

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ**

**Харківський національний університет внутрішніх справ**

**Кафедра кібербезпеки та DATA-технологій, факультет №6**

## **МЕТОДИЧНІ МАТЕРІАЛИ**

### **ДО ПРАКТИЧНИХ ЗАНЯТЬ**

з навчальної дисципліни «**Мережеві технології**»

вибіркових компонент освітньої програми

першого(бакалаврського) рівня вищої освіти

125 «Кібербезпека»

«Безпека інформаційних та комунікаційних систем»

м. Харків  
2023р.

## **ЗАТВЕРДЖЕНО**

Науково-методичною радою  
Харківського національного  
університету внутрішніх справ  
Протокол від 30 .08.23 № 7

## **СХВАЛЕНО**

Вченою радою факультету № 6  
Протокол від 25 .08.23 № 7

## **ПОГОДЖЕНО**

Секцією Науково-методичної ради  
ХНУВС з технічних дисциплін  
Протокол від 29 .08.23 № 7

Розглянуто на засіданні кафедри кібербезпеки та DATA-технологій  
(протокол від 15 .08.23 № 8)

### **Розробники:**

1. Професор кафедри, д.т.н., професор Можєв О.О.
2. Професор кафедри, д.т.н., професор Семенов С.Г.
3. Доцент кафедри, д.т.н., Можєв М.О.

### **Рецензенти:**

1. Доцент кафедри боротьби з кіберзлочинністю ХНУВС, к.т.н., доцент Клімушин П.С.;
2. Завідувач кафедри ЕОМ ХНУРЕ д.т.н., професор, Коваленко А.А.

## 1. Розподіл часу навчальної дисципліни за темами

### 1.1.1. Розподіл часу навчальної дисципліни за темами (денна форма навчання)

Номер та назва навчальної теми	Кількість годин, відведених на вивчення навчальної дисципліни						Література, сторінки	Вид контролю
	Всього	з них:						
		лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття	Самостійна робота		
Семестр 4								
Тема № 1: Математичне моделювання цифрових систем.	30	4		4	4	18	1	
Тема № 2: Аналіз цифрових систем в часовій області.	60	20		8	6	26	1	
Всього за семестр	90	24		12	10	44		залік
Семестр 5								
Тема № 3: Цифрові фільтри	40	8		6	12	14	2	
Тема №4: Аналогово-цифрові перетворювачі та цифро-аналогові перетворювачі	50	8		4	8	30		
Всього за семестр	90	16		10	20	44		екзамен
Всього по дисципліні	180	40		22	30	88		екзамен

### 1.1.2. Розподіл часу навчальної дисципліни за темами (заочна форма навчання)

Номер та назва навчальної теми	Кількість годин, відведених на вивчення навчальної дисципліни						Література, сторінки	Вид контролю
	Всього	з них:						
		лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття	Самостійна робота		
Семестр 4								
Тема № 1: Математичне моделювання цифрових систем.	30	2				28	1	
Тема № 2: Аналіз цифрових систем в часовій області.	60	4				56	1	
Всього за семестр	90	6				84		залік
Семестр 5								
Тема № 3: Цифрові фільтри	40			6	6	28	2	
Тема №4: Аналогово-цифрові перетворювачі та цифро-аналогові перетворювачі	50			4	4	42		
Всього за семестр	90			10	10	70		екзамен
Всього по дисципліні	180	6		10	10	88		екзамен

## 2. Методичні вказівки до практичних занять

### СЕМЕСТР 8

#### ПРАКТИЧНІ ЗАВДАННЯ З ОРГАНІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ

##### 1. Використання особливостей анімації при створенні проекту мережі

Мета роботи: на основі NetCracker та Cisco Packet Tracer вивчити можливості моделювання та навчитися створювати топологію мережі.


#### Теоретичні відомості

Запустіть NetCracker.

Відкрийте файл Router.net.

Розмістіть робоче вікно так, щоб збільшити місце для роботи з проектом (Рис. 1.1).

Для запуску анімації на панелі задач (Рис. 1.2) виберіть кнопку Start

 або з меню Control виберіть команду Start і зачекайте коли почнеться рух пакетів у робочій області.

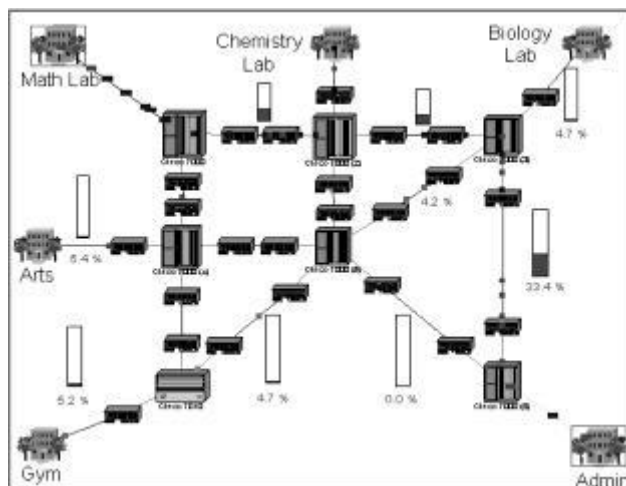


Рис. 1.1 Робоче вікно з анімацією



Рис. 1.2. Панель керування

Для зміни параметрів режиму моделювання натисніть на кнопку

Animation Setup, що показано на рис. 1.3.

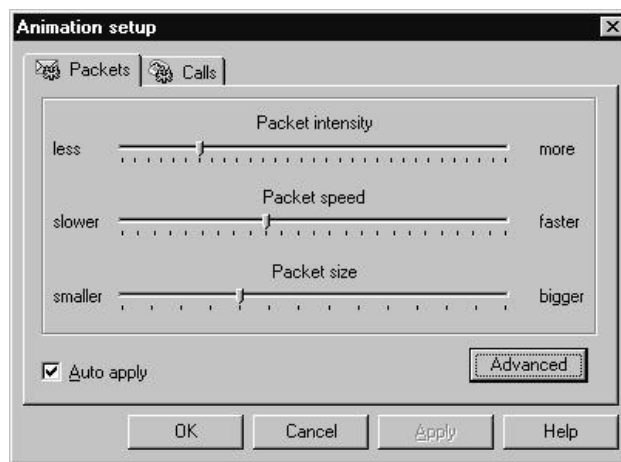


Рис. 1.3. Встановлення режимів моделювання

Скористайтесь лівою кнопкою миші для зміни розміру та швидкості пакетів. Натисніть ОК для збереження параметрів і закриття вікна.


Відкрийте нижчий, другий рівень мережі, двічі клацнувши по будові,

поміченій як Math Lab (Математична лабораторія). Отримаємо робочу область, показану на рис. 1.4.

Перегляньте, скільки інформаційних потоків пропускає підмережа математичної лабораторії, яке комутаційне устаткування задіяно в підмережі. Поверніться у верхній рівень проекту, закривши вікно математична

лабораторія.

Відкрийте підмережу Admin і перегляньте, скільки інформаційних потоків пропускає підмережа Admin, яке комутаційне устаткування задіяно в підмережі. Закрийте вікно Admin кнопкою Close.

Збільшіть масштаб зображення проекту (інструмент ) і розмістіть робочу область таким чином, щоб з'єднання (лінк) між Cisco 7000 (3) та Cisco 7000 (6) знаходилося у центрі робочої області.

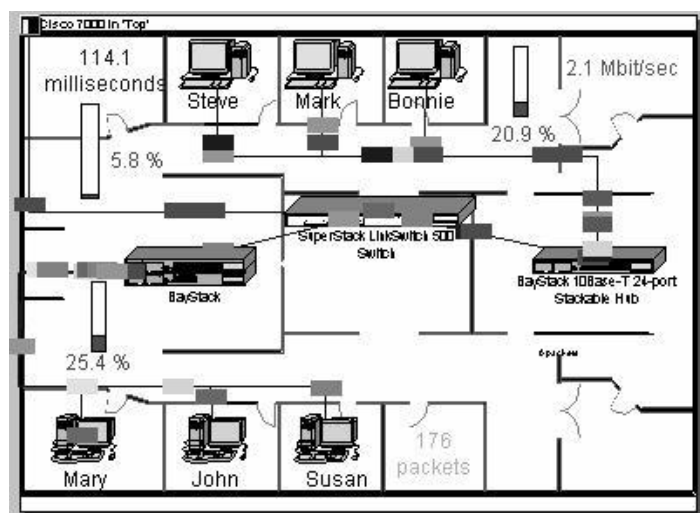



Рис. 1.4. Робоча область Math Lab


10. Щоб фізично розірвати зв'язок, на панелі Modes натисніть на кнопку Break/Restore. Потім помістіть курсор над з'єднанням двох маршрутизаторів Cisco і натисніть на з'єднання. З'явиться червона блискавка, яка сигналізує про те, що з'єднання обірвалося і трафік зупинився. Трафік змінюється згідно

роутинговим протоколом.

Тепер відновіть зв'язок: розмістіть курсор над перерваним зв'язком і клацніть лівою кнопкою миші. Курсор прийме форму гайкового ключа,

указуючи, що ви знаходитесь в режимі Restore. При приміщенні курсора поверх перерваного зв'язку трафік відновлюється, спалахи червоного кольору зникають.

Вимкніть режим Break/Restore, натиснувши на інструментальній панелі Sites Modes кнопку Standard mode .

Для зупинки анімації натисніть кнопку Pause  на панелі інструментів. Щоб отримати інформацію про пакет, помістіть курсор над ним.

Викличте правою клавішою миші локальне меню, що показане на рис. 1.5, та виберіть команду Say Info для інформації про пакет.

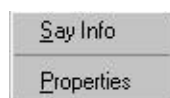


Рис. 1.5. Локальне меню пакетів

Встановіть курсор над пакетом та викличте з локального меню команду Properties. Відкриється діалогове вікно Properties, що показане на рис. 1.6.

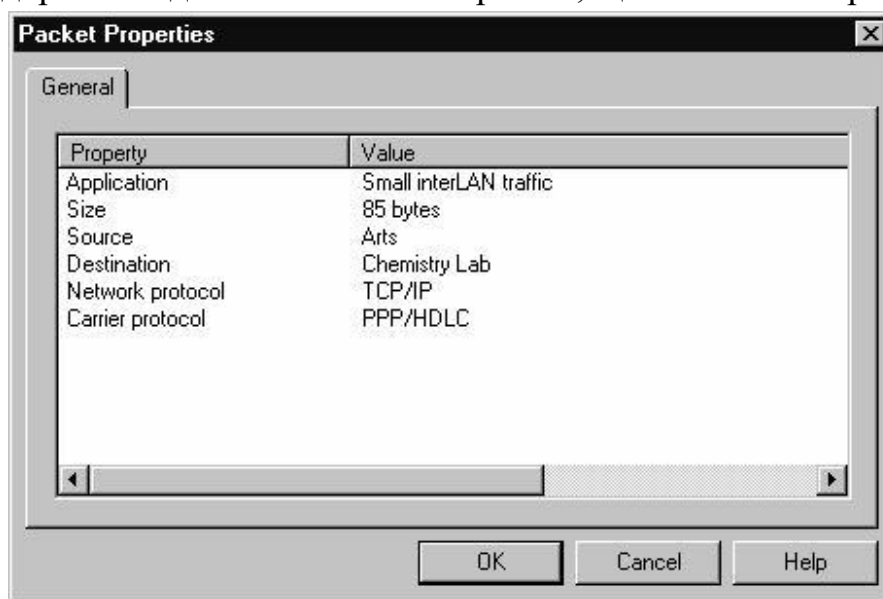


Рис. 1.6. Вікно діалогу Властивості Пакету

У вікно виводиться інформація, що стосується назви програми (Application), розміру пакета (Size), адреси відправника (Source), адреси



призначення (Destination), типу протоколу мережі (Network protocol) та протоколу канального рівня (Carrier protocol). Закрийте діалогове вікно, натиснувши OK або Enter.

Створіть вигин у лінії зв'язку:

переключіть в стан паузи, натиснувши кнопку Pause,

утримуйте кнопку CTRL на клавіатурі, двічі клацніть ліву кнопку миші прямо на зв'язку. На зв'язку з'явиться чорний квадрат,

натисніть і утримайте клавішу миші на чорному квадраті і перетягніть його до нового положення, потім відпустите ліву кнопку миші.

Зв'язок згинається в місці, що вказано і дані слідують навколо вигину зв'язку.

Перейменуйте об'єкт GYM:

клацніть правою кнопкою на будівлі під назвою GYM, щоб звернутися до локального меню, і виберіть команду Properties,

надрукуйте Кафе в полі імені, натисніть OK і закрийте діалог властивостей.

Виведіть звіт про діалоги і затримку передачі інформації в мережі:

Tools - Reports - Wisard - Statistical - Data flows - Application Statistic.

Для закриття проекту спочатку зупиніть анімацію, а потім закрийте файл з меню File. Коли буде виведене повідомлення про те, чи зберігати зміни, натисніть No.

## Тестові завдання

Відкрийте файл проекту NetCracker, вказаний у варіанті індивідуального завдання (Табл. 1.1), ознайомтесь з структурою його верхнього рівня і підрівнів, запустіть анімацію і змодельуйте ситуацію виведення окремого приладу, при необхідності налаштуйте параметри анімації, сформуйте стандартні звіти:

**Табл. 1.1.** Варіанти тестових завдань

Варіант	Файл
<b>1</b>	Hier.net
<b>2</b>	Techno.net
<b>3</b>	Tutor.net
<b>4</b>	Hier.net
<b>5</b>	Techno.net

про використання обладнання (Device Utilization),  
про статистику роботи сегментів WAN (WAN Segments Statistics),  
про статистику роботи сегментів LAN (LAN Segments Statistics).  
Збережіть структурну схему пропонованої мережі і сформовані звіти для включення в загальний протокол роботи.

Вимоги до протоколу роботи

Протокол має містити

Назву, тему та мету роботи.

Розділ 1 «Короткий опис, конфігурація і склад устаткування заданої мережі».

Розділ 2 «Короткий опис, конфігурація і склад устаткування підмереж».

Розділ 3 «Характеристики інформаційних пакетів».

Розділ 4 «Звіти про роботу мережі»:

про використання обладнання (Device Utilization).

про статистику роботи сегментів WAN (WAN Segments Statistics).

про статистику роботи сегментів LAN (LAN Segments Statistics).

Висновки (з відповідями на контрольні питання).

## 2. Розробка нового проекту в середовищі NetCracker

Мета роботи: створювати проекти NetCracker, вибирати комунікаційні пристрої, з'єднувати їх цифровими каналами, генерувати анотований звіт для покращення презентації проекту.

### **Теоретичні відомості**

Запустіть NetCracker.

У меню File виберіть команду New. Якщо проект відкрито (.NET) у робочому вікні, то зберегти цей проект перед відкриттям іншого проекту. Не зберігайте жодного з проектів-прикладів NetCracker. Відкрийте вікно проекту до максимальних розмірів.

В браузері пристроїв виберіть Switches. Переконайтесь, що для браузера пристроїв вибраний вид представлення Types (в списку Hierarchy).

браузері пристроїв послідовно розкрийте групи Switches, Workgroup, Ethernet і виберіть каталог Bay Networks для відображення Bay Networks

switches в області зображень пристроїв.

Для розміщення комутатора у вікні проекту виконайте наступні

кроки:

a. Виділіть модель BayStack 28104/ADV Fast Ethernet Switch в області зображень та перетягніть її у вікно проекту.

b. Збільшіть зображення пристрою, пересуваючи маркери розміру.

c. Збільшіть зображення назви пристрою. Для цього викличте контекстне меню та виберіть в ньому пункт Properties. З'явиться діалог Title Properties, в якому змініть значення розміру з 16 на 26 та натисніть ОК, або Enter, для закриття діалогу.

d. Відтягніть поле з назвою пристрою від зображення пристрою і збільшіть його розміри за допомогою маркерів встановлення розміру.

Додайте дві робочі станції в проект. Для цього:

a. В браузері пристроїв знайдіть і розгорніть групу LAN workstations.

b. Виберіть підгрупу Workstations, а в ній каталог Digital Equipment.

У полі зображень відобразяться робочі станції, виготовлені Digital Equipment Corporation.

с. Виділіть робочу станцію Alpha Station 200 4/166 та перетягніть її у вікно проекту. Змініть її розмір та розмір шрифту в назві.

d. Виберіть каталог PC у групі LAN workstations. Тепер скористайтесь смугою прокручування в браузері та виділіть каталог IBM.

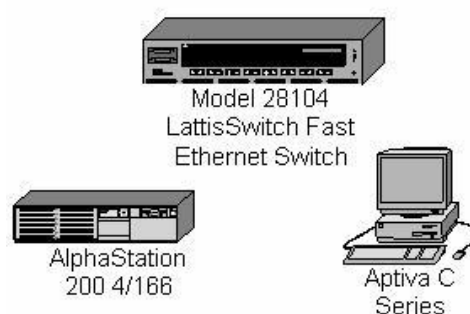
Виділіть Aptiva C Series в області зображень та перетягніть в робоче поле, змініть розмір пристрою і кегль шрифту на 26пт та розмір назви. Зображення виглядатиме як на рис. 1.7.

Розмістіть LAN adapter на кожній з двох робочих станцій.

a. В браузері пристроїв розгорніть групу LAN adapters і підгрупу Ethernet.

b. Виберіть каталог 3COM Corp.

с. В області зображень відшукайте Fast EtherLink 10/100 PCI карту, виділіть її, перетягніть на Alpha Station 200 4/166 та відпустіть кнопку миші після зміни курсора на знак плюс. Курсор змінює своє зображення на знак плюс, якщо карта підходить для робочої станції. Якщо форма курсору не змінюється, значить дана карта не сумісна з робочою станцією.



*Рис. 1.7. Структура простої мережі*

d. Виділіть Fast EtherLink 10/100 PCI карту знову, перетягніть у вікно

проекту та вставте у робочу станцію Aptiva C Series, щоб курсор змінився на знак плюс та відпустіть клавішу мишки.

Для розпізнання пристроїв, які є сумісними з виділеними, виконайте наступне:


Виділіть пристрій.

Виберіть команду Find Compatible з меню Object або натисніть кнопку Compatibles на панелі інструментів Database.

3. Відкрийте список LAN adapters, в ньому - Ethernet список, потім виділіть каталог продавців.

Виберіть сумісний пристрій в області зображень та замініть ним попередньо вибраний пристрій.

Приєднайте робочі станції до комутатора.

а. На панелі Modes натисніть на кнопку Link devices (з'єднання пристроїв) .

Помістіть курсор на Alpha-станцію та натисніть на зображення пристрою, потім перемістіть курсор на комутатор та натисніть на його зображення.

З'явиться вікно діалогу Link Assistant, що показано на рис. 1.8.

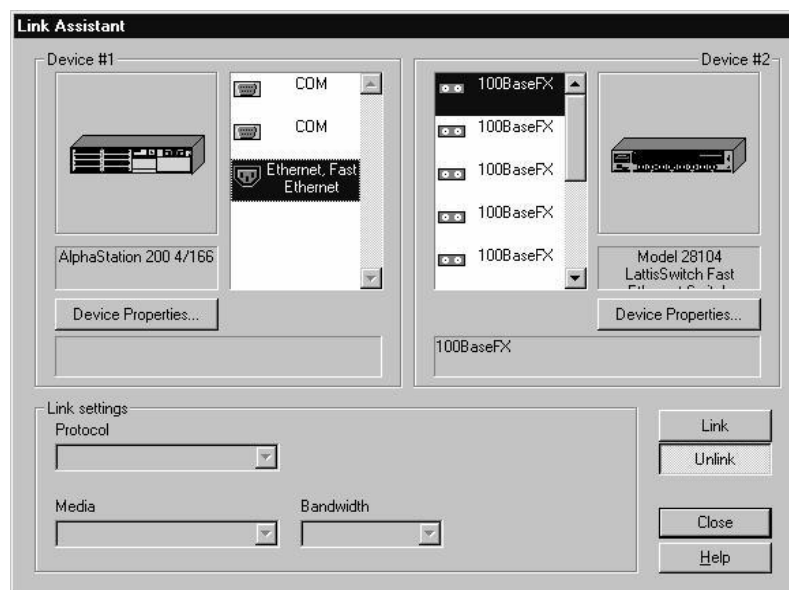


Рис. 1.8. Вікно діалогу Link Assistant

с. Натисніть кнопки Link і далі Close для створення з'єднання та закриття

діалогу.

d. Скористайтесь методом Quick Link для з'єднання робочої станції IBM з комутатором, натиснувши клавішу Shift, лівою клавішою мишки натисніть на комутаторі, потім – на робочій станції IBM (при цьому режим Link Mode з виділеною кнопкою Link devices – активний).

Діалог Link Assistant не з'являється. При цьому зв'язок між другою робочою станцією та комутатором встановлено.

Перевірте тип ліній зв'язку. Ви помітите, що колір з'єднання - жовтий.

a. Для перевірки типу лінії зв'язку вам потрібно звернутися до діалогу Legends, який викликається з меню View командою Legends. Жовтий колір вказує, що тип зв'язку – волоконно-оптичний.

b. Закрийте діалог Legends, натиснувши кнопку Close.

Визначте профіль трафіку до робочої станції.

a. Натисніть на кнопку Set Traffics 

b. Лівою клавішою мишки натисніть на робочій станції Alpha, потім – на робочій станції IBM.

Відкриється вікно діалогу Профілю (рис. 1.9).

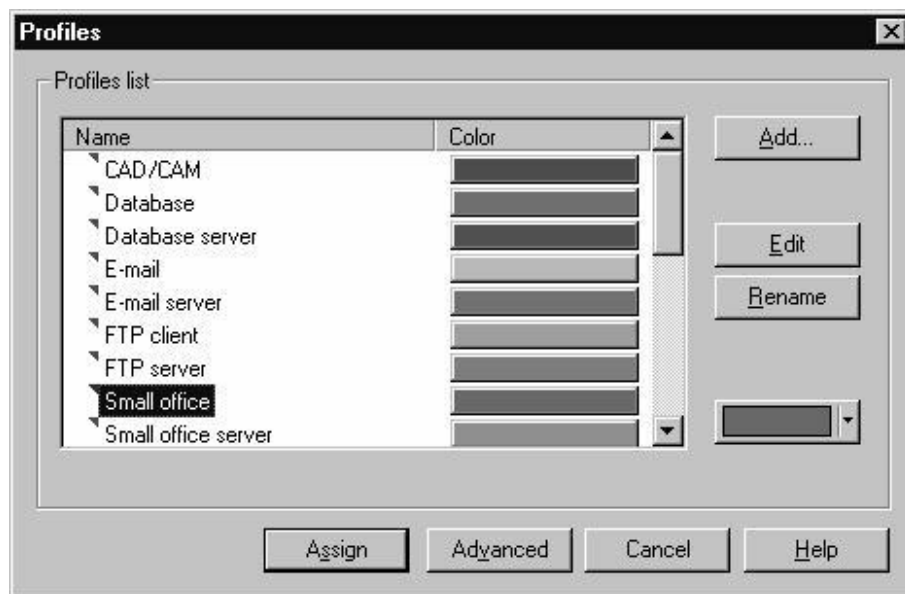


Рис. 1.9. Вікно діалогу Profiles


c. Для визначення трафіку типу Small Office між двома робочими

станціями виберіть з переліку профілів тип Small office. Виберіть колір трафіку з запропонованої кольорової палітри. З'являється повідомлення Do you want to update? ("Бажаєте оновити?"). Натисніть Yes для позначення усього трафіку цього типу певним кольором. Натисніть No для виділення тільки цього трафіку обраним кольором, при цьому колір інших трафіків цього типу змінено не буде.

d. Натисніть кнопку Assign для визначення трафіку та закрийте діалог. e. Повторіть кроки 9b-d, але цього разу виділивши першою робочу станцію IBM, а потім робочу станцію Alpha.

Перевірте встановлення трафіку між двома робочими станціями, запустивши анімацію. На панелі Control натисніть кнопку Start. З'явиться трафік між робочими станціями, який буде проходити через комутатор.

Змініть інтенсивність руху пакетів.


a. Натисніть на кнопку Animation Setup  щоб відкрити вікно діалогу Animation Setup.

b. Перейдіть на вкладку Packet intensity і перетягніть повзунок вліво, потім вправо на 4 пункти.

c. Натисніть OK для закриття вікна діалогу.

Через декілька секунд інтенсивність руху пакетів буде змінено.

Змініть швидкість руху пакетів.

a. Натисніть на кнопку Animation Setup  щоб відкрити вікно діалогу Animation Setup.

b. Перейдіть на вкладку Packet speed та перемістіть бігунок на найбільше



значення.

с. Натисніть ОК для закриття вікна діалогу.

Через декілька секунд швидкість руху пакетів буде змінено.

Зміна розміру пакетів.

а. Натисніть на кнопку Animation Setup щоб відкрити вікно діалогу Animation Setup.

б. Виберіть вкладку Packet size та перемістіть бігунок на найбільше значення.

с. Натисніть ОК для закриття вікна діалогу.

Перегляньте усі пристрої, використані у проекті мережі,

скориставшись вкладкою Recently used області зображень.

Розмістіть карту у вигляді фону.

а. Натисніть правою кнопкою мишки будь-де у вікні проекту (але не на зображенні пристрою і не на з'єднанні) для виклику локального меню, в якому виберіть команду Site Setup. Відкриється вікно діалогу Site Setup.

б. Перейдіть на вкладку Background, потім відмітьте пункт Map.

б. Скористайтесь кнопкою Browse для відкриття вікна діалогу Browse, знайдіть та виберіть бажаний файл карти, натисніть кнопку Open. Ім'я файлу з'явиться у полі Selected map file. Натисніть ОК для прийняття внесених вами змін та закриття вікна діалогу.

Вікно діалогу виглядає як на рис. 1.10. Окрім карт, які поставляються

продуктом, можна використовувати власні карти.

d. Натисніть ОК для прийняття змін та закриття діалогу. 16. Зробіть задній план кольоровим.

З меню Sites виберіть команду Site Setup.

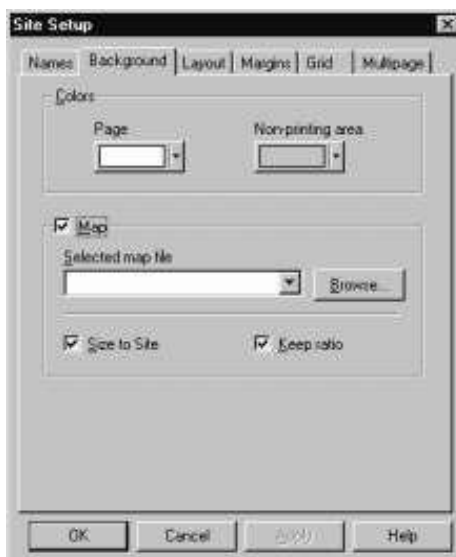


Рис. 1.10. Вікно діалогу Site Setup

Відкриється вікно діалогу Site Setup.

a. Перейдіть на вкладку Background, потім зніміть виділення з пункту

Map.

Поле Selected map file стане неактивним.

b. Натисніть на полі Page. Відобразиться кольорова палітра, виберіть колір та відпустіть кнопку мишки.

c. Натисніть на полі Non-printing area для перегляду кольорової палітри та виберіть інший колір, відпустіть кнопку мишки.

d. Натисніть OK для внесення змін та закриття діалогу.

Новий фон та колір будуть відображені. Також можете додати колір фону у вікно, в якому розміщена карта.

Перегляньте профайли трафіку. Виберіть команду Data flow з меню

Global. Відкриється вікно діалогу Data Flow (рис. 1.11). Два профайли трафіків, які створено, відображені в панелі Flows in the model (Потоки моделі). Натисніть Close щоб закрити вікно діалогу.

В меню File виконайте команду Save. Оскільки ви раніше не зберігали цей файл, то відкриється вікно діалогу команди Save As.

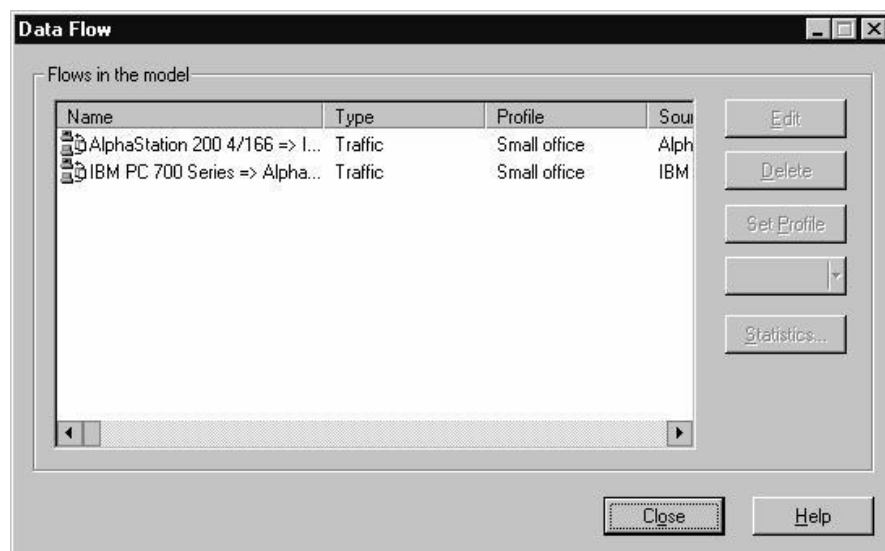


Рис. 1.11. Вікно діалогу Data Flow

За замовчанням файлу дається ім'я Net1.net, воно написане у полі Name. Перейменуйте файл. Розширення .NET автоматично буде додано до

імені файлу.

Зупиніть анімацію проекту кнопкою Stop, потім з меню File виконайте команду Close. Для закриття проекту спочатку зупиніть анімацію,

потім закрийте файл з меню File. На повідомленні про те, чи зберігати зміни, натисніть No.

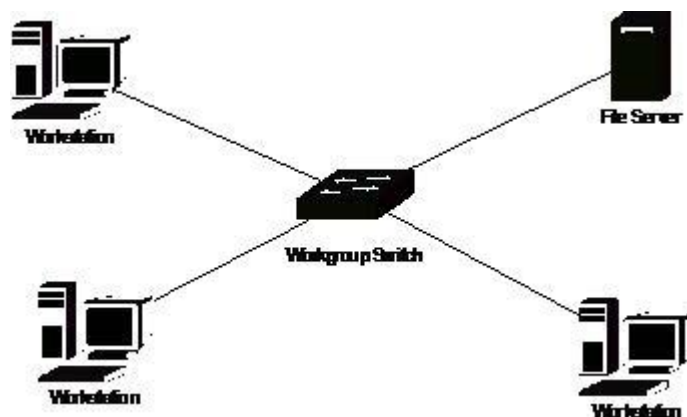
### **Тестові завдання**

Створіть проект мережі з топологією і складом обладнання, показаними на рис. 1.12.

Задайте трафік з профілями згідно варіанту завдання за таблицею 1.2.

Перед установкою трафіку станція-сервер встановіть необхідне серверне програмне забезпечення, вибравши його в групі Network and enterprise software- Server software браузера пристроїв.

Запустіть анімацію і переконайтесь в працездатності моделі.



*Рис. 1.12.* Проект мережі з топологією «зірка»

*Таблиця 1.2.* Варіанти тестового завдання

<b>Варіант</b>	<b>Трафік станція-станція</b>	<b>Трафік станція-сервер</b>
<b>1</b>	Peer-to-peer	SQL server's client
<b>2</b>	Peer-to-peer	FTP client
<b>3</b>	Peer-to-peer	HTTP client
<b>4</b>	Peer-to-peer	E-Mail (SMTP)
<b>5</b>	Peer-to-peer	E-Mail (POP)

Сформууйте стандартні звіти:

використання обладнання (Device Utilization),

про статистику роботи додатків (Application Statistics),

про статистику роботи мережевих пристроїв (Network Devices Statistics).

Збережіть структурну схему пропонованої мережі і сформовані звіти для включення в протокол роботи.

### 3. Розробка багаторівневого проекту мережі

Мета роботи: ознайомитись з методами створення багаторівневих мережевих проектів, моделювання архітектури клієнт/сервер, відображення підсумків моделювання.

### Теоретичні відомості

Відкрийте файл Tutor.net, який знаходиться в підкаталозі Samles.

Переключіться в браузері програми на вкладку Project Hierarchy.



Проекти демонструються як багаторівнева ієрархічна структура. Для проектів з одним рівнем буде відображено тільки верхній (Top) рівень. Кожен рівень має символи розкриття (знак +), або згортання (знак -) ієрархічної структури. Кожному рівню в Project Hierarchy відповідає окреме вікно сайту.

Кожен елемент ієрархії на вкладці Project Hierarchy відображається в робочій області у вигляді окремого вікна після подвійного кліка.

Вкладені рівні у вікні проекту обводяться червоною рамкою.

Виконайте подвійне клацання по об'єкту-контейнеру Building (рис. 1.13).

Тепер відобразіть головне вікно проекту Top, вибравши з меню Для відображення обох вікон проекту виберіть з меню Window команду Cascade.

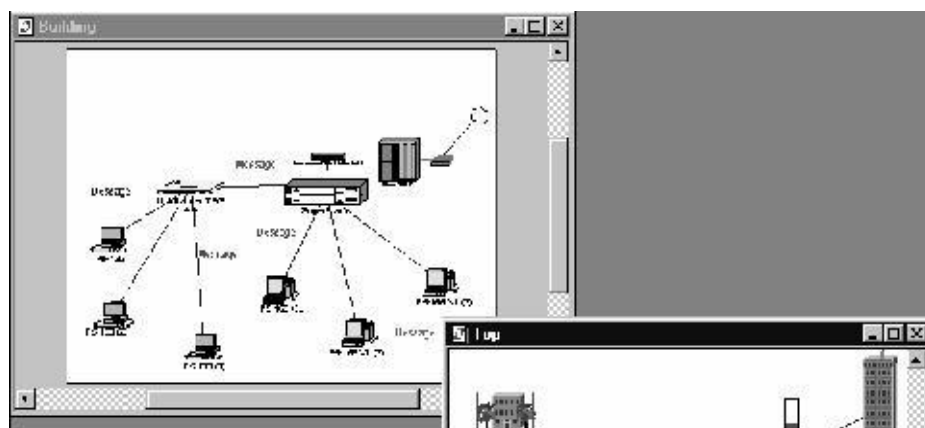
Поміняйте вікна місцями, змініть їхні розміри за допомогою команди Zoom.

*Рис. 1.13. Об'єкт-контейнер*

Залиште головне вікно проекту поточним. Робоче поле може виглядати як на рис. 1.14.

Тепер закрийте вікно Top натиснувши кнопку Close.

Щоб відкрити знову вікно Top, у вікні Building двічі натисніть на позначці з'єднання (connector icon). Вікно Top відкриється знову.



*Рис. 1.14. Багаторівневий проект*

Перейменуйте вікно сайту (проекту).

- a. Спочатку зробіть вікно Top поточним, натиснувши на ньому.
  - b. Відкрийте діалог Setup вибравши з меню Sites команду Site Setup.
  - c. Виберіть вкладку Names. Підсвітіть ім'я (Top) в полі Site name, потім надрукуйте "The MacNally Corporation".
  - d. Натисніть ОК для запам'ятовування змін та закрийте вікно діалогу.
  - e. Аналогічно перейменуйте вікно Building на "The MacNally Building".
- Нові назви The MacNally Corporation та the MacNally Building з'являться в заголовках вікон, у Project Hierarchy та в командах меню

Window.

Анотуйте проект, використовуючи інструментальну палітру (drawing tools).

- a. Зробіть поточним вікном MacNally Building та виділіть знак з'єднання.
- b. В панелі Modes натисніть кнопку Draw.
- c. Виберіть кнопку рисування ліній (Line) на панелі рисування (Drawing

toolbar).

Скористайтесь Line для рисування стрілки, яка вказує на верхній правий куток вікна сайту. Поверніться в стандартний режим (Standard mode) натиснувши на кнопки з рисунком стандартного курсору.

Змініть колір стрілки: виділіть лінію,  
з меню Object, з підменю Styles виберіть команду Draw color,

виберіть колір та натисніть OK,

повторіть попередні дії для кожної ділянки стрілки.

е. Для помітки значка з'єднання, виконайте наступні дії:

панелі Modes виберіть кнопку Draw,

панелі Drawing (рис. 1.15) виберіть інструмент Text,

виділіть прямокутник, в якому буде розміщено текст.




Рис. 1.15. Панель Drawing

Надрукуйте “Link to MacNally Corporation” та натисніть Enter.

Поверніться в режим Standard, натиснувши в панелі інструментів кнопку стандартного режиму (Standard mode).



Виділіть канал зв'язку пристроїв, використавши команду Trace. а.  
Запустіть анімацію, натиснувши кнопку Start.

б. На панелі Modes натисніть кнопку Trace . Потім у вікні MacNally Building кладніть на робочій станції P5-166 XL (3) і на робочій станції P5-133 XL (3).

Канал зв'язку між цими робочими станціями буде підсвічено червоним кольором.

Виділіть канал зв'язку між об'єктами, що знаходяться в різних вікнах проекту.

а. Якщо режим Trace все ще ввімкнутий, натисніть у вікні MacNally Building робочу станцію P5-133 XL (3).

б. Потім натисніть на Building (2) у вікні MacNally Corporation.

Канал зв'язку між цими двома об'єктами буде підсвічено червоним кольором.

с. Поверніть проект до стандартного режиму, натиснувши кнопку Standard mode.

Якщо натиснути на кнопці Trace, підсвічення червоним кольором зніметься, але режим Trace залишиться.

Зупиніть анімацію кнопкою Stop.

Закрийте поточний проект без збереження, вибравши з меню File

команду Close.

Відкрийте новий проект командою New, меню File.

В робочій області з'явиться нове вікно.

У браузері пристроїв натисніть на вкладку Devices. Виберіть список

Buildings, campuses and LAN workgroups.

В області зображень (Image) з'являться об'єкти Buildings, campuses and LAN workgroup.

Виділіть один з об'єктів Building в області зображень та перетягніть

його у вікно Top.

Перетворіть об'єкт Building на контейнерний об'єкт, скориставшись командою Expand.

Виділіть об'єкт Building у вікні проекту, потім правою кнопкою мишки відкрийте локальне меню та виберіть команду Expand, або виберіть команду Expand з меню Object.

Зображення Building у вікні Top тепер обведено червоною рамкою, що означає, що це – контейнерний об'єкт.

Щоб побачити ієрархічну структуру, в Browser натисніть вкладку Project Hierarchy.

Закінчіть проект, побудувавши в будівлі Building найпростішу мережу з архітектурою клієнт/сервер. Краще використовувати типові, вже зконфігуровані пристрої (Generic Devices), які включені в базу пристроїв

NetCracker Professional. Для цього натисніть вкладку Devices в браузері об'єктів, потім відкрийте список LAN workstation та виберіть каталог типових пристроїв Generic Devices. Типові робочі станції відображаються в області зображень Image (рис. 1.16).



*Рис. 1.16. Приклади зображень робочих станцій*

а. У області зображень Image виділіть та перетягніть у вікно Building робочу станцію Ethernet, яка вже має карту LAN-адаптера.

б. З меню Edit виконайте команду Duplicate.

с. В браузері Device відкрийте список концентраторів (Switches), в ньому виберіть Workgroup, потім Ethernet та натисніть на каталозі типових пристроїв.

У полі Image виділіть типовий Ethernet-комутатор (рис. 1.1) та перетягніть його у вікно Building.

Натисніть на кнопці Link devices .

Натисніть на робочій станції та тягніть з'єднання до концентратора.  
Відпустіть кнопку мишки.

З'явилося вікно діалогу Link Assistant, в якому натисніть кнопку Link, потім Close.



*Рис. 1.1. Приклад зображення комутатора*

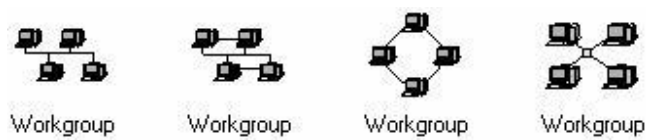
g. Повторіть ці дії і для другої робочої станції.

h. Зробіть вікно Top поточним.

Перейдіть в стандартний режим роботи, потім виділіть список

Buildings, campuses and LAN workgroups у браузері Device.

полі Image з'явилися зображення Buildings, campuses and LAN workgroup device (рис. 1.18)



*Рис.1.18. Приклади типових робочих груп*

Виділіть та перетягніть зображення типової робочої групи у вікно Top

k. У вікні Top з'єднайте робочу групу з Building (в панелі Modes натисніть

на кнопку Link, потім натисніть мишкою на робочій групі, потім на будівлі). Пунктирна лінія вказує, що це з'єднання ще не завершено.

Перейдіть у стандартний режим та двічі натисніть на зображенні Building у вікні Top. Відкриється та стане поточним вікно Building.

m. З панелі Modes виділіть команду Link devices. У вікні Building для завершення з'єднання натисніть на зображенні з'єднання (Connector icon), потім на концентраторі.

Відкриється діалог Link Assistant. Зображення з'єднання (Connector icon) зазвичай розміщено в куточку вікна. Якщо потрібно, скористайтесь збільшувальною кнопкою для знаходження зображення з'єднання.

n. Виберіть Ethernet port в панелі Switch Port Configuration, натисніть кнопку Link, потім Close. Діалог Link Assistant закриється та встановиться з'єднання між Building та Top.

Зробіть одну з робочих станцій сервером.

a. В браузері пристроїв Device спустіться до списку “Network and enterprise software”. В ньому виберіть “Server software”, після чого в області зображень Image з'являться доступні типи серверів.

b. Перетягніть E-mail сервер на зображення робочої станції. Курсор має змінитися на стрілку зі знаком плюс, що буде означати, що ви можете встановити це програмне забезпечення.

Призначте трафік клієнт-сервер.

В інструментальній панелі Modes натисніть кнопку Set Traffic.

У вікні Building натисніть на робочій станції без серверного програмного забезпечення, потім у цьому ж вікні натисніть на робочій станції з серверним програмним забезпеченням.

Відкриється вікно діалогу Profiles.

Виберіть тип трафіку E-mail та натисніть кнопку Assign. 21. Призначте трафік група-станція.

a. У вікні Top натисніть на зображенні робочої групи, потім перейдіть у

вікно Building та натисніть на робочій станції без серверного програмного забезпечення. Відкриється вікно діалогу Profiles.

b. Виберіть тип трафіку Small office та натисніть кнопку Assign.

c. Для запуску анімації натисніть кнопку Start на інструментальній панелі Control.

d. Для зупинки анімації натисніть кнопку Stop.

### **Тестові завдання**

В якості індивідуального завдання додайте у вікно Building об'єкт Room (з групи Buildings, campuses and LAN workgroups браузера) і перетворіть його на контейнер. У вікні контейнера Room побудуйте проект мережі у відповідності з варіантом завдання попередньої роботи. З'єднайте робочу групу у вікні Top з комутатором у вікні Room і задайте для трафіку між робочою групою і робочими станціями тип Small office. Запустіть анімацію і переконайтесь в працездатності моделі.

Сформууйте стандартні звіти:

про використання обладнання (Device Summary),

про статистику роботи додатків (Application Statistics),

про статистику роботи модулів (Module Statistical Items).

Збережіть створений файл проекту NetCracker, скориставшись командою Save з меню File. Закрийте проект, вибравши команду Close з меню File. Закрийте NetCracker, вибравши команду Exit з меню File.

Вимоги до протоколу роботи

Протокол має містити:

Назву, тему та мету роботи.

Розділ 1 «Короткий опис, конфігурація і склад устаткування заданої мережі».

Розділ 2 «Опис мережевого трафіку»

Розділ 3 «Модифікація фонового зображення».

Розділ 4 «Звіти про роботу мережі»:

про використання обладнання (Device Utilization),

про статистику роботи додатків (Application Statistics),

про статистику роботи мережевих пристроїв (Network Devices Statistics).

Висновки (з відповідями на контрольні питання).

#### 4. Налаштування та використання бази пристроїв NetCracker

Мета роботи: знайомство з функціями системи налаштування пристроїв Device Factory та використовувати можливості комбінованого пошуку (Compatible Search) в базі даних.

### Теоретичні відомості

1. В NetCracker відкрийте проект Router.net з підкаталогу Samples.

Перейдіть на вкладку Project Hierarchy.

У браузері вікна Project Hierarchy виберіть вікно Math Lab.

Виділіть робочу станцію Steve.

Викличте Device Factory Wizard одним зі способів:

натисніть на кнопку Device Factory ,

виберіть з меню Database команду Device Factory,

з меню Object виберіть команду Add to Database: Via Factory.

Можна скористатися будь-яким з вказаних способів для виклику Device



Factory. Коли створюється новий пристрій на базі існуючого, то є деякі відмінності у використанні команд Device Factory з меню Database та Add to Database: Via Factory з меню Object. Кнопка Device Factory та команда Device Factory з меню Database звертаються до пристроїв, які розміщені в полі Image, а команда Add to Database: Via Factory звертається до пристроїв, виділених в вікні проекту (Project pane). Відкриється вікно діалогу Device Factory Wizard (рис. 1.19).

вікні Device Factory Wizard є варіанти:

створити з нуля (Create from scratch),

створити з робочої станції Steve (Create from Steve).

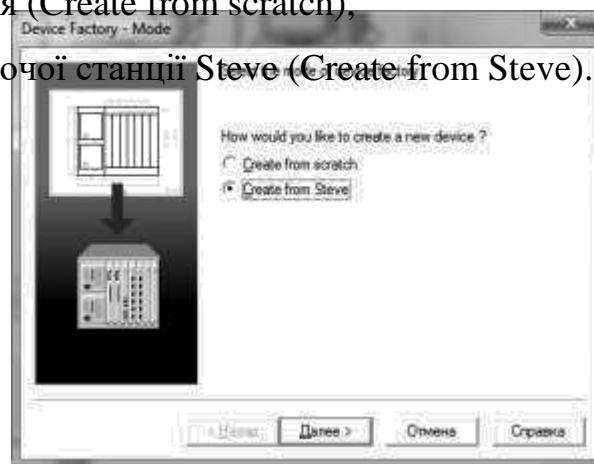
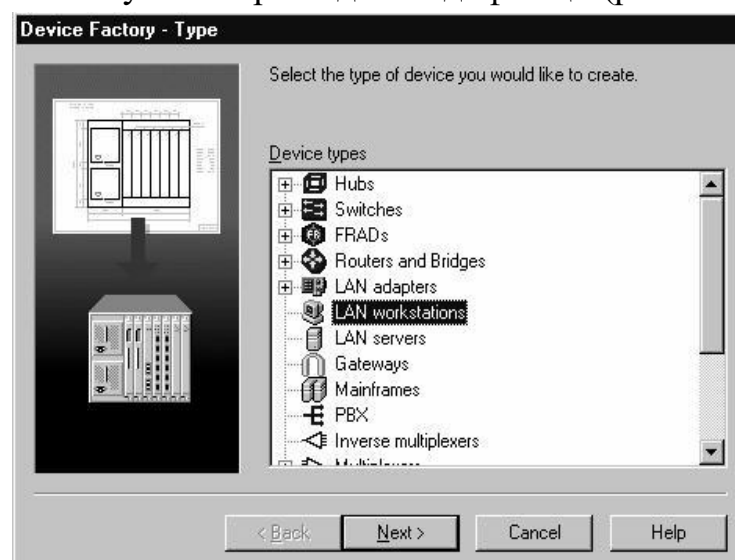


Рис. 1.19. Вікно Device Factory-Mode.

Виберіть варіант “Create from Steve” та натисніть кнопку Next.

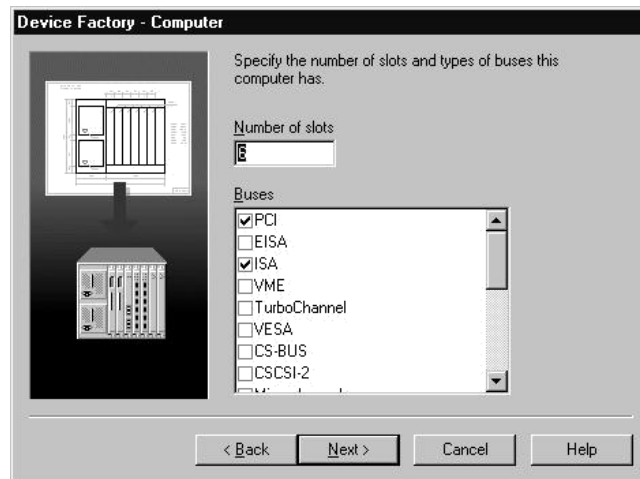
Тип пристрою, який ви вибрали, підсвічено і ви можете впевнитися, що пристрої саме цього типу ви вибрали для модифікації (рис. 1.20).



*Рис. 1.20. Вікно Device Factory Type Screen.*

Натисніть кнопку Next. Відкриється вікно Device Factory Computer (рис. 1.21).

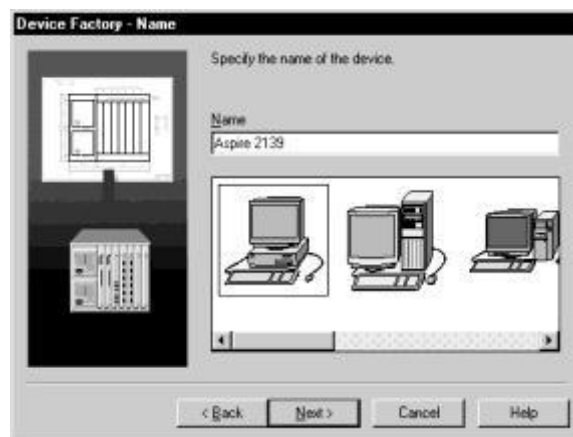
Змініть кількість слотів на 4. Тобто, скільки комп'ютер має місць для установки плат адаптерів та



*Рис. 1.21. Вікно Device Factory Computer*

В секції Buses відмітьте типи шин VESA, PCI та ISA. Новий створений пристрій буде приймати тільки ті адаптери та внутрішні модеми, які мають один з вибраних типів шин.

Натисніть кнопку Next. Відкриється вікно Device Factory Name (рис. 1.22).



*Рис. 1.22. Вікно Device Factory Name*

Надрукуйте ім'я "Development Group Workstation" та натисніть кнопку Next. Відкриється вікно Device Factory Port Groups (рис. 1.23).

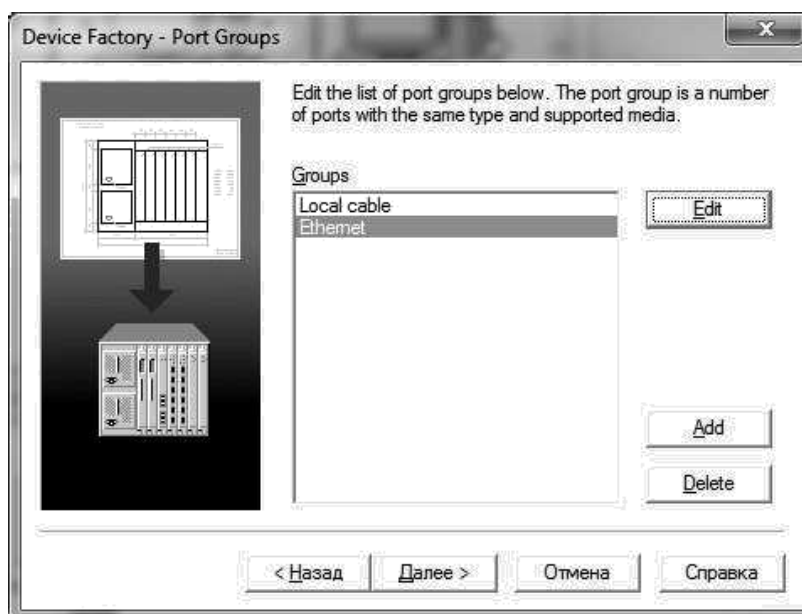


Рис. 1.23. Вікно Device Factory-Port Groups

Добавте групу порту Ethernet, натиснувши кнопку Add. Відкриється вікно Port Factory Number (рис. 1.24).



Рис. 1.24. Вікно Factory Number

Якщо ввести кількість портів більшу, ніж це можливо для цього типу пристрою, з'явиться повідомлення про помилку.

Змініть кількість портів у групі на 2 та натисніть кнопку Next.

З'явиться вікно Port Factory Link Type (рис. 1.25). Як мінімум один тип порту має бути виділений. Якщо нічого не виділено, з'явиться повідомлення про помилку.

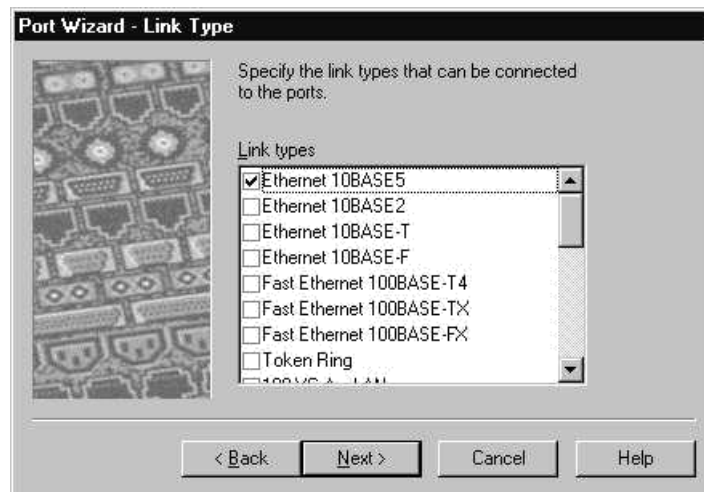


Рис. 1.25. Вікно Port Factory Link Type

Виділіть поля Ethernet 10BASE2 та Ethernet 10BASE-T та натисніть кнопку Next. Відкриється вікно Port Factory wizard Media (рис. 1.26).

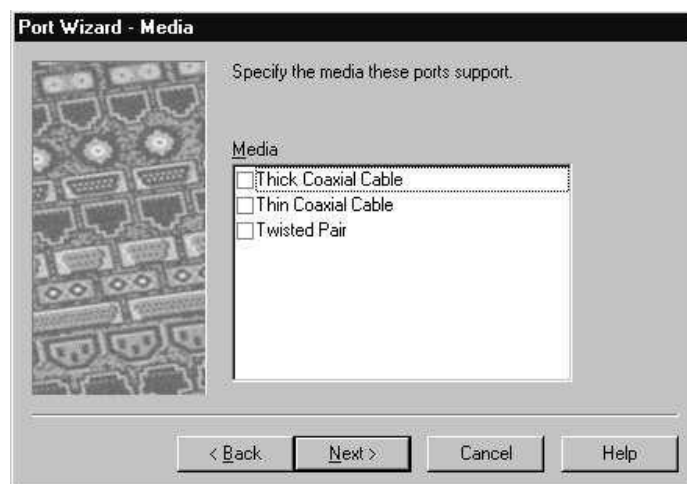


Рис. 1.26. Вікно Port Factory Media

Усі порти в групі All the ports in a group only except media of the type checked.

Як мінімум один тип медіа має бути виділений. Якщо нічого не виділено, з'явиться повідомлення про помилку.

Виділіть типи медіа тонкий коаксіальний кабель (Thin Coaxial Cable) та виту пару (Twisted Pair) і натисніть кнопку Next.

Натисніть кнопку Next на сторінці Device Factory Port Groups та натисніть Finish для збереження тільки що створеного пристрою в базі даних користувачів.

В меню File виберіть команду Close. Не зберігайте змін у проекті

Відкрийте новий проект та відобразіть браузер бази даних (якщо він не відображається), виконавши команду Database Browser з меню View.

Для відображення пристроїв з бази даних користувача, в якій вже є один тільки що створений пристрій:

з випадаючого списку Hierarchy (панель Database) виберіть значення User,

в області перегляду Image перейдіть на вкладку Devices.

Виберіть у області перегляду Image робочу станцію, яку ви тільки що створили - "Development Group Workstation" - та перетягніть її в робочу область проекту.

Тепер знайдіть пристрої, які сумісні з "Development Group Workstation". Для цього в панелі Database натисніть кнопку Compatible, або з меню Object виберіть команду Find Compatible. Браузер автоматично переключиться в режим Compatible Device Browser та відобразить сумісні пристрої. Браузер відображає тільки ті пристрої, які сумісні з виділеним пристроєм (рис. 1.27).

Знайдіть в базі даних сумісну АТМ-карту:

a. Виберіть Types у полі Database Hierarchy,

b. Розкрийте список LAN adapters, у ньому відкрийте список ATM, c.  
Відкрийте каталог Interphase.

Виділіть 5525 PCI ATM адаптер та перетягніть його на нову робочу станцію. Вигляд курсору зміниться на символ +, що означає сумісність карти.

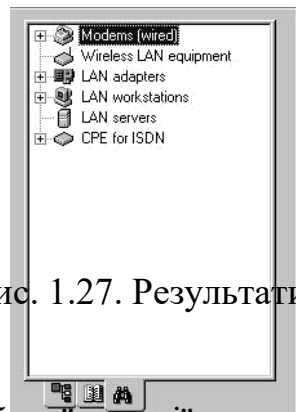


Рис. 1.27. Результати Compatible Search

Для копіювання робочої станції з картою адаптера, виберіть з меню

Edit команду Replicate. Відкриється вікно діалогу Replicate (рис. 1.28).

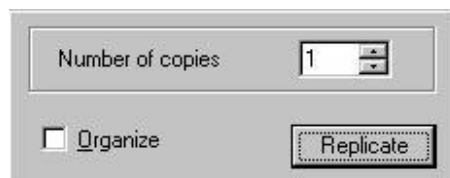


Рис. 1.28. Діалог Replicate

a. Для створення 10 копій, вкажіть число 10 у полі Number of copies,

Для впорядкування нових копій в геометричному порядку,

відмітьте поле Organize.

Натисніть кнопку Replicate. Відкриється вікно Organize (рис. 1.29).

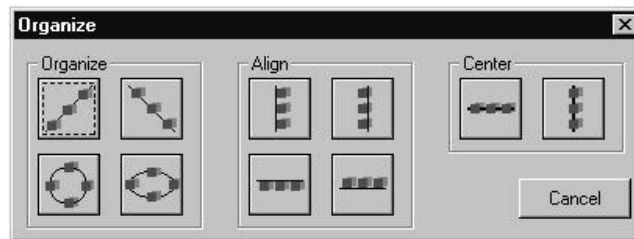


Рис. 1.29. Вікно діалогу Organize

Можна організувати будь-яку групу об'єктів, виділивши цю групу та виконавши команду Organize з меню Object.

Виберіть зразок кола. Після цього вікно діалогу автоматично закривається. Десять скопійованих об'єктів (робочі станції з картами)

розміщуються у формі кола.

Для пошуку пристроїв в базі даних за іншими критеріями скористайтесь пошуком по базі даних (Database search). Для початку пошуку натисніть кнопку Find з меню Database Відкриється вікно діалогу Find (рис.1.30).

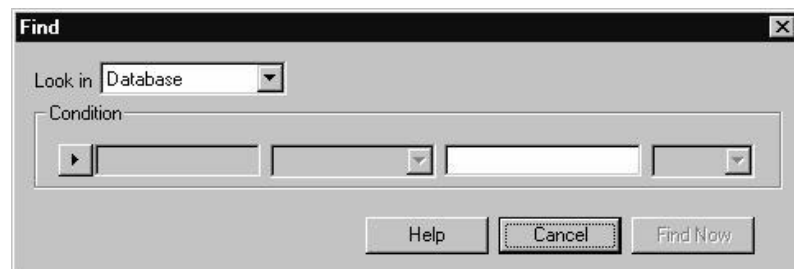


Рис. 1.30. Вікно діалогу Find

Натисніть на кнопку Condition та виберіть модель (Model). В наступному полі зі списком виберіть Includes. В третьому списку напишіть

7000 та натисніть кнопку Find Now. Браузер автоматично переключиться в режим Search Device Browser та відобразить результати пошуку.

Закрийте проект Router.net без збереження.

### **Тестові завдання**

В новому проекті у відповідності з варіантом завдання (Табл. 1.3) створіть і додайте до Device Factory комплект компонентів обладнання для побудови комп'ютерної мережі, що складається з мережевого адаптера, персонального комп'ютера з цим мережевим адаптером і комутатора.

Використайте створені компоненти для побудови 2-рівневого проекту комп'ютерної мережі, верхній рівень якої відображає дві будівлі, а на нижньому рівні – їх локальні мережі ЛОМ1 і ЛОМ2.

Задайте трафік з профілем Small Office Peer-to-peer між комп'ютерами однієї мережі і трафік з профілем Small InterLAN traffic між комп'ютерами різних мереж.

Запустіть анімацію і переконайтесь в працездатності моделі.

Сформууйте стандартні звіти:

про використання обладнання (Device Utilization),

про компонентний склад мережі (Network Summary),



про статистику роботи додатків (Application Statistics).

*Таблиця 1.3. Варіанти тестових завдань*

Варіант	Технологія Ethernet	Шина	Кількість портів комутатора	Кількість комп'ютерів	
				ЛОМ1	ЛОМ2
1	10Base5	ISA	8	7	10
2	10Base2	EISA	14	15	8
3	10Base-T	VESA	12	14	14
4	100Base-T4	PCI	16	20	6
5	100Base-TX	PCI	10	1	14

Збережіть структурну схему пропонованої мережі і сформовані звіти для включення в звіт про виконання роботи. Закрийте NetCracker.

Вимоги до протоколу роботи

Протокол має містити:

Назву, тему та мету роботи.

Розділ 1 «Створення і додавання в базу пристроїв NetCracker нових компонентів»

Розділ 2 «Короткий опис, конфігурація і склад устаткування заданої мережі»

Розділ 3 «Опис мережевого трафіку»

Розділ 4 «Звіти про роботу мережі»:

про використання обладнання (Device Utilization).

про компонентний склад мережі (Network Summary).

про статистику роботи додатків (Application Statistics).

Висновки (з відповідями на контрольні питання).

## 5. Моделювання мережі в Cisco Packet Tracer

Мета роботи: знайомство з багатофункціональною програмою моделювання мереж Cisco Packet Tracer та створення моделі мережі

### **Теоретичні відомості**

Багатофункціональна програма моделювання мереж Cisco Packet Tracer допомагає створювати мережеві топології з використанням маршрутизаторів і комутаторів Cisco, робочих станцій і мережевих з'єднань типу Ethernet, Serial, ISDN, Frame Relay. Дана програма моделювання є корисною як для навчання, так і для роботи. Наприклад, для роботи з налаштування мережі на етапі її створення, або під час усунення несправностей. Загальний вигляд програми

Cisco Packet Tracer показано на рис. 1.31.

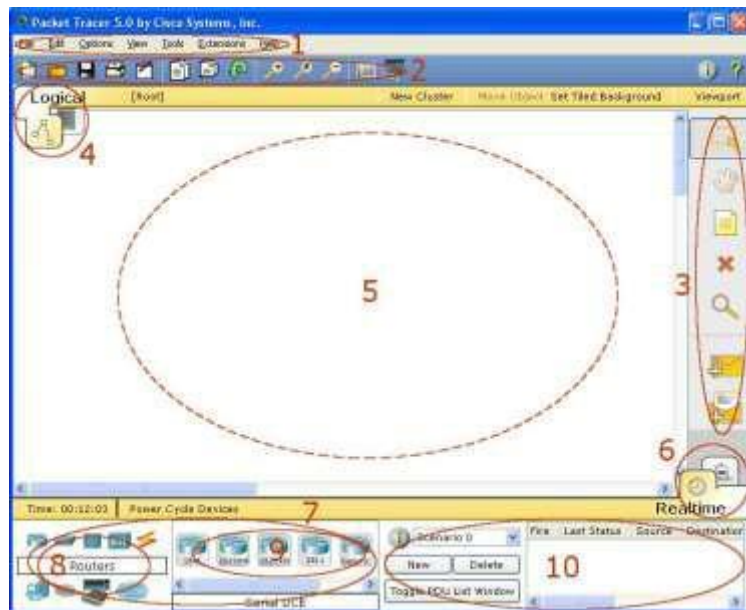


Рис. 1.31. Загальний вигляд програми Cisco Packet Tracer

Робоча область Packet Tracer складається з наступних елементів:

*Menu Bar* – панель, що містить меню File, Edit, Options, View, Tools,

Extensions, Help.

*Main Tool Bar* – панель містить графічні зображення ярликів для доступу до команд меню File, Edit, View і Tools, а також кнопку Network

*Common Tools Bar* – панель, яка забезпечує доступ до основних інструментів програми: Select, Move Layout, Place Note, Delete, Inspect, Add Simple PDU і Add Complex PDU.

*Logical/Physical Workspace and Navigation Bar* – панель, яка дає можливість перемикаати робочу область: фізичну або логічну, а також дозволяє переміщатися за рівнями кластера.

*Workspace* – область для побудови мережі та її моделювання, виведення інформації і статистики.

*Realtime/Simulation Bar* – панель, за допомогою закладок якої включають Realtime/Simulation режим. Панель має кнопки для Power Cycle Devices, Play Control і перемикач Event List в режимі Simulation.

*Network Component Box* – область, де обирають пристрої та зв'язки для розміщення їх на робочому просторі. Містить також області Device-Type

*Device-Type Selection Box* – область містить доступні типи пристроїв і зв'язків в Packet Tracer. Область Device-Specific Selection вказується, залежно від обраного пристрою

*Device-Specific Selection Box* – область використовується для вибору пристроїв і з'єднань, необхідних для побудови в робочому просторі мережі.

*User Created Packet Window* – вікно, що керує пакетами, створеними в мережі під час моделювання.

Для створення топології необхідно обрати пристрій з панелі Network Component, а потім з панелі Device-Type Selection обрати тип пристрою. Після цього – натиснути ліву кнопку миші в полі робочої області програми (Workspace). Також можна перемістити пристрій безпосередньо з області Device-Type Selection – обирається модель пристрою за замовчуванням.

Для швидкого створення декількох екземплярів одного пристрою потрібно, утримуючи кнопку Ctrl, натиснути на пристрій в області Device-Specific Selection і відпустити кнопку Ctrl. Після цього можна кілька разів натиснути на робочій області для додавання копії пристрою.

У Packet Tracer представлені наступні типи пристроїв:

маршрутизатори;

комутатори та мости;

хаби і повторювачі;

кінцеві пристрої – ПК, сервери, принтери, IP-телефони;

бездротові пристрої – точки доступу та бездротові маршрутизатори;

решта пристроїв – хмара, DSL-модем і кабельний модем.

Для прикладу, додамо необхідні типи пристроїв у робочу область програми, як показано на рис. 1.32.

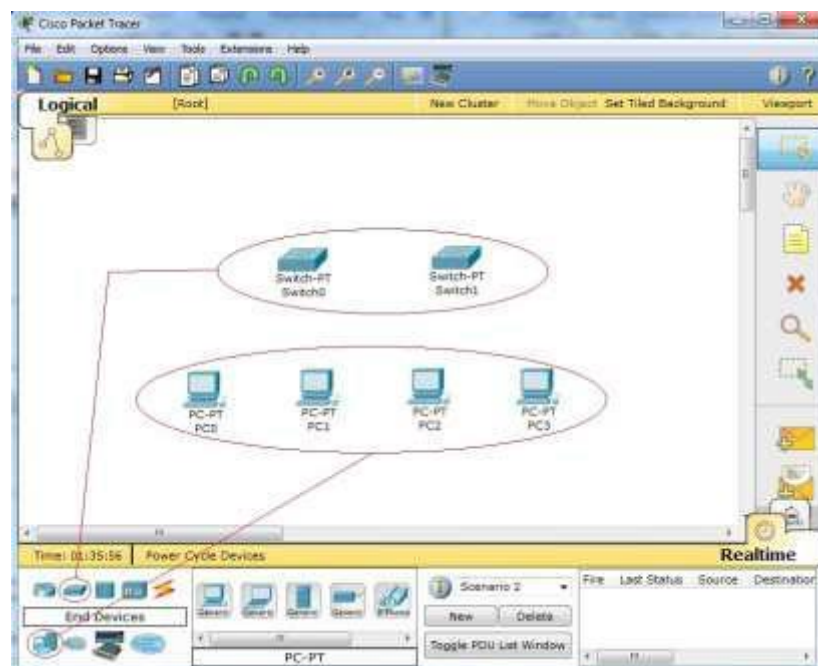


Рис. 1.32. Додавання пристроїв

При додаванні пристрою, користувач може відкрити діалогове вікно пристрою (для цього на пристрої клікають лівою кнопкою миші). Діалогове вікно властивостей має дві вкладки:

*Physical* – містить графічний інтерфейс пристрою і дозволяє моделювати роботу з ним на фізичному рівні;

*Config* - містить всі необхідні параметри для настройки пристрою і має зручний для цього інтерфейс.

вкладці *Config* користувач може присвоїти ім'я пристрою та встановити його параметри.

Також, в залежності від пристрою, властивості можуть мати додаткову вкладку для керування роботою обраного елемента: *Desktop* (якщо обрано кінцевий пристрій), або *CLI* (якщо обрано маршрутизатор). Видалення пристрою з робочою областю відбувається кнопкою *Delete (Del)*.

З'єднаємо додані пристрої за допомогою зв'язків. Для цього оберемо вкладку *Connections* з панелі *Network Component Box*, що показана на рис. 1.33. На вкладці зображено всі можливі типи з'єднань між пристроями. Оберемо відповідний тип кабелю. Показчик миші зміниться на курсор "Connection" – у вигляді роз'єму. Натиснемо на перший пристрій та оберемо інтерфейс, з яким потрібно виконати з'єднання, а потім натиснемо на другий пристрій та виконаємо аналогічну операцію. Пристрої можна також з'єднати за допомогою *Automatically Choose Connection Type* (автоматично з'єднує елементи в мережі). Після натискання на кожному з пристроїв, які потрібно з'єднати, між пристроями з'явиться кабельне з'єднання, а індикатори на кожному кінці покажуть статус з'єднання (для інтерфейсів, що мають індикатори).



Рис. 1.33. Вкладка *Connections* з можливими з'єднаннями

Cisco Packet Tracer підтримує ряд мережевих з'єднань, що показані в табл.



1.4. Кожен тип кабелю може бути з'єднаний лише з певними типами інтерфейсів.

Після створення мережі її топологію потрібно зберегти, обравши пункт меню File -> Save або іконку Save на панелі Main Tool Bar. Файл збереженої топології має розширення \*.pkt.


Packet Tracer дає можливість моделювати роботу з інтерфейсом командного рядка (ІКР) операційної системи IOS, встановленої на всіх комутаторах і маршрутизаторах компанії Cisco.

*Таблиця 1.4. Мережеві з'єднання в*  
**Packet Tracer**

Тип кабелю	Опис
Console	Консольне з'єднання, що може бути між ПК і маршрутизаторами, або комутаторами. Повинні бути виконані деякі вимоги для роботи консольного сеансу з ПК: швидкість з'єднання з обох сторін – однакова, має бути 7 біт даних (або 8 біт) для обох сторін, контроль парності – однаковий, має бути 1 або 2 степових біта (але вони не обов'язково повинні бути однаковими), потік даних може бути довільний для обох сторін.
Copper Straight-	Цей тип кабелю є стандартним середовищем передачі Ethernet для з'єднання пристроїв, які функціонують на різних рівнях OSI. Має бути з'єднаний з наступними типами портів:

through	 <p>Мідний 10 Мбіт/с (Ethernet), мідний 100 Мбіт/с (Fast Ethernet) і мідний 1000 Мбіт/с (Gigabit Ethernet).</p>
 <p>Copper Cross-over</p>	<p>Еthernet</p> <p>Цей тип кабелю є середовищем передачі даних для з'єднання пристроїв, які функціонують на одному рівні моделі OSI. Він може бути з'єднаний з наступними типами портів: мідний 10 Мбіт/с (Ethernet), мідний 100 Мбіт/с (Fast Ethernet) і мідний 1000 Мбіт/с (Gigabit Ethernet).</p>
<b>Тип кабелю</b>	<b>Опис</b>
Fiber	Оптоволоконне середовище використовується для з'єднання між оптичними портами (100 Мбіт/с або 1000 Мбіт/с).
Phone	З'єднання через телефонну лінію може бути здійснено між пристроями, що мають модемні порти. Стандартне уявлення модемного з'єднання - це кінцевий пристрій (наприклад, ПК), що додзвонюється в мережеву хмару.
Coaxial	Коаксіальний кабель використовується для з'єднання між коаксіальними портами, такими як кабельний модем, з'єднаний з хмарою Packet Tracer.
	З'єднання через послідовні порти, які використовуються для зв'язків WAN. Для настройки таких з'єднань необхідно



	<p>установити синхронізацію на стороні DCE-пристрою. Синхронізація DTE виконується за вибором. Сторону DCE можна визначити за іконкою «годинник» поруч з портом. При виборі типу з'єднання Serial DCE перший пристрій, до якого застосовується з'єднання, стає DCE-пристроєм, а другий, автоматично, – стороною DTE. Можливо також зворотнє розташування сторін, якщо обраний тип з'єднання Serial DTE.</p>
---	---

Підключившись до пристрою, ми можемо працювати з ним як з консоллю реального пристрою. Packet Tracer забезпечує підтримку практично всіх команд, доступних на реальних пристроях.

Підключення до ІКР комутаторів, або маршрутизаторів, можна зробити, натиснувши на пристрій і перейшовши через вікно властивостей до вкладки CLI.

Для відтворення роботи командного рядка на кінцевому пристрої необхідно у властивостях обрати вкладку Desktop та натиснути на ярлик Command Prompt.

### ***Робота з файлами в Packet Tracer***

Packet Tracer має можливість зберігати в текстових файлах конфігурацію ряду пристроїв, зокрема маршрутизаторів та комутаторів. Для цього відкрити властивості пристрою і у вкладці Config натиснути кнопку "Export ..." для експорту конфігурації Startup Config, або Running Config. Отримаємо діалогове вікно для збереження необхідної конфігурації в файл, який буде мати розширення \*.txt. Текст файлу з конфігурацією пристрою running-config.txt (ім'я за замовчуванням) аналогічний тексту інформації отриманої з використанням команди show running-config в IOS- пристроях.

Конфігурація кожного пристрою зберігається в окремому текстовому

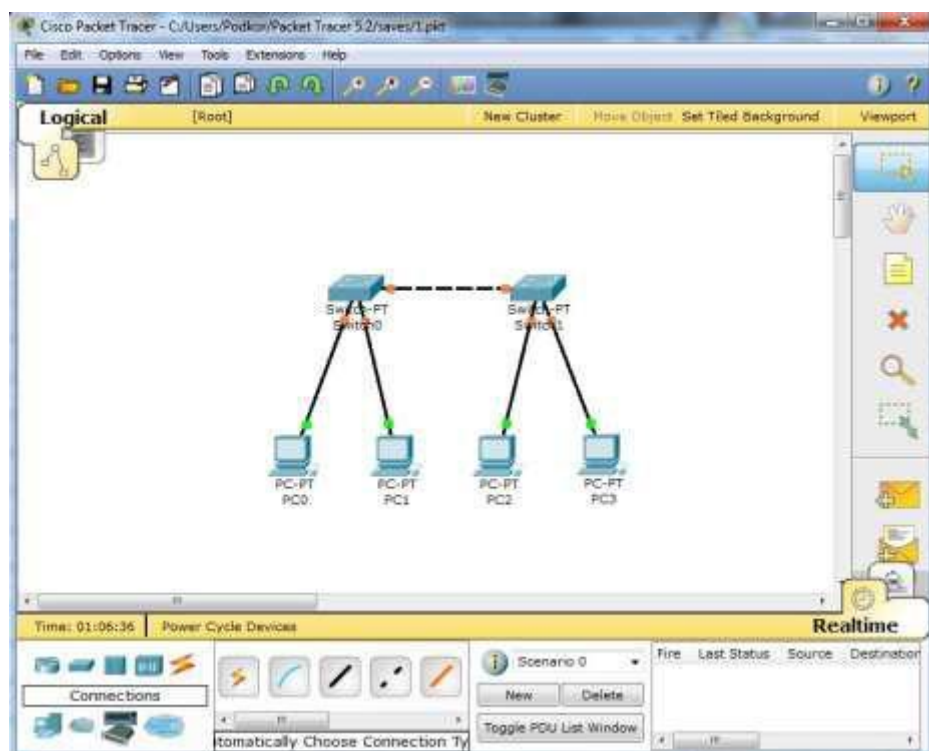
файлі. Користувач може змінювати конфігурації в збереженому файлі за допомогою текстових редакторів. Для завантаження до пристрою збережених, або відредагованих налаштувань потрібно у вкладці Config натиснути кнопку "Load ..." для завантаження необхідної конфігурації Startup Config, або кнопку "Merge ..." для завантаження конфігурації Running Config.

### **Приклад виконання**

Розмістимо в робочій області два комутатора Switch-PT. За замовчуванням – Switch0 і Switch1 та додамо до них чотири комп'ютери з іменами PC0, PC1, PC2, PC3. З'єднаємо пристрої в мережу Ethernet, як показано на рис. 1.34. Збережемо створену топологію, натиснувши кнопку Save (в меню File).

Відкриємо властивості пристрою PC0, клікнувши по його зображенню. Перейдемо до вкладки Desktop і промодельємо його роботу, натиснувши Comand Prompt.

Список команд отримаємо після вказання «?» і натискання Enter. Для конфігурації комп'ютера скористаємося командою `ipconfig`, яку запустимо в командному рядку: `ipconfig 192.168.1.2 255.255.255.0`



*Рис. 1.34. Експериментальна модель мережі*

ІР-адресу і маску мережі також можна вводити в графічному інтерфейсі пристрою, як показано на рис. 1.35.

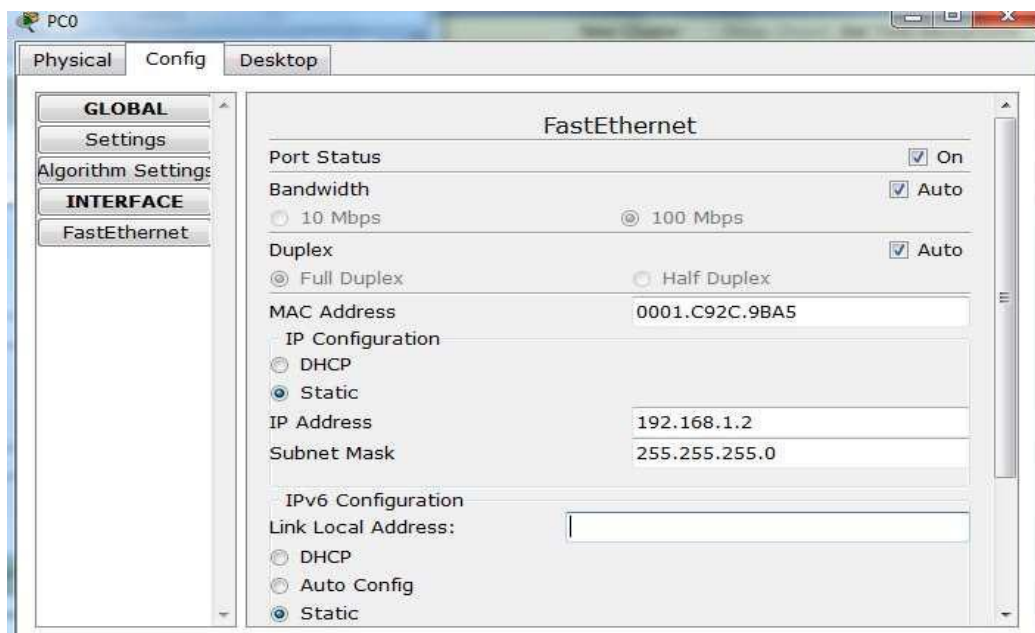


Рис. 1.35. Налаштування пристрою

Поле DEFAULT GATEWAY – адреса шлюзу. В даному випадку не заповнюється, оскільки дана топологія мережі не потребує маршрутизації. Налаштуємо кожен з пристроїв, відповідно до табл. 1.5. У кожного з комп'ютерів переглянемо призначені адреси командою `ipconfig` без параметрів.

Таблиця 1.5 Налаштування комп'ютерів підмережі

Пристрій	IP-address	Subnet mask
PC0	192.168.1.2	255.255.255.0
PC1	192.168.1.3	255.255.255.0
PC2	192.168.1.4	255.255.255.0

PC3	192.168.1.5	255.255.255.0

Packet Tracer передбачено режим моделювання з описанням роботи утиліти ping. Перейдемо в даний режим, натиснувши на однойменний значок

нижньому лівому кутку робочої області, або комбінацію клавіш Shift+S.

При цьому відкриється панель моделювання, показана на рис. 1.36, на якій відобразатиметься виконання ping-процесу.

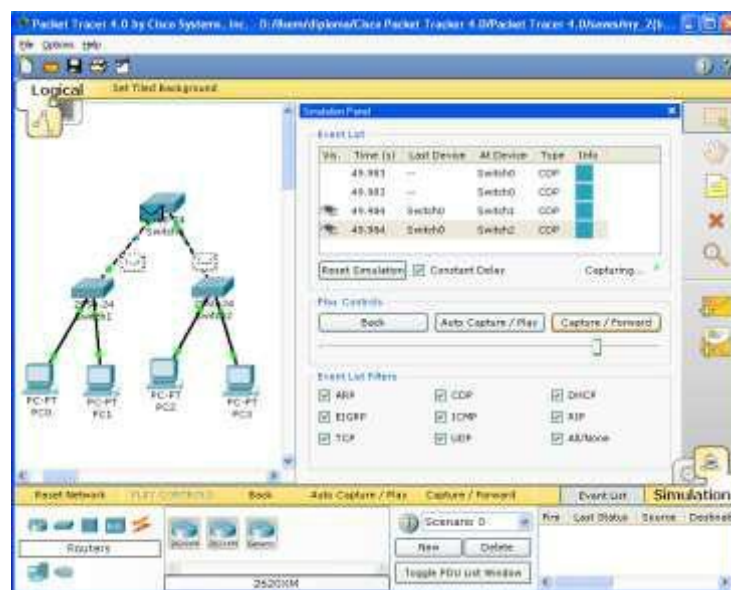


Рис. 1.36. Панель моделювання

Запустимо ping-процес та перейдемо до вкладки «Панель моделювання». На топології спроектованої мережі спостерігатимемо рух пакетів (PDU).

Кнопка «Автоматично» моделює весь ping-процес, а «Покроково» відображає його поетапно. Моделювання припиняється при завершенні ping-процесу, або при закритті вікна «Редагування» відповідного пристрою.

Якщо мережа працює, команда ping пересилатиме пакети на будь-який комп'ютер, як показано на рис. 1.37.

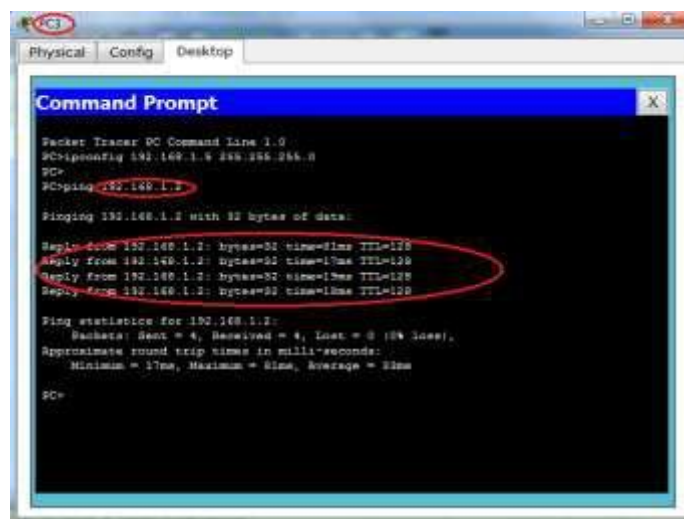


Рис. 1.37. Виконання команди ping в командному рядку

Режим моделювання «Simulation» дозволяє відслідковувати протоколи передачі даних та контролювати рівень моделі OSI, на якому даний протокол задіяний. Інформація про пакет та його структуру відображається натисканням правої кнопки миші на кольоровий квадрат в графі «Інформація», як показано на рис. 1.38.

### Тестові завдання

Створіть топологію мережі, яка показана вище – на рис. 1.36. Кількість комп'ютерів визначається як 4+номер варіанту.

Відповідно до табл. 1.5, призначте всім комп'ютерам IP-адреси з масками.

Запустіть утиліту ping і перевірте стабільність з'єднання всіх об'єктів мережі за протоколом IP.

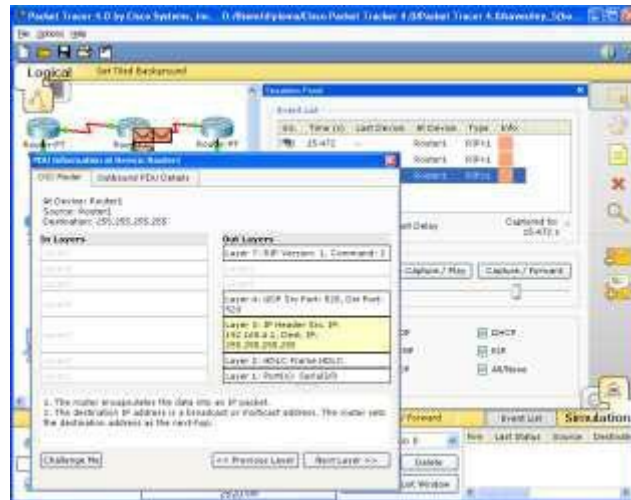


Рис. 1.38. Аналіз моделі OSI в Cisco Packet Tracer.

Переключіться в «Режим моделювання» і виконайте команду Ping з одного комп'ютера на інший. Поясніть обмін даними по протоколу ICMP між пристроями та роль протоколу ARP при цьому.

## 6. Передача повідомлень між клієнтом та сервером на базі TCP-протоколу

Мета роботи: реалізувати в середовищі C++ на базі функцій бібліотеки

WinSock клієнт-серверний додаток для обміну повідомленнями в рамках протоколу TCP для ОС Windows.

## Теоретичні відомості

WinSock або Windows socket – це інтерфейс прикладного програмування (API), створений для реалізації додатків в мережі на основі протоколу TCP/IP. Для роботи використовується WSOCK32.DLL. Ця бібліотека знаходиться в теці \System32 системного каталогу Windows.

Існують дві версії WinSock:

WinSock 1.1 - підтримує тільки протокол TCP/IP;

WinSock 2.0 - дозволяє створювати незалежні від транспортних протоколів додатки, що працюють з TCP/IP (Transmission Control

Protocol/Internet Protocol), IPX/SPX (Internetwork Packet

Exchange/Sequenced Packet Exchange), NETBEUI (NetBios Extended User Interface).

WinSock виділяє функції трьох типів, необхідних для створення додатку, які наведено в таблицях 1.6-1.8:

функції Берклі (блокуючі, що зупиняють роботу програми до свого завершення, і неблокуючі, що виконуються паралельно з програмою).

Ініціалізації і деініціалізації бібліотеки.

Інформаційні (отримання інформації про найменування доменів, служби, протоколи Internet).

*Таблиця 1.6. Функції Берклі (блокуючі)*

Назва функції	Призначення
Асепт	Створює новий сокет і підключає його до віддаленого комп'ютера

Closesocket	Закриває одну з сторін з'єднання
Connect	Ініціалізує з'єднання з боку вказаного сокета
Recv	Приймає дані від підключеного сокета
Recvfrom	Приймає дані від підключеного або непідключеного сокета
Send	Посилає дані підключеному сокету
Sendto	Посилає дані підключеному або непідключеному сокету

*Таблиця 1.7. Функції Берклі (неблокуючі)*

<b>Назва функції</b>	<b>Призначення</b>
Bind	Зв'язує віртуальний сокет з фізичним
Inetaddr	Конвертує рядок адреси в значення, яке можна використовувати в структурі in_addr
Ioctlsocket	Управляє параметрами сокета
Listen	Переводить сокет в режим прослуховування порта.
Socket	Створює точку з'єднання



Таблиця 1.8. Функції ініціалізації і деініціалізації бібліотеки WinSock

Назва функції	Призначення
WSACleanup	Завершує роботу з WinSock DLL
WSAGetLastError	Отримує інформацію про останню помилку
WSASetLastError	Встановлює повернення після помилки
WSAStartup	Ініціалізує WinSock DLL

#### Схема взаємодії функцій WinSock

Для реалізації поставленого завдання необхідно створити клієнтський і серверний додаток. Вони різні по організації, але є загальні дії, необхідні як для клієнтської, так і для серверної частин. Схема взаємодії функцій WinSock показана на Рис. 1. У блоці 1 (рис. 7.1) відображені загальні дії сервера і клієнта, в блоках 2 і 3, відповідно, дії сервера і клієнта, а в блоці 4 – їх взаємодія.

#### Блок 1 (Загальні дії)

роботі реалізовано програмний продукт у вигляді клієнт-серверного додатку. Сервер буде працювати послідовно з встановленням логічного з'єднання на базі протоколу TCP.

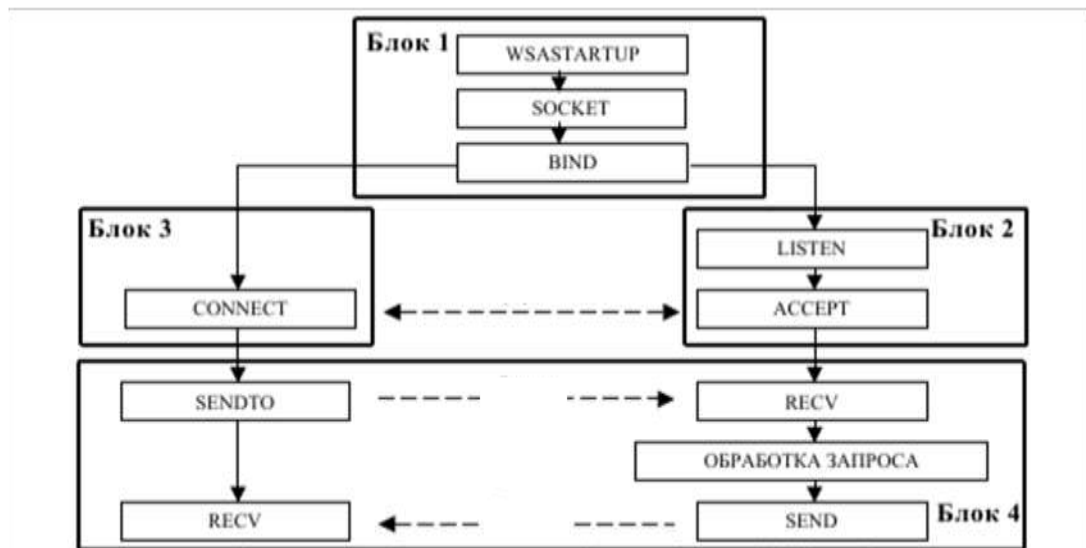
Узагальнений алгоритм роботи послідовного сервера зі встановленням логічного з'єднання на базі протоколу TCP показано на рис. 1.39 та полягає наступному:

Сервер створює сокет, що містить IP адресу та порт .

Сервер переводить створений сокет в режим прослуховування для приймання послідовних запитів від клієнтів.

Сервер приймає запит від певного клієнта на встановлення з'єднання, надсилає підтвердження клієнту та отримує підтвердження з'єднання від клієнта.

Сервер отримує запити від клієнта, опрацьовує їх, та за необхідності надсилає відповіді клієнту.



*Створення віртуального каналу*

*Запит*

*Відгук*

Рис. 1.39. Схема взаємодії функцій WinSock

Після завершення обміну даними із конкретним клієнтом з'єднання закривається. А сам сервер повертається до третього пункту і очікує подальших запитів від клієнтів.

Для використання функцій WinSock 2.x, необхідно в початковий текст програми включити директиву `#include <winsock2.h>` і підготувати її до роботи функцією `WSAStartup`. При вдалому завантаженні бібліотеки, потрібно створити сокет, використовуючи функцію `Socket` і асоціювати сокет

адресною структурою `sockaddr_in`, яка містить інформацію про протокол з'єднання, IP-адреса і порт ПК.

## Блок 2 (Реалізація серверної частини)

Після створення серверного сокета, він прослуховує порт функцією `Listen`, тобто перевіряє його на предмет запиту від клієнта. Після надходження запиту, сервер обробляє його функцією `Accept`.

## Блок 3 (Реалізація клієнтської частини)

Після створення клієнтського сокета, він посилає запит на підключення до сервера, використовуючи функцію `Connect` і вказавши в якості одного з

параметрів IP-адресу сервера. Ця функція є блокуючою, тобто виконання програми припиниться до тих пір, поки не прийде відповідь від сервера. При позитивній відповіді сокет підключається до сервера і може з ним взаємодіяти.

## Блок 4 (Реалізація обміну даними)

Процес взаємодії між клієнтом і сервером зводиться до відправки і прийому повідомлень. Для відправки повідомлення використовується функція `Send`, або `Sendto`. Їх відмінність полягає в тому, що для функції `Send` необхідне

з'єднання (Connect, Accept), для Sendto воно необов'язкове. Для прийому повідомлень застосовують функції Recv або Recvfrom. Для функції Recvfrom з'єднання (Connect, Accept) також необов'язкове.

## **Бібліотека функцій WinSock**

### **ІНІЦІАЛІЗАЦІЯ WINSOCK**

Функція WSStartup ініціалізує бібліотеку WinSock. Вона завжди розташовується першою на початку роботи з WinSock. Її опис:

```
int WSStartup (WORD wVersionRequested, LPWSADATA lpWSADATA).
```

Перший параметр – це версія, що використовується. Молодший байт основна версія, старший байт розширення версії. Якщо ініціалізація відбулася, то повернеться нульове значення. Ініціалізація полягає в

зіставленні номера версії і реально існуючої DLL в системі.

Другий параметр - це вказівник на структуру WSADATA, в яку повернуться параметри ініціалізації:

```
typedef struct WSADATA {  
  
    WORD wVersion;  
  
    WORD wHighVersion;  
  
    char szDescription[WSADESCRIPTION_LEN+1]; char  
szSystemStatus[WSASYS_STATUS_LEN+1]; unsigned short iMaxSockets;  
unsigned short iMaxUdpDg;  
    char FAR * lpVendorInfo;
```

```
}
```

Оператор `WSAGetLastError` видає код помилки при негативному результаті.

Оператор `WSACleanup()` завершує звернення до функцій WinSock. При вдалому виконанні повернеться нуль.

Приклад ініціалізації бібліотеки WinSock:

```
#include <stdio.h>
```

```
#include <winsock2.h>
```

```
#include <windows.h>
```

```
int main()
```

```
{
```

```
//Ініціалізація бібліотеки сокетів
```

```
WSADATA MyWSADATA;
```

```
int ErrWSADATA;
```

```
ErrWSADATA = WSASStartup((0x0202), (WSADATA *) &MyWSADATA);
```

```
if (ErrWSADATA != 0)
```

```

{

printf("Error WSAStartup %d\n", WSAGetLastError());

return -1;

}

printf("WSAStartup OK!");

}

```

**СТВОРЕННЯ І ВИДАЛЕННЯ СОКЕТА** Функція створення сокета має наступний вигляд: `SOCKET socket(int af, int type, int protocol)`, де

`af` – характеризує набір протоколів, в рамках якого взаємодіятимуть клієнт і сервер (це може бути TCP/IP, UDP, IPX і т. д.). Для протоколу TCP/IP параметр `af` повинен бути рівний `AF_INET`, що відповідає формату адреси, прийнятому в Internet.

`type` – визначає тип комунікацій (`SOCK_STREAM`, і `SOCK_DGRAM`). Якщо параметр рівний `SOCK_STREAM`, то сокет буде використаний для передачі даних через канал зв'язку з використанням протоколу TCP/IP. Якщо

використовується `SOCK_DGRAM`, то передача даних виконуватиметься без створення каналів зв'язку через датаграмний протокол UDP.

`protocol` – задає код конкретного протоколу з вказаного набору

(заданого `af`), який буде реалізований в даному з'єднанні. Протоколи позначаються символьними константами з префіксом `IPPROTO_` (наприклад,

IPPROTO\_IP або IPPROTO\_UDP). Допускається значення protocol=0 (протокол не вказаний), в цьому випадку використовується значення за умовчанням для даного виду з'єднань.

Параметр, що повертається, є дескриптором сокета.

Якщо операція socket завершилася успішно, вихідний параметр рівний дескриптору сокета, інакше - INVALID\_SOCKET (-1). За допомогою оператора WSAGetLastError можна одержати код помилки, що прояснює причину негативного результату.

Знищує сокет функція closesocket(SOCKET s), де s – змінна типу SOCKET, одержана в результаті виклику функції socket.

Приклад створення сокета:

```
SOCKET MySocket;
```

```
MySocket = socket(AF_INET, SOCK_STREAM, IPPROTO_IP);
```

```
if (MySocket < 0)
```

```
{
```

```
printf("Error WSAStartup %d\n", WSAGetLastError());
```

```
WSACleanup();
```

```
return -1;
```

```
}
```

## ПРИВ'ЯЗКА АДРЕСИ ДО СОКЕТА

Перш, ніж сервер зможе використовувати сокет, він повинен зв'язати його з локальною адресою, що складається з IP-адреси і номеру порту. Якщо сервер має декілька IP-адрес, то сокет може бути зв'язаний зразу з усіма - для цього як IP-адресу слід вказати константу `INADDR_ANY` рівною нулеві, або конкретну IP-адресу.

Сервер включає в себе певний порт, тому необхідно його прописувати (зв'язувати) вручну. Клієнт теж повинен зв'язувати сокет з конкретною адресою перед його використанням, проте за нього це робить функція `Connect`, асоціюючи сокет з одним з портів, випадково вибраних з діапазону 1024-5000.

Функція зв'язування сокета з фізичною адресою має вид:

`int bind(SOCKET s, const struct sockaddr FAR* name, int namelen)` де `s` – цілочисельний код дескриптора;

`name` – містить три величини: IP-адреса, код протокольного набору, номер порту, який визначає характер додатку;

`namelen` – визначає довжину другого параметра.

Структура адресної інформації має вигляд:

```
struct sockaddr_in{
```

```
short sin_family; /* Вказується протокол*/
```



```
unsigned short sin_port; /* Вказується порт*/

struct inaddr sinaddr; /* Вказується IP-адреса*/

char sin_zero[8];

}
```

У серверній частині додатку IP-адресу можна зробити рівним INADDR\_ANY (або =0), оскільки серверу не обов'язково знати свою IP-адресу. При коректному виконанні функція bind повертає код 0, інакше SOCKET\_ERROR=-1.

Приклад прив'язки адреси до сокета:

//Зв'язування сокета

```
sockaddr_in LocalAddr;
```

```
int ErrBind;
```

```
LocalAddr.sin_family = AF_INET;
```

```
LocalAddr.sin_port = htons(1234); //Функція забезпечує мережевий порядок
```

```
LocalAddr.sin_addr.s_addr = INADDR_ANY;
```

```
ErrBind = bind(MySocket (sockaddr *) &LocalAddr, sizeof(LocalAddr));
```

```
if (ErrBind != 0)
```

```
{
```

```
printf("Error bind %d\n", WSAGetLastError()); closesocket(MySocket);  
WSACleanup();  
  
return -1;  
  
}
```

### **Очікування і обробка запитів на підключення клієнта**

Для підключення сокета, його необхідно перевести в стан очікування функцією Listen:

`int listen(SOCKET s, int backlog)`, де

`backlog` - задає максимальний розмір черги для запитів, тобто скільки запитів може бути прийняте на обслуговування без втрат. Очікуючий сокет надсилає кожному відправнику повідомлення-відгук, яке підтверджує отримання запиту на з'єднання.

Запити з черги, сформованої функцією Listen, обробляються функцією

Accept, що встановлює зв'язок з сокетом клієнта:

`int Accept(SOCKET s, struct sockaddr FAR *addr, int FAR* addrlen)`, де `s` - дескриптор сокета, який прослуховує з'єднання (той, що і в Listen); `addr` - вказівник на структуру, яка містить адресу; `addrlen` - вказівник на довжину адреси `addr`.

При виникненні помилки повертається код INVALID\_SOCKET.

Приклад використання функцій Listen і Accept:

```
//Переведення сервера в режим очікування запитів int ErrListen;
```

```
ErrListen=listen(MySocket,5); //Довжина черги 5
```

```
if (ErrListen != 0)
```

```
{
```

```
printf("Error listen %d\n", WSAGetLastError());
```

```
closesocket(MySocket);
```

```
WSACleanup();
```

```
return -1;
```

```
}
```

```
//Вибір клієнта з черги
```

```
SOCKET AcceptSocket;
```

```
sockaddr_in AcceptAddr;
```

```
int SizeAcceptAddr=sizeof(AcceptAddr);
```

```
AcceptSocket=accept(MySocket, (sockaddr *) &AcceptAddr,
```

```
&SizeAcceptAddr);
```

```
if (AcceptSocket !=INVALID_SOCKET)
```

```
{
```

```
printf("Error accept %d\n", WSAGetLastError());
```

```
return -1;
```

```
}
```

## ПІДКЛЮЧЕННЯ ДО СЕРВЕРА

Клієнт для з'єднання з сервером використовує функцію Connect:

```
int connect(SOCKET s, const struct sockaddr FAR* name, int namelen),
```

де

s – дескриптор сокета;

name – ідентифікатор адреси місця призначення (показчик на

структуру даних);

namelen – довжина цієї адреси.

Таким чином, функція Connect повідомляє IP-адресу і номер порту віддаленої машини. Якщо адресне поле структури name містить нулі, Connect поверне помилку WSAEADDRNOTAVAIL (або SOCKET\_ERROR=-1).

Приклад організації підключення до сервера:

```
//Підключення до сервера
```

```
#define PORT 1234
```

```
#define SERVERADDR "127.0.0.1"
```

```
SOCKET ClientSocket ;
```

```
sockaddr_in ServerAddr;
```

```
int ErrConnect;
```

```
ServerAddr.sin_family=AF_INET;
```

```
ServerAddr.sin_port = htons(PORT);
```

```
ServerAddr.sin_addr.s_addr = inetaddr(SERVERADDR);
```

```
ErrConnect = connect(ClientSocket, (sockaddr *) &ServerAddr,  
sizeof(ServerAddr));
```

```

If (ErrConnect != 0)

{

printf("Error connect %d\n", WSAGetLastError());

return -1;

}

```

### **Відправка і прийом повідомлень**

Для відправки повідомлень використовується функція Send, або Sendto:

```
int Send (SOCKET s, const char FAR* buf, int len, int flags), де
```

s – дескриптор сокета на віддаленій машині;

buf – вказівник на масив символів, що підлягають пересилці; len – розмір другого параметра;

flags – служить для цілей діагностики і управління передачею даних.

Рекомендується прирівнювати його нулю.

Приклад розсилки повідомлень оператором Send:

```
//Пересилка даних у разі установки з'єднання
```

```
char Buff [255];
```

```

int ErrSend;

ErrSend = send(AcceptSocket, &Buff, sizeof(Buff), 0); if (ErrSend =
SOCKET_ERROR) {

    printf("Error send %d\n", WSAGetLastError()); closesocket(MySocket);
    WSACleanup();

    return -1;

}

```

Функція `sendto` служить для пересилки даних без установки з'єднання, тобто відправка даних йде без підтвердження отримання даних:

`int Sendto (SOCKET s, const char FAR* buf, int len, int flags, const struct sockaddr FAR* to, int tolen)`, де

`s` - дескриптор відправника;

`buf` - вказівник на масив даних, призначених для пересилки;

`len` - розмір другого параметра;

`flags` - служить для цілей діагностики і управління передачею даних; `to` - адресна структура сокета віддаленої машини;

`tolen` - розмір адресної структури сокета віддаленої машини. Приклад відправки повідомлень функцією `Sendto`: //Пересилка даних без установки

з'єднання

```
char Buff [255];

int ErrSendTo;

SOCKET MySocket;

sockaddr_in SendToAddr;

ErrSendTo = sendto(MySocket, &Buf, sizeof(Buf), 0, &SendToAddr,
sizeof(SendToAddr));

If (ErrSendTo = SOCKET_ERROR)

{

printf("Error sendto %d\n", WSAGetLastError());

closesocket(MySocket);

WSACleanup();

return -1;

}
```

Для прийому повідомлення застосовується функція Recv або Recvfrom. Функцію Recvfrom так само, як і Sendto, можна застосовувати, не з'єднуючись з відправником.



`int recv (SOCKET s, char FAR* buf, int len, int flags)`, де `s` - дескриптор сокета одержувача;

`buf` - вказівник на масив одержаних даних; `len` - розмір другого параметра;

`flags` - служить для цілей діагностики і управління передачею даних.

Приклад прийому повідомлень функцією `Krecv`:

`//Отримання даних у разі установки з'єднання`

`char Buff [255];`

`SOCKET MySocket`

`int ErrResv;`

`ErrResv = recv(MySocket, &Buff, sizeof(Buff), 0); if (ErrResv = SOCKET_ERROR) {`

`printf("Error send %d\n", WSAGetLastError()); closesocket(MySocket); WSACleanup();`

`return -1;`

`}`

`int Krecvfrom (SOCKET s, const char FAR* buf, int len, int flags, const struct sockaddr FAR* to, int tolen)`, де

s – дескриптор одержувача;

buf – покажчик на масив одержаних даних; len – розмір другого параметра;

flags – служить для цілей діагностики і управління передачею даних; to – адресна структура сокета віддаленої машини;

toLen – розмір адресної структури сокета віддаленої машини.

Приклад прийом повідомлень, використовуючи функцію Recvfrom:

Отримання даних без установки з'єднання

```
char Buff [255];
```

```
SOCKET MySocket;
```

```
int ErrRecvFrom;
```

```
sockaddr_in RecvFromAddr;
```

```
ErrSendTo = recvfrom (MySocket, &Buf, sizeof(Buf), 0, & RecvFromAddr,
sizeof(RecvFromAddr));
```

```
If (ErrRecvFrom = SOCKET_ERROR)
```

```
{
```

```
printf("Error sendto %d\n", WSAGetLastError());
```

```
closesocket(MySocket);
```

```
WSACleanup();
```

```
return -1;
```

```
}
```

### **Тестові завдання**

Створити програмний продукт, що працює як клієнт-серверний додаток

установкою логічного з'єднання за протоколом TCP. Завдання реалізується відповідно до варіанту, що наведений нижче.

Клієнт відправляє на сервер два цілі числа. Сервер перемножує ці числа, додає номер варіанта та відсилає відповідь клієнту.

Клієнт відправляє на сервер два числа. Сервер підносить перше число до степені, що рівна другому числу, і відправляє відповідь клієнту.

Клієнт відправляє на сервер слово. Сервер рахує кількість букв і відправляє число клієнту.

Клієнт відправляє на сервер слово. Сервер видаляє всі голосні та відправляє відповідь клієнту.

Клієнт відправляє на сервер слово. Сервер видаляє всі приголосні та відправляє відповідь клієнту.

Клієнт відправляє на сервер два слова. Сервер їх порівнює і відправляє букви що зустрічаються в обох словах клієнту.

Клієнт відправляє на сервер два числа і отримує від сервера True, якщо друге число ділиться на перше без залишку (або False – в іншому разі).

Клієнт відправляє на сервер слово, що містить пробіли. Сервер видаляє пробіли і повертає слово клієнту без пробілів.

Клієнт відправляє на сервер слово з букв і цифр. Сервер видаляє букви і повертає слово клієнту.

Клієнт відправляє на сервер слово із букв і цифр. Сервер видаляє цифри і повертає клієнту слово.

Клієнт відправляє на сервер слово. Сервер читає слово справа наліво і так повертає його клієнту.

Клієнт надсилає серверу число в двійковій системі від 0 до 1111.

Сервер конвертує число в десяткову систему та повертає клієнту.

Клієнт відправляє на сервер число в шістнадцятковій системі числення від 0 до FF. Сервер конвертує число в десяткову систему та повертає клієнту.

Клієнт відправляє на сервер час в годинах і хвилинах. Сервер отримує і визначає час, що пройшов від початку нового тисячоліття, в днях,

годинах, хвилинах – та повертає клієнту.

Клієнт надсилає на сервер строку, що містить зірочки. Сервер зірочки видаляє і повертає строку клієнту.

Клієнту надсилає на сервер строку написаною латиницею. Сервер транслітерує її в кирилицю і відправляє клієнту.

Клієнт надсилає на сервер число, написане арабськими цифрами.

Сервер записує число в римські цифри та відправляє клієнту.

Клієнт надсилає на сервер число, написане римськими цифрами.

Сервер записує число арабськими цифрами та відправляє клієнту.

## Практична робота №6

### ПІДКЛЮЧЕННЯ ДО ВІДДАЛЕНОГО СЕРВЕРА LINUX І НАЛАШТУВАННЯ МЕРЕЖЕВИХ ПАРАМЕТРІВ

#### Загальні відомості:

Для цієї лабораторної роботи знадобиться сервер або ПК з ОС Linux (бажано, щоб був Debian або Ubuntu). Можете скооперуватися з одногрупниками, використовуючи один комп'ютер або встановивши віртуальну машину.

#### Необхідні команди

• **ssh** - Забезпечує безпечне зашифроване з'єднання між двома хостами через небезпечну мережу. Це з'єднання може також використовуватися для термінального доступу, передачі файлів і для тунелювання інших програм. Графічні програми X11 можна також безпечно запускати через ssh з віддаленого місця.

• **ifconfig**- Видає інформацію про статус активних інтерфейсів. Якщо вказано один аргумент інтерфейсу, видається інформація тільки про умови цього інтерфейсу; якщо вказано один аргумент -а, видається інформація про стан всіх інтерфейсів, навіть відключених. Інакше команда конфігурує зазначений інтерфейс.

• **route** – Маніпулює таблицями маршрутизації IP ядра та показує наявні маршрути. Його основне призначення - встановити статичні маршрути до певних хостів або мереж через інтерфейс після налаштування програми ifconfig.

• **ping** - Дуже простий інструмент для діагностики мережі. Вона дозволяє перевірити доступний віддалений хост. Для цього потрібно перевірити, чи може відповісти на запит про використання протоколу ICMP.

• **dhclient** - Надає засоби для конфігурування одного або декількох мережевих інтерфейсів за допомогою протоколу динамічної конфігурації хоста, протоколу BOOTP або якщо ці протоколи виходять з ладу, шляхом статичного призначення адреси.

• **dig** - Утиліта Unix-систем, що надає користувачеві інтерфейс командного рядка для звернення до системи DNS. Dig - це скорочення від «domain information groper». Дозволяє задавати різні типи запитів і запитувати довільно вказувані сервера. Має більше можливостей ніж подібна утиліта nslookup.

• **netstat** - Службова комп'ютерна програма, призначена для відображення поточного статусу підключень (вхідних та вихідних) по TCP/IP чи UDP, таблиць маршрутизації, кількості мережевих адаптерів та статистики протоколів.

• **ufw** - Інструмент налаштування брандмауера за умовчанням. Розроблений для полегшення конфігурації брандмауера iptables, ufw забезпечує зручний спосіб створення IPv4 або IPv6-хост-брандмауера. За замовчуванням UFW вимкнено.

• **nano** - Консольний текстовий редактор.

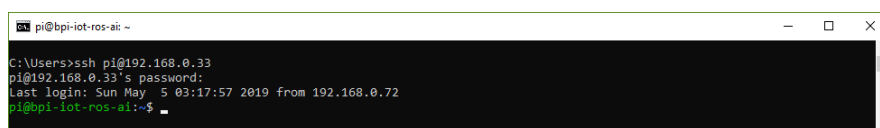
• **scp** - Дозволяє безпечно переносити файли між локальним хостом і віддаленим хостом або між двома віддаленими хостами. Він використовує ту саму аутентифікацію та безпеку, що й протокол Secure Shell (SSH), з якого він базується.

• **Iptables** - Інструмент адміністрування для фільтрації пакетів IPv4 і NAT.

### Хід виконання робіт:

• Під'єднатися до серверу

Команда: *ssh /user/@/IP/*



```

pi@bpi-iot-ros-ai ~
C:\Users>ssh pi@192.168.0.33
pi@192.168.0.33's password:
Last login: Sun May 5 03:17:57 2019 from 192.168.0.72
pi@bpi-iot-ros-ai:~$

```

**Приклад**

• Визначаємо мережеві дані

Команда: *ifconfig, route, ip route*

### Приклад

```

pi@bpi-iot-ros-ai:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 52:1a:f4:87:bb:12
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:510 (510.0 B)
          Interrupt:114

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:21262207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21262207 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3170376548 (3.1 GB)  TX bytes:3170376548 (3.1 GB)

wlan0     Link encap:Ethernet  HWaddr b8:f1:ec:0c:c4:6e
          inet addr:192.168.0.33  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::b2f1:ecff:fe0c:c46e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5724939 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4709303 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1036477310 (1.0 GB)  TX bytes:856070505 (856.0 MB)

pi@bpi-iot-ros-ai:~$

```

```

pi@bpi-iot-ros-ai:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default * 0.0.0.0 UG 0 0 0 wlan0
192.168.0.0 * 255.255.255.0 U 0 0 0 wlan0

pi@bpi-iot-ros-ai:~$ ip route
default via 192.168.0.1 dev wlan0
192.168.0.0/24 dev wlan0 proto kernel scope link src 192.168.0.33

pi@bpi-iot-ros-ai:~$

```

- Змінюємо IP та маску підмережі певного мережевого інтерфейсу  
Команда: *sudo ifconfig /Інтерфейс/ /IP/ netmask /Маска підмережі/*

### Приклад

```

pi@bpi-iot-ros-ai:~$ sudo ifconfig wlan0 192.168.0.33 netmask 255.255.255.0
pi@bpi-iot-ros-ai:~$

```

- Перевірити з'єднання або дізнатися IP іншого серверу  
Команда: *ping /IP/*

```

pi@bpi-iot-ros-ai:~$ ping google.com
PING google.com (172.217.20.174) 56(84) bytes of data:
64 bytes from waw02s07-in-f14.1e100.net (172.217.20.174): icmp_seq=1 ttl=56 time=16.9 ms
64 bytes from waw02s07-in-f14.1e100.net (172.217.20.174): icmp_seq=2 ttl=56 time=95.4 ms
64 bytes from waw02s07-in-f14.1e100.net (172.217.20.174): icmp_seq=3 ttl=56 time=24.1 ms
64 bytes from waw02s07-in-f14.1e100.net (172.217.20.174): icmp_seq=4 ttl=56 time=17.1 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 16.959/38.433/95.402/33.018 ms
pi@bpi-iot-ros-ai:~$

```

Приклад

- Відновити DHCP конфігурацію  
Команда: *sudo dhclient -r /Interface/*

### Приклад

```

pi@bpi-iot-ros-ai:~$ sudo dhclient wlan0 -r
killed old client process
/etc/resolvconf/update.d/libc: Warning: /etc/resolv.conf is not a symbolic link to /run/resolvconf/resolv.conf
pi@bpi-iot-ros-ai:~$

```

- Відтворення DNS інформації про певний домен  
Команда: *dig /IP/ any*

### Приклад

```

pi@bpi-iot-ros-ai:~$ dig kpi.ua any
; <<>> DiG 9.10.3-P4-Ubuntu <<>> kpi.ua any
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 33385
;; flags: qr rd ra; QUERY: 1, ANSWER: 8, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;kpi.ua.
      IN      ANY

;; ANSWER SECTION:
kpi.ua.      3599    IN      SOA      ns.kpi.ua. domainmaster.kpi.ua. 2019050200 14400 3600 1209600 86400
kpi.ua.      3599    IN      NS       ns2.kpi.ua.
kpi.ua.      3599    IN      NS       ns.kpi.ua.
kpi.ua.      3599    IN      A        77.47.133.222
kpi.ua.      3599    IN      MX       10 mail0.kpi.ua.
kpi.ua.      3599    IN      MX       10 mail0.kpi.ua.
kpi.ua.      3599    IN      TXT      "v=spf1 mx ip:77.47.128.135 ip:77.47.128.136 -all"
kpi.ua.      3599    IN      TXT      "google-site-verification=UdsMKVzhK2VF1_xgeVHCu7vZbnpPtDoOgeUtIXnIeBA"

;; Query time: 78 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun May 05 04:46:57 CST 2019
;; MSG SIZE rcvd: 323

pi@bpi-iot-ros-ai:~$

```

## • Відтворення переліку портів

Команда: *netstat -ntulp*

### Приклад

```

pi@bpi-iot-ros-ai:~$ netstat -ntulp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5885            0.0.0.0:*               LISTEN      1404/app.js
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 :::8443                  :::*                    LISTEN      1410/index.js
udp        0      0 0.0.0.0:68              0.0.0.0:*               *
udp        0      0 192.168.0.33:123        0.0.0.0:*               *
udp        0      0 127.0.0.1:123           0.0.0.0:*               *
udp        0      0 0.0.0.0:123             0.0.0.0:*               *
udp6       0      0 fe80::b2f1:ecff:fe0:123 :::*                    *
udp6       0      0 :::123                   :::*                    *
udp6       0      0 :::123                   :::*                    *
pi@bpi-iot-ros-ai:~$

```

## • Вайтліснути певні порти файрволлом

Команда: *sudo ufw allow /Порт/, sudo ufw enable*

```

pi@bpi-iot-ros-ai:~$ sudo ufw status
Status: inactive
pi@bpi-iot-ros-ai:~$ sudo ufw allow 22
Skipping adding existing rule
Skipping adding existing rule (v6)
pi@bpi-iot-ros-ai:~$ sudo ufw allow 3000
Rules updated
Rules updated (v6)
pi@bpi-iot-ros-ai:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y/n)? y
Firewall is active and enabled on system startup
pi@bpi-iot-ros-ai:~$ sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
3000 ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
3000 (v6) ALLOW Anywhere (v6)
pi@bpi-iot-ros-ai:~$

```

Приклад

## • Відтворити tcp з'єднання шляхом передачі html файлу

Команда: *{ echo -ne "HTTP/1.0 200 OK\r\n\r\n"; cat /Файл; } | nc -l -p /Порт/*

### Приклад



```

pi@bpi-iot-ros-ai: ~/app
pi@bpi-iot-ros-ai:~/app$ { echo -ne "HTTP/1.0 200 OK\r\n\r\n"; cat helloworld.html; } | nc -l -p 3000
GET / HTTP/1.1
Host: 192.168.0.33:3000
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: ru,uk-UA;q=0.9,uk;q=0.8,en-US;q=0.7,en;q=0.6
Cookie: connect.sid=s%3ARd0nKAuE8LY5cu8AqXecjiJJ7CKbZnUW.Tixn4v%2FrKDwFBTxVKv1jAvSi6wFShwsSpqjICGbg7SY
pi@bpi-iot-ros-ai:~/app$

```

# Hello world!

- Форвардніть один порт на інший

Команда: `sudo sysctl net.ipv4.ip_forward=1, sudo iptables -t nat -A PREROUTING -p tcp --dport /Порт 1/ -j REDIRECT --to-port /Порт 2/`

## Приклад

```

pi@bpi-iot-ros-ai: ~/app
pi@bpi-iot-ros-ai:~/app$ sudo iptables -t nat -I OUTPUT -p tcp -o lo --dport 3000 -j REDIRECT --to-ports 4000
pi@bpi-iot-ros-ai:~/app$

```

## Завдання

- Спочатку треба під'єднатися до сервера за допомогою команди `ssh`. (Можливо треба сконфігурувати файл `/etc/ssh/sshd_config` для доступу з паролем, якщо ви не використовуєте `ssh` ключі).
- За допомогою `ifconfig` визначте MAC, IPv4, IPv6, маску підмережі та IP шлюзу підмережі. Визначте роути IP-адресації.
- За допомогою `ifconfig` змініть IP, який видавався DHCP на 192.168.181.198, маску підмережі та шлюз на той, який ви визначили у попередньому кроці.
- За допомогою `route` проаналізувати, які маршрути змінилися та чому. За допомогою команди `ping` перевірте, чи можете під'єднатися до зовнішньої мережі.
- Застосуйте команду `dhclient` з інтерфейсом вашої мережі, щоб відновити DHCP конфігурацію та отримати новий IP від системи. Можете переконатись, що IP змінився через команду `ipconfig`.
- За допомогою `dig` визначити IP-адресу `asu.kpi.ua`, визначити TTL та знайти не тільки "A" записи.
- За допомогою `netstat` визначити відкриті порти. Проаналізувати, що саме використовує кожен порт.

- За допомогою *ufw* зробити доступним тільки *ssh* порт (за умовчужанням 22) та порт 3000.

- Створіть деякий html файл *helloworld.html* (наприклад з допомогою *nano* або скопіюйте за допомогою *scp*) та запустіть команду `{ echo -ne "HTTP/1.0 200 OK\r\n\r\n"; cat helloworld.html; } | nc -l -p 3000`

- Перейдіть на 3000 порт сервера (наприклад, використовуючи свій браузер або команду *curl*; ви повинні побачити верстку вашого файлу.)

- Переконайтеся, що ви не можете зайти ззовні на сервер під іншим портом, ані ж 3000.

- Додайте 4000 порт у вайтліст файрволу. За допомогою *iptables* форвардніть порт 3000 на 4000. Таким чином, при запуску команди, що віддає html, цей файл повинен бути на 4000 порті, замість 3000. Перевірте IP маршрути.

### Основна література

1. Навчальний посібник з дисциплін “Комп’ютерні мережі” для студентів / Коган А. В., Роковий О. П., Алєнін. О. І. – Київ: КПІ, 2020. – 77 с.
2. Організація комп’ютерних мереж: підручник: для студ. / КПІ ім. Ігоря Сікорського ; Ю. А. Тарнавський, І. М. Кузьменко. — Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с.

### Допоміжна література

3. Жураковський Ю.П., Полторак П. Теорія інформації та кодування : Підручник. К.: Вища шк., 2011. 255 с.

### Інформаційні ресурси в Інтернеті

Закордонні електронні наукові інформаційні ресурси: European Library.  
Вільний доступ до ресурсів 47 Національних бібліотек Європи, Австралії, Білорусії, Великої Британії, Німеччини, бібліотека коледжу Лондонського університету.

<http://www.irbis-nbuv.gov.ua/>

<https://elibrary.kubg.edu.ua>