, *SCO UNIX*, *Sun Solaris*, *Microsoft Windows NT*.

Стек *NetBIOS/SMB*

Цей стек широко застосовується в продуктах компаній *IBM* і *Microsoft*. На його фізичному і каналному рівнях використовуються всі найбільш поширені протоколи *Ethernet*, *Token Ring*, *FDDI* та інші. На верхніх рівнях працюють протоколи *NETBEUI* і *SMB*.

Протокол *NETBIOS* (Network Basic Input/Output System) з'явився в 1984 році як мережне розширення стандартних функцій базової системи введення/виводу (*BIOS*) *IBM PC* для мережної програми *PC Network* компанії *IBM*. Надалі цей протокол був замінений так званим протоколом розширеного, призначеного для користувача інтерфейсу *NETBEUI* – *NETBIOS Extended User Interface*. Для забезпечення сумісності програмного забезпечення як інтерфейс до протоколу *NETBEUI* був збережений інтерфейс *NETBIOS*. Протокол *NETBEUI* розроблявся як ефективний протокол, який споживає небагато ресурсів і призначений для мереж, що налічують не більше 200 робочих станцій.

Протокол *NETBEUI* виконує багато корисних мережних функцій, які можна віднести до мережного, транспортного і сеансового рівнів моделі *OSI*, проте він не забезпечує можливість маршрутизації пакетів. Це обмежує застосування протоколу *NETBEUI* локальними мережами, не розділеними на підмережі, і робить неможливим його використання в складених мережах. Деякі обмеження *NETBEUI* знімаються в реалізації цього протоколу *NBF (NETBEUI Frame)*, яка включена в операційну систему *Microsoft Windows NT*.

Протокол *SMB* (Server Message Block) виконує функції сеансового, представницького і прикладного рівнів. На основі *SMB* реалізується файлова служба, а також служби друку і передачі повідомлень між застосуваннями. Стеки протоколів *SNA* компанії *IBM*, *DECnet* корпорації *Digital Equipment* і *AppleTalk/AFP* компанії *Apple* застосовуються в основному в операційних системах і мережному устаткуванні цих фірм.

На рис. 1.17 показана відповідність деяких найбільш популярних протоколів рівням моделі *OSI*. Часто ця відповідність вельми умовна, оскільки модель *OSI* – це тільки керівництво до дії, причому достатньо загальне, а конкретні протоколи розроблялися для вирішення специфічних завдань, причому багато з них з'явилися до розробки моделі *OSI*. В більшості випадків розробники стеків віддавали перевагу швидкості роботи мережі, втрачаючи модульність: жоден стек, окрім стека *OSI*, не розбитий на сім рівнів. Найчастіше в стеку явно виділяються 3-4 рівні: рівень мережних адаптерів, в якому реалізуються протоколи фізичного і каналного рівнів, мережний рівень, транспортний рівень і рівень служб, який об'єднує функції сеансового, представницького і прикладного рівнів.

Модель OSI	IBM/Microsoft		TCP/IP	Novell	Стек OSI
Прикладний	SMB		Telnet FTP	NCP SAP	X.200, X.201 FTAM
Представницький			SNMP WWW		Представницький протокол OSI
Сеансовий			TCP		Сеансовий протокол OSI
Транспортний			SPX	Транспортний протокол OSI	
Мережний				IPX RIP NLSP	ES-ES IS-IS
Канальний	802.3 (Ethernet), 802.5 (Token Ring), FDDI, Fast Ethernet, SLIP, 802.11 VG-AnyLAN, X.25, ATM, LAP-B, LAP-D, PPP Коаксіальний кабель, екранована та неекранована вита пара, оптоволокну, радіоканал				
Фізичний					

Рис. 1.17. Відповідність популярних стеків протоколів моделі OSI.

1.3. Основні задачі побудови мереж

При створенні обчислювальних мереж розробникам довелося вирішувати безліч самих різних завдань, пов'язаних з кодуванням і синхронізацією електричних (оптичних) сигналів, вибором конфігурації фізичних і логічних зв'язків, розробкою схем адресації пристроїв, створенням різних способів комутації, мультиплексуванням і демультиплексуванням потоків даних, сумісним використанням середовища передачі. В процесі розвитку і вдосконалення мережних технологій послідовно виникали різні завдання: від найбільш простого випадку безпосереднього з'єднання двох пристроїв фізичним каналом до адресації вузлів мережі.

1.3.1. Зв'язок комп'ютера з периферійними пристроями

Безпосереднє з'єднання двох пристроїв фізичним каналом називається зв'язком "точка-точка" (point-to-point). Окремим випадком зв'язку "точка-точка" є з'єднання комп'ютера з периферійним пристроєм. Оскільки механізми взаємодії комп'ютерів в мережі багато що запозичили зі схеми взаємодії комп'ютера з периферійними пристроями, почнемо розглядати принципи роботи мережі з цього "домережного" випадку.

Для обміну даними комп'ютер і периферійний пристрій (ПП) були оснащені зовнішніми інтерфейсами або портами (рис. 1.18). В даному випадку до поняття "інтерфейс" відносяться:

- електричний роз'єм;
- набір проводів, що сполучають пристрої;
- сукупність правил обміну інформацією по цих проводах.

З боку комп'ютера логікою передачі сигналів на зовнішній інтерфейс управляють:

- **контролер ПП** – апаратний блок, що часто реалізовується у вигляді окремої плати;
- **драйвер ПП** – програма, що управляє контролером периферійного пристрою.

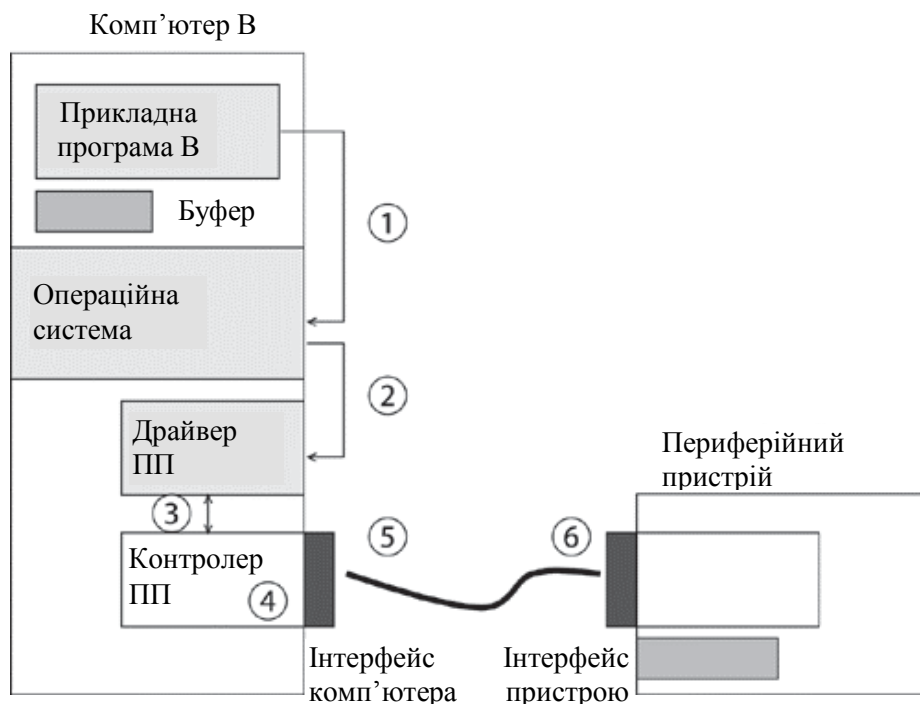


Рис. 1.18. Зв'язок комп'ютера з периферійним пристроєм.

З боку ПП інтерфейс найчастіше реалізується апаратним пристроєм управління ПП, хоча зустрічаються і програмно-керовані периферійні *пристрої*. Обмін даними між ПП і комп'ютером, як правило, є двонаправленим. Так, наприклад, навіть принтер, який є пристроєм виведення інформації, повертає в комп'ютер дані про свій стан. Таким чином, по каналу, що зв'язує зовнішні *інтерфейси*, передається наступна інформація:

- дані, що поступають від *контролера* на ПП, наприклад, байти тексту, який потрібно роздрукувати на папері;
- команди управління, які *контролер* передає на пристрій управління ПП; у відповідь на них воно виконує спеціальні дії, наприклад, переводить головку диска на відповідну доріжку або ж виштовхує з принтера лист паперу;
- дані, які повертаються пристроєм у відповідь на запит від *контролера*, наприклад, дані про готовність до виконання операції.

Розглянемо послідовність дій, які виконуються у тому випадку, коли деякій прикладній програмі потрібно надрукувати текст на принтері. З боку комп'ютера у виконанні цієї операції бере участь, окрім вже названих контролера, драйвера і прикладної програми, ще один найважливіший компонент – операційна система. Оскільки всі операції введення-виводу є привілейованими, всі прикладні програми при виконанні операцій з периферійними *пристроями* використовують ОС як арбітра. Отже, послідовність дій така:

1. Прикладна програма звертається із запитом на виконання операції друку до операційної системи. У запиті указуються: адреса даних в оперативній пам'яті, ідентифікуюча інформація принтера і операція, яку потрібно виконати.

2. Отримавши запит, операційна система аналізує його, вирішує, чи може він бути виконаний, і якщо рішення позитивне, то запускає відповідний *драйвер*, передаючи йому як параметри адресу даних, що виводяться. Подальші дії, що відносяться до операції введення-виводу, з боку комп'ютера реалізуються спільно драйвером і контролером принтера.

3. Драйвер передає команди і дані *контролеру*, який поміщає їх в свій внутрішній буфер. Хай, наприклад, *драйвер* завантажує значення деякого байта в буфер *контролера* ПП.

4. Контролер переміщає дані з внутрішнього буфера в зовнішній порт.

5. Контролер починає послідовно передавати біти в лінію зв'язку, представляючи кожен біт відповідним електричним сигналом. Щоб повідомити пристрій управління принтера про те, що починається передача байта, перед передачею першого біта даних *контролер* формує стартовий сигнал специфічної форми, а після передачі останнього інформаційного біта – стоповий сигнал. Ці сигнали синхронізують передачу байта. Окрім інформаційних біт, *контролер* може передавати біт контролю парності для підвищення достовірності обміну.

6. Пристрій управління принтера, виявивши на відповідній лінії стартовий біт, виконує підготовчі дії і починає приймати інформаційні біти, формуючи з них байт в своєму приймальному буфері. Якщо передача супроводжується бітом парності, то виконується перевірка коректності передачі: при правильно виконаній

передачі у відповідному регістрі пристрою управління принтера встановлюється ознака завершення прийому інформації. Нарешті, прийнятий байт обробляється принтером – виконується відповідна команда або друкується символ.

Обов'язки між драйвером і контроллером можуть розподілятися по-різному, але найчастіше контроллер підтримує набір простих команд, які управляють периферійним пристроєм, а на драйвер зазвичай покладаються найбільш складні функції реалізації обміну. Наприклад, контроллер принтера може підтримувати такі елементарні команди, як "Друк символу", "Переклад рядка", "Повернення каретки" і т.п.

Драйвер же принтера за допомогою цих команд реалізує друк рядків символів, розділення документа на сторінки і інші більш високорівневі операції (наприклад, підрахунок контрольної суми послідовності переданих байтів, аналіз стану периферійного пристрою, перевірка правильності виконання команди). Драйвер, задаючи ту або іншу послідовність команд, визначає тим самим логіку роботи периферійного пристрою. Для одного і того ж контроллера можна розробити різні драйвери, які за допомогою одного і того ж набору доступних команд реалізуюватимуть різні алгоритми управління одним і тим же ПП.

Між драйвером і контроллером можливе розподілення функцій. Функції, що виконуються драйвером:

- ведення черг запитів;
- буферизація даних;
- підрахунок контрольної суми послідовності байтів;
- аналіз стану ПП;
- завантаження чергового байта даних (або команди) в регістр *контролера*;
- читання байта даних або байта стану ПП з регістра *контролера*.

Функції, що виконуються *контролером*:

- перетворення байта з регістра (*порту*) в послідовність біт;
- передача кожного біта в лінію зв'язку;
- обрамлення байта стартовим і стоповим бітами – синхронізація;
- формування біта парності;
- установка ознаки завершення прийому/передачі байта.

1.3.2. Зв'язок двох комп'ютерів

Припустимо, що користувач іншого комп'ютера хотів би роздрукувати текст. Складність полягає в тому, що до його комп'ютера не приєднаний принтер, і потрібно скористатися тим принтером, який пов'язаний з іншим комп'ютером (рис. 1.19).

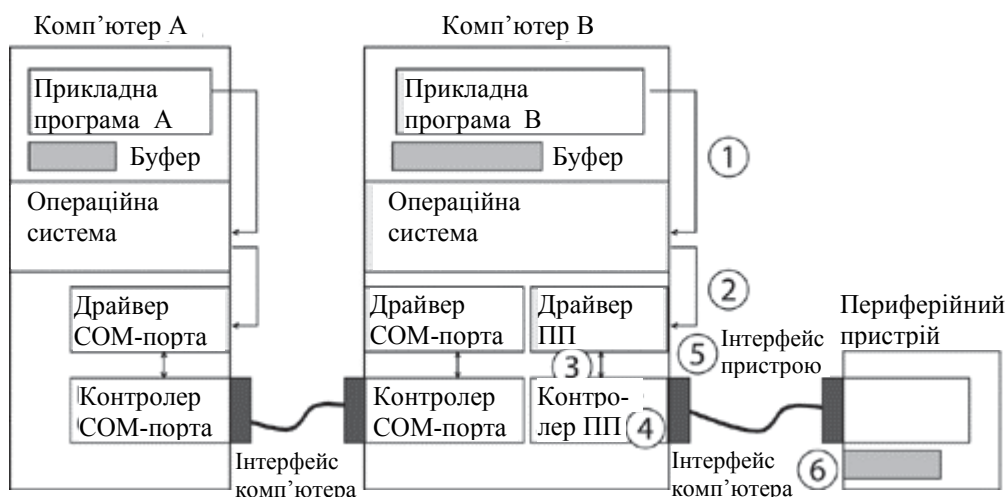


Рис. 1.19. Взаємодія двох комп'ютерів.

Програма, що працює на одному комп'ютері, не може дістати безпосередній доступ до ресурсів іншого комп'ютера – його дисків, файлів, принтера. Вона може тільки "попросити" про це іншу програму, що виконується на тому комп'ютері, якому належать ці ресурси. Ці "прохання" виражаються у вигляді повідомлень, що передаються по каналах зв'язку між комп'ютерами. Така організація друку називається віддаленою.

Припустимо, що ми зв'язали комп'ютери по кабелю через СОМ-порти, які, як відомо, реалізують *інтерфейс RS-232C* (таке з'єднання часто називають нуль-модемним). Зв'язок між комп'ютерами здійснюється аналогічно зв'язку комп'ютера з ПП. Тільки тепер *контролери* і *драйвери* портів діють з двох боків. Разом вони забезпечують передачу по кабелю між комп'ютерами одного байта інформації. (У

"справжніх" локальних мережах подібні функції передачі даних в лінію зв'язку виконуються мережними адаптерами і їх *драйверами*.)

Отже, механізм обміну байтами між двома комп'ютерами визначений. Тепер потрібно домовитися про правила обміну повідомленнями між прикладними програмами А і В. Прикладна програма В повинна "уміти" розшифрувати отриману від прикладної програми А інформацію. Для цього програмісти, що розробляли прикладні програми А і В, строго обумовлюють формати повідомлень, якими обмінюватимуться застосування, і їх семантику. Наприклад, вони можуть домовитися про те, що будь-яке виконання віддаленої операції друку починається з передачі повідомлення, яке запрошує інформацію про готовність застосування В; що на початку повідомлення йде число, яке визначає довжину даних, призначених для друку; що ознакою термінового завершення друку є певна кодова комбінація і т.п. Тим самим, як буде показано далі, визначається протокол взаємодії застосувань.

Повернемося до послідовності дій, які необхідно виконати для роздрукування тексту на принтері "чужого" комп'ютера:

- прикладна програма А формує чергове повідомлення (що містить, наприклад, рядок, який необхідно вивести на принтер) прикладної програми В, поміщає його в буфер оперативної пам'яті і звертається до ОС із запитом на передачу вмісту буфера на комп'ютер В;

- ОС комп'ютера А звертається до драйвера СОМ-порту, який ініціює роботу *контролера*;

- пари *драйверів* і *контролерів* СОМ-порту, що діють з обох боків, послідовно, байт за байтом, передають повідомлення на комп'ютер В;

- драйвер комп'ютера В періодично виконує перевірку на наявність ознаки завершення прийому, що встановлюється *контролером* при правильній виконаній передачі даних, і при його появі прочитає прийнятий байт з буфера *контролера* в оперативну пам'ять, тим самим роблячи його доступним для програм комп'ютера В; в деяких випадках *драйвер* викликається асинхронно, по перериваннях від *контролера*; аналогічно реалізується і передача байта в інший бік – від комп'ютера В до комп'ютера А;

- прикладна програма комп'ютера В приймає повідомлення, інтерпретує його, і залежно від того, що в ньому міститься, формує запит до своєї ОС на виконання тих або інших дій з принтером; у нашому прикладі повідомлення містить вказівку на друк тексту, тому ОС передає драйверу принтера запит на друк рядка;

- далі виконуються всі дії 1-6, що описують виконання запиту прикладної програми до ПП відповідно до розглянутої раніше схеми "локальна ОС – *драйвер* ПП – *контролер* ПП – пристрій управління ПП" (див. попередній розділ); в результаті рядок буде надрукований.

Розглянуто послідовність роботи системи при передачі тільки одного повідомлення від прикладної програми комп'ютера А до прикладної програми комп'ютера В. Однак порядок взаємодії цих двох прикладних програм може припускати неодноразовий обмін повідомленнями різного типу. Наприклад, після успішного друку рядка (у попередньому прикладі) згідно правилам, прикладна програма комп'ютера В повинна послати повідомлення-підтвердження. У відповідь на це повідомлення прикладної програми комп'ютера В поміщає в буферну область оперативної пам'яті, а далі за допомогою *драйвера* СОМ-порту передає його по каналу зв'язку в комп'ютер А, де воно і потрапляє до прикладної програми комп'ютера А.

1.3.3. Клієнт, редиректор і сервер

Можна уявити, що будь-яка програма, якій буде потрібно друк на "чужому" принтері, повинна включати функції, подібні тим, які виконує прикладна програма комп'ютера А. Навантажувати цими стандартними діями кожне застосування – текстові і графічні редактори, системи управління базами даних і інші застосування, – не дуже раціонально (хоча існує велика кількість програм, які дійсно самостійно вирішують всі задачі по обміну даними між комп'ютерами, наприклад, *Kermit* – програма обміну файлами через СОМ-порти, реалізована для різних ОС, *Norton Commander* з його функцією *Link*). Набагато вигідніше створити спеціальний програмний модуль, який (замість прикладної програми комп'ютера А) виконуватиме формування повідомлень-запитів до віддаленої машини і прийом результатів для всіх прикладних програм. Такий службовий модуль називається *клієнтом*.

На стороні ж комп'ютера В (на місці прикладної програми комп'ютера В) повинні працювати інша спеціалізована програма – *сервер*, яка постійно очікує приходу запитів на віддалений доступ до принтера (або до файлів, розташованих на диску) цього комп'ютера. Сервер, прийнявши запит з мережі, звертається до локального ПП, можливо, за участю локальної ОС.

Дуже зручною і корисною функцією клієнтської програми є здатність відрізнити запит до віддаленого файлу від запиту до локального файлу. Якщо клієнтська програма уміє це робити, вона сама розпізнає і перенаправляє (*redirect*) запит до віддаленої машини. Звідси і назва, часто використовувана для клієнтської частини, – *редиректор*. Іноді функції розпізнавання виділяються в особливий програмний модуль, в цьому випадку редиректором називають не всю клієнтську частину, а тільки цей модуль.

Програмний клієнт і сервер виконують системні функції по обслуговуванню запитів всіх програм комп'ютера А на віддалений доступ до файлів комп'ютера В. Щоб прикладні програми комп'ютера В могли користуватися файлами комп'ютера А, описану схему потрібно симетрично доповнити клієнтом для комп'ютера В і сервером для комп'ютера А. Схема взаємодії клієнта і сервера із прикладними програмами і локальною операційною системою приведена на рис. 1.20.

Взаємодія програмних компонентів

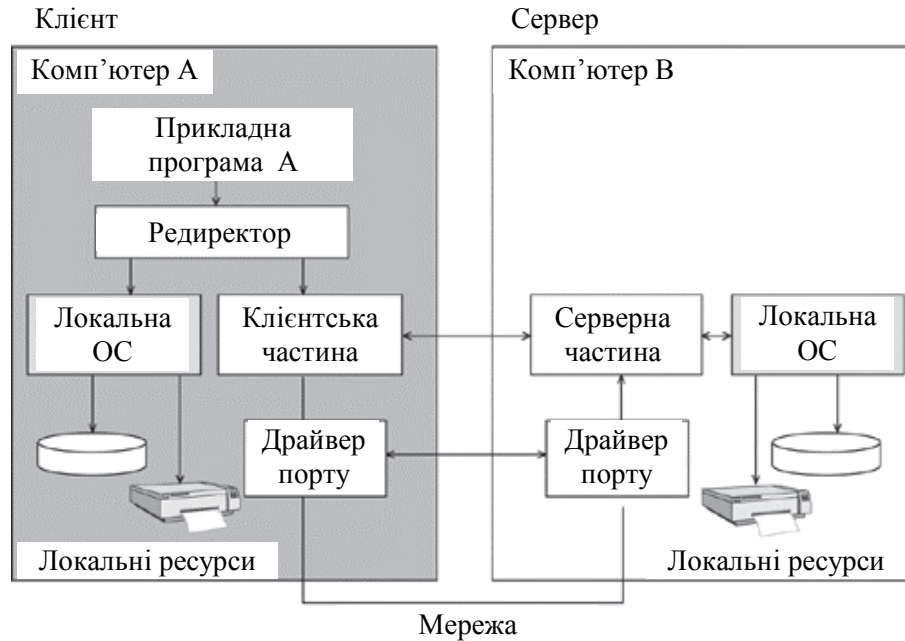


Рис. 1.20. Взаємодія програмних компонентів при зв'язку двох комп'ютерів.

Для того, щоб комп'ютер міг працювати в мережі, його операційна система повинна бути доповнена клієнтським і/або серверним модулем, а також засобами передачі даних між комп'ютерами. В результаті такого додавання операційна система комп'ютера стає *мережною ОС*.

1.3.4. Фізична передача даних по лініях зв'язку

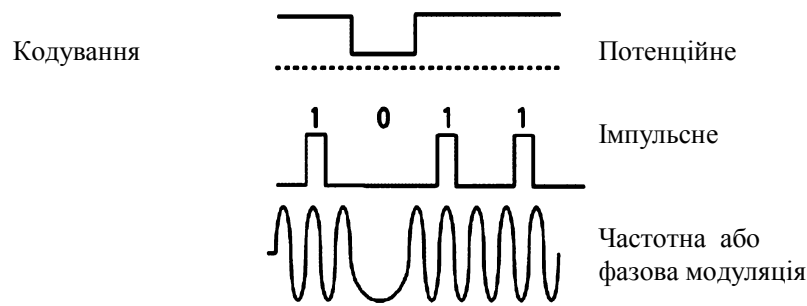
Навіть при розгляді простої мережі, яка складається всього з двох машин, можна побачити багато проблем, властивих будь-якій обчислювальній мережі, зокрема, пов'язаних з фізичною передачею сигналів по лініях зв'язку. При з'єднанні "точка-точка" на перший план виходить завдання фізичної передачі даних по лініях зв'язку. Це завдання серед іншого включає:

- кодування і модуляцію даних;
- взаємну синхронізацію передавача одного комп'ютера з приймачем іншого;
- підрахунок контрольної суми і передачу її по лініях зв'язку після кожного байта або після деякого блоку байтів.

В обчислювальній техніці для представлення даних використовується двійковий код. Представлення даних у вигляді електричних або оптичних сигналів називається кодуванням. Існують різні способи кодування двійкових цифр 1 і 0, наприклад, потенційний спосіб, при якому одиниці відповідає один рівень напруги, а нулю – інший, або імпульсний спосіб, коли для представлення цифр використовуються імпульси різної або однієї полярності.

Аналогічні підходи можуть використовуватися для кодування даних і при їх передачі між двома комп'ютерами по лініях зв'язку. Проте ці лінії зв'язку відрізняються по своїх електричних характеристиках від тих, які існують усередині комп'ютера. Головна відмінність зовнішніх ліній зв'язку від внутрішніх полягає в їх набагато більшій протяжності, а також в тому, що вони проходять поза екранованим корпусом по просторах, часто схильних до дії сильних електромагнітних перешкод. Все це приводить до істотно великих спотворень прямокутних імпульсів (наприклад, "заваленню" фронтів). Тому при передачі даних всередині і поза комп'ютером не завжди можна використовувати одні і ті ж швидкості і способи кодування.

В обчислювальних мережах застосовують як потенційне, так і імпульсне кодування дискретних даних, а також специфічний спосіб представлення даних, який ніколи не використовується усередині комп'ютера, – модуляцію (рис. 1.21). При модуляції дискретна інформація представляється гармонійним сигналом тієї частоти, яку добре передає наявна лінія зв'язку. Для технічної реалізації передачі даних по фізичним каналам використовують мережні адаптери, мережні інтерфейси комутаторів, маршрутизаторів і ін. Для перетворення комп'ютерних даних в аналогову форму використовують модеми (модулятори-демодулятори).



- Компресія – усунення надлишковості інформації
- Перетворення інформації з паралельної в послідовну форму (економія ресурсу каналів зв'язку)
- Забезпечення надійності передачі – перевірка контрольних сум, квітирування

Рис. 1.21. Задачі фізичної передачі даних.

Потенційне або імпульсне кодування застосовується на каналах високої якості, а модуляція на основі синусоїдальних сигналів переважно у тому випадку, коли канал вносить сильні спотворення до переданих сигналів. Зазвичай модуляція використовується в глобальних мережах при передачі даних через аналогові телефонні лінії, які були розроблені для передачі голосу в аналоговій формі і тому не дуже підходять для безпосередньої передачі імпульсів.

На спосіб передачі сигналів впливає і кількість проводів в лініях зв'язку між комп'ютерами. Щоб знизити вартість ліній зв'язку в мережах, розробники прагнуть скоротити кількість проводів і через це використовують не паралельну передачу всіх біт одного байта або навіть декілька байт, як це робиться усередині комп'ютера, а послідовну, побітову передачу, що вимагає всього однієї пари проводів.

При передачі сигналів доводиться ще вирішувати проблему взаємної синхронізації передавача одного комп'ютера з приймачем іншого. При організації взаємодії модулів усередині комп'ютера вона вирішується дуже просто, оскільки в цьому випадку всі модулі синхронізуються від загального тактового генератора. Проблема синхронізації при зв'язку комп'ютерів може вирішуватися різними способами, як за допомогою обміну спеціальними тактовими синхроімпульсами по окремій лінії, так і за допомогою періодичної синхронізації наперед обумовленими кодами або імпульсами характерної форми, відмінної від форми імпульсів даних.

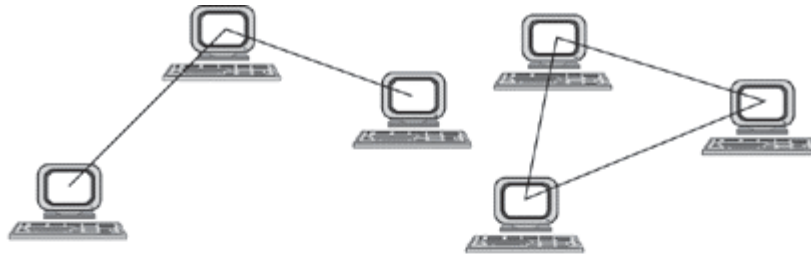
Не дивлячись на прийняті заходи (вибір відповідної швидкості обміну даними, ліній зв'язку з певними характеристиками, способу синхронізації приймача і передавача), існує вірогідність спотворення деяких біт переданих даних. Для надійнішої передачі даних часто використовується стандартний прийом – підрахунок контрольної суми і передача її по лініях зв'язку після кожного байта або після деякого блоку байтів. Часто в протокол обміну даними включається як обов'язковий елемент сигнал-квитанція, яка підтверджує правильність прийому даних і посилається від одержувача відправникові.

У кожен мережний *інтерфейс*, будь то *порт* маршрутизатора, концентратора або комутатора, вбудовані засоби, в тій чи іншій мірі завдання надійного обміну двійковими сигналами, представленими відповідними електромагнітними сигналами. Деякі мережні пристрої, такі як модеми і мережні адаптери, спеціалізуються на фізичній передачі даних. Модеми виконують в глобальних мережах модуляцію і демодуляцію дискретних сигналів, синхронізують передачу електромагнітних сигналів по лініях зв'язку, перевіряють правильність передачі по контрольній сумі і можуть виконувати деякі інші операції. Мережні адаптери розраховані, як правило, на роботу з певним середовищем, що передає, – коаксіальним кабелем, вітою парою, оптоволоконном і т.п. Кожен тип середовища передачі володіє певними електричними характеристиками, які впливають на спосіб використання даного середовища, і визначають швидкість передачі сигналів, спосіб їх кодування і деякі інші параметри. До цих пір розглядалася мережа, яка складається всього з двох машин. При об'єднанні в мережу більшої кількості комп'ютерів виникає цілий комплекс нових проблем.

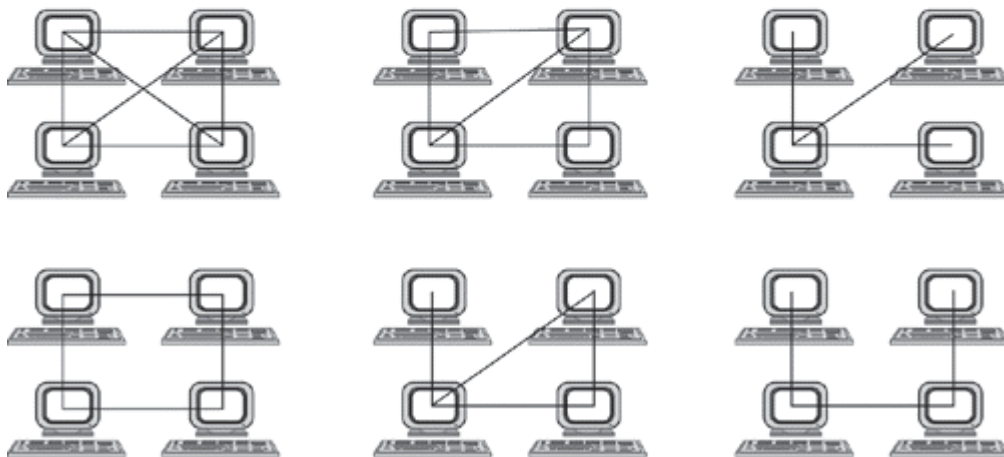
1.3.5. Задачі зв'язку декількох комп'ютерів. Топологія фізичних зв'язків

Як тільки комп'ютерів стає більше двох, виникає проблема вибору конфігурації фізичних зв'язків або топології. Під топологією мережі розуміється конфігурація графа, вершинам якого відповідають кінцеві вузли мережі (наприклад, комп'ютери) і комунікаційне устаткування (наприклад, маршрутизатори), а ребрам – електричні і інформаційні зв'язки між ними.

Число можливих конфігурацій різко зростає при збільшенні числа пов'язаних пристроїв. Так, якщо три комп'ютери ми можемо зв'язати двома способами, то для чотирьох комп'ютерів (рис. 1.22) можна запропонувати вже шість топологічно різних конфігурацій (за умови аналогічної конфігурації комп'ютерів).



а) варіанти зв'язку двох комп'ютерів



б) варіанти зв'язку чотирьох комп'ютерів

Рис. 1.22. Варіанти зв'язку комп'ютерів.

Можливо сполучати кожен комп'ютер з кожним або ж зв'язувати їх послідовно, припускаючи, що вони спілкуватимуться, передаючи один одному повідомлення "транзитом". При цьому транзитні вузли повинні бути оснащені спеціальними засобами, які дозволяють виконувати цю специфічну посередницьку операцію. В ролі транзитного вузла може виступати як універсальний комп'ютер, так і спеціалізований пристрій.

Від вибору топології зв'язків залежить багато характеристик мережі. Наприклад, наявність між вузлами декількох шляхів підвищує надійність мережі і робить можливим балансування завантаження окремих каналів. Простота приєднання нових вузлів, властива деяким топологіям, робить мережу легко розширюваною. Економічні міркування часто приводять до вибору топологій, для яких характерна мінімальна сумарна довжина ліній зв'язку.

Серед безлічі можливих конфігурацій (рис. 1.23) розрізняють повнозв'язні і неповнозв'язні:

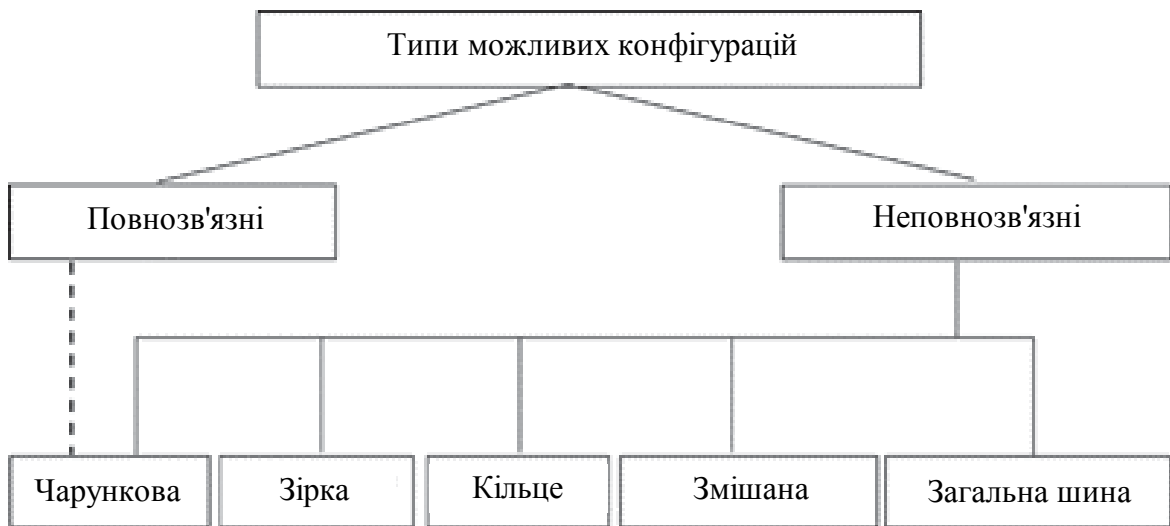


Рис. 1.23. Типи конфігурацій.

Повнозв'язна топологія (fully connected topology) (рис. 1.24) відповідає мережі, в якій кожен комп'ютер безпосередньо пов'язаний зі всіма іншими. Не дивлячись на логічну простоту, це варіант громіздкий і неефективний. Дійсно, кожен комп'ютер в мережі повинен мати велику кількість комунікаційних портів, достатню для зв'язку з кожним з решти комп'ютерів. Для кожної пари комп'ютерів повинна бути виділена окрема фізична лінія зв'язку. (В деяких випадках навіть дві, якщо неможливе використання цієї лінії для двосторонньої передачі.) Повнозв'язні топології в великих мережах застосовуються рідко, оскільки для зв'язку N вузлів потрібно $N(N-1)/2$ фізичних дуплексних ліній зв'язку, тобто має місце квадратична залежність. Частіше цей вид топології використовується в багатомашинних комплексах або в мережах, які об'єднують невелику кількість комп'ютерів.

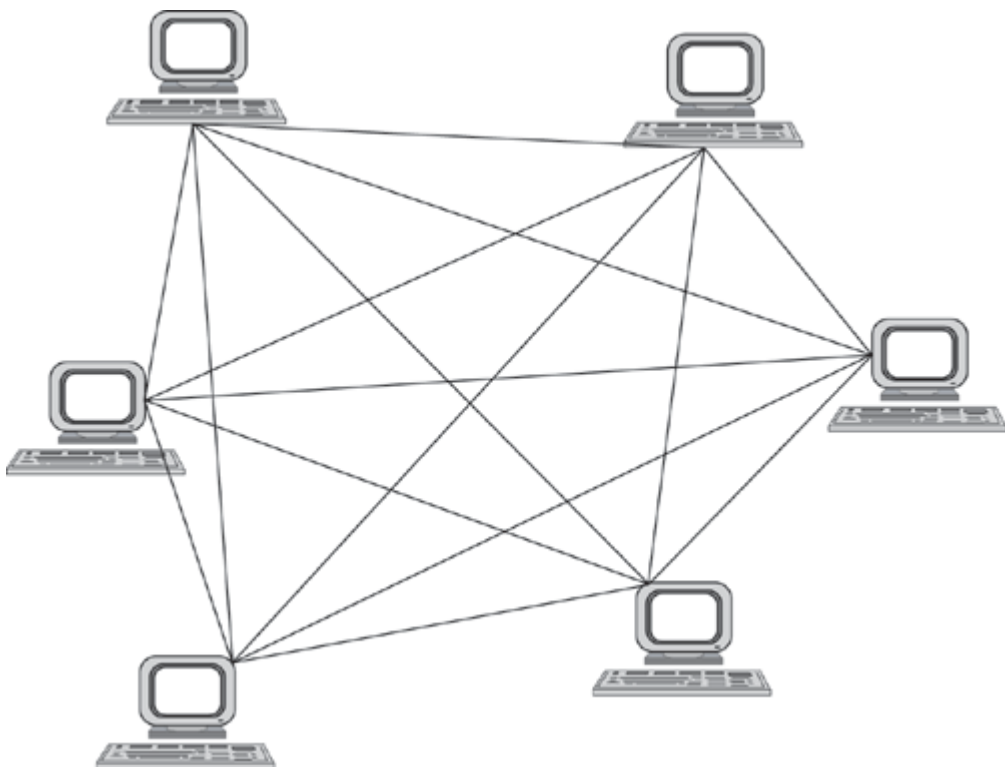


Рис. 1.24. Повнозв'язна конфігурація.

Всі інші варіанти засновані на неповнозв'язних топологіях, коли для обміну даними між двома комп'ютерами може бути потрібна проміжна передача даних через інші вузли мережі.

Чарункова топологія (mesh) походить з повнозв'язної шляхом віддалення деяких можливих зв'язків. Чарункова топологія допускає з'єднання великої кількості комп'ютерів і характерна для великих мереж (рис 1.25).

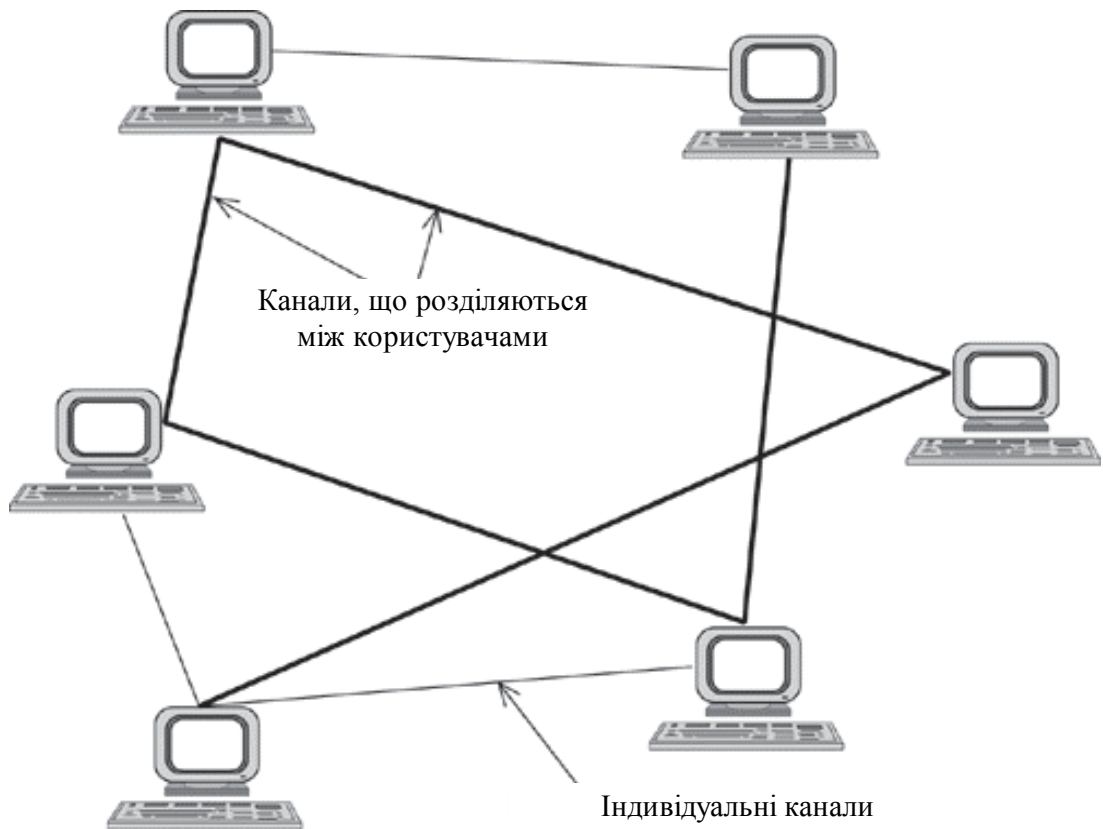


Рис. 1.25. Чарункова топологія.

У мережах з кільцевою конфігурацією (рис. 1.26) дані передаються по кільцю від одного комп'ютера до іншого. Головне достоїнство "кільця" в тому, що воно за своєю природою володіє властивістю резервування зв'язків. Дійсно, будь-яка пара вузлів сполучена тут двома шляхами – за годинниковою стрілкою і проти. "Кільце" є дуже зручною конфігурацією і для організації зворотного зв'язку – дані, зробивши повний оберт, повертаються до вузла-джерела. Тому відправник в даному випадку може контролювати процес доставки даних адресатові. Часто ця властивість "кільця" використовується для тестування зв'язності мережі і пошуку вузла, що працює некоректно. В той же час в мережах з кільцевою топологією необхідно приймати спеціальні заходи, щоб у разі виходу з ладу або відключення якої-небудь станції не уривався канал зв'язку між рештою станцій "кільця".

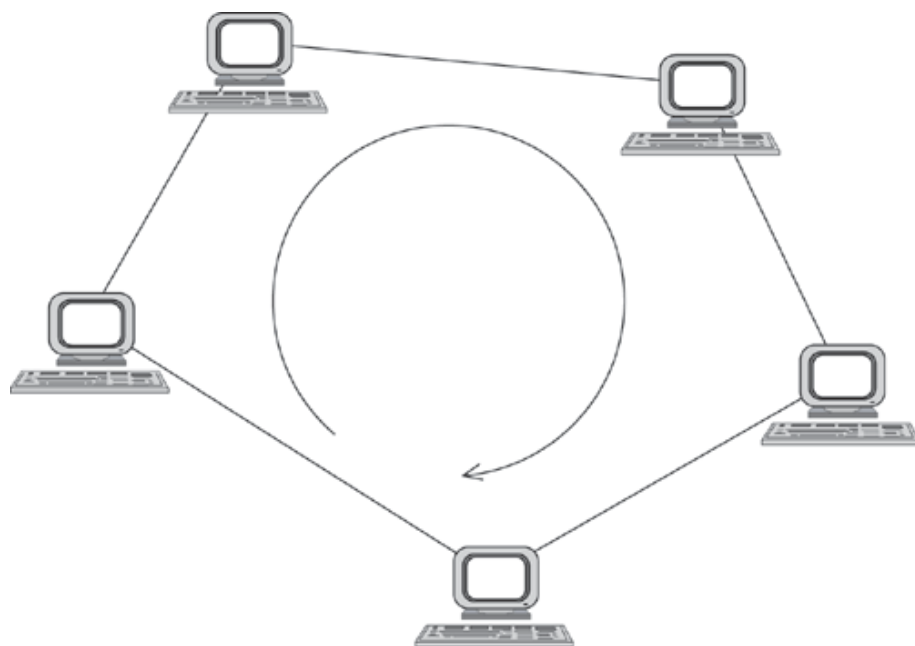


Рис. 1.26. Топологія "кільце".

Топологія "зірка" (рис. 1.27) утворюється у тому випадку, коли кожен комп'ютер за допомогою окремого кабелю підключається до центрального пристрою, який називається вузлом комутації. У функції вузла комутації входить направлення переданої комп'ютером інформації одному або решті всіх комп'ютерів мережі. В ролі вузла комутації може виступати як комп'ютер, так і спеціалізований пристрій, такий як концентратор, багатовходовий повторювач, комутатор або маршрутизатор. До недоліків топології типу "зірка" відноситься вища вартість мережного устаткування, пов'язана з необхідністю придбання спеціалізованого центрального пристрою. Крім того, можливості нарощування кількості вузлів в мережі обмежуються кількістю портів концентратора.

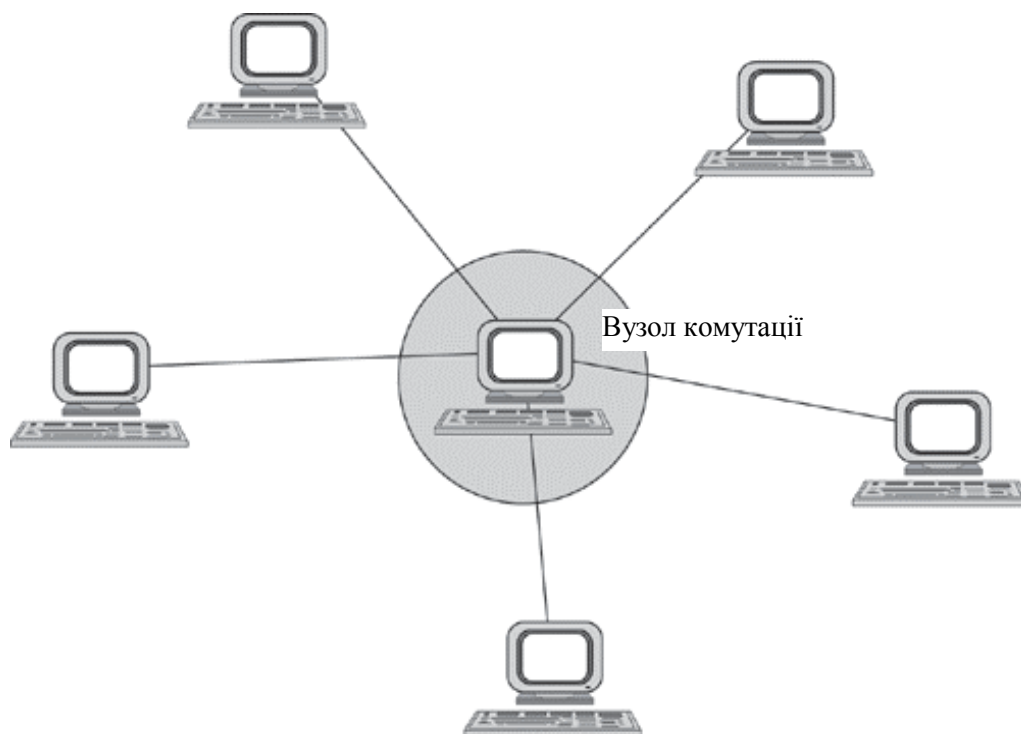


Рис. 1.27. Топологія "зірка".

Іноді має сенс будувати мережу з використанням декількох концентраторів, ієрархічно сполучених між собою зв'язками типу "зірка" (рис. 1.28). Отриману в результаті структуру називають також деревом. В даний час дерево є найпоширенішим типом топології зв'язків, як в локальних, так і в глобальних мережах.

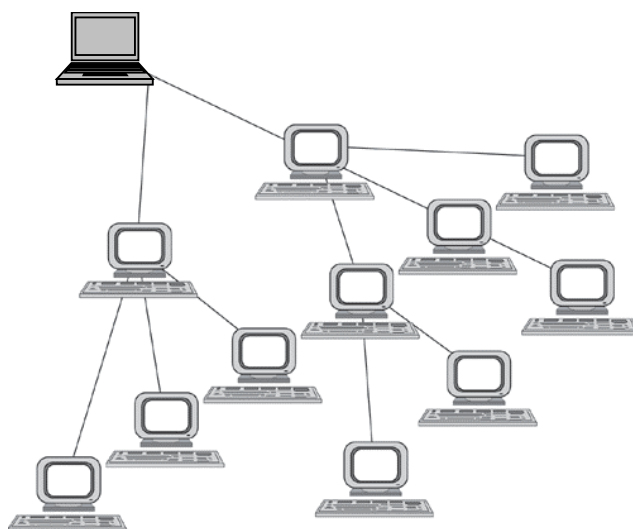


Рис. 1.28. Топологія "ієрархічна зірка" або "дерево".

Особливим окремим випадком конфігурації зірка є конфігурація "загальна шина" (рис. 1.29). Тут в ролі центрального елемента виступає пасивний кабель, до якого по схемі так званого "монтажного АБО" через Т-

конектори підключається декілька комп'ютерів (таку ж топологію мають багато мереж, що використовують бездротовий зв'язок, роль загальної шини грає загальне радіосередовище). Передана інформація розповсюджується по кабелю і доступна одночасно всім приєднаним до нього комп'ютерам. Самі комп'ютери приєднуються до мережі через мережні адаптери (*N/A* – *network adaptors*). Основними перевагами такої схеми є низька вартість і простота нарощування, тобто приєднання нових вузлів до мережі.

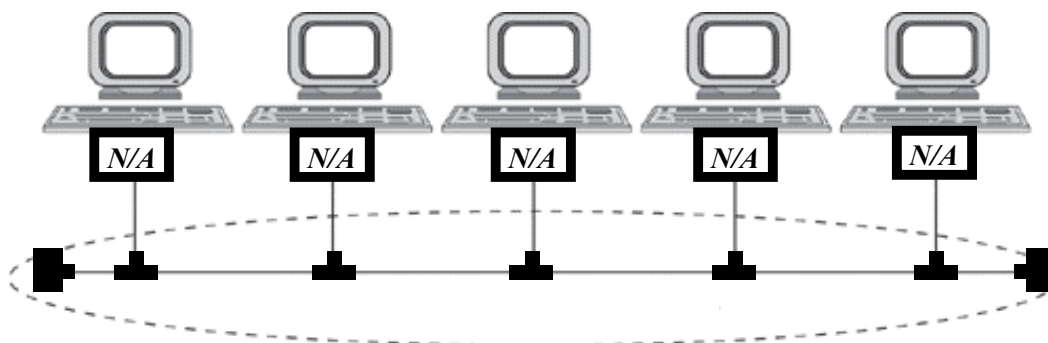


Рис. 1.29 . Топологія "загальна шина".

Недоліком "загальної шини" є її недостатня надійність: будь-який дефект кабелю або якого-небудь з численних роз'ємів повністю паралізує всю мережу. Інший недолік "загальної шини" – невисока продуктивність, оскільки при такому способі підключення в кожен момент часу тільки один комп'ютер може передавати дані по мережі, тому пропускна спроможність каналу зв'язку завжди ділиться між всіма вузлами мережі. Але завдяки простоті розсортування, нарощування та експлуатації "загальна шина" є однією з найпопулярніших топологій локальних мереж.

Тоді як невеликі мережі, як правило, мають типову топологію – "зірка", "кілеце" або "загальна шина", для крупних мереж характерна наявність довільних зв'язків між комп'ютерами. У таких мережах можна виділити окремі довільно зв'язані фрагменти (підмережі), що мають типову топологію, тому їх називають мережами із змішаною топологією (рис. 1.30).

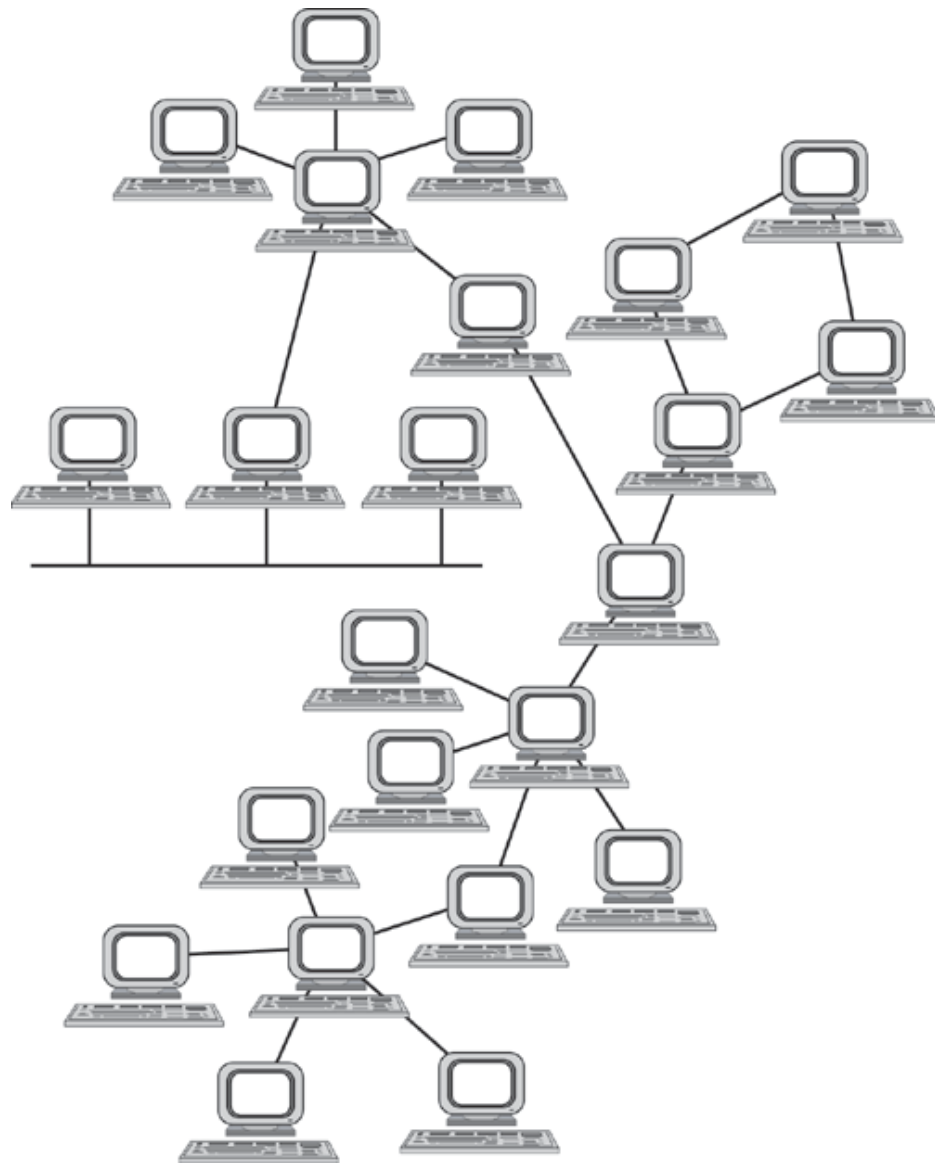


Рис. 1.30. Приклад змішаної топології.

1.3.6. Адресація вузлів мережі

Ще однією проблемою, яку потрібно враховувати при об'єднанні трьох і більш комп'ютерів, є проблема їх адресації, тобто адресації їх мережних інтерфейсів. Один комп'ютер може мати декілька мережних інтерфейсів. Наприклад, для утворення фізичного *кільця* кожен комп'ютер повинен бути оснащений як мінімум двома мережними інтерфейсами для зв'язку з двома сусідами. А для створення *повнозв'язної* структури з N комп'ютерів необхідно, щоб у кожного з них був $N-1$ інтерфейс.

Адреси можуть бути числовими (наприклад, 129.26.255.255) і *символьними* (site.domain.ru). Одна і та ж адреса може бути записана в різних форматах, скажімо, числова адреса в попередньому прикладі 129.26.255.255 може бути записана і в шістнадцятиричному форматі цифрами – 81.1a.ff.ff.

Адреси можуть використовуватися для ідентифікації не тільки окремих інтерфейсів, але і їх груп (групові адреси). За допомогою групових адрес дані можуть прямувати відразу декільком вузлам. У багатьох технологіях комп'ютерних мереж підтримуються так звані широкомовні *адреси*. Дані, направлені за такою адресою, повинні бути доставлені всім вузлам мережі.

Безліч всіх адрес, які є допустимими в рамках деякої схеми адресації, називається адресним простором. Адресний простір може мати плоску (лінійну) (рис. 1.31) або ієрархічну (рис. 1.32) організацію. У першому випадку безліч адрес ніяк не структурована.

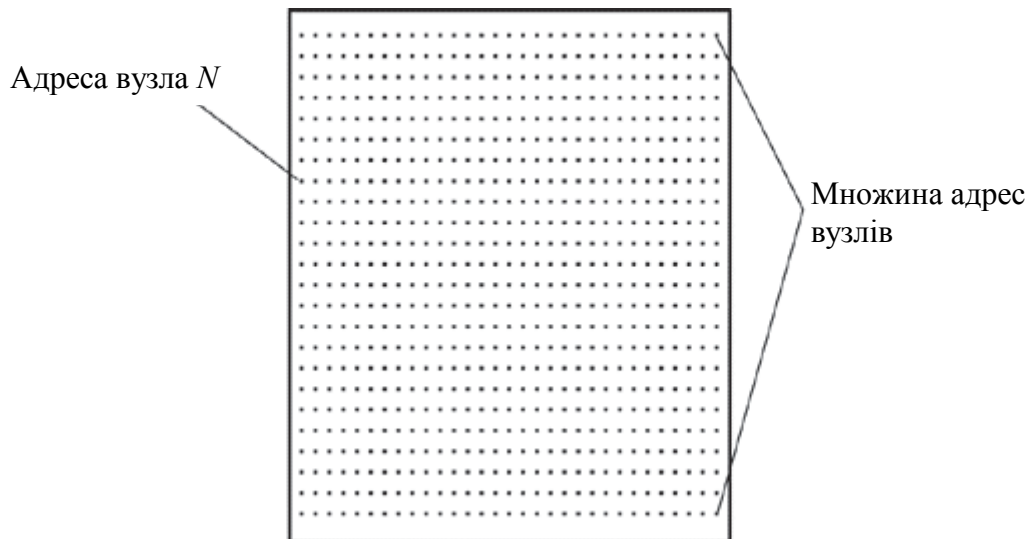


Рис. 1.31. Плоский адресний простір.

При ієрархічній схемі адресації воно організоване у вигляді вкладених один в одного підгруп, які, послідовно звужуючи область, яка адресується, вресіт-решт визначають окремий мережний інтерфейс.

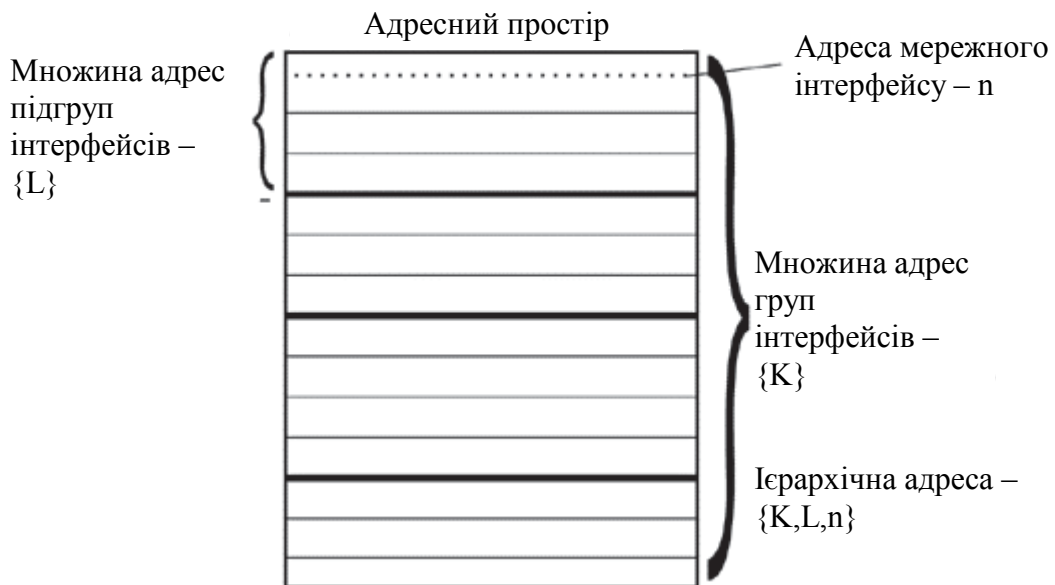


Рис. 1.32. Ієрархічна структура адресного простору.

На рис. 1.33 показана трирівнева структура адресного простору, при якій адреса кінцевого вузла задається трьома складовими: ідентифікатором групи (K), в яку входить даний вузол, ідентифікатором підгрупи (L) і, нарешті, ідентифікатором вузла (n), що однозначно визначає його в підгрупі. Ієрархічна адресація у багатьох випадках виявляється раціональнішою, ніж плоска. У великих мережах, що складаються з багатьох тисяч вузлів, використання плоских адрес може привести до великих витрат – кінцевим вузлам і комунікаційному устаткуванню доведеться працювати з таблицями адрес, що складаються з тисяч записів. А ієрархічна система адресації дозволяє при переміщенні даних до певного моменту користуватися тільки старшою складовою адреси, потім для подальшої локалізації адресата наступною за старшинством частиною, і кінець кінцем – молодшою частиною. Прикладом ієрархічно організованих адрес служать звичайні поштові адреси, в яких послідовно уточнюється місцезнаходження адресата: країна, місто, вулиця, будинок, квартира.

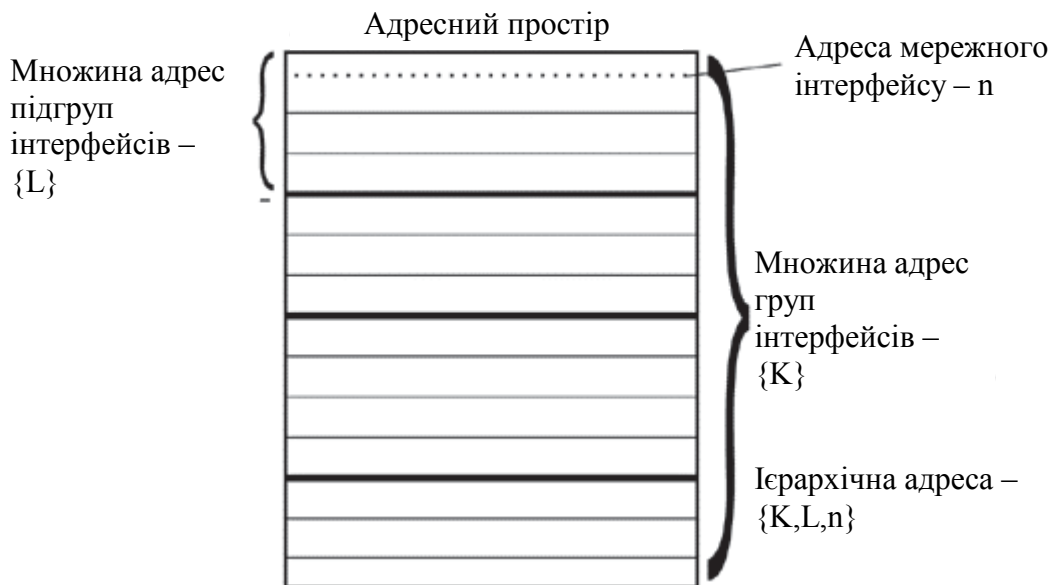


Рис. 1.33. Ієрархічна структура адресного простору.

До адреси мережного інтерфейсу і схеми його призначення можна пред'явити декілька вимог:

- адреса повинна унікально ідентифікувати мережний інтерфейс в мережі будь-якого масштабу;
- схема призначення адрес повинна зводити до мінімуму ручну працю адміністратора і вірогідність дублювання адрес;
- бажано, щоб адреса мала ієрархічну структуру, зручну для побудови великих мереж;
- адреса повинна бути зручна для користувачів мережі, а це означає, що вона повинна допускати символічне уявлення, наприклад, Server3 або www.cite.com;
- адреса повинна бути по можливості компактною, щоб не перенавантажувати пам'ять комунікаційної апаратури – мережних адаптерів, маршрутизаторів і т.п.

Неважко відмітити, що ці вимоги суперечливі – наприклад, адреса, що має ієрархічну структуру, швидше за все, буде менш компактною, чим плоска. Символьні імена зручні, але через змінний формат і потенційно велику довжину їх передачі по мережі не дуже економічні. Оскільки всі перераховані вимоги важко сумістити в рамках якої-небудь однієї схеми адресації, на практиці зазвичай використовується відразу декілька схем, так що мережний інтерфейс комп'ютера може одночасно мати декілька адрес-імен. Кожна адреса використовується в тій ситуації, коли відповідний вид адресації найбільш зручний. А для перетворення адрес з одного вигляду в інший використовуються спеціальні допоміжні протоколи, які називають іноді **протоколами дозволу адрес (address resolution)**.

Прикладом плоскої числової адреси є *MAC*-адреса, яка використовується для однозначної ідентифікації мережних інтерфейсів в локальних мережах. Така адреса зазвичай застосовується тільки апаратурою, тому її прагнуть зробити по можливості компактною і записують у вигляді двійкового або шестнадцятиричного значення, наприклад, 0081005e24a8. Коли задаються *MAC*-адреси, уручну нічого робити не потрібно, оскільки вони зазвичай вбудовуються в апаратуру компанією-виробником; їх називають ще **апаратними (hardware) адресами**. Використання плоских адрес є жорстким рішенням – при заміні апаратури, наприклад, мережного адаптера, змінюється і адреса мережного інтерфейсу комп'ютера.

Типовими представниками ієрархічних числових адрес є мережні *IP*-і *IPX*-адреси. У них підтримується дворівнева ієрархія, адреса ділиться на старшу частину – номер мережі і молодшу – номер вузла. Таке розділення дозволяє передавати повідомлення між мережами тільки на підставі номера мережі, а номер вузла використовується після доставки повідомлення в потрібну мережу; точно так, як і назва вулиці використовується листоношею тільки після того, як лист доставлений в потрібне місто. Останнім часом, щоб зробити маршрутизацію в великих мережах ефективнішою, пропонуються складніші варіанти числової адресації, відповідно до яких адреса має три і більш за складові. Такий підхід, зокрема, реалізований в новій версії протоколу *IPv6*, призначеного для роботи в *Internet*.

Символьні адреси або імена призначені для запам'ятовування людьми і тому зазвичай несуть смислове навантаження. Символьні адреси можна використовувати як в невеликих, так і в великих мережах. Для роботи у великих мережах символічне ім'я може мати ієрархічну структуру, наприклад, ftp-arch1.ucl.ac.uk. Ця адреса говорить про те, що даний комп'ютер підтримує *FTP*-архів в мережі одного з коледжів Лондонського університету (University College London – ucl), і дана мережа відноситься до академічної гілки (ac) *Internet* Великобританії (United Kingdom – uk). При роботі в межах мережі Лондонського університету таке довге символічне ім'я явно надмірно і замість нього можна користуватися коротким символічним ім'ям, на роль якого добре підходить наймолодша складова повного імені, тобто ftp-arch1.

У сучасних мережах для адресації вузлів, як правило, застосовуються всі три приведені вище схеми одночасно. Користувачі адресують комп'ютери символьними іменами, які автоматично замінюються в повідомленнях, що передаються по мережі, на числові номери. За допомогою цих числових номерів повідомлення передаються з однієї мережі в іншу, а після доставки повідомлення в мережу призначення замість числового номера використовується апаратна адреса комп'ютера. Сьогодні така схема характерна навіть для невеликих автономних мереж, де, здавалося б, вона явно надмірна. Це робиться для того, щоб при включенні мережі у велику мережу не потрібно було міняти склад операційної системи.

Проблема встановлення відповідності між адресами різних типів, якою займаються протоколи дозволу адрес, може вирішуватися як централізованими, так і розподіленими засобами. У разі централізованого підходу в мережі виділяється один або декілька комп'ютерів (серверів імен), в яких зберігається таблиця відповідності один одному імен різних типів, наприклад, символьних імен і числових номерів. Решта всіх комп'ютерів звертається до сервера імен, щоб по символьному імені знайти числовий номер комп'ютера, з яким необхідно обмінятися даними.

При розподіленому підході кожен комп'ютер сам вирішує задачу встановлення відповідності між адресами. Наприклад, якщо користувач вказав для вузла призначення числовий номер, то перед початком передачі даних комп'ютер-відправник посилає всім комп'ютерам мережі широкомовне повідомлення з проханням пізнати це числове ім'я. Всі комп'ютери, отримавши таке повідомлення, порівнюють заданий номер з своїм власним. Той комп'ютер, у якого виявився збіг, посилає відповідь, яка містить його апаратну адресу, після чого стає можливою відправка повідомлень по локальній мережі.

Розподілений підхід хороший тим, що не припускає виділення спеціального комп'ютера, на якому, до того ж часто, доводиться вручну вводити таблицю відповідності адрес. Недоліком розподіленого підходу є необхідність широкомовних повідомлень – такі повідомлення перенавантажують мережу, оскільки вони вимагають обов'язкової обробки всіма вузлами, а не тільки вузлом призначення. Тому розподілений підхід використовується тільки в невеликих локальних мережах. У великих мережах розповсюдження широкомовних повідомлень по всіх її сегментах стає практично нереальним, тому для них характерний централізований підхід. Найбільш відомою службою централізованого дозволу адрес є система доменних імен (Domain Name System, *DNS*) мережі *Internet*.

Адреси можуть використовуватися для ідентифікації:

- окремих інтерфейсів;
- їх груп (групові адреси);
- відразу всіх мережних інтерфейсів мережі (широкомовні адреси).

Адреси можуть бути:

- числовими і символьними;
- апаратними і мережними;
- плоскими і ієрархічними.

Для перетворення адрес з одного вигляду в інший використовуються протоколи дозволу адрес (address resolution).

Таким чином адреси мережних інтерфейсів указують на порти вузлів мережі (комп'ютерів і комунікаційних пристроїв), проте кінцевою метою даних, які пересилаються по мережі, є не комп'ютери або маршрутизатори, а виконувані на цих пристроях програми – процеси. Тому в адресі призначення разом з інформацією, що ідентифікує порт пристрою, повинна вказуватися адреса процесу, якому призначені послані дані. Після того, як ці дані досягнуть вказаного в адресі призначення мережного інтерфейсу, програмне забезпечення комп'ютера повинне їх направити відповідному процесу. Зрозуміло, що адреса процесу не обов'язково повинна задавати його однозначно в межах всієї мережі, досить забезпечити його унікальність в межах комп'ютера. Прикладом адрес процесів можуть служити номери портів TCP і UDP, використовуваних в стеку *TCP/IP*.

Ще одним найважливішим завданням побудови мереж є створення ефективного механізму комутації. У наступному підрозділі розглядається це фундаментальне поняття з найзагальніших позицій.

1.4. Комутація і мультиплексування

1.4.1. Узагальнена задача комутації

Якщо топологія мережі не повнозв'язна, то обмін даними між довільною парою кінцевих вузлів (абонентів) повинен йти в загальному випадку через транзитні вузли. Наприклад, в мережі на рис. 1.34 вузли 2 і 4, безпосередньо один з одним не зв'язані, вони вимушені передавати дані через транзитні вузли, в якості яких можуть використовуватися, наприклад, вузли 1 і 5. Вузол 1 повинен виконати передачу даних з інтерфейсу А на інтерфейс В, а вузол 5 – з інтерфейсу F на В.

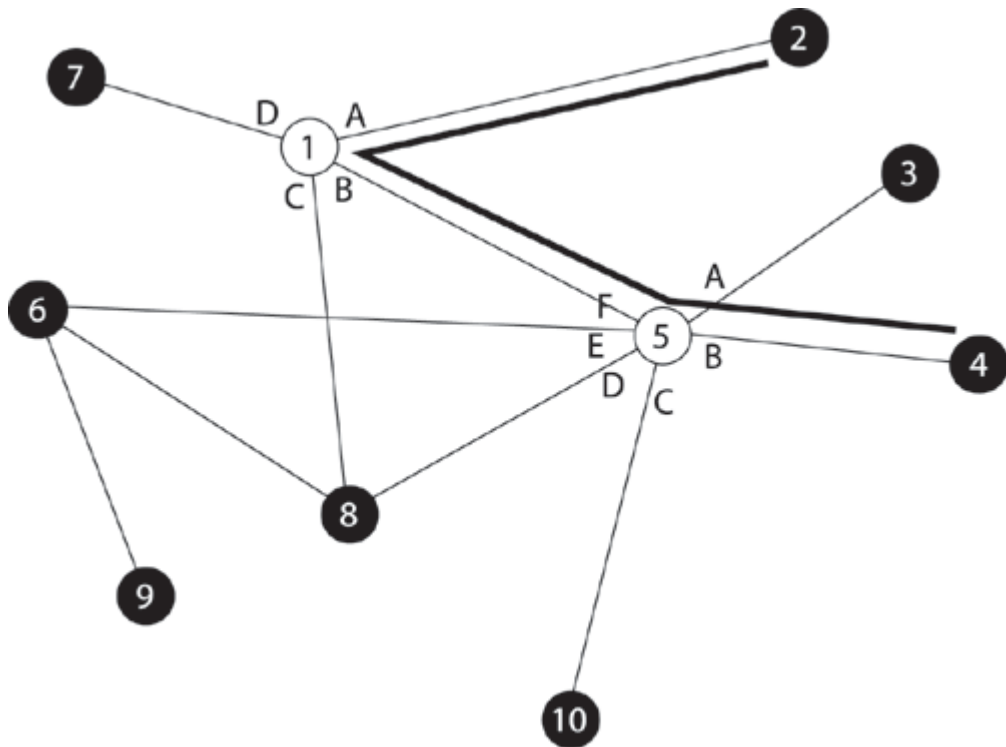


Рис. 1.34. Комутація абонентів через мережу транзитних вузлів.

Послідовність транзитних вузлів (мережних інтерфейсів) на шляху від відправника до одержувача називається маршрутом. У найзагальнішому вигляді завдання комутації – завдання з'єднання кінцевих вузлів через мережу транзитних вузлів – може бути представлено у вигляді декількох взаємозв'язаних приватних завдань:

1. Визначення інформаційних потоків, для яких потрібно прокласти шляхи.
2. Визначення маршрутів для потоків.
3. Повідомлення про знайдені маршрути вузлам мережі.
4. Просування-розпізнавання *потоків* і локальна комутація на кожному транзитному вузлі.
5. Мультиплексування і демультіплексування потоків.

Визначення інформаційних потоків

Зрозуміло, що через один транзитний вузол може проходити декілька маршрутів, наприклад, через вузол 5 проходять дані, що направляються вузлом 4 кожному з решти вузлів, а також всі дані, що поступають у вузли 3 і 10. Транзитний вузол повинен уміти розпізнавати потоки даних, що поступають на нього, щоб забезпечувати їх передачу саме на ті свої інтерфейси, які ведуть до потрібного вузла.

Інформаційним потоком (*data flow, data stream*) називають послідовність даних, об'єднаних набором загальних ознак, який виділяє ці дані із загального мережного трафіку. Дані можуть бути представлені у вигляді послідовності байтів або об'єднані в крупніші одиниці даних – пакети, кадри, осередки. Наприклад, всі дані, що поступають від одного комп'ютера, можна визначити як єдиний потік, а можна представити як сукупність декількох підпотоків, кожен з яких в якості додаткової ознаки має адресу призначення. Кожний з цих підпотоків, у свою чергу, можна розділити на ще дрібніші підпотоки даних, наприклад, що відносяться до різних мережних застосувань – електронної пошти, копіювання файлів, звернення до Web-сервера.

Поняття потоку використовується при рішенні різних мережних задач, і залежно від конкретного випадку, визначається відповідний набір ознак. У завданні комутації, суть якої – передача даних з одного кінцевого вузла в інший, при визначенні потоків в ролі обов'язкових ознак потоку, очевидно, повинні виступати адреса відправника і адреса призначення даних. Тоді кожній парі кінцевих вузлів відповідатиме один потік і один маршрут.

Проте не завжди досить визначити потік тільки парою адрес. Якщо на одній і тій же парі кінцевих вузлів виконується декілька прикладних програм, які взаємодіють по мережі і які пред'являють до неї свої особливі вимоги, потік даних між двома кінцевими вузлами повинен бути розділений на декілька підпотоків так, щоб для кожного з них можна було прокласти свій маршрут. У такому разі вибір шляху повинен здійснюватися з урахуванням характеру переданих даних. Наприклад, для файлового сервера важливо, щоб передані ним великі об'єми даних прямували по каналах з високою пропускнуною спроможністю, а для програмної системи управління, яка посилає в мережу короткі повідомлення, що вимагають обов'язкового і негайного відробітку, при виборі маршруту важливіше надійність лінії зв'язку і мінімальний рівень затримок. У такому прикладі набір ознак потоку повинен бути розширений за рахунок інформації, що ідентифікує прикладну програму.

Крім того, навіть для даних, які пред'являють до мережі однакові вимоги, може прокладатися декілька маршрутів, щоб за рахунок розпаралелювання добитися одночасного використання різних каналів і тим самим прискорити передачу даних. В даному випадку необхідно "помітити" дані, які прямуватимуть по кожному з цих маршрутів.

Ознаки потоку можуть мати глобальне або локальне значення. У першому випадку вони однозначно визначають потік в межах всієї мережі, а в другому – в межах одного транзитного вузла. Пара унікальних адрес кінцевих вузлів для ідентифікації потоку – це приклад глобальної ознаки. Прикладом ознаки, що локально визначає потік в межах пристрою, може служити номер (ідентифікатор) інтерфейсу пристрою, з якого поступили дані. Наприклад, вузол 1 (рис. 1.34) може бути конфігурований так, що він передає всі дані, що поступили з інтерфейсу А, на інтерфейс З, а дані, що поступили з інтерфейсу D, на інтерфейс В. Таке правило дозволяє розділити два потоки даних – що поступає з вузла 2 і що поступає з вузла 7, і направляти їх для транзитної передачі через різні вузли мережі, в даному випадку дані з вузла 2 через вузол 8, а дані з вузла 7 – через вузол 5.

Існує особливий тип ознаки – мітка потоку. Мітка може мати глобальне значення, що унікально визначає потік в межах мережі. У такому разі вона в незмінному вигляді закріплюється за потоком на всьому протязі його шляху проходження від вузла джерела до вузла призначення. У деяких технологіях використовуються локальні мітки потоку, значення яких динамічно міняються при передачі даних від одного вузла до іншого. Визначити потоки – це означає задати для них набір характерних ознак, на підставі яких комутатори зможуть направляти потоки по призначених для них маршрутах.

Визначення маршрутів

Визначення шляху, тобто послідовності транзитних вузлів і їх інтерфейсів, через які треба передавати дані, щоб доставити їх адресатові – складне завдання, особливо коли конфігурація мережі така, що між парою взаємодіючих мережних інтерфейсів існує безліч шляхів. Завдання визначення маршрутів полягає у виборі зі всієї цієї множини одного або декількох шляхів. І хоча в окремому випадку множини наявних і вибраних шляхів можуть співпадати, найчастіше вибір зупиняють на одному оптимальному по деякому критерію маршруті. На практиці для зменшення обчислювальної роботи обмежуються пошуком не оптимального в математичному значенні, а раціонального, тобто близького до оптимального, маршруту. Це стосується, перш за все, випадку, коли завдання вибору маршруту формулюється як завдання багатокритеріальної оптимізації. В якості критерія вибору можуть виступати, наприклад:

- номінальна пропускна спроможність;
- завантаженість каналів зв'язку;
- затримки, що вносяться каналами;
- кількість проміжних транзитних вузлів;
- надійність каналів і транзитних вузлів.

Відмітимо, що навіть у тому випадку, коли між кінцевими вузлами існує єдиний шлях, його визначення при складній топології мережі може бути нетривіальним завданням. Маршрут може визначатися емпірично ("вручну") адміністратором мережі, який, використовуючи різні міркування, що часто не формалізуються, аналізує топологію мережі і задає послідовність інтерфейсів, яку повинні пройти дані, щоб досягти одержувача. Серед спонукальних мотивів вибору того або іншого шляху можуть бути: особливі вимоги до мережі з боку різних типів прикладних програм, рішення передавати трафік через мережу певного провайдера, припущення про пікові навантаження на деякі канали мережі, міркування безпеки.

Проте евристичний підхід до визначення маршрутів для великої мережі з складною топологією не підходить. В цьому випадку таке завдання вирішується найчастіше автоматично. Для цього кінцеві вузли і інші пристрої мережі оснащуються спеціальними програмними засобами, які організують взаємний обмін службовими повідомленнями, що дозволяє кожному вузлу скласти своє уявлення про топологію мережі. Потім на основі цього дослідження і математичних алгоритмів визначаються найбільш раціональні маршрути. Визначити маршрут – це знаяти однозначно задати послідовність транзитних вузлів і їх інтерфейсів, через які треба передавати дані, щоб доставити їх адресатові.

Сповіщення мережі про вибраний маршрут

Після того, як маршрут визначений (вручну або автоматично), слід "повідомити" про нього всім пристроям мережі. Повідомлення про маршрут повинне нести кожному транзитному пристрою приблизно таку інформацію: "Якщо прийдуть дані, які відносяться до потоку n, то потрібно передати їх на інтерфейс F".

Повідомлення про маршрут обробляється транзитним пристроєм, внаслідок чого створюється новий запис в таблиці комутації. Локальний або глобальний ознаці (ознакам) потоку (наприклад, мітці, номеру вхідного інтерфейсу або адресі призначення) ставиться у відповідність номер інтерфейсу, на який пристрій повинен передати дані, що відносяться до цього потоку. Нижче поміщений фрагмент таблиці комутації (табл. 1.3), що містить запис, зроблений на підставі повідомлення про необхідність передачі потоку n на інтерфейс F.

Таблиця 1.3.

Ознаки потоку

n	F
---	---

Звичайно, структура повідомлення про маршрут і вмісту таблиці комутації залежить від конкретної технології, проте ці особливості не міняють суті даних процесів.

Передача інформації про вибрані маршрути так само, як і визначення маршруту, може здійснюватися і вручну, і автоматично. Адміністратор мережі може зафіксувати маршрут, виконавши конфігурацію пристрою вручну, наприклад, жорстко скомутувавши на тривалий час певні пари вхідних і вихідних інтерфейсів (так працювали "телефонні панночки" на перших комутаторах). Він може також за власною ініціативою внести запис про маршрут в таблицю комутації. Проте оскільки топологія мережі і інформаційних потоків може мінятися (відмова або поява нових проміжних вузлів, зміна адрес або визначення нових потоків), то гнучке рішення задач визначення і призначення маршрутів припускає постійний аналіз стану мережі і оновлення маршрутів і таблиць комутації, що вимагає застосування засобів автоматизації.

Оповістити мережу про знайдені маршрути – це означає вручну або автоматично налагодити кожен комутатор так, щоб він "знав", в якому напрямі слід передавати кожен потік. Просування – це розпізнавання потоків і комутація на кожному транзитному вузлі. Коли мережа оповіщена про маршрути, вона може почати виконувати свої функції по з'єднанню або комутації абонентів. Для кожної пари абонентів ця операція може бути представлена сукупністю декількох (по числу транзитних вузлів) локальних операцій комутації. Відправник повинен виставити дані на той свій порт, з якого виходить знайдений маршрут, а всі транзитні вузли повинні відповідним чином виконати "перекидання" даних з одного свого порту на інший, іншими словами – виконати комутацію.

Пристрій, призначений для виконання комутації, називається комутатором (switch). Комутатор проводить комутацію вхідних в його порти інформаційних потоків, направляючи їх у відповідні вихідні порти (рис. 1.35).

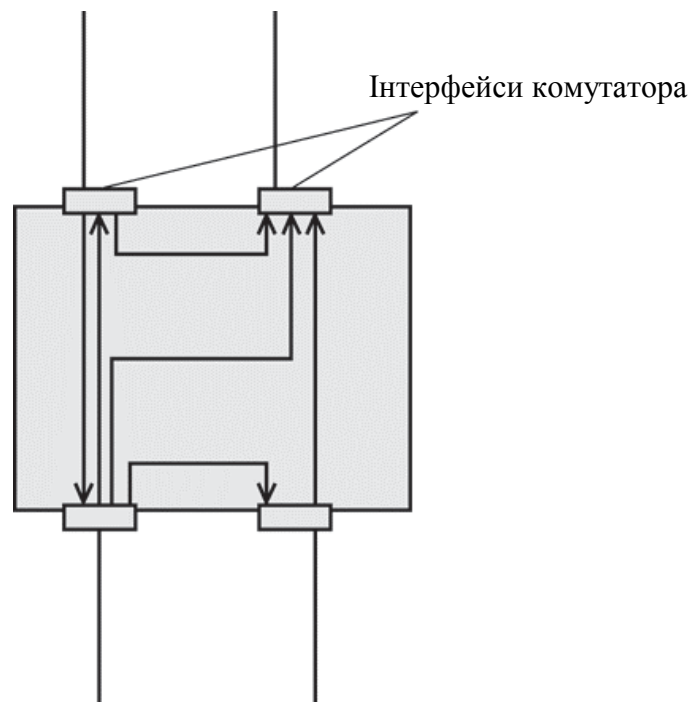


Рис. 1.35. Комутатор.

Проте, перш ніж виконати комутацію, комутатор повинен розпізнати потік. Дані, які для цього поступили, перевіряються на предмет наявності ознак якого-небудь з потоків, заданих в таблиці комутації. Якщо відбувся збіг, то ці дані прямують на той інтерфейс, який був визначений для них в маршруті.

Терміни комутація, таблиця комутації і комутатор в телекомунікаційних мережах можуть трактуватися неоднозначно. Термін комутація визначено як процес з'єднання абонентів мережі через транзитні вузли. Цим же терміном позначають і з'єднання інтерфейсів в межах окремого транзитного вузла. Комутатором в широкому значенні слова називається пристрій будь-якого типу, здатний виконувати операції перемикання потоку даних з одного інтерфейсу на інший. Операція комутації може бути виконана відповідно до різних правил і алгоритмів. Деякі способи комутації і відповідні ним таблиці і пристрої отримали спеціальні назви (наприклад, маршрутизація, таблиця маршрутизації, маршрутизатор). В той же час за іншими спеціальними типами комутації і відповідними пристроями закріпилися ті ж самі назви – комутація, таблиця комутації і комутатор, які тут використовуються у вузькому значенні, наприклад, комутація і комутатор локальної

мережі. У телефонних мережах, які з'явилися набагато раніше комп'ютерних, також використовується аналогічна термінологія, комутатор є тут синонімом телефонної станції. Через солідний вік і набагато більшу (поки) поширеність телефонних мереж найчастіше в телекомунікаціях під терміном «комутатор» розуміють саме телефонний комутатор.

Комутатором може бути як спеціалізований пристрій, так і універсальний комп'ютер з вбудованим програмним механізмом комутації, в цьому випадку комутатор називається програмним. Комп'ютер може суміщати функції комутації даних, що направляються на інші вузли, з виконанням своїх звичайних функцій кінцевого вузла. Проте у багатьох випадках раціональнішим є рішення, відповідно до якого деякі вузли в мережі виділяються спеціально для виконання комутації. Ці вузли утворюють комутаційну мережу, до якої підключаються всі інші. На рис. 1.36 показана комутаційна мережа, утворена з вузлів 1, 5, 6 і 8, до якої підключаються кінцеві вузли 2, 3, 4, 7, 9 і 10.

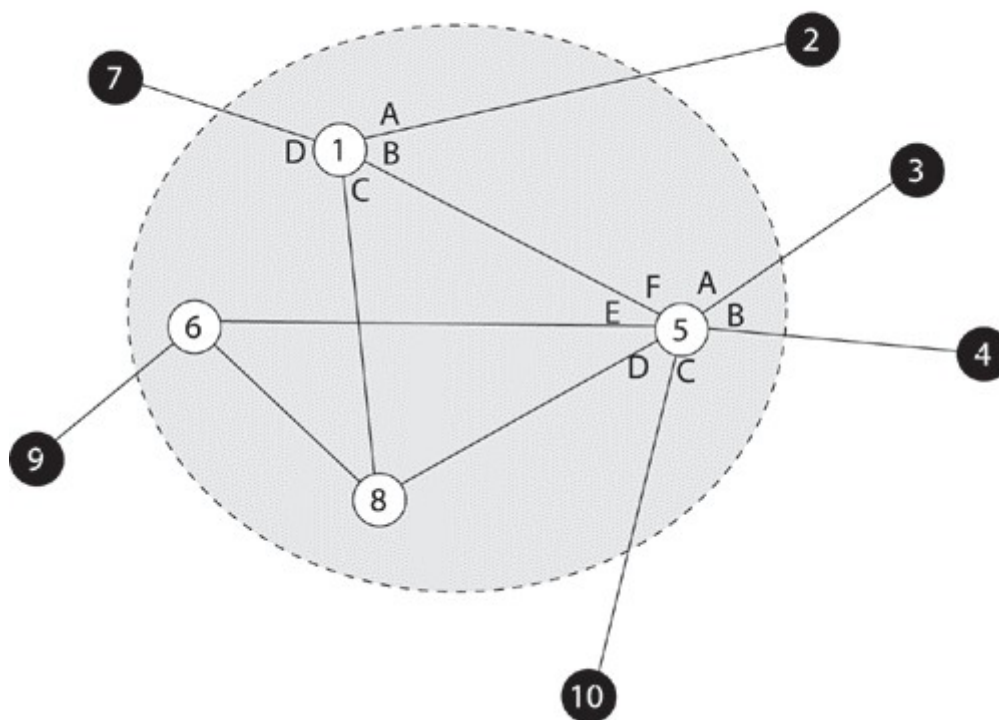


Рис. 1.36. Комутаційна мережа.

1.4.2. Мультиплексування і демюльтиплексування

Як вже було сказано, перш ніж виконати перекидання даних на визначені для них інтерфейси, комутатор повинен зрозуміти, до якого потоку вони відносяться. Це завдання повинне вирішуватися незалежно від того, чи поступає на вхід комутатора тільки один потік в "чистому" вигляді, або "змішаний" потік, який об'єднує в собі декілька потоків. У останньому випадку до завдання розпізнавання додається завдання демюльтиплексування.

Завдання демюльтиплексування (demultiplexing) – це розділення сумарного агрегованого потоку, що поступає на один інтерфейс, на декілька потоків. Як правило, операцію комутації супроводжує також зворотна операція – мультиплексування. Завдання мультиплексування (multiplexing) – утворення з декількох окремих потоків загального агрегованого потоку, який можна передавати поодиночці фізичному каналу зв'язку. Операції мультиплексування/демюльтиплексування мають таке ж важливе значення в будь-якій мережі, як і операції комутації, тому що без них довелось б всі комутатори зв'язувати великою кількістю паралельних каналів, що звало б нанівець всі переваги неповнозв'язної мережі.

На рис. 1.37 показаний фрагмент мережі, яка складається з трьох комутаторів. Комутатор 1 має п'ять мережних інтерфейсів. Розглянемо, що відбувається на інтерфейсі 1. Сюди поступають дані з трьох інтерфейсів – int.3, int.4 і int.5. Все їх треба передати в загальний фізичний канал, тобто виконати операцію мультиплексування. Мультиплексування є спосіб забезпечення доступності наявних фізичних каналів одночасно для декількох сеансів зв'язку між абонентами мережі.

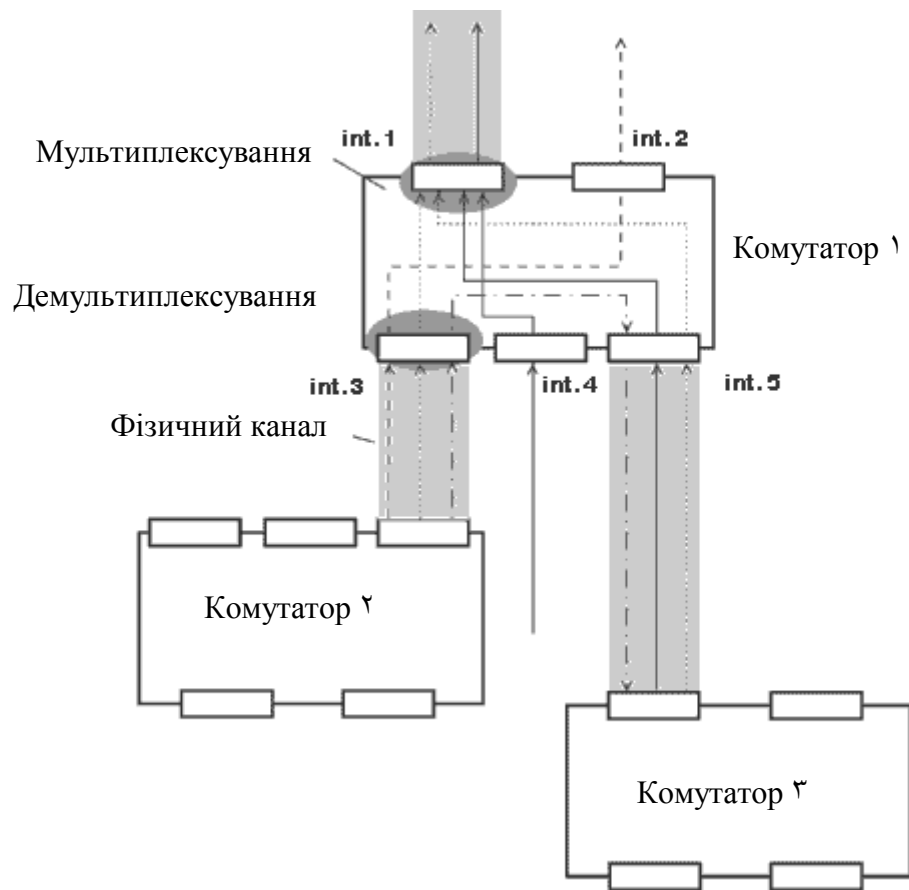


Рис. 1.37. Операції мультиплексування і демюльтиплексування потоків при комутації.

Існує безліч способів мультиплексування потоків в одному фізичному каналі і найважливішим з них є розділення часу. При цьому способі кожен потік час від часу (з фіксованим або випадковим періодом) отримує в своє розпорядження фізичний канал і передає по ньому дані. Дуже поширено також частотне розділення каналу, коли кожен потік передає дані у виділеному йому частотному діапазоні.

Технологія мультиплексування повинна дозволяти одержувачеві такого сумарного потоку виконувати зворотну операцію – розділення (демюльтиплексування) даних на потоки, що становлять. На інтерфейсі int.3 комутатор виконує демюльтиплексування потоку на три складових підпотіку. Один з них він передає на інтерфейс int. 1, інший на int.2, а третій на int.5. А ось на інтерфейсі int.2 немає необхідності виконувати мультиплексування або демюльтиплексування – цей інтерфейс виділений одному потоку в монопольне користування. У загальному випадку на кожному інтерфейсі можуть одночасно виконуватися обидва завдання – мультиплексування і демюльтиплексування.

Окремий вид комутатора (рис. 1.38а), у якого всі вхідні інформаційні потоки комутуються на один вихідний інтерфейс, де мультиплексується в один агрегований потік і прямує в один фізичний канал, називається мультиплексором (multiplexer, mux). Комутатор (рис. 1.38б), який має один вхідний інтерфейс і декілька вихідних, називається демюльтиплексором.

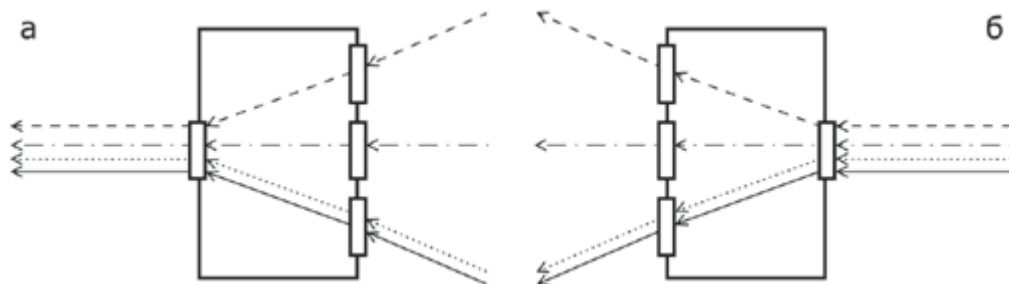


Рис. 1.38. Мультиплексор (а) і демюльтиплексор (б).

Середовище передачі даних, що розділяється

Ще один параметр, який характеризує використання каналів зв'язку, що розділяються, – кількість вузлів, підключених до такого каналу. У приведених вище прикладах до каналу зв'язку підключалися тільки два

взаємодіючі вузли, точніше – два інтерфейси. У телекомунікаційних мережах використовується і інший вид підключення, коли до одного каналу підключається декілька інтерфейсів. Таке множинне підключення інтерфейсів породжує топологію, що вже розглядалася вище, – "загальна шина", іноді звану також шлейфовим підключенням. У всіх цих випадках виникає проблема узгодженого використання каналу декількома інтерфейсами. Нижче на рисунках показані різні варіанти розділення каналів зв'язку між інтерфейсами.

У випадку на рис.1.39 комутатори K1 і K2 зв'язані двома однонаправленими фізичними каналами, тобто такими каналами, по яких інформація може передаватися тільки в одному напрямі. Інтерфейс, який в цьому випадку здійснює передачу, є активним, і фізичне середовище передачі знаходиться цілком і повністю під його управлінням. Пасивний інтерфейс тільки приймає дані. Проблеми розділення каналу між інтерфейсами тут немає. Відмітимо, проте, що завдання мультиплексування потоків даних в каналі при цьому зберігається. На практиці два однонаправлені канали, які реалізують в цілому дуплексний зв'язок між двома пристроями, зазвичай вважаються одним дуплексним каналом, а два інтерфейси одного пристрою розглядаються як частини одного і того ж інтерфейсу, що передає і приймає дані.

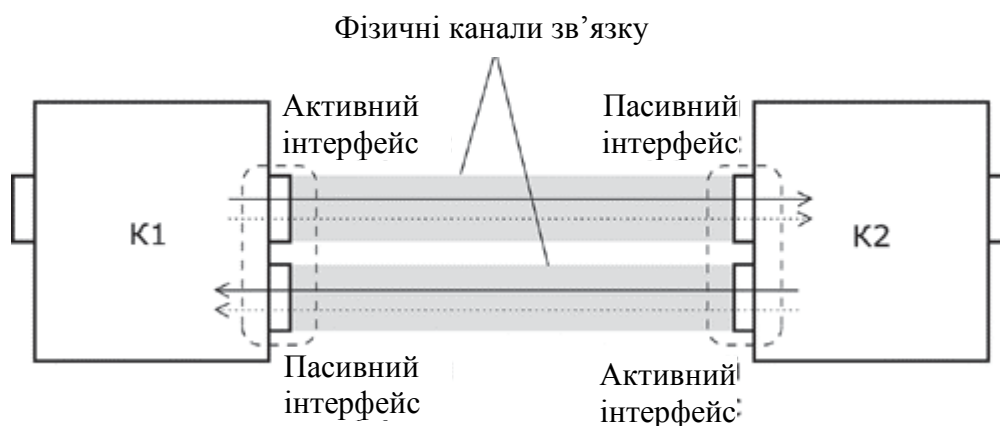


Рис. 1.39. Два однонаправлені фізичні канали.

На рис. 1.40 комутатори K1 і K2 зв'язані каналом, який може передавати дані в обидві сторони, але тільки поперемінно. При цьому виникає необхідність в механізмі узгодження доступу інтерфейсів K1 і K2 до такого каналу. Узагальненням цього варіанту є випадок, показаний на рис. 1.41, коли до каналу зв'язку підключається декілька (більше двох) інтерфейсів, утворюючи загальну шину.

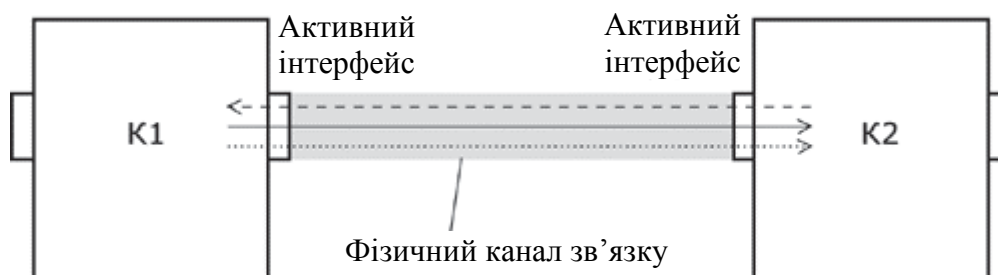


Рис. 1.40. Один напівдуплексний канал.

Фізичний канал, який спільно використовується декількома інтерфейсами, називають таким, що розділяється (*shared*). Потрібно підкреслити, що термін "середовище, що розділяється" (*shared media*), традиційно відносять саме до випадку розділення каналу між інтерфейсами і практично ніколи – до випадку розділення каналу між потоками. Часто використовується також термін "середовище передачі даних, яке розділяється". Канали зв'язку, що розділяються, використовуються не тільки для зв'язків типу комутатор-комутатор, але і для зв'язків комп'ютер-комутатор і комп'ютер-комп'ютер.

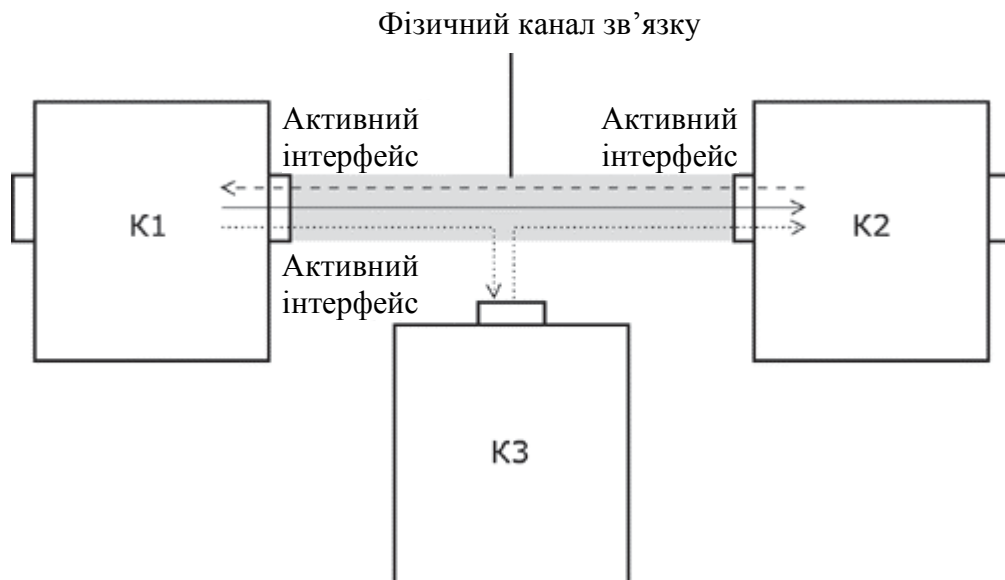


Рис. 1.41. Схема "загальна шина".

Існують різні способи організації сумісного доступу до ліній зв'язку, які розділяються. У одних випадках використовують централізований підхід, коли доступом управляє спеціальний пристрій – арбітр, в інших – децентралізований. У середині комп'ютера проблеми розділення ліній зв'язку між різними модулями також існують – прикладом може служити доступ до системної шини, яким управляє або процесор, або спеціальний арбітр шини. У мережах організація сумісного доступу до ліній зв'язку має свою специфіку через істотно більший час розповсюдження сигналів по лініях зв'язку. Тому процедури узгодження доступу до лінії зв'язку можуть займати дуже багато час і призводити до значного зниження продуктивності мережі.

Не дивлячись на всі ці складнощі, середовища, які розділяються, в локальних мережах, використовуються дуже часто. Цей підхід, зокрема, реалізований в широко поширених класичних технологіях *Ethernet*, *Token Ring*, *FDDI*. У глобальних мережах середовища, які розділяються між інтерфейсами, практично не використовуються. Це пояснюється тим, що великі тимчасові затримки при розповсюдженні сигналів уздовж протяжних каналів зв'язку приводять до дуже тривалих переговорних процедур доступу до середовища, яке розділяється, скорочуючи до неприйняттого рівня частку корисного використання каналу зв'язку на передачу даних абонентів.

Проте останніми роками і в локальних мережах намітилася тенденція відмови від середовищ передачі даних, які розділялися. Це пов'язано з тим, що при зниженні вартості мережі, яке досягається таким чином, доводиться розплачуватися продуктивністю. Мережа з середовищем, що розділяється, при великій кількості вузлів завжди працюватиме повільніше, ніж аналогічна мережа з індивідуальними лініями зв'язку, оскільки пропускна спроможність індивідуальної лінії зв'язку дістається одному комп'ютеру, а при сумісному використанні – ділиться на всі комп'ютери мережі. Часто з такою втратою продуктивності доводиться миритися ради збільшення економічної ефективності мережі. Не тільки у класичних, але і в зовсім нових технологіях, розроблених для локальних мереж, зберігається режим ліній зв'язку, які розділяються. Наприклад, розробники технології *Gigabit Ethernet*, прийнятої в 1998 році в якості нового стандарту, включили режим розділення середовища передачі в свої специфікації разом з режимом роботи по індивідуальних лініях зв'язку.

1.4.3. Різні підходи до виконання комутації

У загальному випадку рішення кожного з приватних завдань комутації – визначення потоків і відповідних маршрутів, фіксація маршрутів в конфігураційних параметрах і таблицях мережних пристроїв, розпізнавання потоків і передача даних між інтерфейсами одного пристрою, мультиплексування/демультиплексування потоків і розділення середовища передачі – тісно пов'язані з рішенням всіх інших. Комплекс технічних рішень узагальненої задачі комутації в сукупності складає базис будь-якої мережної технології. Від того, який механізм прокладки маршрутів, просування даних і сумісного використання каналів зв'язку закладений в тій або іншій мережній технології, залежать її фундаментальні властивості.

Серед безлічі можливих підходів до рішення задачі комутації абонентів в мережах виділяють два основних:

- комутація каналів (*circuit switching*);
- комутація пакетів (*packet switching*).

Зовні обидві ці схеми відповідають приведеній на рис. 1.42 структурі мережі, проте можливості і властивості їх різні.

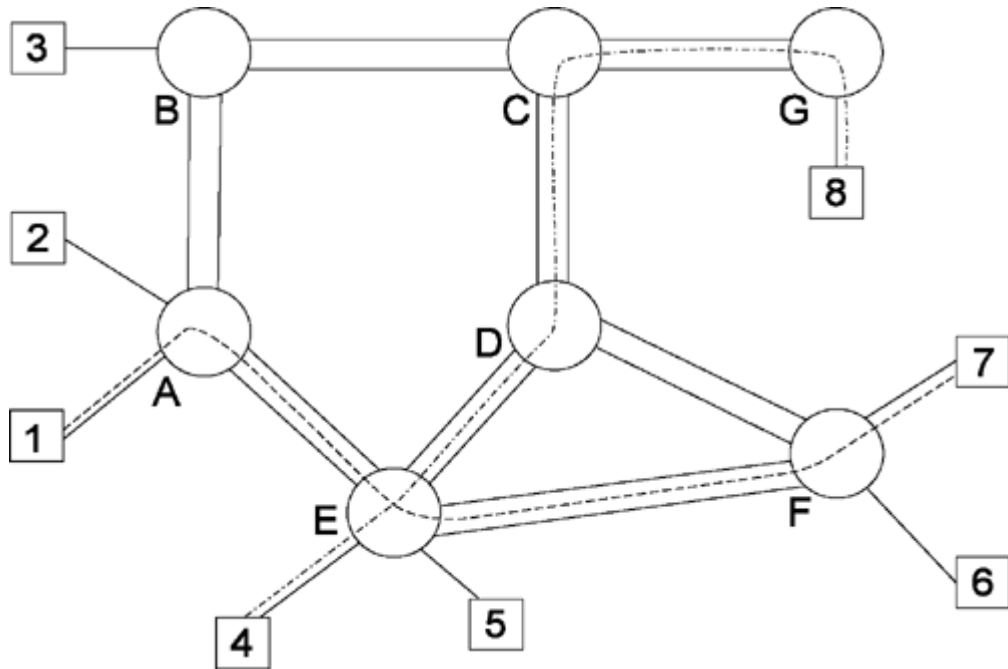


Рис. 1.42. Загальна структура мережі з комутацією абонентів.

Мережі з комутацією каналів мають багату історію, вони походять від перших телефонних мереж. Мережі з комутацією пакетів порівняно молоді, вони з'явилися в кінці 60-х років як результат експериментів з першими глобальними комп'ютерними мережами. Кожна з цих схем має свої достоїнства і недоліки, але за довгостроковими прогнозами багатьох фахівців, майбутнє належить технології комутації пакетів, як гнучкішої і універсальної.

Комутація каналів

При комутації каналів комутаційна мережа утворює між кінцевими вузлами безперервний складений фізичний канал з послідовно сполучених комутаторами проміжних каналних ділянок. Умовою того, що декілька фізичних каналів при послідовному з'єднанні утворюють єдиний фізичний канал, є рівність швидкостей передачі даних в кожному з фізичних каналів. Рівність швидкостей означає, що комутатори такої мережі не повинні буферизувати передані дані.

У мережі з комутацією каналів перед передачею даних завжди необхідно виконати процедуру встановлення з'єднання, в процесі якої і створюється складений канал (рис. 1.43). І лише після цього можна починати передавати дані.

Наприклад, якщо мережа, зображена на рис. 1.42, працює за технологією комутації каналів, то вузол 1, щоб передати дані вузлу 7, спочатку повинен передати спеціальний запит на встановлення з'єднання комутатору А, вказавши адресу призначення 7. Комутатор А повинен вибрати маршрут утворення складеного каналу, а потім передати запит наступному комутатору, в даному випадку Е. Потім комутатор Е передає запит комутатору F, а той, у свою чергу, передає запит вузлу 7. Якщо вузол 7 приймає запит на встановлення з'єднання, він направляє по вже встановленому каналу відповідь початковому вузлу, після чого складений канал вважається скомутованим, і вузли 1 і 7 можуть обмінюватися по ньому даними.

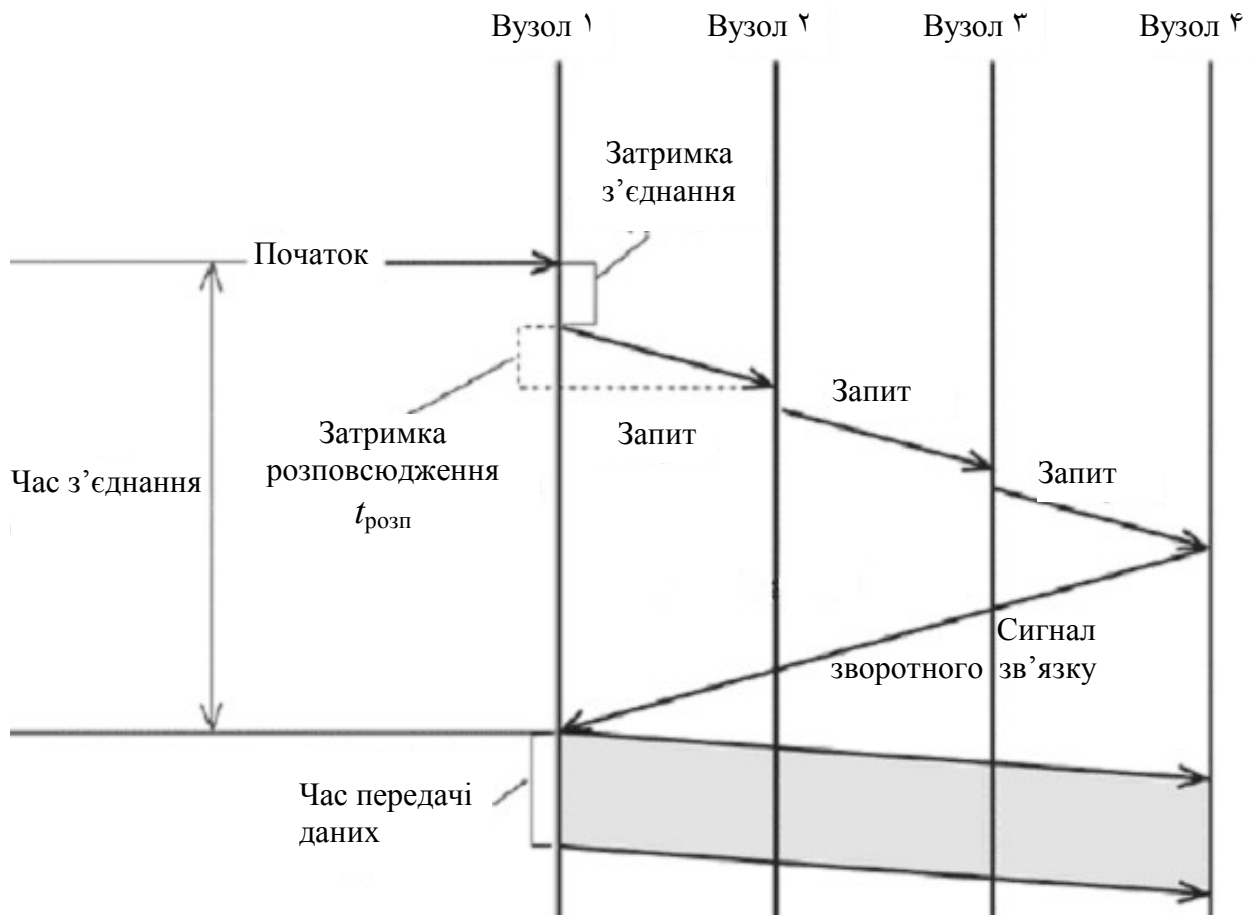


Рис. 1.43. Встановлення складеного каналу.

Техніка комутації каналів має свої переваги і недоліки. Переваги комутації каналів:

1. Постійна і відома швидкість передачі даних по встановленому між кінцевими вузлами каналу. Це дає користувачеві мережі можливість на основі наперед проведеної оцінки встановити в мережі канал потрібної швидкості.

2. Низький і постійний рівень затримки передачі даних через мережу. Це дозволяє якісно передавати дані, чутливі до затримок (звані також трафіком реального часу), – голос, відео, різну технологічну інформацію.

Недоліки комутації каналів:

1. Відмова мережі в обслуговуванні запиту на встановлення з'єднання. Така ситуація може скластися через те, що на деякій ділянці мережі з'єднання потрібно встановити уздовж каналу, через який вже проходить максимально можлива кількість інформаційних потоків. Відмова може трапитися і на кінцевій ділянці складеного каналу, наприклад, якщо абонент здатний підтримувати тільки одне з'єднання, яке характерне для багатьох телефонних мереж. Під час надходження другого виклику до вже розмовляючого абонента мережа передає зухвалому абонентові короткі гудки – сигнал "зайнято".

2. Нераціональне використання пропускної спроможності фізичних каналів. Та частина пропускної спроможності, яка відводиться складеному каналу після встановлення з'єднання, надається йому на весь час, тобто до тих пір, поки з'єднання не буде розірвано. Проте абонентам не завжди потрібна пропускна спроможність каналу під час з'єднання, наприклад, в телефонній розмові можуть бути паузи, ще більш нерівномірною в часі є взаємодія комп'ютерів. Нemoжливість динамічного перерозподілу пропускної спроможності є принципове обмеження мережі з комутацією каналів, оскільки одиницею комутації тут є інформаційний потік в цілому.

3. Обов'язкова затримка перед передачею даних через встановлення фази з'єднання.

Достоїнства і недоліки будь-якої мережної технології відносні. У певних ситуаціях на перший план виходять достоїнства, а недоліки стають неістотними. Так, техніка комутації каналів добре працює в тих випадках, коли потрібно передавати тільки трафік телефонних розмов. При комутації каналів можна миритися з неможливістю "вирізувати" паузи з розмови і раціональніше використовувати магістральні фізичні канали між комутаторами. А ось при передачі дуже нерівномірного комп'ютерного трафіку ця нераціональність вже виходить на перший план.

Комутація пакетів

Ця техніка комутації була спеціально розроблена для ефективної передачі комп'ютерного трафіку. Перші кроки на шляху створення комп'ютерних мереж на основі техніки комутації каналів показали, що цей вид комутації не дозволяє досягти високої загальної пропускної спроможності мережі. Типові мережні застосування генерують трафік дуже нерівномірно, з високим рівнем пульсації швидкості передачі даних. Наприклад, при зверненні до віддаленого файлового сервера користувач спочатку проглядає зміст каталога цього сервера, що породжує передачу невеликого об'єму даних. Потім він відкриває необхідний файл в текстовому редакторі, і ця операція може створити достатньо інтенсивний обмін даними, особливо якщо файл містить об'ємні графічні включення. Після відображення декількох сторінок файлу користувач якийсь час працює з ними локально, що взагалі не вимагає передачі даних по мережі, а потім повертає модифіковані копії сторінок на сервер – і це знову породжує інтенсивну передачу даних по мережі.

Коефіцієнт пульсації трафіку окремого користувача мережі, рівний відношенню середньої інтенсивності обміну даними до максимально можливої, може досягати 1:50 або навіть 1:100. Якщо для описаної сесії організувати комутацію каналу між комп'ютером користувача і сервером, то велику частину часу канал простоюватиме. В той же час комутаційні можливості мережі будуть закріплені за даною парою абонентів і будуть недоступні іншим користувачам мережі.

При комутації пакетів всі передані користувачем повідомлення розбиваються в початковому вузлі на порівняно невеликі частини, які наиваються пакетами. Кожен пакет містить адресу призначення та адресу відправника. Нагадаємо, що повідомленням називається логічно завершена порція даних – запит на передачу файлу, відповідь на цей запит, що містить весь файл і т.д. Повідомлення можуть мати довільну довжину, від декількох байт до багатьох мегабайт. Навпаки, пакети зазвичай теж можуть мати змінну довжину, але у вузьких межах, наприклад, від 46 до 1500 байт. Кожен пакет забезпечується заголовком, в якому указується адресна інформація, необхідна для доставки пакету на вузол призначення, а також номер пакету, який використовуватиметься вузлом призначення для збірки повідомлення (рис. 1.44). Пакети транспортуються по мережі як незалежні інформаційні блоки. Попередня процедура встановлення з'єднання тут не потрібна. Комутатори мережі приймають пакети від кінцевих вузлів і на підставі адресної інформації передають їх один одному, а зрештою – вузлу призначення.

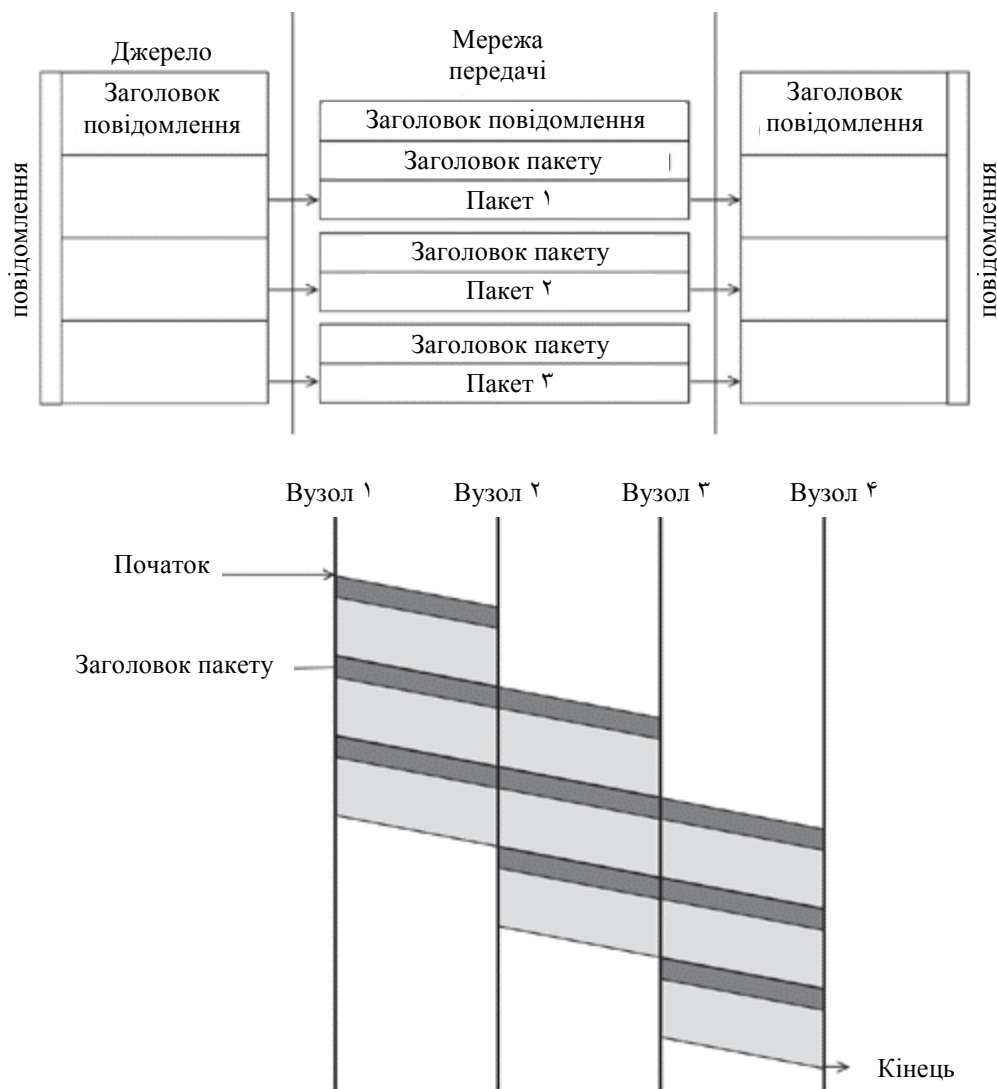


Рис. 1.44. Розбиття повідомлення на пакети.

Комутатори пакетної мережі відрізняються від комутаторів каналів тим, що вони мають внутрішню буферну пам'ять для тимчасового зберігання пакетів, якщо вихідний порт комутатора у момент ухвалення пакету зайнятий передачею іншого пакету (рис. 1.44). В цьому випадку пакет знаходиться якийсь час в черзі пакетів в буферній пам'яті вихідного порту, а коли до нього дійде черга, він передається наступному комутатору. Така схема передачі даних дозволяє згладжувати пульсацію трафіку на магістральних зв'язках між комутаторами і тим самим найефективніше використовувати їх для підвищення пропускної спроможності мережі в цілому.

Дійсно, для пари абонентів найбільш ефективним було б надання їм в одноосібне користування скомутованого каналу зв'язку, як це робиться в мережах з комутацією каналів. У такому разі час взаємодії цієї пари абонентів був би мінімальним, оскільки дані без затримок передавалися б від одного абонента іншому. Простої каналу під час пауз передачі абонентів не цікавлять, для них важливо швидше вирішити своє завдання. Мережа з комутацією пакетів уповільнює процес взаємодії конкретної пари абонентів, оскільки їх пакети можуть чекати в комутаторах, поки по магістральних зв'язках передаються інші пакети, що прийшли в комутатор раніше.

Проте, загальний об'єм переданих мережею комп'ютерних даних в одиницю часу при техніці комутації пакетів буде вищий, ніж при техніці комутації каналів. Це відбувається тому, що пульсації окремих абонентів відповідно до закону великих чисел розподіляються в часі так, що їх піки не співпадають. Тому комутатори постійно і достатньо рівномірно завантажені роботою, якщо число обслуговуваних ними абонентів дійсно велике. На рис. 1.45 показано, що трафік, який поступає від кінцевих вузлів на комутатори, розподілений в часі дуже нерівномірно. Проте комутатори вищого рівня ієрархії, які обслуговують з'єднання між комутаторами нижнього рівня, завантажені більш рівномірно, і потік пакетів в магістральних каналах, які сполучають комутатори верхнього рівня, має майже максимальний коефіцієнт використання. Буферизація згладжує пульсації, тому коефіцієнт пульсації на магістральних каналах набагато нижчий, ніж на каналах абонентського доступу – там він може бути рівним 1:10 або навіть 1:2.

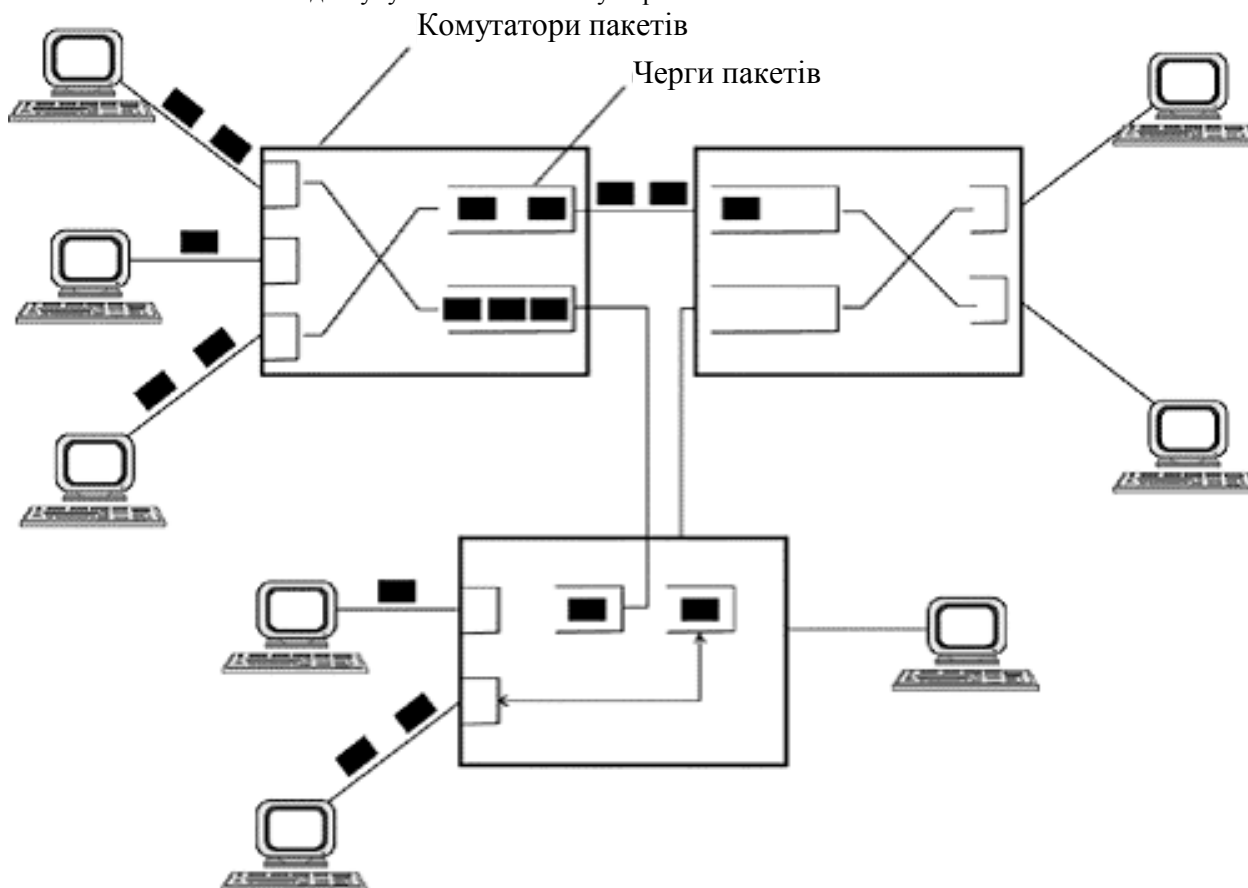


Рис. 1.45. Згладжування пульсацій трафіку в мережі з комутацією пакетів.

Вища ефективність мереж з комутацією пакетів в порівнянні з мережами з комутацією каналів (при рівній пропускній спроможності каналів зв'язку) була доведена в 60-і роки як експериментально, так і за допомогою імітаційного моделювання. Тут доречна аналогія з мультипрограмними операційними системами. Кожна окрема програма в такій системі виконується довше, ніж в однопрограмній системі, коли програми виділяється весь процесорний час, поки її виконання не завершиться. Проте загальне число програм, що виконуються за одиницю часу, в мультипрограмній системі більше, ніж в однопрограмній.

Мережа з комутацією пакетів уповільнює процес взаємодії конкретної пари абонентів, але підвищує пропускну спроможність мережі в цілому.

Достоїнства комутації пакетів:

1. Висока загальна пропускну спроможність мережі при передачі пульсуючого трафіку.
2. Можливість динамічно перерозподіляти пропускну спроможність фізичних каналів зв'язку між абонентами відповідно до реальних потреб їх трафіку.

Недоліки комутації пакетів:

1. Невизначеність швидкості передачі даних між абонентами мережі обумовлена тим, що затримки в чергах буферів комутаторів мережі залежать від загального завантаження мережі.
2. Змінна величина затримки *пакетів* даних, яка може бути достатнє тривалою в моменти миттєвих перевантажень мережі.
3. Можливі втрати даних через переповнювання буферів.

В даний час активно розробляються і упродовжуються методи, що дозволяють подолати вказані недоліки, які особливо гостро виявляються для чутливого до затримок трафіку, що вимагає при цьому постійної швидкості передачі. Такі методи називаються методами забезпечення **якості обслуговування QoS**.

Мережі з комутацією пакетів, в яких реалізовані методи забезпечення якості обслуговування, дозволяють одночасно передавати різні види трафіку, зокрема такі важливі як телефонний і комп'ютерний. Тому методи комутації пакетів сьогодні вважаються найбільш перспективними для побудови конвергентної мережі, яка забезпечить комплексні якісні послуги для абонентів будь-якого типу. Проте, не можна скидати з рахунків і методи комутації каналів. Сьогодні вони не тільки з успіхом працюють в традиційних телефонних мережах, але і широко застосовуються для утворення високошвидкісних постійних з'єднань в так званих первинних (опорних) мережах технологій *SDH* і *DWDM*, які використовуються для створення магістральних фізичних каналів між комутаторами телефонних або комп'ютерних мереж. В майбутньому цілком можлива поява нових технологій комутації, які в тому або іншому виді комбінують принципи комутації пакетів і каналів.

Комутація повідомлень

Комутація повідомлень за своїми принципами близька до комутації пакетів. Під комутацією повідомлень розуміється передача єдиного блоку даних між транзитними комп'ютерами мережі з тимчасовою буферизацією цього блоку на диску кожного комп'ютера. Повідомлення на відміну від пакету має довільну довжину, яка визначається не технологічними міркуваннями, а змістом інформації, складовій повідомлення.

Транзитні комп'ютери можуть з'єднуватися між собою як мережею з комутацією пакетів, так і мережею з комутацією каналів. Повідомлення (це може бути, наприклад, текстовий документ, файл з кодом програми, електронний лист) зберігається в транзитному комп'ютері на диску, причому досить тривалий час, якщо комп'ютер зайнятий іншою роботою або мережа тимчасово переобтяжена. По такій схемі зазвичай передаються повідомлення, які не вимагають негайної відповіді, частіше за все повідомлення електронної пошти. Режим передачі з проміжним зберіганням на диску називається режимом "зберігання і передачі" (store-and-forward).

Режим комутації повідомлень розвантажує мережа для передачі трафіку, наприклад, трафіку служби WWW або файлової служби. Кількість транзитних комп'ютерів зазвичай прагнуть зменшити. Якщо комп'ютери підключені до мережі з комутацією пакетів, то число проміжних комп'ютерів зменшується до двох. Наприклад, користувач передає поштове повідомлення свого сервера вихідної пошти, а той відразу прагне передати його серверу вхідної пошти адресата. Але якщо комп'ютери зв'язані між собою телефонною мережею, то часто використовується декілька проміжних серверів, оскільки прямий доступ до кінцевого сервера може бути в даний момент неможливий через перевантаження телефонної мережі (абонент зайнятий) або економічно не вигідний через високі тарифи на телефонну телекомунікацію.

Техніка комутації повідомлень з'явилася в комп'ютерних мережах раніше техніки комутації пакетів, але потім була витиснена останньою, як ефективніша по критерію пропускну спроможності мережі. Запис повідомлення на диск займає достатньо багато часу, і крім того, наявність дисків припускає використання в якості комутаторів спеціалізованих комп'ютерів, що спричиняє за собою істотні витрати на організацію мережі. Сьогодні комутація повідомлень працює тільки для деяких не оперативних служб, причому найчастіше поверх мережі з комутацією пакетів, в якості служби прикладного рівня.

1.4.4. Порівняння комутації

У табл. 1.4 дані порівняльні характеристики способів комутації.

Таблиця 1.4

Порівняння комутації каналів і комутації пакетів	
Комутація каналів	Комутація пакетів
Гарантована пропускну спроможність (смуга) для взаємодіючих абонентів	Пропускну спроможність мережі для абонентів невідома, затримки передачі носять випадковий характер
Мережа може відмовити абонентові у встановленні з'єднання	Мережа завжди готова прийняти дані від абонента

Трафік реального часу передається без затримок	Ресурси мережі використовуються ефективно при передачі пульсуючого трафіку
Адреса використовується тільки на етапі встановлення з'єднання	Адреса передається з кожним пакетом

Постійна і динамічна комутація

Як мережі з комутацією пакетів, так і мережі з комутацією каналів можна розділити на два класи:

- мережі з динамічною комутацією;
- мережі з постійною комутацією.

У мережах з динамічною комутацією:

- дозволяється встановлювати з'єднання за ініціативою користувача мережі;
- комутація виконується тільки на час сеансу зв'язку, а потім (за ініціативою одного з користувачів)

розривається;

- у загальному випадку користувач мережі може з'єднатися з будь-яким іншим користувачем мережі;
- час з'єднання між парою користувачів при динамічній комутації складає від декількох секунд до декількох годин і завершується після виконання певної роботи – передачі файлу, переглядання сторінки тексту або зображення і т.п.

Прикладами мереж, що підтримують режим динамічної комутації, є телефонні мережі загального користування, локальні мережі, мережі *TCP/IP*.

Мережа, що працює в режимі постійної комутації:

- дозволяє парі користувачів замовити з'єднання на тривалий період часу;
- з'єднання встановлюється не користувачами, а персоналом, обслуговуючим мережу;
- період, на який встановлюється постійна комутація, складає зазвичай декілька місяців;
- режим постійної (permanent) комутації в мережах з комутацією каналів часто називається сервісом виділених (dedicated) або таких, що орендуються (leased), каналів;

у тому випадку, коли постійне з'єднання через мережу комутаторів встановлюється за допомогою автоматичних процедур, ініційованих обслуговуючим персоналом, його часто називають напівпостійним (semi-permanent) з'єднанням, на відміну від режиму ручної конфігурації кожного комутатора.

Найбільш популярними мережами, що працюють в режимі постійної комутації, сьогодні є мережі технології SDH, на основі яких будуються виділені канали зв'язку з пропускнуою спроможністю в декілька гигабіт в секунду. Деякі типи мереж підтримують обидва режими роботи. Наприклад, мережі *X.25* і *ATM* можуть надавати користувачеві можливість динамічно зв'язатися з будь-яким іншим користувачем мережі і в той же час відправляти дані по постійному з'єднанню певному абоненту.

1.4.5. Пропускна спроможність мереж з комутацією пакетів

Однією з відмінностей методу комутації пакетів від методу комутації каналів є невизначеність пропускнуої спроможності з'єднання між двома абонентами. У разі комутації каналів після утворення складеного каналу пропускна спроможність мережі при передачі даних між кінцевими вузлами відома – це пропускна спроможність каналу. Дані після затримки, пов'язаної зі встановленням каналу, починають передаватися на максимальній для каналу швидкості (рис. 1.46). Час передачі повідомлення в мережі з комутацією каналів $T_{к.к.}$ дорівнює сумі затримки розповсюдження сигналу по лінії зв'язку і затримки передачі повідомлення. Затримка розповсюдження сигналу залежить від швидкості розповсюдження електромагнітних хвиль в конкретному фізичному середовищі, яке коливається від 0,6 до 0,9 швидкості світла у вакуумі. Час передачі повідомлення рівний V/C , де V – об'єм повідомлення в бітах, а C – пропускна спроможність каналу в бітах в секунду. У мережі з комутацією пакетів має місце інша картина. Процедура встановлення з'єднання в цих мережах, якщо вона використовується, займає приблизно такий же час, як і в мережах з комутацією каналів, тому порівнюватимемо тільки час передачі даних.

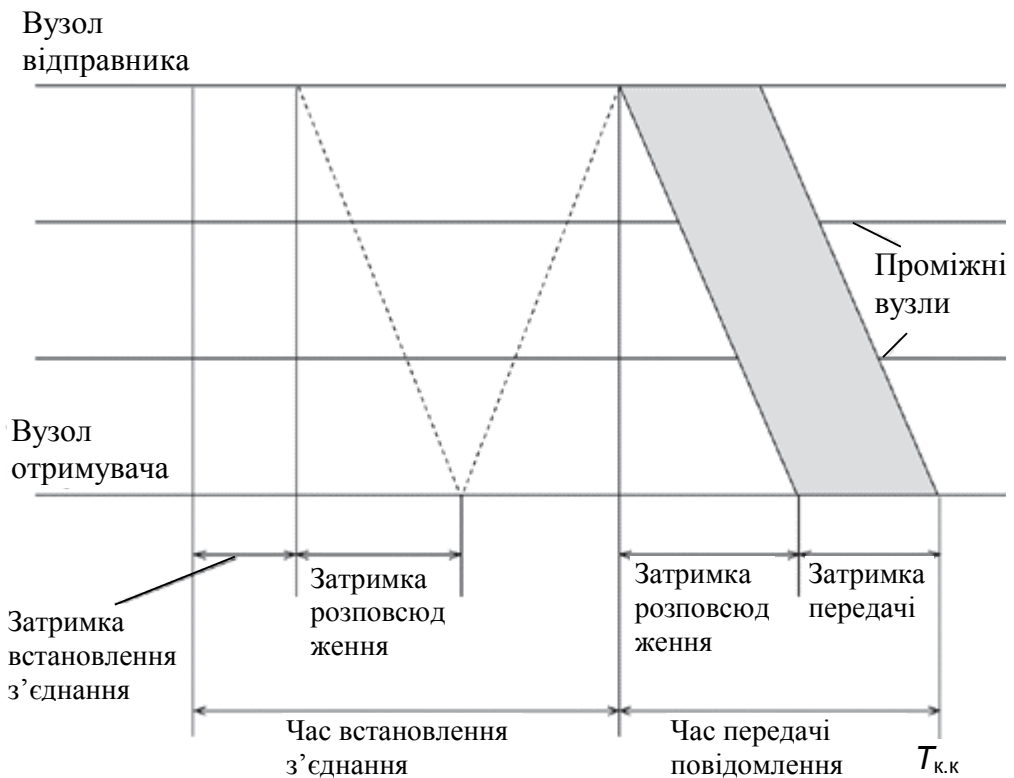


Рис. 1.46. Затримки передачі даних в мережах з комутацією каналів.

На рис. 1.47 показаний приклад передачі даних в мережі з комутацією пакетів. Передбачається, що по мережі передається повідомлення того ж об'єму, що і повідомлення, що передається на рис. 1.46, проте воно розділене на пакети, кожний з яких забезпечений заголовком. Час передачі повідомлення в мережі з комутацією пакетів позначений на рисунку $T_{кп}$. При передачі цього розбитого на пакети повідомлення по мережі з комутацією пакетів виникають додаткові затримки.

По-перше, це затримки в джерелі передачі, яке, крім передачі власне повідомлення, витрачає додатковий час на передачу заголовків $t_{пз}$, до того ж додаються затримки $t_{инт}$, викликані інтервалами між передачею кожного наступного пакету (цей час йде на формування чергового пакету стеком протоколів).

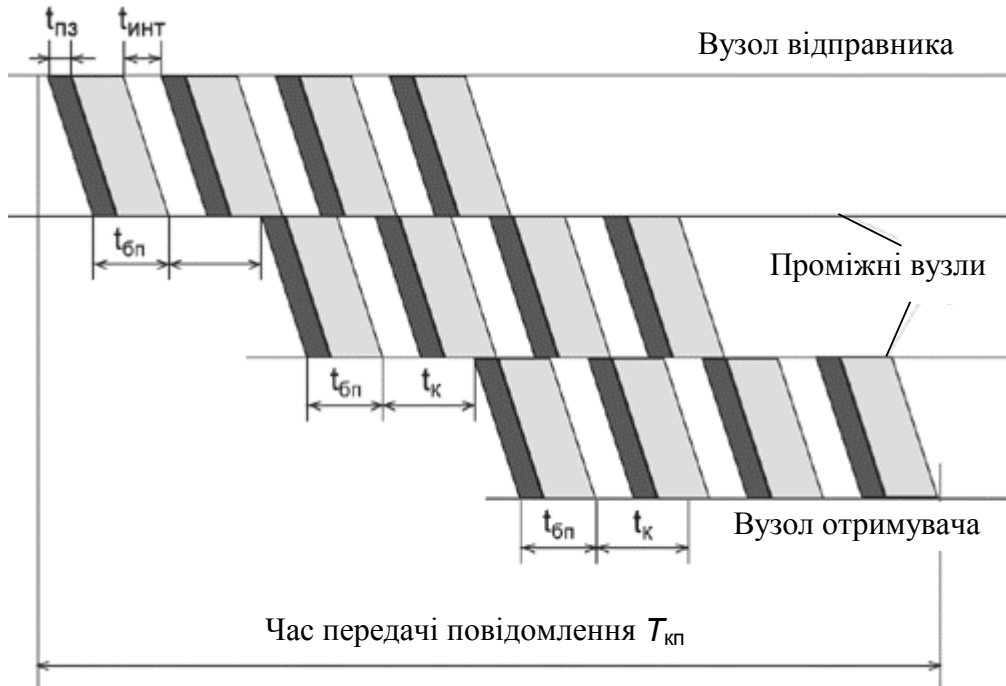


Рис. 1.47. Затримки при передачі даних в мережах з комутацією пакетів.

По-друге, додатковий час витрачається в кожному комутаторі. Тут затримки складаються з часу буферизації пакету $t_{бп}$. (комутатор не може почати передачу пакету, не прийнявши його повністю в свій

буфер) і часу комутації t_k . Час буферизації рівний часу прийому пакету з бітовою швидкістю протоколу. Час комутації складається з часу очікування пакету в черзі і часу переміщення пакету у вихідний порт. Якщо час переміщення пакету фіксований і, як правило, невеликий (від декількох мікросекунд до декількох десятків мікросекунд), то час очікування пакету в черзі коливається в дуже широких межах і наперед невідомо, оскільки залежить від поточного завантаження мережі.

Проведемо грубу оцінку затримки при передачі даних в мережах з комутацією пакетів в порівнянні з мережами з комутацією каналів на простому прикладі. Хай тестове повідомлення, яке потрібно передати в обох видах мереж, має об'єм 200 Кбайт. Відправник знаходиться від одержувача на відстані 5000 км. Пропускна спроможність ліній зв'язку складає 2 Мбіт/с.

Час передачі даних по мережі з комутацією каналів складається з часу розповсюдження сигналу, яке для відстані 5000 км. можна оцінити приблизно в 25 мс (приймаючи швидкість розповсюдження сигналу рівною 2/3 швидкості світла), і часу передачі повідомлення, яке при пропускній спроможності 2 Мбіт/с і довжині повідомлення 200 Кбайт рівні приблизно 800 мс. При розрахунку коректне значення $20 \cdot 10^3$, рівне 1024, округлялося до 1000, аналогічне значення $220 \cdot 10^3$, рівне 1048576, округлялося до 1000000. Таким чином, час передачі даних оцінюється в 825 мс.

Ясно, що при передачі цього повідомлення по мережі з комутацією пакетів, яка володіє такою ж сумарною довжиною і пропускною спроможністю каналів, що пролягають від відправника до одержувача, час розповсюдження сигналу і час передачі даних будуть такими ж – 825 мс. Проте через затримки в проміжних вузлах загальний час передачі даних збільшиться. Давайте оцінимо, на скільки зросте цей час. Вважатимемо, що шлях від відправника до одержувача пролягає через 10 комутаторів. Хай початкове повідомлення розбивається на пакети в 1 Кбайт, всього 200 пакетів. Спочатку оцінимо затримку, яка виникає в початковому вузлі. Припустимо, що частка службової інформації, розміщеної в заголовках пакетів, по відношенню до загального об'єму повідомлення, складає 10%. Отже, додаткова затримка, пов'язана з передачею заголовків пакетів, складає 10% від часу передачі цілого повідомлення, тобто 80 мс. Якщо прийняти інтервал між відправкою пакетів рівним 1 мс, то додаткові втрати за рахунок інтервалів складуть 200 мс. Таким чином, в початковому вузлі через пакетування повідомлення при передачі виникла додаткова затримка в 280 мс.

Кожний з 10 комутаторів вносить затримку комутації, яка може складати від долей до тисяч мілісекунд. У даному прикладі вважатимемо, що на комутацію в середньому витрачається 20 мс. Крім того, при проходженні повідомлень через комутатор виникає затримка буферизації пакету. Ця затримка при величині пакету 1 Кбайт і пропускної спроможності лінії 2 Мбіт/с рівна 4 мс. Загальна затримка, що вноситься 10 комутаторами, складає приблизно 240 мс. В результаті додаткова затримка, створена мережею з комутацією пакетів, склала 520 мс. Враховуючи, що вся передача даних в мережі з комутацією каналів зайняла 825 мс, цю додаткову затримку можна вважати істотною.

Хоча приведенний розрахунок носить дуже приблизний характер, він пояснює, чому процес передачі для певної пари абонентів в мережі з комутацією пакетів є повільнішим, ніж в мережі з комутацією каналів.

Невизначена пропускна спроможність мережі з комутацією пакетів – це плата за її загальну ефективність при деякому утиску інтересів окремих абонентів. Аналогічно, в мультипрограμній операційній системі час виконання прикладної програми передбачити неможливо, оскільки він залежить від кількості інших програм, з якими дана прикладна програма ділить процесор.

На ефективність роботи мережі впливають розміри пакетів, які передає мережа. Дуже великі розміри пакетів наближають мережу з комутацією пакетів до мережі з комутацією каналів, тому ефективність мережі падає. Крім того, при великому розмірі пакетів збільшується час буферизації на кожному комутаторі. Дуже маленькі пакети помітно збільшують частку службової інформації, оскільки кожен пакет містить заголовок фіксованої довжини, а кількість пакетів, на які розбиваються повідомлення, при зменшенні розміру пакету буде різко рости. Існує деяка "золота середина", коли забезпечується максимальна ефективність роботи мережі, проте це співвідношення важко визначити точно, оскільки воно залежить від багатьох чинників, зокрема тих, які змінюються в процесі роботи мережі. Тому розробники протоколів для мереж з комутацією пакетів вибирають межі, в яких може знаходитися розмір пакету, а точніше його поле даних, оскільки заголовок, як правило, має фіксовану довжину. Зазвичай нижня межа поля даних вибирається рівною нулю, що дає можливість передавати службові пакети без призначених для користувача даних, а верхня межа не перевищує 4 Кбайт. Програми при передачі даних намагаються зайняти максимальний розмір поля даних, щоб швидше виконати обмін, а невеликі пакети зазвичай використовуються для коротких службових повідомлень, що містять, наприклад, підтвердження доставки пакету.

При виборі розміру пакету необхідно також враховувати інтенсивність бітових помилок каналу. На ненадійних каналах необхідно зменшувати розміри пакетів, оскільки це скорочує об'єм повторно переданих даних при спотвореннях пакетів.

1.4.6. Ethernet – приклад стандартної технології комутації пакетів

Розглянемо, яким чином описані вище загальні підходи до вирішення проблем побудови мереж втілені в найбільш популярній мережній технології – *Ethernet*. Зупинимось лише на деяких принципових моментах, що ілюструють ряд вже розглянутих базових концепцій.

Мережна технологія – це узгоджений набір стандартних протоколів і програмно-апаратних засобів (наприклад, мережних адаптерів, драйверів, кабелів і роз'ємів), достатній для побудови обчислювальної мережі.

Епітет "достатній" підкреслює ту обставину, що йдеться про мінімальний набір засобів, за допомогою яких можна побудувати працездатну мережу. Цю мережу можна удосконалити, наприклад, за рахунок виділення в ній підмереж, що відразу потребує окрім протоколів стандарту Ethernet застосування протоколу IP, а також спеціальних комунікаційних пристроїв-маршрутизаторів. Вдосконалена мережа буде, швидше за все, надійнішою і швидкодіючою, але за рахунок надбудов над засобами технології Ethernet, яка склала базис мережі.

Термін "мережна технологія" найчастіше використовується в описаному вище вузькому значенні, але іноді застосовується і його розширене тлумачення як будь-якого набору засобів і правил для побудови мережі, наприклад, "технологія кризової маршрутизації", "технологія створення захищеного каналу", "технологія IP-мереж".

Протоколи, на основі яких будується мережа певної технології (у вузькому сенсі), створювалися спеціально для спільної роботи, тому від розробника мережі не вимагається додаткових зусиль по організації їх взаємодії. Іноді мережні технології називають базовими технологіями, маючи на увазі, що на їх основі будується базис будь-якої мережі. Прикладами базових мережних технологій можуть служити разом з *Ethernet* такі відомі технології локальних мереж як *Token Ring* і *FDDI*, або ж технології територіальних мереж *X.25* і *Frame relay*. Для отримання працездатної мережі в цьому випадку досить придбати програмні і апаратні засоби, що відносяться до однієї базової технології, – мережні адаптери з драйверами, концентратори, комутатори, кабельну систему і т. п., і з'єднати їх відповідно до вимог стандарту на дану технологію. Отже, для мережної технології Ethernet характерні:

- комутація пакетів;
- типова топологія "загальна шина";
- плоска числова адресація;
- передає середовище, що розділяється.

Основний принцип, покладений в основу *Ethernet* – це випадковий метод доступу до середовища передачі даних, яке розділяється. Таке середовище може використовувати як товстий або тонкий коаксіальний кабель, виту пару, оптоволокно або радіохвилі.

У стандарті *Ethernet* строго зафіксована топологія електричних зв'язків. Комп'ютери підключаються до середовища, що розділяється, відповідно до типової структури "загальна шина" (рис. 1.29). За допомогою шини, що розділяється в часі, будь-які два комп'ютери можуть обмінюватися даними. Управління доступом до лінії зв'язку здійснюється спеціальними контролерами-мережними адаптерами *Ethernet*. Кожен комп'ютер, а точніше, кожен мережний адаптер, має унікальну адресу. Передача даних відбувається із швидкістю 10 Мбіт/с. Ця величина є пропускною спроможністю мережі *Ethernet*.

Суть випадкового методу доступу полягає в наступному. Комп'ютер в мережі *Ethernet* може передавати дані по мережі, тільки якщо мережа вільна, тобто якщо ніякий інший комп'ютер в даний момент не займається обміном. Тому важливою частиною технології *Ethernet* є процедура визначення доступності середовища. Після того, як комп'ютер переконався, що мережа вільна, він починає передачу і при цьому "захоплює" середовище. Час монопольного використання одним вузлом середовища, яке розділяється, обмежується часом передачі одного кадру. Кадр – це одиниця даних, якою обмінюються комп'ютери в мережі *Ethernet*. Кадр має фіксований формат і разом з полем даних містить різну службову інформацію, наприклад, адресу одержувача і адресу відправника.

Мережа *Ethernet* влаштована так, що при попаданні кадру в середовище передачі даних, що розділяється, всі мережні адаптери починають одночасно приймати цей кадр. Всі вони аналізують адресу призначення, розташовану в одному з початкових полів кадру, і, якщо ця адреса співпадає з їх власною, кадр поміщається у внутрішній буфер мережного адаптера. Таким чином комп'ютер-адресат отримує призначені йому дані.

Може виникнути ситуація, коли декілька комп'ютерів одночасно вирішують, що мережа вільна, і починають передавати інформацію. Така ситуація, називається колізією, перешкоджає правильній передачі даних по мережі. У стандарті *Ethernet* передбачений алгоритм виявлення і коректної обробки колізій. Вірогідність виникнення колізії залежить від інтенсивності мережного трафіку. Після виявлення колізії мережні адаптери, які намагалися передати свої кадри, припиняють передачу і після паузи випадкової тривалості намагаються знову дістати доступ до середовища і передати той кадр, який викликав колізію.

Основні переваги технології *Ethernet*

Головною перевагою мереж *Ethernet*, завдяки якій вони стали такими популярними, є їх економічність. Для побудови мережі досить мати один мережний адаптер для кожного комп'ютера плюс один фізичний сегмент коаксіального кабелю потрібної довжини.

Крім того, в мережах *Ethernet* реалізовані достатньо прості алгоритми доступу до середовища, адресації і передачі даних. Простота логіки роботи мережі веде до спрощення і, відповідно, зниження вартості мережних адаптерів і їх драйверів. З тієї ж причини адаптери мережі *Ethernet* володіють високою надійністю. І, нарешті, ще однією чудовою властивістю мереж *Ethernet* є їх хороша розширюваність, тобто можливість підключення нових вузлів.

Інші базові мережні технології, такі як *Token Ring* і *FDDI*, хоч і володіють індивідуальними рисами, в той же час мають багато загального з *Ethernet*. В першу чергу, це застосування регулярних фіксованих топологій ("ієрархічна зірка" і "кільце"), а також середовищ передачі даних, що розділяються. Істотною відмінністю однієї технології від іншої пов'язані з особливостями використовуваного методу доступу до середовища, яке розділяється. Так, відмінності технології *Ethernet* від технології *Token Ring* багато в чому визначаються специфікою закладених в них методів розділення середовища: випадкового алгоритму доступу в *Ethernet* і методів доступу шляхом передачі маркера в *Token Ring*.

1.4.7. Дейтаграмна передача

У мережах з комутацією пакетів сьогодні застосовується два класи механізмів передачі пакетів:

- дейтаграмна передача;
- віртуальні канали.

Прикладами мереж, що реалізують дейтаграмний механізм передачі, є мережі *Ethernet*, *IP* і *IPX*. За допомогою віртуальних каналів передають дані мережі *X.25*, *Frame relay* і *ATM*. Спочатку розглянемо базові принципи дейтаграмного підходу.

Дейтаграмний спосіб передачі даних заснований на тому, що всі передані пакети обробляються незалежно один від одного, пакет за пакетом. Приналежність пакету до певного потоку між двома кінцевими вузлами і двома програмами, які працюють на цих вузлах, ніяк не враховується.

Вибір наступного вузла, наприклад, комутатора *Ethernet* або маршрутизатора *IP/IPX* відбувається тільки на підставі адреси вузла призначення, що міститься в заголовку пакету. Рішення про те, якому вузлу передати пакет, що прийшов, приймається на основі таблиці, яка містить набір адрес призначення і адресну інформацію, що однозначно визначає наступний (транзитний або кінцевий) вузол. Такі таблиці мають різні назви, наприклад, для мереж *Ethernet* вони зазвичай називаються таблицею просування (*forwarding table*), а для мережних протоколів, таких як *IP* і *IPX*, таблицями маршрутизації (*routing table*). Далі для простоти користуватимемося терміном "таблиця маршрутизації" як узагальненої назви такого роду таблиць, використовуваних для дейтаграмної передачі на підставі тільки адреси призначення кінцевого вузла.

У таблиці маршрутизації для однієї і тієї ж адреси призначення може міститися декілька записів, які вказують, відповідно, на різні адреси наступного маршрутизатора. Такий підхід використовується для підвищення продуктивності і надійності мережі. У прикладі на рис. 1.48 пакети, що поступають в маршрутизатор R1 для вузла призначення з адресою N2, A2, в цілях балансу навантаження розподіляються між двома наступними маршрутизаторами – R2 і R3, що знижує навантаження на кожний з них, а значить, зменшує черги і прискорює доставку. Деяка "розмитість" шляхів проходження пакетів з однією і тією ж адресою призначення через мережу є прямим наслідком принципу незалежної обробки кожного пакету, властивого дейтаграмним протоколам. Пакети, по одному і по тій же адресі призначення, можуть добиратися до нього різними шляхами унаслідок зміни стану мережі, наприклад, відмови проміжних маршрутизаторів.

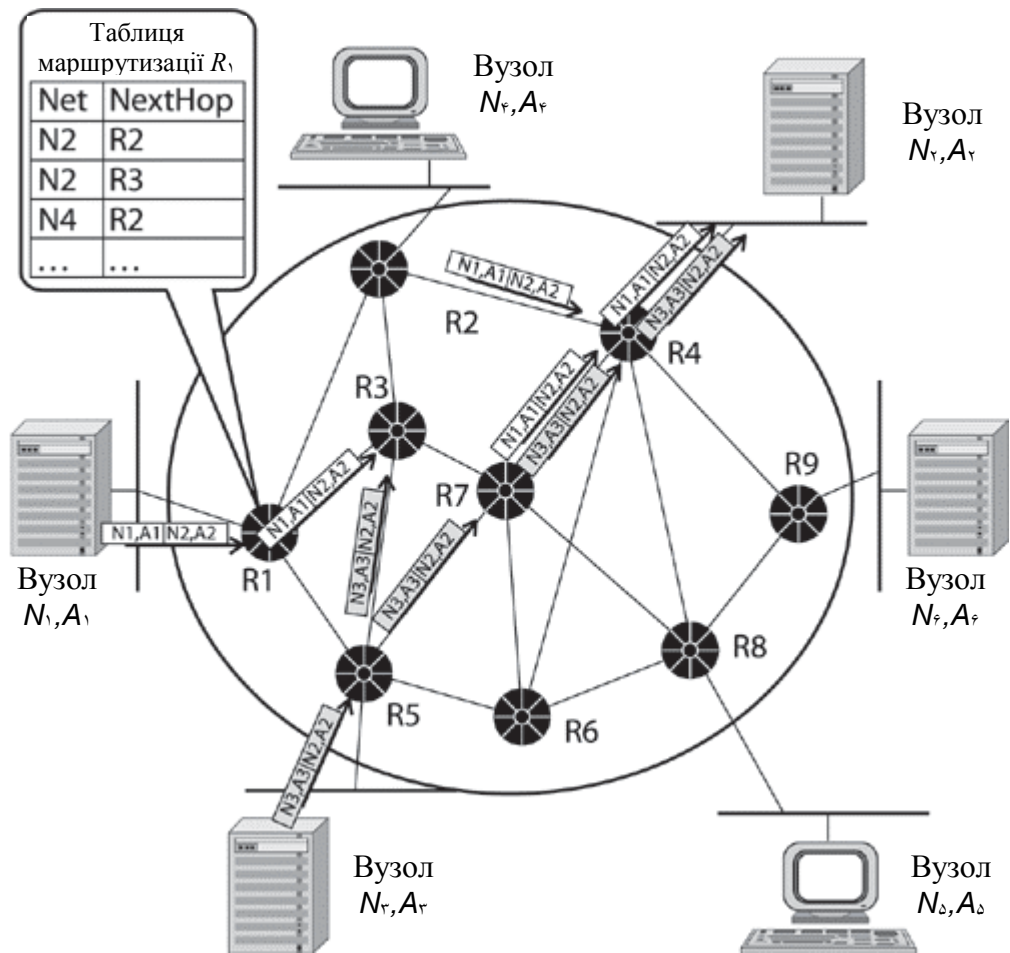


Рис. 1.48. Дейтаграмний принцип передачі пакетів.

Така особливість дейтаграмного механізму як розмитість шляхів проходження трафіку через мережу також в деяких випадках є недоліком. Наприклад, якщо пакетам певної сесії між двома кінцевими вузлами мережі необхідно забезпечити задану якість обслуговування. Сучасні методи підтримки *QoS* працюють ефективніше, коли трафік, якому потрібно забезпечити гарантії обслуговування, завжди проходить через одні і ті ж проміжні вузли.

1.4.8. Віртуальні канали в мережах з комутацією пакетів

Механізм віртуальних каналів (virtual circuit або virtual channel) створює в мережі стійкі шляхи проходження трафіку через мережу з комутацією пакетів. Цей механізм враховує існування в мережі потоків даних.

Якщо метою є прокладка для потоку всіх пакетів єдиного шляху через мережу, то необхідною (але не завжди єдиною) ознакою такого потоку повинна бути наявність для всіх його пакетів загальних точок входу і виходу з мережі. Саме для передачі таких потоків в мережі створюються віртуальні канали. На рис. 1.49 показаний фрагмент мережі, в якій прокладено два віртуальні канали. Перший проходить від кінцевого вузла з адресою N₁, A₁ до кінцевого вузла з адресою N₂, A₂ через проміжні комутатори мережі R₁, R₃, R₇ і R₄. Другий забезпечує просування даних по дорозі N₃, A₃ – R₅ – R₇ – R₄ – N₂, A₂. Між двома кінцевими вузлами може бути прокладені декілька віртуальних каналів, як повністю співпадаючих відносно шляху проходження через транзитні вузли, так і таких, що відрізняються.

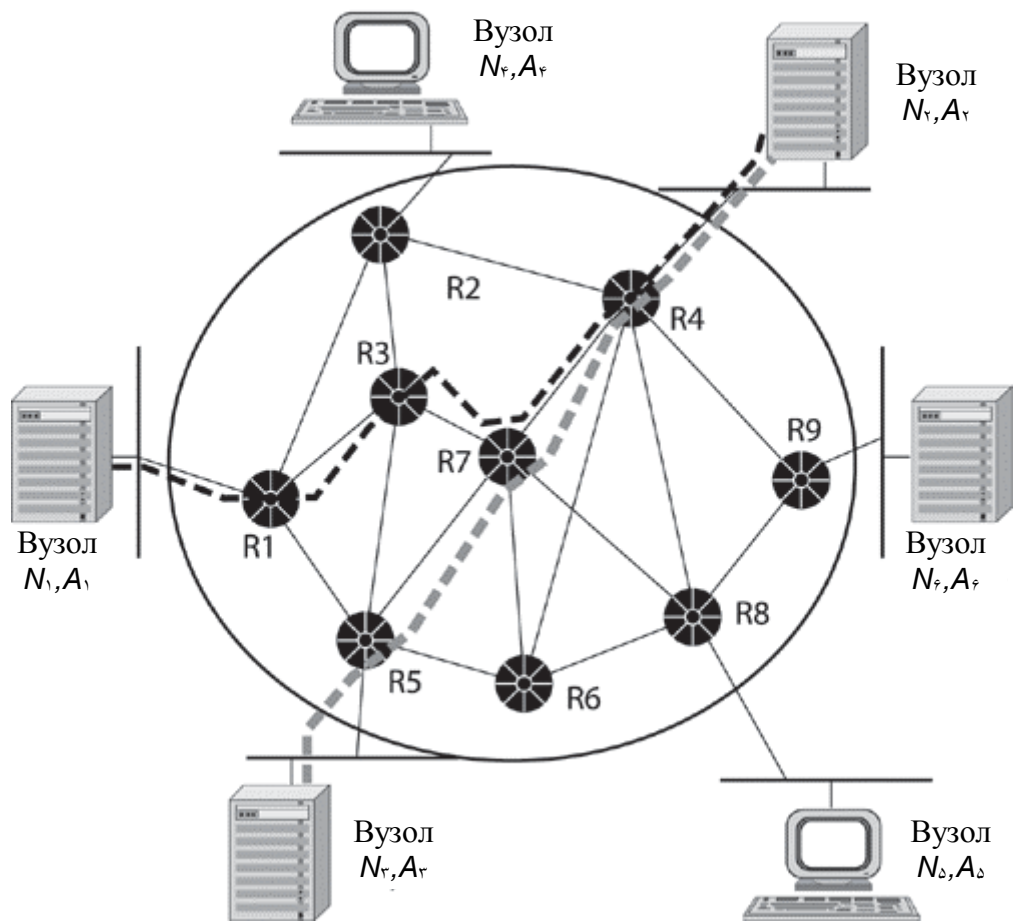


Рис. 1.49. Принцип роботи віртуального каналу.

Мережа тільки забезпечує можливість передачі трафіку уздовж віртуального каналу, а які саме потоки передаватимуться по цих каналах, вирішують самі кінцеві вузли. Вузол може використовувати один і той же віртуальний канал для передачі всіх потоків, які мають загальні з даним віртуальним каналом кінцеві точки, або ж тільки частини з них. Наприклад, для потоку реального часу можна використовувати один віртуальний канал, а для трафіку електронної пошти – інший. У останньому випадку різні віртуальні канали пред'являтимуть різні вимоги до якості обслуговування, і задовольнити їх буде простіше, ніж у тому випадку, коли поодинокі віртуальному каналу передається трафік з різними вимогами до параметрів *QoS*.

Важливою особливістю мереж з віртуальними каналами є використання локальних адрес пакетів при ухваленні рішення про передачу. Замість достатньо довгої адреси вузла призначення (його довжина повинна дозволяти унікально ідентифікувати всі вузли і підмережі в мережі, наприклад, технологія *ATM* оперує адресами завдовжки в 20 байт) застосовується локальна, тобто змінна від вузла до вузла, мітка, якою позначаються всі пакети, переміщувані по певному віртуальному каналу. Ця мітка в різних технологіях називається по-різному: у технології *X.25* – номер логічного каналу (Logical Channel number, *LCN*), в технології *Frame relay* – ідентифікатор з'єднання рівня каналу даних (Data Link Connection Identifier, *DLCI*), в технології *ATM* – ідентифікатор віртуального каналу (Virtual Channel Identifier, *VCI*). Проте призначення її скрізь однаково – проміжний вузол, званий в цих технологіях комутатором, читає значення мітки із заголовка пакету, що прийшов, і проглядає свою таблицю комутації, в якій указується, на який вихідний порт потрібно передати пакет.

Таблиця комутації містить записи тільки про ті віртуальні канали, які проходять через даний комутатор, а не про всі наявні в мережі вузли (або підмережах, якщо застосовується ієрархічний спосіб адресації). Зазвичай в крупній мережі кількість прокладених через вузол віртуальних каналів істотно менше кількості вузлів і підмереж, тому за розмірами таблиця комутації набагато менше таблиці маршрутизації, а, отже, перегляд займає значно менше часу і не вимагає від комутатора великої обчислювальної потужності. Ідентифікатор віртуального каналу також набагато коротший за адресу кінцевого вузла (з тієї ж причини), тому і надлишковість заголовка пакету, який тепер не містить довгої адреси, а переносить по мережі тільки ідентифікатор, істотно менший.

1.5. Практичні відомості

Приклад розрахунку швидкості передачі даних

Для якісної передачі голосу в методі **ІКМ** використовується частота квантування амплітуди звукових коливань в 8000 Гц. Це пов'язано з тим, що в аналоговій телефонії для передачі голосу був вибраний діапазон від 300 до 3400 Гц, який **достатньо** якісно передає всі основні **гармоніки** спектру людської **мови**. У методі **ІКМ** звичайно використовується 7 або 8 біт коду для представлення амплітуди одного виміру. Відповідно це дає 127 або 256 градацій звукового сигналу, що **виявляється цілком** достатнім для якісної передачі голосу.

При використанні методу **ІКМ** для передачі одного голосового каналу необхідна пропускна спроможність 56 або 64 Кбіт/с залежно від того, якою кількістю біт представляється кожен вимір. Якщо для цих цілей використовується 7 біт, то при частоті передачі вимірів в 8000 Гц **одержуємо**:

$$8000 * 7 = 56000 \text{ біт/с або } 56 \text{ Кбіт/с};$$

а для випадку 8-ми бітів:

$$8000 * 8 = 64000 \text{ біт/с або } 64 \text{ Кбіт/с}.$$

Стандартним є цифровий канал 64 Кбіт/с, який також називається елементарним каналом цифрових телефонних **мереж**.

Приклад вибору кабелю для лінії передачі

Для передачі даних з необхідною достовірністю на відстань 2000 м необхідно, щоб **загасання** в кабелі не перевищувало 0,5 **дБ**. Виберіть тип кабелю і довжину хвилі передачі (рис. 1.50). Охарактеризуйте режим **розповсюдження** променя в кабелі (одномодовий або багатомодовий).



Рис.1.50. Залежність загасання від довжини хвилі.

Для передавання інформації використовують світло з довжиною хвилі 1550 нм (1,55 мкм), 1300 нм (1,3 мкм) та 850 нм (0,85 мкм). Світлодіоди можуть випромінювати світло довжиною хвилі 1300 нм та 850 нм. Діодні випромінювачі з довжиною хвилі 850 нм суттєво дешевше, ніж випромінювачі з довжиною хвилі 1300 нм, але смуга пропускання кабелю для хвиль довжиною 850 нм складає всього 200 МГц/км, у той час як на хвилі 1300 нм вона складає 500 МГц/км.

Лазерні випромінювачі працюють на хвилях 1550 нм та 1300 нм. З використанням лазерних випромінювачів можна модулювати світловий потік сигналами з шириною спектру 10 ГГц і навіть вище. Завдяки когерентності світлового потоку лазерних випромінювачів втрати в оптоволоконних кабелях стають менше, ніж при використанні некогерентного світлового потоку діодних випромінювачів.

Таким чином, вибір довжини хвилі, типу режиму розповсюдження променя в кабелі (одно- чи багатомодовий) залежить від багатьох факторів. Деякі з них є суперечними, але для наведеного прикладу протиріччя не мають місця. Основним фактором є граничне загасання сигналу. Судячи по графіку, його можна забезпечити на хвилях від 1400 нм до 1600 нм. Тому відповідь – однозначна: у вікні прозорості 1400 – 1600 нм вибираємо хвилю довжиною 1550 нм, випромінювання якої може забезпечити лазерний випромінювач. Логічно при цьому використовувати одномодовий кабель, в якому менше спотворення форми імпульсів, що розповсюджуються (дисперсія сигналу), та менше загасання завдяки когерентності випромінюваних коливань.

1.6. Контрольні запитання

1. Які нові можливості з'являються при використанні обчислювальних мереж?
2. Особливості взаємодії комп'ютерів в мережі і взаємодії комп'ютера з периферійним пристроєм.
3. Призначення і функції мережного адаптера і його драйвера.
4. Типи топологій мереж. Переваги та недоліки різних топологій.
5. Принципи побудови і функціональне призначення повторювачів, мостів, концентраторів, комутаторів, маршрутизаторів.
6. Логічна і фізична структуризація мережі.
7. Які з наступних тверджень вірні завжди?
 - a. Кожен порт моста/комутатора має MAC - адреса.
 - b. Кожен міст/комутатор має мережну адресу.
 - c. Кожен порт моста/комутатора має мережну адресу.
 - d. Кожен маршрутизатор має мережну адресу.
 - e. Кожен порт маршрутизатора має MAC - адреса.
 - f. Кожен порт маршрутизатора має мережну адресу.
8. Принципи побудови відкритих систем. Приклади закритих систем.
9. Еталонна модель OSI і стік OSI. Рівні взаємодії, протоколи, інтерфейси.
10. Функції різних рівнів і стандартні стеки протоколів TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA, OSI для кожного рівня моделі OSI.
11. Чи є терміни «специфікація» і «стандарт» синонімами?
12. Основні стандарти і принципи побудови мереж Ethernet.
13. Основні стандарти і принципи побудови мереж Token Ring.
14. Основні стандарти і принципи побудови мереж FDDI.
15. Основні різновиди комп'ютерних мереж, їх особливості.
16. У чому полягає відмінність локальних мереж від глобальних на рівні служб? На рівні транспортної системи?
17. Характеристики продуктивності мережі.
18. Дати визначення наступних характеристик мережі:
 - availability;
 - fault tolerance;
 - security;
 - extensibility;
 - scalability;
 - transparency.
19. Вимоги, що пред'являються до обчислювальних мереж.
20. Основні характеристики ліній зв'язку.
21. Методи кодування. Основні фізичні і логічні коди, їх достоїнства і недоліки.
22. Поясніть, з яких міркувань вибирається пропускна спроможність елементарного каналу цифрових телефонних мереж.
23. Порівняльна ефективність способів комутації.
24. Характеристики віртуальних каналів в мережах з комутацією пакетів.
25. Характеристики складових каналів в мережах з комутацією каналів.
26. Технології Ethernet: метод доступу CSMA/CD. Поняття про колізії.
27. Технології Ethernet: етапи доступу до середовища.
28. Технології Ethernet: час подвійного оберту і розпізнавання колізій.
29. Поясніть різницю між розширюваністю і масштабованістю на прикладі технології Ethernet.
30. Що таке домен колізій? Чи є доменами колізій фрагменти мережі, показані на рис. 1.51?

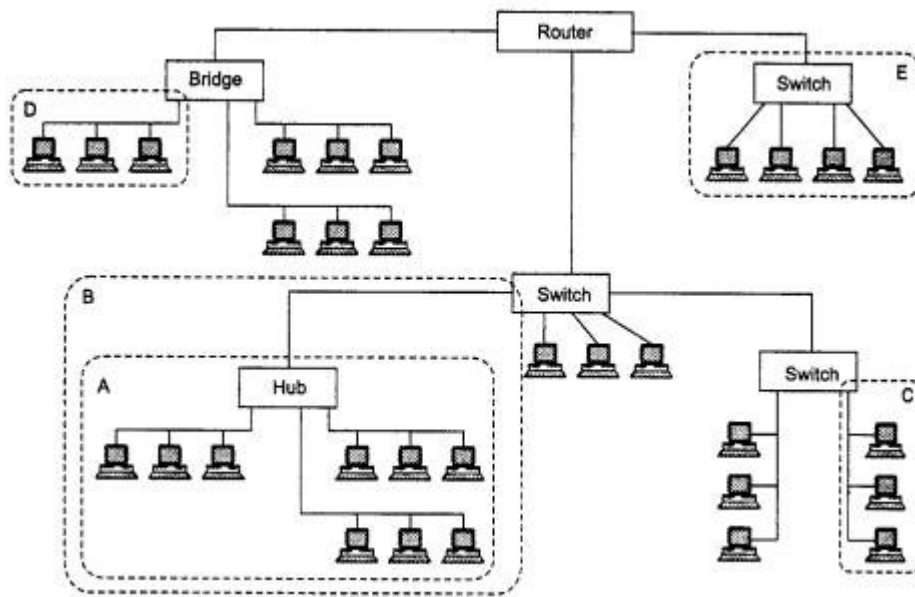


Рис1.51. Домени колізій.

31. Чому дорівнюють значення **наступних** характеристик стандарту 10Base-5:
 - a. номінальна пропускна спроможність (біт/с);
 - b. ефективна пропускна спроможність (біт/с);
 - c. пропускна спроможність (кадр/с);
 - d. внутрішньопакетна швидкість передачі (біт/с);
 - e. міжбітовий інтервал (с).
32. Чим пояснюється, що мінімальний розмір кадру в стандарті 10Base-5 був вибраний рівним 64 байт?
33. Поясніть **значення** кожного поля кадру Ethernet. Як величина MTU впливає на роботу **мережі**? Які проблеми несуть **дуже** довгі кадри? Чому короткі кадри не ефективні?
34. Як коефіцієнт **використовування** впливає на продуктивність **мережі** Ethernet?
35. Якщо один варіант технології Ethernet має вищу швидкість передачі даних, ніж **інший** (наприклад, Fast Ethernet і Ethernet), то який **з** них підтримує велику максимальну довжину **мережі**?
36. З яких міркувань вибрана максимальна довжина фізичного сегменту в стандартах Ethernet?
37. Перевірте коректність конфігурації **мережі** Fast Ethernet, приведеної на рис. 1.52.

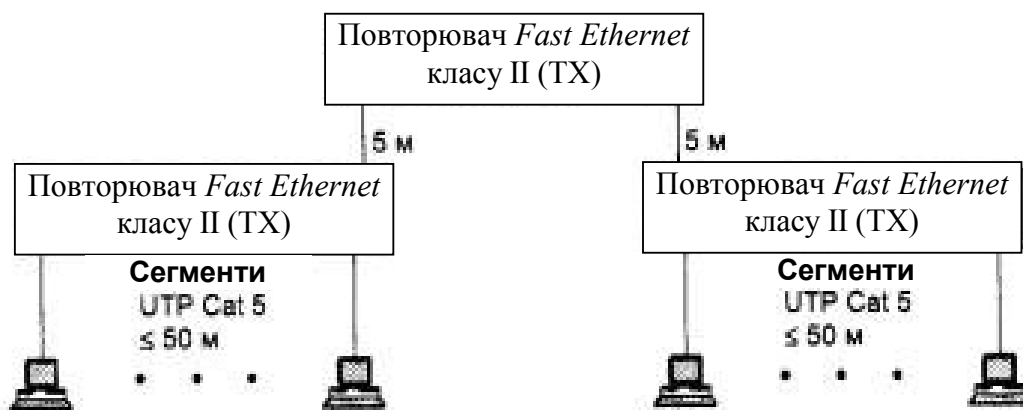


Рис. 1.52. Приклад конфігурації мережі.

38. Вкажіть максимально допустимі значення MTU для:
 - Ethernet;
 - Token Ring;
 - FDDI;

– АТМ.

39. Опишіть алгоритм доступу до середовища технології Token Ring. З яких міркувань вибирається максимальний час обертку маркера по кільцю?

1.7. Задачі і вправи

1. Якщо всі комунікаційні пристрої в приведеному нижче фрагменті мережі (рис.1.53) є концентраторами, то на яких портах з'явиться кадр, якщо його відправив комп'ютер А комп'ютеру В? Комп'ютеру 3? Комп'ютеру D?

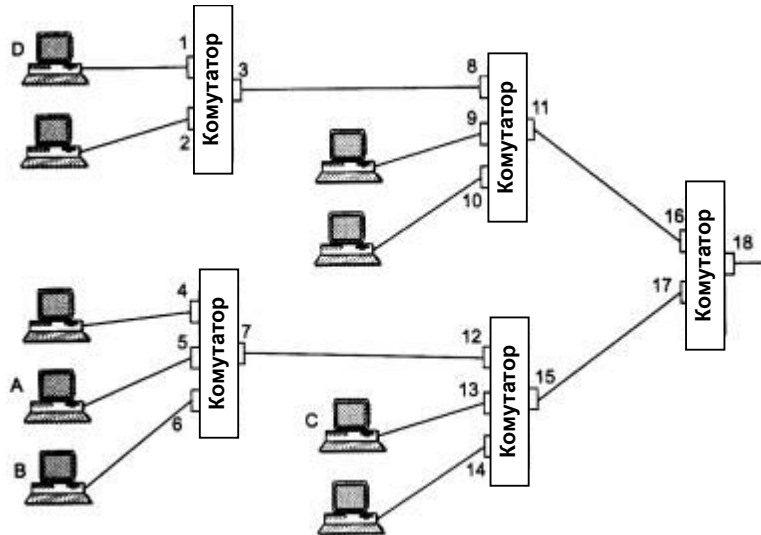


Рис. 1.53. Фрагмент мережі.

2. Якщо всі комунікаційні пристрої в приведеному нижче фрагменті мережі (рис. 1.54) є комутаторами, то на яких портах з'явиться кадр, якщо його відправив комп'ютер А комп'ютеру В? Комп'ютеру С? Комп'ютеру D?

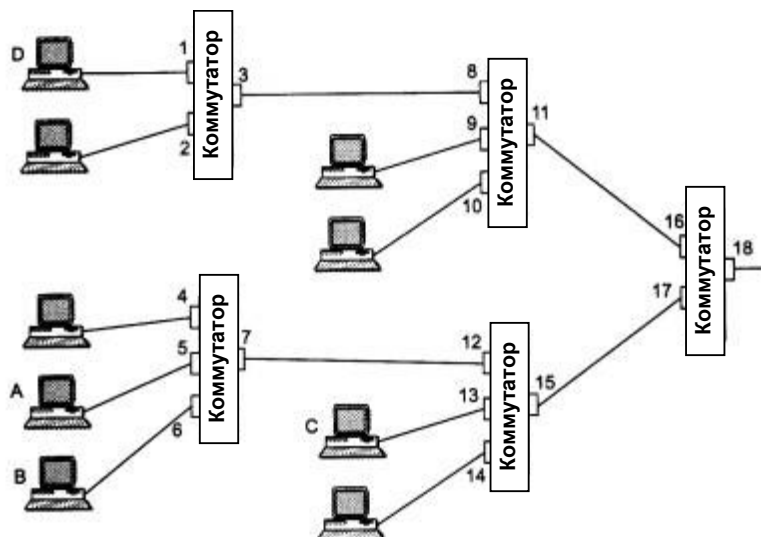


Рис. 1.54. Фрагмент мережі.

3. Якою буде теоретична межа швидкості передачі даних в бітах в секунду по каналу з шириною смуги пропускання в 20 КГц, якщо потужність передавача складає 0,01 мВт, а потужність шуму в каналі рівна 0,0001 мВт?

4. Якою буде теоретична межа швидкості передачі даних в бітах в секунду по каналу з шириною смуги пропускання в 20 КГц, якщо потужність передавача складає 0,01 мВт, а потужність шуму в каналі рівна 0,0001 мВт?

5. Визначте пропускну спроможність каналу зв'язку для кожного з **напрямів** дуплексного режиму, якщо відомо, що його смуга пропускання рівна 600 КГц, а в методі кодування використовується 10 станів сигналу.
6. Розрахуйте затримку **розповсюдження** сигналу і затримку передачі даних для випадку передачі пакету в 128 байт:
- по кабелю витой пари завдовжки в 100 м при швидкості передачі 100 Мбіт/с;
 - коаксіальному кабелю завдовжки в 2 км при швидкості передачі в 10 Мбіт/с;
 - супутниковому **геостаціонарному** каналу протяжністю в 72 000 км при швидкості передачі 128 Кбіт/с.
- Вважайте швидкість **розповсюдження** сигналу:
- по кабелю дорівнює 200000 км/с;
 - по супутниковому каналу дорівнює швидкості світла у вакуумі (300000 км/с).
7. **Мережа з** комутацією пакетів **знає** переваження. Для усунення цієї ситуації розмір вікна в протоколах комп'ютерів **мережі** потрібно збільшити або зменшити? Чому?

1.4. Список літератури для модуля 1

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2001. – 672 с.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов, 2-е изд. – СПб.: Питер, 2003. – 864 с.
3. Електронна версія цих же підручників: <http://kafvt.narod.ru/Osia/frameset.htm>
4. Вишневикий В.М. Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003. – 512 с.
5. Столлингс В. Современные компьютерные сети. 2-е издание. – СПб.: Питер, 2003. – 783с.

Модуль 2

2. Теоретичні відомості

2.1. Прикладне мережне програмне забезпечення

2.1.1. Перспективи розвитку розподіленої обробки даних

Серед основних напрямків розвитку інформаційних технологій необхідно відмітити наступні тенденції розробки прикладних програм широкого призначення: об'єктно-орієнтований підхід, створення візуальних засобів розробки, підтримка різноманітних систем керування базами даних (СКБД), використання автоматизованих засобів для проектування даних – Computer-Aided System Engineering (CASE), підтримка програмного забезпечення класу оперативного аналізу даних – On-Line Analytical Processing (OLAP) для систем прийняття рішень.

Розвиток і запровадження в практику об'єктно-орієнтованого підходу в цілому (об'єктно-орієнтоване програмування і проектування програмних систем, об'єктно-орієнтовані технології організації інтерфейсів користувача, виникнення розподілених об'єктних систем і т.п.) зумовило виникнення об'єктно-орієнтованих СКБД (ОО СКБД).

Програмний продукт, в тому числі і клієнтська прикладна програма, створюється вже на етапі розробки. Помилки при проектуванні даних приводять як мінімум до написання коду більшого об'єму, а як максимум до неможливості виконати запит. Вартість її виправлення дуже висока. Тому важливо, щоб ця частина роботи над проектом була виконана максимально якісно. Саме важливість цього етапу зумовила стрімкий ріст популярності такої категорії програмного забезпечення, як автоматизовані засоби проектування даних – CASE-засоби.

Оперативний аналіз даних OLAP – інформаційна технологія, яка забезпечує аналітикам, управляючим, керівникам можливість вивчати великі об'єми взаємозв'язаних даних за допомогою швидкого інтерактивного їх відображення на різних рівнях деталізації з різних точок зору у відповідності з уявленнями кінцевого користувача про сферу підприємницької діяльності. Засоби OLAP становлять невід'ємну частину сучасних корпоративних систем підтримки прийняття рішень. Цей напрямок індустрії створення програмного забезпечення є одним з тих, який найбільш динамічно розвивається. Більшість провідних виробників програмного забезпечення, включаючи Arbor Software, IBM, Informix, Oracle, Microsoft, Sybase, пропонують OLAP-системи в даному секторі ринку.

2.1.2. Загальна характеристика стану ринку СКБД

Сучасний ринок систем управління базами даних можна розділити на дві великі групи. Перша група – це так звані настільні, а друга група – серверні СКБД [1],[2].

Настільні СКБД не містять спеціальних прикладних програм і сервісів, які керують даними, взаємодія з ними здійснюється за допомогою файлових сервісів операційної системи. Нерідко подібні СКБД мають у своєму складі і засоби розробки, орієнтовані на роботу з даними формату, характерного для цієї СКБД, які

дозволяють створити більш комфортний інтерфейс користувача. Що ж стосується обробки даних, то вона повністю здійснюється в прикладній програмі користувача (клієнтській прикладній програмі).

У даний момент багато настільних СКБД мають мережні версії, які дозволяють обробляти дані, що знаходяться в загальнодоступному сховищі (наприклад, на мережному диску) декільком користувачам одночасно. Від чисто настільних СКБД версії для багатьох користувачів відрізняються наявністю механізму блокування частин файлів даних (які містять одну або декілька записів таблиці), що дозволяє звертатися до того самого файлу декільком користувачам одночасно.

Недоліки подібних СКБД не очевидні і стають помітні, як правило, при рості обсягів даних, які зберігаються, і збільшенні кількості користувачів. Звичайно недоліки виявляються в зниженні продуктивності й у виникненні збоїв при роботі після деякого часу використання клієнтських прикладних програм. Причини подібних проблем криються, в основному, в принципі роботи таких СКБД і заснованих на них інформаційних системах (ІС). Вони полягають в опрацюванні даних усередині клієнтської прикладної програми.

Інша проблема настільних СКБД полягає в можливості порушення цілісності даних тому, що єдиним механізмом, що контролює її, є програма користувача. Тому всі клієнтські програми повинні містити відповідний код, і доступ до файлів бази даних (БД) із будь-яких інших програм повинний бути заборонений. На сьогоднішній день відомо більш двох десятків форматів даних настільних СКБД. Найбільш популярними, виходячи з числа проданих копій, є *Paradox*, *FoxPro*, *Access* і *Microsoft Data Engine*, "полегшена" версія *Microsoft SQL Server*.

СКБД архітектури клієнт-сервер реалізовані для декількох платформ; мають зручні адміністративні утиліти; дозволяють здійснювати резервне копіювання даних; підтримують кілька сценаріїв реплікацій; підтримують *OLAP* і створення сховищ даних; підтримують виконання розподілених запитів і транзакцій; дозволяють використовувати різні засоби проектування даних для створення своїх об'єктів; підтримують засоби розробки і генератори звітів як власного виробництва, так і інших виробників; підтримують як мінімум публікацію даних в *Internet*. На сьогоднішній день відомо більш двох десятків серверних СКБД. Найбільш популярними, виходячи з числа продажів і інсталяцій, є *Oracle*, *Microsoft SQL Server*, *Informix*, *Sybase*, *DB2*.

Можливості найбільш популярних серверних СКБД відбивають сучасні тенденції розвитку ІС, такі як використання багатопроцесорних систем і розподіленої обробки даних, створення розподілених систем, у тому числі з використанням технологій *Internet*, застосування засобів швидкої розробки прикладних програм, створення систем підтримки прийняття рішень з використанням аналітичної обробки даних, а також вимоги до надійності ІС.

2.1.3. Засоби розробки прикладних програм

Процес проектування даних для реляційних СКБД є у відомій мірі процесом логічним і підкоряється єдиній стандартній методології. Це обумовлює низький ступінь залежності послідовності виконуваних при проектуванні даних дій від того, який саме засіб проектування даних застосовується, чи застосовується він взагалі. Саме тому засоби проектування даних у більшому або меншому ступені за своїм інтерфейсом подібні один одному.

Засоби розробки прикладних програм як категорія програмних продуктів існують набагато довше, ніж засоби проектування даних, і вони більш різноманітні. Класифікувати засоби розробки можна з різних позицій, наприклад, виходячи з підтримуваної ними мови програмування, або працездатності створених прикладних програм на тій або іншій платформі, або наявності в них тих або інших бібліотек і візуальних засобів. Практично будь-який засіб розробки, що претендує на універсальність, можна змусити працювати з будь-якою БД: досить підтримки застосування в цьому засобі розробки сторонніх бібліотек і наявності в цій базі даних набору клієнтських інтерфейсів – *Application Programming Interface (API)* для платформи, на якій повинні функціонувати створені прикладні програми. Однак далеко не будь-яка пара продуктів "засіб розробки плюс СКБД" приваблива за трудовитратами, які пов'язані зі створенням подібних прикладних програм. Можна написати повноцінну прикладну програму за допомогою мови C та відповідних бібліотек, але витрати не будуть виправданими, тому що вже існують потужні засоби, орієнтовані на створення прикладних програм саме для роботи з БД.

Виділимо засоби розробки, орієнтовані на конкретні СКБД. Наприклад, *Oracle Forms Developer* призначений для створення прикладних програм, які надають інтерфейс користувача до СКБД *Oracle*. Майже усі виробники серверних СКБД виробляють засоби розробки прикладних програм. У переважній більшості випадків сучасні версії цих засобів розробки підтримують доступ до СКБД інших виробників як мінімум за допомогою одного з універсальних механізмів доступу до даних – *Open DataBase Connectivity (ODBC)*, *Object Linking and Embedding DataBase (OLE DB)*, *Borland DataBase Engine (BDE)* [3]. В окрему категорію можна виділити середовища розробки настільних СКБД. Як відомо, переважна більшість настільних СКБД, таких як *Microsoft Visual FoxPro*, *Microsoft Access*, *Corel Paradox*, *Visual dBase* підтримують доступ до серверних СКБД, як мінімум, за допомогою універсальних механізмів доступу до даних, що дозволяє умовно віднести їх і до категорії засобів розробки. Відмітимо, що створення прикладних програм в архітектурі клієнт-сервер з їх допомогою – явище рідкісне.

Наступна категорія – засоби розробки, універсальні стосовно СКБД, як правило, є послідовниками звичайних засобів розробки прикладних програм, які не мають прямого відношення до БД. Типові приклади таких засобів розробки – *Borland Pascal*, *Borland C++*, *Microsoft Quick*. Здатні використовувати бібліотеки сторонніх виробників, ці засоби дозволяють звертатися до функцій клієнтських *API*, а з розвитком універсальних механізмів доступу до даних (таких як *ODBC*) і до функцій *API* бібліотек, що реалізують такі механізми. Відзначимо, що нерідко за допомогою цих засобів розробки створювалися середовища настільних СКБД (таких як *dBase*, *FoxBase*) або псевдокомпілятори для мов сімейства *xBase* (наприклад, *Clipper*). Більш пізні версії зазначених засобів розробки придбали бібліотеки функцій і класів, призначених для доступу до даних за допомогою тих або інших універсальних механізмів. Подальший розвиток засобів розробки привів до появи двох категорій продуктів подібного призначення.

До першої категорії відносяться засоби розробки, що володіють великими бібліотеками класів, великою кількістю "майстрів" і кодогенераторів, але орієнтовані на "ручне" створення коду і досить рідко застосовуються для створення "стандартних" прикладних програм для роботи з БД. Тут під словосполученням "стандартна прикладна програма" мається на увазі прикладна програма, яка має безпосередній доступ до БД, з якою взаємодіє користувач, який є "класичним" клієнтом серверної СКБД. Типовим (і єдиним дійсно популярним на ринку програмного забезпечення) представником цього класу продуктів є *Microsoft Visual C++*. За допомогою *Microsoft Visual C++* і бібліотеки *MFC (Microsoft Foundation Classes)* можна створювати будь-які прикладні програми, якщо мати знання, уміння, навичку і час. Проте, прикладні програми, які мають складний інтерфейс користувача (наприклад, що використовують БД), з його допомогою розробляють не так часто. В основному цей продукт застосовується для створення клієнтських прикладних програм у випадку пред'явлення до них особливих вимог, таких, наприклад, як висока продуктивність, здатність здійснювати які-небудь нестандартні операції й т.п.

До другої категорії відносяться засоби розробки з розвитими візуальними інструментами, що дозволяють буквально "малювати" інтерфейс користувача, частково стираючи розходження між роботою програміста і користувача й зменшуючи вартість кінцевого продукту. Саме ця категорія засобів розробки найбільше часто застосовується при створенні клієнтських прикладних програм. До найбільш популярних продуктів подібного класу варто віднести [Microsoft Visual Basic](#), [Borland Delphi](#), [Sybase PowerBuilder](#) і [Borland C++ Builder](#).

Останнім часом дуже популярним стало створення *Web*-прикладних програм. При написанні таких модулів можуть використовуватись інтерпретовані середовища розробки – мови *Perl* і *Java*. Мова *Perl* пристосована для обробки великих текстів і дозволяє здійснювати доступ до різних БД. Це обумовлене насамперед тим, що *Perl* надає розроблювачам прості і зручні засоби обробки довільних текстів і взаємодії з БД і технологією *Web*. *Java*-технологія створена з метою використання на різних платформах, має всі вище згадані можливості і найбільш стандартні засоби для легшої адаптації під конкретне середовище.

Відзначимо також, що, крім перерахованих вище "універсальних" засобів створення як прикладних програм в архітектурі клієнт-сервер, так і бізнес-об'єктів для розподілених систем, на ринку засобів розробки є і спеціалізовані засоби, призначені саме для створення бізнес-об'єктів (як правило, *Web*-прикладних програм). З засобів розробки такого класу для платформи *Windows* найбільш популярний *Microsoft Visual InterDev*.

2.2. Системи керування базами даних

2.2.1. Технологія файл-сервер і клієнт-сервер

Розпізнають централізовані та розподілені системи БД [4]. Централізована БД зберігається в пам'яті однієї обчислювальної системи, тобто розміщена на одному комп'ютері. Розподілена БД складається з декількох частин, розміщених на різних машинах обчислювальної мережі.

Системи централізованих БД поділяються в залежності від архітектури. Системи централізованих БД, в яких СКБД і прикладна програма розміщені на одному комп'ютері і мережна підтримка не потрібна, є застарілою. Розглянемо архітектури файл-сервер і клієнт-сервер. Цими термінами визначають перш за все архітектуру або логічне розподілення функцій. Клієнт – це прикладна програма, яку також називають прикладною програмою переднього плану (*frontend*), а сервер – це прикладна програма заднього плану (*backend*).

В архітектурі файл-сервер БД розміщується на файл-сервері (або на декількох серверах). В якості останнього може використовуватись більш потужна машина із усіх, об'єднаних в мережу. Функції файл-сервера полягають в основному в збереженні БД та забезпеченні доступу до них користувачів, які працюють на різних комп'ютерах. Файли БД у відповідності з запитом користувачів передаються на робочі станції, де в основному і виконується обробка. Передані дані оброблюються СКБД, яка знаходиться на комп'ютерах користувачів. Після того як користувачі виконають необхідні зміни, вони копіюють файли знову на файл-сервер, де інші користувачі, в свою чергу, можуть знову їх використовувати. Явним недоліком подібного підходу є висока ймовірність втрати змін, виконаних одними користувачами, при записуванні змінених файлів на центральний сервер.

За технологією клієнт-сервер (рис. 2.1) припускається, що, крім зберігання БД, центральний комп'ютер (сервер БД) повинен забезпечувати виконання основного об'єму обробки даних. При технології клієнт-

сервер запит на виконання операції з даними (наприклад, звичайний вибір), який видає клієнт (робоча станція), примушує сервер виконувати пошук і вилучення даних. Отримані дані, але не файли, транспортуються по мережі від серверу до клієнта.

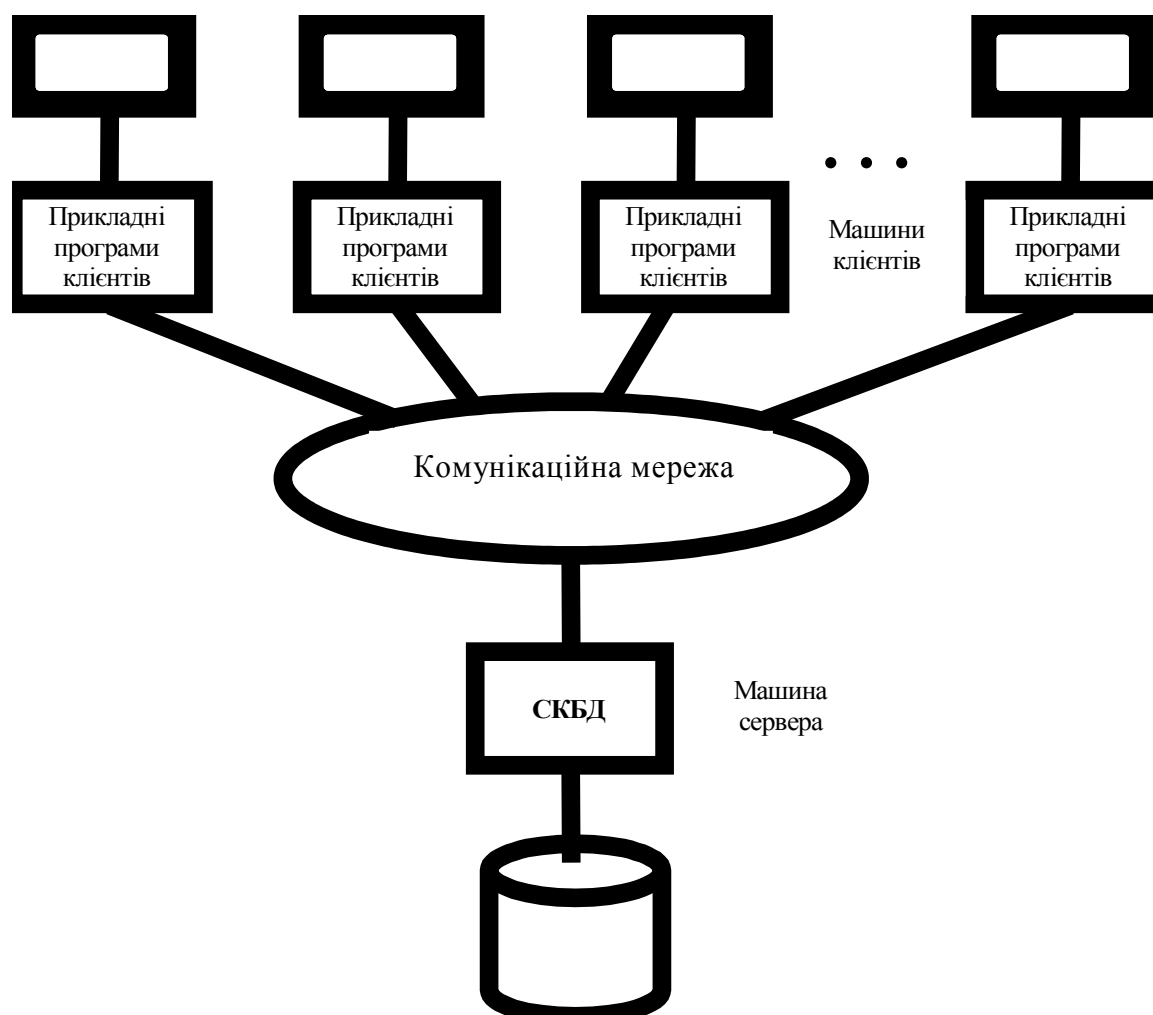


Рис. 2.1 Архітектура клієнт-сервер.

Архітектура клієнт-сервер, для якої призначені серверні СКБД, є деякою мірою поверненням до колишньої "мейнфреймової" моделі, заснованої на централізації збереження й обробки даних на одному виділеному комп'ютері, де функціонує спеціальна прикладна програма або сервіс, що називається сервером БД. Сервер БД відповідає за роботу з файлами БД, підтримку цілісності посилань, резервне копіювання, забезпечення авторизованого доступу до даних, протоколювання операцій і, звичайно, за виконання запитів користувача на вибір і модифікацію даних і метаданих (визначення інших об'єктів системи). Клієнтські прикладні програми, що є джерелами цих запитів, функціонують на персональних комп'ютерах у мережі. Можна відзначити, що при використанні серверних СКБД виконання запитів виробляється самим сервером, тому клієнтські прикладні програми одержують від сервера тільки результати самого запиту і не вимагають передачі всього індексу чи всієї таблиці, що істотно знижує мережний трафік при обробці запитів.

Однією з важливих переваг архітектури клієнт-сервер є зниження інтенсивності мережного трафіку при виконанні запитів. Наступна перевага полягає в можливості зберігання бізнес-правил (наприклад, правил цілісності посилань або обмежень на значення даних) на сервері, що дозволяє уникнути дублювання коду в різних клієнтських прикладних програмах, які використовують загальну базу даних.

Сучасні серверні СКБД мають такі сервіси: реалізація для декількох платформ; зручні утиліти адміністрування; резервне копіювання даних; обслуговування реплікацій (копіювання даних в декілька інших БД); паралельна обробка даних в багатопроцесорних системах; підтримка OLAP і створення сховищ даних; виконання розподілених запитів і транзакцій; автоматизовані засоби проектування даних; підтримка Web-технологій.

У процесі роботи СКБД виникає необхідність захисту БД від випадкових та навмисних ситуацій, коли існує імовірність втрати даних. Одним із засобів вирішення цієї проблеми є механізм транзакцій. Транзакція (*Transaction*) – це логічна послідовність операцій над даними (читання, віддалення, вставки, модифікації), неподільна з точки зору впливу на БД, яка або виконується вся разом, або вся разом скасовується.

Завершення (*Commit*) транзакції означає, що всі операції, що входять до складу транзакції, успішно завершені, і результат їхньої роботи збережений у базі даних. Відкат (*Rollback*) транзакції означає, що усі уже виконані операції, що входять до складу транзакції, скасовуються і всі об'єкти БД, порушені цими операціями, повернуті у вихідний стан. Для реалізації можливості відкату транзакцій деякі СКБД підтримують запис у *log*-файли, що дозволяють відновити вихідні дані при відкоті. Транзакція може складатися з декількох вкладених транзакцій. Деякі СКБД підтримують двофазне завершення транзакцій – процес, що дозволяє здійснювати транзакції над декількома БД, що відносяться до однієї і тієї ж СКБД. Для підтримки розподілених транзакцій (тобто транзакцій над БД, керованих різними СКБД), існують спеціальні засоби, що зветься моніторами транзакцій.

Крім зазначених переваг сучасні серверні СКБД володіють можливістю оптимізації запитів, яка виконується спеціальним компонентом СКБД – оптимізатором запитів. В оброблювачі запитів СКБД MS SQL Server [5] реалізовано нові методи пошуку, які підвищують швидкість обробки комплексних запитів. SQL Server використовує техніку перетину та з'єднання індексів для таблиць з декількома індексами, нові методи пошуку, які підвищують швидкість обробки комплексних запитів. SQL Server підтримує паралельне виконання запитів, підтримку розподілених транзакцій, що дозволяє сполучати в один запиті віддалені сервери. Не дивлячись на це, не можна вважати, що знайдено адекватне вирішення задачі глобальної оптимізації в розподіленій СКБД.

2.2.2. Розподілені бази даних

Основна задача систем управління розподіленими базами даних полягає в забезпеченні засобів інтеграції локальних БД, розташованих у вузлах обчислювальної мережі, для того, щоб користувач, працюючий в будь-якому вузлі мережі, мав доступ до всіх її частин, як до єдиної БД.

Розподілена система БД (рис. 2.2) означає, що окрема прикладна програма може “прозора” обробляти дані, розподілені між множиною різних БД, керування якими здійснюють різні СКБД, які працюють на з'єднаних комунікаційними мережами машинах різних типів з різними операційними системами. Поняття “прозора” означає, що прикладна програма виконує обробку даних з логічної точки зору так, якби управління даними повністю здійснювалося однією СКБД, яка працює на одній машині. Тобто клієнт може отримувати доступ до будь-якої кількості серверів одночасно (за один запит можливо отримати дані з двох і більше серверів). В цьому випадку сервери розглядаються клієнтом як єдиний сервер (з логічної точки зору) і користувач може не знати, на якій машині знаходиться та або інша частина даних. Фундаментальний принцип розподіленої системи БД можна сформулювати так: для користувача розподілена система повинна виглядати так, як нерозподілена [4].

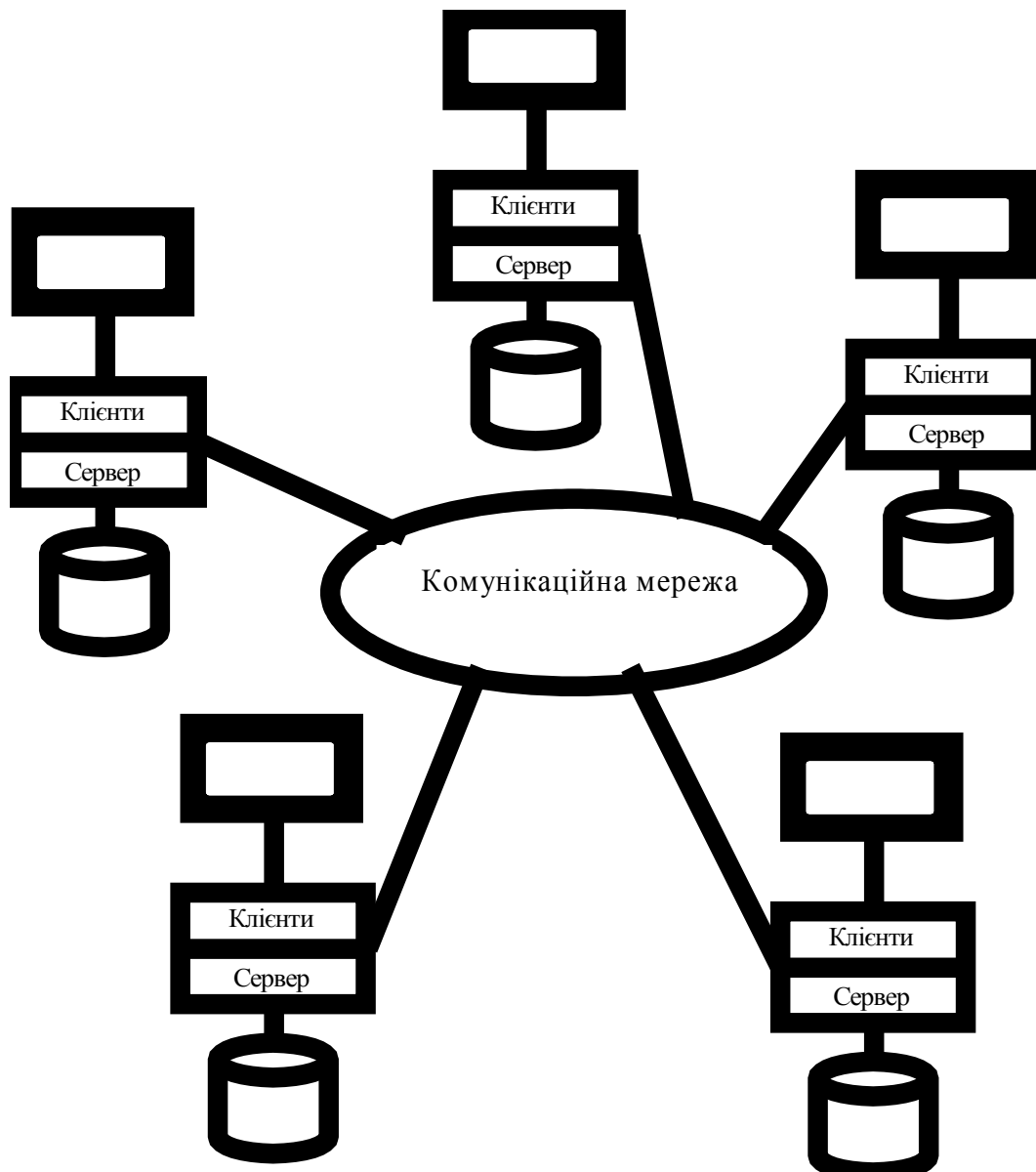


Рис. 2.2 Розподілена система БД.

Реалізація розподілених БД переслідує наступні цілі: локальна незалежність, відсутність опори на центральний вузол, неперервне функціонування, незалежність від розташування, незалежність від фрагментації, незалежність від реплікації (фрагментарне копіювання), обробка розподілених запитів, управління розподіленими транзакціями, апаратна незалежність, незалежність від операційної системи, незалежність від мережі, незалежність від типу СКБД.

Незалежність від типу СКБД є дуже важливою вимогою. В цьому випадку звертаються за допомогою до спеціальних програм, так званих шлюзів. Робота шлюзів полягає в тому, що за рахунок їх використання одна СКБД бачить роботу іншої в зрозумілому для неї вигляді, для чого використовуються загальні протоколи обміну інформацією, типи даних (або здійснюється перетворення одних типів даних в інші) та забезпечується сумісна реалізація блокувань, виконання транзакцій і т.ін.

Для забезпечення інтеграції локальних БД потрібно вирішити проблеми декомпозиції початкового запита, оптимального засобу виконання запита, забезпечення синхронізації, знаходження та рішення розподілених тупиків, відновлення стану БД.

Традиційно виділяють однорідні і неоднорідні розподілені БД. В однорідних розподілених БД кожна локальна БД керується однією й тією ж СКБД. На відміну від цього, в неоднорідній системі локальні БД можуть відноситись до різних моделей даних, тому мережна інтеграція неоднорідних БД – це актуальна, але складна проблема. Багато рішень відомі на теоретичному рівні, але поки що не вдається подолати головну проблему – недостатню ефективність інтегрованих систем. Вирішенні цієї проблеми в значній мірі сприяє стандартизація мови SQL та загальне слідування виробників СКБД принципам відкритих систем.

Основною ідеєю відкритих систем є спрощення сполучення обчислювальних систем за рахунок стандартизації апаратної та програмної частини. Головною причиною розвитку концепції відкритих систем є перехід до використання локальних обчислювальних мереж та проблеми спряження апаратно-програмних

засобів, які викликали цей перехід. В зв'язку з розвитком технології глобальних комунікацій відкриті системи набувають ще більше значення та масштабність.

Стосовно розподіленої СКБД архітектура клієнт-сервер цікава та актуальна головним чином тому, що вона забезпечує просте і відносно дешеве рішення проблеми колективного доступу до БД в локальній мережі. В деякому роді системи, засновані на архітектурі клієнт-сервер, є спрощеним наближенням до розподілених систем, які не вимагають рішення основного набору проблем дійсно розподілених БД. Архітектура клієнт-сервер – це окремий (поодинокий) випадок розподілених систем в цілому. Реальне розповсюдження архітектури клієнт-сервер стало можливим завдяки розвитку і широкому впровадженню в практику концепції відкритих систем.

Розглянемо класифікацію прикладних програм, які використовують БД. Почнемо їхній розгляд з прикладних програм, що працюють в архітектурі клієнт-сервер. ІС, створені в такій архітектурі, являють собою сервер БД, який маніпулює даними, і клієнтську прикладну програму, що звертається до нього і використовує для цього або клієнтські *API*, або один з універсальних механізмів доступу до даних. Звичайно при використанні такої архітектури прикладних програм на сервер БД покладається також контроль дотримання бізнес-правил, реалізованих у виді збережених процедур, тригерів, серверних обмежень і інших об'єктів БД. Для створення клієнтських прикладних програм у цьому випадку найчастіше застосовуються засоби розробки, що володіють розвинутими візуальними інструментами, такі як *Microsoft Visual Basic, Borland Delphi, Sybase PowerBuilder, Borland C++Builder*.

Слід зазначити, однак, що вибір архітектур сучасних прикладних програм не вичерпується "класичною" архітектурою клієнт-сервер, яка припускає, що прикладна програма складається із сервера БД і клієнтських прикладних програм, взаємодіючих з цим сервером. Існують так звані розподілені прикладні програми. Розподілені (або багатоланкові) прикладні програми звичайно складаються з презентаційних сервісів, сервісів бізнес-логіки, реалізованих у вигляді бізнес-об'єктів і сервісів даних (звичайно складаються із сервера БД і механізмів доступу до даних). Сервіси бізнес-логіки призначені для одержання введених користувачем даних від презентаційних сервісів, взаємодії із сервісами даних для виконання бізнес-операцій (наприклад, обробки або замовлень розрахунку бухгалтерського балансу) і повернення результатів цих операцій презентаційним сервісам. Деякі з бізнес-об'єктів можуть звертатися до сервісів даних, використовуючи ті або інші механізми доступу до даних. Оскільки кінцевий користувач не взаємодіє безпосередньо з бізнес-об'єктами, останні звичайно не мають інтерфейсу користувача у звичному розумінні. Фізично бізнес-об'єкти можуть бути реалізовані у виді сервісів операційної системи, консольних прикладних програм або *Windows*-прикладних програм, а також у вигляді бібліотек, які завантажуються в адресний простір спеціально призначеної для цієї мети серверної прикладної програми (*Web*-сервера, сервера прикладних програм і т. ін.). Для створення бізнес-об'єктів застосовуються як засоби розробки з розвинутими візуальними інструментами, так і засоби розробки, орієнтовані на "ручне" створення коду. Відзначимо, що новітні версії майже усіх найбільш популярних засобів розробки *Windows*-прикладних програм (*Microsoft Visual Basic, Visual FoxPro і Visual C++, Borland Delphi і C++Builder, Sybase PowerBuilder*) підтримують створення різних типів бізнес-об'єктів (*Web*-прикладних програм, *ASP*-об'єктів, *COM*-серверів і ін.), за винятком, мабуть, *Microsoft Access* – цей продукт розрахований скоріше на кваліфікованих користувачів, ніж на розроблювачів розподілених систем. Нерідко для цієї мети використовуються і засоби створення *Java*-прикладних програм (такі як *Borland JBuilder*).

2.2.3. Технологія аналітичної обробки даних OLAP та багатовимірні сховища даних

Ефективне управління крупним і середнім бізнесом сьогодні неможливе без застосування новітніх інформаційних технологій, зокрема без систем підтримки прийняття рішень (СППР). СППР – це системи, які призначені для аналізу ділової інформації. Подібні системи створюються на основі таких теорій, як дослідження операцій, теорія поведінки, наукова теорія управління, а також методів статистичної обробки і дозволяють вирішувати наступні класи задач [6].

Аналітичні – обчислення заданих показників та статистичних характеристик бізнес-діяльності на основі інформації із БД.

Візуалізація даних – наочне графічне і табличне представлення інформації.

Зобуття знань – визначення взаємозв'язків та взаємозалежностей бізнес-процесів на основі існуючої інформації. Наприклад, перевірка статистичних гіпотез. Шляхом аналізу економічних та фінансових показників діяльності компаній, які потім збанкрутіли, банк може виявити деякі стереотипи, які можна буде урахувати при оцінці показника ризику кредитування.

Імітаційні – проведення на комп'ютері експериментів з математичними моделями, які описують поведінку складних систем на протязі визначеного або формуемого інтервалу часу.

Синтез управління – використовується для визначення досяжності намічених цілей, визначення множини можливих управляючих впливів, які приводять до наміченої мети.

Оптимізаційні – засновані на інтеграції імітаційних, управлінських, оптимізаційних та статистичних методів моделювання та прогнозування. Задачі даного класу дозволяють вибрати з множини можливих управлінь ті з них, які забезпечують найефективніше (з точки зору заданого критерію) просування до поставленої мети.

СППР засновані на пов'язаних між собою технологіях: сховища даних (data warehouse), магазини даних (data mart), банки оперативних даних (operational data store), оперативна аналітична обробка OLAP.

OLAP-система повинна базуватися на спеціальній БД – сховищі даних. В залежності від задач сховище даних може бути реалізоване як на основі багатовимірної бази, так і на основі реляційної. Дані, які зберігаються в базах з реляційною моделлю, зовні представлені у вигляді таблиць, які складаються із рядків та стовпчиків. Такі структури даних найкращим чином забезпечують оперативну обробку інформації, яка постійно надходить (On-Line Transaction Processing, OLTP), але в меншому ступені пристосовані на побудові на їхній основі СППР. OLTP системи орієнтовані на прикладні програми, які виконують за відносно короткий проміжок часу велику кількість транзакцій, пов'язаних з додаванням, та поновлень достатньо невеликих об'ємів інформації, яка розміщена по різним взаємозв'язаним таблицям. В типовій ситуації, наприклад, потрібно поновити усі дані, пов'язані з конкретним замовником. Кожна таблиця містить ідентифікаційний номер замовника, який використовується для встановлення зв'язків між різними таблицями. Такий простий зв'язок між записами дозволяє при додаванні чи зміні даних модифікувати за одну транзакцію відносно невелику кількість записів.

В багатовимірній моделі дані представлені у вигляді багатовимірного куба, у якого виміри відповідають всім куба, а змінні – індивідуальним коміркам куба. Наприклад, якщо аналізувати об'єм продажу по товарам в залежності від регіону і часу, то в такому випадку треба розглядати модель багатовимірної БД з трьома вимірами (товар, регіон, місяць) та одним показником об'ємом продажу (рис.2.3).

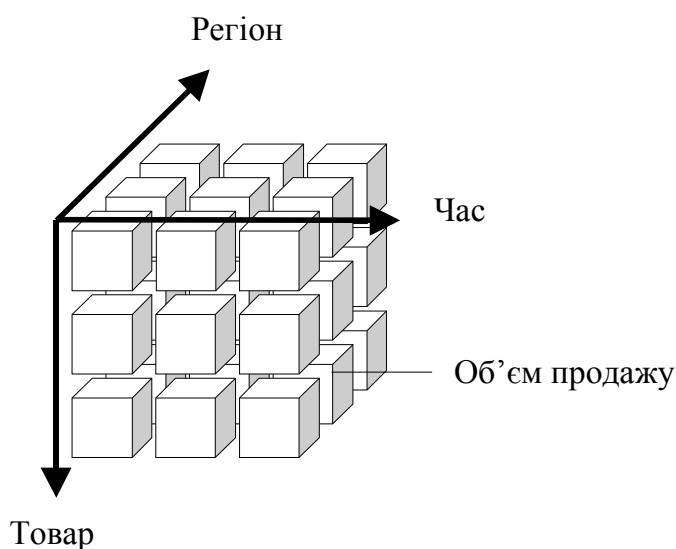


Рис. 2.3 Багатовимірна модель представлення даних.

Багатовимірна модель дозволяє робити плоскі зрізи куба даних та повертати його потрібною гранню будь-яким зручним способом. Використовуючи багатовимірну модель, аналітик може легко отримати представлення даних у відповідності до власних інтересів.

Як реляційні моделі з рядків та стовпчиків, так і багатовимірні моделі кубів, є логічним представленням інформації. Насправді дані на диску зберігаються у вигляді деякої впорядкованої структури, яка відображається в логічну модель тільки при запитах користувача. Тому система оперативного аналізу може бути побудована і на основі реляційної бази. Цей напрямок розвитку програмне забезпечення отримало назву Relational OLAP (ROLAP). Типовий представник – Oracle Discoverer.

На відміну від ROLAP оперативний аналіз даних, який реалізується на основі багатовимірних БД, отримав назву Multidimensional OLAP (MOLAP). Перевага використання багатовимірних БД – їх потужність. Час виконання OLAP-операцій в багатовимірних базах даних може на декілька порядків перевищувати відповідний показник для реляційних БД.

2.2.4. Об'єктно-орієнтовані БД

Великий інтерес викликають результати досліджень, отримані за останній час в галузі об'єктно-орієнтованих систем БД або скорочено – об'єктних систем. Деякі автори вважають їх серйозним конкурентом реляційним системам [4].

Об'єктні системи беруть свій початок від об'єктно-орієнтованих мов програмування. Основна ідея, яка поєднує ці дві області, полягає в тому, щоб відгородити користувача від конструкцій, пов'язаних з апаратним забезпеченням, таким як біти або байти (а може записи і поля). Замість цього користувач має справу з об'єктами і операціями над цими об'єктами, які більше відповідають своїм аналогам в реальному часі. Наприклад, використовуючи спеціальні терміни, можливо уявляти відділ, як “кортеж ОТД” з набором відповідних “кортежів РАБ”, тобто співробітників, які мають “значення зовнішніх ключів”, які

“посилаються” на “значення первинного ключа” в “кортежі ОТД”. В новій технології користувач буде мати справу з об’єктом відділу, який вміщує відповідну множину об’єктів співробітників. Інакше кажучи, фундаментальна ідея об’єктного підходу – підвищення рівня абстракції.

В загальній та класичній постановці об’єктно-орієнтований підхід базується на концепціях: об’єкта та ідентифікатора об’єкта, атрибутів і методів, класів, ієрархії та наслідування класів. При такому наборі базових понять об’єктно-орієнтований підхід дуже близький до мов програмування з абстрактними чи довільними типами даних. З іншого боку, об’єктно-орієнтований підхід дуже близький до семантичного моделювання. Фундаментальні абстракції, які лежать в основі семантичних моделей, неявно використовують і в об’єктно-орієнтованому підході.

2.2.5. Мова побудови запитів SQL

Structured Query Language (SQL) [7] – це назва структурованої мови запитів, яка містить засоби створення та обробки даних в реляційних моделях. Незалежність від специфіки комп’ютерних технологій, а також підтримка SQL лідерами промисловості в галузі реляційних БД зробили його індустріальним стандартом. SQL – це не процедурна і реляційно повна мова, яка використовується для формулювання запитів до БД у більшості сучасних СКБД. Непроцедурність мови означає, що на ній можна вказати, що потрібно зробити з БД, але не можна описати алгоритм цього процесу. Реляційна повнота мови означає, що вирази цієї мови дозволяють визначити кожне відношення за допомогою алгебраїчних виразів початкової алгебри. В основу мови покладено як реляційну алгебру, так і реляційне числення.

Всі алгоритми обробки *SQL*-запитів генеруються самою СКБД і не залежать від користувача. Мова *SQL* складається з операторів, які можна розділити на кілька категорій:

- *Data Definition Language (DDL)* – мова визначення даних, що дозволяє створювати, видаляти і змінювати об’єкти в базах даних
- *Data Manipulation Language (DML)* – мова керування даними, що дозволяє модифікувати, додавати і видаляти дані в наявних об’єктах БД;
- *Data Control Languages (DCL)* – мова, яка використовується для керування привілеями користувачів;
- *Transaction Control Language (TCL)* – мова для керування змінами, зробленими групами операторів;
- *Cursor Control Language (CCL)* – оператори для визначення курсору, підготовки операторів *SQL* до виконання і деяких інших операцій.

Один з самих важливих операторів цієї мови – оператор *Select*, його застосовують для вибору даних.

2.3. База даних як модель предметної області

Система БД – це комп’ютеризована система збереження інформації, основне призначення якої – зберігати інформацію, надаючи користувачам засоби її вилучення та модифікації. Інформація – відомості про об’єкти і явища навколишнього середовища, їхні параметри, властивості і стан, що сприймають інформаційні системи (живі організми, що керують машинами й ін.) у процесі життєдіяльності і роботи. Дані – це будь-який сигнал, одержуваний і оброблений або людиною за допомогою органів почуттів, або технічним пристроєм. Зазначимо, що деякі автори не розрізняють цих понять [4].

Стосовно комп’ютерної обробки даних під інформацією розуміють деяку послідовність символічних позначень (букв, цифр, закодованих графічних образів і звуків і т.п.), яка несе значне навантаження і відображена в зрозумілому комп’ютеру виді. Предмети, процеси, явища матеріальної чи нематеріальної властивості, розглянуті з погляду їхніх інформаційних властивостей, називаються інформаційними об’єктами. Усі ці процеси, пов’язані з визначеними операціями над інформацією, називаються інформаційними процесами. Інформація має наступні властивості: вірогідність, повнота, точність, цінність, своєчасність, зрозумілість, доступність, стислість і т.д.

Будь-яка БД є складовою частиною деякої ІС, що має на увазі не тільки збереження даних, але і їхню обробку. При розробці БД звичайно виділяється кілька рівнів, за допомогою яких відбувається перехід від предметної області до конкретної реалізації БД засобами конкретної СКБД.

Предметна область – частина реального світу, яка підлягає вивченню з метою організації керування і, у кінцевому рахунку, автоматизації. Предметна область представляється великою кількістю фрагментів, наприклад, університет – факультетами, кафедрами, групами і т.д. Кожен фрагмент предметної області характеризується великою кількістю об’єктів і процесів, що використовують об’єкти, а також великою кількістю користувачів, які характеризуються різними поглядами на предметну область. У теорії проектування ІС предметну область (чи, якщо завгодно, увесь реальний світ у цілому) прийнято розглядати у виді трьох уявлень: уявлення предметної області в тім виді, як вона реально існує, як її сприймає людина (мається на увазі проектувальник БД), як вона може бути описана за допомогою символів. Тобто, кажуть, що ми маємо справу з реальністю, описом (уявленням) реальності і з даними, що відбивають це уявлення.

Дані, які використовуються для опису предметної області, можна представити у вигляді трьохрівневої схеми (так звана модель *ANSI/SPARC*) (рис. 2.4).

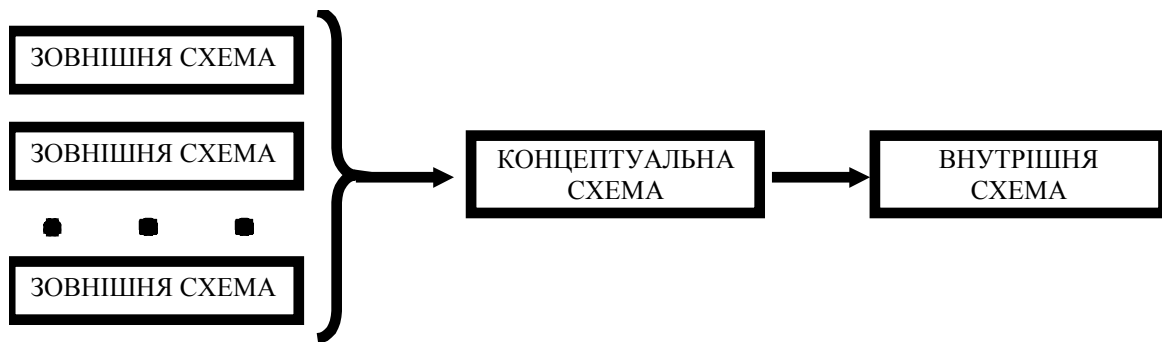


Рис. 2.4 Архітектура ANSI/SPARC.

Розрізняють концептуальний, внутрішній і зовнішній рівні даних БД, яким відповідають моделі аналогічного призначення. Концептуальний рівень відповідає логічному уявленню даних предметної області в узагальненому виді. Це – ядро проектування БД. Концептуальна модель складається з логічно структурованих різних типів даних. Внутрішній рівень відображає організацію даних у середовищі збереження і відповідає фізичному уявленню даних. Внутрішня модель складається з окремих записів, фізично збережених у зовнішніх носіях. Зовнішній рівень підтримує подання даних, які необхідні конкретним користувачем. За допомогою зовнішніх моделей підтримується доступ до різних прикладних програм.

Зовнішня схема даних є сукупністю вимог до даних з боку деякої конкретної функції, виконуваної користувачем. Концептуальна схема є повною сукупністю усіх вимог до даних, отриманої з уявлень користувача про реальний світ. Внутрішня схема – це сама БД. Звідси випливають основні етапи, на які розбивається процес проектування БД для ІС.

2.4. Моделі організації даних

Коли мова йде про моделі організації даних, то, як правило, розглядають ієрархічну, мережну та реляційну модель організації даних [8].

2.4.1. Ієрархічна модель

Організація даних у СКБД ієрархічного типу визначається в термінах: елемент, агрегат, запис (група), групове відношення, БД. Атрибут (елемент даних) – найменша одиниця структури даних. Як правило, кожному елементу при описі БД привласнюється унікальне ім'я. По цьому імені до нього звертаються при обробці. Елемент даних також часто називають полем. Запис – іменована сукупність атрибутів. Використання записів дозволяє за одне звертання до бази одержати деяку логічно зв'язану сукупність даних. Саме записи змінюються, додаються і знищуються. Тип запису визначається складом її атрибутів. Екземпляр запису – конкретний запис з конкретним значенням елементів. Групове відношення – ієрархічне відношення між записами двох типів. Батьківський запис (власник групового відношення) називається вихідним записом, а дочірні записи (члени групового відношення) – підлеглими.

Ієрархічна БД може зберігати тільки такі деревоподібні структури. Кореневий запис кожного дерева обов'язково повинен містити ключ з унікальним значенням. Ключі некоренових записів повинні мати унікальне значення тільки в рамках групового відношення. Кожен запис ідентифікується повним складеним ключем, під яким розуміється сукупність ключів усіх записів від кореневої по ієрархічному шляху. При графічному зображенні групові відносини зображують дугами орієнтованого графа, а типи записів – вершинами. Для групових відносин в ієрархічній моделі забезпечується автоматичний режим включення і фіксоване членство. Це означає, що для запам'ятовування будь-якого некоренового запису в БД повинен існувати її батьківський запис. Ієрархічна модель даних припускає жорстке підпорядкування даних, підлеглий елемент не може існувати без свого попередника по ієрархії. В цій моделі реалізовано зв'язок “один-до-багатьох”.

Ієрархічна модель задовольняє вимоги багатьох, але не всіх реальних задач. Недоліки можна проілюструвати при виконанні кожної з операцій маніпулювання даними. Розглянемо можливу ієрархічну модель БД “Постачальники_Деталі” (рис.2.5). В цій моделі дані представлені в вигляді чотирьох окремих дерев або записів, по одному на кожну деталь. Кожне дерево складається з одного запису деталі та підлеглому запису постачальника, тобто в моделі є записи двох типів. Додавання нового постачальника неможливе без включення нової деталі. При віддаленні відомостей про деталь губиться інформація про постачальника. При необхідності зміни опису постачальника виникає проблема переглядання всієї моделі з метою пошуку всіх записів даного постачальника. Частково недоліки ієрархічної моделі зняті в мережній моделі даних.

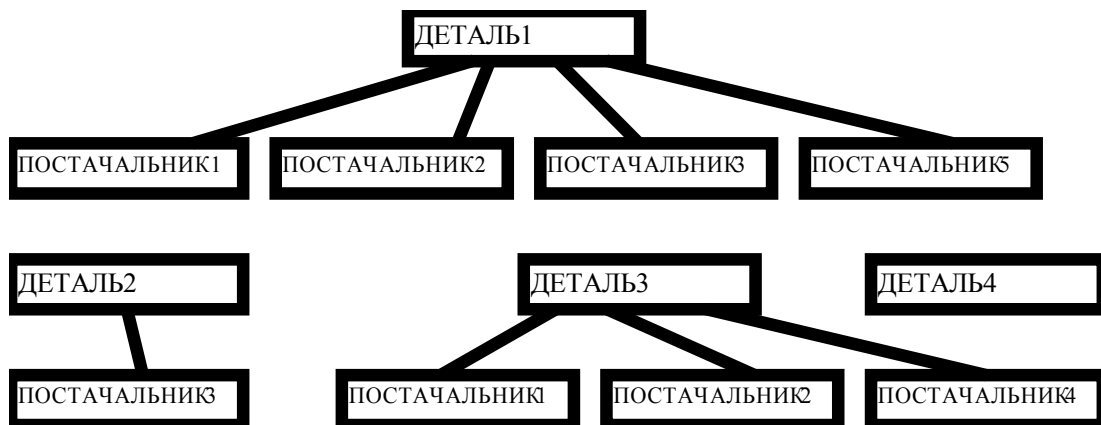


Рис. 2.5. Приклад ієрархічної моделі БД “Постачальники_Деталі”.

2.4.2. Мережна модель

Мережна модель даних свого часу була дуже популярною, а у деяких системах використовується і нині (термін “мережна” не має відношення до комп’ютерної мережі). Мережна модель даних визначається в тих же термінах, що й ієрархічна. Вона складається з багатьох записів, що можуть бути власниками, або членами групових відносин. Зв’язок між записом-власником і записом-членом також має вид $1:N$. Основне розходження цих моделей полягає в тому, що в мережній моделі запис може бути членом більш ніж одного групового відношення. Відповідно до цієї моделі кожне групове відношення іменується і проводиться розходження між його типом і екземпляром. Тип групового відношення задається його ім’ям і визначає властивості загальні для всіх екземплярів даного типу. Екземпляр групового відношення представляється записом-власником і множиною (можливо порожньою) підлеглих записів. При цьому є наступне обмеження: екземпляр запису не може бути членом двох екземплярів групових відносин одного типу.

В мережній моделі будь-який елемент може бути пов’язаний з будь-яким іншим елементом. Підлеглий елемент може мати більш одного початкового. В цій моделі реалізовано зв’язок “багато_до_багатьох” (рис. 2.6). Будь-яка мережна структура може бути приведена до більш простого виду шляхом введення надлишковості даних (рис. 2.7). Мережу можливо перетворити в дерево, якщо вказати деякі імена декілька разів.

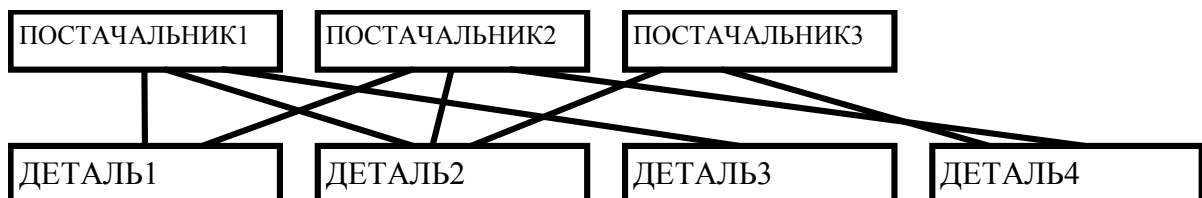


Рис. 2.6. Приклад мережної моделі БД “Постачальники_Деталі”.

В деяких випадках надлишковість, яка при цьому виникає, мала і є допустимою, в інших вона виявляється занадто високою. Багато методів, які широко використовуються для фізичного представлення деревовидних структур, є непридатними для представлення мережних. Саме цьому більшість програм, які оброблюють деревовидні структури, не можуть працювати з мережними. Завдання виявляється ще більш складним ще й тому, що методи, які застосовуються для одного виду мережних структур, є незадовільними для іншого виду. Мережна модель не є вільною від основного недоліку – чітко визначати на фізичному рівні зв’язки між даними і також чітко дотримуватись цій структурі при організації запитів до БД. Основна перевага мережної моделі – можливість визначати дуже складні інформаційні структури, які максимально відповідають природі інформаційних процесів. Запобігти зростаючій складності ієрархічної і мережної моделі даних вдалося введенням реляційної моделі.

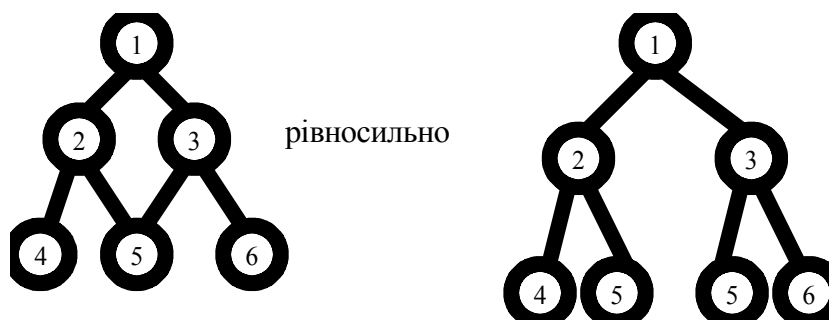


Рис. 2.7. Введення надлишковості в мережній моделі БД.

2.4.3. Багатовимірна модель даних

Багатовимірна модель даних використовується в технології OLAP. Відмітимо, що багатовимірність моделі даних означає логічне представлення інформації і не пов'язана з багатовимірністю візуалізації. Багатовимірні структури представляються як гіперкуби даних, Кожна грань куба є розмірністю. Основними поняттями, які використовуються в багатовимірних моделях даних, є “вимір” (dimension) та “комірка” (cell). Вимір – впорядкований набір значень, які приймає конкретний параметр, і який відповідає одній з граней гіперкуба. Комірка – це поле, відповідне атрибуту сутності, значення якого однозначно визначається фіксованим набором значень параметрів (значеннями “вимірів”).

В багатовимірній моделі задаються операції, серед яких можна виділити операції “формування зрізу” і “агрегації”. При формуванні зрізу користувачу на його запит надається деяка підмножина гіперкуба, отримана в результаті фіксації користувачем одного або декількох значень параметрів. Операція агрегація забезпечує перехід до більш загального представлення інформації із гіперкуба користувачу, наприклад, при додаванні значень показників по всім значенням одного з параметрів.

Така модель дозволяє порівнювати дані при різних значеннях параметрів, будувати графіки залежності значень конкретних атрибутів від значень заданих параметрів. Тому основне призначення технології OLAP – обробка інформації для проведення аналізу і прийняття рішень. Багатовимірна модель вважається перспективною технологією організації даних.

2.5. Реляційна модель даних

2.5.1. Концепція реляційної моделі даних

На сьогоднішній день майже всі сучасні СКБД підтримують реляційну модель даних. Концепція реляційної моделі даних модель була запропонована вченим Коддом у 1970 році. В основі реляційної моделі лежить поняття відношення (від англ. relation). Відношення зручно представляється у вигляді двомірної таблиці при дотриманні деяких обмежувальних умов. Таблиця зрозуміла і звична для людини. Доведено, що набір відношень (таблиць) може бути використаний для зберігання даних про об'єкти реального світу і моделювання зв'язків між ними [9]. При обговоренні поняття інформації можливо виділити три аспекти:

- інформація як відомості про реальний світ;
- інформація як опис;
- інформація як дані.

Взаємозв'язок понять, які відображають три аспекти представлення інформації представлені на рис. 2.8.

інформація як відомості про реальний світ	інформація як опис	інформація як дані
об'єкт	запис про об'єкт	логічний запис, кортеж
властивості об'єктів	атрибут	поле, елемент даних
набори об'єктів	набори записів про об'єкт	файл

Рис. 2.8. Аспекти представлення інформації.

Наприклад, для зберігання об'єкту або сутності “студент” використовують відношення СТУДЕНТ, в якому властивості сутності розміщуються в стовпчиках таблиці (рис. 2.9).

СТУДЕНТ

прізвище ім'я по-батькові	дата народження	курс	спеціальність
Петренко А.І.	12.02.92	2	прикл. мат
Іванов І.І.	01.08.94	3	механіка
Сидоров А.А.	29.05.93	1	економіка
...

Рис. 2.9. Відношення СТУДЕНТ.

Об'єкт – елемент реального світу, інформацію про який ми зберігаємо, визначена частина дійсності, яка нас оточує. Наприклад, предмет, процес, явище і т. ін. Об'єкти характеризуються властивостями. Наприклад, властивості об'єкту студент: прізвище та ім'я по-батькові, дата народження, курс, спеціальність. Сукупність однорідних об'єктів зветься набором об'єктів. Наприклад, група студентів.

У другому аспекті представлення інформація звичайно зберігається на паперових носіях, в яких зберігаються записи про об'єкти, які згруповані в набори записів про об'єкти. Властивості об'єктів називають атрибутами. Наприклад, атрибути: “прізвище_ім'я_по-батькові”, ”дата_народження”, “курс”, “спеціальність”. Атрибути містять значення. Наприклад, значення атрибутів відповідно: “Петренко А.І.”, ”12.02.92”, “2”, “прикл. мат”.

Елемент даних – найменша одиниця поійменованих даних. Елемент даних має назву – поле. Кожне поле має свій унікальний англословний ідентифікатор. Наприклад, FIO, DN, KURS, SPEC. Набір взаємозв'язаних елементів даних запису називають кортежем, логічним записом. Наприклад, “Петренко А.І. 12.02.92 2 прикл. мат”. Файл – поійменована сукупність всіх логічних записів заданого типу. Наприклад, файл групи студентів. Структура запису файлу групи студентів Student може мати вигляд (рис. 2.10):

Student		
Поле	Тип	Кількість знаків
FIO	char	15
DN	char	8
KURS	char	1
SPEC	char	17

Рис. 2.10 Структура запису файлу групи студентів Student.

Оригінальність підходу Кодда полягає в тому, що він запропонував застосовувати до відношень струнку систему операцій, які дозволяють отримувати одні відношення з інших. Це дає можливість поділяти інформацію на дві частини, яка зберігається і яка обчислюється, економити пам'ять, при необхідності обчислюючи частину інформації з тієї, що зберігається.

2.5.2. Реляційна алгебра і реляційне числення. Операції над відношеннями

Формальною основою реляційної моделі є реляційна алгебра, яка заснована на теорії множин, і реляційне числення, яке базується на математичній логіці. Ефективність реляційної СКБД визначається можливістю виконувати над відношеннями вісім операцій алгебри. Основними операціями над відношеннями в реляційній базі даних є наступні: об'єднання, перетин, різниця, декартовий добуток, ділення, проекція, з'єднання, вибір.

Властивості відношень

Відношення реляційної БД в залежності від змісту поділяють на об'єктні і зв'язні. В об'єктному відношенні один з атрибутів однозначно ідентифікує окремий об'єкт. Такий атрибут називають первинним ключем (*primary key*). У відношенні СТУДЕНТ на роль ключа претендує атрибут “прізвище_ім'я_по-батькові”. Для зручності первинний ключ записують в першому стовпчику таблиці. Якщо первинний ключ складається з більш ніж одного стовпчика, він називається складеним первинним ключем (*composite primary key*). В об'єктному відношенні не повинно бути рядків з однаковими ключами, тобто не повинно бути дублювання об'єктів. Це основне обмеження реляційної моделі для забезпечення цілісності даних.

Зв'язне відношення зберігає ключі двох або більше об'єктних відношень, тобто по ключам встановлюються зв'язки між об'єктами відношень. Наприклад, розглянемо об'єктне відношення СТУДЕНТ (рис. 2.11), об'єктне відношення ПРЕДМЕТ (рис. 2.12) і зв'язне відношення ВИВЧАС (рис. 2.13).

СТУДЕНТ

прізвище_ім'я_по-батькові	курс	спеціальність
Петренко А.І.	2	прикл. мат
Іванов І.І.	3	механіка
Сидоров А.А.	1	економіка

Рис. 2.11 Об'єктне відношення СТУДЕНТ.

ПРЕДМЕТ

назва	кількість семестрів
алгебра	4

історія	3
програмування	2
ін. мова	2

Рис. 2.12. Об'єктне відношення ПРЕДМЕТ.

ВІВЧАЄ

прізвище ім'я по-батькові	предмет
Іванов І.І.	алгебра
Іванов І.І.	програмування
Сидоров А.А.	ін. мова
Петренко А.І.	історія

Рис. 2.13. Зв'язне відношення ВІВЧАЄ.

Зв'язне відношення окрім полів, які воно зв'язує, може мати і інші атрибути, які залежать від цього зв'язку. Прикладом може бути відношення ВІВЧАЄ_ОЦІНКА (рис. 2.14).

ВІВЧАЄ ОЦІНКА

прізвище ім'я по-батькові	предмет	оцінка
Іванов І.І.	алгебра	4
Іванов І.І.	програмування	5
Сидоров А.А.	ін. мова	3
Петренко А.І.	історія	4

Рис. 2.14. Зв'язне відношення ВІВЧАЄ_ОЦІНКА.

Ключі в зв'язних відношеннях називаються зовнішніми ключами (*foreign key*) тому, що вони є первинними ключами інших відношень. Реляційна модель накладає на зовнішні ключі обмеження для забезпечення цілісності даних, яке називають довідковою цілісністю (*referential integrity*) (рис. 2.15). Це означає, що кожному зовнішньому ключу повинен відповідати рядок якого-небудь об'єктного відношення. Без такого обмеження може трапитись, що зовнішній ключ посилається на не існуючий об'єкт. Іншими словами, зовнішній ключ – це колонка або набір стовпчиків, чії значення збігаються з наявними значеннями первинного ключа іншої таблиці.

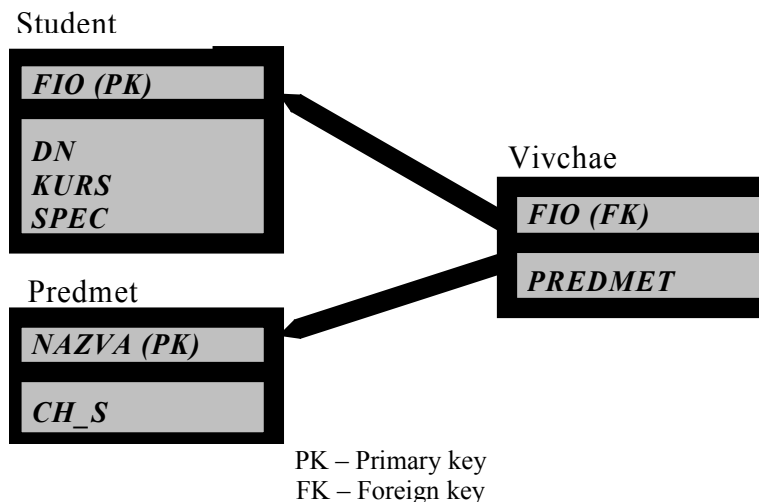


Рис. 2.15. Довідкова цілісність.

В реляційній моделі на відношення накладається і інше обмеження – вони повинні бути нормалізовані.

Подібні взаємовідносини між таблицями називаються зв'язком (*relationship*). Зв'язок між двома таблицями встановлюється шляхом присвоювання значень зовнішнього ключа однієї таблиці значенням первинного ключа іншої. Співвідношення можуть бути один-до-одного (*one-to-one relationship*), один-до-багатьох (*one-to-many relationship*) або співвідношенням *master-detail*. У останньому випадку таблиця, що містить зовнішній ключ, називається *detail*-таблицею, а таблиця, що містить первинний ключ, що визначає можливі значення зовнішнього ключа, називається *master*-таблицею. Група зв'язаних таблиць називається схемою БД (*database schema*).

2.6. Проектування реляційних баз даних

2.6.1. Процес нормалізації відношень

Однією з складних проблем, з якою стикається більшість розробників БД, є проблема оцінки, які елементи даних належить збирати в один кортеж. Непродумане групування елементів даних може призвести як мінімум до дублювання даних, а як максимум до неможливості виконати запит. Головна мета раціонального групування даних в кортежі – зменшення надмірності даних. Для рішення цієї проблеми застосовують підхід, який зветься нормалізація відношень [4], [10].

Відношення реляційної БД містять як структурну, так і семантичну інформацію. Структурна інформація задається схемою відношення, а семантична виражена функціональними зв'язками між атрибутами. Склад атрибутів БД повинен задовольняти дві основні вимоги:

- між атрибутами не може бути небажаної функціональної залежності;
- групування атрибутів повинне забезпечувати мінімальне дублювання даних.

Задоволення цих вимог досягається шляхом нормалізації відношень БД. Нормалізація відношень – це покроковий зворотній процес розкладання початкових відношень на більш дрібні і прості. При цьому встановлюються всі можливі функціональні залежності. Апарат нормалізації був розроблений Коддом. В ньому визначаються різні нормальні форми. Кожна з нормальних форм обмежує типи функціональних залежностей відношень. Кодд виділив три нормальні форми: 1НФ, 2НФ, 3НФ. Вважається достатнім приведення до 3НФ.

2.6.2. Перша нормальна форма

Кожен атрибут повинен бути простим, тобто мати атомарні або неподільні значення. Наприклад, відношення СТУДЕНТ_СПОРТ (рис. 2.16) ненормалізоване, оскільки містить складний атрибут “спорт”.

СТУДЕНТ_СПОРТ

прізвище_ім'я_по-батькові	курс	спеціальність	спорт	
			вид	розряд
Петренко А.І.	2	прикл. мат	плавання	мс
Іванов І.І.	3	механіка	футбол	кмс
Сидоров А.А.	1	економіка	шахи	1р

Рис. 2.16. Об'єктне відношення СТУДЕНТ_СПОРТ.

Первинним ключем цього відношення є атрибут “прізвище_ім'я_по-батькові”. Відношення, у якого всі атрибути прості, називається приведеним до першої нормальної форми (1НФ). Приведемо це відношення до 1НФ, тобто звільнимися від складного атрибута “спорт”. В отриманому відношенні ключ складається з атрибутів “прізвище_ім'я_по-батькові” та “вид_спорт” (рис. 2.17).

СТУДЕНТ ВИД СПОРТ

прізвище_ім'я_по-батькові	вид_спорт	курс	спеціальність	розряд
Петренко А.І.	плавання	2	прикл. мат	мс
Іванов І.І.	футбол	3	механіка	кмс
Сидоров А.А.	шахи	1	економіка	1р

Рис. 2.17. Об'єктне відношення СТУДЕНТ_ВИД_СПОРТ в першій нормальній формі.

2.6.3. Типи функціональних залежностей

Вважається, що атрибут В відношення R функціонально залежить від атрибута А того ж відношення, якщо в кожному момент часу кожному значенню атрибута А відповідає не більш чим одне значення атрибута В. Функціональна залежність відображається так: $A \rightarrow B$. Розглянемо відношення ВИКЛАДАЧ_ПРЕДМЕТ (рис. 2.18) з складеним первинним ключем “табельний_номер” і “назва_предмету”. Вважаємо, що викладач закріплений за однією кафедрою.

ВИКЛАДАЧ ПРЕДМЕТ

табельний_номер	назва_предмету	кількість_годин	прізвище	посада	оклад	кафедра	телефон
201	ІС	36	Федін	доцент	1000	ЕОМ	5-89
201	Бух. облік	72	Федін	доцент	1000	ЕОМ	5-89
202	СКБД	48	Васильєв	доцент	1000	ЕОМ	5-89

301	СКБД	48	Костенко	професор	2000	АСУ	5–89
401	Оптика	24	Петренко	асистент	500	Фізика	3–10
401	Механіка	30	Петренко	асистент	500	Фізика	3–10

Рис. 2.18. Відношення ВИКЛАДАЧ_ПРЕДМЕТ.

В цьому відношенні атрибут “посада” функціонально залежить від атрибута “оклад”, “табельний_номер” від атрибута “прізвище”, “прізвище” від атрибута “табельний_номер” і т. ін. Атрибут може функціонально залежати від цілої групи атрибутів. Якщо відношення знаходиться в 1НФ, то всі неключові атрибути функціонально залежать від ключа, але ступінь залежності може бути різним. Якщо неключовий атрибут залежить тільки від частини ключа, то кажуть про часткову залежність, наприклад, неключовий атрибут “кількість_годин” функціонально залежить тільки від частини ключа, тобто тільки від атрибута “назва_предмету”.

Якщо неключовий атрибут залежить від всього складеного ключа і не знаходиться в частковій залежності від його частин, то говорять про його повну функціональну залежність від складеного ключа. В відношенні ВИКЛАДАЧ_ПРЕДМЕТ не має атрибутів, які знаходяться в повній функціональній залежності від складеного ключа.

Якщо всі можливі ключі відношення містять по одному атрибуту, то таке відношення є відношенням в 2НФ, так як всі атрибути, які не є первинними, повністю залежать від можливих ключів. Якщо ключі складаються більш ніж з одного атрибута, то відношення, яке задане в 1НФ може не бути відношенням у 2НФ.

Якщо для атрибутів А, В, С виконуються умови $A \rightarrow B$, $B \rightarrow C$, а зворотна залежність відсутня, то говорять, що С залежить від А транзитивно. Наприклад, “прізвище” \rightarrow “кафедра” \rightarrow “телефон”.

У відношеннях між атрибутами може існувати ще один тип залежності – багатозначна залежність. У відношенні R атрибут В багатозначно залежить від А ($A \twoheadrightarrow B$), якщо кожному значенню А відповідає множина значень В, ніяк не пов’язаних з іншими атрибутами з R. Багатозначна залежність можлива при наявності у відношенні хоча б трьох атрибутів: ключа і не менш двох незалежних один від одного атрибутів. Наприклад, розглянемо відношення ВИКЛАДАЧ_РОЗКЛАД (рис. 2.19).

ВИКЛАДАЧ РОЗКЛАД

прізвище	група	предмет
Іванов	410	СКБД
Іванов	410	ІС
Іванов	309	СКБД
Іванов	309	ІС
Сидоров	510	СКБД

Рис. 2.19. Відношення ВИКЛАДАЧ_РОЗКЛАД.

Між викладачами і групами студентів є зв’язок типу “один-до-багатьох” (1:M), оскільки викладач може вести курсові проекти в одній і більше групах, однак кожній групі відповідає один викладач. Між викладачем і предметами є зв’язок типу “багато-до-багатьох” (M:N), оскільки викладач може читати один і більше предметів, і навпаки, один предмет можуть читати декілька викладачів. В відношенні існує незалежні багатозначні залежності “прізвище” \twoheadrightarrow “група” і “прізвище” \twoheadrightarrow “предмет”, так як значення багатозначних атрибутів “група” і “предмет” ніяк не пов’язані між собою і можлива зміна їх значень в будь-якому рядку відношення.

2.6.4. Друга нормальна форма

В відношенні ВИКЛАДАЧ_ПРЕДМЕТ можна відмітити часткову функціональну залежність атрибутів “прізвище”, “посада”, “оклад”, “кафедра”, “телефон” від частини “табельний_номер” складеного ключа. Така часткова залежність приводить до наступних аномалій:

1. Має місце дублювання даних про викладача, оскільки викладач може читати декілька предметів.
2. Існує проблема контролю надлишковості даних, так як зміна, наприклад, окладу спричиняє необхідність пошуку та зміни значень окладів у всіх кортежах з даним викладачем.
3. Виникає проблема з викладачами, які вданий час не ведуть предмети. Викладача без предмета неможливо включити у відношення і, навпаки, якщо викладач звільняється і вилучається із відношення, то буде віддалений і предмет, хоча предмет продовжує читатися. Таким чином, відношення у 2НФ потребує подальших перетворень.

Відношення знаходиться у другій нормальній формі, якщо воно знаходиться у 1НФ і кожний неключовий атрибут функціонально повно залежить від складеного ключа. Для того, щоб усунути часткову залежність і привести відношення до 2НФ, необхідно розкласти його на два відношення наступним чином:

- побудувати проекцію на частину складеного ключа і атрибути, залежні від цієї частини;

– побудувати проекцію без атрибутів, які знаходяться в функціональній залежності від складеного ключа.

Отримаємо два відношення ВИКЛАДАЧ і ПРЕДМЕТ, які знаходяться у 2НФ (рис. 2.20).

ВИКЛАДАЧ

табельний номер	прізвище	посада	оклад	кафедра	телефон
201	Федін	доцент	1000	ЕОМ	5–89
202	Васильєв	доцент	1000	ЕОМ	5–89
301	Костенко	професор	2000	АСУ	5–89
401	Петренко	асистент	500	Фізика	3–10

ПРЕДМЕТ

табельний номер	назва предмету	кількість годин
201	ІС	36
201	Бух. облік	72
202	СКБД	48
301	СКБД	48
401	Оптика	24
401	Механіка	30

Рис. 2.20. Відношення ВИКЛАДАЧ і ПРЕДМЕТ у другій нормальній формі.

Таким чином, приведення до 2НФ можна описати наступним чином. Нехай дана змінна-відношення $R(A,B,C,D)$ PRIMARY KEY $\{A,B\}$ і є функціональна залежність $A \rightarrow D$. Процедура нормалізації в 2НФ передбачає заміну цієї змінної-відношення наступними двома проекціями R1 і R2:

R1(A,D) PRIMARY KEY $\{A\}$

R2(A,B,C) PRIMARY KEY $\{A,B\}$

FOREIGN KEY $\{A\}$ REFERENCES R1

В ряді випадків і 2НФ породжує незручності, пов'язані з надлишковістю даних. Для їх усунення використовується наступний крок нормалізації, який перетворює 2НФ у 3НФ.

2.6.5. Третя нормальна форма та подальша нормалізація

В отриманому відношенні ВИКЛАДАЧ є наступні транзитивні функціональні залежності:

“табельний номер” → “кафедра” → “телефон”

“табельний номер” → “посада” → “оклад”

Наявність транзитивних функціональних залежностей викликає аномалії наступного характеру (на прикладі атрибуту “телефон”):

1. Має місце дублювання інформації про телефон для викладача однієї кафедри.

2. Існує проблема контролю надлишковості даних, оскільки зміна номеру телефону кафедри викликає необхідність пошуку і зміни всіх номерів телефонів всіх викладачів кафедри.

3. Не можна додати дані про нову кафедру (назву і номер телефону), якщо на даний момент відсутні викладачі. І, навпаки, при звільненні усіх викладачів кафедри дані про неї не можна зберегти.

Таким чином, відношення у 2нф потребує подальших перетворень. Відношення знаходиться у 3НФ, якщо воно знаходиться у 2НФ і в ньому відсутні транзитивні залежності неключових атрибутів від ключа.

Отримаємо три відношення ВИКЛАДАЧ, ПОСАДА і КАФЕДРА, які знаходяться у 3НФ (рис. 2.21).

ВИКЛАДАЧ

табельний номер	прізвище	посада	кафедра
201	Федін	доцент	ЕОМ
202	Васильєв	доцент	ЕОМ
301	Костенко	професор	АСУ
401	Петренко	асистент	Фізика

ПОСАДА

посада	оклад
доцент	1000
професор	2000
асистент	500

КАФЕДРА

кафедра	телефон
---------	---------

ЕОМ	5–89
АСУ	5–89
Фізика	3–10

Рис. 2.21. ВИКЛАДАЧ, ПОСАДА і КАФЕДРА у третій нормальній формі.

Приведення до 3НФ можна описати наступним чином. Нехай дана змінна-відношення $R(A,B,C)$ PRIMARY KEY {A} і є функціональна залежність $B \rightarrow C$. Процедура нормалізації в 3НФ передбачає заміну цієї змінної-відношення наступними двома проекціями R1 і R2:

R1(B,C) PRIMARY KEY {B}
R2(A,B) PRIMARY KEY {A}
FOREIGN KEY {B} REFERENCES R1

Таким чином, концептуальна модель БД у третій нормальній формі складається з чотирьох взаємопов'язаних відношень: ВИКЛАДАЧ, ПРЕДМЕТ, ПОСАДА і КАФЕДРА (рис. 2.22):

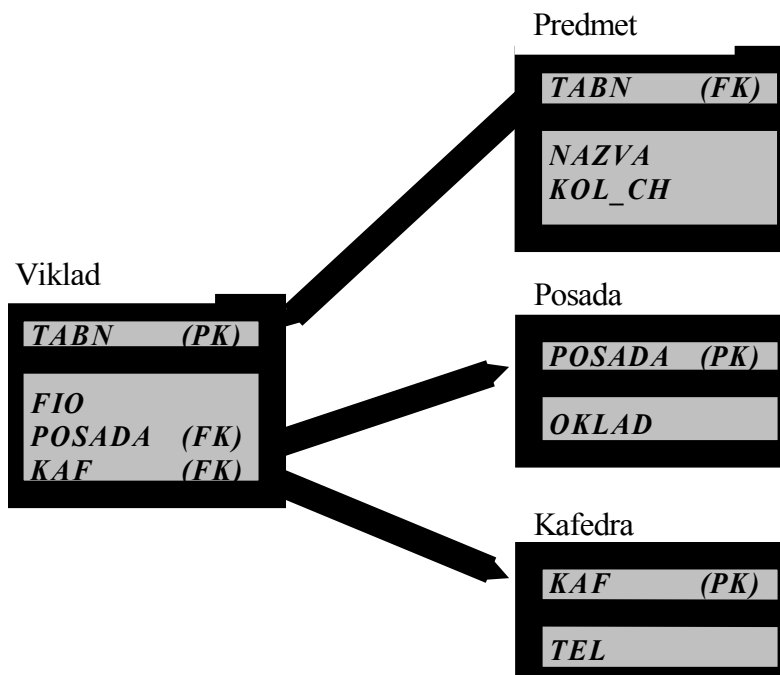


Рис. 2.22. Концептуальна модель БД.

Третя нормальна форма визволяє від надлишковості і аномалій виконання операцій включення, віддалення і поновлення, якщо відношення має один ключ, і інші залежності, в тому числі і багатозначні, в ньому відсутні. Але, якщо при цьому існують інші залежності, окрім залежності від ключа, то 3НФ не забезпечує відсутність аномалій операцій. В цьому випадку застосовують посилену 3НФ, так звану нормальну форму Бойса-Кодда (НФБК). Існують також 4НФ і 5НФ. Рівень нормалізації відношення залежить від його семантики, яка задається функціональними зв'язками. Вважається достатнім приведення до 3НФ.

2.7. Семантичне моделювання даних, ER-діаграми

Широке використання і розповсюдження реляційних БД показує, що реляційна модель даних достатня для моделювання предметних областей. Однак проектування реляційної БД в термінах відношень на основі методології нормалізації відношень являє собою складний процес. Необхідність в більш зручних і потужних засобах моделювання поставила перед проектувальниками задачу семантичного моделювання даних. Проводяться дослідницькі роботи над БД в семантичній моделі даних. При цьому розглядається два варіанти – забезпечення інтерфейсу користувача на основі семантичної моделі з автоматичним відображенням відношень в реляційну модель даних і пряма реалізація СКБД, заснована на якій-небудь реляційній моделі. Найбільш близько до другого підходу знаходяться сучасні об'єктно-орієнтовані СКБД, моделі даних яких по багатьом параметрам близькі до семантичних моделей.

Процес моделювання полягає в виділенні об'єктів (сутностей предметної області), встановленні властивостей виділених об'єктів і виявленні існуючих між ними зв'язків. Одна з найбільш важливих і розповсюджених семантичних моделей є модель "сутність-зв'язок", ER-модель (*Entity Relationship*). Основними поняттями ER-моделі є сутність, зв'язок і атрибут. Як і в реляційних схемах, в ER-моделях

вводиться поняття нормальних форм, їх зміст дуже близький до змісту реляційних нормальних форм. Цей підхід дозволяє на початковій стадії правильно спроектувати логічну структуру БД.

2.8. Засоби проектування даних

Існує декілька типів інструментальних засобів, так звані CASE-засоби, які дозволяють створювати діаграми “сутність-зв’язок” і проектувати дані. Процес проектування даних можна умовно розбити на два етапи: логічне і фізичне проектування. Результатом першого з них є так звана логічна або концептуальна модель даних, яка виражається звичайно діаграмою “сутність-зв’язок” або ER-діаграмою, яка представлена в одній із стандартних схем, прийнятих для відображення подібних діаграм. Результатом другого етапу є готова БД або DDL-скрипт (Data Definition Language) для її створення. Найбільш популярні засоби проектування даних наведені в Таблиці (рис. 2.23).

CASE - засіб	Виробник	URL
Designer 2000	Oracle	http://www.oracle.com/
ERwin	Computer Associates	http://www.cai.com/
PowerDesigner	Sybase	http://www.sybase.com/
ER/Studio	Embarcadero	http://www.embarcadero.com/
Visible Analyst	Visible Systems	http://www.visible.com
Visio Enterprise	Microsoft	http://www.Microsoft.com/

Рис. 2.23. Найбільш популярні засоби проектування даних.

Відмітимо, що багато з цих продуктів призначені не тільки для проектування даних, але і для вирішення інших завдань, наприклад, для моделювання потоків даних чи бізнес-процесів, функціонального моделювання, документування прикладних програм, управління проектами і т. ін.

2.9. Загальна характеристика баз знань та експертних систем

Для науково-технічного прогресу характерна зростаюча необхідність вирішення неформалізуємих або погано формалізуємих проблем, алгоритмічне рішення яких або не існує, або не може бути реалізоване в силу їх неповноти, невизначеності, неточності, творчого підходу. Прикладами можуть бути задачі ідентифікації, інтерпретації, розпізнавання, класифікації, прогнозування, діагностики, проектування, контролю та попередження виникнення нештатних ситуацій, тестування, навчання. Вирішення цих проблем потребує “інтелектуалізації” комп’ютера, реалізації технології обробки даних із застосуванням ідей штучного інтелекту. Термін “штучний інтелект” трактується двояко, а саме:

- штучна система, що імітує вирішення людиною складних задач в процесі її життєдіяльності;
- вибір напрямків, супроводжуючих та обумовлюючих створення систем штучного інтелекту (проблеми представлення знань та роботи з ними, планування доцільної поведінки, проблеми спілкування на природній мові, проблеми сприйняття зорових, акустичних та інших видів інформації, яка надходить із зовнішнього середовища).

При вивченні інтелектуальних систем традиційно присутня інформація, яку називають знаннями. Інтелектуальну систему можна сприймати як деякий комп’ютерний аналог спеціаліста вузького профілю, який буде здатний вирішувати проблеми в якій-небудь предметній області на тому ж рівні компетентності, на якому працюють спеціалісти в даній області, експерти. В 80-ті роки завдяки досягненням в галузі штучного інтелекту з’явилося багато систем, що базувались на використанні знань [11]. Найбільш розповсюдженим видом інтелектуальних систем є експертні системи (ЕС). ЕС – це система, яка забезпечує створення та використання за допомогою обчислювальної машини баз знань експертів, тобто спеціалістів в конкретній предметній галузі. Основні компоненти ЕС показані на рис. 2.24.

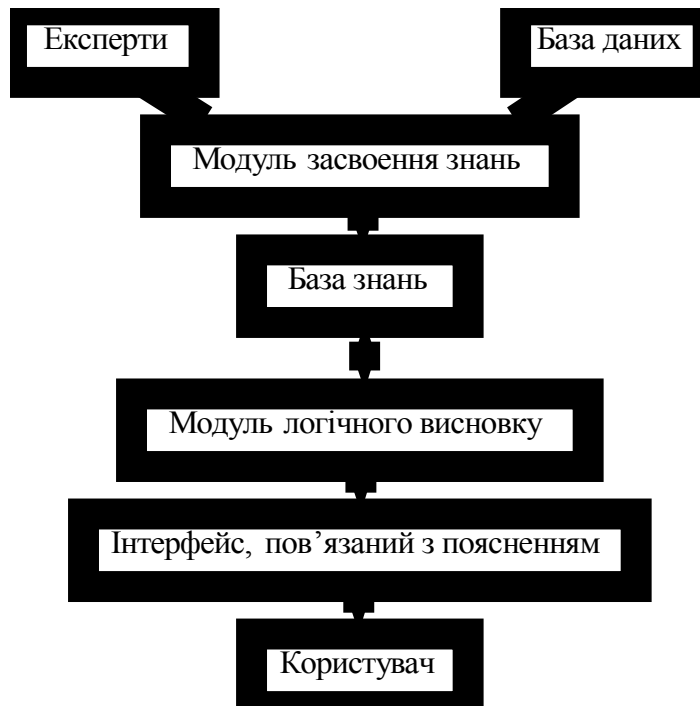


Рис. 2.24. Структура експертної системи.

Для експертної системи, за аналогією з відомим виразом Вірта: “дані+алгоритми=програми”, справедливий наступний вираз: “знання+висновок=експертна система”. Таким чином, база знань і модуль логічного висновку утворюють ядро експертної системи. База знань може трактуватись як сукупність основних відомостей, які відносяться до певної області знань, тобто деяким чином представлена інформація про конкретну предметну область; факти, закономірності, правила, евристичні припущення.

Знання – вид інформації (подібно до програм та даних), яка зберігається в базах знань і відображає знання людини – спеціаліста (експерта) в певній предметній галузі. Знання можуть бути представлені в формі даних. Зокрема, в вигляді тексту на деякій формальній мові, в вигляді мережі, яка задає зв'язки різного роду між елементами знань. Отже, знання є деяка більш висока ступінь організації даних, що допускає спеціальну інтерпретацію. Від мови програмування потребується забезпечити такі засоби представлення даних, щоб інтерпретатор міг розглядати дану сукупність даних як знання [12].

Знання відображає безліч всіх поточних ситуацій в об'єктах даного типу та способи переходу від одного опису об'єкта до іншого. Для знань характерні внутрішня інтерпретуємість, структурованість, зв'язність та активність. Образне формулювання знань: “знання=факти+переконання+правила”. Найбільш часто в системах штучного інтелекту знання представлені у вигляді:

- семантичних мереж;
- фреймів;
- правил продукції;
- формальних логічних моделей.

Семантична мережа – спосіб представлення знань в вигляді позначеного орієнтованого графа, в якому вершини відповідають поняттям, об'єктам, діям, ситуаціям або складним відношенням, а дуги – властивостям або елементарним відношенням. Основною перевагою семантичної мережі є те, що вона більше чим інші моделі відповідає сучасним уявленням про організацію довготривалої пам'яті людини. Недоліком є складність організації процедури виводу.

Фрейми – структура для представлення стереотипної ситуації. Фрейм можливо зобразити у вигляді мережі із вузлів та відношень (дуг). Верхній рівень фрейму зафіксований і містить знання завжди істинні для ситуацій, що представляються ним. На більш низьких рівнях знаходяться слоти – “дірки”, котрі не обов'язково заповнені. “Розуміння”, тобто опис конкретної ситуації, відбувається шляхом пристосування до неї певного фрейму, а саме, заповненням слотів відповідними конкретними даними. Слотам можуть бути задалегідь приписані деякі значення. Споріднені фрейми можуть сполучатися в систему фреймів, утворюючи таким чином інформаційно-пошукову мережу. Значеннями слотів можуть бути конкретні дані, посилання на інші файли, імена процедур обчислення тих чи інших значень.

Семантична мережа може бути представлена як мережа фреймів та навпаки. Перевага фреймів – елементи, які традиційно присутні в описі об'єкта, згруповуються і завдяки цьому можуть здобуватися та оброблятися як єдине ціле. Фрейми відображають концептуальну основу організації пам'яті людини, фреймова модель наочна та гнучка.

Продукційна модель або модель, заснована на правилах, дозволяє представляти знання у вигляді речень “якщо (умова), то (дія)”. Правило продукції – це програма із одного оператора виду “якщо (умова), то (дія)”. В основі правил продукції лежить простий причинно-наслідковий механізм. Вони визначають набір дозволених перетворень, за допомогою яких відбувається просування від початкового стану до остаточного рішення поставленої задачі. Поточний (проміжний) стан відображається за допомогою безлічі фактів, що фіксуються в базі даних. В перебігу розв’язання задачі відбувається порівняння одної із частин правила з вмістом бази даних. В реальній системі можлива ситуація, коли можуть бути одночасно застосовані декілька правил. Тому виникає необхідність в управляючій структурі, яка повинна вирішувати, яке із правил застосувати.

Продукційні системи складаються з трьох компонентів:

- бази даних, що містить правила продукції;
- бази даних, яка відображає поточний стан деякої задачі;
- управляючої структури (інтерпретатора), що вирішує, яке з правил продукції потрібно застосувати наступним.

Механізм виводу в продукційній моделі заснований на двох засобах – прямій та зворотній аргументації. Пряма аргументація – кожне правило обробляється від посилання до закінчення. Пряма аргументація, як правило, використовується для знаходження результатів або наслідків існуючої ситуації. Зворотна аргументація – кожне правило обробляється від закінчення до посилання.

Модуль логічних висновків розрізняє правило, в висновку якого міститься рішення проблеми користувача. Після цього модуль визначає чи істинне посилання даного правила. В випадку істини правило можна активізувати з метою отримання рішення. Разом з тим, посилання правила може бути невідомим. В таких випадках модуль логічних висновків розглядає невідомі змінні в посиланні як нові підзадачі і намагається вирішити всі підзадачі шляхом подальшої обробки інших правил в зворотному напрямку. Це відбувається доти, доки не буде знайдено рішення чи поки задача не буде визнана такою, що не має розв’язку.

Продукційна модель частіше всього застосовується в промислових експертних системах. Вона приваблює своєю наочністю, легкістю внесення доповнень і змін, високою модульністю та простотою механізму логічного виводу.

Традиційно в представленні знань виділяють формальні логічні моделі, засновані на численні предикатів. Предикат – це вираз, який може приймати значення “да” або “ні”. Ця модель застосовується в дослідницьких “ігрових” системах, тому що пред’являє дуже високі вимоги та обмеження до предметної області.

Ключовим поняттям в системах баз даних є база даних, а для систем баз знань – поняття бази знань (БЗ). Під системою баз даних (СБД) розуміють як інструментальну систему, яка забезпечує створення і використання БД, так і систему, яка забезпечує функціонування конкретної прикладної БД або декількох баз, тобто прикладну систему. В першому значенні мають на увазі СКБД. Аналогічно СБД системою баз знань (СБЗ) називають систему, яка забезпечує створення і використання баз знань.

В багатьох СБЗ і ЕС зміст бази знань поділяють на “факти” і “правила”, причому факти грають роль елементарних “одиниць знання”, а правила використовуються для виразу зв’язків, залежностей між фактами і їх комбінаціями. В системах баз даних цьому діленню відповідає ділення на об’єкти і зв’язки, що прийняте в багатьох моделях даних. БД містить не тільки “факти” (записи об’єктів з їх атрибутами), але й зв’язки. Отже, традиційну БД можливо розглядати як своєрідну базу неалгоритмічних знань.

СБЗ відрізняються від СБД способами представлення неалгоритмічних знань з відповідними механізмами отримання рішень і характером знань, які містяться в БЗ. Пряме використання знань із БЗ для вирішення задач забезпечується механізмом отримання рішень – другою основною частиною СБЗ, яку називають також механізмом виводу, процедурою пошуку, планування, рішення і т. п. Механізм отримання рішень дає можливість вилучати із БЗ відповіді на запитання, отримувати рішення задач. Принципи роботи механізму отримання рішень тісно зв’язані зі способами представлення знань в БЗ. Для знань, представлених в БЗ рівняннями, механізм отримання рішень являє собою процедуру рішення рівняння.

2.10. Аналіз мереж методами теорії масового обслуговування

Теорія масового обслуговування – найбільш природний інструмент аналізу комп’ютерних мереж. Вона може використовуватися для отримання відповідей, наприклад, на такі запитання:

- як змінюється тривалість пошуку файлу при збільшенні частоти звернень до диску?
- чи зміниться тривалість відповіді, якщо в однакове число разів збільшиться як швидкість процесора, так і число користувачів системи?
- скільки каналів повинна мати система з розділенням часу при використанні набірною доступу?
- скільки терміналів потрібно мати у довідковому або операторському центрі?

Число задач, які можуть вирішуватися методами теорії масового обслуговування, надзвичайно велике, і такі задачі виникають фактично у кожній області інформатики. Здатність зробити такий аналіз – істотна перевага спеціалістів, пов’язаних з розподіленими інформаційними системами.

Хоча теорія масового обслуговування є достатньо складним розділом математичної статистики, її застосування до аналізу роботи будь-яких підприємств та організацій, технічних систем у багатьох випадках дає суттєве покращання результатів. В першу чергу потрібно знання елементарних статистичних основ та розуміння застосовності теорії масового обслуговування.

Мета цього розділу – описати практичні методи застосування теорії масового обслуговування. Виникає багато випадків, коли важливо прогнозувати вплив деякої зміни в конструкції та/або топології мережі: або очікується зріст навантаження на мережу, або планується модифікація чи розширення мережі.

Продуктивність мережі – системна характеристика. У інтерактивній системі або у системі реального часу параметр продуктивності – час реакції. У багатьох випадках забезпечення потрібної продуктивності – головна проблема. Для її вирішення можливі такі підходи.

1. Аналіз, заснований на фактичних значеннях. Він робиться після практичної реалізації проекту.
2. Аналіз на основі принципу подібності. Робиться екстраполяція результатів функціонування існуючої мережі на майбутню мережу більшого масштабу.
3. Розробка аналітичної моделі, заснованої на теорії масового обслуговування.
4. Розробка і дослідження імітаційної моделі.

Перший метод – практично метод проб та помилок. Це досить дорогий та неефективний метод. Другий метод є більш перспективним. Однак неможливо створити точну модель. Скоріше, можна дати оцінки у деякому діапазоні для розробленого проекту. Недоліком цього методу є те, що поведінка більшості систем при поточних змінах навантаження не співпадає з інтуїтивно очікуваними результатами. Якщо є якась система з розподілим середовищем, наприклад, телекомунікаційна чи комп’ютерна мережа, лінія високовольтної передачі, система з розподілом, то попит у послугах такої системи зазвичай збільшується в показовому ступені.

Типовий приклад наведений на рис. 2.25. Верхня лінія показує, як змінюється затримка відповіді t_B системи з розподілом ресурсів при збільшенні коефіцієнту використання k_B мережі (нагадаємо, що коефіцієнт використання мережі є відношення навантаження на мережу до пропускнуої спроможності мережі).

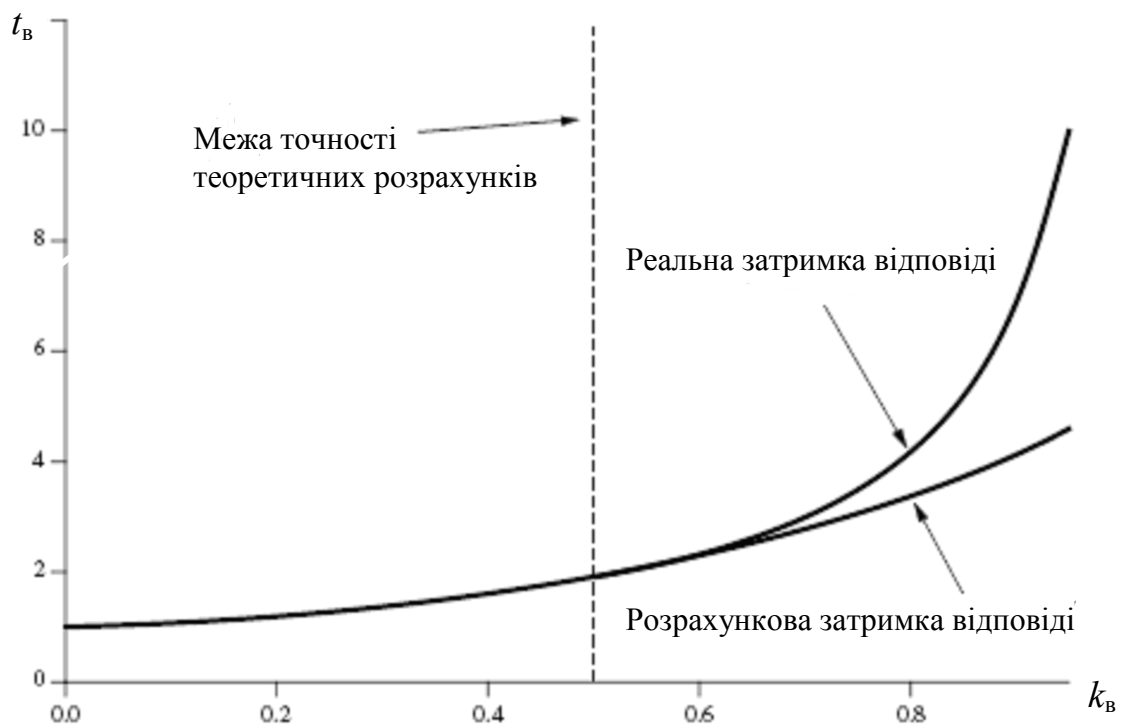


Рис. 2.25. Зміна затримка відповіді.

Можна зробити висновок, що при простій екстраполяції реальна якість роботи системи не відповідає розрахунковій вже при $k_B \geq 0,6 \dots 0,7$, а при $k_B \geq 0,8 \dots 0,9$, мережа взагалі починає працювати “на себе”, передаючи втрачені пакети знову і знову.

Тому, потрібний точніший інструмент прогнозу. Доцільно використовувати аналітичну модель у вигляді набору рівнянь, у результаті рішення яких можуть бути отримані бажані параметри (затримка відповіді, продуктивність, і т.п.). Для комп’ютерів, операційних систем, мережних технологій, інших практичних задач аналітичні моделі, засновані на теорії масового обслуговування, забезпечують прийнятну збіжність теорії та

практики. Трудність використання теорії масового обслуговування заключається в тому, що для розв'язання рівнянь і отримання рішень у замкнутій формі треба робити цілий ряд спрощуючих припущень.

Останній метод – імітаційна модель. Використовуючи спеціалізовані мови програмування для створення імітаційних моделей, можна з достатньою гнучкістю та детальністю моделювати реальні процеси та об'єкти і уникати введення багатьох припущень, потрібних при використанні теорії масового обслуговування. Проте, в більшості випадків, імітаційна модель не може служити першим етапом аналізу. Точність результатів імітаційного моделювання в усіх випадках обмежена точністю вхідних даних. Крім того, навіть за наявності багатьох припущень, що вводяться при використанні теорії масового обслуговування, отримані результати часто бувають дуже близькими до тих, які були б одержані б при більш детальному імітаційному моделюванні. До того ж, аналіз на основі теорії масового обслуговування може виконаний за значно коротший термін, ніж моделювання, яке може потребувати дні, тижні, або місяці для програмування і відладки моделі.

Відповідно, спеціалісту з розподілених інформаційних систем належить опанувати теорію масового обслуговування або хоча б ознайомитися з її основами.

2.10.1. Математичні моделі систем масового обслуговування

Модель одноканальної системи

Найпростішу систему масового обслуговування (СМО) зображено на рис. 6.2. Центральний елемент системи - сервер, який обслуговує деякі заявки. Ці заявки поступають в систему обслуговування. Якщо сервер вільний, заявка обслуговується негайно. Інакше заявка, що прибуває, стає в чергу. Коли сервер завершив обслуговування заявки, вона відбуває. Якщо є заявки, що чекають в черзі, одна з них негайно поступає на обслуговування до сервера. Сервер в цій моделі може виконувати деяку функцію обслуговування заявок. Приклади: процесор надає послугу процесам; лінія передачі даних надає послугу передачі пакетам або кадрам; пристрій вводу-виводу забезпечує читання або запис запитів.

Параметри потоку заявок

На рис. 2.26 ілюструються деякі важливі параметри, пов'язані із моделями масового обслуговування. Заявки прибувають в буфер з деякою середньою інтенсивністю λ (число заявок у секунду). У будь-який даний час, в черзі буде знаходитись певна кількість заявок (нуль або більше); позначимо середнє число заявок у черзі через W , середнє число заявок, що обслуговуються – через ρ , а середній час очікування T_w . Цей час усереднюється по всім заявкам, що поступають на вхід, з урахуванням тих, які не чекають взагалі. Сервер обслуговує заявки, що поступають, з середнім часом обслуговування T_s . Це часовий інтервал між посылкою заявки до сервера і виходу обслугової заявки з сервера. Інтенсивність обслуговування μ – це число обслужених заявок за одиницю часу. Загальне середнє число заявок, що знаходяться в системі, в тому числі заявка, що обслуговується (якщо вона є) і заявки, що очікують обслуговування (якщо вони є), означимо T і середній час, впродовж якого заявка знаходиться в системі (чекає своєї черги і обслуговується,) - T_r ; цей час розглядаємо як середній час загального знаходження заявки в системі (очікування плюс обслуговування).

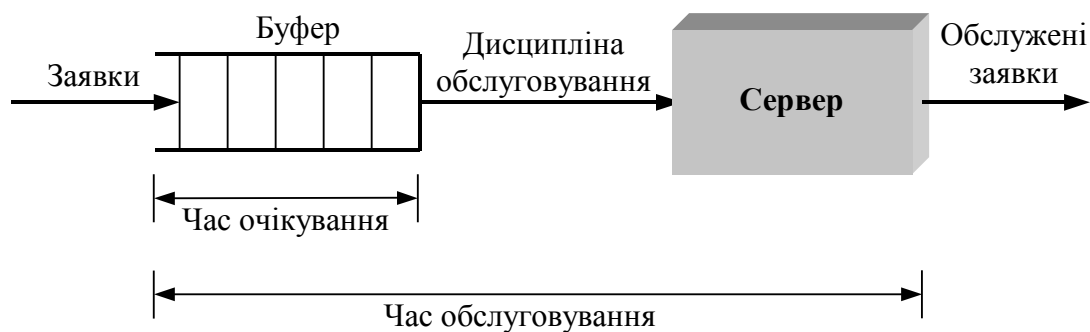


Рис. 2.26. Параметри потоку заявок.

Якщо ми припускаємо, що місткість черги нескінченна, то ніякі заявки ніколи в системі не втрачаються; вони тільки затримуються впродовж часу очікування та обслуговування. При цих обставинах, середнє число відправлених заявок дорівнює середньому числу прибуваючих заявок у одиницю часу. При збільшенні інтенсивності прибуття заявок на вхід системи час знаходження заявок в системі також збільшується, що призводить до заторів. Черга стає довшою, час очікування збільшується. При $\rho = 1$, тобто $\lambda = \mu$, сервер

насичується, працюючи 100% часу. Тому теоретична максимальна інтенсивність вхідного потоку пов'язана з середнім часом обслуговування T_s як $\lambda_{\max} = \frac{1}{T_s}$.

Проте при насиченні системи, коли $\rho = 1$, практично черга зростає до нескінченності. На практиці при обмеженому розмірі буферної пам'яті та наявності обмежень на затримку відповіді зазвичай обмежують інтенсивність вхідного потоку в одноканальній системі лімітом від 70% до 90% відносно теоретичного максимуму. Для подовження аналізу зробимо деяке припущення щодо цієї моделі.

Довжина потоку заявок. У першому наближенні приймається модель нескінченного потоку. Це означає, що середня частота появлень заявок не змінюється при їх втратах. Якщо довжина потоку обмежена, то обсяг заявок, які можна очікувати на вході системи, зменшується на число заявок, що зараз знаходяться в системі; це зазвичай приводить до пропорційного зменшення середньої частоти появлень заявок.

Розмір черги. Якщо приймається нескінченний розмір черги, час очікування може рости до нескінченності. За умов обмеженої черги деякі заявки в системі можуть втрачатися. На практиці, звичайно, будь-яка черга є обмеженою. У багатьох випадках це не приводить до суттєвої різниці в аналізі.

Дисципліна обслуговування. Коли сервер закінчує обслуговування поточної заявки, і, якщо в черзі є більш ніж одна заявка, повинно бути прийнято рішення щодо того, яку заявку обслуговувати далі. Найпростіша дисципліна обслуговування – так звана дисципліна *FIFO* (First In – First Out) - перший увійшов, перший вийшов; ця дисципліна – те, що зазвичай мається на увазі при використанні власно терміну *черга*. Інший вид дисципліни обслуговування – *LIFO* (Last In - First Out) останній зайшов – перший вийшов. На практиці вибирають дисципліну обслуговування з міркувань припустимого часу обслуговування. Наприклад, у вузлі з комутацією пакетів можна передбачити відправку спочатку найкоротших або навпаки, найдовших пакетів. Цей вибір визначається характером трафіку та вимогами до якості обслуговування.

Модель багатоканальної системи

На рис. 2.27 зображено узагальнену модель багатоканальної системи обслуговування з загальним буфером. Якщо заявка прибуває в момент, коли хоча б один сервер вільний, вона негайно відсилається до того сервера. Передбачається, що всі сервери ідентичні; тому, якщо доступний більш ніж один сервер, не має значення, який сервер вибраний для обслуговування. Якщо всі сервери зайняті, починає формуватися черга. Як тільки один сервер звільняється, заявка вибирається з черги, відповідно з діючою дисципліною обслуговування.

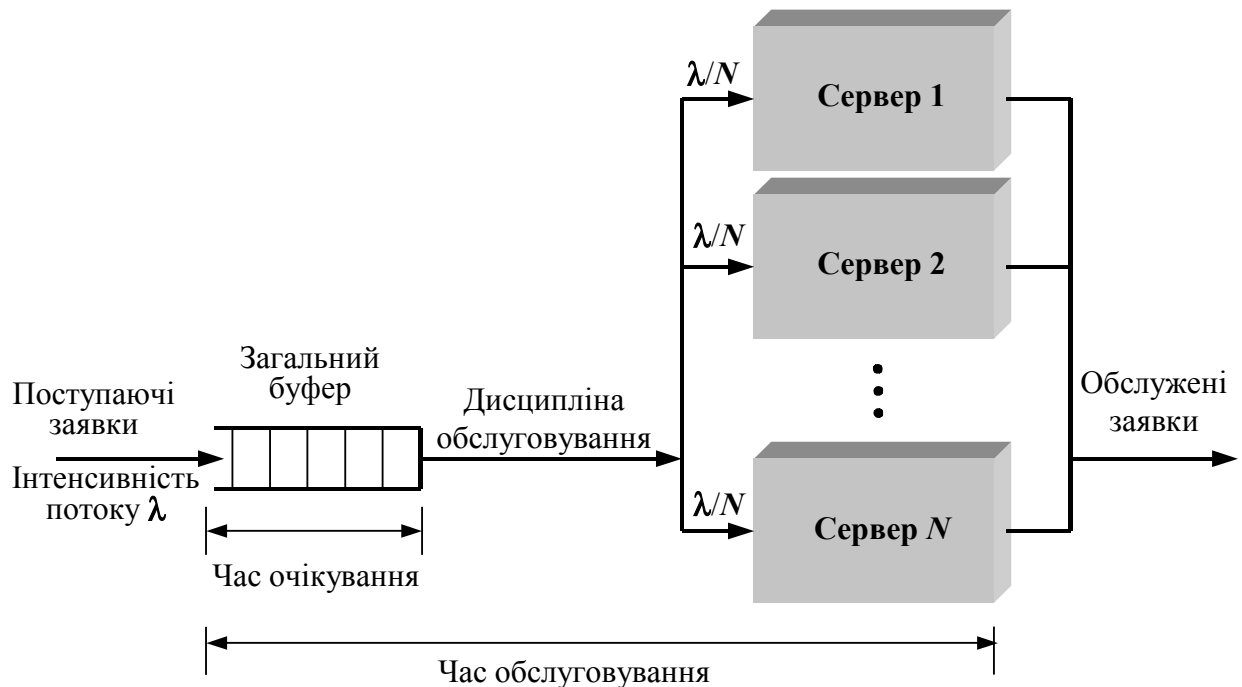


Рис. 2.27. Багатоканальна система обслуговування з загальною буферною пам'яттю (загальна черга з заданою дисципліною обслуговування).

За винятком інтенсивності обслуговування ρ , всі параметри, використані при аналізі одноканальної системи, мають те ж саме значення. Якщо ми маємо N ідентичних серверів з однаковою інтенсивністю обслуговування кожним сервером, що дорівнює ρ , то можна вважати, що середня інтенсивність

обслуговування системи в цілому дорівнює $N\rho$; цей останній термін часто співвідносять з інтенсивністю трафіку u , що чисельно дорівнює інтенсивності вхідного потоку заявок λ . Теоретичний максимум відносної інтенсивності обслуговування дорівнює $N \times 100\%$, а теоретичний максимум інтенсивності вхідного потоку є $\lambda_{\max} = \frac{N}{T_s}$.

Ключові характеристики для черги з декількома обслуговуючими пристроями аналогічні характеристикам для одноканальної системи. Припускається нескінченний об'єм буферної пам'яті і нескінченний розмір черги, з розподілом черги між всіма обслуговуючими пристроями (серверами). Звичайно вважають, що реалізується дисципліна обслуговування в порядку надходження (*FIFO*). Для випадку багатоканальної системи обслуговування, якщо всі сервери передбачаються ідентичним, вибір специфічного сервера для чергової заявки не впливає на час обслуговування.

На рис. 2.28 зображено багатоканальну систему з розділеною буферною пам'яттю, яку можна трактувати як паралельну структуру з одноканальною системою обслуговування. Хоча зміни в структурі не є принциповими, робочі характеристики зображеної системи можуть істотно відрізнятись від тієї, яка розглянута раніше.

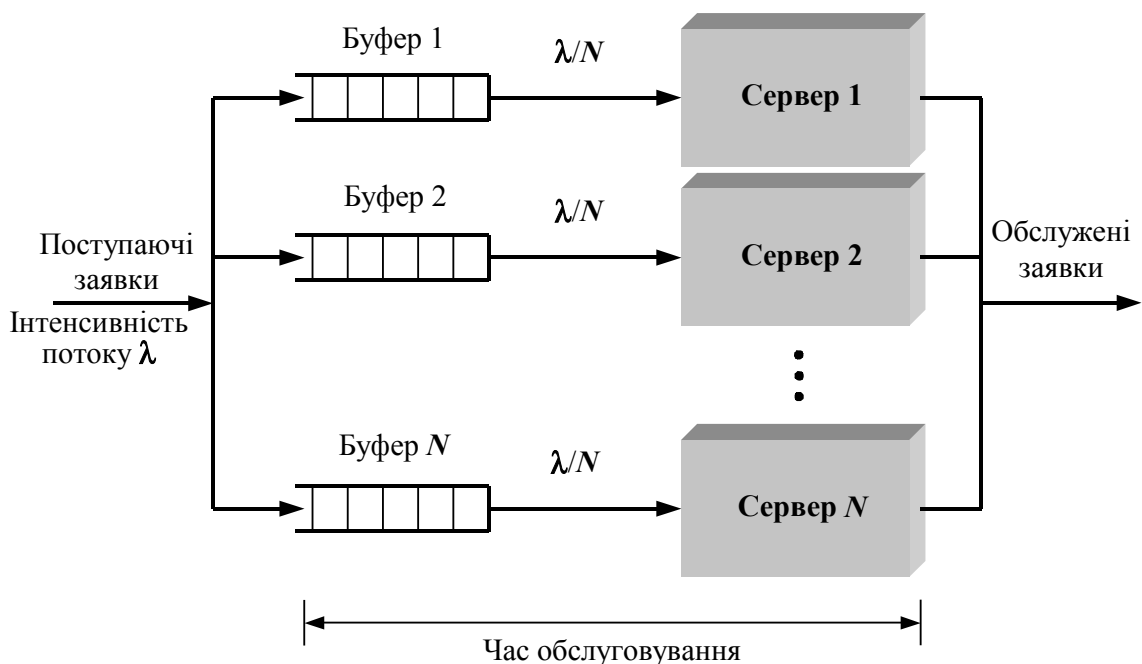


Рис. 2.28. Багатоканальна система обслуговування з розділеною буферною пам'яттю (індивідуальні черги з заданими дисциплінами обслуговування).

2.10.2. Основні співвідношення теорії масового обслуговування

Для продовження аналізу зробимо деякі спрощуючі припущення. Ці припущення, звичайно, можуть погіршити адекватність моделі реальним ситуаціям, але в більшості випадків результати будуть достатньо точні для вирішення задач планування і розробки конкретних проектів.

Наведемо деякі співвідношення, справедливі, наприклад, для стаціонарних та ергодичних процесів приходу заявок на обслуговування. Ці співвідношення можуть бути корисні як асимптотичні наближення реальних процесів.

Для оцінювання середнього розміру черги r за умов стаціонарності та ергодичності процесу приходу заявок використовуються так звані формула Літгла:

- для одноканальної системи обслуговування $r = \lambda T_r$, $r = w + \rho$;

- для N -канальної системи обслуговування $\rho = \frac{\lambda T_r}{N}$, $u = \lambda T_s = \rho N$, $r = w + N\rho$, де

$$T_r = T_w + T_s.$$

Відповідно можна через формули Літтла зв'язати число ρ з інтенсивністю приходу заявок λ та часом знаходження заявки в системі T_s . Воно дорівнює $\rho = \lambda T_s$.

Таким чином, для аналізу системи масового обслуговування необхідно мати таку апріорну інформацію:

- інтенсивність вхідного потоку заявок;
- середній час обслуговування;
- число каналів обслуговування.

На основі даної інформації можна отримати асимптотичні оцінки середнього числа заявок у черзі, середній час очікування та загальний час знаходження заявки в системі.

Необхідно враховувати, що потоки заявок можуть бути розподілені не по закону Пуассона, а по іншим імовірнісним законам з так званими "важкими хвостами". Це розподіли Парето, Вейбулла, логарифмічно-нормальний розподіл, гамма-розподіл, бета-розподіл та деякі інші, менш популярні. Наприклад, вираз для

щільності імовірності розподілу Парето має наступний вид: $f(x) = \frac{\alpha}{k} \left(\frac{k}{x}\right)^{\alpha+1}$, де k і α ($k, \alpha > 0$) -

параметри розподілу. Відповідно функція імовірності $F(x) = 1 - \left(\frac{k}{x}\right)^\alpha$ ($x > k; \alpha > 0$), середнє значення

$$E[X] = \frac{\alpha}{\alpha-1} k \quad (\alpha > 1).$$

Показано, що трафік даних, що циркулює в цифрових мережах, і, зокрема, у комп'ютерних мережах з комутацією пакетів, має так звані самоподібні, або фрактальні властивості. "Самоподібність" являє собою властивість процесу зберігати своє поведіння і зовнішні ознаки при розгляді в різному масштабі. Для тимчасових послідовностей масштабуємою величиною є час. Виходячи з визначення самоподібності, можна стверджувати, що часові і спектральні характеристики випадкового процесу (у нашому випадку – трафіку) при зміні масштабу усереднення будуть описуватися тими самими рівняннями, функціями, але з відповідними масштабними коефіцієнтами. Іншими словами, самоподібність якого-небудь процесу (явища) можна трактувати як інваріантність до змін чи масштабу розміру.

Математичною основою самоподібних процесів є замкнуті множини, зокрема, так звана *канторова множина*. Розглянемо структуру канторової множини. Нехай T_0 - відрізок одиничної тривалості: $T_0 [0, 1]$. Множина T_0 містить свої кінцеві крапки (є замкнутою). Виключимо з неї інтервал $(1/3, 2/3)$, тобто всі точки, що належать середньої частини відрізка T_0 , за винятком кінцевих. Множина $T_1 [[0, 1/3], [2/3, 1]]$, що залишилася, також містить свої кінцеві точки, отже, також є замкнутою. Потім викинемо з T_1 інтервали $(1/9, 2/9)$ й $(7/9, 8/9)$, а множина, що залишилася, також буде замкнутою. Означимо її T_2 . В кожному з чотирьох відрізків довжиною $(1/3)^2 = 1/9$, що лишилися, викинемо середній інтервал довжиною $(1/3)^3 = 1/27$ і т.д. Продовжуючи цей процес, одержимо спадну послідовність замкнутих множин T_n . При $n \rightarrow \infty$ з відрізка $[0, 1]$ віддаляється злічене число інтервалів перемінної (кратної) довжини. Одержуємо замкнуту безліч $T = \bigcap_{n=0}^{\infty} T_n$. На рис. 2.29 зображено кілька етапів «проріджування» відрізка T_0 .

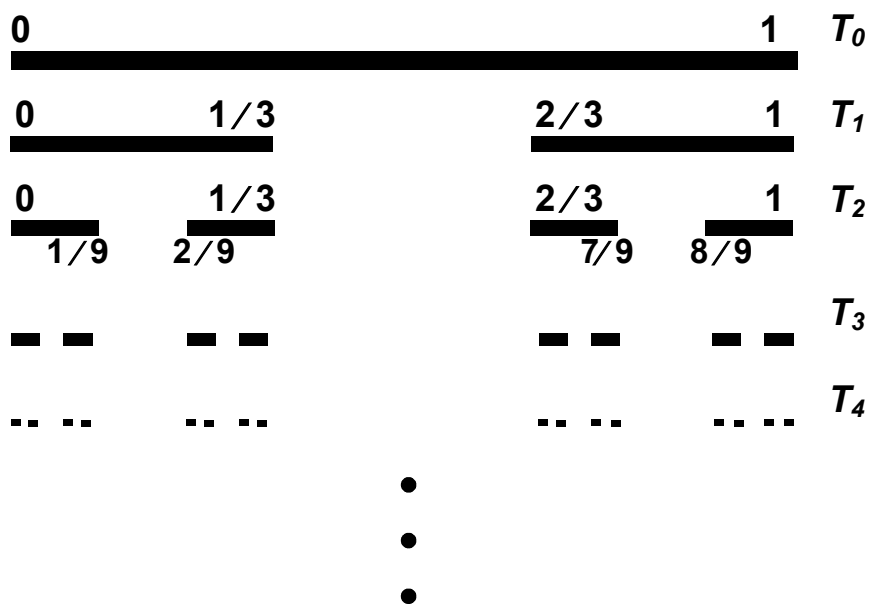


Рис. 2.29. Етапи «проріджування» відрізка T_0 .

Реальні випадкові процеси, звичайно, зберігають властивість самоподібності тільки до визначеної межі. Ця міра статистичної утсаленості процесу при багаторазовому масштабуванні визначається так званим параметром Херста чи параметром самоподібності. Випадковий процес $x(t)$ є статистично самоподібним з параметром Херста H ($0,5 \leq H \leq 1$), якщо для будь-якого $a > 0$ процес $\frac{x(at)}{a^H}$ має

ті ж статистичні характеристики, що і сам процес $x(t)$: математичне чекання $M[x(t)] = \frac{M[x(at)]}{a^H}$

; дисперсія $D[x(t)] = \frac{D[x(at)]}{a^{2H}}$; кореляційна функція $R(t, \tau) = \frac{R(at, a\tau)}{a^{2H}}$. Чим більше H , тим

довше зберігається властивість самоподібності при багаторазовому масштабуванні. При $H = 0,5$ ця властивість практично відсутня.

Кореляційні функції самоподібних процесів з великим параметром Херста загасають повільніше, ніж у звичайних випадкових процесів, причому спадання має, як правило, коливальний характер. Установлено, що спадання постійної складової кореляційної функції відбувається за законом $c_1 t^{-c_2 a}$, де c_1, c_2 — константи, a — параметр масштабу. Відповідно і спектральна щільність процесу теоретично прагне до нескінченності при частоті, що наближується до нуля.

Такі специфічні характеристики властиві не тільки трафіку даних (протоколи *TCP, FTP*), але і сигнальному трафіку (протокол *SS7*), *VBR*-відео, *Ethernet/ISDN* і деяких інших. Фізично вони обумовлені високим ступенем групування пакетів на клієнтських ділянках, у маршрутизаторах і вузлах комутації інфокомунікаційних мереж. Навіть якщо джерело породжує регулярний потік пакетів, дані до споживача доставляються серіями, які перемежуються інтервалами простою. Причинами цього є обмежена швидкість роботи мережних пристроїв, недостатній обсяг буферів і ін.

Крім того, самоподібний трафік має особливу структуру, що зберігається при багаторазовому масштабуванні. У реалізації, як правило, присутня деяка кількість викидів при відносно невеликому середньому рівні трафіку. Через такі сплески навантаження характеристики мережі також погіршуються: збільшуються втрати, затримки при проходженні через вузли мережі.

Методи розрахунку вимог до мереж нових поколінь (пропускної здатності каналів, ємності буферів і ін.), засновані на марковських моделях і формулах Ерланга чи Літгла, що з успіхом використовувалися при проектуванні телефонних мереж, можуть давати невиправдано оптимістичні рішення і приводити до недооцінки навантаження.

При самоподібній природі трафіку залежність середньої тривалості черги (відповідно, необхідного розміру буфера) q від середнього коефіцієнта використання має наступний вид: $q = \frac{\rho^{1/2(1-H)}}{(1-\rho)^{H/(1-H)}}$.

При $H=0,5$ ця формула спрощується: $q = \rho/(1-\rho)$, що являє собою класичний результат СМО з

найпростішим вхідним потоком і показово розподіленим часом обслуговування (M/M/1). Для системи з детермінованим часом обслуговування (M/D/1) класичний результат виглядає в такий спосіб:

$$q = \frac{\rho}{1-\rho} - \frac{\rho^2}{2(1-\rho)}$$

Наведемо приклад розрахунку залежності середнього числа заявок в одноканальній системі від коефіцієнту використання системи $\rho = \lambda/\mu$. Розрахунки зроблено по наведених формулах як для найпростішого (Пуассонівського) потоку заявок, так і для самоподібних потоків. Результати розрахунків наведено на рис. 2.30.

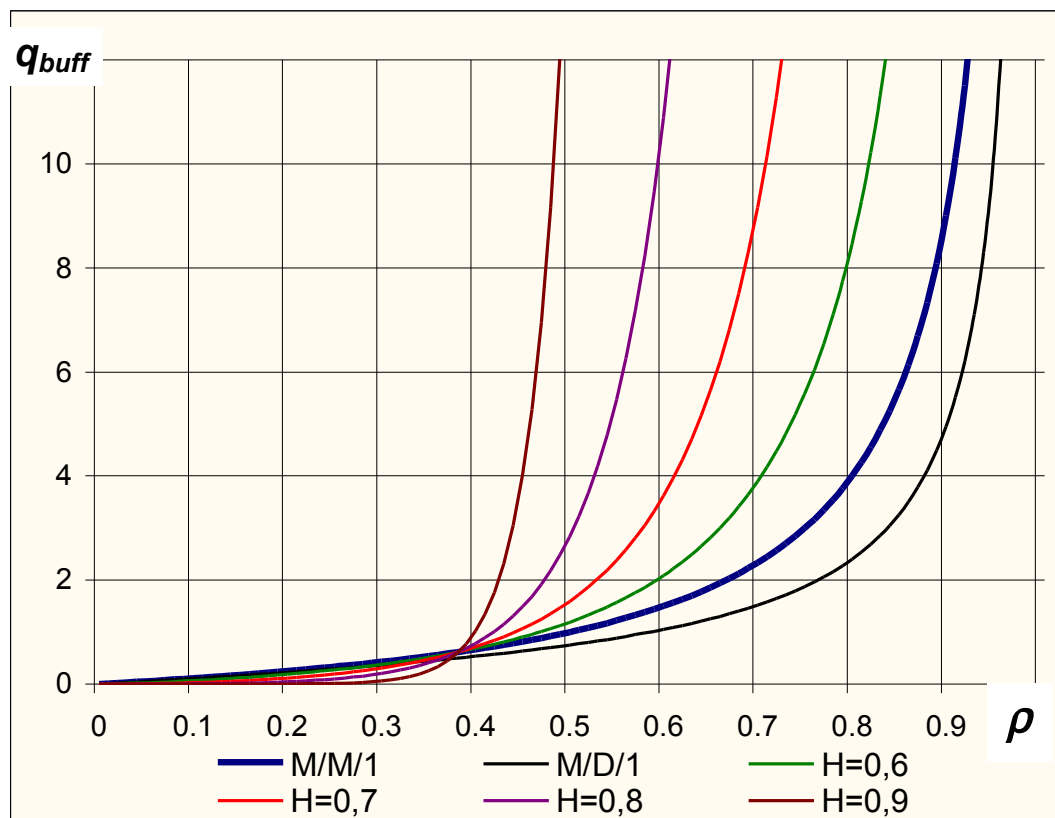


Рис. 2.30. Залежності довжини черги заявок (потребной пам'яті буфера q_{buff}) від коефіцієнта використання ρ для різних моделей вхідного трафіку.

На графіках добре видно, що для самоподібного трафіку вже при $\rho \approx 0,4$ потрібно більший ресурс пам'яті буферних пристроїв, чим для класичної моделі M/M/1, що вважається найменш сприятливим в порівнянні з іншими (наприклад, з постійним чи гауссовски розподіленим часом обслуговування). Швидкість росту необхідного обсягу пам'яті росте при збільшенні параметра Херста, що обумовлено, в основному, ступенем групування однорідних пакетів і сплесками навантаження на мережу.

Можна також зробити висновок, що просте нарощування буферної пам'яті (апаратним чи програмним способом) є малоефективним. При очікуваному збільшенні частки трафіку даних у загальному обсязі ступінь самоподібності буде збільшуватися, і залежність $\rho(q_{buff})$ буде зростати все більш різко.

2.11. Функціональні ролі комп'ютерів в мережі

2.11.1. Багат шарова модель мережі

Навіть при поверхневому аналізі роботи мережі можна зробити висновок, що обчислювальна мережа – це складний комплекс взаємозв'язаних і погоджено функціонуючих програмних і апаратних компонентів [14]. Вивчення мережі в цілому припускає знання принципів роботи окремих її елементів, таких як:

- комп'ютери;
- комунікаційне устаткування;
- операційні системи;
- мережні застосування.

Весь комплекс програмно-апаратних засобів мережі може бути описаний багат шаровою моделлю (рис. 2.31). В основі будь-якої мережі лежить апаратний шар стандартизованих комп'ютерних платформ. В даний час в мережах успішно застосовуються комп'ютери різних класів – від персональних комп'ютерів до мейнфреймів і СУПЕР-ЕВМ. Набір комп'ютерів в мережі повинен відповідати набору вирішуваних мережею задач.

Другий шар – це комунікаційне устаткування. Хоча комп'ютери і є центральними елементами обробки даних в мережах, останнім часом не менш важливу роль стали грати комунікаційні пристрої. Кабельні системи, повторювачі, мости, комутатори, маршрутизатори і модульні концентратори з допоміжних компонентів мережі перетворилися на основні разом з комп'ютерами і системним програмним забезпеченням, як по впливу на характеристики мережі, так і за вартістю. Сьогодні комунікаційний пристрій може бути складним спеціалізованим мультипроцесором, який потрібно конфігурувати, оптимізувати і адмініструвати. Вивчення принципів роботи комунікаційного устаткування вимагає знайомства з великою кількістю протоколів, які використовуються як в локальних, так і в глобальних мережах.

Прикладні програми
Операційні системи
Комунікаційне обладнання
Комп'ютерна платформа

Рис. 2.31. Багат шарова модель мережі.

Третім шаром, створюючим програмну *платформу* мережі, є операційні системи (ОС). Від того, які концепції управління локальними і розподіленими ресурсами покладені в основу мережної ОС, залежить ефективність роботи всієї мережі. При проектуванні мережі важливо враховувати, наскільки легко операційна система може взаємодіяти з іншими ОС мережі, який вона забезпечує рівень безпеки і захищеності даних, до якого ступеня дозволяє нарощувати число користувачів, чи можна перенести її на *комп'ютер* іншого типу і багато інших міркувань.

Самий верхній шар мережних засобів утворюють різні *мережні програми*, такі як мережні *бази даних*, поштові системи, засоби архівації даних, системи автоматизації колективної роботи і т.д. Дуже важливо представляти діапазон можливостей, що надаються програмами для різних областей застосування, а також знати, наскільки вони сумісні з іншими *мережними програмами* і операційними системами.

Обчислювальна мережа – це багат шаровий комплекс взаємозв'язаних і погоджено функціонуючих *програмних і апаратних компонентів: комп'ютерів, комунікаційного устаткування, операційних систем, мережних програм*. Залежно від того, які розподілені функції між комп'ютерами мережі, вони можуть виступати в трьох різних ролях:

комп'ютер грає роль виділеного сервера мережі (рис. 2.32);

- комп'ютер грає роль вузла-клієнта (рис. 2.33);
- комп'ютер є одноранговим вузлом (рис. 2.34).

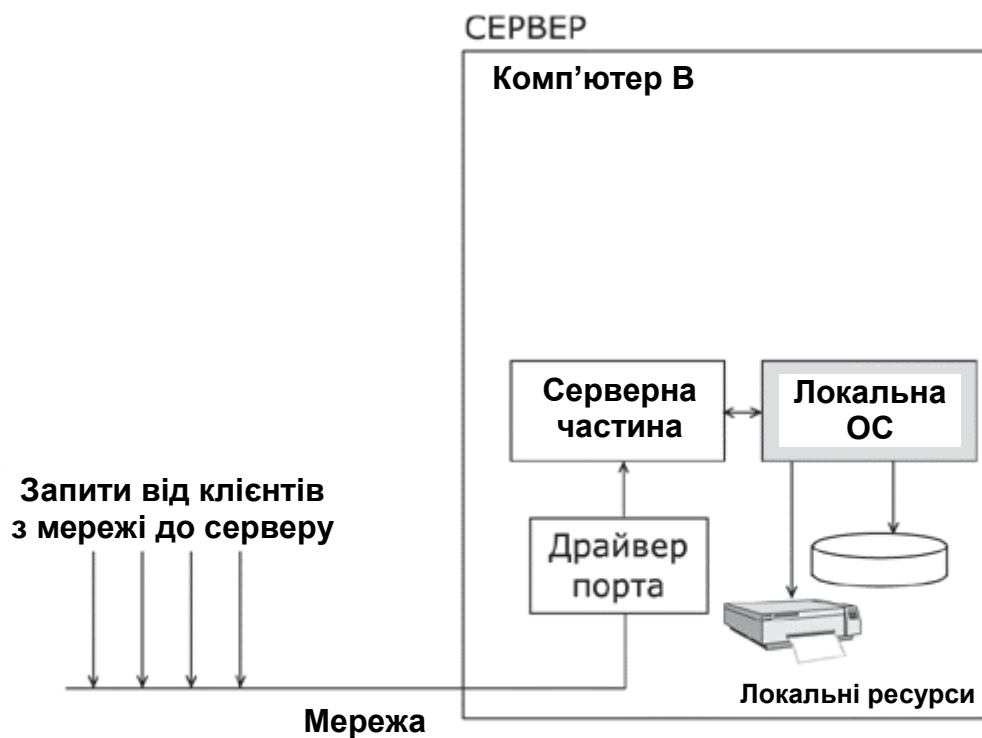


Рис. 2.32. Комп'ютер, що займається виключно обслуговуванням запитів інших комп'ютерів.

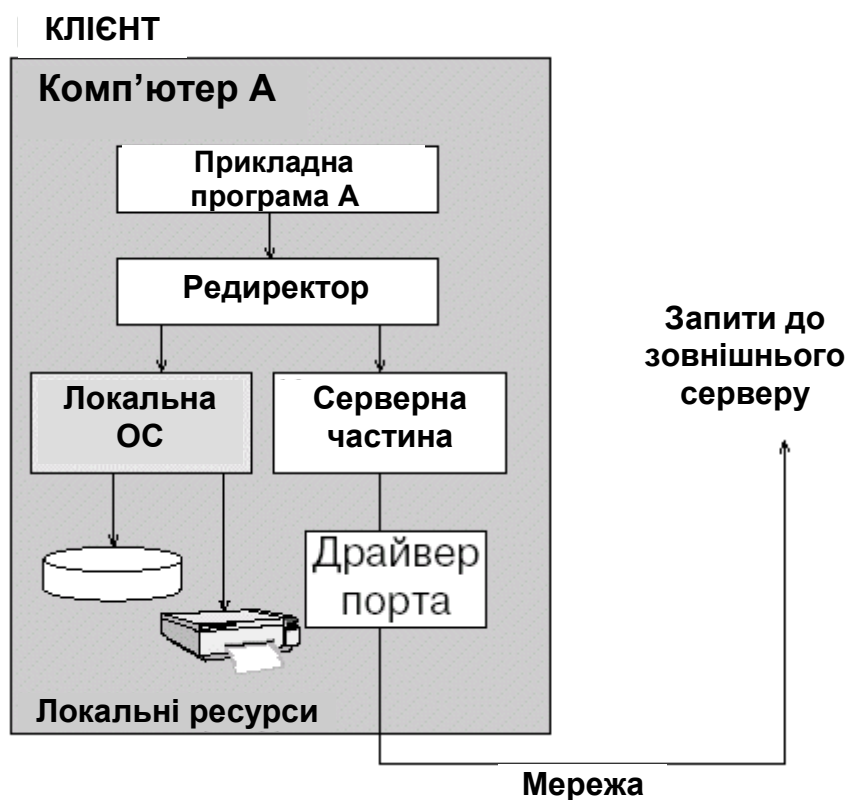


Рис. 2.33. Комп'ютер, який звертається із запитом до ресурсів іншої машини.

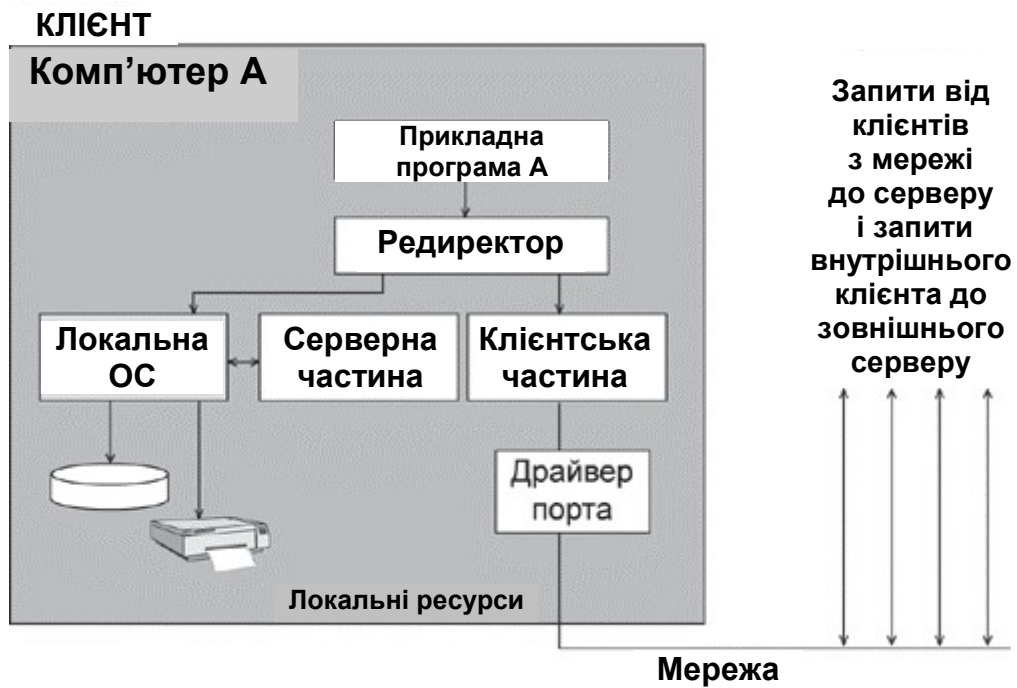


Рис. 2.34. Комп'ютер, що суміщає функції клієнта і сервера.

Очевидно, що мережа не може складатися тільки з клієнтських або тільки з серверних вузлів. Мережа може бути побудована по одній з трьох схем:

- мережа на основі однорангових вузлів – однорангова мережа;
- мережа на основі клієнтів і серверів – мережа з виділеними серверами;
- мережа, що включає вузли всіх типів, – гібридна мережа.

Кожна з цих схем має свої переваги і недоліки, які визначають їх області застосування.

2.11.2. Однорангові мережі

У однорангових мережах (рис. 2.35) всі комп'ютери рівні в можливостях доступу до ресурсів один одного. Кожен користувач може по своєму бажанню оголосити який-небудь ресурс свого комп'ютера таким, що розділяється, після чого інші користувачі можуть з ним працювати. У однорангових мережах на всіх комп'ютерах встановлюється така операційна система, яка надає всім комп'ютерам в мережі потенційно рівні можливості. Мережні операційні системи такого типу називаються одноранговими ОС. Очевидно, що однорангові ОС повинні включати як серверні, так і клієнтські компоненти мережних служб (на малюнку вони позначені буквами, відповідно, С і К). Прикладами однорангових ОС можуть служити *LANtastic*, *Personal Ware*, *Windows for Workgroups*, *Windows NT Workstation*, *Windows 95/98*.

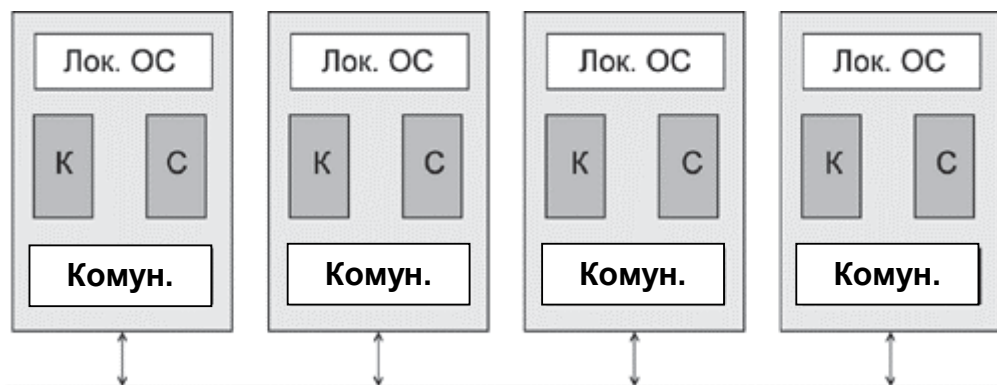


Рис. 2.35. Однорангова мережа (словом "Комун." позначені комунікаційні засоби).

При потенційному рівноправ'ї всіх комп'ютерів в одноранговій мережі часто виникає функціональна несиметрія. Зазвичай деякі користувачі не бажають надавати свої ресурси для сумісного доступу. У такому разі серверні можливості їх операційних систем не активізуються, і комп'ютери грають роль "чистих" клієнтів (на малюнку невживані компоненти ОС зображені затемненими).

В той же час адміністратор може закріпити за деякими комп'ютерами мережі тільки функції, які пов'язані з обслуговуванням запитів від решти комп'ютерів, перетворивши їх таким чином в "чисті" сервери, за якими користувачі не працюють. У такій конфігурації однорангові мережі стають схожими на мережі з виділеними серверами, але це тільки зовнішня схожість – між цими двома типами мереж залишається істотна відмінність. Спочатку в однорангових мережах відсутня спеціалізація ОС залежно від того, яку роль грає комп'ютер – клієнта або сервера. Зміна ролі комп'ютера в одноранговій мережі досягається за рахунок того, що функції серверної або клієнтської частин просто не використовуються.

Однорангові мережі простіше в розгортанні і експлуатації; по цій схемі організовується робота в невеликих мережах, в яких кількість комп'ютерів не перевищує 10-20. В цьому випадку немає необхідності в застосуванні централізованих засобів адміністрування – декільком користувачам неважко домовитися між собою про перелік ресурсів, що розділяються, і паролі доступу до них. Проте у великих мережах засоби централізованого адміністрування, зберігання і обробки даних, а особливо захисту даних необхідні. Такі можливості легше забезпечити в мережах з виділеними серверами.

2.11.3. Мережі з виділеним сервером

У мережах з виділеними серверами (рис. 2.36) використовуються спеціальні варіанти мережних ОС, які оптимізовані для роботи в ролі серверів і називаються серверними ОС. Призначені для користувача комп'ютери в таких мережах працюють під управлінням клієнтських ОС.

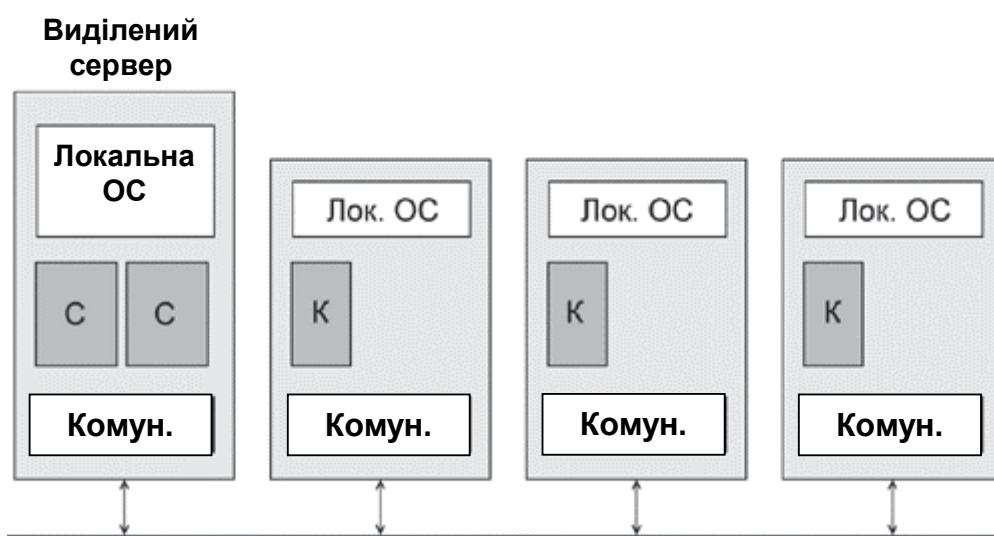


Рис. 2.36. Мережа з виділеним сервером.

Спеціалізація операційної системи для роботи в ролі сервера є природним способом підвищення ефективності серверних операцій. А необхідність такого підвищення часто відчувається вельми гостро, особливо у великій мережі. При існуванні в мережі сотень або навіть тисяч користувачів інтенсивність запитів до ресурсів, що розділяються, може бути дуже значною, і сервер повинен справлятися з цим потоком запитів без великих затримок. Очевидним рішенням цієї проблеми є використання в якості серверу комп'ютера з могутньою апаратною платформою і операційною системою, оптимізованою для серверних функцій.

Чим менше функцій виконує ОС, тим більше ефективно можна їх реалізувати, тому для оптимізації серверних операцій розробники ОС вимушені ущемляти деякі інші її функції, причому іноді навіть повністю відмовлятися від них. Одним з яскравих прикладів такого підходу є серверна ОС NetWare. Її розробники поставили перед собою мету оптимізувати виконання файлового сервісу і сервісу друку. Для цього вони повністю виключили з системи багато елементів, важливих для універсальної ОС, зокрема, графічний інтерфейс користувача, підтримку універсальних програм, захист програм мультимедійного режиму один від одного, механізм віртуальної пам'яті. Все це дозволило добитися унікальної швидкості файлового доступу і вивело *NetWare* в лідери серверних ОС на довгий час.

Проте дуже вузька спеціалізація деяких серверних ОС є одночасно і їх слабкою стороною. Так, відсутність в *NetWare 4* універсального інтерфейсу програмування і засобів захисту програм, що не дозволило використовувати цю ОС як середовище для виконання застосувань, приводить до необхідності застосування в мережі інших серверних ОС в тих випадках, коли потрібне виконання функцій, відмінних від файлового сервісу і сервісу друку. Тому розробники багатьох серверних операційних систем відмовляються від функціональної обмеженості і включають до складу серверних ОС всі компоненти, які дозволяють задіювати їх в якості універсальних серверів і навіть клієнтських ОС. Такі серверні ОС забезпечуються розвиненим графічним призначенням для користувача інтерфейсом і підтримують універсальні API. Це

зближує їх з одноранговими операційними системами, але існує декілька відмінностей, які дозволяють віднести їх саме до класу серверних ОС:

- підтримка могутніх апаратних платформ, зокрема мультипроцесорних;
- підтримка великого числа одночасно виконуваних процесів і мережних з'єднань;
- включення до складу ОС компонентів централізованого адміністрування мережі (наприклад, довідкової служби або служби аутентифікації і авторизації користувачів мережі);
- ширший набір мережних служб.

Клієнтські операційні системи в мережах з виділеними серверами зазвичай звільняються від серверних функцій, що значно спрощує їх організацію. Розробники клієнтських ОС приділяють основну увагу призначеному для користувача інтерфейсу і клієнтським частинам мережних служб. Найбільш прості клієнтські ОС підтримують тільки базові мережні служби, зазвичай файлової і служби друку. В той же час існують так звані універсальні клієнти, які підтримують широкий набір клієнтських частин, що дозволяють їм працювати практично зі всіма серверами мережі.

Багато компаній, які розробляють мережні ОС, випускають дві версії однієї і тієї ж операційної системи. Одна версія призначена для роботи в якості серверної ОС, а інша – для роботи на клієнтській машині. Ці версії найчастіше засновані на одному і тому ж базовому коді, але відрізняються набором служб і утиліт, а також параметрами конфігурації, зокрема встановлюваними за умовчанням і невіддатливими зміни.

Наприклад, операційна система *Windows NT* випускалася у версії для робочої станції – *Windows NT Workstation* і у версії для виділеного сервера – *Windows NT Server*. Обидва ці варіанти операційної системи включають клієнтські і серверні частини багатьох мережних служб.

Так, ОС *Windows NT Workstation*, окрім виконання функцій мережного клієнта, може надавати мережним користувачам файловий сервіс, сервіси друку, віддаленого доступу та інші, а, отже, може служити основою для однорангової мережі. З іншого боку, ОС *Windows NT Server* містить всі необхідні засоби, які дозволяють задіювати комп'ютер як клієнтську робочу станцію. Під управлінням ОС *Windows NT Server* локально запускаються прикладні програми, які можуть зажадати виконання клієнтських функцій ОС при появі запитів до ресурсів інших комп'ютерів мережі. *Windows NT Server* має такий же розвинений графічний інтерфейс, як і *Windows NT Workstation*, що дозволяє з рівним успіхом застосовувати ці ОС для інтерактивної роботи користувача або адміністратора.

Проте версія *Windows NT Server* має більше можливостей для надання ресурсів свого комп'ютера іншим користувачам мережі, оскільки може виконувати ширший набір функцій, підтримувати більшу кількість одночасних з'єднань з клієнтами, реалізовувати централізоване управління мережею, має розвиненіші засоби захисту. Тому рекомендується застосовувати *Windows NT Server* як ОС для виділених серверів, а не клієнтських комп'ютерів.

2.11.4. Гібридна мережа

У великих мережах разом з відносинами клієнт-сервер зберігається необхідність і в однорангових зв'язках, тому такі мережі найчастіше будуються за гібридною схемою (рис. 2.37).

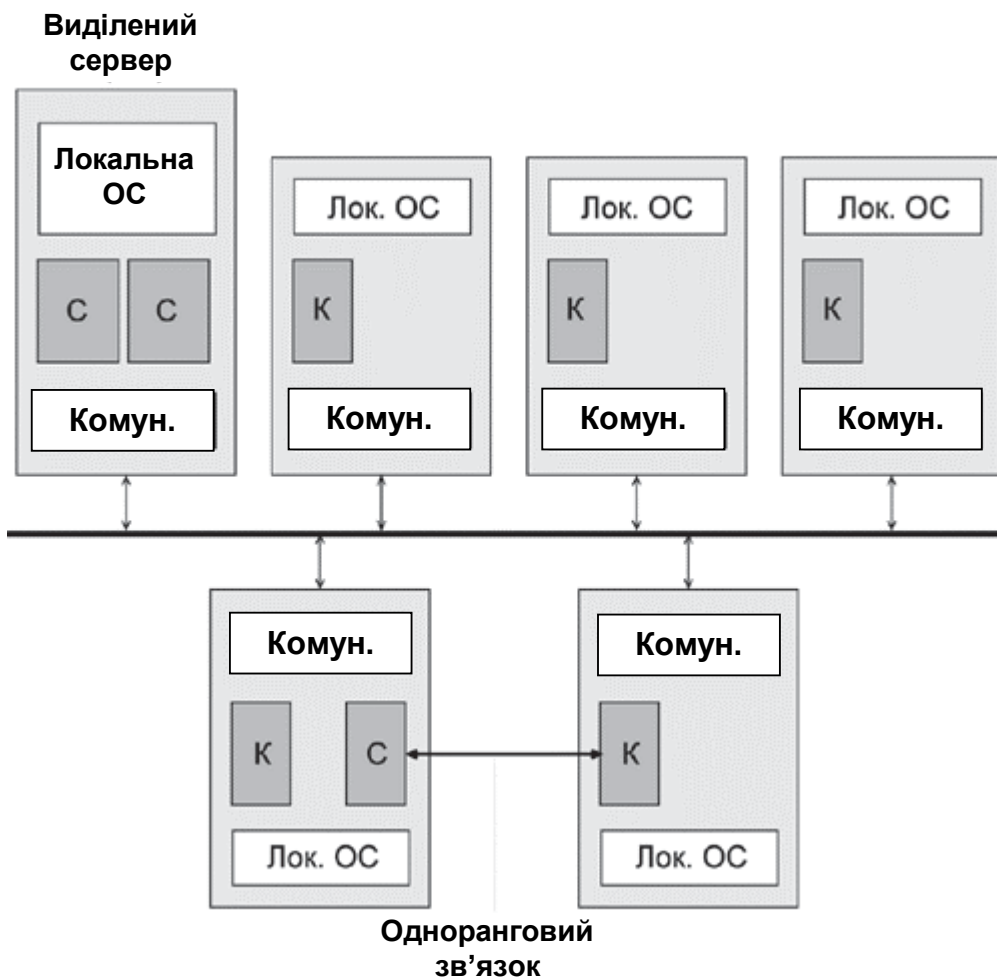


Рис.2.37. Гібридна мережа.

2.11.5. Мережні служби і операційна система

Для кінцевого користувача мережа – це не комп'ютери, кабелі і концентратори і навіть не інформаційні потоки, для нього мережа – це, перш за все, набір мережних служб, за допомогою яких він дістає можливість проглянути список наявних в мережі комп'ютерів, прочитати віддалений файл, роздрукувати документ на "чужому" принтері або послати поштове повідомлення. Саме сукупність можливостей, що надаються, – наскільки широкий їх вибір, наскільки вони зручні, надійні і безпечні – визначає для користувача зовнішність тієї або іншої мережі.

Окрім власне обміну даними, мережні служби повинні вирішувати та інші, більш специфічні, завдання, наприклад, завдання, пов'язані з розподіленою обробкою даних. До таких завдань відносяться забезпечення несуперечності декількох копій даних, розмішених на різних машинах (служба реплікації), або організація виконання одного завдання паралельно на декількох машинах мережі (служба виклику віддалених процедур). Серед мережних служб можна виділити адміністративні, тобто такі, які в основному орієнтовані не на простого користувача, а на адміністратора, і служать для забезпечення правильної роботи мережі в цілому. Служба адміністрування призначених для користувача облікових записів, яка дозволяє адміністраторові вести загальну базу даних про користувачів мережі, система моніторингу мережі, що дозволяє захоплювати і аналізувати мережний трафік, служба безпеки, у функції якої може входити, крім іншого, виконання процедури логічного входу з подальшою перевіркою пароля, – все це приклади адміністративних служб.

Реалізація мережних служб здійснюється програмними засобами. Всі мережні служби побудовані в архітектурі "клієнт-сервер".

Основні служби – файлова служба і служба друку – зазвичай надаються мережною операційною системою, а допоміжні, наприклад, служба баз даних, зв'язку факсиміле або передачі голосу, – системними мережними застосуваннями або утилітами, що працюють в тісному контакті з мережною ОС. Взагалі кажучи, розподіл служб між ОС і утилітами достатньо умовно і міняється залежно від реалізації ОС.

При розробці мережних служб доводиться вирішувати задачі, властиві будь-яким розподіленим застосуванням: визначення протоколу взаємодії між клієнтською і серверною частинами, розподіл функцій між ними, вибір схеми адресації застосувань і т.д.

Одним з головних показників якості мережної служби є її зручність. Для одного і того ж ресурсу може бути розроблено декілька служб, по-різному вирішальних загалом одні і те ж завдання. Відмінності можуть полягати в продуктивності або в рівні зручності послуг, що надаються. Наприклад, файлова служба може бути заснована на використанні команди передачі файлу з одного комп'ютера в іншій по імені файлу, а це вимагає від користувача знання імені потрібного файлу. Та ж файлова служба може бути реалізована і так, що користувач вмонтовує віддалену файлову систему до локального каталогу, а далі звертається до віддалених файлів як до своїх власних, що набагато зручніше. Якість мережної служби залежить і від якості призначеного для користувача інтерфейсу – інтуїтивної зрозумілості, наочності, раціональності.

При визначенні ступеня зручності ресурсу, що розділяється, часто використовують термін "прозорість". Прозорий доступ – це такий доступ, при якому користувач не помічає, де розташований потрібний йому ресурс – на його комп'ютері або на віддаленому. Після того, як він змонтував віддалену файлову систему в своє дерево каталогів, доступ до віддалених файлів стає для нього абсолютно прозорим. Сама операція монтування також може мати різний ступінь прозорості – в мережах з меншою прозорістю користувач повинен знати і задавати в команді ім'я комп'ютера, на якому зберігається віддалена файлова система, в мережах з більшим ступенем прозорості відповідний програмний компонент мережі проводить пошук томів файлів, що розділяються, незалежно від місць їх зберігання, а потім надає їх користувачеві в зручному для нього вигляді, наприклад, у вигляді списку або набору піктограм.

Для забезпечення прозорості важливий спосіб адресації (іменування) мережних ресурсів, що розділяються. Імена мережних ресурсів, що розділяються, не повинні залежати від їх фізичного розташування на тому або іншому комп'ютері. У ідеалі користувач не повинен нічого міняти в своїй роботі, якщо адміністратор мережі перемістив том або каталог з одного комп'ютера на іншій. Сам адміністратор і мережна операційна система мають інформацію про розташування файлових систем, але від користувача вона прихована. Такий ступінь прозорості поки рідко зустрічається в мережах, – зазвичай для діставання доступу до ресурсів певного комп'ютера спочатку доводиться встановлювати з ним логічне з'єднання. Такий підхід застосовується, наприклад, в мережах *Windows NT*.

2.12. Структуризація мереж

2.12.1. Причини структуризації транспортної інфраструктури мереж

У мережах з порівняно невеликою (10-50) кількістю комп'ютерів найчастіше використовується одна з типових топологій – "загальна шина", "кілець", "зірка" або повнозв'язна мережа. Всі перераховані топології володіють властивістю однорідності, тобто всі комп'ютери в такій мережі мають однакові права відносно доступу до інших комп'ютерів (за винятком центрального комп'ютера при з'єднанні "зірка"). Така однорідність структури спрощує процедуру нарощування числа комп'ютерів, полегшує обслуговування і експлуатацію мережі.

Проте при побудові великих мереж однорідна структура зв'язків перетворюється з переваги в недолік [15]. У таких мережах використання типових структур породжує різні обмеження, найважливішими з яких є:

- обмеження на довжину зв'язку між вузлами;
- обмеження на кількість вузлів в мережі;
- обмеження на інтенсивність трафіку, який генерують вузли мережі.

Наприклад, технологія *Ethernet* на тонкому коаксіальному кабелі дозволяє використовувати кабель завдовжки не більше 185 метрів, до якого можна підключити не більше 30 комп'ютерів. Проте, якщо комп'ютери інтенсивно обмінюються інформацією, іноді доводиться знижувати число підключених до кабелю машин до 20, а то і до 10, щоб кожному комп'ютеру діставалася прийнятна частка загальної пропускної спроможності мережі.

Для зняття цих обмежень використовуються особливі методи структуризації мережі і спеціальне устаткування – повторювачі, концентратори, мости, комутатори, маршрутизатори. Такого роду устаткування також називають комутаційним, маючи на увазі, що з його допомогою окремі сегменти мережі взаємодіють між собою. Розрізняють:

1. Топологію фізичних зв'язків (фізичну структуру мережі). В цьому випадку конфігурація фізичних зв'язків визначається електричними з'єднаннями комп'ютерів, тобто ребрам графа відповідають відрізки кабелю, що зв'язують пари вузлів.

2. Топологію логічних зв'язків (логічну структуру мережі). В цьому випадку в якості логічних зв'язків виступають маршрути передачі даних між вузлами мережі, які утворюються шляхом відповідної настройки комунікаційного устаткування.

2.12.2. Фізична структуризація мережі

Простіший з комутаційних пристроїв – це повторювач (repeater). Він використовується для фізичного з'єднання різних сегментів кабелю локальної мережі з метою збільшення загальної довжини мережі. Повторитель передає сигнали, що приходять з одного сегменту мережі, в інші її сегменти (рис. 2.38). Повторювач дозволяє подолати обмеження на довжину ліній зв'язку за рахунок поліпшення якості переданого сигналу – відновлення його потужності і амплітуди, поліпшення фронтів і т.п.

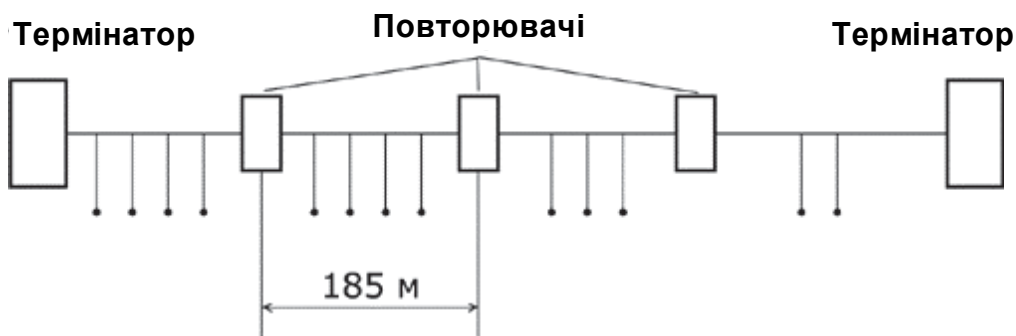


Рис. 2.38. Повторювач дозволяє збільшити довжину мережі *Ethernet*.

Повторювач, який має декілька портів і сполучає декілька фізичних сегментів, часто називають концентратором (concentrator) або хабом (*hub*). Ці назви (*hub* – основа, центр діяльності) відображають той факт, що в даному пристрої зосереджені всі зв'язки між сегментами мережі. Використання концентраторів характерне практично для всіх базових технологій локальних мереж – *Ethernet*, *ArcNet*, *Token Ring*, *FDDI*, *Fast Ethernet*, *Gigabit Ethernet*.

Потрібно підкреслити, що в роботі будь-яких концентраторів багато загального – вони повторюють сигнали, що прийшли з одного з портів на інші порти. Різниця полягає в тому, на яких саме портах повторюються вхідні сигнали. Так, концентратор *Ethernet* повторює вхідні сигнали на всіх своїх портах, крім того, з якого сигнали поступають (рис. 2.39).

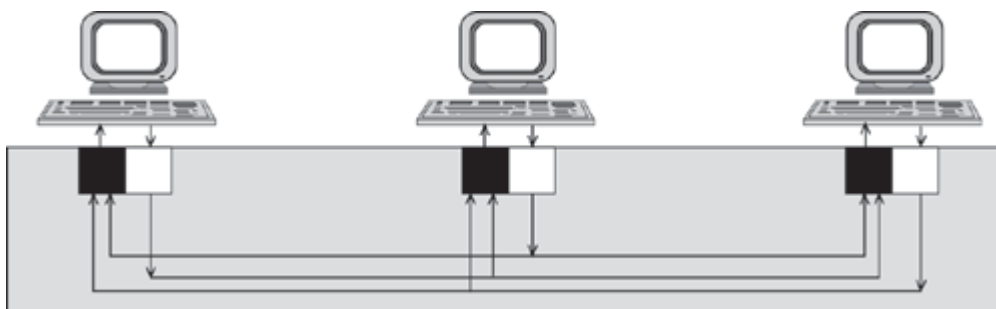


Рис. 2.39. Концентратор *Ethernet*.

На виході концентратора *Token Ring* (рис. 2.40) вхідні сигнали, що поступають з деякого порту, повторюються тільки на одному порті – на тому, до якого підключений наступний в кільці комп'ютер.

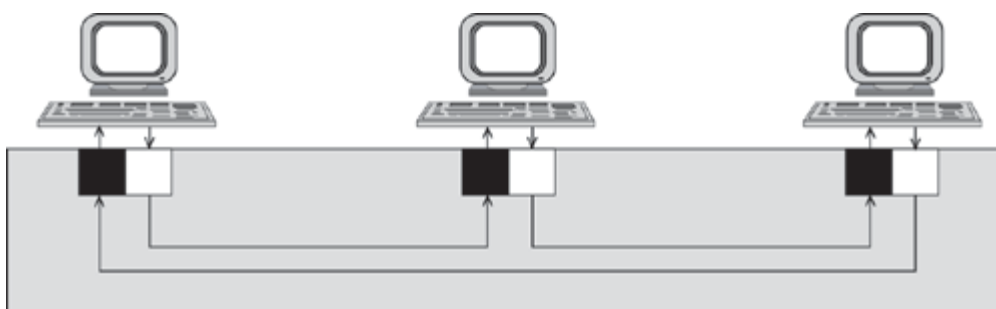


Рис. 2.40. Концентратор *Token Ring*.

Додавання в мережу концентратора завжди змінює фізичну топологію мережі, але при цьому залишає без змін її логічну топологію.

Як вже було сказано, під фізичною топологією розуміється конфігурація зв'язків, утворених окремими частинами кабелю, а під логічною – конфігурація інформаційних потоків між комп'ютерами мережі. У багатьох випадках фізична і логічна топології мережі співпадають. Наприклад, мережа, представлена на рис. 2.41а, має фізичну топологію "кільце". Комп'ютери такої мережі дістають доступ до кабелів кільця за рахунок передачі один одному спеціального кадру – маркера, причому цей маркер також передається послідовно від комп'ютера до комп'ютера в тому ж порядку, в якому комп'ютери утворюють фізичне кільце, тобто комп'ютер А передає маркер комп'ютеру В, комп'ютер В – комп'ютеру С і т.д.

Мережа, показана на рис. 2.41б, демонструє приклад неспівпадання фізичної і логічної топології. Фізично комп'ютери сполучені по топології "загальна шина". Доступ же до шини відбувається не по алгоритму випадкового доступу, який вживається в технології *Ethernet*, а шляхом передачі маркера в

кільцевому порядку: від комп'ютера А – комп'ютеру В, від комп'ютера В – комп'ютеру С і т.д. В цьому випадку порядок передачі маркера вже не повторює фізичні зв'язки, а визначається логічною конфігурацією драйверів мережних адаптерів. Ніщо не заважає налагодити мережні адаптери і їх драйвери так, щоб комп'ютери утворили кільце в іншому порядку, наприклад: В, А, С... При цьому фізична структура мережі не змінюється.

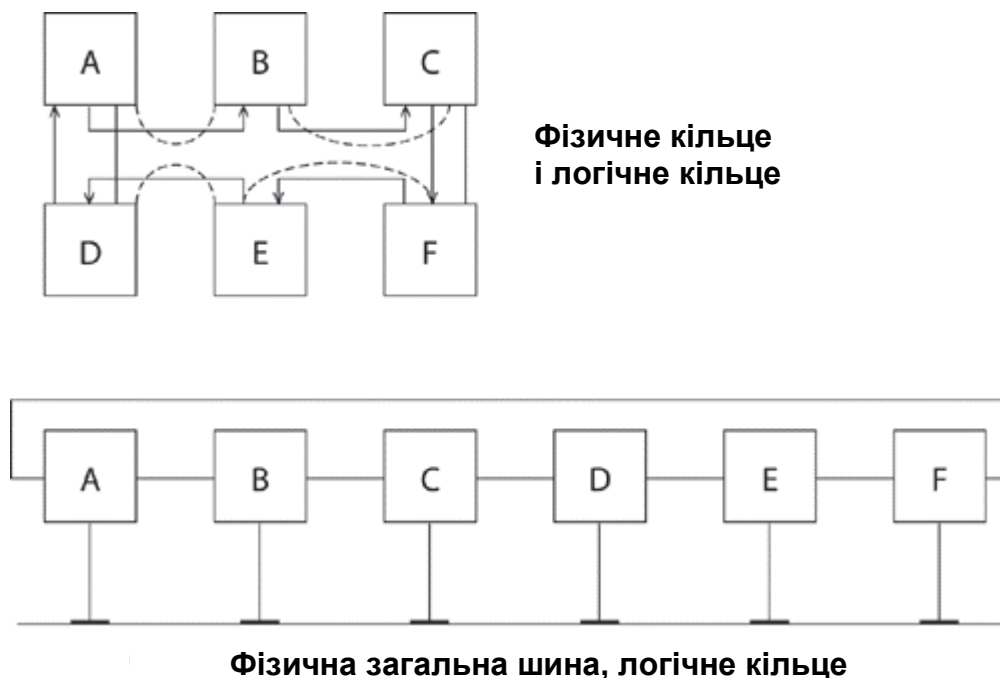


Рис. 2.41.

- а) логічна і фізична структури мережі співпадають;
- б) логічна структура не співпадає з фізичною.

Іншим прикладом неспівпадання фізичної і логічної топологій мережі є вже розглянута мережа на рис. 2.39. Концентратор *Ethernet* підтримує в мережі фізичну топологію "зірка". Проте логічна топологія мережі залишилася без змін – це "загальна шина". Оскільки концентратор повторює дані, що прийшли з будь-якого порту, на решті всіх портів, то вони з'являються на всіх фізичних сегментах мережі одночасно, як і в мережі з фізичною загальною шиною. Логіка доступу до мережі не міняється: всі компоненти алгоритму випадкового доступу – визначення незайнятості середовища, захоплення середовища, розпізнавання і відробіток колізій залишаються в силі.

Фізична структуризація мережі за допомогою концентраторів корисна не тільки для збільшення відстані між вузлами мережі, але і для підвищення її надійності. Наприклад, якщо який-небудь комп'ютер мережі *Ethernet* з фізичною загальною шиною через збій починає безперервно передавати дані по загальному кабелю, то вся мережа виходить з ладу, і залишається тільки одне – вручну від'єднати мережний адаптер цього комп'ютера від кабелю. У мережі *Ethernet*, побудованій з використанням концентратора, ця проблема може бути вирішена автоматично – концентратор відключає свій порт, якщо виявляє, що приєднаний до нього вузол дуже довго монополює мережу. Концентратор може блокувати некоректно працюючий вузол і в інших випадках, виконуючи роль деякого управляючого вузла.

2.12.3. Логічна структуризація мережі

Фізична структуризація мережі корисна у багатьох відношеннях, проте у ряді випадків, що зазвичай відносяться до мереж великого і середнього розміру, без логічної структуризації мережі обійтися неможливо. Найбільш важливою проблемою, що не вирішується шляхом фізичної структуризації, залишається проблема перерозподілу переданого трафіку між різними фізичними сегментами мережі.

У великій мережі природним чином виникає неоднорідність інформаційних потоків: мережа складається з безлічі підмереж робочих груп, відділів, філіалів підприємства і інших адміністративних утворень. У одних випадках найбільш інтенсивний обмін даними спостерігається між комп'ютерами, що належать одній підмережі, і лише невелика частина звернень відбувається до ресурсів комп'ютерів, що знаходяться поза локальними робочими групами. На інших підприємствах, особливо там, де є централізовані сховища корпоративних даних, які активно використовуються всіма співробітниками підприємства, спостерігається зворотна ситуація: інтенсивність зовнішніх звернень вище інтенсивності обміну між "сусідніми" машинами. Але незалежно від того, як розподіляються зовнішній і внутрішній трафік, для підвищення ефективності роботи мережі неоднорідність інформаційних потоків необхідно враховувати.

Мережа з типовою топологією ("шина", "кільце", "зірка"), в якій всі фізичні сегменти розглядаються як одне середовище, що розділяється, виявляється неадекватній структурі інформаційних потоків у великій мережі. Наприклад, в мережі із загальною шиною взаємодія будь-якої пари комп'ютерів займає її на весь час обміну, тому при збільшенні числа комп'ютерів в мережі шина стає вузьким місцем. Комп'ютери одного відділу вимушені чекати, коли завершить обмін пари комп'ютерів іншого відділу.

Цей випадок ілюструє рис. 2.42. На цьому рисунку показана *мережа*, побудована з використанням концентраторів. Хай комп'ютер А, що знаходиться в одній *підмережі* з комп'ютером В, посилає йому дані. В результаті фізичної структуризації логічна структура на змінилася. Не дивлячись на розгалужену фізичну структуру *мережі*, концентратори поширюють будь-який кадр по всіх її сегментах. Тому кадр, що посилається комп'ютером А комп'ютеру В, хоч і не потрібний комп'ютерам відділів 2 і 3, відповідно до логіки роботи концентраторів поступає на ці сегменти теж (на малюнку кадр, посланий комп'ютером А, показаний у вигляді заштрихованого кружка, який повторюється на всіх мережних інтерфейсах даної *мережі*). І до тих пір, поки комп'ютер В не отримає адресований йому кадр, жоден з комп'ютерів цієї *мережі* не зможе передавати дані.

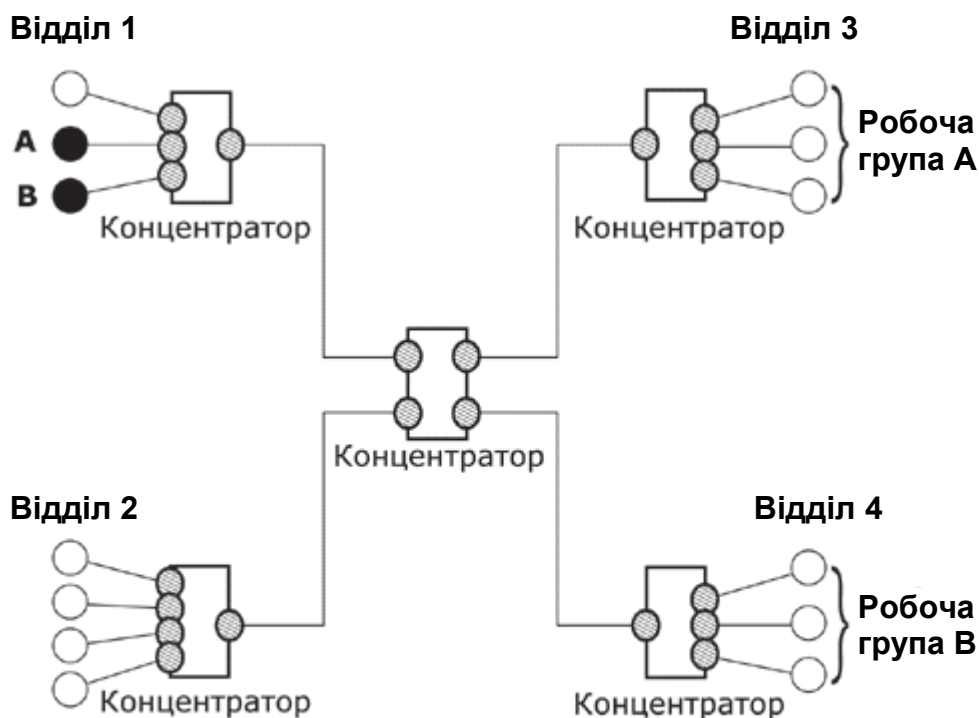


Рис. 2.42. Фізична структуризація на основі концентраторів.

Така ситуація виникає через те, що логічна структура даної *мережі* залишилася однорідною – вона ніяк не враховує можливість локальної обробки трафіку усередині відділу і надає всім парам комп'ютерів рівні можливості по обміну інформацією (рис. 2.43).

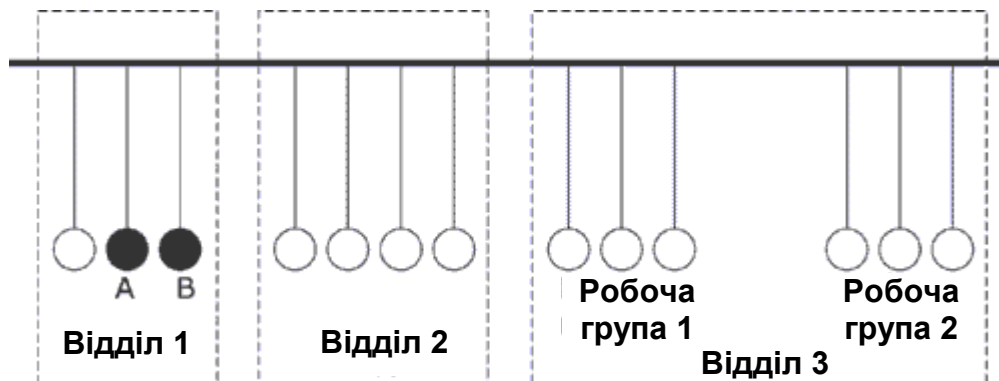


Рис. 2.43. Логічна структура мережі відповідає "загальній шині".

Для вирішення проблеми доведеться відмовитися від ідеї єдиного однорідного середовища, що розділяється. Наприклад, в розглянутому вище прикладі бажано було б зробити так, щоб кадри, які

передають комп'ютери відділу 1, виходили б за межі цієї частини *мережі* в тому і лише в тому випадку, якщо ці кадри направлені якому-небудь комп'ютеру з інших відділів. З іншого боку, в мережу кожного з відділів повинні потрапляти тільки ті кадри, які адресовані вузлам цієї *мережі*. При такій організації роботи *мережі* її продуктивність істотно підвищиться, оскільки комп'ютери одного відділу не простоюватимуть в той час, коли обмінюються даними комп'ютери інших відділів.

Неважко відмітити, що в запропонованому рішенні ми відмовилися від ідеї загального середовища, що розділялося, в межах всієї *мережі*, хоч і залишили її в межах кожного відділу (рис. 2.44). Пропускна спроможність ліній зв'язку між відділами не повинна співпадати з пропускну спроможністю середовища усередині відділів. Якщо трафік між відділами складає тільки 20% трафіку усередині відділу (як вже наголошувалося, ця величина може бути іншою), то і пропускна спроможність ліній зв'язку і комунікаційного устаткування, що сполучає відділи, може бути значно нижче за внутрішній трафік *мережі* відділу.

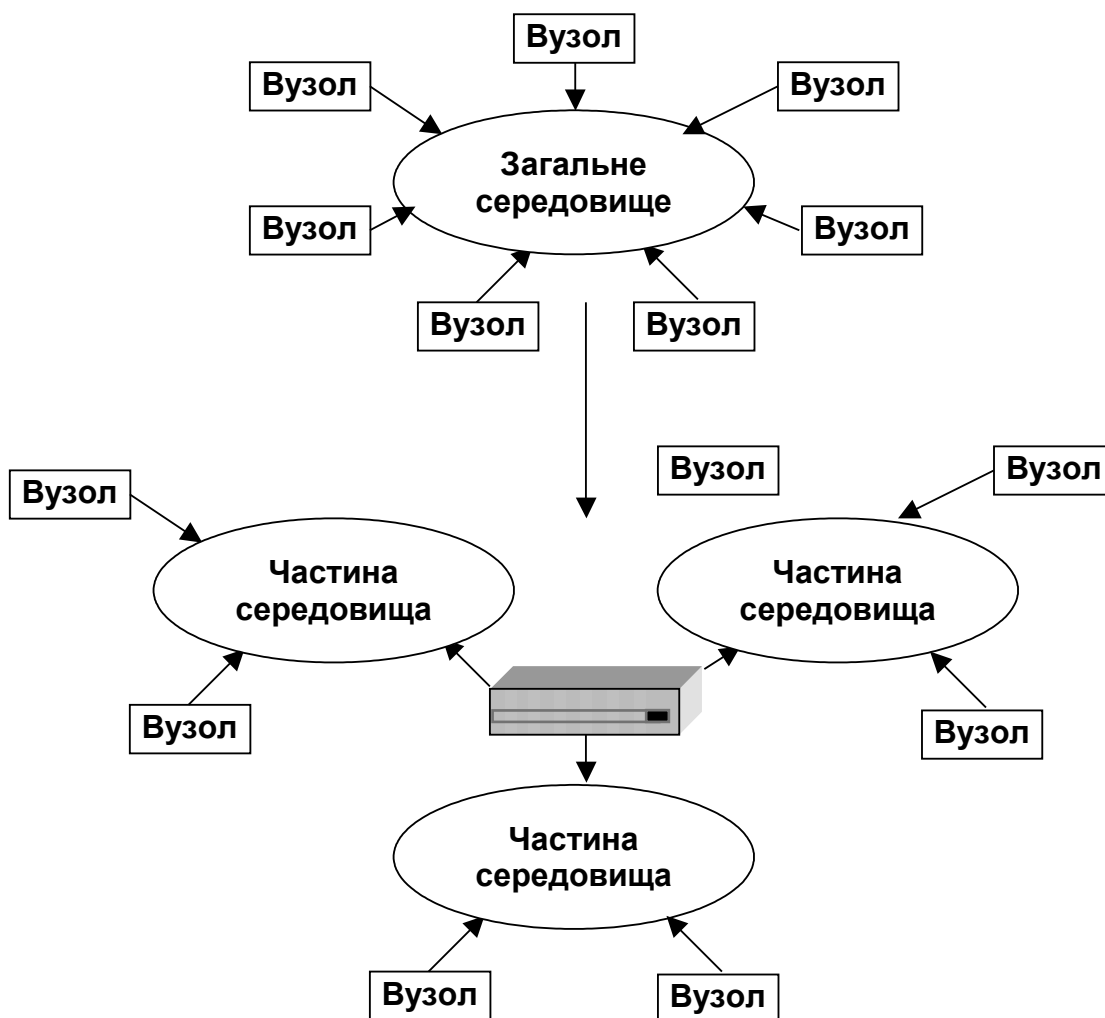


Рис. 2.44. Відмова від єдиного середовища, що розділяється.

Розповсюдження трафіку, призначеного для комп'ютерів деякого сегменту мережі, тільки в межах цього сегменту, називається локалізацією трафіку. Логічна структуризація мережі – це процес розбиття *мережі* на сегменти з локалізованим трафіком. Для логічної структуризації мережі використовуються комунікаційні пристрої:

- мости;
- комутатори;
- маршрутизатори;
- шлюзи.

Міст (*bridge*) ділить середовище передачі мережі, яка розділяється, на частини (часто звані логічними сегментами), передаючи інформацію з одного сегменту в іншій тільки в тому випадку, якщо така передача дійсно необхідна, тобто якщо адреса комп'ютера призначення належить іншій підмережі. Тим самим міст ізолює трафік однієї підмережі від трафіку іншої, підвищуючи загальну продуктивність передачі даних в мережі. Локалізація трафіку не тільки економить пропускну спроможність, але і зменшує можливість

несанкціонованого доступу до даних, оскільки кадри не виходять за межі свого сегменту, і зловмисникові складніше перехопити їх.

На рис. 2.45 показана мережа, яка була отримана з мережі з центральним концентратором (див. рис. 2.42) шляхом його заміни на міст. Мережі 1-го і 2-го відділів складаються з окремих логічних сегментів, а мережа відділу 3 – з двох логічних сегментів. Кожен логічний сегмент побудований на базі концентратора і має просту фізичну структуру, утворену відрізками кабелю, що пов'язують комп'ютери з портами концентратора. Якщо користувач комп'ютера А буде посилати дані користувачеві комп'ютера В, який знаходиться в одному з них сегменті, то ці дані будуть повторені тільки на тих мережних інтерфейсах, які відмічені на малюнку заштрихованими кругами.

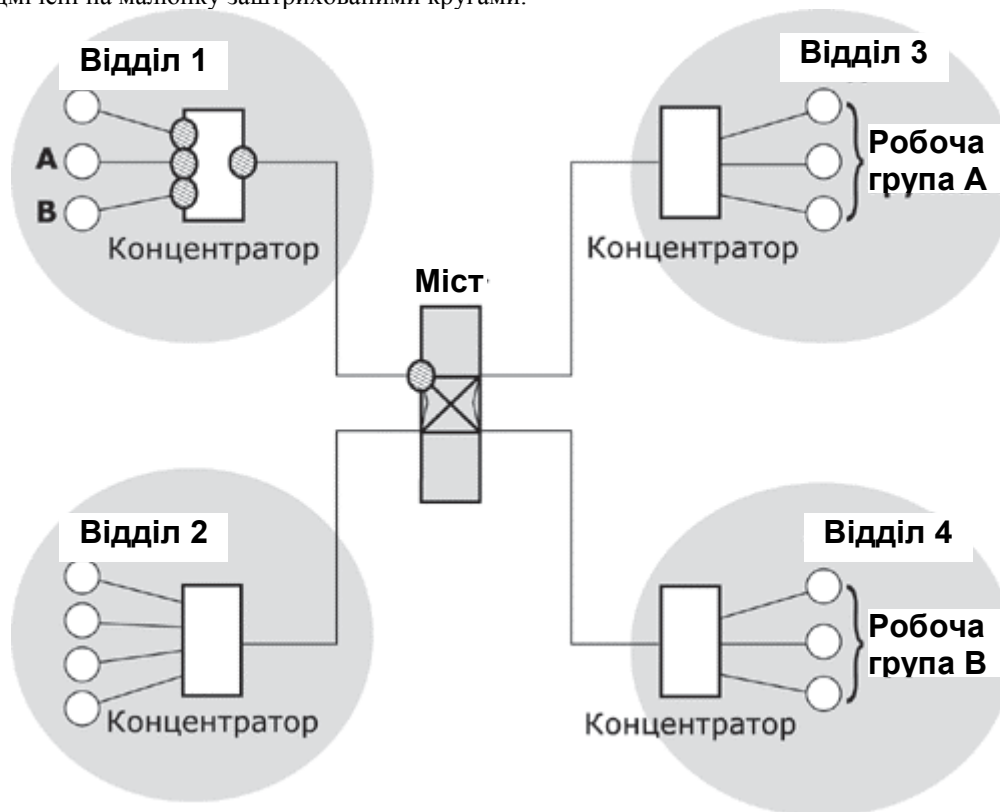


Рис. 2.45. Логічна структуризація мережі за допомогою моста.

Мости використовують для локалізації трафіку апаратні адреси комп'ютерів. Це затруднює розпізнавання приналежності того або іншого комп'ютера до певного логічного сегменту – сама адреса не містить подібної інформації. Тому міст достатньо спрощено представляє ділення мережі на сегменти – він запам'ятовує, через який порт на нього поступив кадр даних від кожного комп'ютера мережі, і надалі передає кадри, призначені для даного комп'ютера, на цей порт. Точної топології зв'язків між логічними сегментами міст не знає. Через це застосування мостів приводить до значних обмежень на конфігурацію зв'язків мережі – сегменти повинні бути сполучені так, щоб в мережі не утворювалися замкнуті контури.

Комутатор (*switch*) за принципом обробки кадрів від моста практично нічим не відрізняється. Єдина його відмінність полягає в тому, що він є свого роду комунікаційним мультипроцесором, оскільки кожен його порт оснащений спеціалізованою мікросхемою, яка обробляє кадри по алгоритму моста незалежно від мікросхем інших портів. За рахунок цього загальна продуктивність комутатора зазвичай набагато вище за продуктивність традиційного моста, що має один процесорний блок. Можна сказати, що комутатори – це мости нового покоління, які обробляють кадри в паралельному режимі.

Обмеження, зв'язані із застосуванням мостів і комутаторів, по топології зв'язків, а також ряд інших, привели до того, що у ряді комунікаційних пристроїв з'явився ще один тип устаткування – маршрутизатор (*router*). У термінології *Internet* маршрутизатор часто називають шлюзом (*gateway*).

Маршрутизатори надійніше і ефективніше ніж мости, ізолюють трафік окремих частин мережі один від одного. Маршрутизатори утворюють логічні сегменти за допомогою явної адресації, оскільки використовують не плоскі апаратні, а складені числові адреси. У цих адресах є поле номера мережі, так що всі комп'ютери, у яких значення цього поля однакове, належать одному сегменту, що називається в даному випадку підмережею (*subnet*).

Окрім локалізації трафіку, маршрутизатори виконують ще багато інших корисних функцій. Так, маршрутизатори можуть працювати в мережі із замкнутими контурами, при цьому вони здійснюють вибір найбільш раціонального маршруту з декількох можливих. Мережа, представлена на рис. 2.46, відрізняється

від своєї попередниці (рис. 2.45) тим, що між підмережами відділів 1 і 2 прокладений додатковий зв'язок, який може використовуватися для підвищення як продуктивності мережі, так і її надійності.

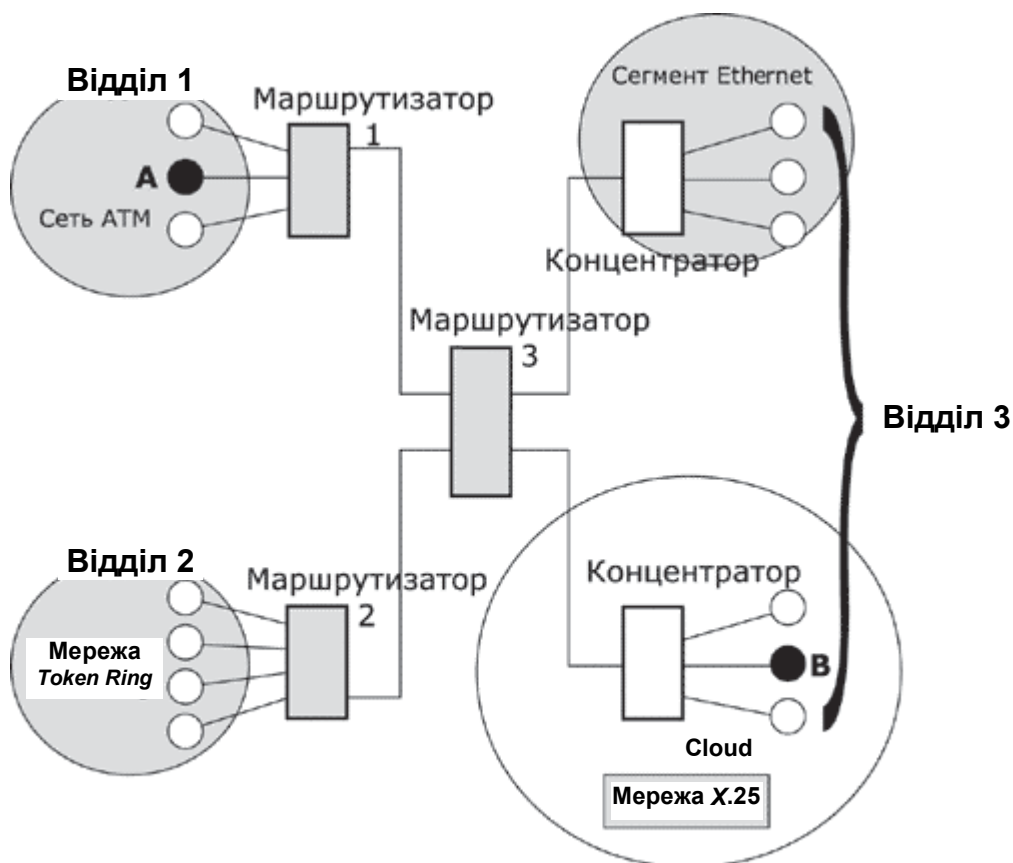


Рис. 2.46. Логічна структуризація мережі за допомогою маршрутизаторів.

Іншою дуже важливою функцією маршрутизаторів є їх здатність зв'язувати в єдину мережу підмережі, побудовані з використанням різних мережних технологій, наприклад, *Ethernet* і *X.25*.

Окрім перерахованих пристроїв, окремі частини мережі може сполучати шлюз (*gateway*). Зазвичай основною причиною використання шлюзу в мережі є необхідність об'єднати мережі з різними типами системного і прикладного програмного забезпечення, а не бажання локалізувати трафік. Проте, шлюз забезпечує і локалізацію трафіку, що можна розглядати як деякий побічний ефект.

Великі мережі практично ніколи не будуються без логічної структуризації. Для окремих сегментів і підмереж характерні типові однорідні топології базових технологій, і для їх об'єднання завжди використовується устаткування, що забезпечує локалізацію трафіку: мости, комутатори, маршрутизатори і шлюзи.

2.13. Конвергенція комп'ютерних і телекомунікаційних мереж

Яскраво виражена останнім часом тенденція зближення різних типів мереж характерна не тільки для локальних і глобальних комп'ютерних мереж, але і для телекомунікаційних мереж інших типів. До телекомунікаційних мереж в даний час можна віднести:

- телефонні мережі;
- радіомережа;
- телевізійні мережі;
- комп'ютерні мережі

У всіх цих мережах ресурсом, що надається клієнтам, є інформація. У табл. 2.1 наведено класифікацію видів інформації та мереж, послуг, які надаються.

Таблиця 2.1.

Вид телекомунікаційної мережі	Вид послуг	Вид представлення інформації
телефонні мережі	<i>інтерактивні послуги</i>	тільки голосова інформація
радіомережі	широкомовні послуги	тільки голосова інформація
телевізійні мережі	широкомовні послуги	голос і зображення
комп'ютерні мережі		алфавітно-цифрове

Таблиця характеризує розподіл виду послуг і форми представлення інформації в мережах різного типу. Телефонні мережі надають інтерактивні послуги (interactive services), оскільки два абоненти, що беруть участь в розмові (або декілька абонентів, якщо це конференція), поперемінно проявляють активність. Радіомережі і телевізійні мережі надають ширококомовні послуги (broadcast services), при цьому інформація розповсюджується тільки в один бік – з мережі до абонентів, по схемі "один-до-багатьох" (point-to-multipoint). Сьогодні по багатьом напрямкам йде конвергенція. різних видів телекомунікаційних мереж.

2.13.1. Загальна структура телекомунікаційної мережі

Не дивлячись на те, що відмінності між комп'ютерними, телефонними, телевізійними і радіомережами мережами, безумовно, істотні, всі ці мережі на достатньо високому рівні абстракції мають схожі структури.

Телекомунікаційна мережа (рис. 2.47) в загальному випадку включає наступні компоненти:

- мережа доступу (access network) призначена для концентрації інформаційних потоків, які поступають по численних каналах зв'язку від устаткування користувачів, в порівняно невеликій кількості вузлів магістральної мережі;
- магістраль (backbone або core network) об'єднує окремі мережі доступу, забезпечуючи транзит трафіку між ними по високошвидкісних каналах;
- інформаційні центри або центри управління сервісами (data centers або services control point) – це власні інформаційні ресурси мережі, на основі яких здійснюється обслуговування користувачів.

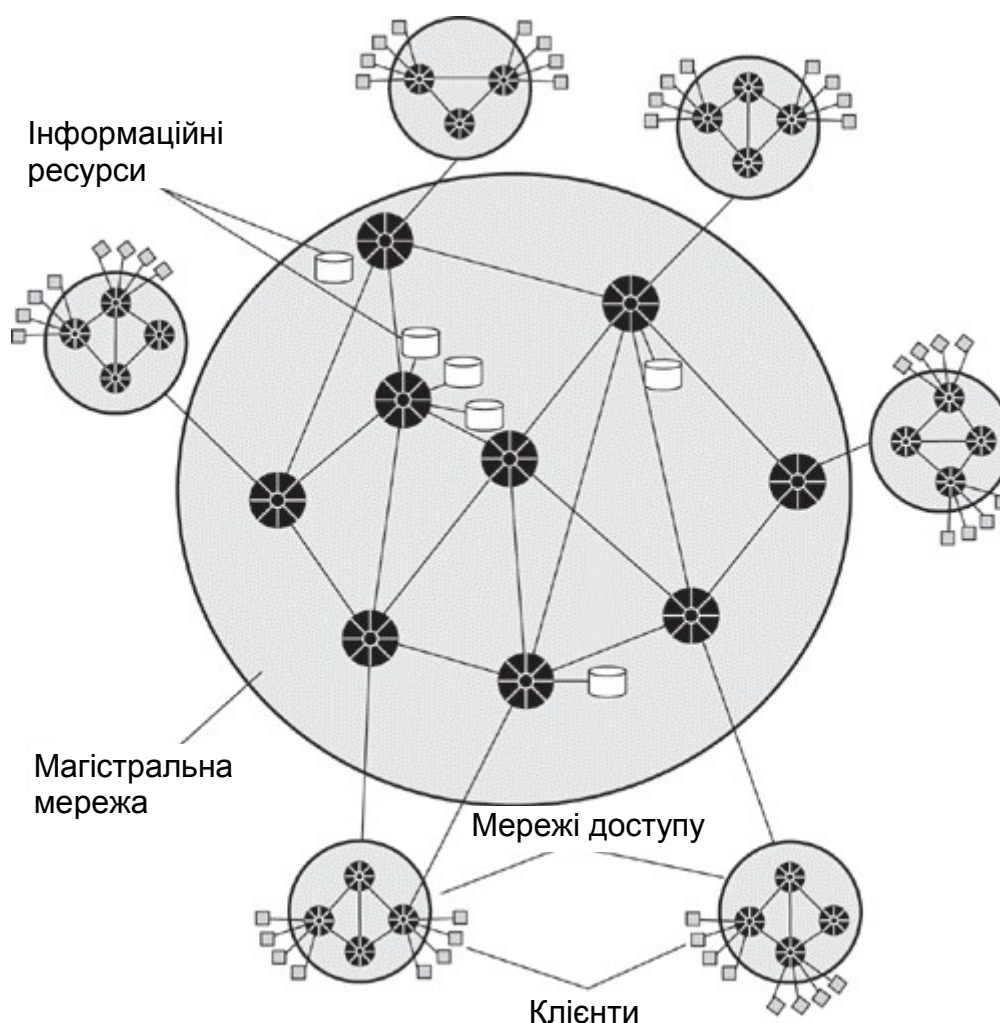


Рис. 2.47. Структура телекомунікаційної мережі.

І мережа доступу, і магістральна мережа будуються на базі комутаторів. Кожен комутатор оснащений деякою кількістю портів, які з'єднуються з портами інших комутаторів каналами зв'язку.

Мережа доступу

Мережа доступу – це нижній рівень ієрархії телекомунікаційної мережі. До цієї мережі підключаються кінцеві (термінальні) вузли – устаткування, встановлене у користувачів (абонентів, клієнтів) мережі. У разі комп'ютерної мережі кінцевими вузлами є комп'ютери, у разі телефонної – телефонні апарати, а у разі телевізійної або радіомережі – відповідні теле- і радіоприймачі.

Основне призначення мережі доступу – концентрація інформаційних потоків, що поступають по численних каналах зв'язку від устаткування користувачів, в порівняно невеликій кількості вузлів магістральної мережі.

Мережа доступу, як і телекомунікаційна мережа в цілому, може складатися з декількох рівнів (на малюнку показано два). Комутатори, встановлені у вузлах нижнього рівня, мультиплекують інформацію, що поступає по численних абонентських каналах (які часто називають абонентськими закінченнями, local loop) і передають її комутаторам верхнього рівня, щоб ті у свою чергу передали її комутаторам магістралі. Кількість рівнів мережі доступу залежить від її розміру, невелика мережа доступу може складатися з одного рівня, а велика – з двох-трьох. Наступні рівні здійснюють подальшу концентрацію трафіку, збираючи його і здійснюючи мультиплексування в швидкісні канали.

Магістральна мережа

Магістральна мережа об'єднує окремі мережі доступу, виконуючи функції транзиту трафіку між ними по високошвидкісних каналах. Комутатори магістралі можуть оперувати не тільки інформаційними з'єднаннями між окремими користувачами, але і агрегованими інформаційними потоками, що переносять дані великої кількості призначених для користувача з'єднань. В результаті інформація за допомогою магістралі потрапляє в мережу доступу одержувачів, демультиплексується там і комутується таким чином, що на вхідний порт устаткування користувача поступає тільки та інформація, яка йому адресована.

У тому випадку, коли абонент-одержувач підключений до того ж комутатору доступу, що і абонент-відправник (безпосередньо або через підлеглі за ієрархією зв'язків комутатори), останній виконує необхідну операцію комутації самостійно.

Інформаційні центри

Інформаційні центри/центри управління сервісами – це власні інформаційні ресурси мережі, на основі яких здійснюється обслуговування користувачів. У таких центрах може зберігатися інформація двох типів:

- призначена для користувача інформація, тобто ті дані, які безпосередньо цікавлять користувачів мережі;
- допоміжна службова інформація, що дозволяє надавати користувачам деякі послуги.

Прикладом інформаційних ресурсів першого типу можуть служити Web-портали, на яких розташована різноманітна довідкова інформація і новини, інформація електронних магазинів і т.п. У телефонних мережах роль таких центрів грають служби екстреного виклику (наприклад, міліції, швидкої допомоги) і довідкові служби різних організацій і підприємств – вокзалів, аеропортів, магазинів і т.п. У телевізійних мережах такими центрами є телестудії, що поставляють "живу" картинку або ж відтворюють раніше записані сюжети або фільми.

До ресурсів другого типу відносяться, наприклад, різні системи аутентифікації і авторизації користувачів, за допомогою яких організація, що володіє мережею, перевіряє права користувачів на отримання тих або інших послуг; системи білінгу, які в комерційних мережах підраховують плату за надані послуги; бази даних облікової інформації користувачів, що зберігають імена і паролі, а також переліки послуг, на які підписаний кожен користувач. У *телефонних мережах* існують центри управління сервісами (Services Control Point, SCP), де встановлені комп'ютери, на яких зберігаються програми нестандартної обробки телефонних викликів користувачів, наприклад, викликів безкоштовних довідкових служб комерційних підприємств (так звані служби 800) або викликів при проведенні телеголосування. Ще одним з поширених видів допоміжного інформаційного центру є централізована система управління мережею, яка є програмним забезпеченням, що працює на одному або декількох комп'ютерах.

Природно, у мереж кожного типу є багато особливостей, проте їх структура в цілому відповідає описаній вище. В той же час, залежно від призначення і розміру мережі, деякі складові узагальненої структури можуть в ній бути відсутніми або ж мати неістотне значення. Наприклад, в невеликій локальній комп'ютерній мережі немає яскраво виражених мереж доступу і магістралі – вони зливаються в загальну достатньо просту структуру. У корпоративній мережі, як правило, система білінгу відсутня, оскільки послуги співробітникам підприємства надаються не на комерційній основі. У деяких телефонних мережах можуть бути відсутніми інформаційні центри, а в телевізійних мережах доступу набуває вид розподільної мережі, оскільки інформація в ній розповсюджується тільки в одному напрямі – з мережі до абонентів.

2.13.2. Мережі операторів зв'язку

Комп'ютерні мережі можна класифікувати по різних критеріях. Відоме ділення на локальні і глобальні мережі, яке відбувається за територіальною ознакою, тобто за розмірами території, яку покриває мережа. Іншою важливою ознакою класифікації мереж є призначення послуг, що надаються:

- мережі операторів зв'язку (мережі провайдерів послуг) надають загальнодоступні послуги;
- корпоративні мережі надають послуги тільки співробітникам того підприємства, яке володіє мережею.

Оператори зв'язку і клієнти

Існують мережі, які створюються спеціально для надання загальнодоступних (публічних, public) телекомунікаційних послуг. Прикладами таких мереж можуть служити міські, регіональні, національні і міжнародні телефонні мережі. Їх послугами користуються численні клієнти – власники домашніх і мобільних телефонів, а також підприємства (корпоративні користувачі). Ще однією традиційною

телекомунікаційною послугою є надання в оренду каналів зв'язку. У первинних мережах *PDH/SDH*, що створюються телекомунікаційним підприємством для об'єднання своїх АТС, зазвичай залишається не використовуваною для внутрішніх потреб канална місткість, яку логічно здавати в оренду. Типовими споживачами цієї послуги є крупні підприємства, які створюють за допомогою орендованих каналів власні мережі – телефонні або комп'ютерні.

У міру зростання популярності комп'ютерної обробки даних до набору телекомунікаційних послуг додалася можливість об'єднання локальних мереж підприємств за допомогою загальнодоступної територіальної мережі передачі даних, наприклад, мережі технології *X.25, Frame relay, ATM* або *IP*. *Internet*-революція 90-х років породила таку поширену загальнодоступну послугу, як доступ в *Internet* для обміну повідомленнями електронної пошти і використання ресурсів численних *Web*-сайтів. Незабаром середовище *Internet* стало використовуватися підприємствами не тільки для доступу до "чужих" інформаційних ресурсів, але і для об'єднання власних, тобто як типова мережа передачі даних, що надає транспортні послуги. На стику *телефонних і комп'ютерних мереж* почали з'являтися нові типи загальнодоступних послуг, які використовують можливості комплексного застосування різних технологій.

Спеціалізоване підприємство, яке створює телекомунікаційну мережу для надання загальнодоступних послуг, володіє цією мережею і підтримує її роботу, традиційно називається оператором зв'язку (*telecommunication carrier*).

Оператори зв'язку відрізняються один від одного:

- набором послуг, що надаються;
- територією, в межах якої надаються послуги;
- типом клієнтів, на яких орієнтовані послуги;
- інфраструктурою, що є у володінні оператора: лініями зв'язку, комутаційним устаткуванням, інформаційними серверами і т.п.

Послуги, провайдери послуг і мережна інфраструктура

Оператори зв'язку здійснюють свою діяльність на комерційній основі, укладаючи договори із споживачами послуг. Послуги можна розділити на декілька рівнів і груп, використовуючи різні критерії класифікації.

На рис. 2.48 групи послуг визначаються за типом мереж, які їх надають: телефонні або комп'ютерні (для повноти картини потрібно було б доповнити малюнок послугами телевізійних і радіомереж). На рисунку показані тільки деякі основні рівні і групи, але і ця неповна картина добре ілюструє широту спектру сучасних телекомунікаційних послуг і складність їх взаємозв'язків. Послуги кожного вищого рівня спираються на послуги нижніх рівнів. Послуги надання в оренду каналів зв'язку є послугами самого нижнього рівня, оскільки користувач спочатку повинен побудувати за допомогою цих каналів власну мережну інфраструктуру (встановити телефонні комутатори або комутатори пакетних мереж).

Наступний рівень складають дві великі групи послуг:

- телефонні послуги;
- послуги комп'ютерних мереж.

Комбіновані послуги:	
<ul style="list-style-type: none"> • IP-телефонія • Універсальна служба повідомлень (Unified Messaging) 	
Послуги телефонії:	Послуги комп'ютерних мереж:
<ul style="list-style-type: none"> • З'єднання двох абонентів • Доступ до довідкових служб • Переадресація викликів • Голосова пошта • Інші 	<ul style="list-style-type: none"> • Доступ в Internet • Електронна пошта • Об'єднання LAN • Віртуальні приватні мережі • Інформаційні портали (WWW) • Інші
Надання каналів зв'язку в оренду	

Рис. 2.48. Класифікація послуг телекомунікаційної мережі (закрашені області відповідають традиційним послугам операторів зв'язку).

Цей рівень, у свою чергу, можна ділити на підрівні, оскільки з простіших послуг будуються складніші. Наприклад, на основі послуги доступу до *Internet*, яка полягає в простому транспортному підключенні комп'ютера або локальної мережі до усесвітньої загальнодоступної мережі, можна надавати послуги електронної пошти. Або ж на основі базової телефонної послуги з'єднання абонентів можна створити послугу голосової пошти.

Верхній рівень сьогодні займають комбіновані послуги, реалізація яких вимагає сумісної оперативної взаємодії комп'ютерних і телефонних мереж.

Послуги можна класифікувати залежно від того, чи надається клієнтові додаткова інформація:

- послуги, що полягають в передачі трафіку в незмінному вигляді між абонентами мережі;
- послуги, що полягають в наданні користувачеві інформації, створеної оператором або операторами зв'язку.

Телефонна розмова – це приклад послуги першого типу, оскільки інформація створюється абонентами мережі, а оператор тільки доставляє її від одного абонента до іншого. До цього ж типу відноситься послуга з'єднання двох локальних мереж клієнта за допомогою мережі передачі даних оператора. Прикладами послуг другого типу (їх зазвичай називають інформаційними) є послуги довідкових служб *телефонної мережі* або послуги якого-небудь *Web-сайту*.

Ще одна загальноживана ознака класифікації послуг – наявність або відсутність інтерактивності. Якщо при наданні послуги поперемінно і в реальному масштабі часу активні два (або більш) абоненти, то це інтерактивна послуга. Телефонна розмова – класичний приклад інтерактивної послуги. Якщо ж абонент тільки отримує інформацію, що передається по мережі, то це не інтерактивна послуга. Наприклад, такими є послуги радіо і телебачення, аналогічні послуги надають *Web-сайти*, проте тут грань між інтерактивністю і неінтерактивністю провести складно, оскільки користувач може брати активну участь в прогляданні змісту сайту, граючи або відповідаючи на питання анкети.

Кожен оператор зв'язку надає ті послуги із загального набору, які йому представляються найбільш відповідними його профілю і економічно вигідними. У більшості країн світу (і в Росії теж) оператори зв'язку повинні отримувати від державних органів ліцензії на надання тих або інших послуг зв'язку. Таке положення існувало не завжди – практично у всіх країнах були оператори, які фактично були монополістами на ринку телекомунікаційних послуг в масштабах країни. Сьогодні відбувається процес демонополізації цієї області діяльності, колишні монополісти продовжують працювати, але їм доводиться боротися за клієнтів з новими операторами, яких часто називають альтернативними, оскільки завдяки ним у користувачів дійсно з'являється свобода вибору.

Клієнти

Всю безліч клієнтів – споживачів инфотелекоммунікаційних послуг можна розділити на дві великі групи:

- масові індивідуальні клієнти;
- корпоративні клієнти.

Масові індивідуальні клієнти

У першому випадку місцем споживання послуг є квартира або приватний будинок, а клієнтами – мешканці цієї квартири, яким потрібні, перш за все, базові послуги – телефонний зв'язок, телебачення, радіо і вихід в Internet.

Для масових клієнтів дуже важлива економічність послуги – низька місячна плата, можливість використання стандартних термінальних пристроїв (телефонні апарати, телевізійні приймачі, персональні комп'ютери), а також проводки, що існують в квартирі, у вигляді телефонної пари і телевізійного коаксіального кабелю. Складні в обігу і дорогі термінальні пристрої, такі як, наприклад, які включають комп'ютер телевізори або *IP-телефони*, навряд чи стануть масовими до тих пір, поки не наблизяться за вартістю до звичайного телевізора або телефону і не підтримуватимуть простий, призначений для користувача інтерфейс.

Проводка, що існує в наших будинках – це серйозне обмеження для надання послуг доступу в Internet і нових послуг *комп'ютерних мереж*, оскільки вона не розрахована на передачу даних, а підведення до кожного будинку нового якісного кабелю, наприклад, волоконно-оптичного, обійдеться недешево. Тому доступ в Internet найчастіше здійснюється за допомогою низькошвидкісного модемного з'єднання по *телефонній мережі*. Проте поступово розвиваються нові технології – так звані технології цифрових абонентських ліній, що дозволяють передавати по існуючій телефонній проводці дані з набагато вищою швидкістю, ніж звичайні модеми.

Корпоративні клієнти

Корпоративні клієнти – це підприємства і організації різного профілю. Невеликі підприємства по набору переважних послуг не дуже відрізняються від масових клієнтів – це та ж базова телефонія і телебачення, тільки телефонних номерів такому підприємству може потрібно не один, а два-три, та і потреби в передачі даних зводиться до стандартного модемного доступу до Internet.

Крупні ж підприємства, що складаються з декількох територіально розосереджених відділень і філіалів, а також ті, які мають співробітників, що часто працюють удома, потребують розширеного набору послуг. Перш за все, такою послугою є віртуальна приватна мережа (*Virtual Private Network, VPN*), коли оператор зв'язку створює для підприємства ілюзію того, що всі його відділення і філіали сполучені приватною, тобто повністю належною і керованою підприємством-клієнтом мережею. Тоді як насправді при цьому використовується мережа оператора, тобто загальнодоступна мережа, що одночасно передає дані багатьох клієнтів. Послуги *VPN* можуть надаватися як для телефонії (наприклад, корпоративні користувачі дзвонять по скорочених внутрішніх номерах), так і для мереж передачі даних.

Останнім часом корпоративні клієнти все частіше користуються не тільки комунікаційними, але і інформаційними послугами операторів, наприклад, переносять власні Web-сайти і бази даних на територію оператора, доручаючи останньому підтримувати їх роботу і забезпечувати швидкий доступ до них для співробітників підприємства і, можливо, інших користувачів мережі оператора.

Крупні корпоративні клієнти вимагають розширеного набору послуг і згодні платити за нього більше, ніж за стандартний, якщо послуги надаються з високим рівнем якості. Тому оператор може узятися за прокладку нової фізичної лінії зв'язку до приміщення такого клієнта і установку складних комунікаційних пристроїв. Одні операторів надають послуги як масовим, так і корпоративним клієнтам, інші спеціалізуються тільки на одній категорії споживачів.

2.13.3. Інфраструктура

Окрім суб'єктивних і правових причин на формування набору пропонованих оператором послуг робить вплив матеріально-технічний чинник, оскільки для надання певної послуги оператор повинен володіти відповідною апаратно-програмною інфраструктурою. Так, для надання послуг з оренди каналів оператор повинен мати в своєму розпорядженні транспортну мережу, наприклад, первинну мережу *PDH/SDH* або ж мережу з комутацією каналів, таку як *ISDN*. Для надання інформаційних *Web*-послуг він повинен створити власний сайт, який повинен бути сполучений з Internet, щоб користувачі Internet могли дістати доступ до нього.

У тих випадках, коли у оператора відсутня необхідна інфраструктура для надання деякої послуги, він може скористатися послугами іншого оператора, на основі яких необхідна послуга може бути сконструйована. Наприклад, для створення загальнодоступного *Web*-сайту електронної комерції оператор зв'язку може не мати власної IP-мережі, сполученої з Internet. Для цього йому досить тільки створити інформаційне наповнення сайту і помістити його на комп'ютері іншого оператора, мережа якого має підключення до Internet. Оренда фізичних каналів зв'язку для створення власної телефонної або комп'ютерної мережі є іншим типовим прикладом надання послуг за відсутності одного з елементів апаратно-програмної інфраструктури. Оператора, який надає послуги іншим операторам зв'язку, часто називають оператором операторів (*carrier of carriers*).

Фізичні канали зв'язку і інші елементи первинної транспортної інфраструктури грають ключову роль в наданні будь-яких комунікаційних послуг, оскільки без них передача інформації стає просто неможливою. Власне, від наявності або відсутності власної транспортної інфраструктури залежить і назва підприємства, що надає інформаційно-комунікаційні послуги – традиційне "оператор зв'язку" або нове "провайдер (постачальник) послуг". Хоча кожен оператор зв'язку, безумовно, надає послуги своїм клієнтам, тобто є провайдером послуг, в цих термінах є деяка різниця. Кажучи "оператор зв'язку", зазвичай підкреслюють ту обставину, що компанія володіє власною транспортною інфраструктурою, підтримує її функціонування і на цій основі надає послуги. Зазвичай традиційний оператор зв'язку в першу чергу надає низькорівневі транспортні послуги – просту передачу трафіку (телефонного трафіку або трафіку даних) між географічними пунктами без його додаткової обробки. Коли ж говорять "провайдер послуг", то акцент робиться на тому, що підприємство надає нові високорівневі послуги, наприклад, доступ в Internet, розміщення в своїй мережі інформаційних ресурсів (*Web*-сайти, бази даних підприємств), але не обов'язково володіє власною розвинутою транспортною інфраструктурою, оскільки часто для їх ефективної реалізації достатньо орендованих мережних ресурсів.

Територія покриття

По ступеню обхвату території, на якій надаються послуги, оператори діляться на:

- локальних;
- регіональних;
- національних;
- транснаціональних.

Локальний оператор працює на території міста або сільського району. Традиційний локальний оператор – це оператор міської телефонної мережі, який володіє всією відповідною транспортною інфраструктурою: фізичними каналами між приміщеннями абонентів (квартирами, будинками і офісами) і вузлом зв'язку, телефонними станціями (АТС) і каналами зв'язку між телефонними станціями.

Сьогодні до традиційних локальних операторів додалися альтернативні, які часто є провайдерами послуг нового типу, перш за все, пов'язаних з Internet, але іноді конкурують з традиційними операторами і в області телефонії. Не дивлячись на монополізацію телекомунікаційної галузі, фізичними каналами доступу до абонентів як і раніше володіють традиційні локальні оператори, тобто "Міські телефонні мережі" (МТС), тому альтернативним операторам простіше надавати високорівневі послуги, не так сильно залежні від наявності прямого зв'язку з абонентом, такі, наприклад, як доступ в Internet, розміщення в своїх вузлах інформаційних ресурсів і т.п. А для організації доступу до цих ресурсів абонентам доводиться укласти договір з традиційним оператором, який направляє трафік безпосередньо підключених до нього абонентів в мережу альтернативного оператора, якщо абонент підписався на відповідну послугу. В цьому випадку ми бачимо природну спеціалізацію операторів – кожен займається тією справою, для якої більше підходить наявна інфраструктура.

Регіональні і національні оператори надають послуги на великій території, маючи в своєму розпорядженні відповідну інфраструктуру. Традиційні оператори цього масштабу виконують транзитну передачу телефонного трафіку між телефонними станціями локальних операторів, маючи в своєму розпорядженні крупні транзитні АТС, зв'язані високошвидкісними фізичними каналами зв'язку. Це оператори операторів, їх клієнтами є, як правило, локальні оператори або крупні підприємства, що мають відділення і філіали в різних містах регіону або країни. Маючи в своєму розпорядженні розвинену транспортну інфраструктуру, такі оператори зазвичай надають послуги телекомунікації, передаючи транзитом великі об'єми інформації без якої-небудь обробки. Альтернативні оператори регіонального і національного масштабу можуть мати власну транспортну інфраструктуру, але це не обов'язково. У першому випадку вони конкурують з традиційними операторами на ринку послуг телекомунікації, а в другому прагнуть надавати додаткові інформаційні послуги, укладаючи договори з великим числом локальних операторів, а також з яким-небудь з операторів, який надає послуги телекомунікації, – для організації взаємодії між своїми інформаційними ресурсами.

Мережі відділів

Мережі відділів – це мережі, які використовуються порівняно невеликою групою співробітників, що працюють в одному відділі підприємства. Ці співробітники вирішують деякі загальні задачі, наприклад, ведуть бухгалтерський облік або займаються маркетингом. Вважається, що відділ може налічувати до 100-150 співробітників.

Головною метою мережі відділу є розділення локальних ресурсів, таких як програмне забезпечення, дані, лазерні принтери і модеми. Зазвичай мережі відділів мають один або два файлові сервери, не більше тридцяти користувачів (рис. 2.49) і не розділяються на підмережі. У цих мережах локалізується велика частина трафіку підприємства. Мережі відділів зазвичай створюються на основі якої-небудь однієї мережної технології – Ethernet, Token Ring. У такій мережі найчастіше використовується один або, максимум, два типи операційних систем. Невелика кількість користувачів дозволяє застосовувати в мережах відділів однорангові мережні ОС, наприклад, *Windows 98/XP*.

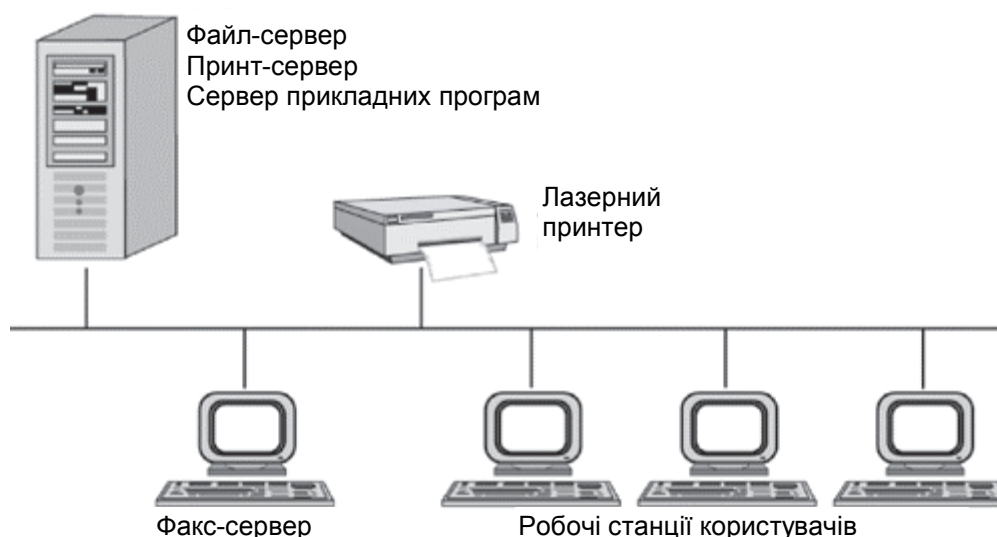


Рис. 2.49. Приклад мережі масштабу відділу.

Завдання управління мережею на рівні відділу відносно прості: додавання нових користувачів, усунення простих відмов, інсталяція нових вузлів і установка нових версій програмного забезпечення. Такою мережею може управляти співробітник, що присвячує виконанню обов'язків адміністратора тільки частину свого часу. Найчастіше адміністратор мережі відділу не має спеціальної підготовки, але є тією людиною у відділі, який краще за всіх розбирається в комп'ютерах, і само собою виходить так, що він і займається адмініструванням мережі.

Існує і інший тип мереж, близький до мереж відділів, – мережі робочих груп. До таких мереж відносять зовсім невеликі мережі, що включають до 10-20 комп'ютерів. Характеристики мереж робочих груп практично не відрізняються від описаних вище характеристик мереж відділів. Такі властивості, як простота мережі і однорідність, в цьому випадку виявляються найбільшою мірою, тоді як мережі відділів можуть наближатися в деяких випадках до наступного по масштабу типу мереж – *мереж кампусів*.

Мережі кампусів

Мережі кампусів отримали свою назву від англійського слова campus – студентське містечко. Саме на території університетських городків часто виникала необхідність в об'єднанні декількох дрібних мереж в

одну велику. Зараз цю назву не пов'язують із студентськими городками, а використовують для позначення мереж будь-яких підприємств і організацій.

Мережі кампусів (рис. 2.50) об'єднують безліч мереж різних відділів одного підприємства в межах окремої будівлі або однієї території, що покриває площу в декілька квадратних кілометрів. При цьому глобальні з'єднання в мережах кампусів не використовуються. Служби такої мережі включають взаємодію між мережами відділів, доступ до загальних баз даних підприємства, доступ до загальних факс-серверів, високошвидкісних модемів і високошвидкісних принтерів. В результаті співробітники кожного відділу підприємства дістають доступ до деяких файлів і ресурсів інших відділів. Мережі кампусів забезпечують доступ до корпоративних баз даних незалежно від того, на яких типах комп'ютерів вони розташовуються.

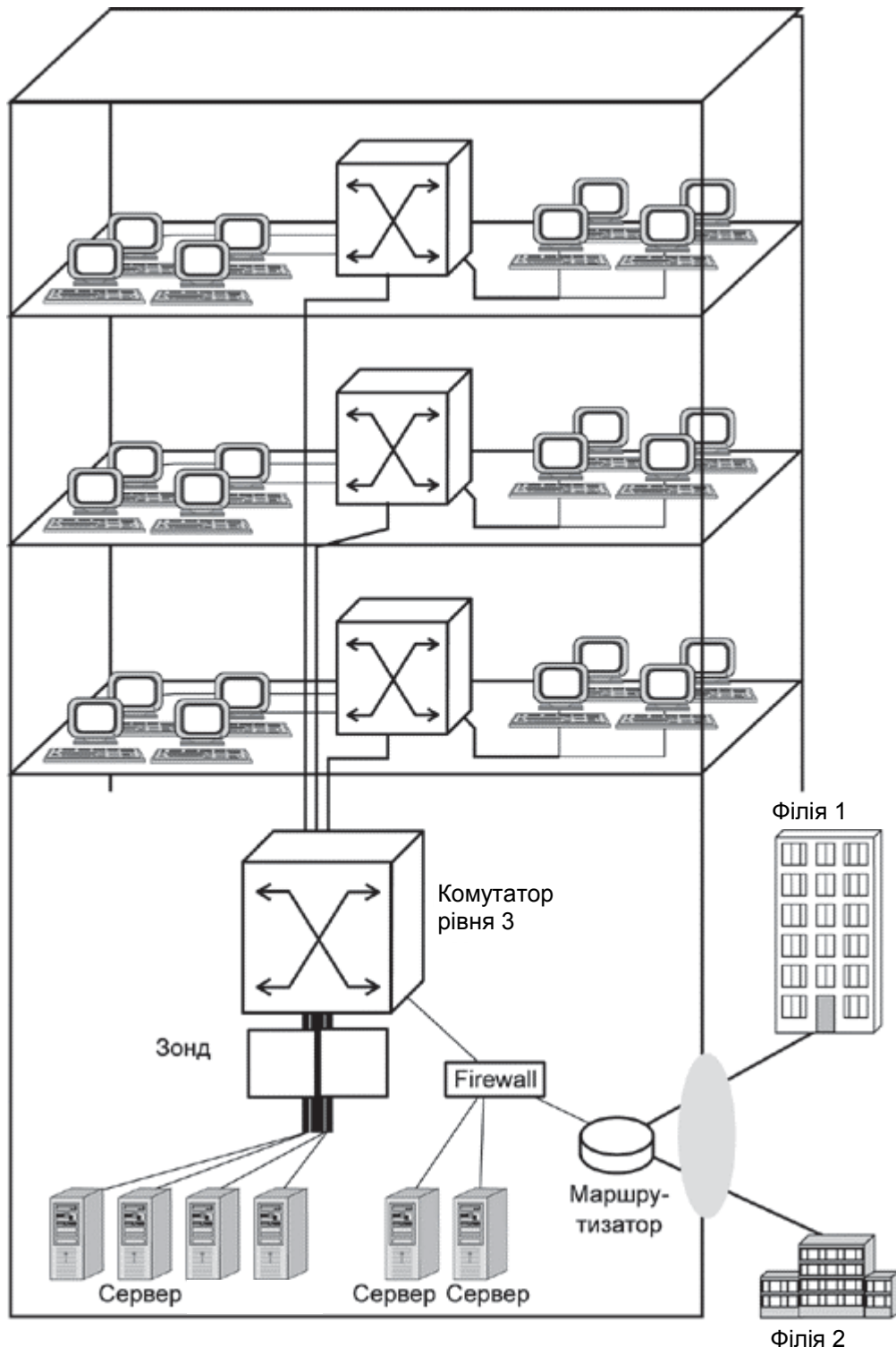


Рис. 2.50. Приклад мережі кампусу, Firewall - міжмережний екран.

Саме на рівні мережі кампусу виникають проблеми інтеграції неоднорідного апаратного і програмного забезпечення. Типи комп'ютерів, мережних операційних систем, мережного апаратного забезпечення в кожному відділі можуть відрізнятися. Звідси витікають складнощі управління мережами кампусів. Адміністратори повинні бути в цьому випадку більш кваліфікованими, а засоби оперативного управління мережею – ефективнішими.

Корпоративні мережі

Корпоративна мережа – це мережа, головним призначенням якої є забезпечення функціонування конкретного підприємства, що володіє даною мережею. Користувачами корпоративної мережі є тільки співробітники даного підприємства. На відміну від мереж операторів зв'язку, корпоративні мережі, в загальному випадку, не надають послуг іншим організаціям або користувачам.

Залежно від масштабу підприємства, а також від складності і різноманіття вирішуваних задач розрізняють мережі відділу, мережі кампусу і корпоративні мережі (термін "корпоративні" в даній класифікації набуває вузького значення – мережа великого підприємства). Перш ніж обговорювати характерні особливості кожного з перерахованих типів мереж, зупинимось на тих чинниках, які примушують підприємства обзаводитися власною комп'ютерною мережею.

Корпоративні мережі називають також мережами масштабу підприємства, що відповідає дослівному перекладу терміну "Enterprise-wide networks", який використовується в англійській літературі для позначення цього типу мереж. Мережі масштабу підприємства (корпоративні мережі) об'єднують велику кількість комп'ютерів на всіх територіях окремого підприємства. Вони можуть бути складно зв'язані і здатні покривати місто, регіон або навіть континент. Число користувачів і комп'ютерів може вимірюватися тисячами, а число серверів – сотнями, відстані між мережами окремих територій бувають такими, що доводиться використовувати глобальні зв'язки. Для з'єднання віддалених локальних мереж і окремих комп'ютерів в корпоративній мережі застосовуються різноманітні телекомунікаційні засоби, зокрема телефонні канали, радіоканали, супутниковий зв'язок. Корпоративну мережу можна представити у вигляді "острівців локальних мереж", плаваючих в телекомунікаційному середовищі.

Неодмінним атрибутом такої складної і великомасштабної мережі є високий ступінь неоднорідності (гетерогенності) – не можна задовольнити потреби тисяч користувачів за допомогою однотипних програмних і апаратних засобів. У корпоративній мережі обов'язково використовуватимуться різні типи комп'ютерів – від мейнфреймів до персональних ЕОМ, декілька типів операційних систем і безліч різних програм. Неоднорідні частини корпоративної мережі повинні працювати як єдине ціле, надаючи користувачам по можливості зручний і простий доступ до всіх необхідних ресурсів. Мережі підприємств (корпоративні мережі) об'єднують велику кількість комп'ютерів на всіх територіях окремого підприємства. Для корпоративної мережі характерні:

- масштабність – тисячі призначених для користувача комп'ютерів, сотні серверів, величезні об'єми даних, що зберігаються і передаються по лініях зв'язку, безліч різноманітних програм;
- високий ступінь гетерогенності – різні типи комп'ютерів, комунікаційного устаткування, операційних систем і програм;
- використання локальних зв'язків – з'єднання в рамках мережі окремої філії може здійснюватися через цифрові АТС установ, мережі *ISDN* та ін;
- використання глобальних зв'язків – мережі філій з'єднуються за допомогою телекомунікаційних засобів, зокрема телефонних каналів, радіоканалів, супутникового зв'язку.

Поява корпоративних мереж – це хороша ілюстрація відомого постулату про перехід кількості в якість. При об'єднанні окремих мереж крупного підприємства, що має філіали в різних містах і навіть країнах, в єдину мережу багато кількісних характеристик об'єднаної мережі переходять деякий критичний поріг, за яким починається нова якість. У цих умовах існуючі методи і підходи до рішення традиційних задач мереж менших масштабів для корпоративних мереж виявилися непридатними. На перший план вийшли такі завдання і проблеми, які в мережах робочих груп, відділів і навіть кампусів або мали другорядне значення, або взагалі не виявлялися. Прикладом може служити просте (для невеликих мереж) завдання – ведення облікових даних про користувачів мережі.

Найбільш простий спосіб її рішення – помістити облікові дані кожного користувача в локальну базу облікових даних кожного комп'ютера, до ресурсів якого користувач повинен мати доступ. При спробі доступу ці дані витягуються з локальної облікової бази, і на їх основі надається або не надається доступ. У невеликій мережі, що складається з 5-10 комп'ютерів і приблизно такої ж кількості користувачів, такий спосіб працює дуже добре. Але якщо в мережі налічується декілька тисяч користувачів, кожному з яких потрібний доступ до декількох десятків серверів, то, очевидно, це рішення стає край неефективним. Адміністратор повинен повторити декілька десятків разів (по числу серверів) операцію занесення облікових даних кожного користувача. Сам користувач також вимушений повторювати процедуру логічного входу кожного разу, коли йому потрібний доступ до ресурсів нового сервера. Правильне рішення цієї проблеми для крупної мережі – використовувати централізовану довідкову службу, в базі даних якої зберігаються облікові записи всіх користувачів мережі. Адміністратор один раз виконує операцію занесення даних користувача в цю базу, а користувач один раз виконує процедуру логічного входу, причому не в окремий сервер, а в мережу цілком.

При переході від простішого типу мереж до складнішого – від мереж відділу до корпоративної мережі, територія обхвату збільшується, підтримувати зв'язки комп'ютерів стає все складнішим. У міру збільшення масштабів мережі підвищуються вимоги до її надійності, продуктивності і функціональних можливостей. По мережі циркулює вся зростаюча кількість даних і необхідно забезпечувати їх безпеку і захищеність разом з доступністю. Все це призводить до того, що корпоративні мережі будуються на основі найбільш могутнього і різноманітного устаткування і програмного забезпечення.

2.13.4. Переваги і проблеми використання мереж

Розглянемо наступні питання:

1. У яких випадках розгортання на підприємстві обчислювальних мереж переважно використання автономних комп'ютерів або багатомашинних систем?
2. Які нові можливості з'являються на підприємстві з появою обчислювальної мережі?
3. І, нарешті, чи завжди підприємству потрібна мережа?

Якщо не вдаватися в подробиці, то кінцевою метою використання комп'ютерних мереж на підприємстві є підвищення ефективності його роботи, яке може виражатися, наприклад, в збільшенні прибутку. Дійсно, якщо завдяки комп'ютеризації знизилася витрати на виробництво вже існуючого продукту, скоротилися терміни розробки нової моделі або прискорилося обслуговування замовлень споживачів – це означає, що даному підприємству насправді потрібна була мережа.

Концептуальною перевагою мереж, яка витікає з їх приналежності до розподілених систем, перед автономно працюючими комп'ютерами є їх здатність виконувати паралельні обчислення. За рахунок цього в системі з декількома оброблювальними вузлами в принципі можна досягти продуктивності, що перевищує максимально можливу на даний момент продуктивність будь-якого окремого, скільки завгодно могутнього, процесора. Розподілені системи потенційно мають краще співвідношення продуктивність/вартість, чим централізовані системи.

Ще одна очевидна і важлива перевага розподілених систем – це їх вища відмовостійкість. Під відмовостійкістю слід розуміти здатність системи виконувати свої функції (можливо, не в повному об'ємі) при відмовах окремих елементів апаратури і неповної доступності даних. Основою підвищеної відмовостійкості розподілених систем є надмірність. Надмірність оброблювальних вузлів (процесорів в багатопроцесорних системах або комп'ютерів в мережах) дозволяє при відмові одного вузла перепризначувати приписані йому завдання на інші вузли. З цією метою в розподіленій системі можуть бути передбачені процедури динамічної або статичної реконфігурації. У обчислювальних мережах деякі набори даних можуть дублюватися на зовнішніх пристроях декількох комп'ютерів мережі, так що при відмові одного з них дані залишаються доступними.

Використання територіально розподілених обчислювальних систем більше відповідає розподіленому характеру прикладних завдань в деяких предметних областях, таких як автоматизація технологічних процесів, банківська діяльність і т.п. У всіх цих випадках є розосереджені по деякій території окремі споживачі інформації – співробітники, організації або технологічні установки. Ці споживачі автономно вирішують свої задачі, тому слід було б надавати їм власні обчислювальні засоби, але в той же час, оскільки вирішувані ними задачі логічно тісно взаємозв'язані, їх обчислювальні засоби повинні бути об'єднані в загальну систему. Оптимальним рішенням в такій ситуації є використання обчислювальної мережі.

Для користувача розподілені системи дають ще і такі переваги, як можливість сумісного використання даних і пристроїв, а також можливість гнучкого розподілу робіт по всій системі. Таке розділення дорогих периферійних пристроїв, таких як дискові масиви великої місткості, кольорові принтери, графічні пристрої, модеми, оптичні диски, у багатьох випадках є основною причиною розгортання мережі на підприємстві. Користувач сучасної обчислювальної мережі працює за своїм комп'ютером, часто не віддаючи собі звіту в тому, що він користується даними іншого могутнього комп'ютера, що знаходиться за сотні кілометрів від нього. Він відправляє електронну пошту через модем, підключений до комунікаційного сервера, загального для декількох відділів його підприємства. У користувача створюється враження, що ці ресурси підключені безпосередньо до його комп'ютера або ж "майже" підключені, оскільки для роботи з ними потрібні незначні додаткові дії в порівнянні з використанням дійсно власних ресурсів.

Останнім часом став переважати інший мотив розгортання мереж, набагато важливіший в сучасних умовах, чим економія засобів за рахунок розділення між співробітниками корпорації дорогої апаратури або програм. Цим мотивом стало прагнення забезпечити співробітникам оперативний доступ до обширної корпоративної інформації. В умовах жорсткої конкурентної боротьби в будь-якому секторі ринку виграє, кінець кінцем, та компанія, співробітники якої можуть швидко і правильно відповісти на будь-яке питання клієнта – про можливості їх продукції, про умови її застосування, про вирішення різних проблем і т.п. На крупному підприємстві навіть хороший менеджер навряд чи знає всі характеристики кожного з продуктів, що випускаються, тим більше що їх номенклатура може оновлюватися кожного кварталу, якщо не місяць. Тому дуже важливо, щоб менеджер мав можливість з свого комп'ютера, підключеного до корпоративної мережі, скажімо, в Магадані або Житомері, передати питання клієнта на сервер, розташований в центральному відділенні підприємства в Москві або в Новосибірську, і оперативно отримати відповідь, що задовольняє клієнта. У такому разі клієнт не звернеться в іншу компанію, а користуватиметься послугами даного менеджера і надалі.

Використання мережі приводить до вдосконалення комунікацій між співробітниками підприємства, а також його клієнтами і постачальниками. Мережі знижують потребу підприємств в інших формах передачі інформації, таких як телефон або звичайна пошта. Часто саме можливість організації електронної пошти є однією з причин розгортання на підприємстві обчислювальної мережі. Всього більшого поширення набувають нові технології, які дозволяють передавати по мережних каналах зв'язку не тільки комп'ютерні дані, але і голосову і відеоінформацію. Корпоративна мережа, яка інтегрує дані і мультимедійну інформацію, може використовуватися для організації аудіо- і відеоконференцій, крім того, на її основі може бути створена власна внутрішня телефонна мережа.

Переваги, які дає використання мереж:

1. Інтегральна перевага – підвищення ефективності роботи підприємства.
2. Здатність виконувати паралельні обчислення, за рахунок чого може бути підвищена продуктивність і відмовостійкість.
3. Більша відповідність розподіленому характеру деяких прикладних завдань.

Можливість сумісного використання даних і пристроїв.

4. Можливість гнучкого розподілу робіт по всій системі.
5. Оперативний доступ до обширної корпоративної інформації.
6. Вдосконалення комунікацій.

Проблеми використання мереж:

1. Складність розробки системного і прикладного програмного забезпечення для розподілених систем.
2. Проблеми з продуктивністю і надійністю передачі даних по мережі.
3. Проблема забезпечення безпеки.

Звичайно, при використанні обчислювальних мереж виникають і проблеми, пов'язані в основному з організацією ефективної взаємодії окремих частин розподіленої системи.

По-перше, це неполадки в програмному забезпеченні операційних системах і прикладних програмах. Програмування для розподілених систем принципово відрізняється від програмування для централізованих систем. Так, мережна операційна система, виконуючи в загальному випадку всі функції по управлінню локальними ресурсами комп'ютера, понад те вирішує численні задачі, пов'язані з наданням мережних послуг. Розробка мережних програмних продуктів ускладнюється із-за необхідності організувати спільну роботу їх частин, що виконуються на різних машинах. Завдає багато клопоту і забезпечення сумісності програмного забезпечення, що встановлюється у вузлах мережі.

По-друге, багато проблем пов'язано з транспортуванням повідомлень по каналах зв'язку між комп'ютерами. Основні завдання в цьому випадку – забезпечення надійності (щоб передані дані не втрачалися і не спотворювалися) і продуктивності (щоб обмін даними відбувався з прийнятними затримками). У структурі загальних витрат на обчислювальну мережу витрати на рішення "транспортних питань" складають істотну частину, тоді як в централізованих системах ці проблеми повністю відсутні.

По-третє, це питання, пов'язані із забезпеченням безпеки, які набагато складніше вирішувати в обчислювальній мережі, чим в автономно працюючому комп'ютері. В деяких випадках, коли безпека особливо важлива, від використання мережі краще відмовитися.

Можна приводити ще багато "за і проти", але головним доказом ефективності використання мереж є безперечний факт їх повсюдного розповсюдження. Сьогодні важко знайти підприємство, на якому немає хоч би односегментної мережі персональних комп'ютерів; все більше і більше з'являється мереж з сотнями робочих станцій і десятками серверів, деякі крупні організації обзаводяться приватними глобальними мережами, об'єднуючими їх філіали, віддалені на тисячі кілометрів. У кожному конкретному випадку для створення мережі були свої підстави, але вірно і загальне твердження: щось в цих мережах все-таки є.

2.14. Практичні відомості

2.14.1. Приклад розрахунку затримки передачі даних

Оцінимо затримку передачі даних в мережах з комутацією пакетів порівняно з мережами з комутацією каналів.

Нехай тестове повідомлення, яке потрібно передати в обох видах мереж, складає 200 Кбайт. Відправник знаходиться від одержувача на відстані 5000 км. Пропускна спроможність ліній зв'язку складає 2 Мбіт/с. Час передачі даних по мережі з комутацією каналів складається з часу розповсюдження сигналу та часу передачі повідомлення. Прийемо, що швидкість розповсюдження електромагнітних хвиль у кабельній (електромагнітній або оптоволоконній) мережі складає c_k ; 200000км/с .

Тоді затримка передачі сигналу на відстань $r = 5000$ км дорівнює $\tau_p = \frac{r}{c_k} = 25$ мс .

Час передачі повідомлення $\Delta\tau_m$ визначається за виразом $\Delta\tau_m = \frac{t_s}{C_n}$, де $t_s = 200\text{Кбайт}$ – довжина повідомлення, а $C_n = 2\text{Мбит/с}$ – пропускна спроможність мережі. Тоді $\Delta\tau_m$; $\frac{200 \times 10^3 \times 8}{2 \times 10^6} = 25\text{мс}$.

Таким чином, загальна затримка передачі даних складає 825 мс.

Оцінимо додатковий час, який потрібно для передачі цього повідомлення по мережі з комутацією пакетів. Вважатимемо, що шлях від відправника до одержувача пролягає через 10 комутаторів. Початкове повідомлення розбивається на пакети в 1 Кбайт, всього 200 пакетів. Спочатку оцінимо затримку, яка виникає в початковому вузлі. Припустимо, що частка службової інформації, розміщеної в заголовках пакетів, по відношенню до загального об'єму повідомлення складає 10 %. Отже, додаткова затримка, пов'язана з передачею заголовків пакетів, складає 10 % від часу передачі цілого повідомлення, тобто 80 мс. Якщо прийняти інтервал між відправкою пакетів рівним 1 мс, тоді додаткові втрати за рахунок інтервалів складуть 200 мс. Тобто, в початковому вузлі через пакетування повідомлення при передачі виникла додаткова затримка в 280 мс.

Кожний з 10 комутаторів вносить затримку комутації, яка може мати великий розкид, від часток до тисяч мілісекунд. У даному прикладі приймемо, що на комутацію в середньому витрачається 20 мс. Крім того, при проходженні повідомлень через комутатор виникає затримка буферизації пакету. Ця затримка при величині пакету 1 Кбайт і пропускну спроможність лінії 2 Мбіт/с рівна 4 мс. Загальна затримка, що вноситься 10 комутаторами, складе приблизно 240 мс. В результаті додаткова затримка, створена мережею з комутацією пакетів, складе 520 мс. Враховуючи, що вся передача даних в мережі з комутацією каналів зайняла 825 мс, цю додаткову затримку можна вважати істотною.

2.14.2. Розрахунок коректності конфігурації мережі Ethernet

В цьому прикладі розглянуто найпоширеніші мережі Ethernet і Fast Ethernet, проте деякі з розглянутих обмежень будуть справедливі і в інших мережах (природно, з іншими значеннями параметрів). Для того, щоб мережа Ethernet могла функціонувати коректно, її конфігурація повинна задовольняти певним вимогам, які включають:

1) **Обмеження на максимальну/мінімальну довжину кабелю.** Основним недоліком будь-якого типу кабелю є загасання сигналу в кабелі. Якщо не використовувати повторювачі (концентратори), які ретранслюють і підсилюють сигнал, то відстань між будь-якими двома комп'ютерами в мережі з топологією "загальна шина" не може перевищувати деякого граничного значення (табл. 2.2). При топології "зірка" або "кільце" таке ж обмеження накладається на довжину кабелю комп'ютер-комп'ютер або комп'ютер-концентратор (hub). Існують також обмеження на мінімальну довжину кабелю між двома мережевими пристроями, що пов'язане з фізичними особливостями розповсюдження сигналу в кабелі.

2) **Обмеження на кількість комп'ютерів в одному сегменті мережі.** Сегмент утворюють комп'ютери, сполучені між собою за допомогою повторювачів (концентраторів). Два різні сегменти об'єднуються між собою за допомогою мостів, комутаторів, маршрутизаторів. У стандарті Ethernet передбачене обмеження на максимальне число комп'ютерів в одному сегменті мережі (табл. 2.2).

3) **Обмеження на кількість повторювачів між будь-якими двома комп'ютерами мережі.** Кількість повторювачів (концентраторів) між будь-якими двома комп'ютерами в мережі Ethernet не може бути більше чотирьох. Це обмеження називають "правилом чотирьох хабів". Обмеження пов'язане із затримками в розповсюдженні сигналу, які вносить повторювач (докладніше, пояснення див. в розрахунку PDV).

Обмеження на конфігурацію мережі Ethernet

Таблиця 2.2

Стандарт	10Base-5	10Base-2	10Base-T	10Base-F
Кабель	Товстий коаксіальний кабель RG-8 або RG-11	Тонкий коаксіальний кабель RG-58	Неекранована вита пара категорій 3, 4, 5	Багатомодовий волоконно-оптичний кабель
Максимальна довжина кабелю, м	500	185	100	2000
Мінімальна довжина кабелю, м	2,5	2,5	2,5	
Максимальна кількість комп'ютерів в одному сегменті	100	30	1024	1024
Максимальне число	4			

повторювачів між будь-якими станціями мережі	
PDV не більш	575 бітових інтервалів
PVV не більш	49 бітових інтервалів

4) Обмеження на час подвійного обороту сигналу (Path Delay Value, PDV). Правило "чотирьох хабів" є достатньо простим, проте гарантує коректність конфігурації мережі із зайвим "запасом". В деяких випадках можна побудувати мережу і з великим числом повторювачів між будь-якими двома комп'ютерами в мережі. Крім того, правило чотирьох хабів не розраховано на змішані мережі (коаксіальний кабель+вита пара+оптоволокну). Для точнішої перевірки використовується розрахунок часу подвійного обороту сигналу (PDV). Пояснимо термін PDV.

Чітке розпізнавання колізій всіма комп'ютерами мережі є необхідною умовою коректної роботи мережі Ethernet. Якщо який-небудь передаючий комп'ютер не розпізнає колізію і вирішить, що кадр даних переданий вірно, то цей кадр даних буде загублений. Швидше за все, загублений кадр буде повторно переданий яким-небудь протоколом верхнього рівня (транспортним або прикладним), але відбудеться це через значно триваліший інтервал часу (іноді навіть через декілька секунд), в порівнянні з мікросекундними інтервалами, якими оперує протокол Ethernet. Тому якщо колізії надійно не розпізнаватимуться вузлами мережі Ethernet, то це приведе до помітного зниження корисної пропускної спроможності мережі.

Оскільки швидкість розповсюдження електричного сигналу кінцева, то кожен метр кабелю вносить затримку в розповсюдження сигналу. Істотну затримку також вносять повторювачі, вимушені побітно приймати і підсилувати сигнал. Для спрощення розрахунків існує спеціальна таблиця, яка містить величини затримок, вказаних в бітових інтервалах (табл. 2). Сумарна величина PDV, розрахована по таблиці, не повинна перевищувати 575 бітових інтервалів. Для збільшення надійності мережі, на випадок відхилення параметрів кабелю і повторювачів, краще залишати "про запас" 4 бітові інтервали, тобто PDV не повинне перевищувати 571 бітовий інтервал. Дані для розрахунку значення PDV наведені в табл.2.3.

Таблиця 2.3

Тип сегменту	Повторювач** лівого сегменту, bt	Повторювачі проміжного сегменту, bt	Повторювачі правого сегменту, bt	Затримка*** середовища на 1 м кабелю, bt	Максимальна довжина сегменту, м	
10Base-5	11,8	46,5	169,5	0,0866	500	
10Base-2	11,8	46,5	169,5	0,1026	185	
10Base-T	15,3	42,0	165,0	0,113	100	
10Base-F	10Base-FB	-	24,0	-	0,1	2000
	10Base-FL	12,3	33,5	156,5	0,1	2000
	FOIRL	7,8	29,0	152,0	0,1	1000
AUI (> 2 м)	0	0	0	0,1026	2+48	

** У стандарті використовуються точніші терміни: база лівого, проміжного і правого сегментів.

*** Для того, щоб не потрібно було двічі складати затримки, що вносяться кабелем, в таблиці даються подвоєні величини затримок для кожного типу кабелю.

У таблиці використовуються також такі поняття, як лівий сегмент, правий сегмент і проміжний сегмент. Лівим сегментом вважається сегмент комп'ютера-відправника, а правим сегментом - сегмент комп'ютера-одержувача (рис. 2.51).

Оскільки лівий і правий сегменти мають різні величини затримки повторювачів, то у разі різних типів (коаксіальний, вита пара, оптоволокну) сегментів на видалених краях мережі необхідно виконати розрахунки двічі: один раз прийняти в якості лівого сегменту сегмент одного типу, а в другій - сегмент іншого типу. Результатом вважається максимальне значення PDV. У мережі, зображеній на малюнку обидва крайні сегменти належать до одного типу (10Base-T, вита пара) тому подвійний розрахунок не потрібен.

Розрахунок PDV для цієї мережі приведений нижче:

Лівий сегмент 1: 15,3 (повторювач) + 100 x 0,113 (кабель) = 26,6.

Проміжний сегмент 2: 33,5 (повторювач) + 1000 x 0,1 (кабель) = 133,5.

Проміжний сегмент 3: 24 (повторювач) + 500 x 0,1 (кабель) = 74,0.

Проміжний сегмент 4: 24 (повторювач) + 500 x 0,1 (кабель) = 74,0.

Проміжний сегмент 5: 24 (повторювач) + 600 x 0,1 (кабель) = 84,0.

Правий сегмент 6: 165 (повторювач) + 100 x 0,113 (кабель) = 176,3.

Разом PDV: 568,4 бітові інтервали.

Оскільки розрахункове значення PDV менше максимально допустимої величини 575, то ця мережа проходить по критерію часу подвійного обороту сигналу, не дивлячись на те, що кількість повторювачів –

більше 4-х. Проте, щоб визнати конфігурацію мережі коректною, потрібно також розрахувати зменшення міжкадрового інтервалу (PVV).

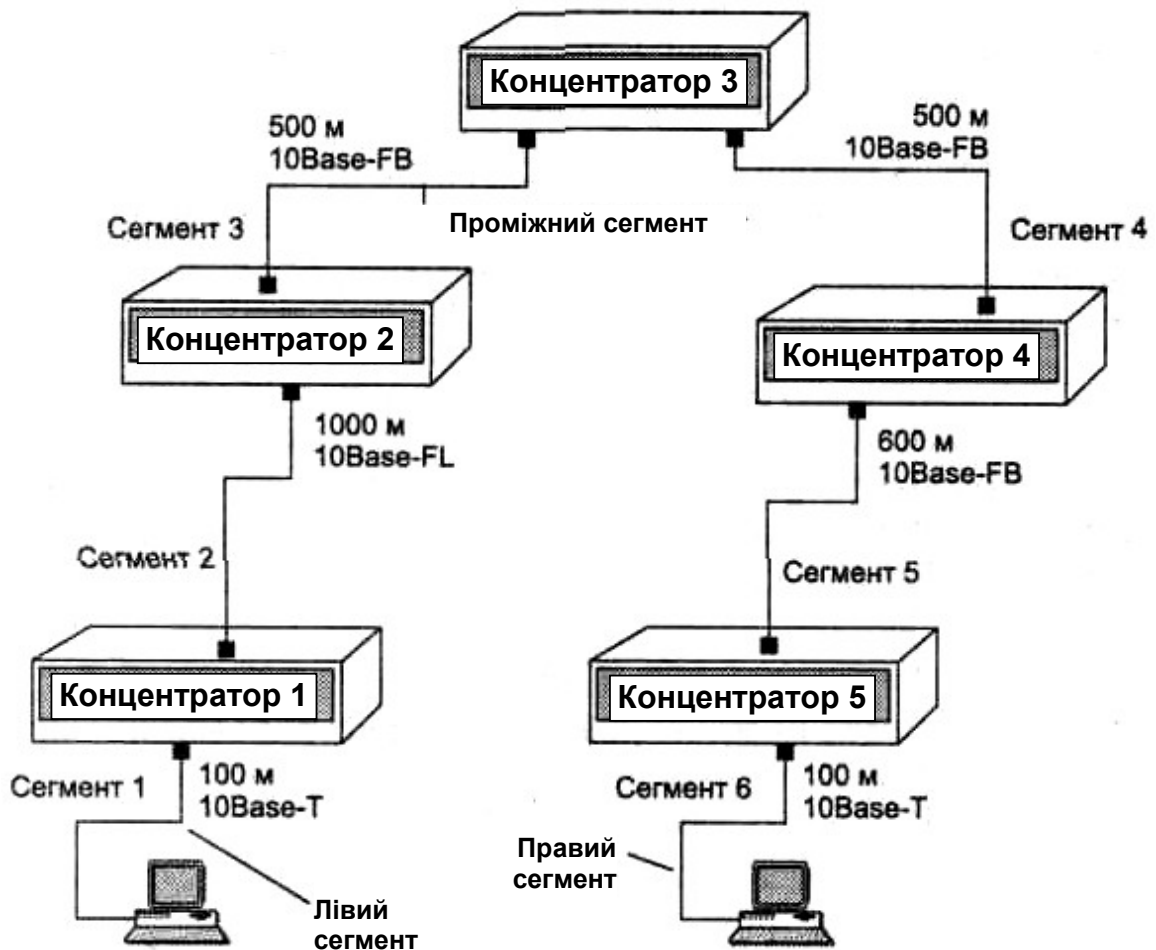


Рис. 2.51. Розрахунок коректності конфігурації мережі Ethernet.

5) Обмеження на скорочення міжкадрового інтервалу (Path Variability Value, PVV).

При відправці кадру, комп'ютери забезпечують початкову міжкадрову відстань в 96 бітових інтервалів. При проходженні через повторювачі, міжкадровий інтервал зменшується. Сумарне скорочення міжкадрового інтервалу (PVV) не повинне перевищувати 49 бітових інтервалів. Для розрахунку PVV використовується таблиця 2.4.

Таблиця 2.4

Скорочення міжкадрового інтервалу повторювачами

Тип сегменту	Передаючий сегмент, bt	Проміжний сегмент, bt
10Base-5 або 10Base-2	16	11
10Base-FB	-	2
10Base-FL	10,5	8
10Base-T	10,5	8

Відповідно до даних таблиці, розрахуємо значення PVV для нашого прикладу.

Лівий сегмент 1 10Base-T: скорочення в 10,5 bt.

Проміжний сегмент 2 10Base-FL: 8.

Проміжний сегмент 3 10Base-FB: 2.

Проміжний сегмент 4 10Base-FB: 2.

Проміжний сегмент 5 10Base-FB: 2.

Разом PVV: 24,5 бітові інтервали.

Розраховане значення PVV 24,5 менше граничного значення в 49 бітових інтервалу. В результаті, наведена в прикладі мережа відповідає стандартам Ethernet по всіх параметрах, хоча і включає більше чотирьох повторювачів.

2.14.3. Розрахунок коректності конфігурації мережі Fast Ethernet

Порядок розрахунку коректності конфігурації мережі Fast Ethernet декілька відрізняється від розрахунку мережі Ethernet, як по параметрах, так і по схемі розрахунку. Стандарт Fast Ethernet не підтримує коаксіальний кабель і мережа будується виключно по топології “зірка”. Обмеження на довжину кабелю комп'ютер, комп'ютер-комп'ютер приведені в табл. 2.5.

Таблиця 2.5

Обмеження на довжину кабелю в стандарті Fast Ethernet

Тип кабелю	Стандарт	До повторювача підключений	Максимальна довжина кабелю, м
Вита пара категорії 5	100Base-TX	-	100 м
Вита пара категорії 3, 4	100Base-T4	-	100 м
Багатомодове оптоволоконно 62,5/125 мкм	100Base-FX	Тільки оптоволоконний кабель	412 м (напівдуплекс) 2000 м (повний дуплекс)
		Один оптоволоконний кабель і декілька кабелів вита пара	160 м
		Декілька оптоволоконних кабелів і декілька кабелів вита пара	136 м

2.14.4. Обмеження на кількість повторювачів

Повторювачі Fast Ethernet діляться на два класи. Повторювачі класу 1 мають порти всіх типів (стандарт 100Base-TX, 100Base-FX і 100Base-T4). Повторювачі класу 2 мають або всі порти 100Base-T4, або порти 100Base-TX і 100Base-FX. Між будь-якими двома комп'ютерами в мережі може бути не більш двох повторювачів класу 2 або тільки один повторювач класу 1. Між собою повторювачі класу 1 повинні об'єднуватися за допомогою комутаторів, мостів, маршрутизаторів. Приклад конфігурації мережі Fast Ethernet приведений на рис. 2.52.

Приведених правил побудови мережі цілком достатньо для визначення коректності конфігурації мережі, оскільки ці правила вибрані з мінімальним "запасом міцності". Проте, при бажанні, можна провести і розрахунок PDV, виходячи з таких міркувань. Максимально допустима величина PDV = 512 бітових інтервалів. При розрахунку сегменти не діляться на правий і лівий. Для розрахунку беруться затримки, які вносять дві взаємодіючих через повторювач мережеві карти комп'ютерів (або мережева карта комп'ютера і порт комутатора). Також враховується затримка сигналу в повторювачі і затримка, що вноситься кабелем. Початкові дані для розрахунку приведені в табл. 2.6.

Таблиця 2.6

Розрахунок затримок розповсюдження сигналу

Затримка, що вноситься кабелем		Затримка, що вноситься мережевими картами		Затримка, що вноситься повторювачем	
Тип кабелю	Подвоєна затримка, bt на 1 м	Тип мережевих карт, що взаємодіють через повторювач	Подвоєна затримка, bt	Клас повторювача	Подвоєна затримка, bt
UTP Cat 3	1,14 bt	Два адаптери TX/FX	100 bt	1	140
UTP Cat 4	1,14 bt	Два адаптери T4	138 bt	2	92
UTP Cat 5	1,112 bt	Один адаптер TX/FX і один T4	127 bt		
Оптоволоконно	1,0 bt				

Розрахуємо, для прикладу, PDV між двома комп'ютерами, підключеними до повторювача 1 класу, який розташовано в правій частині рис. 2.52.

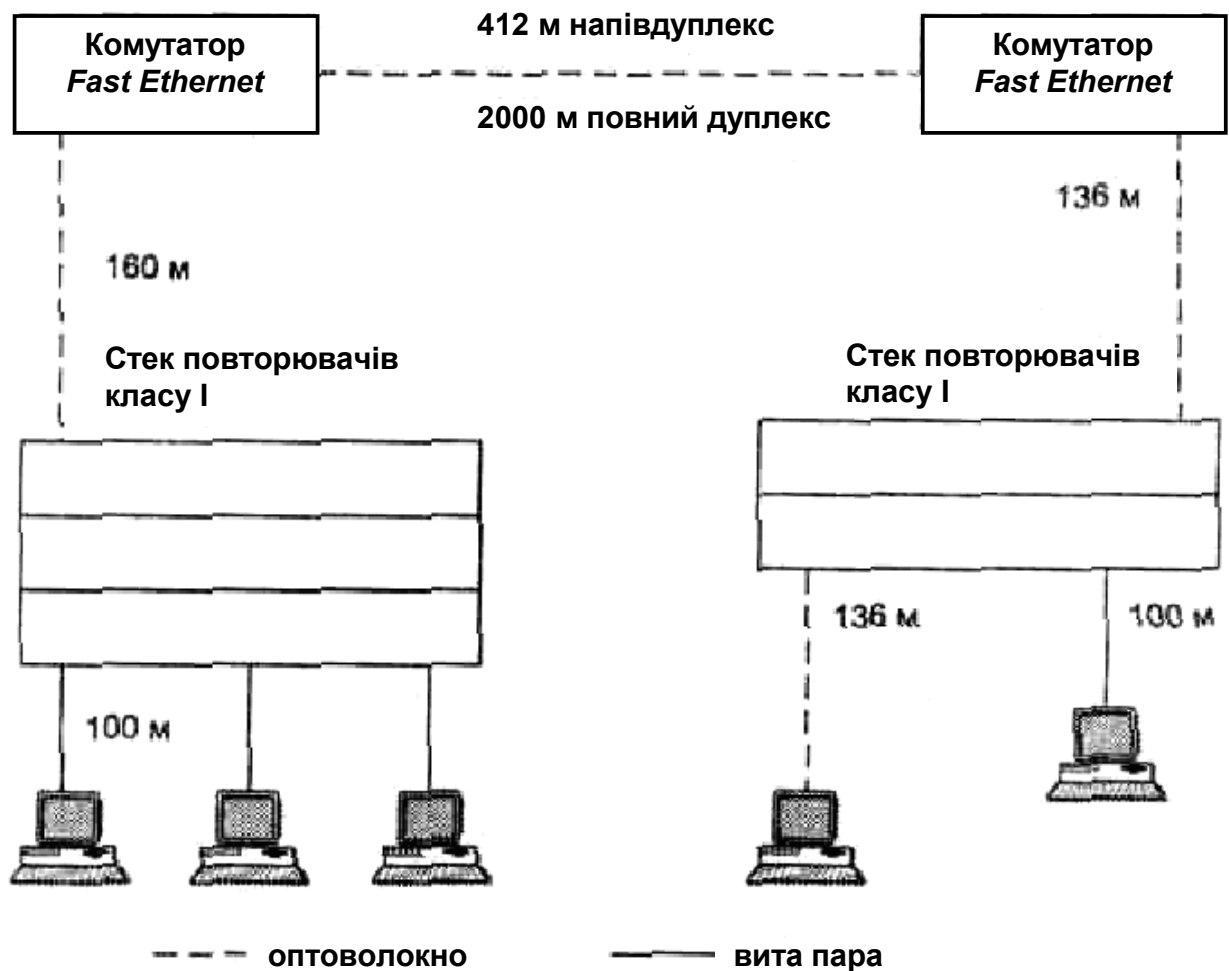


Рис. 2.52. Конфігурація мережі Fast Ethernet.

Припускаємо, що використовується вита пара 5-ої категорії (TX). $PDV=100*1,112$ (кабель) + $136*1,0$ (кабель) + 100 (мережеві карти) + 140 (повторювач) = $487,2 < 512$.

Розраховане значення PDV менше граничного значення в 512 бітових інтервалів, відповідно розрахунки поки не виявили некоректність конфігурації мережі. Для визнання конфігурації коректною, необхідно продовжити розрахунки для решти пар комп'ютер-комп'ютер, комп'ютер-комутатор.

2.14.5. Проектування логічної структури бази даних

Розглянемо процес проектування логічної структури бази даних. Для прикладу побудуємо структуру бази даних, яка може бути використана збереження інформації про кліматичні умови аеропорту.

Дослідження предметної області

Кожна інформаційна система в залежності від її призначення має справу з тією або іншою частиною реального світу, яку прийнято називати предметною областю системи. Дослідження предметної області є необхідним початковим етапом розробки будь-якої інформаційної системи. Саме на цьому етапі визначаються інформаційні потреби всієї сукупності користувачів майбутньої системи, що, у свою чергу, визначає зміст її бази даних.

Першим етапом проектування бази даних є аналіз предметної області, який закінчується побудовою логічної структури (концептуальної моделі) бази даних аеропорту. На даному етапі аналізуються запити користувачів, вибираються інформаційні об'єкти, їхні характеристики, і на основі проведеного аналізу структурується предметна область. Аналіз предметної області, який є загальнозначущим етапом, що не залежить від програмного і технічного середовищ, у яких буде реалізовуватися база даних кліматичних характеристик аеропорту, доцільно розбити на три етапи:

- аналіз концептуальних вимог і інформаційних потреб;
- виявлення інформаційних об'єктів і зв'язків між ними;
- побудова концептуальної моделі предметної області і проектування концептуальної схеми бази даних по кліматичних характеристиках аеропорту.

На етапі аналізу концептуальних вимог і інформаційних потреб метеорологічної служби аеропорту вирішуються такі задачі:

- аналіз вимог користувачів до бази даних;
- виявлення наявних **задач по опрацюванню** інформації, що повинна бути подана в базі даних;
- документування результатів аналізу.

Обстеження предметної **області** виконується сумісно із Замовником, майбутніми користувачами інформаційної системи. Вимоги користувачів до розроблювальної бази даних являють собою перелік об'єктів для збереження в базі даних, список їхніх атрибутів, вимоги до **вводу**, відновлення, редагування інформації, списку запитів із указівкою їхньої інтенсивності й обсягів даних. Ці дані отримуються в результаті діалогу зі співробітниками метеорологічної служби аеропорту. Вимоги користувачів уточнюються і доповнюються при аналізі наявних і перспективних потреб.

У процесі обстеження предметної **області** визначено, що база даних про кліматичні характеристики аеропорту може проектуватися для таких типів **задач**:

- одержання **даних** про метеорологічні явища і величини, які впливають на роботу авіації;
- розрахунок сезонів року або частини доби зі **сприятливими** і несприятливими умовами для **зльоту і посадки ПК**;
- складання кліматичної характеристики аеропорту за даними щогодинних багаторічних спостережень;
- одержання графічної інформації, що відбиває кліматичні умови аеропорту (графіки повторюваності будь-яких кліматичних характеристик, **троянди вітрів** і т.д.).

Відзначені задачі характеризуються **такими** інформаційними об'єктами:

- інформація про кліматичні характеристики щоденника спостережень;
- інформація про особливі метеорологічні явища;
- **дані** про максимуми і мінімуми температури й опадів;
- інформація довідкового характеру.

Визначено, що необхідно одержувати **відомості** по **всіх** кліматичних характеристиках аеропорту: рік, місяць, число, час спостережень; напрямок, швидкість, пориви вітру у землі; напрямок, швидкість вітру на **висоті 100м**; напрямок, швидкість вітру на колі; горизонтальна дальність видимості; хмарність; температура; температура **точки** роси; вологість відносна й абсолютна; тиск; барометрична тенденція; **відомості** про небезпечні метеорологічні явища; курс **посадки**; прогноз на **посадку**; температура; **відомості про опади**; **відомості** про середні, максимальні і мінімальні значення параметрів. Потрібно **одержувати інформацію про повторюваність метеорологічних характеристик, троянди вітрів у вигляді діаграм, графіків.**

Визначення переліку атрибутів та їх характеристик

Друга фаза аналізу предметної **області** полягає у виборі інформаційних об'єктів. Потрібно вивчити усі вхідні і вихідні документи, наставляння, форми, довідки, повідомлення, що циркулюють при метеорологічному забезпеченні польотів:

- інструкція з метеорологічного забезпечення польотів;
- інструкція інженера-синоптика з технології роботи автоматичної метеорологічної станції;
- **загальна** інструкція з технології роботи автоматичної метеорологічної станції;
- збірник міжнародних метеорологічних авіаційних кодів (METAR, SPECI, TAF);
- наставляння **гідрометеорологічним** службам і **постам**;
- наставляння по виконанню польотів;
- щоденники спостережень **УАМЦ-Бориспіль** і **АМСЦ-Київ**;
- рекомендації по веденню щоденника погоди;
- оперативні зведення погоди кодів METAR, SPECI, TAF.

Виділення інформаційних об'єктів – процес ітеративний. Він здійснюється на підставі аналізу перерахованих вище джерел і діалогу з користувачами. Визначення інформаційних об'єктів починається з упорядкування переліку атрибутів. Перелік атрибутів являє собою таблицю, в котру заносяться ім'я атрибута, розгорнутий коментар для уточнення змісту атрибута і **зведення** про документи, у яких фігурує цей атрибут, якщо це необхідно. Після **опрацювання** вихідної інформації формується перелік атрибутів, тобто характеристик інформаційних об'єктів, поданий у колонках 1 і 2 таблиці 2.7.

Далі для кожного атрибута **встановлюються** і **записуються** основні характеристики:

- шаблон значень атрибута;
- **область припустимих** значень.

Шаблон значень атрибута визначає категорію і довжину значень атрибута. Значення можуть бути різних типів: числового, текстовими, типу "дата" і т.д. У таблиці 2.7 поданий перелік атрибутів із типами даних (колонка 3) та припустимою значністю (колонки 4, 5, 6). **Область припустимих** значень задається обов'язково для кожного атрибута. **Областю** може бути перелік можливих значень, інтервали значень, що використовуються в алгоритмі перевірки коректності значення.

Отже, були **введені такі** обмеження на значення атрибутів:

1. Значення атрибута "рік **спостережень**" задається цілим позитивним **чотирицифровим** числом.
2. Значення атрибута "місяць **спостережень**" задається значеннями: "січень", "лютий", "березень", "квітень", "травень", "червень", "липень", "серпень", "вересень", "жовтень", "листопад", "грудень".

3. Значення атрибута "день місяця **спостережень**" задається цілим позитивним числом із значенням від 1 до 31.
4. Значення атрибута "**час спостережень**" задається цілим позитивним числом із значенням від 0 до 24.
5. Значення атрибута "напрямок вітру у землі" задається текстовою величиною довжиною не більше п'яти знаків.
6. Значення атрибута "швидкість вітру у землі" задається позитивним числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
7. Значення атрибута "**пориви** вітру у землі" задається позитивним числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
8. Значення атрибута "напрямок вітру на **висоті 100м**" задається текстовою величиною довжиною не більше п'яти знаків.
9. Значення атрибута "швидкість вітру на **висоті 100м**" задається позитивним числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
10. Значення атрибута "напрямок вітру на колі" задається текстовою величиною довжиною не більше п'яти знаків.
11. Значення атрибута "швидкість вітру на колі" задається позитивним числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
12. Значення атрибутів "горизонтальна дальність видимості по **приладам**", "горизонтальна дальність видимості - група яскравості", "горизонтальна дальність видимості **ВВІ**" задається текстовою величиною довжиною не більше шести знаків
13. Значення атрибута "**загальна** кількість **хмар**" задається позитивним числом із значенням від 0 до 10.
14. Значення атрибута "кількість **хмар нижнього** ярусу" задається позитивним числом із значенням від 0 до 10.
15. Значення атрибута "**висота нижньої межі хмар**" задається позитивним числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
16. Значення атрибута "температура повітря" задається числом із значенням від –100 до +100.
17. Значення атрибута "температура **точки роси**" задається числом із значенням від –100 до +100.
18. Значення атрибута "вологість відносна" задається числом із значенням від 0 до 100.
19. Значення атрибута "вологість абсолютна" задається числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
20. Значення атрибута "**тиск**" задається числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
21. Значення атрибута "**барометрична** тенденція" задається числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
22. Значення атрибута "курс **посадки**" задається числом із значенням від –360 до +360.
23. Значення атрибута "особливі **явища** погоди – **початок (хвилини)**" задається цілим позитивним числом із значенням від 0 до 60.
24. Значення атрибута "особливі **явища** погоди – кінець (день місяця)" задається цілим позитивним числом із значенням від 1 до 31.
25. Значення атрибута "особливі **явища** погоди – кінець (**години**)" задається цілим позитивним числом із значенням від 0 до 24.
26. Значення атрибута "особливі **явища** погоди – кінець (**хвилини**)" задається цілим позитивним числом із значенням від 0 до 60.
27. Значення атрибута "особливі **явища** погоди – інтенсивність" задається значеннями "0", "1", "2".
28. Значення атрибута "мінімум температур за день" задається числом із значенням від –100 до +100.
29. Значення атрибута "**опад** в 9.00 **годин**" задається числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.
30. Значення атрибута "максимум температур за день" задається числом із значенням від –100 до +100.
31. Значення атрибута "**опад** в 21.00 **годину**" задається числом, у якому ціла частина – тризначне число, а **дробова** – однозначне.

Таблиця 2.7

Перелік атрибутів бази даних кліматичних характеристик аеропорту

№	Назва атрибута	Тип атрибута	Кількість знаків	Ціла частина	Дробова частина	Ідентифікатор
1	2	3	4	5	6	7
1.	рік спостережень	числовий	4	4		GOD
2.	місяць спостережень	текстовий	8			MES

3.	день місяця спостережень	числовий	2	2		DAYM
4.	час спостережень	числовий	2	2		CH
5.	напрямок вітру у землі	текстовий	5			VN
6.	швидкість вітру у землі	числовий	5	3	1	VS
7.	пориви вітру у землі	числовий	5	3	1	VP
8.	напрямок вітру на висоті 100м	текстовий	5			V100
9.	швидкість вітру на висоті 100м	числовий	5	3	1	V100S
10.	напрямок вітру на колі	текстовий	5			VK
11.	швидкість вітру на колі	числовий	5	3	1	VKS
12.	горизонтальна дальність видимості по приладам	текстовий	6			GVDVP
13.	горизонтальна дальність видимості - група яскравості	текстовий	6			GVG
14.	горизонтальна дальність видимості ВВІ	текстовий	6			GVDV
15.	загальна кількість хмар	числовий	3	1	1	OKO
16.	кількість хмар нижнього ярусу	числовий	3	1	1	OKN
17.	форма хмар	текстовий	16			OF
18.	висота нижньої межі хмар	числовий	5	3	1	OVNG
19.	температура повітря	числовий	5	3	1	T
20.	температура точки роси	числовий	5	3	1	TTR
21.	вологість відносна	числовий	4	2	1	VO
22.	вологість абсолютна	числовий	5	3	1	VA
23.	тиск	числовий	5	3	1	D
24.	барометрична тенденція	числовий	5	3	1	BT
25.	зведення про зледеніння і бовтанку	текстовий	15			SV
26.	курс посадки	числовий	5	3	1	KP
27.	додаткова інформація	текстовий	50			DI
28.	прогноз на посадку	текстовий	30			PP
29.	особливі явища погоди – початок (хвилини)	числовий	2	2		NMIN
30.	особливі явища погоди – кінець (день місяця)	числовий	2	2		DAYMK
31.	особливі явища погоди – кінець (години)	числовий	2	2		KCH
32.	особливі явища погоди – кінець (хвилини)	числовий	2	2		KMIN
33.	особливі явища погоди – назва явища	текстовий	50			OSOB
34.	особливі явища погоди – інтенсивність	текстовий	1			INTEN
35.	особливі явища погоди – видимість	текстовий	4			VID
36.	мінімум температури за день	числовий	5	3	1	MINIMUM
37.	опади в 9.00 годин	числовий	5	3	1	OSAD09H
38.	максимум температури за день	числовий	5	3	1	MAX
39.	опади в 21.00 годину	числовий	5	3	1	OSAD21H

Зведення про максимуми і мінімуми температури й опади за день по аеропорту можна замінити одним об'єктом. Тоді початкові атрибути: мінімум температури за день, опади в 9.00 часів, максимум температури за день, опади в 21.00 час можливо замінити значеннями нового – узагальненого об'єкта: відомості про максимуми і мінімуми температури і опадів за день. Атрибути, що залишилися, можна об'єднати в об'єкт: відомості щоденника спостережень.

У такий спосіб при осмисленні предметної області виділяються наступні інформаційні об'єкти: ВІДОМОСТІ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ, ВІДОМОСТІ ПРО МАКСИМУМИ І МІНІМУМИ ЗА ДЕНЬ, ВІДОМОСТІ ПРО ОСОБЛИВІ ЯВИЩА. Для кожного об'єкта зазначено відповідний набір атрибутів.

У процесі визначення інформаційних об'єктів можна виділити декілька нових об'єктів типу "довідник" для різних атрибутів. Наприклад, довідник особливих метеорологічних явищ, довідник форм хмар, довідник видимості по приладам, довідник напрямків вітру. Всі ці об'єкти мають однакову структуру і характеризуються ідентичним використанням, у зв'язку з чим цілком логічно в схемі предметної області лишити один об'єкт "Довідники".

На цьому ж етапі **роблять** так назване зовнішнє кодування. Воно **полягає** в заміні текстових значень атрибутів короткими ідентифікаторами. У табл. 2.7 поданий перелік атрибутів із ідентифікаторами (колонка 7).

Встановлення і опис запитів

Визначення **всіх** запитів користувача до бази даних виконується в контактї з Замовником. **В результаті чого здійснюється** упорядкування повного переліку запитів для кожного фрагмента предметної області. За кожним запитом **стоять визначені** процеси **опрацювання** даних і запитальні зв'язки, по яких повинна відбуватися навігація від об'єкта до об'єкта для задоволення запиту. Аналіз і формалізація процесів **опрацювання** інформації передують в проектуванні встановленню структурних зв'язків між об'єктами. Це **пояснюється** тим, що одне з основних призначень структурних зв'язків – забезпечити **виконання всіх** процесів **опрацювання** інформації з усіх фрагментів предметної області.

Проектування концептуальної моделі бази даних кліматичних умов аеропорту

Заключний етап аналізу предметної області складається в проектуванні її концептуальної схеми або моделі. При виборі моделі враховується та обставина, що будь-яким моделям властиві **визначені** обмеження, тому **пошук** ідеальної моделі, що цілком відбиває реальний **світ**, дуже проблематичний. Вибір моделі диктується насамперед характером предметної області і вимогам до бази даних. **Іншою** немаловажною обставиною є незалежність концептуальної моделі від системи **керування** базами даних, яка повинна бути **обрана** після побудови концептуальної моделі.

Абстрактна інформаційна модель, тобто виділені в ній сутності, атрибути, зв'язки описуються **засобами** моделі даних, що підтримує СКБД. Практично усі СКБД персональних комп'ютерів підтримують **реляційну** модель даних – найбільше популярний засіб збереження даних, який використано для розробленої бази даних кліматичних характеристик аеропорту. В основі реляційної моделі лежить поняття відношення, яке зручно уявляти в вигляді таблиці. Набір **відношень** (таблиць) може бути використаний для збереження даних про об'єкти, що існують при метеорологічному забезпеченні польотів, і моделюванні зв'язків між ними.

Виділені атрибути повинні бути **агреговані** в інформаційні об'єкти. **Угруповання** об'єктів повинно бути раціональним, тобто **мінімізуючим** дублювання даних і зрощуючим процедури їх **опрацювання** і відновлення. Саме на цьому етапі виконуються процедури нормалізації **відношень**. **Склад** атрибутів **відношень** бази даних кліматичних характеристик аеропорту повинний задовольняти двом основним вимогам: між атрибутами не повинно бути небажаних функціональних залежностей; **угруповання** атрибутів повинно забезпечувати мінімальне дублювання даних. Задоволення цих вимог досягається нормалізацією **відношень** бази даних. Приведемо відношення послідовно до третьої нормальної форми.

У табл. 2.7 наведене **відношення, приведенне до першої нормальної форми** – в ньому подано прості атрибути. Позначимо його R1NF. Для його наступної нормалізації необхідно для визначити первинний ключ. На роль первинного складового ключа цього відношення претендують атрибути: рік спостережень, місяць спостережень, день місяця спостережень, години спостережень, особливі явища погоди – початок (хвилини).

З погляду нормалізації **відношення**, поданого в табл. 2.7, атрибути з **6-го** по **28-й** знаходяться в частковій функціональній залежності від первинного ключа **відношення** R1NF, оскільки вони функціонально залежать від частини ключа – рік спостережень, місяць спостережень, день місяця спостережень, години спостережень.

Атрибути з **30-го** по **35-й** знаходяться в повній функціональній залежності від усього ключа **відношення** R1NF: рік спостережень, місяць спостережень, день місяця спостережень, години спостережень, особливі явища погоди – початок (хвилини).

Атрибути з **36-го** по **39-й** не знаходяться в повній функціональній залежності від усього ключа **відношення** R1NF, оскільки вони функціонально залежать від частини ключа – рік спостережень, місяць спостережень, день місяця спостережень.

Усунемо у відношенні R1NF всі часткові залежності неключових атрибутів від ключа, тобто залишимо тільки повні функціональні залежності неключових атрибутів від ключа. **Після всіх** перетворень одержимо три **відношення** в другій нормальної формі: R2NF_OSN (**ВІДОМОСТІ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ**), R2NF_OSOB (**ВІДОМОСТІ ПРО ОСОБЛИВІ ЯВИЩА**) і R_2NF_MM (**ВІДОМОСТІ ПРО МАКСИМУМИ І МІНІМУМИ ЗА ДЕНЬ**).

Відношення R2NF_OSN містить відповідно атрибути: **ВІДОМОСТІ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ**(**рік спостережень, місяць спостережень, день місяця спостережень, години спостережень**, напрямок вітру у землі, швидкість вітру у землі, пориви вітру у землі, напрямок вітру на **висоті 100м**, швидкість вітру на **висоті 100м**, напрямок вітру на колі, швидкість вітру на колі, горизонтальна дальність видимості по **приладам**, горизонтальна дальність видимості - група яскравості, горизонтальна дальність видимості **ВВІ**, загальна кількість хмар, кількість хмар **нижнього** ярусу, форма хмар, **висота нижньої межі** хмар, температура повітря, температура **точки** роси, вологість відносна, вологість абсолютна, тиск, барометрична тенденція, **зведення** про зледеніння і бовтанку, курс **посадки**, додаткова інформація, прогноз на **посадку**);

Відношення R2NF_OSОВ містить відповідно атрибути: ВІДОМОСТІ ПРО ОСОБЛИВІ ЯВИЩА(рік спостережень, місяць спостережень, день місяця спостережень, години спостережень, особливі явища погоди – початок (хвилини), особливі явища погоди – кінець (години), особливі явища погоди – кінець (хвилини), особливі явища погоди – назва явища, особливі явища погоди – інтенсивність, особливі явища погоди – видимість);

Відношення R2NF_MM містить відповідно атрибути: ВІДОМОСТІ ПРО МАКСИМУМИ І МІНІМУМИ ЗА ДЕНЬ(рік спостережень, місяць спостережень, день місяця спостережень, мінімум температури за день, опади в 9.00 годин, максимум температури за день, опади в 21.00 годину).

Між об'єктами досліджуваної предметної області ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ і ЗВЕДЕННЯ ПРО ОСОБЛИВІ ЯВИЩА існує зв'язок "один-до-багатьох" – на протязі одного часу може відбутися декілька особливих явищ погоди; між об'єктами ЗВЕДЕННЯ МАКСИМУМАХ І МІНІМУМАХ ЗА ДЕНЬ і ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ також існує зв'язок "один-до-багатьох" – тільки одне значення максимуму і мінімуму визначається протягом усього дня. Для об'єднання пов'язаних об'єктів ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ і ЗВЕДЕННЯ ПРО ОСОБЛИВІ ЯВИЩА, ЗВЕДЕННЯ ПРО МАКСИМУМИ І МІНІМУМИ ЗА ДЕНЬ служить зовнішній ключ рік спостережень, місяць спостережень, день місяця спостережень. Таким чином, було виділено три відношення: ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ; ЗВЕДЕННЯ ПРО ОСОБЛИВІ МЕТЕОРОЛОГІЧНІ ЯВИЩА; ЗВЕДЕННЯ ПРО МАКСИМУМИ І МІНІМУМИ ЗА ДЕНЬ.

Тільки в відношенні ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ є транзитивна функціональна залежність: атрибут “форма хмар (на українській мові)” від “форма хмар (абревіатура на латинській мові)”, а останній атрибут – від усього складеного ключа цього відношення. Таким чином, відношення ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ потребує усунення транзитивної залежності. В результаті отримуємо два відношення: R3NF_OSN_2 (ЗВЕДЕННЯ ЩОДЕННИКА СПОСТЕРЕЖЕНЬ_2) і R3NF_FH (ФОРМА ХМАР). Відношення R3NF_OSN_2 містить відповідно усі атрибути, крім “форма хмар (на українській мові)”, відношення R3NF_FH містить атрибути: “форма хмар (абревіатура на латинській мові)”, “форма хмар (на українській мові)”.

Концептуальна схема бази даних кліматичних умов аеропорту подана на рис. 2.53. При відображенні концептуальної схеми на логічну реляційну схему кожний прямокутник схеми відображається у відповідне відношення – таблицю.

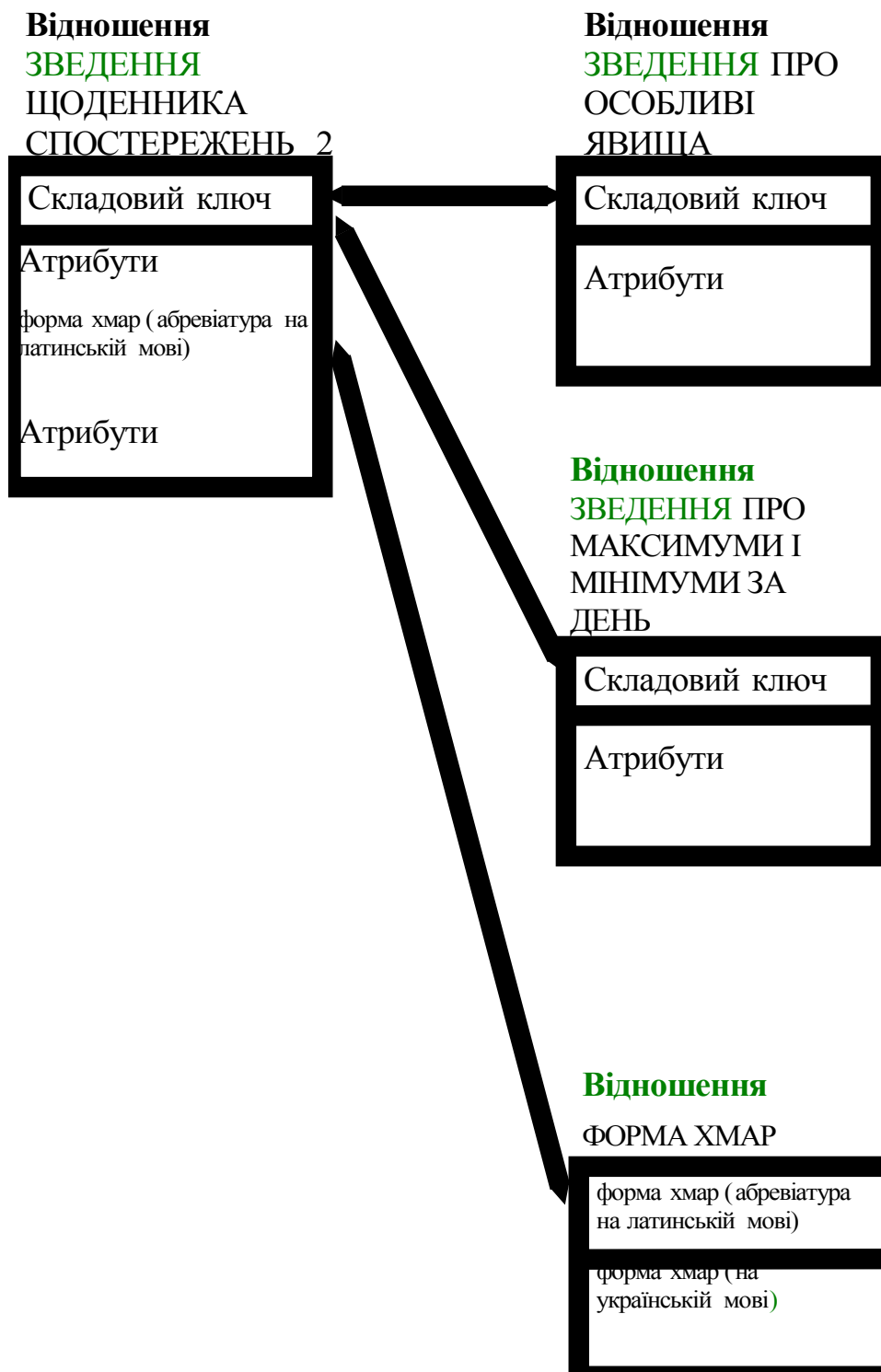


Рис. 2.53. Концептуальна схема бази даних кліматичних умов аеропорту.

Нормалізовані відношення призначені для формування структури основних файлів бази даних кліматичних умов аеропорту: кліматичні характеристики щоденника спостережень; особливі метеорологічні явища; максимуми і мінімуми за день.

Структура файла бази даних "Кліматичні характеристики щоденника спостережень", відповідного відношенню R2NF_OSN, подана в табл. 2.8.

Таблиця 2.8

Структура файла бази даних "кліматичні характеристики щоденника спостережень"

№	Назва атрибута	Тип атрибута	Кількість знаків	Ціла частина	Дробова частина	Ідентифікатор
---	----------------	--------------	------------------	--------------	-----------------	---------------

1.	рік спостережень	числовий	4	4		GOD
2.	місяць спостережень	текстовий	8			MES
3.	день місяця спостережень	числовий	2	2		DAYM
4.	час спостережень	числовий	2	2		CH
5.	напрямок вітру у землі	текстовий	5			VN
6.	швидкість вітру у землі	числовий	5	3	1	VS
7.	пориви вітру у землі	числовий	5	3	1	VP
8.	напрямок вітру на висоті 100м	текстовий	5			V100
9.	швидкість вітру на висоті 100м	числовий	5	3	1	V100S
10.	напрямок вітру на колі	текстовий	5			VK
11.	швидкість вітру на колі	числовий	5	3	1	VKS
12.	горизонтальна дальність видимості по приладам	текстовий	6			GVDVP
13.	горизонтальна дальність видимості - група яскравості	текстовий	6			GVG
14.	горизонтальна дальність видимості ОВИ	текстовий	6			GVDV
15.	загальна кількість хмар	числовий	3	1	1	OKO
16.	кількість хмар нижнього ярусу	числовий	3	1	1	OKN
17.	форма хмар	текстовий	16			OF
18.	висота нижньої межі хмар	числовий	5	3	1	OVNG
19.	температура повітря	числовий	5	3	1	T
20.	температура точки роси	числовий	5	3	1	TTR
21.	вологість відносна	числовий	4	2	1	VO
22.	вологість абсолютна	числовий	5	3	1	VA
23.	тиск	числовий	5	3	1	D
24.	барометрична тенденція	числовий	5	3	1	BT
25.	зведення про зледеніння і бовтанку	текстовий	15			SV
26.	курс посадки	числовий	5	3	1	KP
27.	додаткова інформація	текстовий	50			DI
28.	прогноз на посадку	текстовий	30			PP

Структура файлу бази даних "особливі метеорологічні явища", відповідного відношенню R2NF_OSOB, подана в табл. 2.9.

Таблиця 2.9

Структура файлу бази даних "особливі метеорологічні явища"

№	Назва атрибута	Тип атрибута	Кількість знаків	Ціла частина	Дробова частина	Ідентифікатор
1.	рік спостережень	числовий	4	4		GOD
2.	місяць спостережень	текстовий	8			MES
3.	день місяця спостережень	числовий	2	2		DAYM
4.	особливі явища погоди – початок (години)	числовий	2	2		NCH
5.	особливі явища погоди – початок (хвилини)	числовий	2	2		NMIN
6.	особливі явища погоди – кінець (день місяця)	числовий	2	2		DAYMK
7.	особливі явища погоди – кінець (години)	числовий	2	2		KCH
8.	особливі явища погоди – кінець (хвилини)	числовий	2	2		KMIN
9.	особливі явища погоди – назва явища	текстовий	50			OSOB
10.	особливі явища погоди – інтенсивність	текстовий	1			INTEN
11.	особливі явища погоди – видимість	текстовий	4			VID

Структура файлу бази даних, відповідного відношенню R2NF_MM, подана в табл. 2.10.

Таблиця 2.10

Структура файлу бази даних "максимуми і мінімуми за день"

№	Назва атрибута	Тип атрибута	Кількість знаків	Ціла частина	Дробова частина	Ідентифікатор
1.	рік спостережень	числовий	4	4		GOD
2.	місяць спостережень	текстовий	8			MES

3.	день місяця спостережень	числовий	2	2		DAYM
4.	мінімум температури за день	числовий	5	3	1	MINIMUM
5.	опади в 9.00 годин	числовий	5	3	1	OSAD09H
6.	максимум температури за день	числовий	5	3	1	MAX
7.	опади в 21.00 годину	числовий	5	3	1	OSAD21H

Структура файлу бази даних, відповідного відношенню R3NF_FH, подана в табл. 2.11.

Структура файлу бази даних "форма хмар"

Таблиця 2.11

№	Назва атрибута	Тип атрибута	Кількість знаків	Ціла частина	Дробова частина	Ідентифікатор
1.	форма хмар (абревіатура на латинській мові)	текстовий	10			OF
2.	форма хмар (на українській мові)	текстовий	20			OFU

Подані структури файлів бази даних засобами обраної СУБД можуть бути введені в ЕОМ. Модифікація і вибір даних здійснюється за допомогою запитів безпосередньо на потужній непроцедурній мові запитів SQL.

2.15. Задачі і вправи

1. Вкажіть в таблиці 2.12 застосування того або іншого типу кабелю для різних підсистем.

Таблиця 2.12

Дані	Горизонтальна система	Вертикальна система	Підсистема кампусу
Неекранована вита пара			
Екранована вита пара			
Товстий коаксіальний кабель			
Тонкий коаксіальний кабель			
Оптоволоконний кабель			
Бездротовий зв'язок			

2. Хай IP-адреса деякого вузла підмережі рівна 198.65.12.67, а значення маски для цієї підмережі – 255.255.255.240. Визначте номер підмережі. Яке максимальне число вузлів може бути в цій підмережі?

3. Підприємство вирішило створити власну глобальну мережу. Який тип глобальних зв'язків буде найбільш ефективний, якщо підприємству необхідно з'єднати локальну мережу в штаб-квартирі з трьома локальними мережами регіональних підрозділів, розташованими в різних містах? Середня інтенсивність трафіку між мережами підрозділів і центральною мережею оцінюється діапазоном значень від 500 Кбіт/с до 1 Мбіт/с.

4. Кабель (вита пара категорії 5) сполучає два елементи мережі, що знаходяться на відстані 50 м друг те друга. Визначити загасання сигналу.

5. Для передачі даних з необхідною достовірністю на відстань 2000 м необхідно, щоб загасання в кабелі не перевищувало 0,5 дБ. Виберіть тип кабелю і довжину хвилі передачі (рис. 2.54). Охарактеризуйте режим розповсюдження променя в кабелі (одномодовий або багатомодовий).

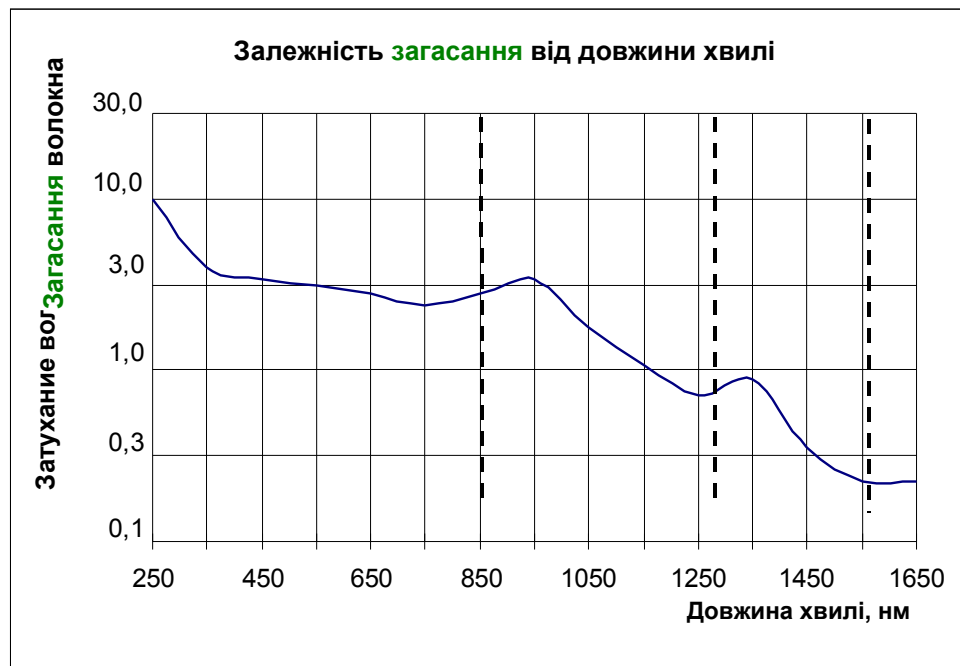


Рис. 2.54. Залежність загасання від довжини хвилі.

6. Визначити необхідну смугу пропускання цифрової лінії передачі, якщо кількість джерел, які передають дані, дорівнює 10, ширина спектру початкового (аналогового) сигналу – 3400 Гц, а частотне рознесення між сусідніми цифровими каналами – не менше 2400 Гц. Яка теорема використовується для визначення необхідної частоти дискретизації безперервного сигналу?

7. Визначити середню потужність помилки квантування безперервного сигналу, якщо динамічний діапазон сигналу (діапазон від мінімального до максимального значень напруги) рівний 2,4 V, а розрядність коду рівна 8.

8. Визначити середнє, максимальне і мінімальне значення помилки квантування безперервного сигналу, якщо динамічний діапазон сигналу (діапазон від мінімального до максимального значень напруги) рівний 2,4 V, а розрядність коду рівна 8.

2.14. Список літератури для модуля 2

1. Федоров А., Елманова Н. Введение в базы данных. Часть 2. Настольные СУБД //Комп'ютер-пресс. -2000. -№4. -с.123-127
2. Федоров А., Елманова Н. Введение в базы данных. Часть 3. Серверные СУБД //Комп'ютер-пресс. -2000. -№5. -с.143-150.
3. Федоров А., Елманова Н. Введение в базы данных. Часть 4. Механизмы доступа к данным. Borland Database Engine и альтернативы //Комп'ютер-пресс. -2000. -№6. -с.146-153.
4. Дейт К. Дж. Введение в системы баз данных, 7-е издание.: Пер. с англ. -М.: Издательский дом "Вильямс", 2001. -1072с.
5. Мамаев Е., Вишневский А. Microsoft SQL Server 7 для профессионалов. -СПб.: Питер, 2001.-896с.
6. Архипенков С. Аналитические системы на базе Oracle Express OLAP. Проектирование, создание, сопровождение. -М.: ДИАЛОГ-МИФИ, 1999. -320с.
7. Microsoft SQL Server 6. Transact SQL Reference. Microsoft Corporation, 1995. -728с.
8. Мартин Дж. Организация баз данных в вычислительных системах.: Пер. с англ. -М.: Мир, 1980. -662с.
9. Ульман Дж. Основы систем баз данных: Пер. с англ. - М.: Финансы и статистика, 1983. - 334с.
10. Гусева Т.И., Башинов Ю.Б. Проектирование баз данных в примерах и задачах. -М.: Радио и связь, 1992. -160с.
11. Осуга С. Обработка знаний: Пер. с япон. - М.: Мир, 1989. -293с.
12. Уотермен Д. Руководство по экспертным системам: Пер. с англ. - М.: Мир, 1989. -264с.
13. Элти Дж., Кумбс М. Экспертные системы: концепции и примеры. - М.: Финансы и статистика, 1987.-191с.
14. Богуславский Л.Б., Дрожжинов В.И. Основы построения вычислительных сетей для автоматизированных систем. – М.: Энергоатомиздат, 1990. – 256 с. :ил.
15. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003. – 512 с.

3. Методичні вказівки для курсового проектування по дисципліні Комп'ютерні системи і мережі»

3.1. Цілі і задачі курсового проектування

Курсове проектування повинне сприяти закріпленню, поглибленню і узагальненню знань, одержаних студентами в процесі вивчення лекційного курсу по дисципліні "Комп'ютерні системи та мережі", а також умінь і навиків, одержаних при виконанні практичних і лабораторних робіт, і застосуванню цих знань, умінь і навиків для рішення конкретних інженерних задач, розвитку навиків роботи із спеціальною літературою і навиків інженерного проектування. Дисципліна "Комп'ютерні системи та мережі" спирається на знання, одержані при вивченні наступних дисциплін: "Теоретичні основи інформаційних процесів", "Теоретичні і технічні основи обчислювальних машин і обчислювальних мереж".

3.2. Тематика і зміст курсових проектів

Завдання на курсове проектування присвячене проектуванню обчислювальних мереж (ОМ) як основи комплексу технічних засобів інформаційних систем різних предметних областей (організацій, підприємств, установ і їх підрозділів). При цьому передбачається закріплення знань і навиків системного проектування. При виконанні курсового проекту студент повинен:

- провести порівняльний аналіз різних варіантів архітектури ОМ з системних позицій по основних параметрах: швидкодія, надійність, інформаційна безпека, вартість;
- розробити структурну схему ОМ (для завдань по проектуванню локальних обчислювальних мереж) або структуру апаратного і програмного забезпечення для надання вибраного переліку послуг глобальної обчислювальної мережі (для завдання по визначенню способу використання глобальної обчислювальної мережі);
- оформити пояснювальну записку і графічну частину проекту.

Крім того, за рішенням кафедри до складу проекту можуть бути включені додаткові розділи, пов'язані з науково-дослідною роботою.

3.3. Завдання по курсовому проектуванню

Індивідуальне завдання на курсове проектування складається з одного з двох основних варіантів проектів ОМ (табл. 2.13) і варіанту предметної області, в якій повинна функціонувати інформаційна система (табл. 2.14).

Таблиця 2.13

Варіант 1	Розробити проект локальної обчислювальної мережі як основи комплексу технічних засобів інформаційної системи в заданій предметній області.
Варіант 2	Визначити спосіб використання глобальної обчислювальної мережі і необхідний перелік послуг для інформаційної системи в заданій предметній області.

Таблиця 2.14

№ варіанту	Предметна область
1	Інформаційна система для автоматизації організаційно-розпорядливого документообігу виробничого підприємства.
2	Інформаційна система для автоматизації документообігу оперативного управління виробничого підприємства.
3	Інформаційна система для автоматизації документообігу підсистеми збуту виробничого підприємства.
4	Інформаційна система для організаційно-розпорядливого документообігу установи
5	Інформаційна система для факультету університету
6	Інформаційна система для кафедри університету
7	Інформаційна система для торговельного підприємства
8	Інформаційна система для авіапідприємства
9	Інформаційна система для лікувальної установи (лікарні)
10	Інформаційна система для лікувальної установи (поліклініки)
11	Інформаційна система для банку
12	Інформаційна система для культурно-спортивного центру
13	Інформаційна система для видавництва, що має філіали в інших містах
14	Інформаційна система для автотранспортного підприємства
15	Інформаційна система для підприємства зв'язку
16	Інформаційна система для залізничного вокзалу
17	Інформаційна система для школи (коледжу, гімназії)
18	Інформаційна система для виставкового центру
19	Інформаційна система для центру служби зайнятості
20	Інформаційна система для акціонерного суспільства, що має філіали в інших містах

21	Інформаційна система для невеликої фінансової компанії
22	Інформаційна система для невеликої інвестиційної фірми
23	Інформаційна система для архітектурної організації
24	Інформаційна система для машинобудівного підприємства

3.4. Правила оформлення пояснювальної записки

Пояснювальна записка пишеться чорнилом на одній стороні **листа** паперу формату А4. **Загальний об'єм** 25-30 сторінок. З лівого боку **листу** повинні бути поля шириною 20 мм, листи підшиваються в папку разом з діаграмами, схемами і **іншими** ілюстраціями. Всі схеми, формули, графіки повинні бути пронумеровані і забезпечені підписами і **посиланнями** в тексті. Допускається підготовка пояснювальної записки з використанням комп'ютера. Матеріали в пояснювальній записці слід розташовувати в **наступному порядку**:

- титульний **лист**;
- **завдання** на проектування;
- **зміст**;
- **вступ**;
- техніко-економічне обґрунтування розробки обчислювальної **мережі**;
- вибір конфігурації обчислювальної **мережі**;
- проектування структурної схеми обчислювальної **мережі**;
- теоретична та розрахункова **частини**;
- **висновок**;
- список літератури;
- **додатки**.

У пояснювальній записці повинні бути витримані єдині позначення і єдині розмірності для параметрів, які було використано. Допускаються **тільки** загальноприйняті скорочення слів, термінів, позначень. Закінчена пояснювальна записка підписується студентом і керівником проекту. Виклад повинен бути ясним і чітким, без повторень, кількість ілюстрацій-мінімальною, **але** достатньою для пояснення викладеного.

3.5. Правила оформлення графічного матеріалу

Графічна **частина** проекту **є** не ілюстративним матеріалом, а технічною документацією на розроблений студентом проект обчислювальної **мережі**. Графічний матеріал, поміщений в пояснювальній записці, а також на листах, по формату, умовним позначенням, шрифтам і масштабам повинен відповідати вимогам єдиної системи конструкторської документації (**ЕСКД**).

3.6. Методика курсового проектування

3.6.1. Загальні теоретичні відомості про обчислювальні мережі

Обчислювальна **мережа** складається з обчислювальних машин і **мережі** передачі даних (**мережі** зв'язку). **ОМ** класифікуються по геометричних масштабах на **наступні** класи **мереж**: глобальна обчислювальна **мережа**; широкомасштабна **мережа**; регіональна **мережа**; локальна **мережа**. Вимоги, що пред'являються до обчислювальної **мережі** в даній прикладній області, визначають географічні масштаби **ОМ** і швидкості передачі даних.

Під локальною обчислювальною **мережею** (**ЛОМ**) звичайно розуміють **ОМ**, які сполучають обчислювальні машини в одній кімнаті, будівлі або декілька **близько розташованих** будівлях. **Мережі** зв'язку **ЛОМ** мають **в даний час наступні** типові характеристики: високу швидкість передачі даних (0.1-100 Мбіт/с), невелику протяжність (0.1-50км), малу імовірність помилки передачі даних (+1E-8 - +1E-11).

Глобальні комп'ютерні **мережі** (**ГОМ**) включають комп'ютери і канали зв'язку, які забезпечують **можливість** передачі інформації від комп'ютера до комп'ютера. **ГОМ**, наприклад, **мережа мереж** – Інтернет охоплює мільйони комп'ютерів практично у всіх країнах світу. Послугами **ГОМ** користуються десятки мільйонів чоловік-бізнесменів, **вчених, урядовців, викладачів, студентів**. Основний ефект від **використання ГОМ** для кінцевого користувача полягає в скороченні часу обміну або доступу до інформації при зниженні **витрат**. **ГОМ** являють собою специфічні підприємства по виробництву (наданню) інформаційного сервісу: електронна пошта, телеконференції, новини, біржові зведення, передача файлів, доступ до **мережевих архівів** і баз даних і т.д.

Міжнародна організація по стандартизації (International Standards Organization – ISO) розробила модель архітектури **мережі**, яка називається взаємодія **відкритих** систем – **ВВС** (Open System Interconnection reference model – OSI). Модель **ВВС** – це багаторівнева модель, яка за умови її **використання** всіма постачальниками ЕОМ і **мереж**, забезпечить зв'язок будь-якої ЕОМ з будь-якою **іншою** в будь-якій **мережі**. У моделі взаємодії **відкритих** систем **є** сім рівнів управління **мережею** і відповідних протоколів. Протоколи – це набір правил і процедур, які регулюють **порядок** здійснення деякого зв'язку. **Мережеві** протоколи – це правила і технічні процедури, що дозволяють декільком комп'ютерам при об'єднанні в **мережу** спілкуватися один з одним [1]. Основні поняття, що стосуються протоколів:

- існує безліч протоколів, і хоча вони всі беруть участь в реалізації зв'язку, кожен протокол має різні цілі, виконує різні задачі, володіє своїми перевагами і обмеженнями;
- протоколи працюють на різних рівнях OSI; функція протоколу визначається рівнем, на якому він працює;

– декілька протоколів можуть працювати спільно; це так званий стек, або набір, протоколів.

Передача даних по **мережі**, з технічної точки зору, повинна бути розбита на **ряд** послідовних кроків, кожному з яких відповідають свої правила і процедури, або протокол. Ці дії повинні бути виконані в одній і тій же послідовності на кожному комп'ютері. На комп'ютері-відправнику ці дії виконуються в **напрямі** зверху вниз, а на комп'ютері-одержувачі – з низу в верх. Процес, який називається прив'язка, дозволяє з достатньою гнучкістю **настроювати мережу**, тобто **поеднувати** протоколи і плати **мережевих** адаптерів, як того вимагає ситуація. Наприклад, два стеки протоколів, IPX/SPX і TCP/IP (Transmission Control Protocol / Internet Protocol), можуть бути прив'язані до однієї плати **мережевого** адаптера. **Порядок** прив'язки визначає черговість, з якою операційна система виконує протоколи.

У комп'ютерній промисловості в якості стандартних моделей протоколів розроблено декілька стеків. **Найважливіші** з них: набір протоколів ISO/OSI; IBM System Network Architecture (SNA); Novell Netware; набір протоколів Інтернету, TCP/IP.

Модель OSI допомагає визначити, які протоколи потрібно використовувати на кожному рівні. ISO, IEEE (Institute of Electrical and Electronical Engineers), ANSI (American National Standards Institute), CCITT (Comite Consultatif Internationale de Telegraphie et Telephonie) і **інші** організації по стандартизації розробили протоколи, які відповідають деяким рівням моделі OSI. Так, наприклад, IEEE протоколи канального рівня підтримують зв'язок на підрівні управління доступом до **середовища**: 802.3 (Ethernet), 802.4 (передача маркера), 802.5 (Token Ring), 802.12 (100VG – AnyLAN). Протокол управління доступом до **середовища** визначає, якій саме комп'ютер може використовувати **мережевий** кабель, якщо декілька комп'ютерів одночасно **намагаються** дістати до нього доступ.

До основних методів доступу в сучасних **ЛОМ** відносяться:

- метод **множинного** доступу з контролем несучої і виявленням конфліктів (CSMA/CD, протокол 802.3) в **мережах** Ethernet;
- метод доступу з використанням маркера в **мережах** Token-Ring(протокол 802.5);
- доступ по пріоритету запиту, розроблений для стандарту **мережі** Ethernet із швидкістю передачі даних 100 Мбіт/с-100VG–AnyLAN топології зірка-шина.

Глобальна **мережа** Інтернет **спирається** на сімейство протоколів TCP/IP – промисловий стандартний набір протоколів, які забезпечують зв'язок в гетерогенному (неоднорідному) **середовищі**, **тобто** забезпечують сумісність між комп'ютерами різних типів. Як **видно з табл. 2.15**, TCP і IP приблизно відповідають транспортному і **мережевому** рівням моделі OSI.

Таблиця 2.15

Прикладний	Простий протокол пересилки пошти (SMTP)
Представлення даних	Протокол передачі файлів (FTP)
Сеансовий	Програма емуляції терміналу (Telnet)
Транспортний	TCP, UDP
Мережевий	IP, ARP
Канальний	Ethernet

Першим кроком до **відкриття** Інтернет для масового користувача була поява технології WorldWide Web – WWW (всесвітня павутина). Web-сервери і Web-навігатори використовуються **не тільки** у світовому масштабі, у Всесвітній Павутині. Web – це інфраструктурний сервіс, необхідний кожній організації з **скільки-небудь** помітними інформаційними потоками.

Наступним етапом в розвитку технологій, які були розроблені в Інтернет, в **першу чергу** Web-технології, став їх **додаток** до корпоративних інформаційних систем. Такі системи **одержали** назву системи Інтранет. Таким чином, Інтранет – це **застосування** технології Інтернет в **рамках** корпоративних систем.

Застосування технології Інтранет для **побудови** корпоративних інформаційних систем, **перш за все** для зручного, призначеного для користувача інтерфейсу **видаленого** доступу, забезпечує **можливість** підвищення ефективності управління діловими процесами.

Упровадження технології Інтранет дає відчутний економічний ефект в діяльності організації. Зміни, пов'язані в **першу чергу** з різким **поліпшенням** якості **споживання** інформації, **напряму** впливають на продуктивність праці співробітників організації. Результат **застосування** Інтранет – різке скорочення паперових архівів, легкість і простота публікації інформації, універсальний і природний доступ до інформації за допомогою навігаторів, **істотне** скорочення **витрат** на адміністрування **додатків** на робочих місцях користувачів, негайна актуалізація будь-яких змін в інформаційному сховищі організації, **зсув** акцентів від **створення** інформації до її ефективного **споживання**. **Ключовими** якостями Інтранет, **напряму** пов'язаними з економічними аспектами діяльності сучасної організації є:

- простота і природність технології;
- низький ризик і швидка віддача інвестицій;

- інтеграційний і каталітичний **характер** технології;
 - ефективне управління і комунікації в організації.
- Схема корпоративної обчислювальної **мережі** показана на рис.2.55.

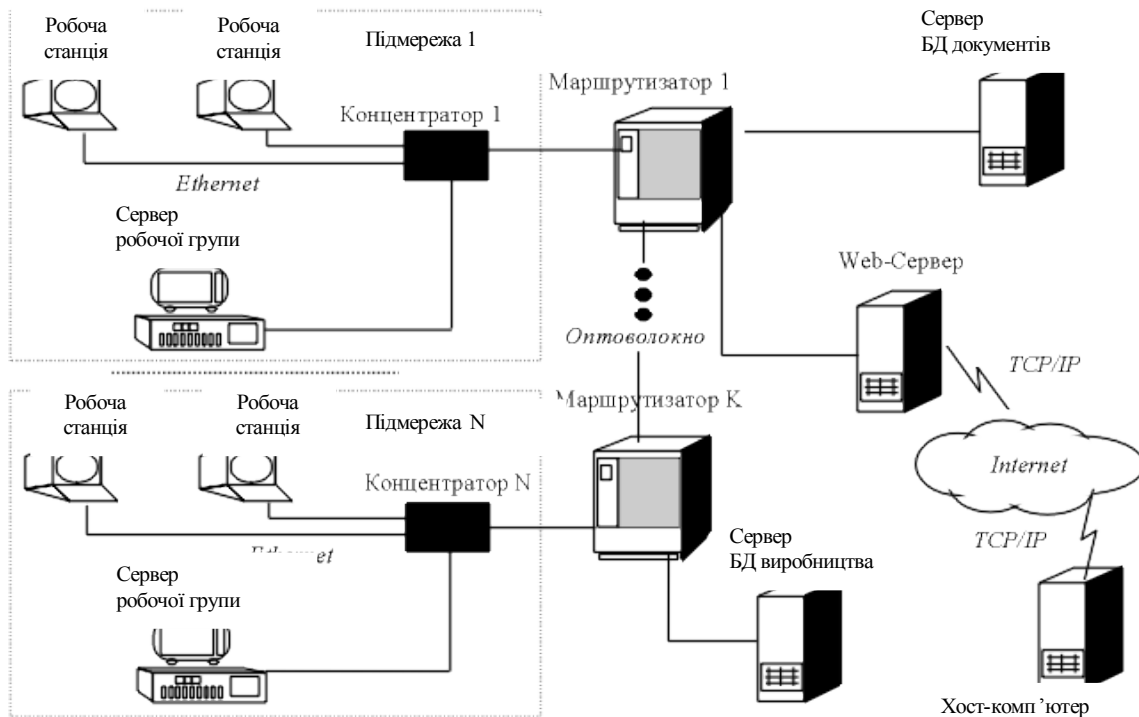


Рис.2.55. Схема корпоративної обчислювальної **мережі**.

Відповідно до двох основних компонентів корпоративної обчислювальної **мережі** Інтранет для курсового проектування вибрані два основні варіанти – проектування локальної обчислювальної **мережі** і проектування апаратного і програмного забезпечення для **використання** послуг глобальної обчислювальної **мережі** в інформаційних системах різних підприємств і організацій.

3.6.2. Проектування локальної обчислювальної **мережі**

При **створенні ЛОМ** перед розробником стоїть проблема: при відомих даних про призначення, перелік функцій **ЛОМ** і основні вимоги до комплексу технічних і програмних засобів **ЛОМ**, **побудувати мережу** для інформаційної системи в заданій **предметній** області. Методика проектування локальних обчислювальних **мереж** складається з етапів, показаних на рис. 2.56.

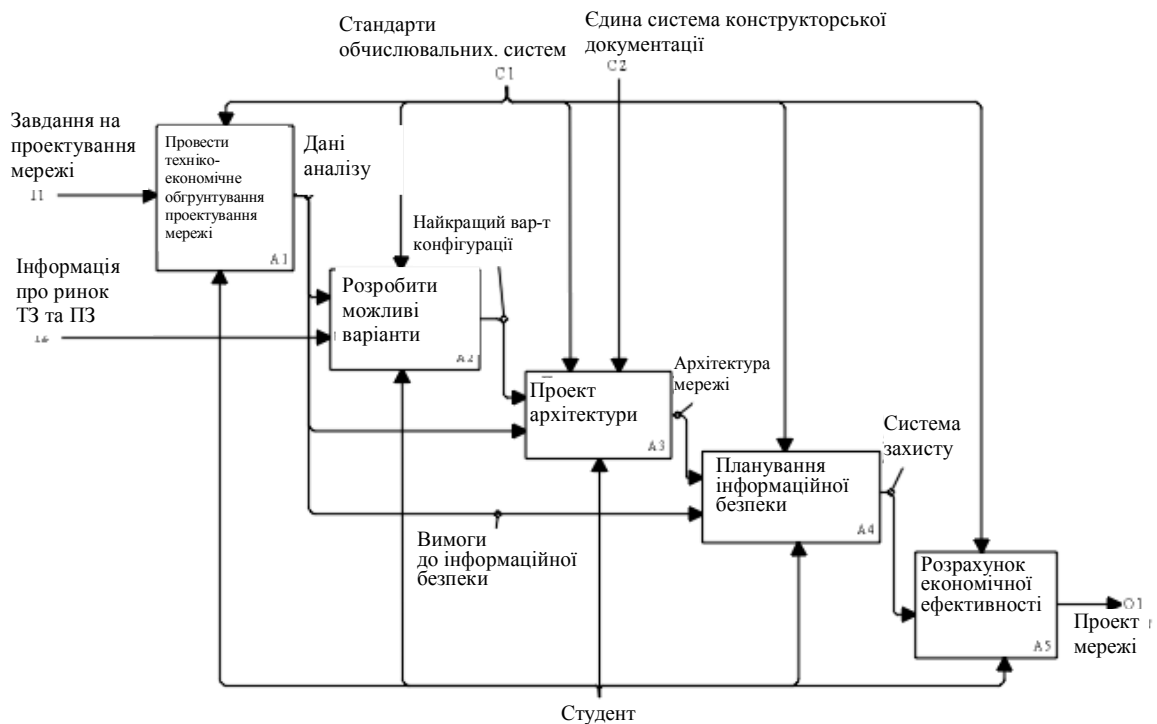


Рис. 2.56. Етапи проектування **ОМ**.

Короткий зміст етапів проектування

Техніко-економічне обґрунтування розробки обчислювальної мережі (рис.2.57) включає аналіз предметної області і обґрунтування потреби проектування обчислювальної мережі. На цьому етапі необхідно визначити цілі створення мережі. Основними цілями звичайно є:

- зв'язок;
- сумісна обробка інформації;
- сумісне використання файлів;
- централізоване управління комп'ютерами;
- контроль за доступом до важливих даних;
- централізоване резервне копіювання всіх даних;
- децентралізація обробки інформації на відміну від централізованої обробки на мейнфреймі і ін.

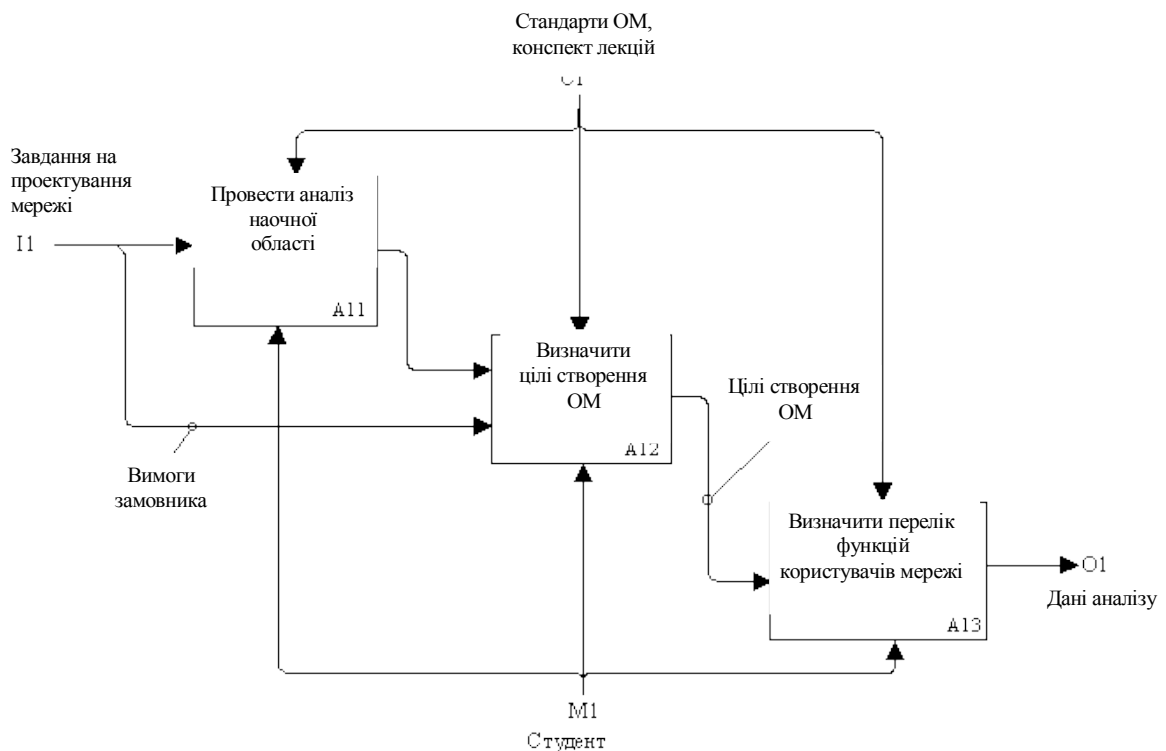


Рис. 2.57. Етапи техніко-економічного обґрунтування проектування ОМ.

Наступним кроком аналізу предметної області є розробка функціональної, інформаційної і динамічної моделей предметної області по IDEF – технології або визначення переліку функцій користувачів в обчислювальній мережі і розробка моделі "сутність-зв'язок" (ER- моделі).

Розробка можливих варіантів конфігурації ЛОМ

Мережеву архітектуру можна розуміти як підтримуючу конструкцію або інфраструктуру, яка лежить в основі функціонування мережі. Така інфраструктура складається з декількох головних складових, зокрема компонови або топології мережі, кабельної проводки, пристрої: мости, маршрутизатори і комутатори, які сполучають елементи мережі. Проектуючи мережу, необхідно брати до уваги кожний з цих мережевих ресурсів і визначити, які конкретно засоби слід вибрати і як їх треба розподілити по мережі, щоб оптимізувати продуктивність, спростити управління устаткуванням і залишити можливості для подальшого зростання.

Розробку можливого варіанту конфігурації ЛОМ слід починати із створення загального плану мережі, оформленого як таблиця конфігурації. Представлена в таблиці 4 конфігурація широко поширена при установці мереж, в яких кількість користувачів не перевищує 50 чоловік. Ця стандартна конфігурація підходить для більшості випадків проектування мереж і проста в реалізації. У курсовому проекті слід створити свою конфігурацію мережі відповідно до конкретного завдання (табл. 2.16).

Таблиця 2.16

Компонент/характеристика	Реалізація
Топологія	Зірка-шина
Лінія зв'язку	Неекранована вита пара категорії 5
Мережеві адаптери	Ethernet 10BaseT
Ретранслятори (повторювачі, концентратори, комутатори, мости, маршрутизатори, шлюзи)	Концентратор Ethernet 10BaseT
Управління сумісним використанням ресурсів	Мережа на основі серверу з комп'ютерами-клієнтами. Ресурси, які вимагають централізованого управління, знаходяться на сервері, а інші – на комп'ютерах-клієнтах.
Сумісне використання периферійних пристроїв	Підключення мережевого принтера безпосередньо до мережевого кабелю через мережеву плату, управління чергами до принтера за допомогою програмного забезпечення серверу
Підтримувані програмне забезпечення:	Електронна пошта, обробка повідомлень факсиміле, організація колективних робіт в середовищі електронного документообігу, робота з базами даних з використанням спеціалізованих серверів

При розробці можливих варіантів конфігурації **мережі** слід пам'ятати, що вони повинні бути складені на основі вимог, сформульованих для виконання функцій інформаційної системи в заданій **предметній** області, і повинні задовольняти умові сумісності апаратних і програмних засобів. Для аналізу варіантів складається таблиця, аналогічна таблиці 2.16, в якій наводяться характеристики 2-3 варіантів конфігурації локальної обчислювальної **мережі**.

Оцінка різних варіантів архітектури **ОМ** проводиться з системних позицій по основних критеріях: швидкодія, надійність, інформаційна безпека, вартість. Далі виконується вибір **якнайкращого** варіанту по основних критеріях. При цьому залежно від **встановлених** цілей проектування обчислювальної **мережі** вибирається один головний критерій ефективності досягнення мети, а решта критеріїв враховується як обмеження. Наприклад, можлива **наступна** постановка задачі оптимізації: забезпечити мінімальну затримку передачі **повідомлень** в **мережі** при виконанні **встановлених** обмежень на значення показників надійності **мережі**, вартості **мережі** і достатню якісну оцінку інформаційної безпеки **мережі**.

Для вирішення поставленої задачі можуть використовуватися відомі методи рішення задачі вибору. Такими методами є: метод аналізу ієрархій Сааті [2], метод зважування, метод гілок і **меж**. При рішенні задачі вибору в якості початкових даних можуть використовуватися як числові характеристики, так і якісні експертні оцінки. Можна якісно визначити **ступінь** переваг користувача при оцінці якості і важливості послуг **ОМ**, використовуючи для цього лінгвістичні змінні, наведені в таблиці 2.17.

Таблиця 2.17

Лінгвістична змінна	Опис лінгвістичної змінної
Відмінно	необхідні функції виконуються бездоганно
Дуже добре	відповідає всім основним вимогам і забезпечує істотні переваги
Добре	відповідає основним вимогам і має додаткову нагоду
Задовільно	відповідає основним вимогам
Погано	має обмежену нагоду
Неприйнятно	не відповідає мінімальним вимогам або не забезпечує даної можливості

Розробка структурної схеми **ЛОМ**

На даному етапі для вибраного варіанту конфігурації **ЛОМ** необхідно:

- розробити архітектуру **ЛОМ**;
- розробити структурну схему **ЛОМ**;
- вибрати типи компонент **ЛОМ**;
- розрахувати кількість компонент **ЛОМ**;
- скласти специфікацію **ЛОМ**.

При цьому повинні враховуватися правила **з'єднання** компонентів **ЛОМ**, засновані на стандартизації **мереж** і їх обмеження, специфіковані виготівниками компонент **ЛОМ**. Етапи проектування архітектури **ЛОМ** **представлені на рис.2.58**. Розробці структурних схем **ЛОМ** і вибору типів компонент **ЛОМ** присвячені роботи [3].

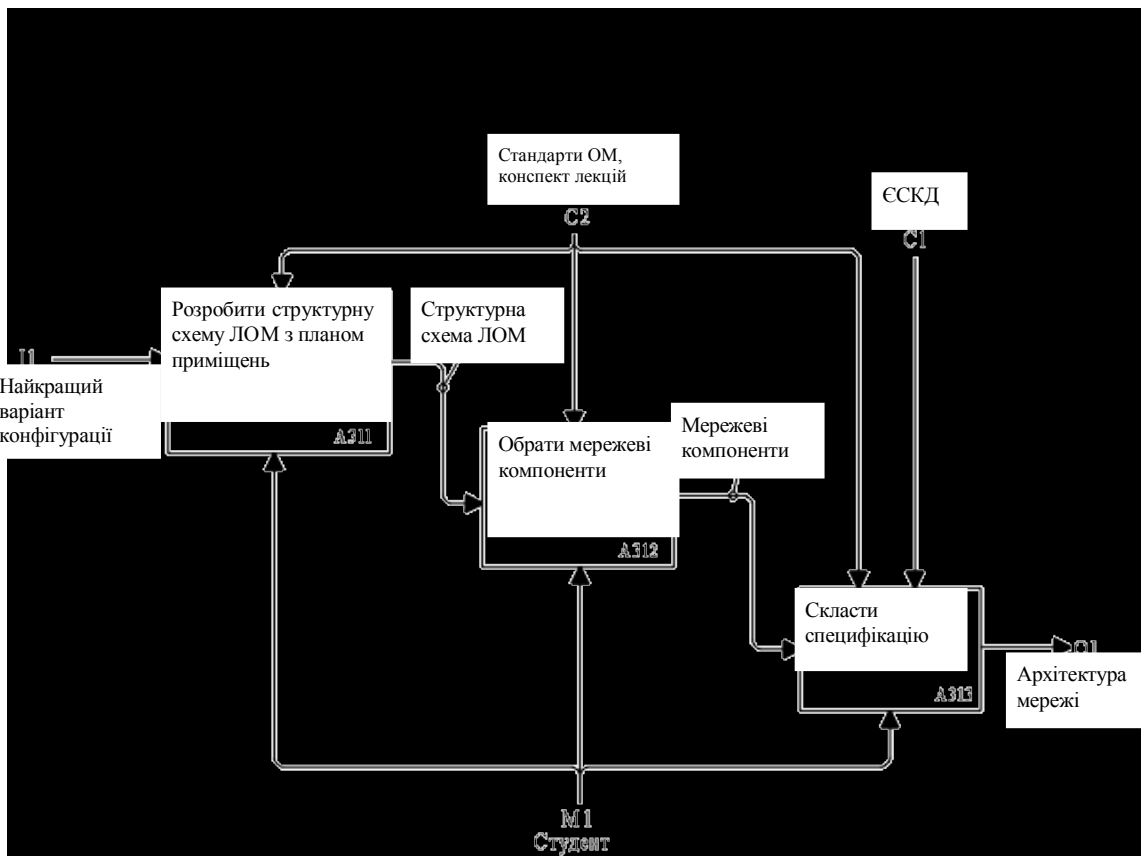


Рис. 2.58. Етапи проектування архітектури ЛОМ.

Приклад структурної схеми ЛОМ для інформаційної системи кафедри університету наведено на мал. (мал. ЛОМ кафедри АСУ, новий). Специфікація ЛОМ, конфігурація якої приведена в таблиці 2.16, показана в табл. 2.18.

Технічні засоби (ТЗ) обчислювальної мережі

Таблиця 2.18

N	Тип ТЗ	Найменування ТЗ
1	Сервер	Комп'ютер Pentium III-IV
2	Мережевий адаптер	Ethernet
3	Лінія зв'язку	Кабель коаксіальний RG 58 або вита пара (вибрати категорію з використанням джерела [5]).
4	Активні концентратори	HUB 8-port 10Mbps
5	Роз'єми	BNC connector
6	Роз'єми	T-connector
7	Мережевий принтер	
Програмне забезпечення		
8	Мережева операційна система	MS Windows NT 4/2000/2003 Server
9	Інтегрована офісна система	MS OFFICE'97/2k/XP RUS

Планування інформаційної безпеки

Захист інформації включає комплекс заходів, **направлених** на забезпечення інформаційної безпеки. На практиці під цим розуміється підтримка цілісності, доступності і, якщо потрібно, конфіденційності інформації і ресурсів, які використовуються для введення, зберігання, обробки і передачі даних.

Інформаційна безпека – це захищеність інформації і підтримуючої інфраструктури від випадкових або **навмисних** дій природного або штучного **характеру**, через які трапляється нанесення **збитку** власникам або користувачам інформації і підтримуючій інфраструктури.

Основний критерій для вибору рівня захисту – важливість інформації. Якщо в обчислювальній **мережі, яка проектується**, оброблятимуться конфіденційні дані, слід вибрати централізований захист на основі серверу (незалежно від кількості обслуговуваних користувачів). Етапи **планування** інформаційної безпеки представлено на рис.2.59.

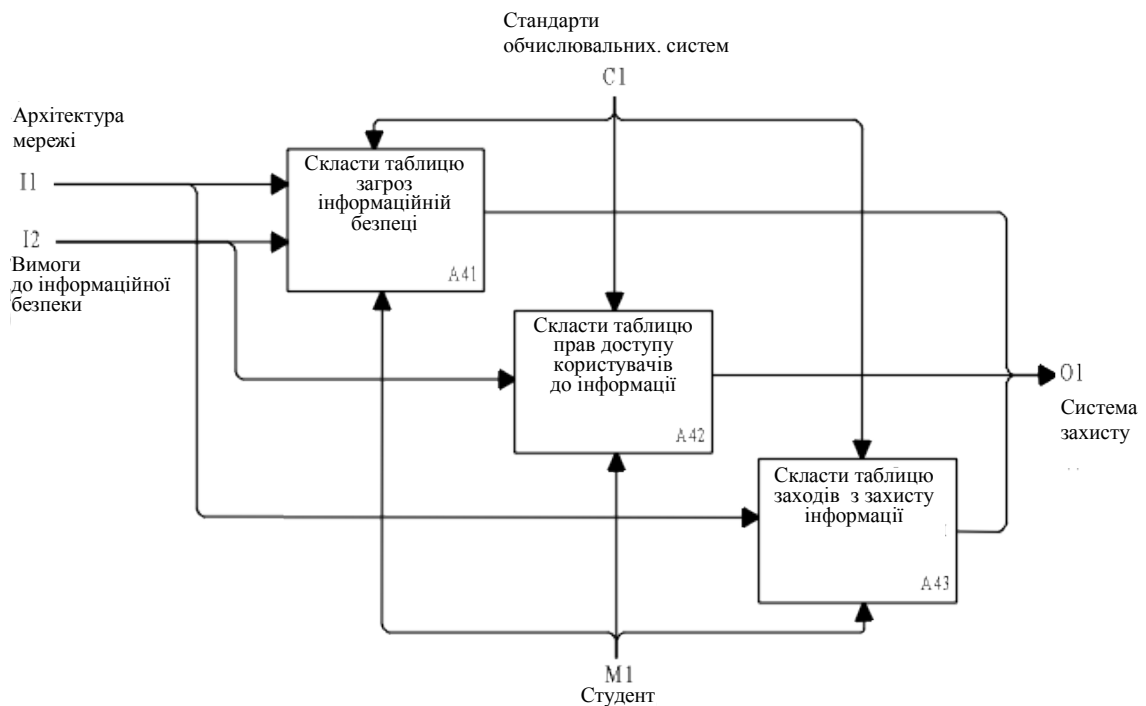


Рис. 2.59. Етапи планування інформаційної безпеки.

Всім захисним заходам повинен передувати аналіз загроз. До загроз можна віднести все, що спричиняє за собою втрату даних в мережі, зокрема:

- крадіжка або вандалізм;
- пожежа;
- відмови джерел живлення і скачки напруги;
- відмови компонентів;
- природні явища (блискавка, повені, бурі і землетруси).

Існують методи і системи, що запобігають катастрофічній втраті даних:

- резервне копіювання;
- джерела безперебійного живлення;
- відмовостійкі системи;
- попередження крадіжки даних;
- паролі і шифрування;
- аудит;
- комп'ютери без дисків;
- навчання користувачів;
- фізичний захист устаткування;
- захист від вірусів.

Аудит – це запис певних подій в журнал безпеки серверу. У проєкті потрібно вибрати методи і системи для запобігання втраті даних, відповідні переліку найреальніших загроз безпеки в заданій предметній області, які приводять до найважчих наслідків для обчислювальної мережі.

Необхідною функцією засобів забезпечення безпеки є реєстрація діяльності користувачів. Для кожної бази даних, окремого документа і навіть окремого поля запису у файлі бази даних можуть бути встановлені:

- список користувачів, які мають право доступу;
- функції, які може виконувати користувач;
- привілеї для доступу до вибраної інформації.

У проєктованій обчислювальній мережі потрібно вибірково наділити користувачів правами доступу до каталогів і створити групи для надання доступу до загальних мережевих ресурсів. Приклад визначення прав доступу для груп користувачів показаний в табл. 2.19.

Таблиця 2.19

Назва групи	Внутрішні ресурси	Рівні доступу до внутрішніх ресурсів	Доступ в Internet і електронна пошта
Адміністратори	Всі мережеві ресурси	Права адміністратора в каталогах, зокрема зміна рівня і прав доступу	Всі мережеві ресурси
Розробники	Бази даних документів, що розробляються	Створення, читання файлів, запис у файл, створення підкаталогів і файлів, видалення каталогів, пошук файлів, зміна каталогів	Всі мережеві ресурси
Співробітники в офісі	Вся інформація підприємства (установи)	Обмеження доступу (з потреби)	Обмеження за IP- адресою (адресата і джерела), обмеження за змістом (вхідної і витікаючої кореспонденції)
Співробітники поза офісом	Вся інформація підприємства (установи)	Обмеження доступу (з потреби)	Обмеження за IP- адресою (адресата і джерела), обмеження за змістом (вхідної і кореспонденції, яка виходить), аутентифікація видаленого користувача перед здійсненням доступу
Постачальники, ділові партнери, клієнти	Спеціальні каталоги для виробників, партнерів і клієнтів	Доступ тільки до спеціально відведених областей	Обмеження за IP- адресою (адресата і джерела). Ідентифікація і аутентифікація видаленого користувача
Потенційні клієнти	Спеціальні каталоги для клієнтів	Проглядання об'єктів (читання і пошук файлів)	При відкритому доступі Інтрасеть повинна бути ізольована; ідентифікація користувача не потрібна

3.6.3. Проектування апаратного і програмного забезпечення для використання глобальних обчислювальних мереж

Короткий зміст етапів проектування

Зміст 1, 4 і 5 етапів проектування відповідає змісту аналогічних етапів проектування локальних обчислювальних мереж.

Вибір послуг, які надаються глобальною обчислювальною мережею

Вибір послуг, які надаються глобальною обчислювальною мережею, для виконання необхідних функцій, представлено на рис. 2.60.

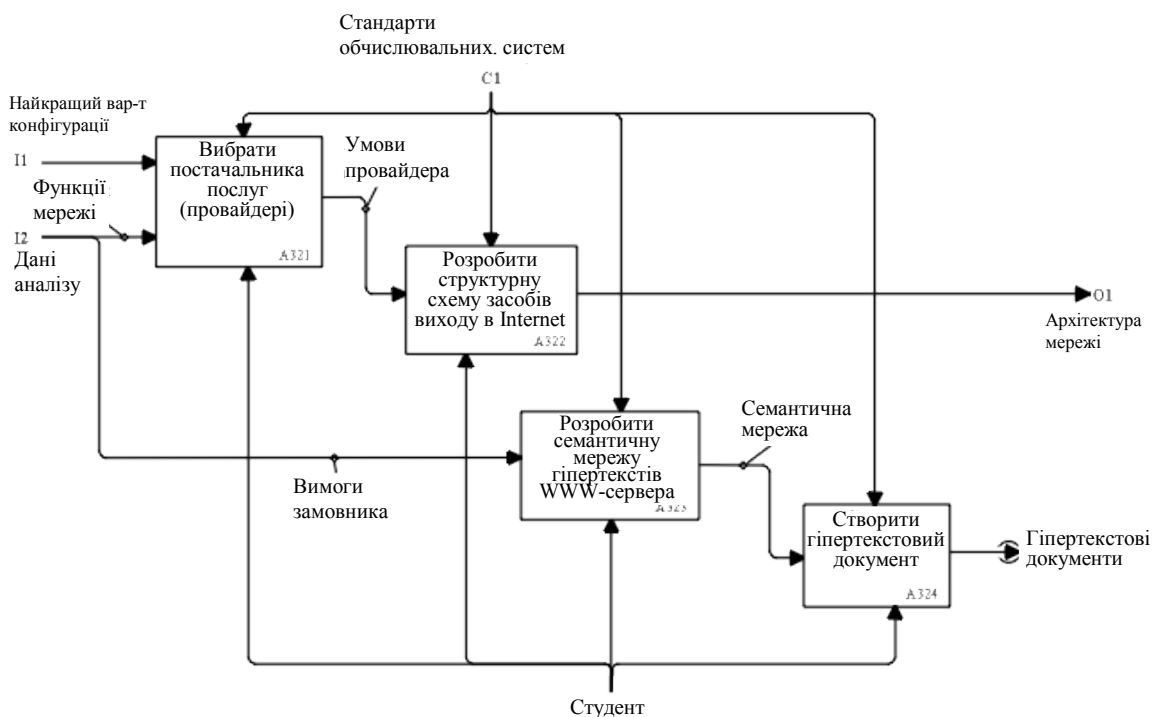


Рис. 2.60. Вибір послуг, що надаються ГОМ.

Глобальні **мережі** є специфічними підприємствами по виробництву (наданню) інформаційного сервісу: електронна пошта, телеконференції, новини, біржові зведення, доступ до **мережових** архівів і баз даних і т.д. Глобальні **мережі** можуть бути орієнтовані на широкий спектр послуг для широкого кола користувачів (загальнодоступні глобальні **мережі**) і можуть бути спеціалізованими (наприклад, глобальні **мережі** банківських розрахунків) і корпоративними (наприклад, **мережа** міжнародної корпорації DEC).

Найважливішими комп'ютерними **мережами**, які діють або в окремих рамках, або утворюють спільно з Internet світовий **мережовий** простір, є **мережі** ARPANET, NSFNET, BITNET, EARN, USENET, EUnet, NASA Science Internet, FidoNet, Compu Serve, MCI, Mail, GLASNET і багато інших. Щоб підключитися до серверів Internet, ваш комп'ютер повинен бути підключений до цієї глобальної **мережі** через постачальника послуг Internet (рис. 2.61).

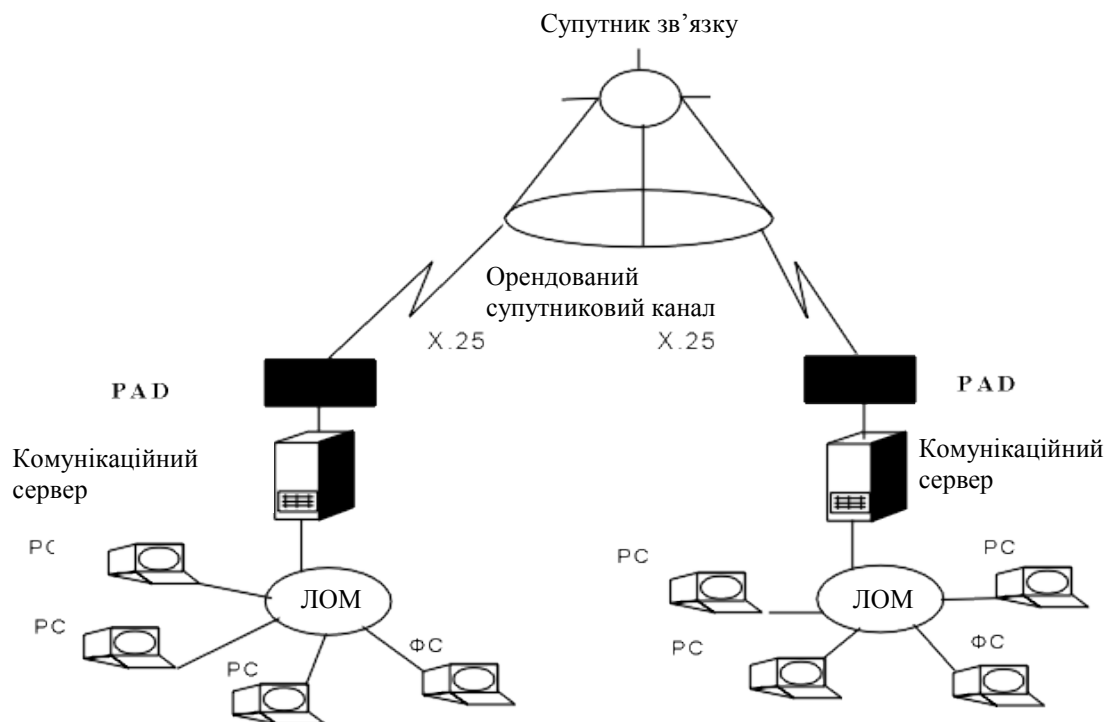


Рис. 2.61. Схема ГОМ із засобами супутникового зв'язку.

Для підключення комп'ютера до Internet потрібно щонайменше підтримка протоколу TCP/IP, конфігурованого так, щоб комп'ютер був вузлом Internet. Іншими словами, комп'ютер повинен мати Internet-адресу, досягну для передбачуваних користувачів серверу.

Для вибору постачальника послуг обчислювальних мереж потрібно скласти таблицю, розмістивши в ній характеристики трьох-чотирьох постачальників послуг обчислювальних мереж в Києві. Ці характеристики є початковими даними для вибору постачальника глобальних обчислювальних мереж по одному з відомих методів рішення задачі вибору, аналогічно задачі вибору конфігурації локальної обчислювальної мережі в п. 1.

Розробка структури програмного забезпечення WWW- серверу

WWW – інформаційна гіпертекстова система, що забезпечує доступ до документів з вказівкою шляху пересування від документа до документа в межах кінцевої їх множини. WWW функціонує за принципом "клієнт-сервер". Web-сервер виступає як інформаційний концентратор, який одержує інформацію з різних джерел, а потім однорідним чином надає її користувачу. WWW клієнти існують для різних типів ЕОМ і операційних систем і вільно розповсюджуються в Internet. Програма для проглядання гіпертекстів – навігатор, який забезпечений універсальним і природним інтерфейсом для спілкування з людиною, дозволяє останньому легко проглядати інформацію незалежно від її природи. Інтеграційні якості Web-технології виявляються виключно важливими для застосування в корпоративних мережах. Архітектура клієнт/сервер з використанням Web-технології представлена на рис. 2.62.

Web-технологія підходить для підтримки інфраструктури комунікацій і координації в середовищі розподіленого ухвалення рішень. Web-стандарти прості, чітко організовані і широко використовуються у всьому світі. Програмне забезпечення для ведення бази даних WWW складається з:

- програм-клієнтів протоколу HTTP;
- програми-серверу протоколу HTTP;
- CGI-скриптів.

З цього набору 80% складає готове програмне забезпечення, яке використовується в мережі Internet, і лише 20% (CGI-скрипти) розробляється користувачами і адміністраторами.

З погляду проектувальника WWW-сервер – це просто програма, яка відповідає на запит документа від WWW-клієнта через Internet. Всі WWW-сервери використовують загальний протокол для спілкування з клієнтами, який одержав назву HyperText Transfer Protocol (HTTP) – протокол передачі гіпертексту. Це протокол прикладного рівня, розроблений для обміну гіпертекстовою інформацією в мережі Internet. Використовуючи цей протокол, можна передавати всі типи даних, включаючи HTML-документи, графіку,

звук і відео. HTTP дозволяє реалізувати в рамках обміну даними набір методів доступу, які базуються на специфікації універсального ідентифікатора ресурсів (Universal Resource Identifier), вживаного у формі універсального локатора ресурсів (Universal Resource Locator – URL) або універсального імені ресурсу (Universal Resource Name). Повідомлення по мережі при використуванні протоколу HTTP передаються у форматі, подібному до формату поштового повідомлення Internet (RFC-822) або формату повідомлень MIME (Multiperposal Internet Mail Exchange). HTTP використовується для взаємодії програм-клієнтів з програмами-шлюзами, які надають доступ до ресурсів електронної пошти Internet (SMTP), списків новин (NNTP), файловим архівам (FTP) системам GOPFER і WAIS.

Протокол розроблено для доступу до цих ресурсів за допомогою проміжних програм-серверів (проху), які дозволяють передавати інформацію між різними інформаційними службами без втрат. Функції WWW-серверу, окрім його прямого обов'язку передавати потрібні дані, можна умовно розбити на чотири групи:

- а) управління передачею документів;
- б) ведення журналу активності клієнтів;
- в) підтримка безпеки даних;
- г) забезпечення роботи засобів інтерактивної роботи з клієнтом.

За процес передачі інформації відповідає HTTP-протокол. І від серверу залежить, наскільки повно він підтримує стандарт передачі даних, а також розширення, що допомагають в роботі, але не узаконені як стандарт. Сюди ж відноситься взаємодія серверу з шлюзами, проху-серверами і брандмауерами, які використовуються для захисту локальних мереж від несанкціонованого доступу ззовні. WWW-сервер може вести журнал активності клієнтів, записуючи Internet-адреси, час, зміст запиту і інші параметри кожного з'єднання.

Функції захисту мають декілька рівнів:

- а) контроль доступу до файлів на сервері;
- б) обмеження на допустимі клієнтські Internet-адреси і імена доменів; в) кодування мережевого трафіку (технологія SSL).

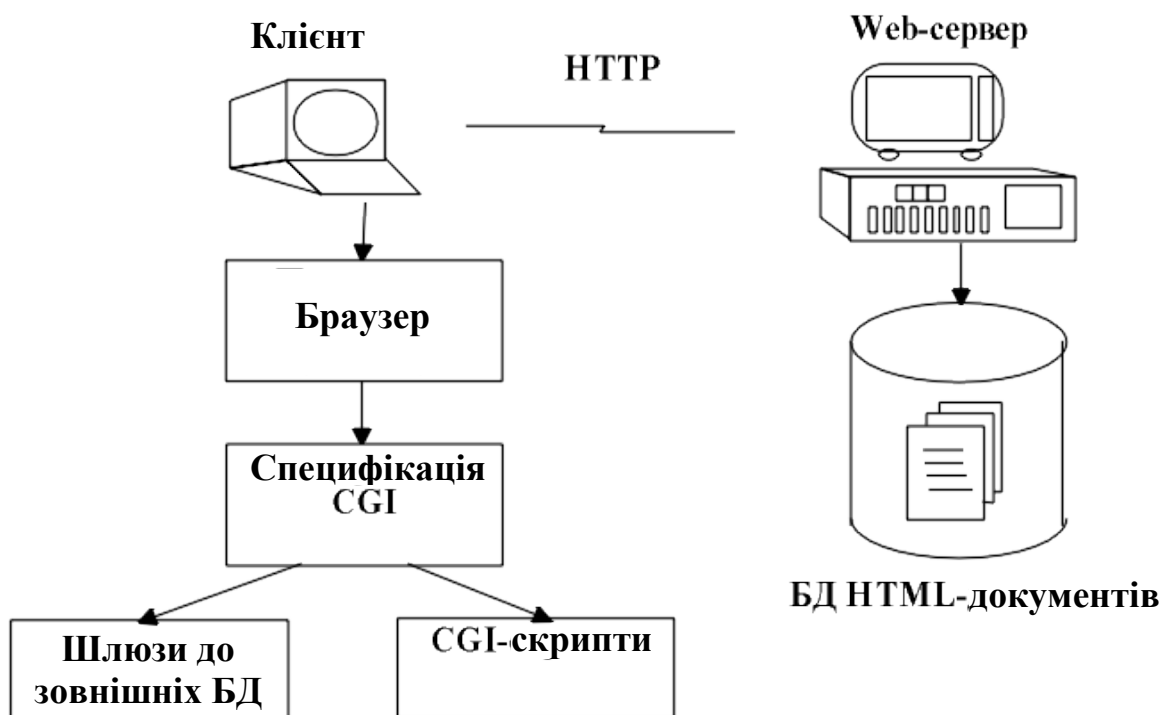


Рис. 2.62. Архітектура клієнт/сервер з використанням Web-технології.

CGI (Common Gateway Interface) – забезпечення однакового пакету даних між сервером і прикладною програмою, яка запускається під керуванням серверу. CGI-скрипт – програма, яка написана відповідно до специфікації Common Gateway Interface, і яку сервер HTTP може запускати для реалізації механізмів, не передбачених в протоколі. Багато достатньо могутніх інформаційних механізмів WWW реалізовано за допомогою CGI-скриптів. До них відносяться: програми пошуку за ключовими словами, програми реалізації графічних гіпертекстових посилань – imagermap, програми сполучення з системами управління базами даних

і т.п. CGI-скрипти можуть бути написані на будь-якій мові програмування (наприклад, C, C++, PASCAL) або мові команд.

Шлюз – це CGI-скрипт, який використовується для обміну даними з іншими інформаційними ресурсами Internet або програмами-доменами. Звична CGI-програма запускається сервером HTTP для виконання деякої роботи, повертає результати до серверу і завершує своє виконання. Шлюз виконується так само, тільки фактично він ініціює взаємодію в якості клієнта з третьою програмою. Якщо ця третя програма є сервісом Internet, наприклад, сервер GOPHER, то шлюз стає клієнтом GOPHER, який посилає запит по порту GOPHER, а після отримання відповіді пересилає його серверу WWW. Аналогічно відбувається взаємодія з серверами розподілених баз даних, наприклад, ORACLE.

Найпоширенішими програмами-клієнтами WWW є Netscape Navigator, Internet Explorer. Вони забезпечують графічний інтерфейс доступу до WWW. Програма інтерпретує мову гіпертекстової розмітки HTML і дозволяє обмінюватися даними по протоколу HTTP. При використуванні гіпертекстових посилань перехід від одного документа до іншого відбувається після вибору "мишею" відповідної переходу фрази або слова, яке виділене кольором в тексті і є ідентифікатором посилання. При використанні графічного стека гіпертекстових посилань користувач вибирає крапку на графічному образі, координати якої і передаються програмі-серверу.

Програма-сервер WWW – це програма, яка приймає запити від WWW-клієнтів і відповідає на них. Як відповідь може бути повернений HTML-документ, який зберігається в базі даних серверу, графічний образ, аудіозапис, фільм або відповідь зовнішньої програми. Сервер обмінюється даними не тільки з клієнтами, але і з CGI-скриптами. В даний час сервери WWW існують для всіх типів комп'ютерних платформ і операційних систем.

До створення власного WWW-серверу необхідно його спланувати. Термін "сервер" використовується для позначення як програмного забезпечення, що оперує даними, так і комп'ютера, до якого звертається користувач за цими даними. Тому при плануванні необхідно враховувати обидва аспекти при виборі апаратного і програмного забезпечення, яке задовольняє потребам виконання функцій інформаційної системи, що розробляється.

Вибір апаратного забезпечення WWW-серверу

Переважає більшість доступних комп'ютерних платформ використовується для побудови на їх базі невеликого серверу на 8-10 одночасних з'єднань з користувачами. Навіть для простих серверів на базі PC або Macintosh найістотнішим є об'єм оперативної пам'яті. Бажано мати принаймні 16 Мбайт для серверу, не рахуючи потреб мережевої операційної системи. Значно менше число платформ підходять для побудови могутнього серверу. Кластери таких комп'ютерів можуть функціонувати не менш ефективно, особливо якщо ретельно спланувати розподіл документів між ними. Можна також продублювати важливу інформацію на декількох серверах, а у разі раптового виходу з ладу одного з них змінити посилання в документах. Схема організації Web-вузла в мережі Novell Netware представлена на рис.2.63.

Якщо необхідний крупний інформаційний сервер (для університету, бібліотеки, архіву і т. п.), рекомендується використовувати одну або декілька робочих станцій Unix, наприклад, робочі станції таких постачальників, як Sun Microsystems (Sun Solaris), Hewlett-Packard (HPUX), IBM (AIX) або Digital Equipment Corporation (OSF).

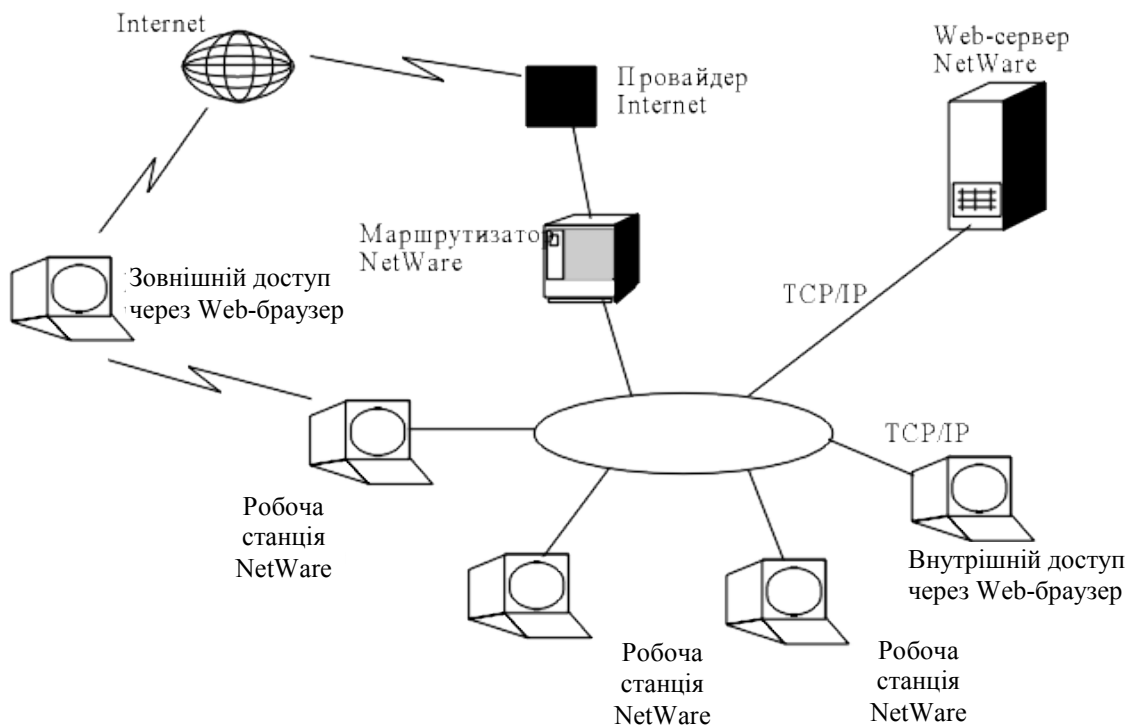


Рис. 2.63. Схема організації Web-вузла в мережі Novell Netware.

Існує декілька типів програмного забезпечення Web-серверів. Воно буває як безкоштовним, так і комерційним. В ОС UNIX є 4 різновиди Web-серверів: сервер NCSA, сервер Apache, сервер CERN, сімейство серверів Netscape. Для Windows 95 Windows NT розроблений сервер WebSite. Конфігурація декількох найпопулярніших серверів освітлюється в [4].

Хоча у принципі Web-документи можуть існувати в будь-якому форматі, як стандарт прийнято мову розмітки гіпертексту HyperText Markup Language (HTML) – мову, призначену для створення форматowanego тексту, який насичений зображеннями, звуком, анімацією і гіпертекстовими посиланнями на інші документи, які розкидані по всьому Web-простору.

Якщо статичних документів для задоволення потреб Web-вузла недостатньо, на вузлі використовуються такі інструментальні засоби, як CGI, Java і JavaScript. Java – це об'єктно-орієнтована мова для написання всіляких програм, від анімаційних програм до електронних таблиць, які можна завантажувати по Web в локальні машини. Тексти програм, написані на мові JavaScript, можна інтегрувати безпосередньо в HTML-документ.

Обслуговування Web-вузлів

На типовому Web-вузлі обов'язки персоналу можуть розподілятися серед чотирьох основних груп співробітників:

- постачальники даних працюють з даними – створюють і редагують HTML-документи, розміщують в них зображення і форми, забезпечують цілісність посилань;
- дизайнери створюють зображення і визначають "зовнішній вигляд" вузла;
- програмісти пишуть програми з використанням CGI, Java і JavaScript, які включаються в ресурси вузла;
- адміністратори забезпечують безперервну, нормальну і ефективну роботу серверу, вони також можуть відповідати за створення нових областей вмісту і нових сценаріїв, а також за підтримку належної безпеки важливих документів і вузла в цілому.

На крупному вузлі може бути приблизно 50 постачальників даних, п'ять дизайнерів, три-чотири програмісти і два адміністратори. На маленькому вузлі всі ці функції звичайно суміщає одна людина.

Огляд HTML

Гіпертекст – це звичний текст, який містить посилання як на власні фрагменти, так і на інші тексти. Концепція гіпертексту запропонована Теодором Хольмом Нельсоном. Простий приклад гіпертексту – книга, зміст якої містить посилання на розділи і розділи книги (тут посилання – це номер сторінки, з якою починається відповідні глава або розділ). Подивившись в зміст, людина знає номер сторінки потрібного їй

розділу, і відкриває книгу на сторінці, яку шукає. Для того, щоб знайти розділ книги, який цікавить її, людина не повинна проглядати всю книгу. Зміст надає їй можливість швидкого потрапляння на той розділ або розділ, який їй необхідний. За наявності мережі тексти, пов'язані один з одним посиланням, можна розміщувати на різних, територіально видалених комп'ютерах, і створювати і редагувати тексти можуть різні люди. Таким чином, створюється павутина взаємозв'язаних текстів, здатна стати гігантським інформаційним сховищем.

Гіпертекстові документи пов'язані один з одним виділеним набором слів, тобто до виділених слів поточного документа прив'язуються інші документи, і т.д. У середовищі WWW інші документи можуть розташовуватися на інших видалених ЕОМ.

Подальшим розвитком ідеї гіпертексту є гіпермедіа. Гіпермедіа – це синтез гіпертексту і мультимедіа. Тобто гіпермедіа-документ може включати не тільки гіпертекст, але і графіку, а також звук і відео.

HTML – це мова компоновки документів і специфікації гіперпосилань, яка використовується для кодування документів в системі World Wide Web. Засобами HTML задаються синтаксис і розміщення спеціальних вбудованих вказівок, відповідно до яких браузер відображає зміст документа: текст, зображення і дані інших типів, які підтримує даний браузер. Текст самих вбудованих вказівок Web-браузером не відображається. У мові HTML, крім того, реалізована підтримка механізму спеціальних гіпертекстових посилань, які забезпечують зв'язок даного документа з іншими документами (останні можуть знаходитися в локальній системі, в системі World Wide Web або можуть бути одержані за допомогою інших ресурсів Internet – FTP, Gopher і т.д.). Застосування механізму гіпертекстових посилань дозволяє зробити документ інтерактивним.

Базовий синтаксис і семантика мови HTML визначені в стандарті HTML. Стандарт HTML, як і всі інші стандарти, які мають відношення до Web, розробляються під егідою консорціуму World Wide Web Consortium (W3C). Специфікації стандартів і проекти нових пропозицій можна знайти за адресою <http://www.w3.org>. Найбільш функціональні реалізації HTML базуються на наборах тегів, які використовуються в більшості популярних браузерів.

Структура HTML-документа

HTML-документ складається з тексту, який є вмістом документа, і тегів, які визначають структуру і зовнішній вигляд документа при його відображенні браузером. Структура HTML-документа проста. Текст всього документа береться в теги:

```
<html>
<head>
<title>Barebones HTML Document</title>
</head>
<body>This illustrates in a very <i>simple</i> way,
the basic structure of an HTML document.</body>
</html>
```

Текст документа складається з елементів заголовка і тіла, які виділяються відповідно тегамі. У заголовку вказуються назва HTML-документа і інші параметри, які браузер буде використовувати при відображенні документа. Тіло – це та частина, в яку поміщається власне вміст HTML-документа. Тіло включає призначений для відображення текст і управляючі маркери документа (теги), які використовуються браузером. Теги містять вказівки про спосіб відображення тексту. За допомогою тегів, крім того, створюються посилання на файли, які містять додаткові дані (графіку, звук), і позначаються точки прив'язки (гіперпосилання, або якорі), за допомогою яких документи зв'язуються з іншими документами.

Синтаксис HTML-тегів

Теги HTML-документів в більшості своїй прості для розуміння і використання, бо вони утворені за допомогою загальноживаних слів англійської мови, зрозумілих скорочень і позначень. HTML-тег складається з імені, за яким може слідувати необов'язковий список атрибутів тега. Текст тега розміщується в кутових дужках (< і >). Простий варіант тега – ім'я, укладене в кутові дужки, наприклад, <head> або <i>. Для складніших тегів характерна наявність атрибутів, які можуть мати конкретні значення, визначені автором для видозміни функції тега.

Атрибути тега слідують за ім'ям і відділяються один від одного одним або декількома знаками табуляції, пропусками або символами повернення до початку рядка. Порядок запису атрибутів в тегу значення не має. Значення атрибуту, якщо таке є, слідує за знаком рівності, що стоїть після імені атрибуту. Якщо значення атрибуту – одне слово або число, то його можна просто вказати після знаку рівності, не виділяючи додатково. Всю решту значень необхідно брати в одинарні або подвійні лапки, особливо якщо вони містять декілька розділених пропусками слів. Довжина значення атрибуту обмежена 1024 символами.

Ось декілька прикладів тегів з атрибутами:

```
<a href="http://www.ora.com/catalog.html">
<ul compact>
<input name=filename size=24 maxlength=80>
<link title="Table of Contents">
```

Регістр символів в іменах **тегов** і атрибутів не враховується, чого не можна сказати про значення атрибутів. Наприклад, особливо **важливо** використовувати потрібний **регістр** при введенні URL **інших** документів в якості значення атрибуту href.

Найчастіше HTML-теги складаються з початкового і кінцевого компонентів, між якими розміщуються текст і **інші** елементи документа. Ім'я кінцевого **тега** ідентично імені початкового, **але** перед ім'ям кінцевого **тега** ставиться коса межа (/). Кінцеві **теги** ніколи не **містять** атрибутів. Наприклад, щоб **відобразити** текст курсивом, його потрібно укласти в **теги** `<i>`:

```
<i>This text in italics. </i>
```

При **використовуванні** вкладених **тегов** в документі потрібно дотримуватись особливої акуратності. Вкладені **теги** потрібно **закривати**, **починаючи** з самого останнього і **рухаючись** до першого. У приведеному нижче **прикладі** фраза, виділена жирним шрифтом (``), включена в текст **посилання** (``), яке **знаходиться** в тілі документа:

```
<body>
This is some text in the body, with a
<a href="another doc.html">link, a portion on which
is <b>set in bold </b> </a> </body>
```

Деякі HTML-теги не мають кінцевого компоненту, оскільки вони є автономними елементами. Наприклад, **тег** зображення ``, який служить для вставки в документ графічного зображення, кінцевого компоненту не вимагає. До автономних тегів також відносяться розрив рядка (`
`), горизонтальна лінійка (`<hr>`) і **теги**, що містять таку інформацію про документ, яка не впливає на його вміст, який відображається, наприклад, **теги** `<meta>` і `<base>`.

В деяких випадках кінцеві **теги** в документі можна опускати. Більшість **браузерів** реалізовано так, що при обробці тексту документа початковий **тег** сприймається як кінцевий **тег** попереднього. Найпоширеніший тог такого типу – **тег** абзацу `<p>`. Оскільки він використовується в документі дуже часто, то його звичайно ставлять **тільки** на початку кожного абзацу. Коли один абзац закінчується, **наступний тег** `<p>` сигналізує **браузеру** про те, що потрібно завершити даний абзац і **почати наступний**. Більшість авторів **тегом** абзацу взагалі не користуються. Є і **інші** кінцеві **теги**, без яких **броузери** **відмінно** працюють, наприклад, **тег** `</html>`. Проте, рекомендується включати по можливості більше кінцевих **тегів**, щоб уникнути плутанини і помилок при відтворенні документа.

Для **компоновки** і творчого оформлення інформації в Web-документах існує **ряд** інструментальних засобів: таблиці, кадри, **кольори**. Особливості їх **використання** детально **розглядаються** в [4]. Проте при проектуванні гіпертекстових документів необхідно мати на увазі **наступне**. Канали глобальних **мереж**, принаймні ті, які можуть собі дозволити більшість компаній, мають **вельми** обмежену пропускну спроможність в порівнянні з каналами локальних **мереж**. Багато хто **вважає** канал T-1 (1,544 Мбіт/с) високошвидкісним, **але** йому надзвичайно далеко до Ethernet на 10 Мбіт/с. Тому необхідно ретельно **планувати** графічне наповнення сторінок Web. Графіка повинна бути невелика за розміром, і одна сторінка не повинна **містити** більше декількох картинок. Інакше читачам сторінок **доведеться** **дуже** довго чекати появи сторінок гіпертексту.

Основна ідея гіпертекстових систем полягає в концепції автоматично підтримуваних зв'язків як усередині одного документа, так і між різними документами. Підтримка таких зв'язків дозволяє організувати гнучкі нелінійні текстові структури. Гіпертекстову систему складають вузли-поняття, і зв'язки-відносини між ними, **тобто** вона є семантичною **мережею**. Вузол звичайно є одиночним поняттям або ідеєю і може **містити** текстову, графічну або яку-небудь **іншу** форму інформації. На екрані користувача вузли можуть бути **представлені** як текст, як **частини** екрану і т.д., спеціально виділені, щоб передати їх семантичне **значення**. Доступ до об'єктів або вузлів може бути здійснений шляхом **використання** їх ідентифікаторів.

Зв'язки **представляють** відносини між вузлами-поняттями. Вони можуть бути двонаправленими, дозволяючи повертатися від **подальшого** вузла до попереднього. Зв'язки можуть бути **представлені** як набір мінімум двох ідентифікаторів об'єктів. Зв'язки також можуть **розглядатися** як об'єкти з власними ідентифікаторами (ідентифікаторами зв'язку), які можуть бути використані для **розділення** вказівної і змістовної інформації.

Складні вузли (що складаються з вузлів і зв'язків) можуть бути **представлені** як складні об'єкти або сукупність простих об'єктів. Визначення методів **створення**, знищення і зміни вузлів і зв'язків може здійснюватися через концепції абстрактних типів даних і інкапсуляції.

Вузли і зв'язки можуть бути розбиті на класи по їх семантичних ознаках. Вузли і зв'язки певного класу можуть **успадковувати** властивості **споріднених** суперкласів. Ці властивості можуть бути використані при **створенні** і управлінні шаблонів гіпертекстів. Властивості вузлів і зв'язків можна **змінювати** через схему еволюції. Інструменти **створення** і управління вузлами і зв'язками об'єднує в собі система-менеджер, вбудована в WWW-сервер.

При проектуванні бази даних Web-документів слід розробити семантичну **мережу** понять, які описують задану **предметну** область, і написати Web- документи.

3.6.4. Розрахунок економічної ефективності від **упровадження** обчислювальної **мережі** Джерела економічної ефективності

За оцінкою зарубіжних фахівців в області автоматизації управління, автоматизація роботи службовців в умовах промислових підприємств може скоротити загальні витрати на конторську діяльність приблизно на 25%. Проте, найважливішою метою автоматизації роботи є підвищення якості адміністративних рішень (якості інформації, що виробляється).

Джерелами економічної ефективності, що виникає від застосування ЕОМ в організаційному управлінні, є:

- зменшення витрат на обробку одиниці інформації;
- підвищення точності розрахунків;
- збільшення швидкості виконання обчислювальних і друкарських робіт;
- можливість моделювання зміни деяких змінних і аналіз результатів;
- здатність автоматично збирати, запам'ятовувати і накопичувати розрізнені дані;
- систематичне ведення баз даних;
- зменшення об'ємів інформації, що зберігається, і вартості зберігання даних;
- стандартизація ведення документів;
- істотне зменшення часу пошуку необхідних даних;
- поліпшення доступу до архівів даних;
- можливість використання обчислювальних мереж при зверненні до баз даних.

При аналізі ефективності АСУ важливо враховувати, що кінцевий ефект від застосування ЕОМ пов'язаний не тільки з відшкодуванням витрат на покупку, монтаж і експлуатацію устаткування, а, в першу чергу, за рахунок додаткового поліпшення якості ухвалюваних рішень (в той же час, як показує досвід, ефект, як правило, в розрахунку переоцінюють, а витрати недооцінюють).

Економічна ефективність інформаційних процесів визначається співвідношенням витрат на технічні засоби і на заробітну платню працівників з результатами їх діяльності. Відомо декілька підходів до визначення основних складових ефекту інформаційної діяльності. В основу цих понять встановлені поняття інформаційної продукції (різні види інформації), інформаційного ефекту, величини запобігання втратам, суспільно необхідного рівня інформованості і інші.

3.6.5. Розрахунок суми витрат на розробку, впровадження і експлуатацію обчислювальної мережі

Капітальні вкладення при впровадженні пропонованої задачі або підсистеми АСУ (К) розраховуються в тому випадку, якщо впровадження задачі спричиняє за собою придбання додаткових технічних засобів. Таким чином, витрати на впровадження обчислювальної мережі повинні розраховуватися по наступній формулі:

$$D_o = K_{ao} + K_{po} + K_{pl} + K_{mn} + K_{pp} \quad (1)$$

де

K_{ao} – вартість апаратного забезпечення ОМ;

K_{po} – вартість програмного забезпечення ОМ;

K_{pd} – вартість додаткових площ;

K_{mn} – одноразові витрати на наладку, монтаж і пуск ОМ;

K_{pp} – передвиробничі витрати (на науково-дослідницькі, дослідно-конструкторські роботи, підготовку і освоєння виробництва).

Якщо нові технічні засоби не будуть повністю завантажені пропованою задачею або підсистемою, то капітальні витрати визначаються з урахуванням коефіцієнта завантаження технічних засобів:

$$K_{зад} = K * r \quad (2)$$

де

r – коефіцієнт завантаження технічних засобів пропованою задачею.

$$r = T_v / T_{эфф.v} \quad (3)$$

де

T_v – час рішення задачі на v -му виді технічних засобів; $T_{эфф.v}$ – річний ефективний фонд часу роботи технічних засобів v -го вигляду.

Витрати на постановку задач, які вирішуються з використанням ОМ, їх програмування і впровадження визначаються на підставі експертних оцінок. В якості експертів виступають фахівці, які створюють і експлуатують інформаційні системи.

Ці витрати носять одноразовий характер і при розрахунку ефективності враховуються разом з додатковими капітальними витратами.

Використання обчислювальної мережі вимагає додаткових витрат на її експлуатацію і обслуговування. Витрати на витратні матеріали при використанні ЕОМ і периферійного устаткування (придбання паперу і стрічки для принтера, гнучких магнітних дисків, картриджів для заправки принтера і т.д.) в порівнянні з витратами на витратні матеріали при рішенні задач вручну, як свідчать експертні дані, навіть збільшуються приблизно на 5 %.

Експлуатаційні витрати на обчислювальну мережу визначаються по наступній формулі:

$$P_e = P_{zp} + P_{otc} + P_{накл} + P_{a.o} + P_{ел} + P_{pm} + P_{об} + P_{аб} \quad (4)$$

де

Ре – експлуатаційні витрати на **ОМ**;
 Рзп – витрати на сумарну заробітну **платню** працівників, обслуговуючих **ОМ**;
 Ротч – витрати по відрахуванню **із заробітної платні** до фондів соціального захисту;
 Р **накл** – витрати по відрахуваннях **із заробітної платні** на **утримання адміністративного апарату**;
 Ра.о – амортизаційні відрахування;
 Рел – витрати на електроенергію в рік при **використовуванні ОМ**;
 Ррм – **витрати** на витратні матеріали;
 Роб – витрати на навчання користувачів ЕОМ;
Раб – абонентська плата постачальнику послуг ОМ (для глобальної **мережі**).

$$Рзп = (Ор * Курн + Ор * Кдз) * m \quad (5)$$

де

m – кількість працівників;

$$Ротч = Рзп * Отч \quad (6)$$

$$Рнакл = Рзп * Накл \quad (7)$$

$$Ра.о = Цоб * a / 100 \quad (8)$$

Рел розраховується для кожного виду **устаткування** окремо і потім **одержані** результати підсумовуються:

$$Рел = N * Fд * Цен \quad (9)$$

де

Fд – дійсний річний **фонд часу роботи устаткування**

$$Fд = D * T \quad (10)$$

де

D – кількість робочих днів в році;

T – час роботи **устаткування** в добу;

$$Ррм = Цгд * n + Цп * m \quad (11)$$

де

n і m – відповідно кількість використаних витратних матеріалів.

2.20. **Початкові** дані для розрахунку **витрат** повинні бути зведені в таблицю, **приклад** якої наведено в таблиці

Таблиця 2.20

N	Стаття витрати	Скорочене найменування
1	Оклад працівника, який обслуговує ОМ	(Ор)
2	Регіональний коефіцієнт	(Кур)
3	Коефіцієнт додаткової зар. Платня	(Кдз)
4	Відрахування	(Отч)
5	Накладні витрати	(Накл)
6	Витрати на навчання персоналу	Роб
7	Ціна устаткування ОМ і ПО	(Цоб)
8	Норма амортизації	(a)
9	Нормативний коефіцієнт ефективності	(Ен)
10	Вартість гнучкого диска	(Цгд)
11	Вартість витратних матеріалів для принтера	(Цп)
12	Ціна 1 Квт. Година електроенергії	(Цен)
13	Потужність ЕОМ	(Nэвм)
14	Потужність принтера	(Nпр)
15	Потужність UPS	(Nups)
16	Час роботи ЕОМ в добу: 10 ЕОМ 2 ЕОМ	(Тэвм)
17	Час роботи принтера в добу	(Тпр)
18	Час роботи UPS в добу	(Tups)

3.6.6. Вибір методики розрахунку економічної ефективності

Ухвалюється рішення про галузь і умови **впровадження** обчислювальної **мережі**. Залежно від цього вибирається відповідна методика.

Вибір бази для порівняння

Вибір бази залежить від того, яка задача ставиться при розрахунку ефективності. Для цього аналізуються в сукупності основні **напрями** економічної ефективності інформаційних систем і схема їх впливу на техніко-економічні показники. У даному курсовому проекті необхідно виконати порівняння декількох варіантів інженерних **рішень** по розрахунковому коефіцієнту ефективності капітальних вкладень (Ер).

$$Ер = (Рэ1 - Рэ2) / K2 - K1; \quad (12)$$

Дані інженерні рішення признаються ефективними при $E_p \geq E_n$.

Для деяких інформаційних систем спочатку не мається на меті скорочення робочих місць, економія засобів, що відводяться на трудовий процес, а установка обчислювальної мережі проводиться з метою підвищення якості ухвалюваних рішень, установки єдиного регламенту проведення ділових процесів, підвищення якості обслуговування клієнтів, забезпечення колективної роботи службовців, які працюють в територіально видалених підрозділах, прагнення перевершити конкурентів і так далі. У зв'язку з цим обчислення показників економічної ефективності не представляється можливим, а ефект від установки обчислювальної мережі можна визначити на якісному рівні. Світова практика показує, що Intranet-мережі забезпечують величезні переваги і підприємствам і організаціям слід підготуватися до їх використання.

3.6.7. Порядок захисту

Захист проекту відбувається перед комісією, яка затверджується кафедрою.

Студент допускається до захисту за умови наявності підписаної керівником і студентом пояснювальної записки і розрахунково-графічної частини проекту.

Для захисту студенту відводиться 10-15 хвилин на виклад змісту роботи; в процесі захисту комісія виказує свої зауваження; виявлені помилки проекту повинні бути відмічені червоним олівцем.

За наслідками захисту (доповідь, відповіді на питання, якість проекту) виставляється оцінка у відомості і на титульному листі записки пояснення. У разі виявлення принципових помилок проект повертається на доробку.

Після захисту студент повинен здати пояснювальну записку керівнику проекту. У разі незадовільної оцінки призначається повторний захист з усуненням всіх помилок проекту або з видачею нового завдання. За відсутності достатнього матеріалу за проектом в контрольні терміни, студенту, як правило, видається нове завдання.

Список літератури для використання при курсовому проектуванні

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 2-е издание. – СПб.: Питер, 2003. – 894с.
2. Саати Т.Л. Математические модели конфликтных ситуаций. Пер.с англ. – М.: Сов. Радио, 1977. – 304 с.
3. Столлингс В. Современные компьютерные сети. 2-е издание. – СПб.: Питер, 2003. – 783с.
4. С. Спейнаур, В. Куэрсиа. Справочник Web-мастера: Пер. С англ. – К.Ж Издательская группа ВHV, 1997. – 368 с.