

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Р.Є.Рикалюк

ЛАБОРАТОРНИЙ ПРАКТИКУМ
з курсу
«КОМП'ЮТЕРНІ МЕРЕЖІ»

Львів ЛНУ 2017

Рикалюк Р.Є. Лабораторний практикум з курсу “Комп’ютерні мережі”.
Видавн. центр Львів. ун-ту, . - 112 с.

У практикумі представлено курс лабораторних робіт з локальних обчислювальних мереж, який побудований на використанні віртуальних машин. Лабораторні роботи охоплюють принципи та апаратні засоби створення комп’ютерних мереж, а також програмні налаштування мережі в різних операційних системах. Викладено методику роботи з віртуальними машинами та сформульовано завдання до проведення лабораторних робіт з курсів “Інформаційні мережі” та «Комп’ютерні мережі».

Для студентів факультету прикладної математики та інформатики.

Рецензент:.

Редактор

©Р.Є.Рикалюк, 2009-2017

Програма курсу “Комп’ютерні мережі” передбачає виконання лабораторних робіт з основ побудови локальних комп’ютерних мереж, функціонування протоколів та налаштування програмного забезпечення для організації передавання даних у мережах ЕОМ з використанням різних типів операційних систем. Практикум складається з десяти лабораторних робіт: однієї – присвяченої апаратним мережевим засобам; двох змішаних апаратно-програмних робіт по налаштуванню локальної мережі; двох лабораторних робіт спрямованих на ознайомлення з особливостями середовища керування віртуальними машинами та вивчення мережових налаштувань різних операційних систем, використовуючи віртуальні машини. Інші лабораторні роботи знайомлять студентів з роботою бездротових мереж, плануванням адресного простору локальної мережі та дають змогу дослідити роботу локальної мережі за допомогою спеціальних програм – сніфферів. Для реалізації мережних налаштувань запропоновано програмний продукт VMWare Workstation. За допомогою цього середовища управління віртуальними машинами можна створити віртуальну мережу і проводити там лабораторні роботи. Крім цього, можна налаштувати віртуальні машини так, щоб вони після вимкнення поверталися до попереднього стану, що автоматично звільняє адміністратора лабораторії від додаткової роботи по відновленню системних налаштувань на “піддослідних” комп’ютерах.

У процесі виконання цього циклу робіт студенти вивчають принципи роботи та методи передавання даних у мережах ЕОМ.

ЗАГАЛЬНІ МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Виконання лабораторних робіт спрямоване на практичне засвоєння студентами спеціальностей 080201, 080202, 080203 та 080204 теоретичного матеріалу лекційного курсу “Комп’ютерні мережі”.

Перш ніж допустити студента до виконання лабораторної роботи, викладач перевіряє його теоретичну підготовку з теми роботи, знання методики проведення лабораторної роботи та наявність оформленого звіту про попередню роботу.

Лабораторну роботу виконують згідно з планом, що наведений в інструкції. Для виконання окремих робіт студенти отримують індивідуальні завдання від викладача.

Звіти про виконання лабораторних робіт потрібно оформити у текстовому редакторі Microsoft Word (або аналогічному) окремими файлами з іменем:

“прізвище_”lan№лабораторної роботи”.doc.

У звіті необхідно подати:

- 1) назву та мету роботи;
- 2) хід виконання роботи з детальним описом методики її проведення;
- 3) копії екранів відповідних налаштувань;
- 4) висновки.

За результатами виконаної лабораторної роботи, які наведені у звіті, відбувається захист лабораторної роботи.

ЛАБОРАТОРНА РОБОТА №1

Тема: Вивчення мережних апаратних засобів та обладнання.

Мета роботи. Ознайомлення з основними апаратними засобами та обладнанням для створення локальної обчислювальної мережі. Лабораторна робота містить практичні вказівки по виконанню найбільш поширених завдань адміністратора мережі, наприклад – приєднання інтерфейсного роз'язку до мережевого кабелю.

Апаратні засоби та обладнання, що використовують у роботі:

1. Кабелі для організації з'єднань у мережах (коаксіальний, неекранована скручена пара, оптоволокно).
2. Пристрої з'єднання BNC, RJ-45, настінні і модульні розетки, термінатори.
3. Елементи ЛКМ: монтажні коробки, патч-панелі, патч-корди, абонентські шнури. Розділення кабелю UTP за стандартами TIA/EIA - 568 A/B.
4. Варіанти виконання активних концентраторів (хаби, комутатори, MAU).

Теоретичні відомості

Коаксіальні кабелі

На початку розвитку локальних мереж коаксіальний кабель як засіб передачі був найбільш поширений. Він використовувався і використовується переважно в мережах Ethernet і частково ARCnet. Розрізняють “товстий” і “тонкий” кабелі.

“Товстий Ethernet”, як правило, використовують так: прокладають по периметру приміщення або будівлі, і на його кінцях встановлюють 50-омні термінатори. Через свою товщину і жорсткість кабель не може бути під'єднаним безпосередньо до мережевої плати. Тому на кабель в потрібних місцях встановлюють “вампири” – спеціальні пристрої, що проколюють оболонку кабелю і під'єднуються до нього. “Вампір” настільки міцно сидить на кабелі, що після установки його неможливо зняти без спеціального інструменту. До “вампіра”, у свою чергу, під'єднують трансивер – пристрій,

що погоджує мережеву плату і кабель. І, нарешті, до трансивера приєднують гнучкий кабель з 15-контактними роз'єднаннями на обох кінцях – другим кінцем він під'єднується до роз'єднання АUI (attachment unit interface) на мережевій платі.

Вся ця складність була виправдана тільки одним – допустима максимальна довжина “товстого” коаксіального кабелю складала 500 метрів. Відповідно одним таким кабелем можна покрити набагато більшу площу, ніж “тонким” кабелем, максимально допустима довжина якого складає 185 метрів. Можна уявляти собі, що “товстий” коаксіальний кабель – це розподілений в просторі Ethernet-концентратор, тільки повністю пасивний і такий, що не вимагає живлення. Інших переваг у нього немає, недоліків багато – перш за все висока вартість самого кабелю (близько 2,5 дол. за метр), необхідність використання спеціальних пристроїв для монтажу (25-30 дол. за штуку), незручність прокладки і т.п. Це поступово привело до того, що “товстий Ethernet” поволі зійшов зі сцени, і в даний час мало де застосовується.

“Тонкий Ethernet” був поширений значно ширше, ніж його “товстий” побратим. Принцип використання у нього той самий, але завдяки гнучкості кабелю він може приєднуватися безпосередньо до мережевої плати. Для під'єднання кабелю використовуються роз'єднання BNC (bayonet nut connector), що встановлюються власне на кабель, і Т-конектори, що виконують відведення сигналу від кабелю в мережеву плату. Роз'єднання типу BNC бувають обтискові і розбірні (приклад розбірного роз'єднання – вітчизняний СР-50-74Ф).



Т-конектор



BNC конектор

Для монтажу роз'єднання на кабель потрібно або спеціальний інструмент для обтискання, або паяльник і плоскогубці.

Кабель підготовляють так:

1. Акуратно відріжте кабель так, щоб його торець був рівним. Надіньте на кабель металеву муфту (відрізок трубки), який поставляється в комплекті з BNC-роз'єдками.

2. Зніміть з кабелю зовнішню пластикову оболонку на довжину приблизно 20 мм. Будьте акуратні, щоб не пошкодити по можливості жоден провідник екрану.

3. Переплетені провідники екрану акуратно розплетіть і розведіть в різні боки. Зніміть ізоляцію з центрального провідника на довжину приблизно 5 мм.

4. Встановіть центральний провідник в штир, який також поставляється в комплекті з роз'єдками BNC. Використовуючи спеціальний інструмент, надійно обтисніть штир, фіксуючи в ньому провідник, або впаяйте провідник в штир. При паянні будьте особливо акуратні і уважні – погане паяння через деякий час стане причиною відмов в роботі мережі, причому локалізувати це місце буде достатньо важко.

5. Вставте центральний провідник зі встановленим на нього штирем в тіло роз'єдкати доки не почуєте клацання. Клацання означає, що штир сів на своє місце в роз'єдці і зафіксувався там.

6. Рівномірно розподіліть провідники екрану по поверхні роз'єдкати, якщо необхідно, обріжте їх до потрібної довжини. Насуньте на роз'єдкати металеву муфту.

7. Спеціальним інструментом (або плоскогубцями) акуратно обтисніть муфту до забезпечення надійного контакту екрану з роз'єдками. Не обтискайте дуже сильно – можна пошкодити роз'єдкати або перетиснути ізоляцію центрального провідника. Останнє може привести до нестійкої роботи всієї мережі. Але і обтискати дуже слабо теж не можна – поганий контакт екрану кабелю з роз'єдками також приведе до відмов в роботі.

Відзначимо, що вітчизняне роз'єдкати CP-50 вмонтовується приблизно так само, за винятком того, що екран в ньому закладається в спеціальну розрізну втулку і закріплюється гайкою. В деяких випадках це може виявитися навіть зручнішим.



CP-50

Кабелі на основі скрученої пари дротів

Скручена пара дротів (UTP/STP, unshielded / shielded twisted pair) в даний час є найбільш поширеним засобом передачі даних в локальних мережах. Кабелі UTP/STP використовуються в мережах Ethernet, Token Ring і ARCnet. Вони розрізняються за категоріями (залежно від смуги пропускання) і типами провідників (гнучкі або одножильні). Кабель 5-ої категорії, як правило, складається з восьми провідників, скручених попарно (тобто чотири пари).



Кабель UTP

Структурована кабельна система, побудована на основі скрученої пари 5-ої категорії, має дуже велику гнучкість у використанні. Її ідея полягає в наступному.

На кожне робоче місце встановлюється не менше двох (рекомендується три) чотирипарні розетки RJ-45. Кожна з них окремим кабелем 5-ої категорії з'єднується з кросом або патч-панеллю, встановленою в спеціальному приміщенні, - серверній. У це приміщення заводяться кабелі зі всіх робочих місць, а також міські телефонні лінії, виділені лінії для під'єднання до глобальних мереж і т.п. У цьому приміщенні монтуються сервери, а також офісна АТС, системи сигналізації і інше комунікаційне устаткування.

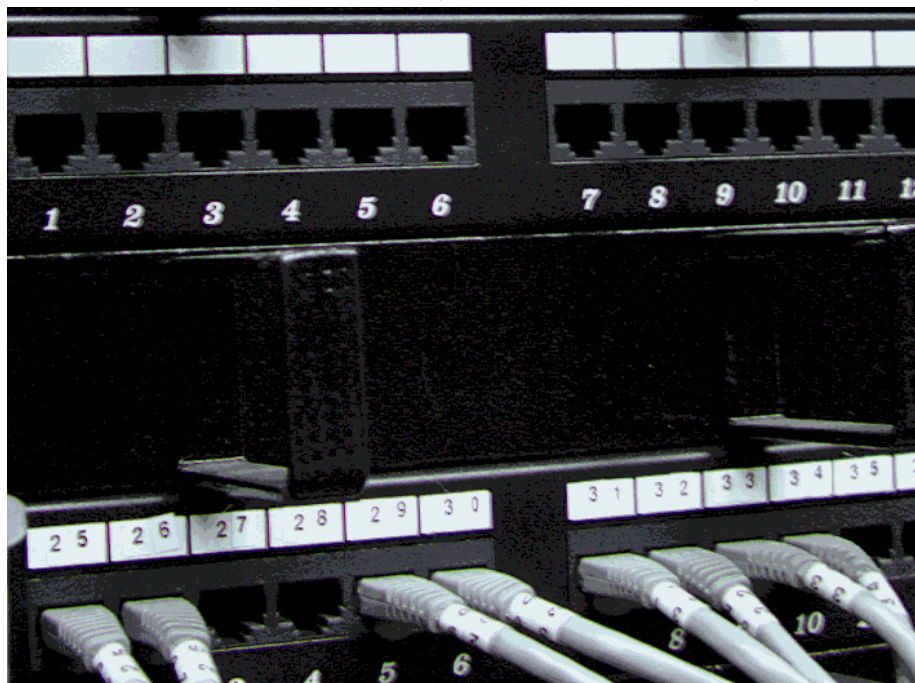
Завдяки тому що кабелі зі всіх робочих місць зведені на загальну панель, будь-яку розетку можна використовувати як для під'єднання робочого місця до ЛКМ, так і для телефонії або взагалі чого завгодно. Припустимо, дві розетки на робочому місці було під'єднано до комп'ютера і принтера, а третя – до телефонної станції. В процесі роботи з'явилася необхідність прибрати принтер з робочого місця і встановити замість нього другий телефон. Немає нічого простішого – патч-корд відповідної розетки відмикається від

концентратора і перемикається на телефонний крос, що займе у адміністратора мережі не більше декількох хвилин.



Розетка на 2 порти

Патч-панель, або панель з'єднань, є групою розеток RJ-45, змонтованих на пластині шириною 19 дюймів. Це стандартний розмір для універсальних комунікаційних шаф – “реків” (rack), в яких встановлюється устаткування (концентратори, сервери, джерела безперебійного живлення і т.п.). На зворотному боці панелі змонтовані з'єднувачі, в які вмонтовуються кабелі.



Патч-панель

Крос на відміну від патч-панелі розеток не має. Замість них він несе на собі спеціальні сполучні модулі. Крім того, крос можна вмонтовувати прямо на стіну - наявності комунікаційної шафи він не вимагає. Насправді, немає сенсу купувати дорогі комунікаційні шафи, якщо вся ваша мережа складається з одного-двох десятків комп'ютерів і сервера.

Кабелі з багатожильними гнучкими провідниками використовуються в якості патч-кордів, тобто сполучних кабелів між розеткою і мережевою платою, або між розетками на панелі з'єднань або кросі. Кабелі з одножильними провідниками – для прокладки самої кабельної системи. Монтаж рознять і розеток на ці кабелі абсолютно ідентичний, але зазвичай кабелі з одножильними провідниками вмонтовуються на розетки робочих місць користувачів, панелі з'єднань і кроси, а розняття встановлюють на гнучкі сполучні кабелі.

Як правило, застосовують наступні види рознять:

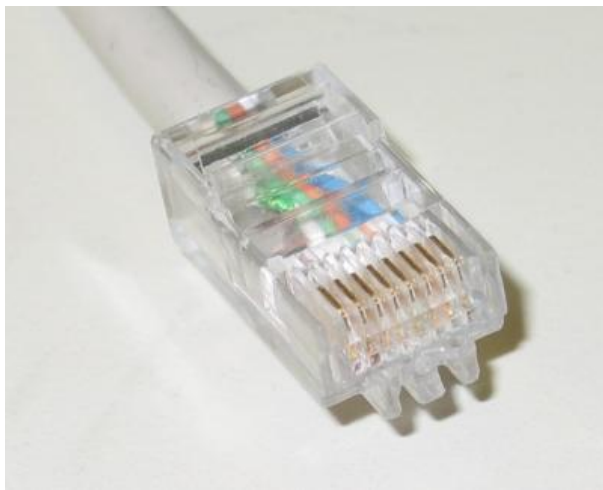
S110 – загальна назва рознять для під'єднання кабелю до універсального кросу “110” або комутації між входами на кросі;

RJ-11 і RJ-12 – розняття з шістьма контактами. Перші зазвичай застосовуються в телефонії загального призначення – ви можете зустріти такі розняття на шнурах телефонних апаратів. Другий зазвичай використовується в телефонних апаратах, призначених для роботи з офісними МІНІ-АТС, а також для під'єднання кабелю до мережевих плат ARCnet;



RJ-11

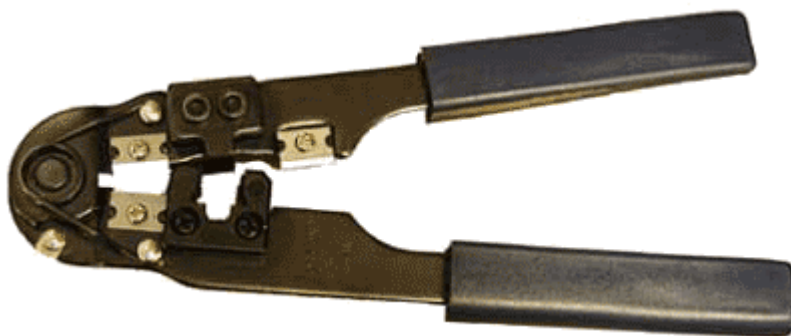
В телекомунікаціях використовують **конектор 8P8C** (8 Position, 8 Contact) — уніфікований конектор, що містить 8 контактів та фіксатор. Помилково вважають за RJ-45, хоч справжній RJ-45 фізично не сумісний з 8P8C, оскільки RJ45 використовує схему 8P4C. Тут будемо вважати, що RJ-45 – восьмиконтактне розняття, що використовується зазвичай для під'єднання кабелю до мережевих плат Ethernet або для комутації на панелі з'єднань.



RJ-45(насправді 8P8C)

Залежно від того, що з чим потрібно комутувати, застосовуються різні патч-корди: "45- 45" (з кожного боку по роз'єднню RJ-45), "110-45" (з одного боку S110, з іншої - RJ-45) або "110- 110".

Для монтажу роз'єднень RJ-11, RJ-12 і RJ-45 використовуються спеціальні інструменти для обтискання, що розрізняються між собою кількістю ножів (6 або 8) і розмірами гнізда для фіксації роз'єднень. Як приклад розглянемо монтаж кабелю 5-ої категорії на роз'єднень RJ-45.



Інструмент для обтискання RJ-45

1. Акуратно обріжте кінець кабелю. Край кабелю повинен бути рівним.
2. Використовуючи спеціальний інструмент, зніміть з кабелю зовнішню ізоляцію на довжину приблизно 30 мм і обріжте нитку, вмонтовану в кабель (нитка призначена для зручності зняття ізоляції з кабелю на велику довжину). Будь-які пошкодження (надрізи) ізоляції провідників абсолютно недопустимі – саме тому бажано використовувати спеціальний інструмент, лезо різача якого виступає рівно на товщину зовнішньої ізоляції.
3. Акуратно розведіть, розплетіть і вирівняйте провідники. Вирівняйте їх в один ряд, при цьому дотримуючись колірного порядку. Існує два найбільш поширених стандарти по розводці кольорів по парах: T568A (рекомендований

компанією Siemon) і T568B (рекомендований компанією AT&T і фактично найчастіше вживаний).

Номер пари	Колір за T568B	Колір за T568A
1	Синя	Синя
2	Оранжева	Зелена
3	Зелена	Оранжева
4	Коричнева	Коричнева

На рознятті RJ-45 кольори провідників розташовуються так:

Номер контакту	Колір за T568B	Колір за T568A
1	Біло-оранжевий	Біло-зелений
2	Оранжевий	Зелений
3	Біло-зелений	Біло-оранжевий
4	Синій	Синій
5	Біло-синій	Біло-синій
6	Зелений	Оранжевий
7	Біло-коричневий	Біло-коричневий
8	Коричневий	Коричневий

Провідники повинні розташовуватися строго в один ряд, не накладаючись один на одного. Утримуючи їх однією рукою, іншою рівно обрізайте провідники так, щоб вони виступали над зовнішньою обмоткою на 8-10 мм.

4. Тримавши розняття заглушкою донизу, вставте в нього кабель. Кожен провідник повинен потрапити на своє місце в рознятті і упертися в обмежувач. Перш ніж обтискати розняття, переконайтеся, що ви не помилилися в розводці провідників. При неправильній розводці крім відсутності відповідності номерам контактів на кінцях кабелю, що легко виявляється за допомогою простого тестера, можлива неприємніша річ – поява “розбитих пар” (splitted pairs). Для виявлення цього браку звичайного тестера недостатньо, оскільки електричний контакт між відповідними контактами на кінцях кабелю існує і з вигляду все нібито нормально. Але такий кабель ніколи не зможе забезпечити нормальну якість з’єднання навіть в 10-мегабітній мережі на відстань більше

40-50 метрів. Тому потрібно бути уважним і не поспішати, особливо якщо у вас немає достатнього досвіду.

5. Вставте роз'єд в гніздо на інструменті для обтискання і обтисніть його до упору-обмежувача на інструменті. В результаті фіксатор на роз'єдті встане на своє місце, утримуючи кабель в роз'єдті нерухомим. Контактні ножі роз'єдта вріжуться кожен в свій провідник, забезпечуючи надійний контакт.

Аналогічно можна здійснити монтаж роз'єдів RJ-11 і RJ-12, використовуючи відповідний інструмент.

Для монтажу роз'єдта S110 спеціального інструмента для обтискання не потрібно. Саме роз'єдта поставляється в розібраному вигляді. До речі, на відміну від “одноразових” роз'єдів типу RJ роз'єдта S110 допускає багатократне розбирання і збірку, що дуже зручно. Послідовність дій при монтажі наступна:

1. Зніміть зовнішню ізоляцію кабелю на довжину приблизно 40 мм, розведіть в сторони пари провідників, не розплітаючи їх.

2. Закріпіть кабель (у тій половинці роз'єдта, на якій немає контактної групи) за допомогою пластмасової стяжки і відріжте “хвіст”, що вийшов.

3. Акуратно укладіть кожен провідник в органайзер на роз'єдті. Не розплітайте пару на більшу, ніж потрібно, довжину – це погіршить характеристики всього кабельного з'єднання. Послідовність укладання пар звичайна – синя-оранжева-зелена-коричнева; при цьому світлий дріт кожної пари укладається першим.

4. Гострим інструментом (бокорізами або ножем) обріжте кожен провідник по краю роз'єдта.

5. Встановіть на місце другу половинку роз'єдта і руками обтисніть її до замикання всіх фіксаторів. При цьому ножі контактної групи вріжуться в провідники, забезпечуючи контакт.



4-парне роз'єдта S110

Оптоволоконні кабелі

Оптоволоконні кабелі – найбільш перспективний і найшвидший спосіб розповсюдження сигналів для локальних мереж і телефонії. У локальних мережах оптоволоконні кабелі використовуються для роботи по протоколах ATM, FDDI а тепер і Gigabit Ethernet.

Оптоволокно, як зрозуміло з його назви, передає сигнали за допомогою імпульсів світлового випромінювання. Як джерела світла використовуються напівпровідникові лазери, а також світлодіоди. Оптоволокно поділяється на одно- і багатомодове.

Одномодове волокно дуже тонке, його діаметр складає близько 10 мікрон. Завдяки цьому світловий імпульс, проходячи по волокну, рідше відбивається від його внутрішньої поверхні, що забезпечує менше згасання. Відповідно одномодове волокно забезпечує велику дальність без застосування повторювачів. Теоретична пропускна спроможність одномодового волокна складає десятки Гбіт/с.

Багатомодове волокно має більший діаметр – 50 або 62,5 мікрона. Цей тип оптоволокна найчастіше застосовується в комп'ютерних мережах. Більше загасання в багатомодовому волокні пояснюється вищою дисперсією світла в ньому, через яку його пропускна спроможність істотно нижча – теоретично вона складає 2,5 Гбіт/с.

Для з'єднання оптичного кабелю з активним устаткуванням застосовуються спеціальні роз'єднання. Найбільш поширені роз'єднання типу SC і ST.



Конектори SC та ST.

Монтаж з'єднувачів на оптоволоконний кабель – дуже відповідальна операція, що вимагає досвіду і спеціального обладнання, тому зараз не розглядається.

Хід роботи

Ознайомитись з технічними характеристиками наступних апаратних засобів і устаткування для створення ЛКМ:

1. Мережеві адаптери Ethernet і Token Ring для шин ISA, PCI, MCA.
2. Види кабелів для мереж (коаксіальний, неекранована скручена пара, оптоволокно).
3. Пристрої з'єднання BNC, RJ-45, настінні і модульні розетки, термінатори.
4. Елементи ЛКМ: монтажні коробки, патч-панелі, патч-корди, абонентські шнури. Розділення кабелю UTP за стандартами TIA/EIA - 568 A/B.
5. Варіанти виконання активних концентраторів (хаби, комутатори, MAU).
6. Виготовити кабелі і виконати з'єднання двох – трьох комп'ютерів за допомогою неекранованої скрученої пари дротів і концентратора та перевірити функціонування обладнання за допомогою індикаторів концентратора та мережної карти.
7. Оформити звіт про виконання лабораторної роботи у якому подати:
 - a. тему, мету та завдання лабораторної роботи;
 - b. прізвище, ініціали та назву групи студента, що виконав роботу;
 - c. короткий опис досліджуваних вузлів;
 - d. висновки.
8. Звіт оформити у вигляді файла з іменем:
“Прізвище”lan1.doc.

ЛАБОРАТОРНА РОБОТА № 2

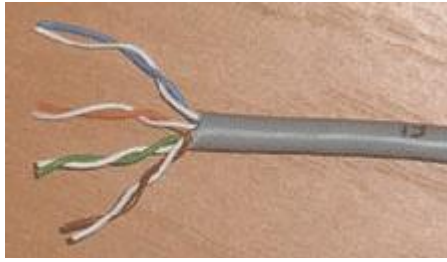
З'єднання комп'ютерів за допомогою cross-over або прямого кабелю в мережу.

Мета роботи: отримання знань і практичних навичок, необхідних для з'єднання комп'ютерів за допомогою cross-over або прямого кабелю в мережу на базі операційної системи MS Windows 7.

Теоретичні відомості

При необхідності з'єднання пари комп'ютерів через мережеві інтерфейси знадобляться встановлені і настроєні мережеві карти в обох комп'ютерах, мережевий кабель UTP/FTP/STP/SFTP 4pair (мал. 2.1), з якого необхідно зробити прямий або кроссовер (crossover) кабель, два конектори

RJ-45 (мал. 2.2) для приєднання на кінцях кабелю і інструмент для обтискання (мал. 2.3).



мал. 2.1



мал. 2.2



мал. 2.3

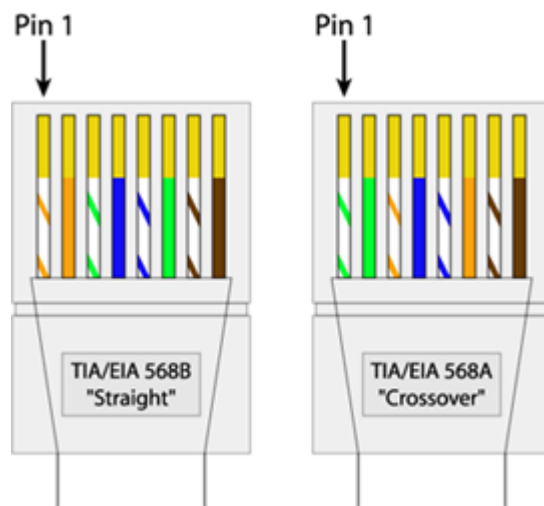
Використовується звичайний кабель ("скручена пара") для локальних мереж UTP/FTP/STP/SFTP що має 4 пари провідників. Необхідно визначити, скільки кабелю потрібно для з'єднання 2-х комп'ютерів, враховуючи, що довжина не може перевищувати 90 м і бути не меншою 1,5 м.

Кабель обтискається з обох боків роз'ємними RJ-45, за певним правилом.

Cross-over ("нуль хабний") – використовується для з'єднання двох комп'ютерів через мережеві карти безпосередньо, тобто не використовуючи активне мережеве обладнання (концентратор – hub, коммутатор – switch). У такий спосіб можна приєднати тільки два комп'ютери одночасно. Для приєднання трьох і більше комп'ютерів потрібно додаткове мережеве устаткування. Сучасні мережеві карти дозволяють також з'єднати два комп'ютери через мережеві карти безпосередньо за допомогою прямого кабелю.

При підключенні трьох і більше комп'ютерів через концентратор або комутатор використовується кабель типу Straight-through (такий, що проходить прямо). Назва цього виду кабелю говорить сама за себе – він передає сигнал безпосередньо з одного кінця в іншій, а саме з 1-го контакту на 1, 2-2, 3-3 і т.д. Використовується для різних видів з'єднань (комп'ютер – концентратор, комп'ютер – ADSL/ISDN/кабельний модем, або з'єднання концентратор і комутатор між собою).

При обтисканні провідників скористаємося стандартом TIA/EIA-568B.



З однієї сторони провідники повинні бути розташовані в наступному порядку:

1	Біло-оранжевий
2	Оранжевий
3	Біло-зелений
4	Синій
5	Біло-синій
6	Зелений
7	Біло-коричневий
8	Коричневий

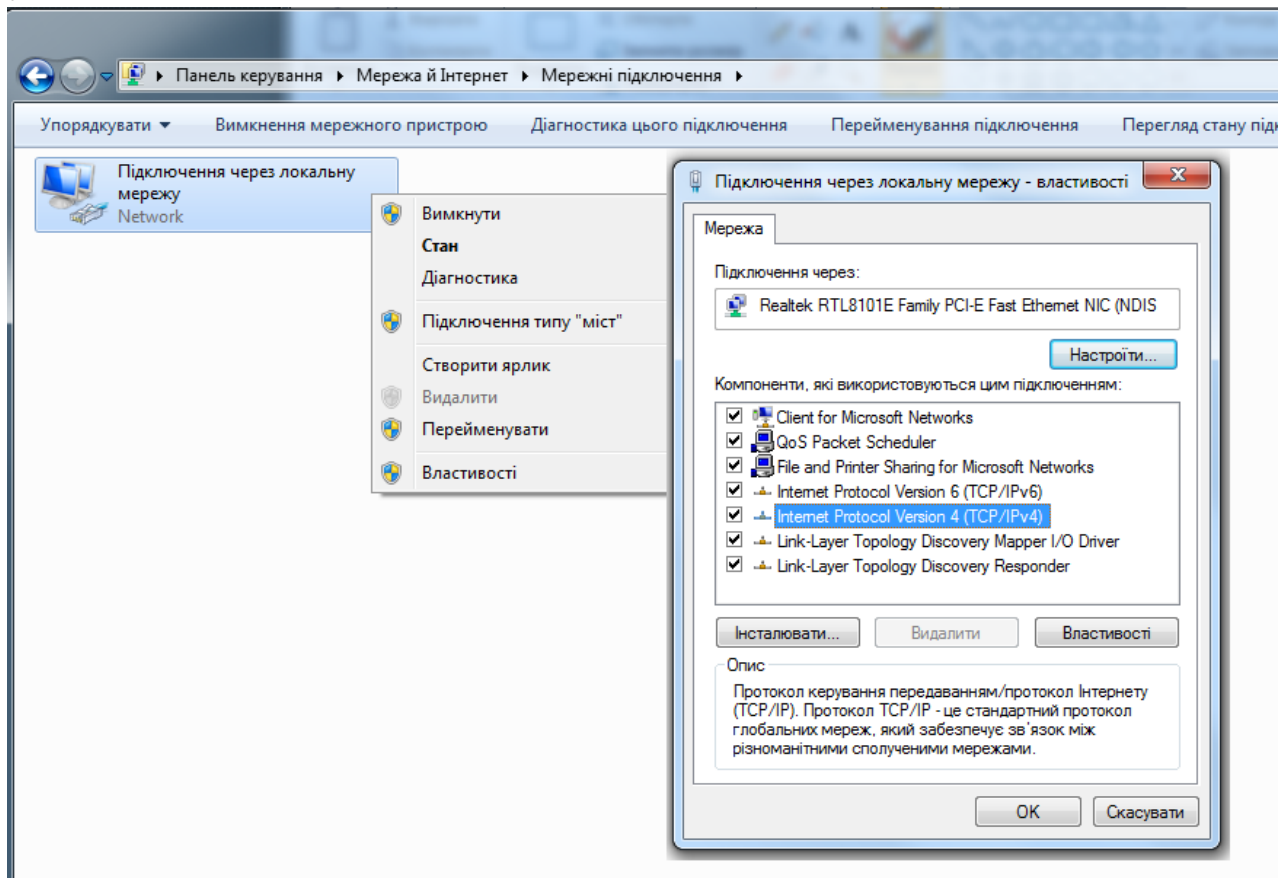


З іншого боку провідники повинні бути розташовані в іншому порядку:

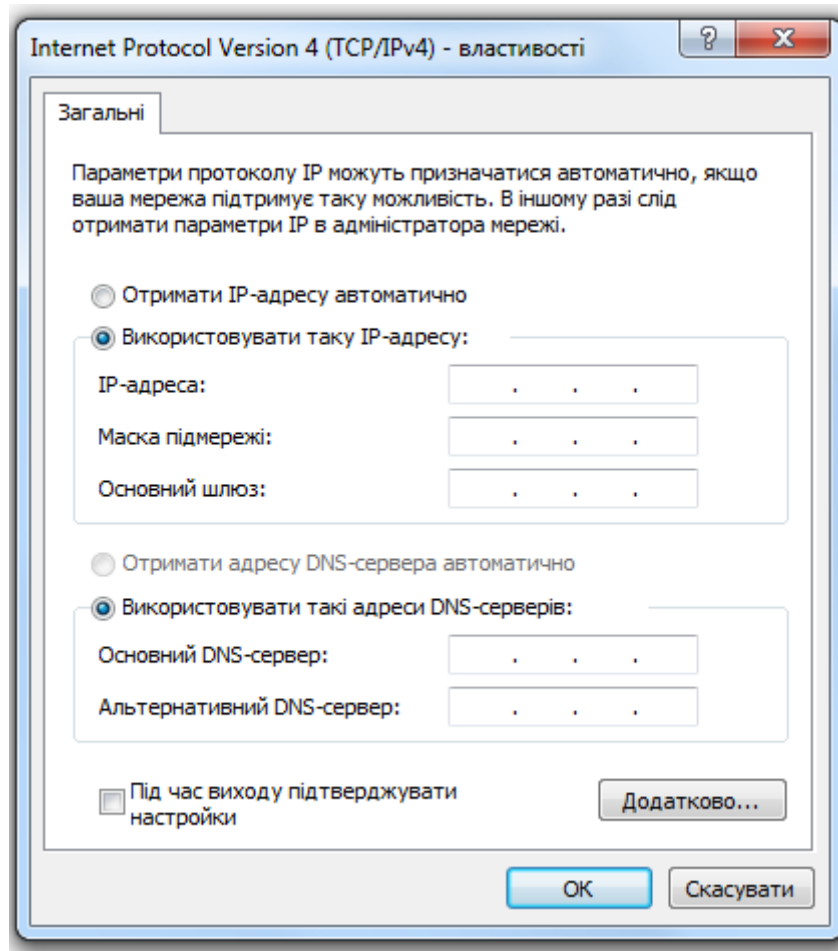
1	Біло-зелений
2	Зелений
3	Біло-оранжевий
4	Синій
5	Біло-синій
6	Оранжевий
7	Біло-коричневий
8	Коричневий

Вмикаємо отриманий кабель в мережеві карти комп'ютерів і приступаємо до налаштування операційної системи. У *Панель Керування* вибираємо ярлик *Мережа й Інтернет* і у вікні, що з'явилося, знаходимо ярлик *Мережні підключення через локальну мережу*, натискаємо праву кнопку і обираємо пункт *Властивості*:

На закладці *Мережа* в списку *Компоненти, які використовуються цим підключенням* вибираємо *Інтернет Протокол версії4(TCP/IP)* і натискаємо кнопку *Властивості*

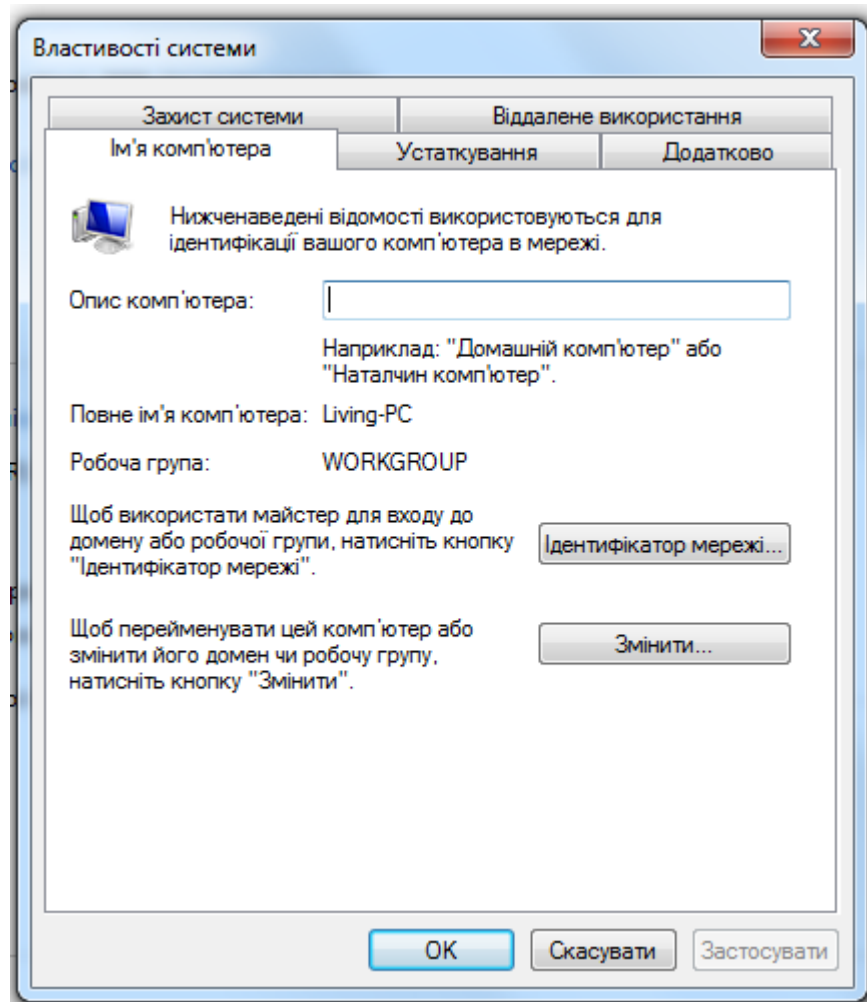


У вікні, що з'явилося вибираємо *Використовувати таку IP адресу*.

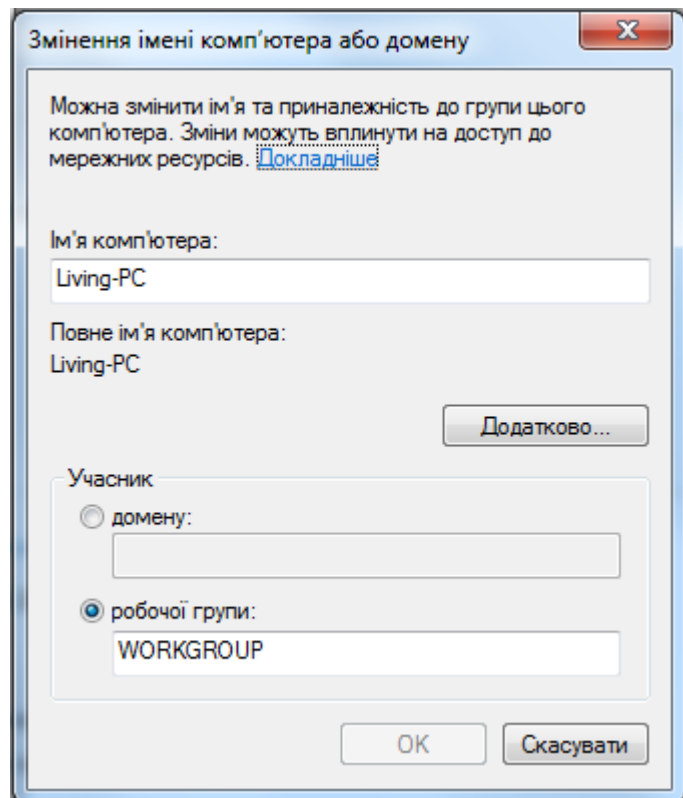


Для локальної мережі ми використовуємо адресний простір мережі класу С 192.168.0.x, де x – від 1 до 255. На одному комп'ютері вказуємо адресу 192.168.0.1, а на іншому адресу 192.168.0.2. Дуже важливо, щоб IP адреси відрізнялися одна від одної останньою цифрою. Маска підмережі може бути вказана 255.255.255.0, вона встановлюється обов'язково однакова на всі комп'ютери локальної мережі.

Тепер необхідно налаштувати робочу групу, а також ввести ім'я комп'ютера для представлення в мережі. Для цього натискаємо правою кнопкою миші на іконці *Мій комп'ютер* і вибираємо пункт *Властивості*. У вікні, що з'явилося, переходимо на закладку *Ім'я комп'ютера* і натискаємо кнопку *Змінити настройки*.



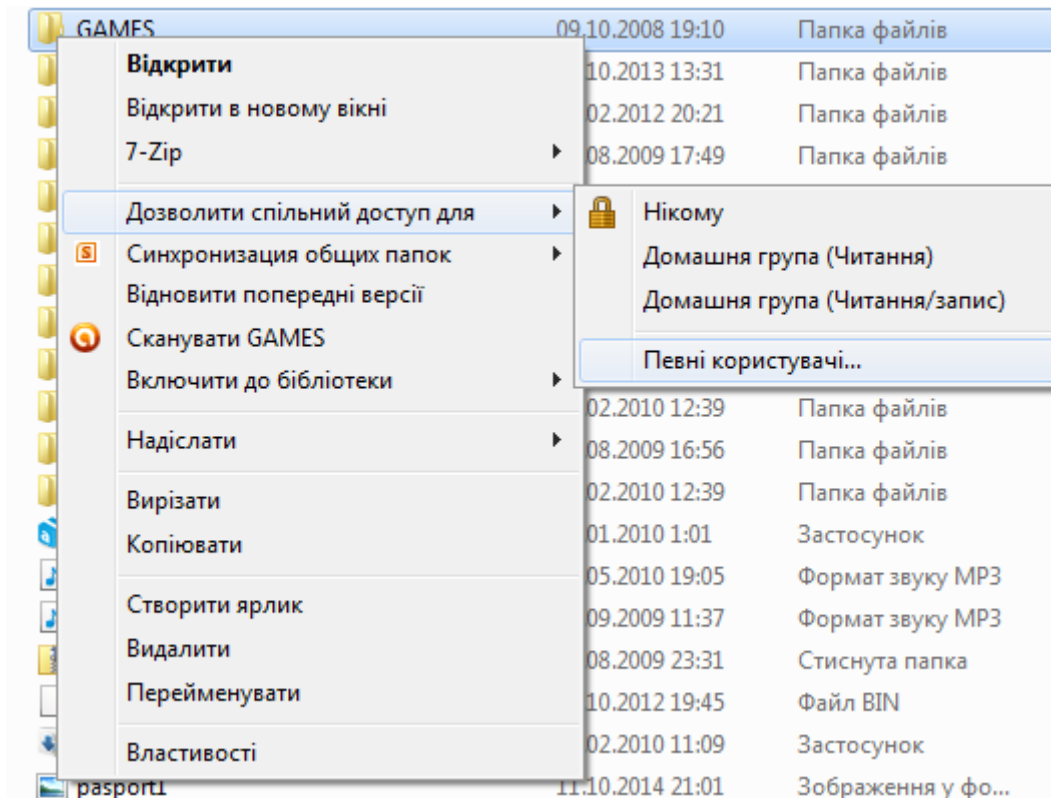
У полі *Ім'я комп'ютера* вписуємо ім'я, яким комп'ютер буде представлений в мережі. Використовуйте англійські букви, цифри. Намагайтесь не використовувати інші символи, оскільки при цьому можливі проблеми в роботі мережі.



З'єднання вже налаштоване, тепер залишилося зробити декілька відкритих ресурсів (“розшарити” від англійського слова "share"— ділитися, розділяти). Для цього вибираємо теку яку хочемо відкрити для доступу іншому комп'ютеру, і натискаємо на ній правою кнопкою миші. У меню, що з'явилося, вибираємо пункт *Дозволити спільний доступ для* (таким чином ви відкриваєте вікно властивостей даної теки).

У вікні, що з'явилося, маєте можливість *заборонити доступ до цієї папки* або дозволити різні рівні доступу (тільки читання або читання і запис) користувачам мережі.

Вибравши пункт Властивості отримаєте можливість обмежити кількість користувачів мережі, які отримують доступ до вказаного ресурсу.



Хід роботи

1. Ознайомитись з технічними характеристиками та підготувати до роботи для з'єднання комп'ютерів мережеві адаптери Ethernet.
2. Виготовити або перевірити готовий прямий чи cross-over кабель.
3. Виконати з'єднання двох комп'ютерів за допомогою прямого чи cross-over кабелів (для сучасних ПК з'єднання двох комп'ютерів можна здійснити за допомогою прямого кабеля).
4. Налаштувати комп'ютери для спільної роботи, вказавши відповідні IP-адреси, ім'я комп'ютера, робочу групу.
5. Створити відкриті ресурси на обох комп'ютерах і виконати передавання файлів між комп'ютерами.
6. Оформити звіт про виконання лабораторної роботи у якому подати:
 - a. тему, мету та завдання лабораторної роботи;
 - b. прізвище, ініціали та назву групи студента, що виконав роботу;
 - c. короткий опис проведених досліджень;
 - d. висновки.
7. Звіт оформити у вигляді файла з іменем:
"Прізвище"lan2.doc.

ЛАБОРАТОРНА РОБОТА № 3

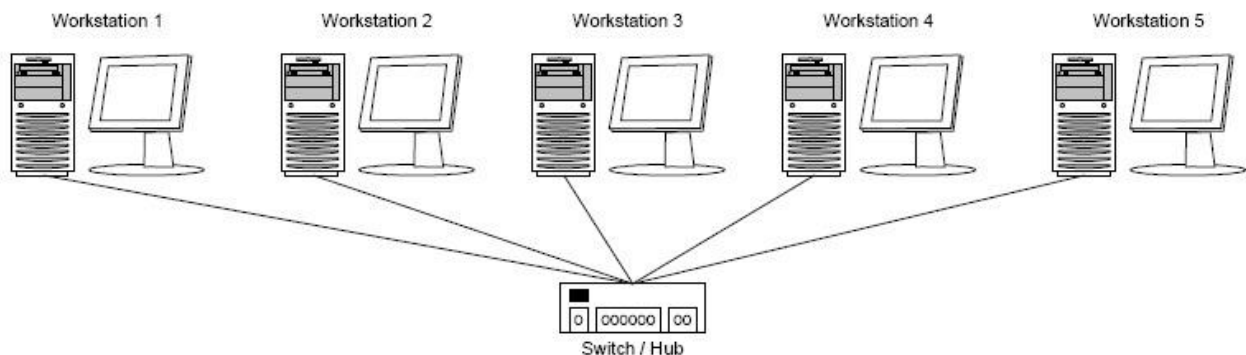
Побудова локальної обчислювальної мережі (ЛКМ) за мережевою технологією Fast Ethernet з використанням комутатора.

Мета роботи: отримання знань і практичних навичок, необхідних для з'єднання комп'ютерів за технологією Fast Ethernet (100 Base TX) в мережу на базі операційної системи MS Windows 7 та організація доступу до мережі Internet з використанням комутатора.

Теоретичні відомості

Мережева технологія – це узгоджений набір стандартних протоколів і програмно-апаратних засобів, що їх реалізують (наприклад, мережевих адаптерів, драйверів, кабелів і роз'єдів), достатній для побудови обчислювальної мережі. Протоколи, на основі яких будується мережа певної технології, спеціально розроблялися для спільної роботи, тому від розробника мережі не вимагаються додаткові зусилля з організації їх взаємодії.

Для з'єднання N комп'ютерів (два і більше) попередньо необхідно встановити і налаштувати у всіх комп'ютерах мережеві карти необхідної технології, підготувати мережевий кабель UTP/FTP/STP/SFTP 5e 4pair, з якого необхідно зробити N прямопрохідних кабелів (Straight-through), 2*N конекторів RJ-45 для “обтискання” кабелю і інструмент для обтискання. Технологія Fast Ethernet передбачає об'єднання комп'ютерів в мережу за допомогою мережевого устаткування. В ролі активного мережевого устаткування використовуємо концентратор або комутатор, що має N портів.



Об'єднання комп'ютерів в мережу за допомогою мережевого обладнання.

Для приєднання конекторів до провідників використовуємо стандарт TIA/EIA-568B (див. лаб. №2). Один кінець отриманого кабелю вмикаємо в мережеві карти комп'ютерів, а інший в порт мережевого устаткування і приступаємо до налаштування операційної системи Windows 7.

ЛКМ будують також за допомогою повторювачів, мостів, та маршрутизаторів. Існує безліч різних передавальних середовищ, кожне з яких має свої переваги й недоліки. Одним з недоліків кабелю UTP є обмеження на довжину кабелю. Максимальна довжина кабелю UTP для однієї ділянки мережі становить 100 м. Якщо потрібна більша відстань, то потрібно



використати *повторювачі*. Призначення повторювачів полягає у регенерації й ресинхронізації мережних сигналів на бітовому рівні для того, щоб вони могли пройти більшу відстань по передавальному середовищу. Повторювачі звичайно використовуються в тих випадках, коли в мережі є занадто багато вузлів або довжини наявного кабелю недостатньо для досягнення віддалених точок. Правило чотирьох повторювачів для шинної топології Ethernet 10 Мбіт/с, також відоме як правило 5-4-3, використовується як

стандарт при розширенні сегментів локальних мереж. Це правило стверджує, що не більше п'яти сегментів мережі можуть бути з'єднані один з одним за допомогою чотирьох повторювачів, але тільки три сегменти можуть при цьому мати підключені до них робочі станції (комп'ютери). Хоча правило 5-4-3 справедливо для мереж із шинною топологією, для більше складних мереж з комутаторами й зіркоподібною топологією воно не завжди застосовно.

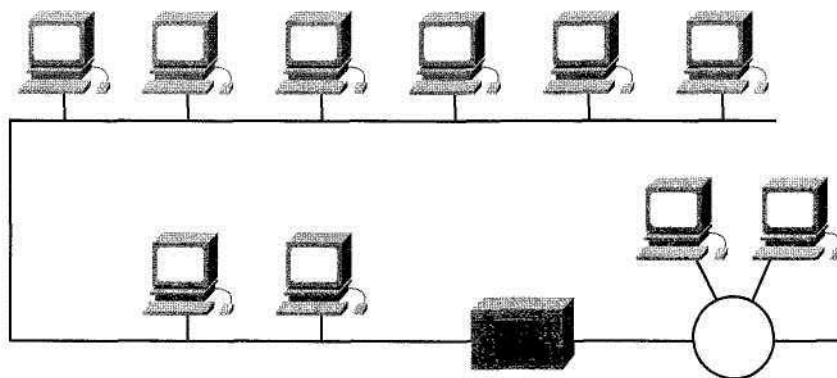


Рис. 1. Повторювач

Функції різних пристроїв добре видно з наступної таблиці, де мережі зображені у вигляді двох взаємодіючих відкритих інформаційних систем (модель ISO)

Мережа 1	Обладнання для з'єднань	Мережа 2
Прикладний (Application)	Шлюз (Gateway)	Прикладний (Application)
Представницький (Presentation)		Представницький (Presentation)
Сеансовий (Session)		Сеансовий (Session)
Транспортний (Transport)		Транспортний (Transport)
Мережний (Network)	Маршрутизатор (Router)	Мережний (Network)
Канальний (Data Link)	Міст (Bridge)	Канальний (Data Link)
Фізичний (Physical)	Ретранслятор (Repeater), Концентратор (hub)	Фізичний (Physical)

У більшості сучасних мереж Ethernet замість повторювачів використовуються концентратори, які є багатопортовими повторювачами або комутуючими пристроями, створеними на основі нових технологій.

Концентратори

Концентратори власне кажучи є багатопортовими повторювачами. У багатьох випадках різниця між цими двома пристроями складається тільки в кількості надаваних ними портів. У той час як типовий повторювач має тільки 2 порти, концентратор звичайно має від 4 до 24 портів, як показано на мал.2



Рис. 2. Восьмипортовий концентратор

Крім того, концентратори найчастіше використовуються в мережах 10BASE-T і 100BASE-T, хоча можуть використовуватися й в інших типах мереж. Використання концентратора перетворить мережну топологію із шинної, у якій кожний пристрій безпосередньо приєднаний до загальної шини, у зіркоподібну. При використанні концентраторів дані, що надходять на один з портів концентратора, повторюються за допомогою мікросхем на всіх інших портах, приєднаних до цього ж мережного сегмента, за винятком порту, з якого ці дані були відправлені.

Альтернативний способом підключення до ЛКМ є бездротове середовище, про що мова йтиме згодом у окремій лабораторній роботі. При її використанні не потрібно прокладати кабель, можна без використання постійного кабельного з'єднання передавати сигнали від одного комп'ютера іншому за допомогою радіозв'язку (radio frequency - RF), лазера, інфрачервоних променів (infrared - IR), а також супутникових сигналів або частот діапазону НВЧ.

Мости

Іноді потрібно розділити більшу локальну мережу на менші, легше керовані сегменти. Така стратегія дозволяє обмежити потік даних через окрему частину ЛКМ і розширити підтримувану мережею географічну область, як показано на мал. 3. Як пристрої, які можуть бути використані для з'єднання між собою мережних сегментів, можуть бути використані мости, комутатори, маршрутизатори й шлюзи. Комутатори й мости функціонують на канальному рівні моделі OSI. Функція мосту складається у визначенні (ухваленні осмисленого рішення) того, потрібно чи відправляти сигнали, що надійшли на нього, в інший сегмент мережі. Мости можуть також бути використані для з'єднання мереж, що використовують різні протоколи або різні передавальні середовища, як, наприклад, у випадку бездротових мостів, що з'єднують мережі Ethernet у мережу міського масштабу.

A bridge connecting two LAN segments

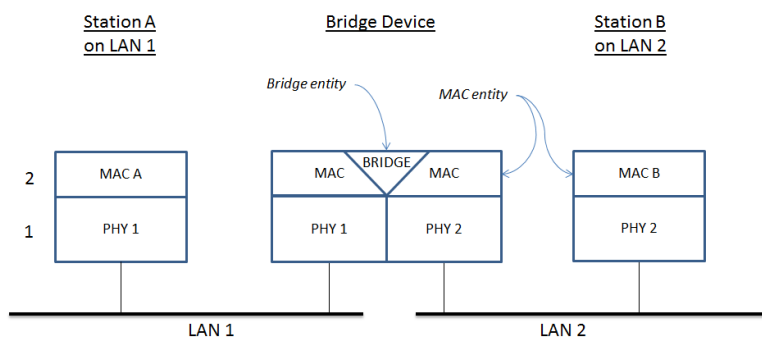
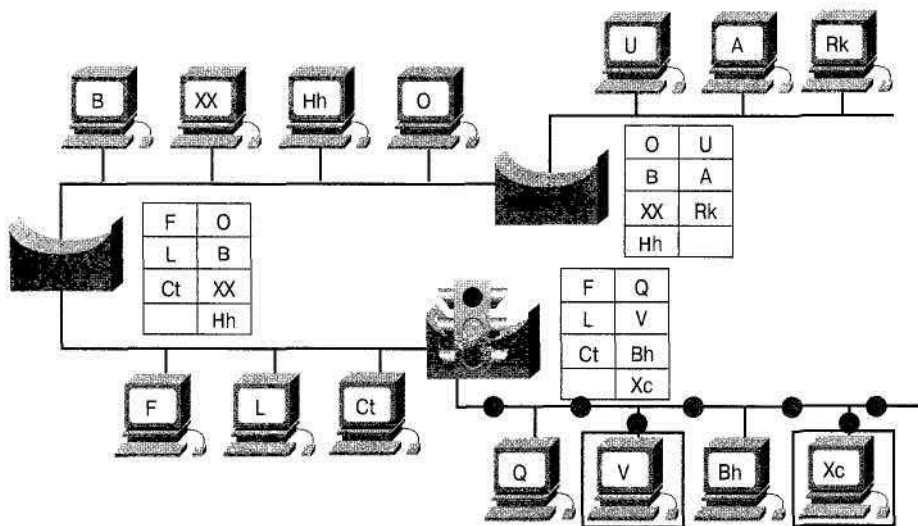


Рис. 3. Мости ділять мережу на сегменти

Коли міст одержує кадр (фрейм), він порівнює Мас-адресу відправника з наявною в нього адресною таблицею для визначення того, чи варто відфільтрувати цей фрейм (відкинути), розіслати його лавинним способом або скопіювати фрейм в інший сегмент. Прийняття такого рішення відбувається в такий спосіб:

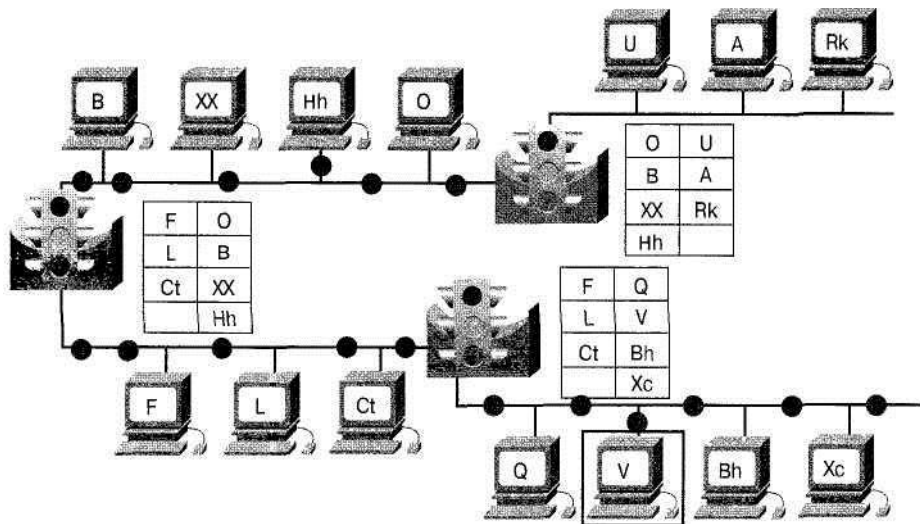
- якщо пристрій-одержувач перебуває в тім же сегменті, з якого цей фрейм був отриманий, то міст запобігає його передачі в інші сегменти, як показано на мал. 4. Цей процес називається *фільтрацією (filtering)*;
- якщо пристрій-одержувач перебуває в іншому сегменті і його адреса присутня в адресній таблиці, то міст пересилає фрейм у відповідний сегмент, як показано на мал. 5;
- якщо пристрій-одержувач відсутній у таблиці адрес (тобто "невідомо" мосту), то міст розсилає фрейм в усі сегменти за винятком того, звідки був отриманий фрейм. Такий стан називають *лавинним розсиленням повідомлень*.

Стратегічно правильно встановлений міст може значно збільшити продуктивність мережі.



У даному прикладі пакет даних відправляється з комп'ютера V, а пунктом призначення є комп'ютер Xc. Пакет надходить у пункт призначення й не розсилається широкомовно в інші сегменти мережі.

Рис. 4. Мости сегментують мережу: фільтрація



У даному прикладі пакет даних відправляється з комп'ютера V, а пунктом призначення є комп'ютер Hh. Міст переглядає свою адресну таблицю й визначає, чи варто відправляти сигнал в інші сегменти мережі.

Рис. 5. Мости сегментують мережу: пересилання

Комутатори

Комутатор іноді називають багатопортовим мостом. У той час як типовий міст має тільки два порти (з'єднує два мережних сегменти), комутатор може мати кілька портів, залежно від кількості мережних сегментів, які необхідно з'єднати. Як і мости, комутатори витягають певну інформацію з пакетів даних, які вони одержують від різних комп'ютерів мережі. Надалі ця інформація використовується для побудови таблиць комутації даних, які потім використовуються для визначення напрямку потоків даних, що відправляються одним з комп'ютерів мережі іншому, як показано на мал. 6.

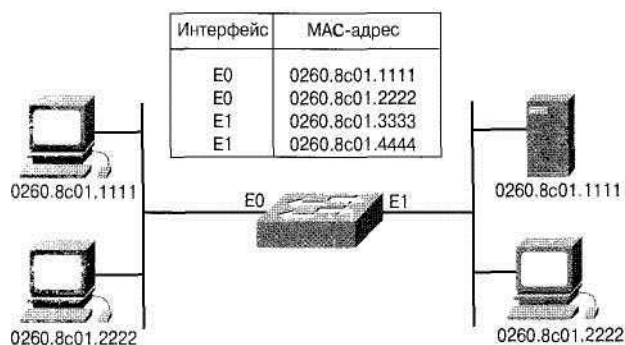


Рис. 6. Таблиця комутації

Хоча в роботі мостів і комутаторів є багато загального, комутатор являє собою більше складний пристрій, ніж міст. Міст визначає, чи направляється фрейм в інший мережний сегмент, на основі Мас-адреси одержувача. Комутатор має кілька портів, до яких приєднані сегменти мережі. Комутатор вибирає порт, до якого приєднаний пристрій-одержувач або робоча станція. Комутатори Ethernet стають усе більше популярними, оскільки, як і мости,

значно підвищують продуктивність мережі (швидкість передачі й смугу пропускання).

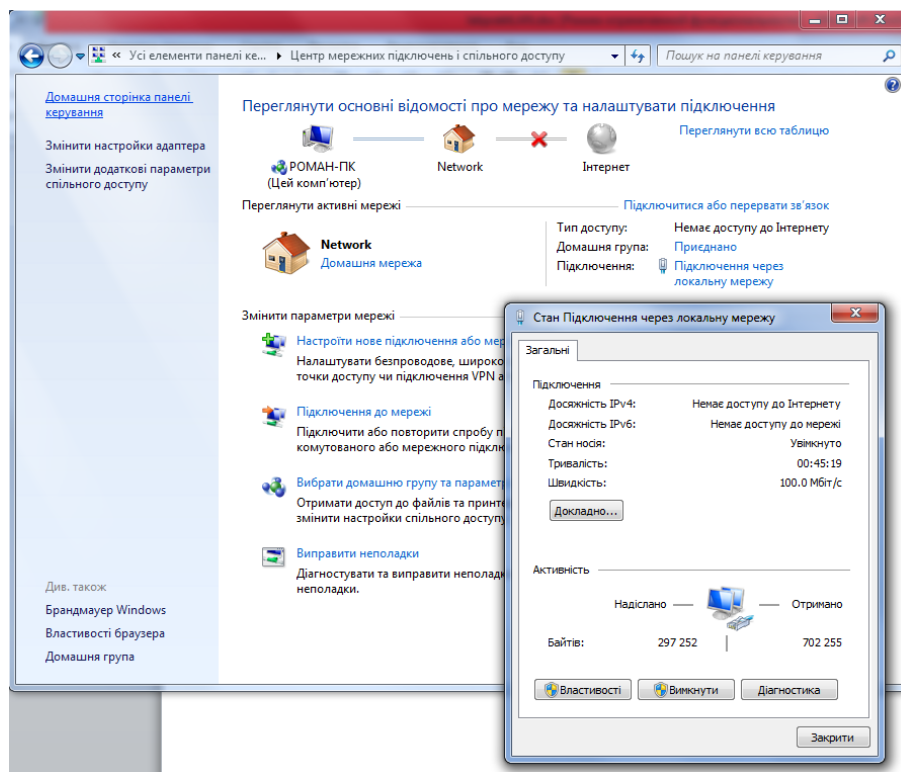
Комутація являє собою технологію, що знижує ймовірність виникнення в мережах Ethernet заторів за рахунок зменшення обсягів переданих по мережі даних і збільшення смуги пропускання. Комутатори часто використовуються для заміни концентраторів, оскільки не вимагають зміни існуючої кабельної інфраструктури, що дозволяє підвищити продуктивність мережі з мінімальною кількістю змін у вже існуючій мережі. У цей час у сфері передачі даних все комутуюче устаткування виконує дві основні операції:

Комутатори працюють із більшими швидкостями, ніж мости, а також можуть підтримувати додаткові й досить важливі функції, такі, як віртуальні локальні мережі VLAN (Virtual LAN).

Комутатор Ethernet має багато переваг, зокрема, дозволяє багатьом користувачам здійснювати зв'язок паралельно за рахунок використання віртуальних каналів і створювати виділені мережні сегменти, вільні від колізій. Такий підхід дозволяє максимізувати доступну смугу пропускання в загальному середовищі. Іншою перевагою є можливість повторно використати вже існуюче апаратне забезпечення й кабельну інфраструктуру, що робить перехід до використання комутаторів фінансово ефективним.

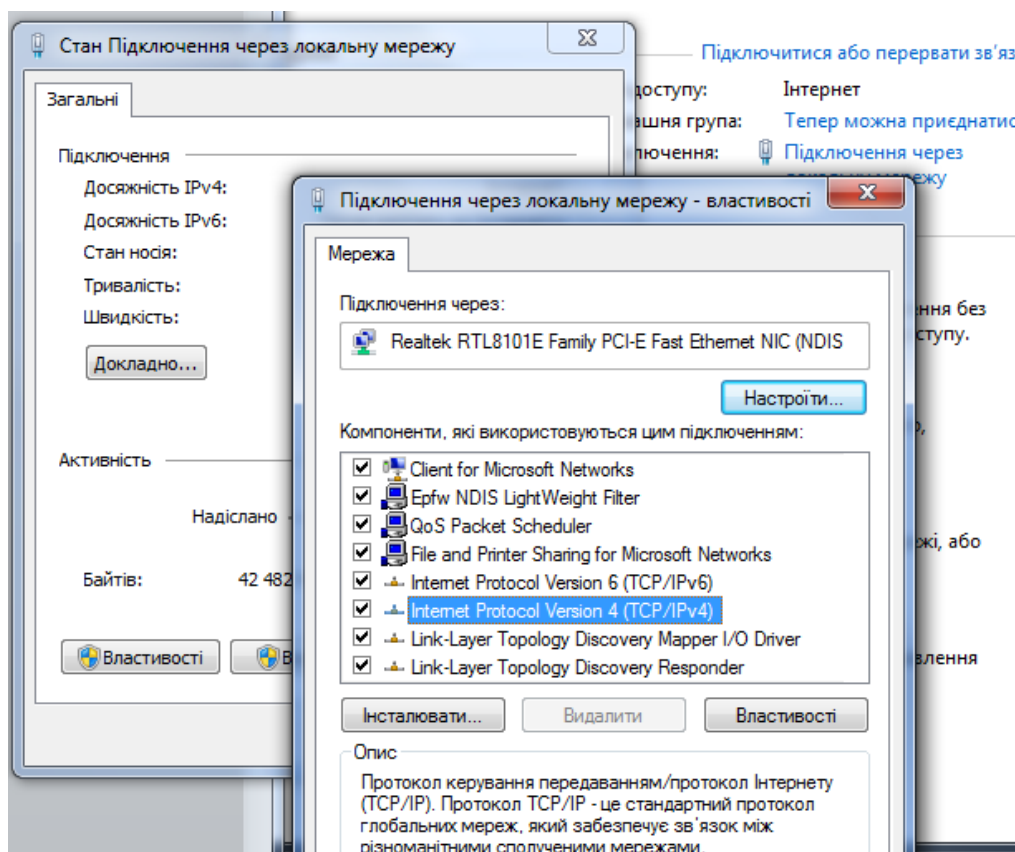
Приєднавши станції до комутатора за допомогою кабелів, як це ми робили у попередніх лабораторних роботах, переконуємось у надійності з'єднання, спостерігаючи за станом індикаторів комутатора і мережевих карт. Після цього приступаємо до налаштування адаптерів станцій.

У *Панелі Управління* послідовно вибираємо шлях до вікна *Мережеві підключення*: Пуск – Панель управління – Мережа та Інтернет – Центр управління мережами і загальним доступом – Зміна параметрів адаптера. Запускаємо його



і, натиснувши кнопку *Властивості* встановлюємо наступні параметри:

У вікні ви зможете побачити список компонентів. Тут необхідно вибрати пункт «Протокол Інтернету TCP/IP» і натиснути на кнопку «Властивості».



Після цього потрібно відкрити вікно властивостей обраного компонента. Тут необхідно відзначити пункт «Використовувати наступну IP адресу» і вписати необхідні IP адреси (не плутати з мас адресою).

Для першого комп'ютера:

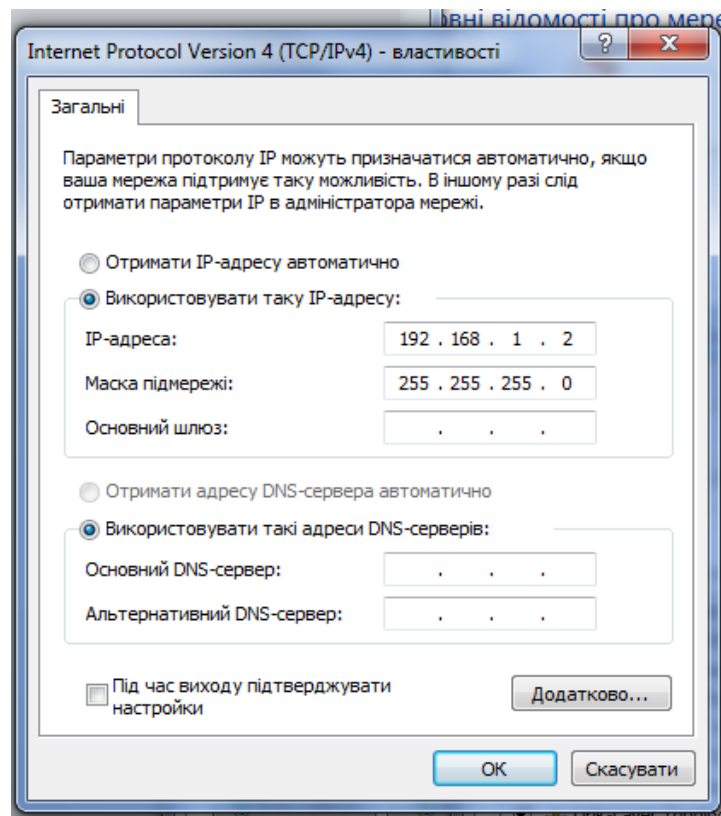
- «IP-адреса»- 192.168.1.1
- «Маска підмережі»- 255.255.255.0

Для другого комп'ютера:

- «IP-адреса»- 192.168.1.2
- «Маска підмережі»- 255.255.255.0

Інші поля залишаємо порожніми.

Якщо комп'ютерів у мережі буде більше двох, то їм необхідно присвоїти IP-адреси з наступними порядковими номерами, а маска підмережі, як і для попередніх, залишиться 255.255.255.0.



Після цього необхідно закрити всі відкриті вікна натисненням на кнопку «ОК». Тепер налаштування локальної мережі під Windows 7 закінчене і мережа запрацює протягом декількох секунд. Для перевірки роботи підключення можна виконати команду ping. Для цього на одному з комп'ютерів натисніть комбінацію клавіш Win-R та введіть команду “cmd”. Після чого в чорному вікні введіть команду ping і IP-адресу іншого комп'ютера. Результат виконання команди повинен бути таким як на картинці.

```
C:\Windows\system32\cmd.exe

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Роман>ping 192.168.0.1

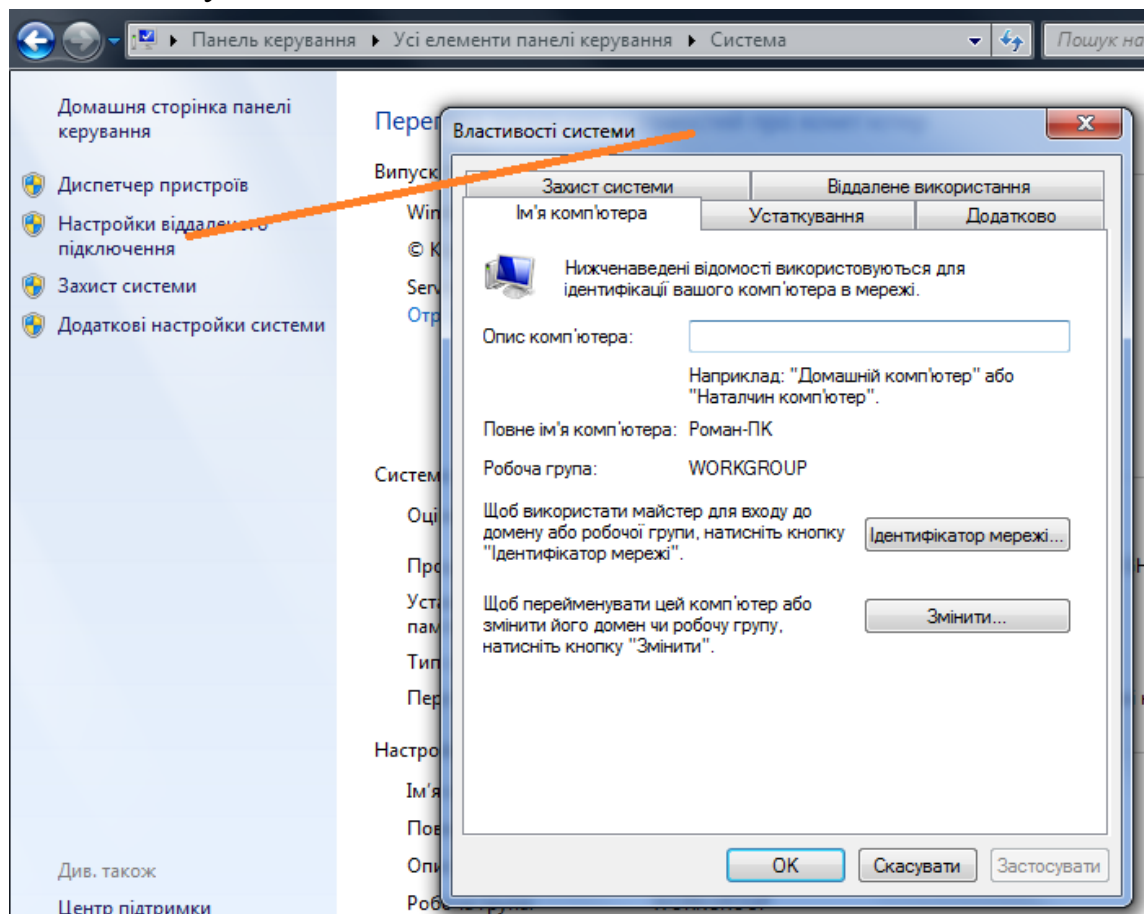
Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Роман>
```

Це означає, що все працює правильно і тепер ви знаєте, як налаштувати локальну мережу під Windows 7.

Тепер необхідно налаштувати робочу групу, а також ввести ім'я комп'ютера для представлення в мережі. Для цього натискаємо правою кнопкою миші на іконці *Мій комп'ютер* і вибираємо пункт *Властивості системи*. У вікні, що з'явилося, переходимо на закладку *Ім'я комп'ютера* і натискаємо кнопку *Змінити*.



У полі *Ім'я комп'ютера* вписуємо ім'я, яким комп'ютер буде представлений в мережі. Використовуйте англійські букви, цифри. Намагайтесь не використовувати інші символи, оскільки при цьому можливі проблеми в роботі мережі. Імена комп'ютерів повинні бути різними.

На даному етапі ми отримали локальну мережу, до якої входять всі наші комп'ютери.

Хід роботи

1. З'єднати комп'ютери між собою з використанням активного мережного обладнання.

2. Для з'єднання комп'ютерів з активним обладнанням необхідно підготувати додаткові кабелі (по одному на кожен комп'ютер). Кабель повинен бути прямим, а не перехресним, тобто мати однаковий стандарт обтискання з обох кінців. Хоча, в даний час часто зустрічається обладнання, яке автоматично визначає тип підключеного кабелю, і здатне працювати як з тим, так і з іншим.

3. Підключити всі комп'ютери до комутатора або концентратора за допомогою підготовлених кабелів.

4. Перевірити, що всі комп'ютери знаходяться в одній підмережі, і якщо це не так, змінити IP-адреси належним чином.

5. Перевірити, чи є доступ до загальних ресурсів всіх комп'ютерів. При відсутності доступу перевірити правильність обтиску кабелів і справність мережевих карт.

6. Оформити звіт про виконання лабораторної роботи у якому подати:

- a. тему, мету та завдання лабораторної роботи;
- b. прізвище, ініціали та назву групи студента, що виконав роботу;
- c. короткий опис процесу налагодження мережі;
- d. висновки.

7. Звіт оформити у вигляді файла з іменем: "Прізвище"lan3.doc.

ЛАБОРАТОРНА РОБОТА № 4

Ознайомлення з середовищем VMware 5.0. Створення та налаштування віртуальної машини.

Мета роботи: Ознайомитись з середовищем керування віртуальними машинами VMware 5.0. Навчитись створювати нові віртуальні машини та налаштовувати їх залежно від завдань, які буде виконувати віртуальна машина.

Хід роботи.

1. Створення віртуальної машини.

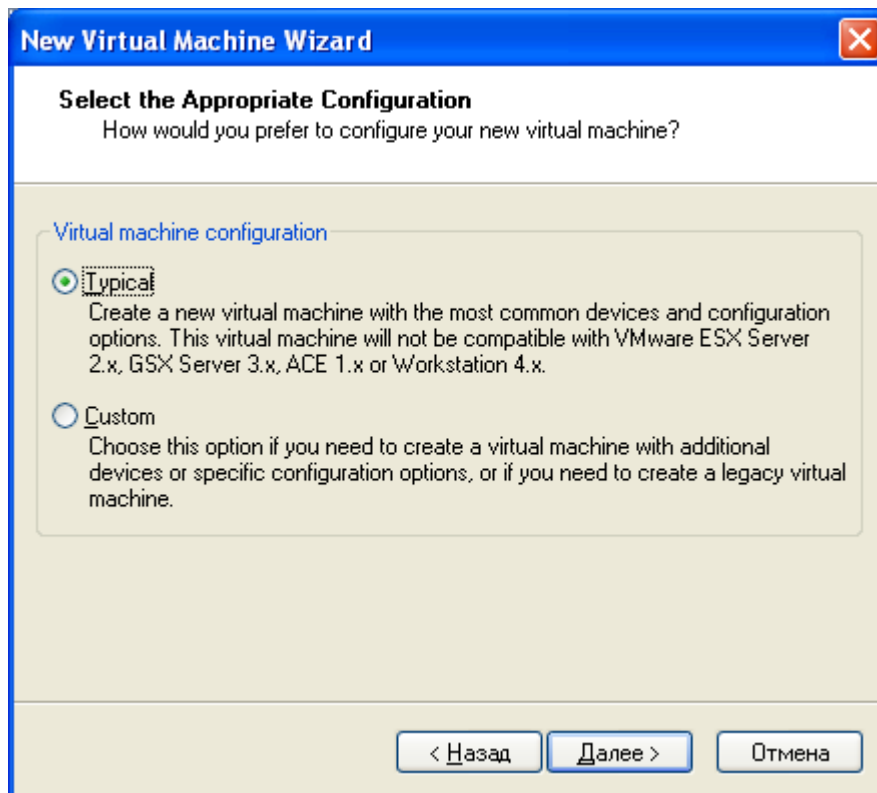
Для створення нової ВМ потрібно запустити середовище VMware 5.0 та зайти в пункт меню *File => New => Virtual Machine*. Перед нами з'явиться “майстер налаштувань” – діалог, який дозволяє ввести всі необхідні дані для створення ВМ. Розглянемо по пунктах всі вікна цього діалогу.

1. Вікно-привітання.



В цьому вікні натискаємо *Далее* для продовження роботи.

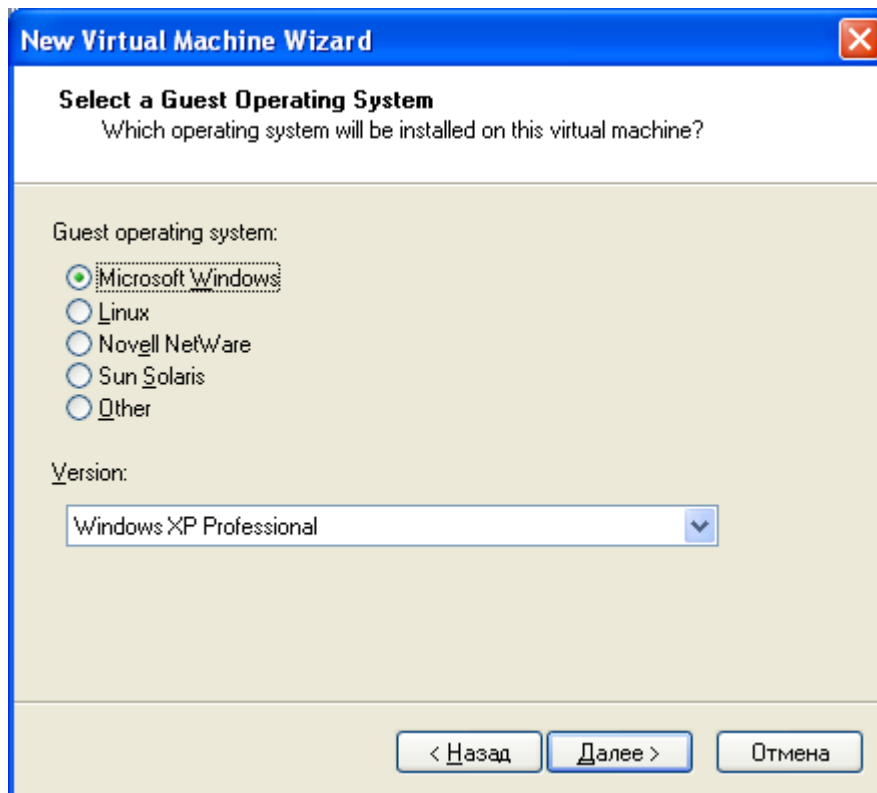
2. Вікно, в якому пропонується обрати один з типів введення налаштувань:



- a) *Typical* (типовий) – будуть введені типові налаштування та створені типові пристрої для ВМ автоматично.
- b) *Custom* (ручний) – для досвідчених користувачів. Надає можливість ввести всі необхідні налаштування вручну.

На даному етапі оберемо *Typical*, а всі необхідні налаштування введемо вже після створення ВМ.

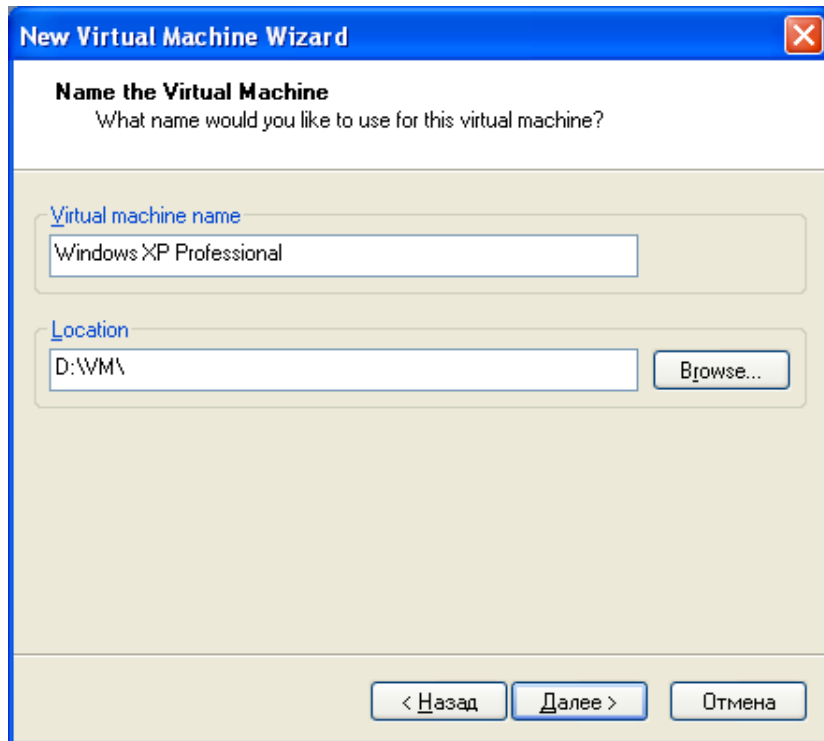
- 3. Вікно вибору операційної системи, яка буде встановлюватися на ВМ.



Список доступних ОС достатньо обширний і включає всі найбільш популярні ОС на даний час. Хоча, навіть якщо Ви не знайдете в даному списку тої ОС, яку хочете встановити (наприклад одного з численних дистрибутивів Linux), це не означає, що встановити її на ВМ неможливо. Присутність ОС в списку означає, що вона протестована на працездатність в середовищі VMware 5.0 та існують спеціальні налаштування оптимізації для ефективнішої роботи даної ОС. Якщо ж ОС відсутня у даному списку, то можна вибрати на даному етапі ОС “схожу” до бажаної (для Linux, наприклад – *Other Linux*). Крім того тут є пункт *Other* – варіант для довільної ОС.

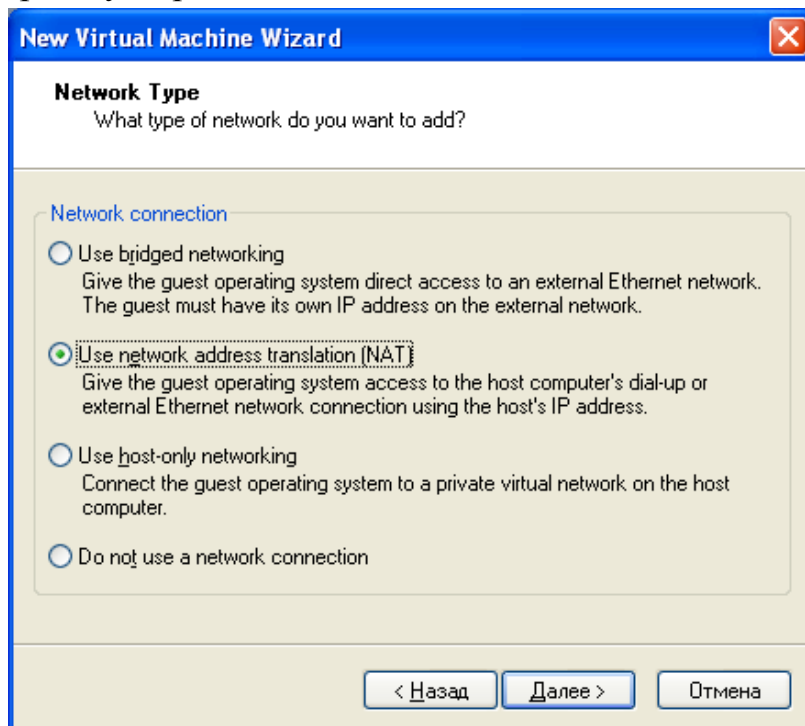
Ми оберемо пункт *Microsoft Windows*, а серед ОС – *Windows XP Professional*. Тиснемо *Далее*.

4. Ім'я та розміщення ВМ.



В даному вікні необхідно ввести ім'я майбутньої VM та каталог, в якому будуть розміщені всі файли віртуальної машини.

5. Вибір типу мережі.



Тут нам пропонується обрати тип мережі з чотирьох можливих:

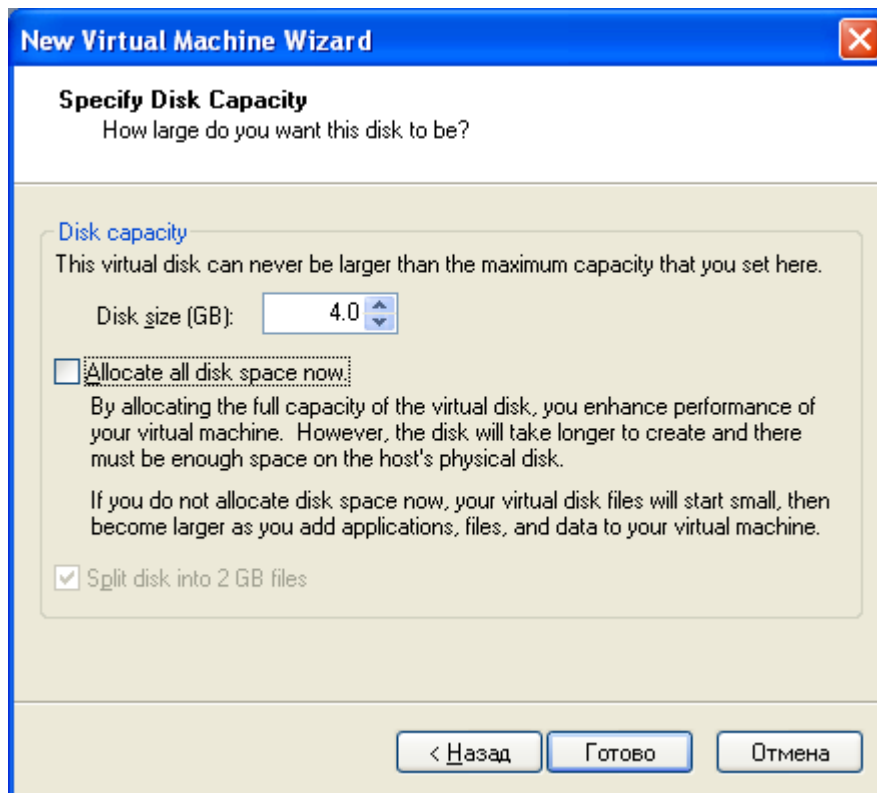
- а) *Bridged networking* (мостове з'єднання). Цей тип з'єднання надає VM доступ до зовнішньої мережі. При цьому VM ідентифікується власною IP-адресою та розглядається зовні, як окремий незалежний комп'ютер. Це досягається тим, що

створюється віртуальний комутатор, до якого під'єднується хост та всі ВМ, що використовують мостове з'єднання. Таким чином, утворюється віртуальна підмережа, яка стає частиною зовнішньої мережі, до якої під'єднаний хост.

- b) *NAT – Network Address Translation* (трансляція мережевої адреси). Надає можливість ВМ доступатися до зовнішньої мережі та Інтернет, але під виглядом хоста, тобто з його IP-адресою. NAT використовує трансляцію адрес вихідного трафіка. Тобто коли пакет даних пересилається з ВМ, то мережна адреса ВМ в ньому замінюється на адресу хост-машини. При цьому запит зберігається в таблиці запитів. Отримані відповіді від віддалених систем звіряються з цією таблицею і якщо знаходиться відповідність, то повідомлення перенаправляється до ВМ, при цьому відбувається зворотна заміна адреси хост-машини на адресу ВМ.
- c) *Host-only networking* (мережа лише з хостом). Цей спосіб дозволяє приєднати дану ВМ до віртуальної мережі (віртуального комутатора), в яку входить хост та інші ВМ, але яка є ізольованою від зовнішньої (реальної) мережі, до якої належить хост. Даний спосіб під'єднання надає можливість створити віртуальну мережу з кількох ВМ, на якій пізніше можна проводити лабораторні роботи по адмініструванню мереж.
- d) *No network* – відсутність мережевого з'єднання у ВМ.

Вибір *NAT* на даному етапі дозволить нам в майбутньому приєднати ВМ до віртуальної мережі та налаштувати доступ в Інтернет, використовуючи хост як шлюз.

6. Вибір розміру жорсткого диску.



В даному вікні можна ввести розмір віртуального жорсткого диску, що буде використовуватися ВМ. Буде створено спеціальний файл (або кілька файлів), в який насправді буде записуватися вся інформація. Можна обрати опцію *Allocate all disk space now*, що призведе до моментального виділення пам'яті на жорсткому диску хоста під зберігання всього об'єму віртуального жорсткого диску. Це дещо підвищить швидкодію ВМ. В іншому випадку місце під віртуальний жорсткий диск буде виділятися по мірі його заповнення.

Для виконання лабораторних робіт нам цілком достатньо диску об'ємом 4 Gb.

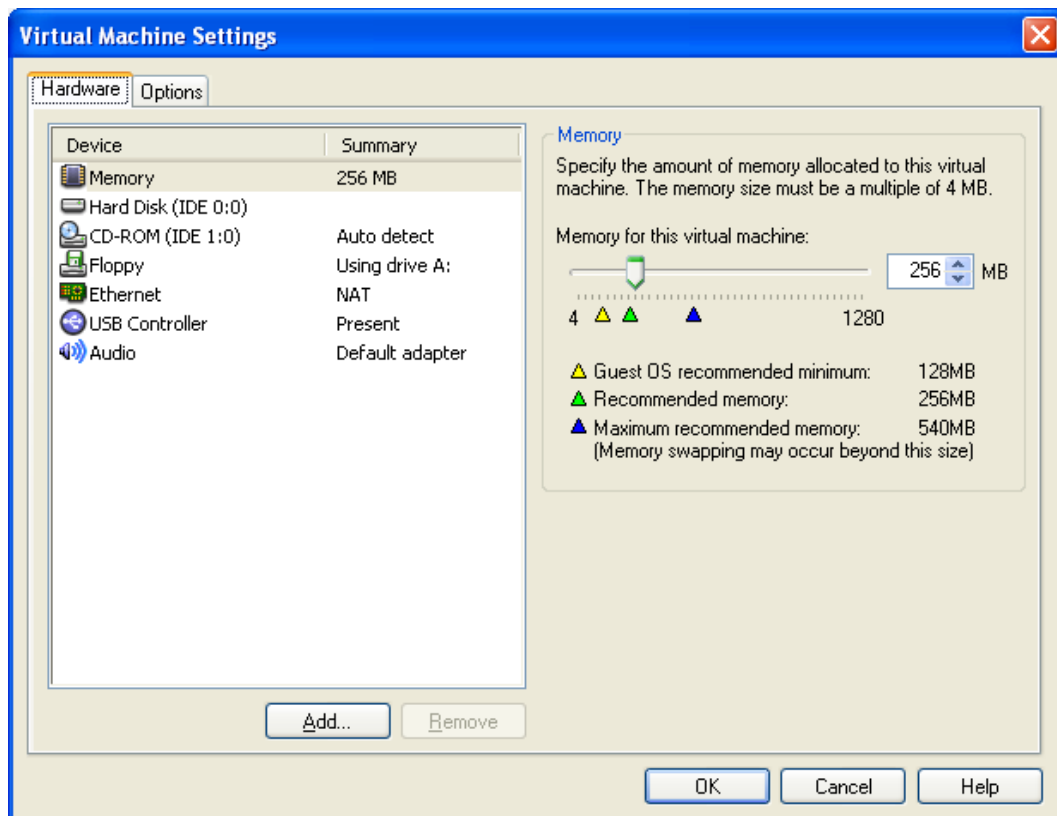
Тиснемо *Готово* – ВМ створено.

2. Налаштування віртуальної машини.

Щоб змінити налаштування ВМ необхідно зайти у меню *VM => Settings*.

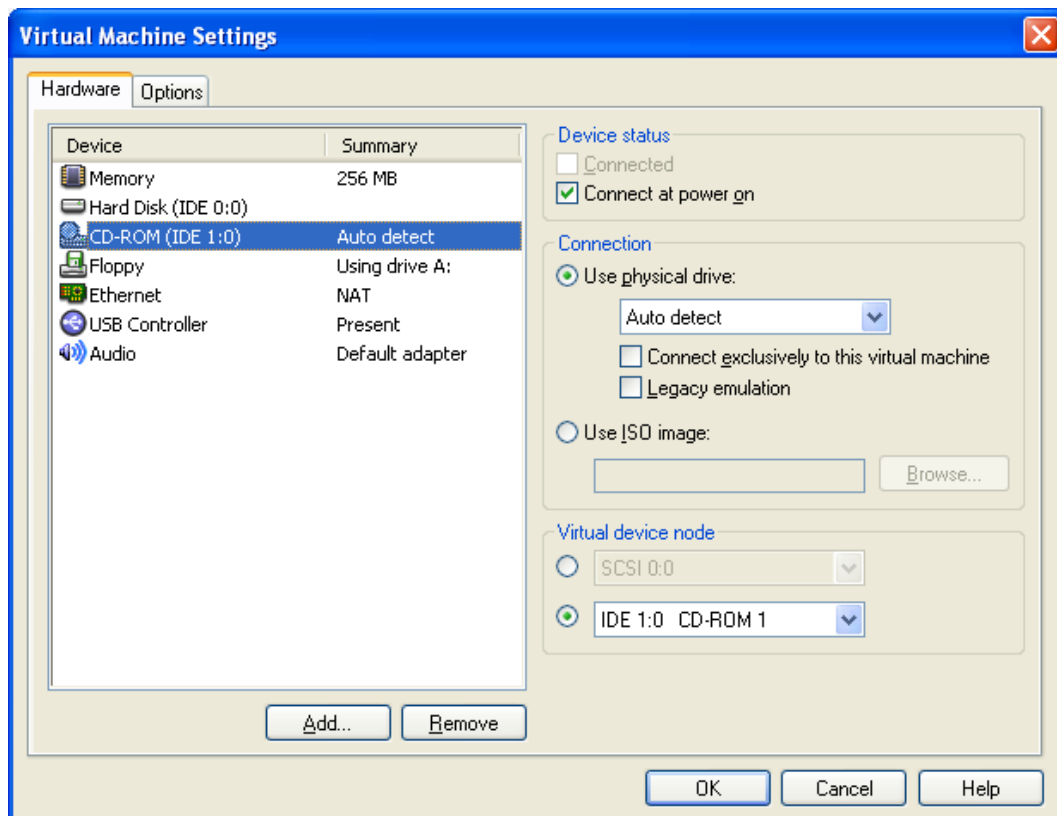
Розглянемо пункти закладки *Hardware*, які можуть нам знадобитися:

1. **Memory.** В цьому вікні можна визначити об'єм оперативної пам'яті, що буде доступний віртуальній машині.



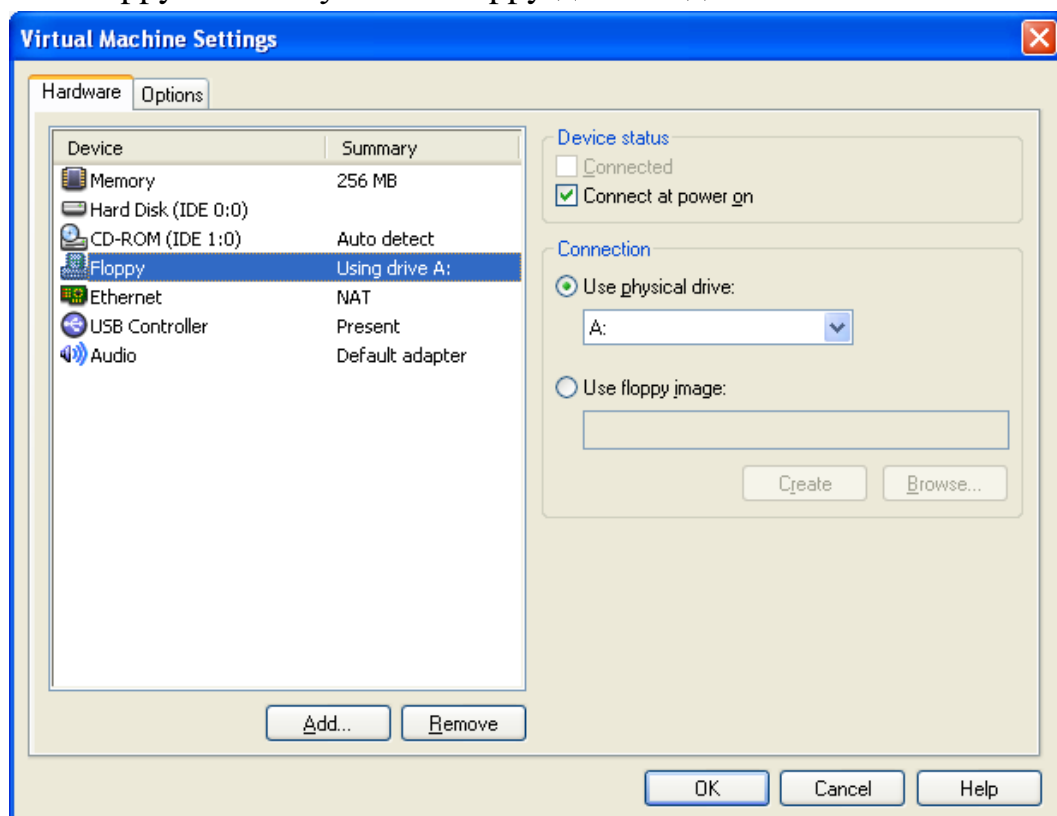
Звернемо увагу на те, що оперативну пам'ять ВМ захоплює динамічно по мірі заповнення. Крім того розмір оперативної пам'яті для ВМ не обмежується кількістю оперативної пам'яті в реальній машині. Можна виставити таку кількість, що перевищує реальну, але при цьому нестача буде компенсуватися використанням файлу підкачки, що значно сповільнить роботу як ВМ, так і хоста в цілому.

2. *Hard Disk*. В цьому пункті відображені властивості віртуального жорсткого диску. Також тут є можливість виконання дефрагментації диску.
3. *CD-ROM*. Це вікно управління дисководом для оптичних дисків.



Корисною можливістю є під'єднання в ролі оптичного диска – ISO-образу диска. Для цього необхідно в радіо-групі *Connection* обрати кнопку *Use ISO image*, та в полі ввести шлях до файлу-образу.

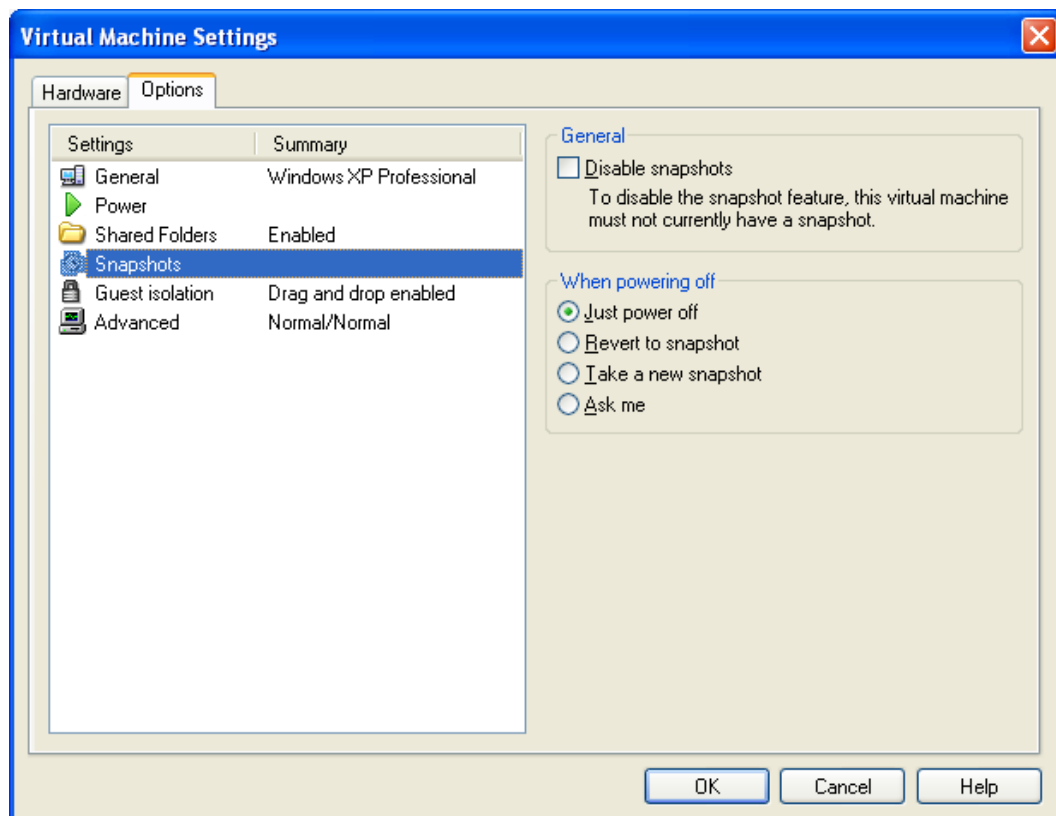
4. Флорру. Налаштування Флорру-дисководу.



Аналогічно як і для CD-ROMa, для Floppy можна зарезервувати фізичний дисківід або файл-образ дискети.

5. Ethernet. Це вікно призначене для налаштування мережі. Тут можна змінити тип мережі ВМ на один з доступних, описаних в підпункті 5 попереднього пункту цієї лабораторної роботи.
6. *USB Controller*. В цьому підпункті можна ввімкнути або вимкнути автоматичне під'єднання пристроїв USB до ВМ коли вона є активною (знаходиться в фокусі).
7. *Audio*. В даному підпункті можна змінити налаштування звукової карти ВМ.

Перейдемо на закладку *Options*. Тут для нас потрібним є пункт *Snapshots*.

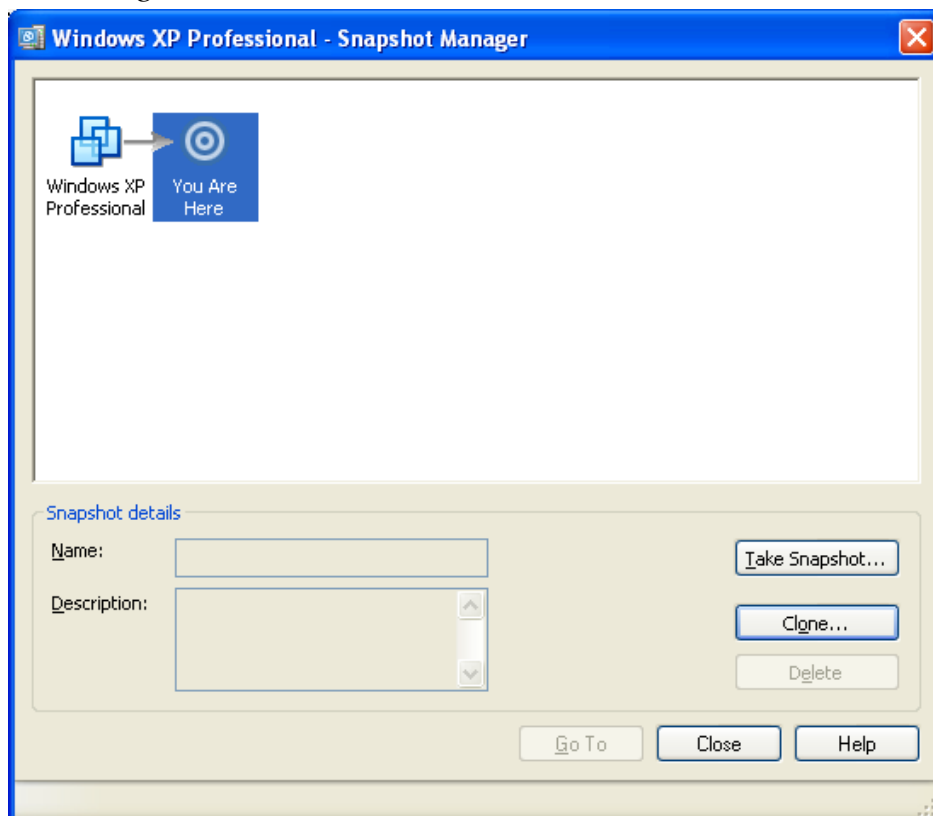


Snapshot (моментальний знімок) – це збережений моментальний стан ВМ, з метою наступного відновлення її до цього стану.

В даному вікні можна ввести налаштування для автоматичного відновлення системи до початкового стану після вимкнення ВМ. Зокрема в радіогрупі *When powering off* можна вказати дії, що виконуватимуться при вимкненні ВМ:

- 1) Просто вимкнути.
- 2) Повернути ВМ до попереднього стану. Цей пункт дуже добре підходить при виконанні лабораторних робіт студентами на ВМ. Після закінчення роботи студент просто вимикає ВМ і вона сама, без допомоги системного адміністратора, повертається до попереднього стану, і вже готова для виконання на ній наступної лабораторної роботи.
- 3) Зберегти “моментальний знімок” системи. Якщо вибрати цей пункт, то після закінчення роботи користувача всі зміни збережуться в окремий “знімок”. Це дозволяє в будь який момент повернутися до довільного етапу в роботі користувача, оскільки кожен етап в кінці роботи зберігається.
- 4) Видати запит для вибору дії. При виборі такої опції, щоразу при вимкненні ВМ користувача буде запитано, яку з трьох попередніх дій виконати.

Крім пункту в налаштуваннях є спеціальний “менеджер знімків” (*Snapshot Manager*). Щоб у нього потрапити потрібно зайти у *VM => Snapshot => Snapshot Manager*.



Snapshot Manager створений для управління “знімками”. В ньому можна створювати, переглядати, видаляти “знімки” та переходити до певного “знімка”. Крім того тут доступна можливість “клонування” ВМ – створення повної копії ВМ, яку можна перенести на інший комп’ютер та завантажити

там в середовищі віртуальних машин VMware 5.0. За цю можливість відповідає кнопка *Clone*.

Налаштування ВМ завершено. Тепер, ми маємо готове віртуальне середовище, на яке можемо встановити операційну систему. Навіть після встановлення ОС на ВМ, ми можемо змінювати більшість налаштувань: додавати та знищувати пристрої, змінювати розмір оперативної пам'яті та інше.

6. Оформити звіт про виконання лабораторної роботи у якому подати:

- а. тему, мету та завдання лабораторної роботи;
- б. прізвище, ініціали та назву групи студента, що виконав роботу;
- с. короткий опис проведених налаштувань;
- д. висновки.

7. Звіт оформити у вигляді файлу з іменем:

“Прізвище”lan4.doc.

ЛАБОРАТОРНА РОБОТА № 5

Налаштування локальної мережі та Інтернет в VMware 5.0 на ОС Windows та Linux

Мета роботи: Навчитись налаштовувати локальну мережу між хостовою ОС та віртуальною машиною. Ознайомитись з сервісом NAT. З допомогою сервісу NAT налаштувати вихід у Інтернет на віртуальній машині, використовуючи реальну машину, як шлюз. Вивчити особливості налаштувань в ОС Windows та Linux на прикладах Windows 7, Windows XP та Fedora Linux4.

1. Вступ.

Здійснити зв'язок між хостом та віртуальною машиною (ВМ) у VMware можна декількома способами:

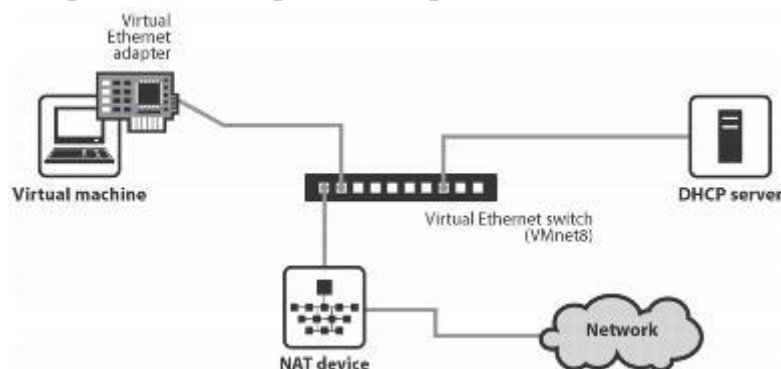
1) Сервіс Shared folders – дозволяє використовувати папку з реальної ОС як мережевий диск у ВМ. На жаль цей сервіс не підтримує всі ОС встановлені на ВМ (зокрема версії Windows до Windows 98 включно та деякі версії Linux).

2) Drag'n'Drop – можливість “перетягувати” файли з віртуальної машини на реальну та навпаки. Цей спосіб теж має обмеження, бо не працює, коли у ВМ встановлено ОС Linux.

3) Створення локальної мережі між хостом та ВМ. Цей спосіб найкращий, оскільки підтримується всіма ОС. Крім того, ми отримуємо не лише можливість переносу файлів, а повноцінну мережу з кількох комп'ютерів, яку можна використати для тестування інших мережевих сервісів.

2. Теоретичні відомості про NAT.

Для створення мережі в VMware скористаємось сервісом NAT (network address translation) – Трансляція мережної адреси.



NAT використовує трансляцію адрес вихідного трафіка, тобто коли пакет даних пересилається з ВМ, то мережна адреса ВМ в ньому замінюється на адресу хост-машини. При цьому запит зберігається в таблиці запитів. Отримані відповіді від віддалених систем звіряються з цією таблицею і якщо знаходиться відповідність, то повідомлення перенаправляється до ВМ, при цьому відбувається зворотна заміна адреси хост-машини на адресу ВМ.

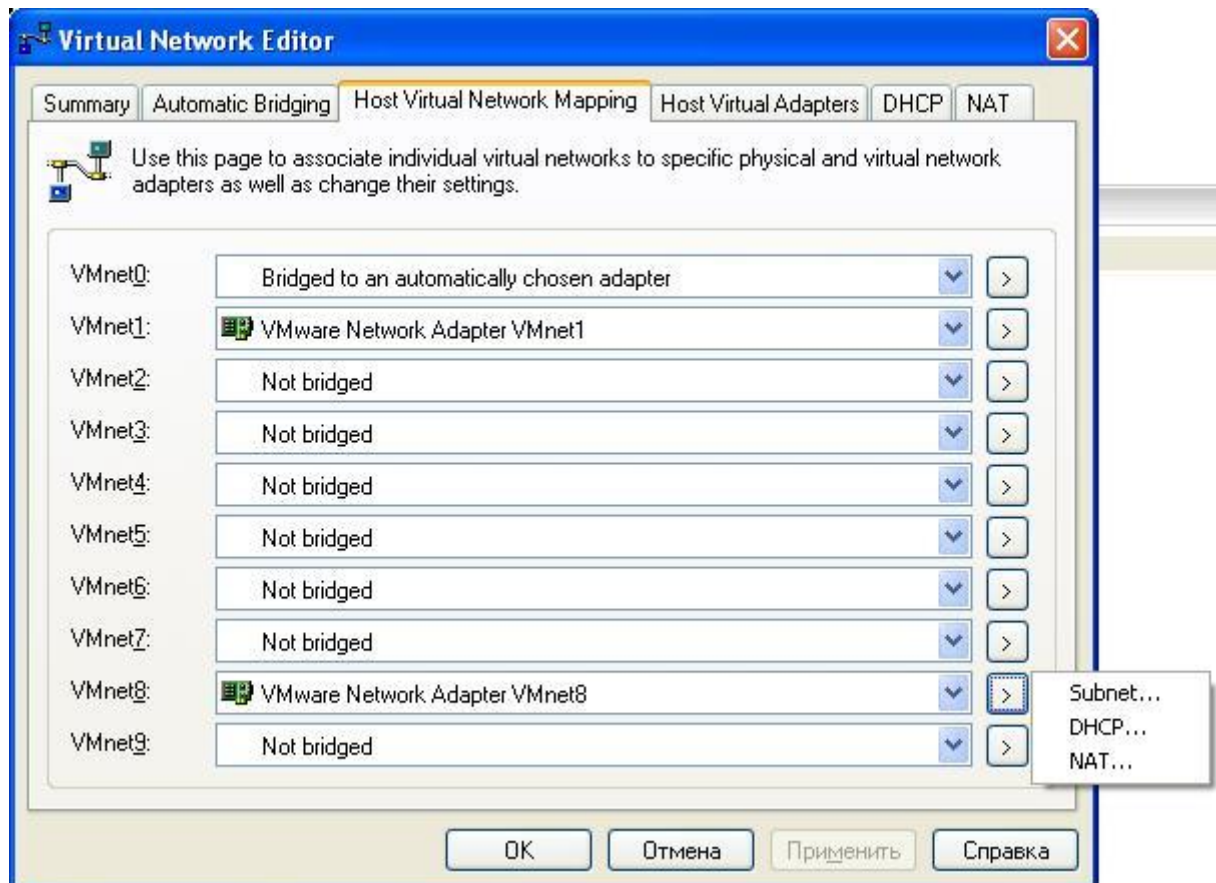
Таким чином ВМ отримує повноцінний доступ до зовнішньої мережі, але в зовнішньому світі вона ідентифікується так само, як відповідна хост-машина.

Хід роботи.

1. Налаштування віртуальної мережевої картки в хостовій ОС.

При встановленні VMware в операційній системі автоматично з'являються дві віртуальні мережеві картки, одна з яких розрахована для забезпечення сервісу NAT. По замовчуванню вона носить назву "VMware Network Adapter VMnet8".

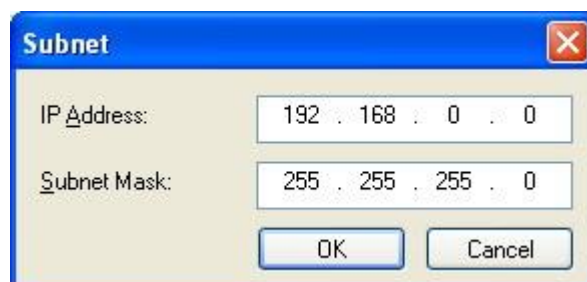
Щоб змінити налаштування віртуальної мережевої картки потрібно зайти у вікні VMware в меню *Edit => Virtual Network Settings*. Переходимо на закладку *Host Virtual Network Mapping*. Нам потрібно ввести налаштування адаптера *VMware Network Adapter VMnet8*. Для цього натискаємо на стрілку і бачимо список з трьох компонентів налаштування адаптера: *Subnet*, *DHCP* і *NAT*.



Спершу заходимо в *Subnet* і вводимо IP-адресу віртуальної мережі та маску підмережі, наприклад:

IP Address: 192.168.0.0

Subnet Mask: 255.255.255.0

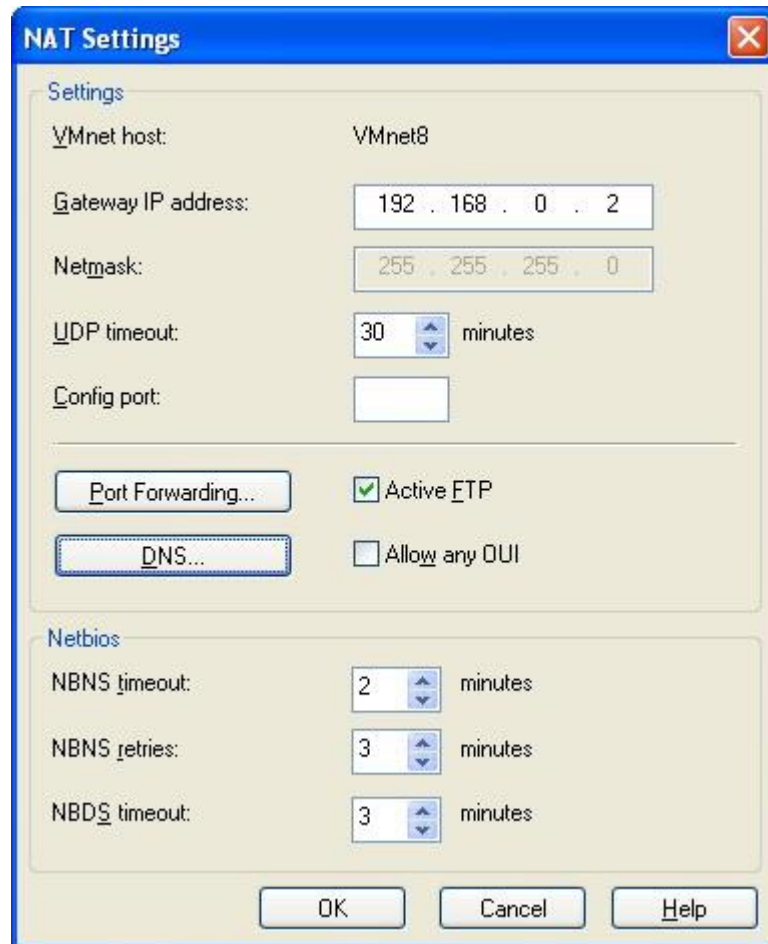


Тиснемо *OK*.

Далі заходимо в *NAT* і вводимо IP-адресу та маску NAT-шлюза, наприклад так:

Gateway IP address: 192.168.0.2

Netmask: 255.255.255.0



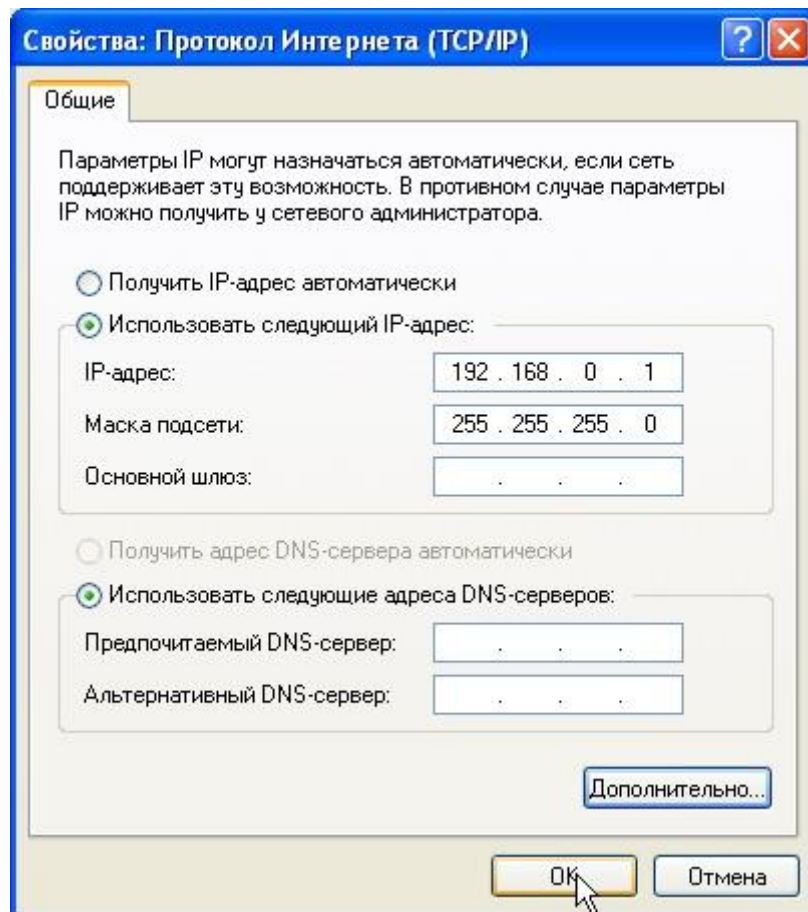
Решту налаштувань можна залишити, як є.

Закриваємо вікно *Virtual Network Settings*.

Тепер переходимо в *Панель управления => Сетевые подключения*. На іконці *VMware Network Adapter VMnet8* викликаємо контекстне меню, вибираємо *Свойства*. У вікні, що з'явилося на закладці *Общие* обираємо *Протокол Интернета (TCP/IP)*, тиснемо *Свойства*. Вводимо IP-адресу хоста, який в майбутньому буде відігравати роль шлюза для ВМ, та маску підмережі.

IP-адрес: 192.168.0.1

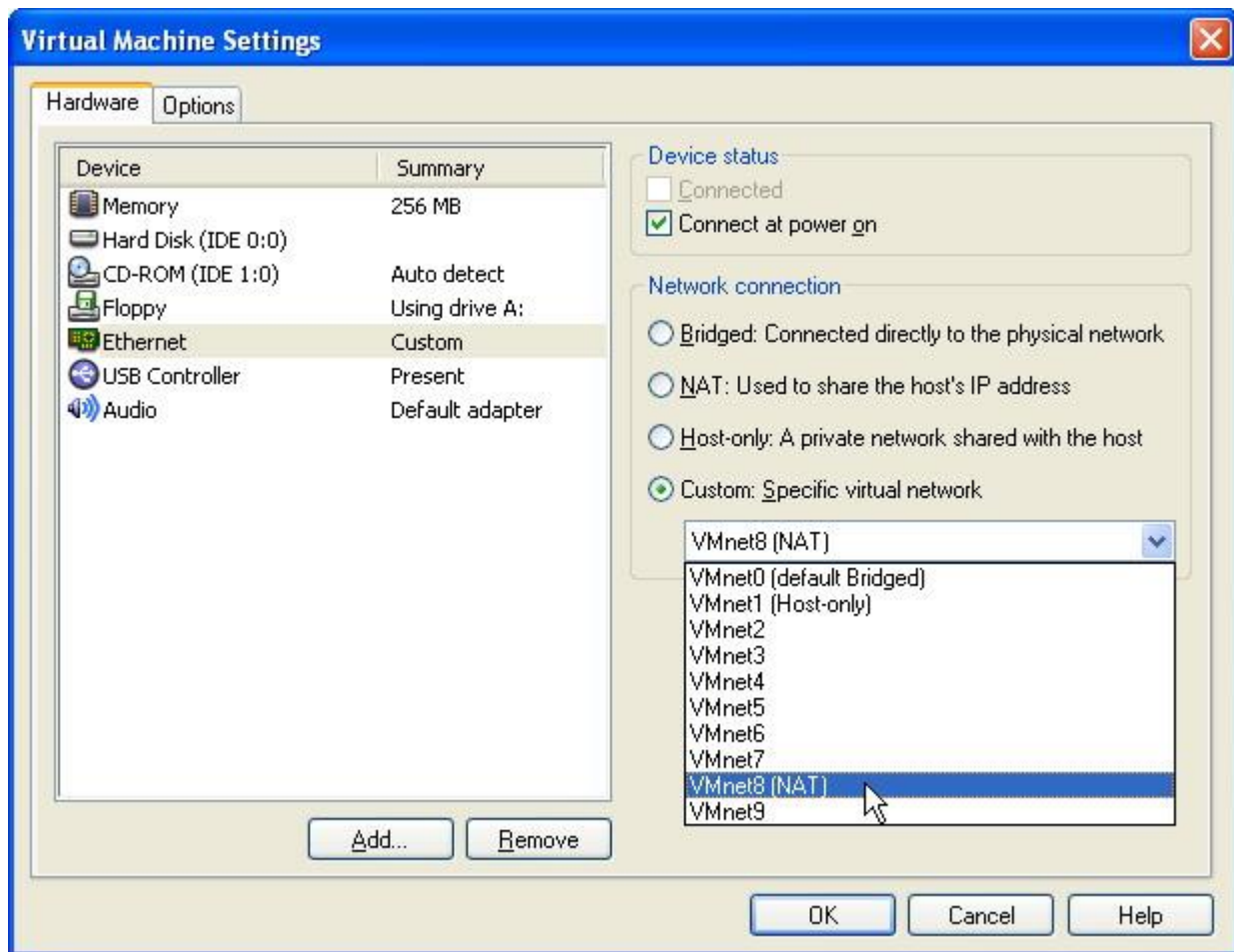
Маска подсети: 255.255.255.0



Адаптер налаштовано.

2. Налаштування мережі в віртуальній машині.

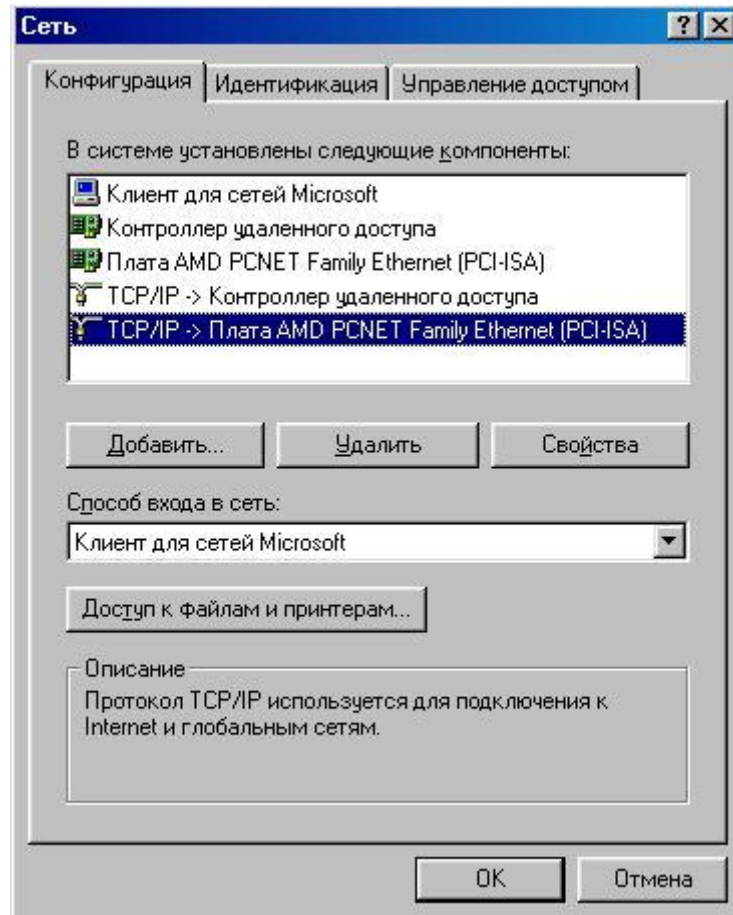
Для початку виберемо віртуальну мережу, до якої буде під'єднуватися наша віртуальна машина. Для цього у вікні VMware вибираємо потрібну віртуальну машину, далі в меню *VM => Settings* на закладці *Hardware* обираємо мережевий адаптер. В налаштуванні мережевого з'єднання (*Network connection*) обираємо *Custom: Specific virtual network*. В випадаючому списку обираємо *VMnet8 (NAT)*.



Всі наступні налаштування потрібно виконувати вже при запусненій ОС у ВМ, тому вони відрізняються для різних ОС. Розглянемо окремо налаштування для Windows 98, Windows XP та Fedora Linux 4.

2.1. Windows 98. потрібно замінити на Windows 7 або 10

Заходимо в *Панель Управлення => Сеть*. На закладці *Конфигурація* обираємо налаштування TCP/IP нашої мережевої плати та тиснемо *Свойства*.

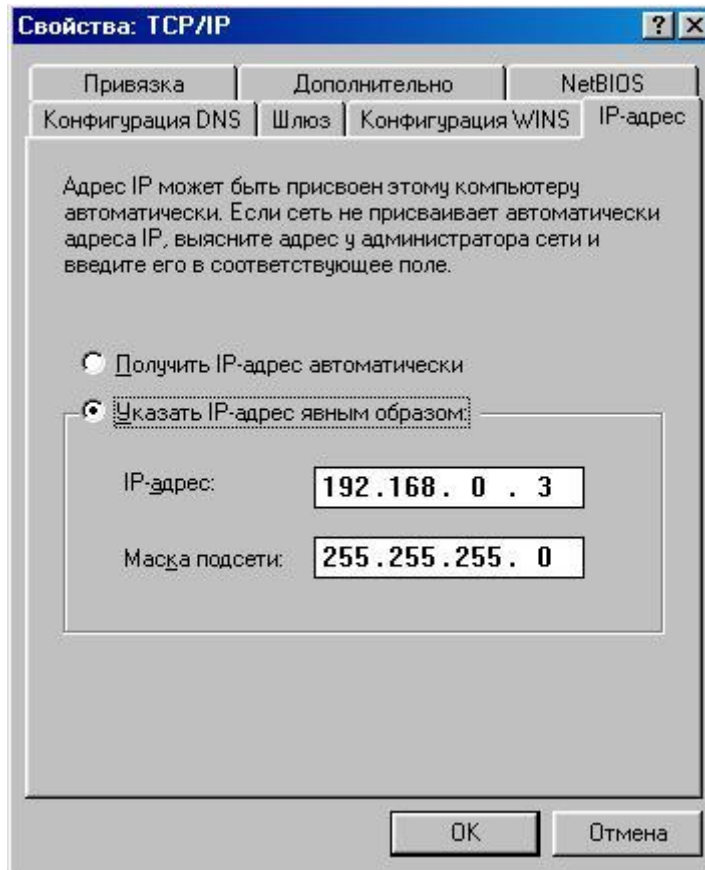


На закладці *IP-адрес* вводимо IP-адресу віртуальної машини та маску підмережі, наприклад:

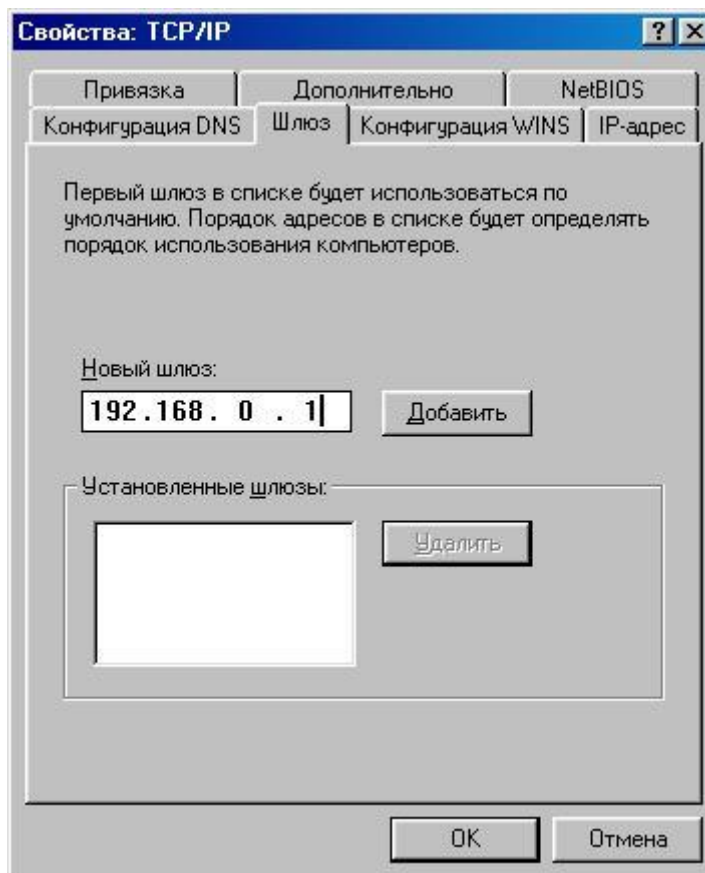
IP-адрес: 192.168.0.3

Маска підсети: 255.255.255.0

Введена IP-адреса, це адреса, за якою віртуальна машина буде ідентифікуватися в віртуальній мережі.

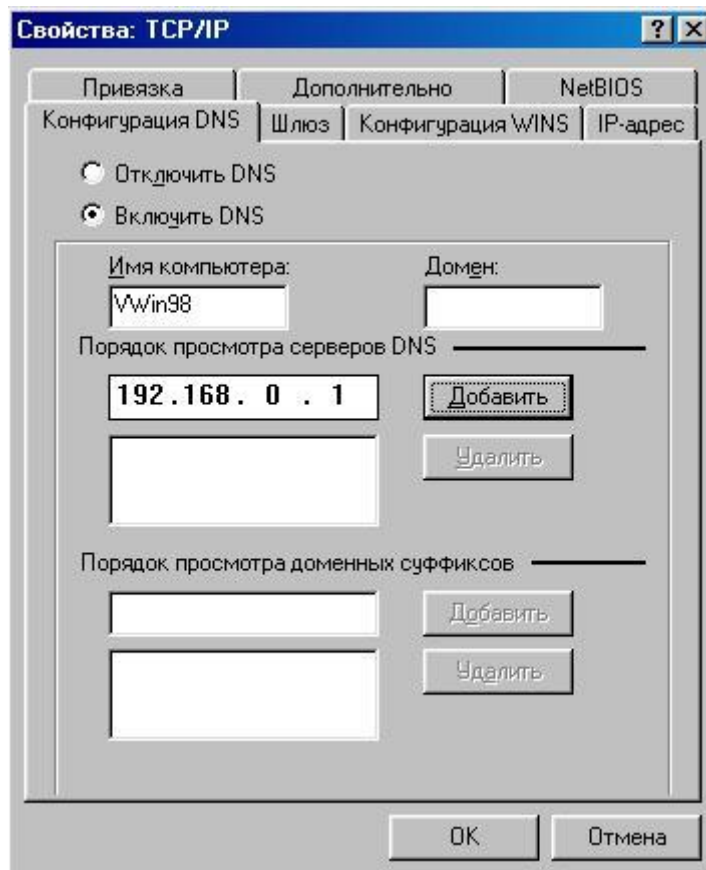


На закладці *Шлюз* вводимо IP-адресу шлюзу 192.168.0.1 та тиснемо *Добавить*.



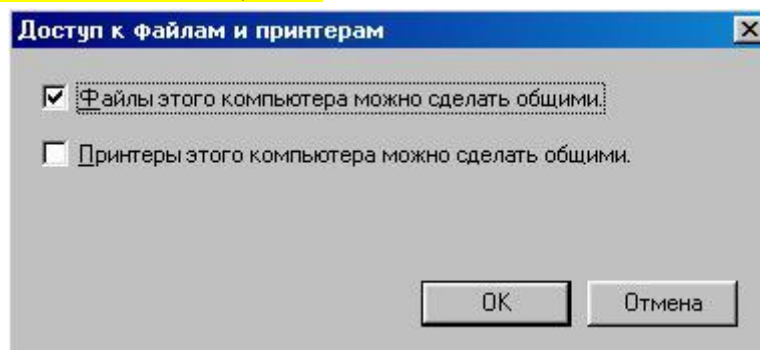
На закладці *Конфигурация DNS* вибираємо *Включить DNS* та заповнюємо поле *Имя компьютера* тут можна ввести таке ім'я, яке вам до

вподоби, його надалі ми використовувати не будемо (але заповнити поле обов'язково). В полі нижче заповнюємо адресу *DNS-сервера* 192.168.0.1 та тиснемо *Добавить*.



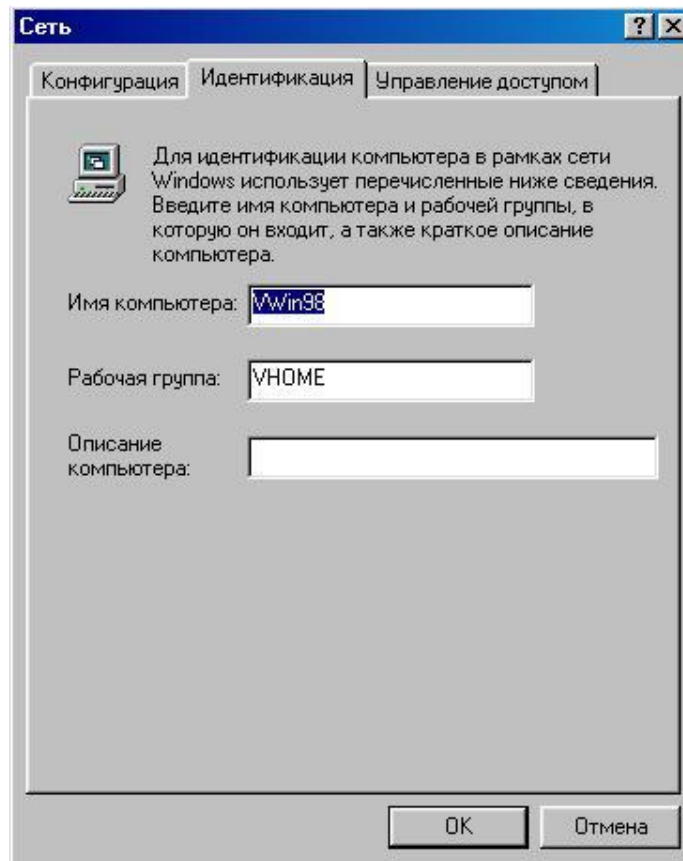
Натискаємо *OK*.

На закладці *Конфигурация*, що перед нами, тиснемо *Доступ к файлам и принтерам*. В вікні що з'явилося ставимо галочку біля *Файлы этого компьютера можно сделать общими*.



Тиснемо *OK*.

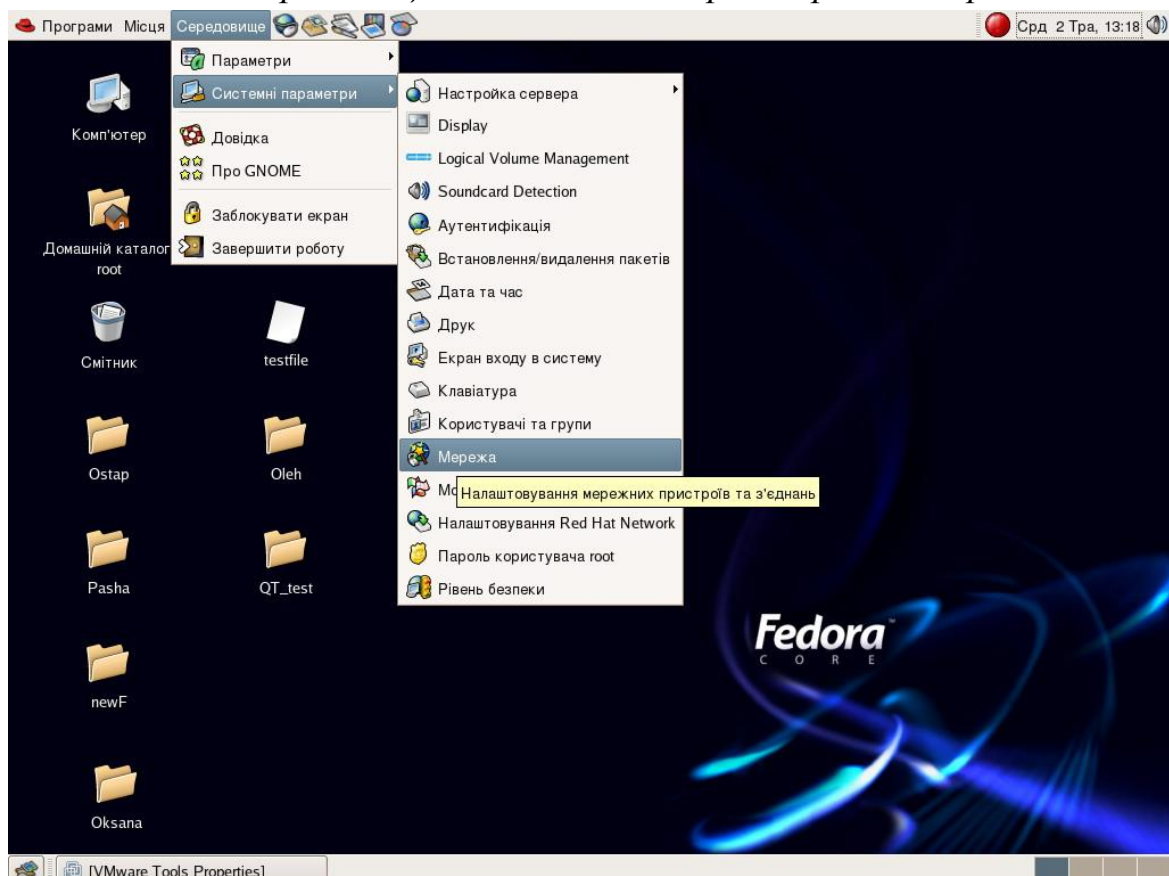
Переходимо на закладку *Идентификация* і вводимо ім'я комп'ютера та робочої групи. Саме з цим іменем буде існувати наша ВМ в мережевому оточенні всіх інших машин під'єднаних до даної віртуальної мережі.



Тиснемо ОК. Перезавантажуємо ОС. Мережу налаштовано.

2.2. Fedora Linux 4.

Заходимо в *Середовище* => *Системні параметри* => *Мережа*.

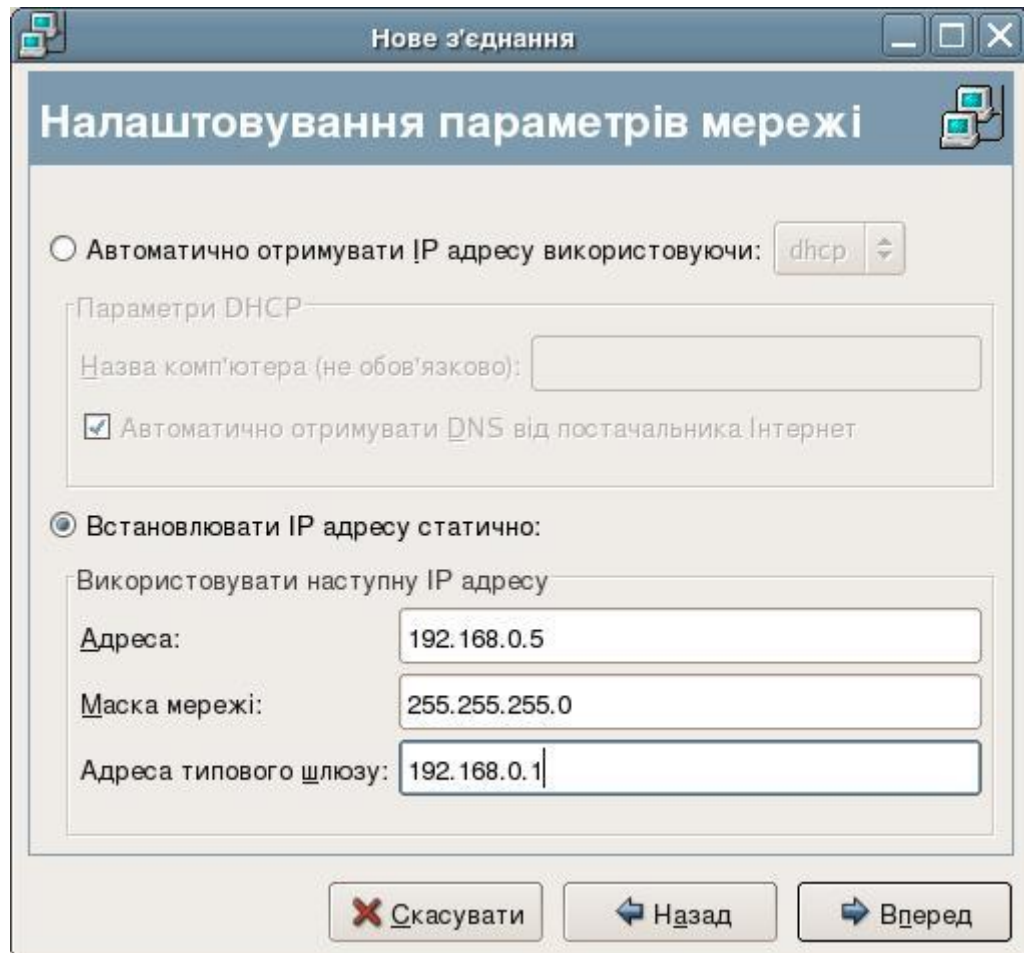


Тут, перебуваючи на закладці *Пристрої*, натискаємо кнопку *Створити*. Далі обираємо *З'єднання Ethernet*. В наступному вікні обираємо нашу існуючу мережеву картку. В третьому вікні обираємо *Встановлювати IP-адресу статично* та вводимо майбутню IP-адресу нашої ВМ, маску мережі та адресу шлюзу. Наприклад:

Адреса: 192.168.0.5

Маска мережі: 255.255.255.0

Адреса типового шлюзу: 192.168.0.1



Нове з'єднання

Налаштовування параметрів мережі

☐ Автоматично отримувати IP адресу використовуючи: dhcp

Параметри DHCP

Назва комп'ютера (не обов'язково):

☒ Автоматично отримувати DNS від постачальника Інтернет

☒ Встановлювати IP адресу статично:

Використовувати наступну IP адресу

Адреса: 192.168.0.5

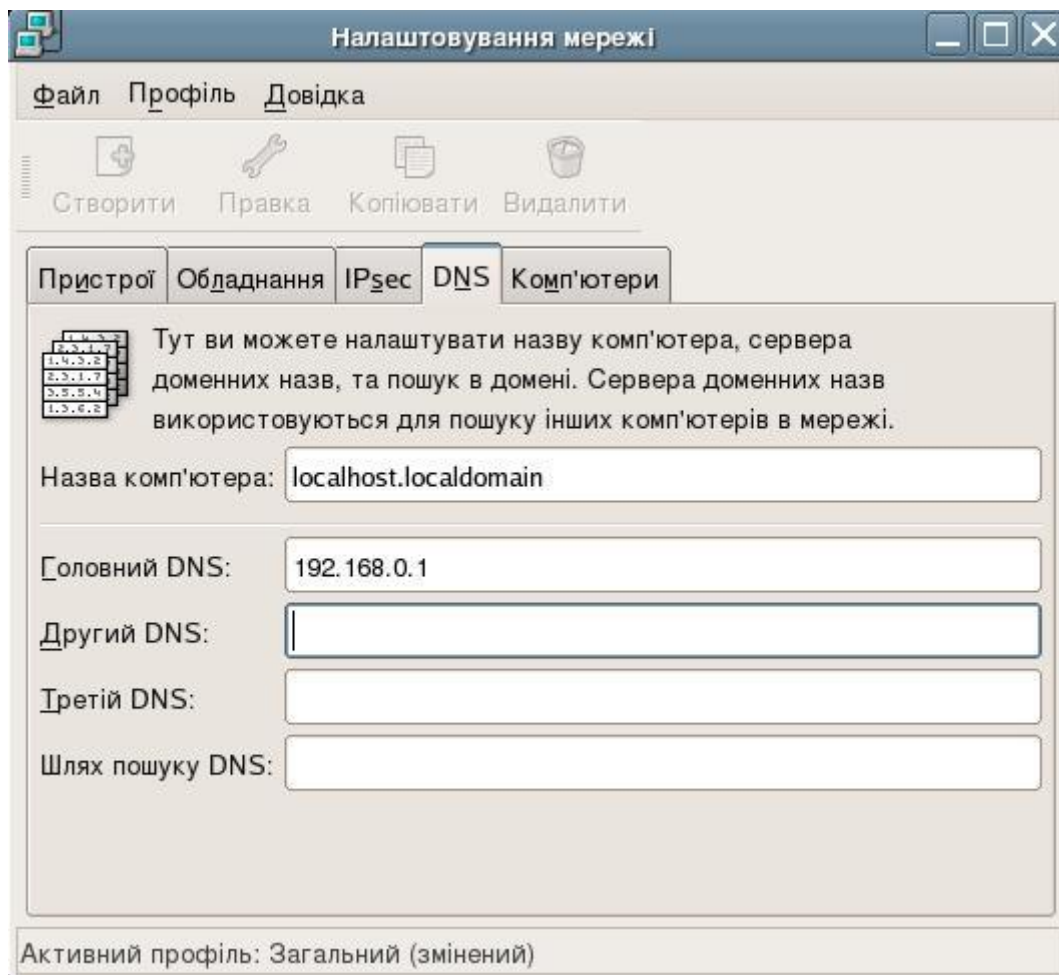
Маска мережі: 255.255.255.0

Адреса типового шлюзу: 192.168.0.1

Скасувати Назад Вперед

Далі підтверджуємо наші налаштування. В головному вікні налаштувань мережі тиснемо *Активувати*.

Переходимо на закладку *DNS*. В полі *Головний DNS* заповнюємо 192.168.0.1 .



Закриваємо вікно налаштувань мережі. На запит “*Бажаєте зберегти зміни?*” відповідаємо *Так*.

Знаходимо в каталозі */etc* файл *hosts*, відкриваємо його в текстовому редакторі і перевіряємо наявність рядка

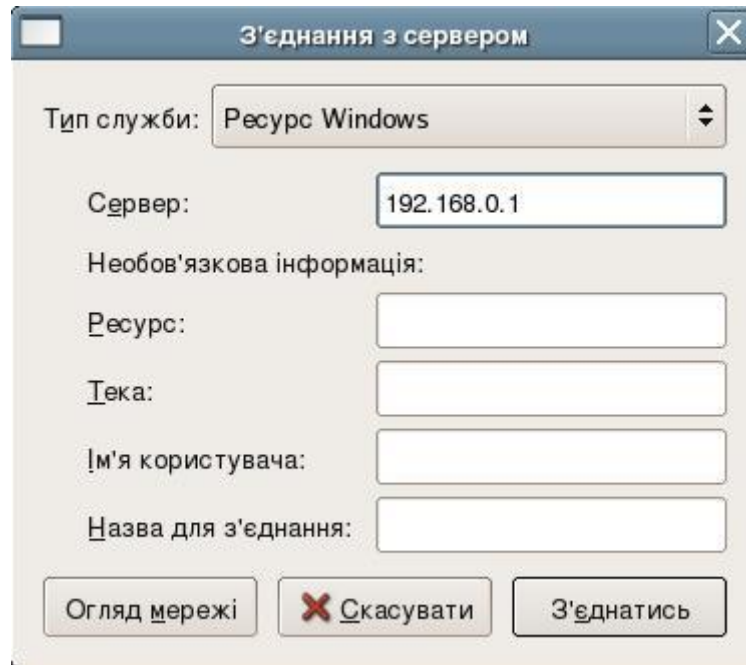
```
127.0.0.1 localhost.localdomain localhost
```

Якщо його немає, то дописуємо.

Все, мережу налаштовано. Перезавантажуємо ОС і можемо користуватися мережею.

Тепер, щоб доступитися до папок з відкритим доступом на певному комп'ютері з нашої віртуальної мережі, потрібно під'єднати мережевий диск. Це робиться так:

- 1) Заходимо у *Місця => З'єднання з сервером*.
- 2) У вікні, що з'явилося в полі *Тип служби* обираємо *Ресурс Windows*.
- 3) В полі *Сервер* записуємо IP-адресу комп'ютера в нашій віртуальній мережі, до якого хочемо під'єднатись (це може бути як хост-система так і одна з інших VM), наприклад 192.168.0.1.
- 4) Тиснемо *З'єднатись*.



З'являється вікно зі всіма папками, до яких відкритий доступ. Крім того, створюється мережевий диск, зв'язаний з "віддаленим" комп'ютером, який можна використовувати і надалі для обміну файлами між віртуальною машиною та цим комп'ютером. В ролі "віддаленого комп'ютера" може виступати і реальна ОС, і одна зі створених раніше віртуальних машин.

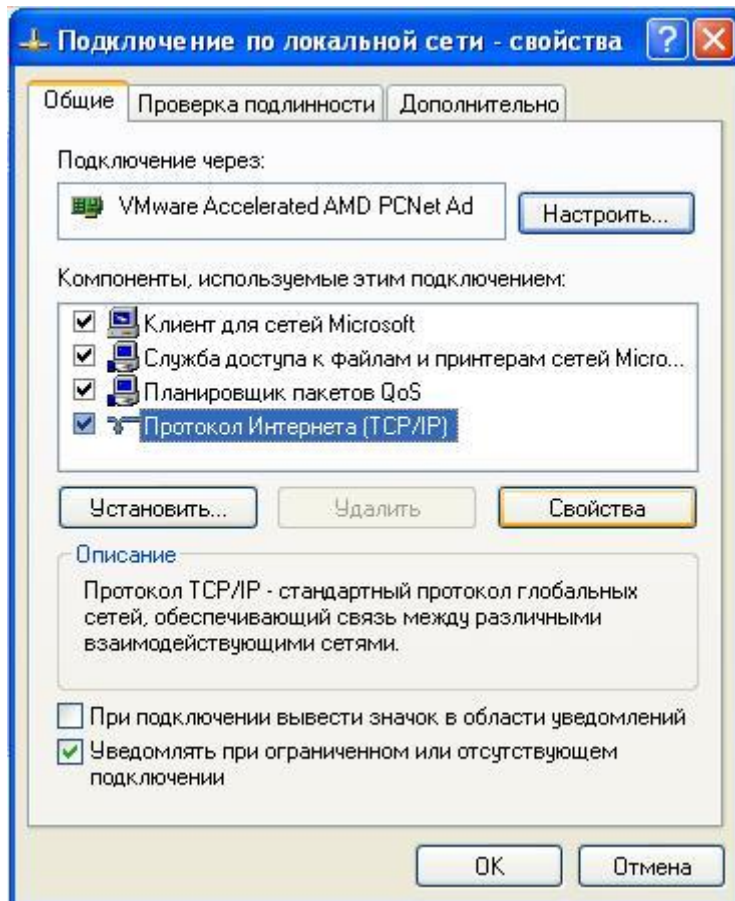
Щоб відкрити доступ до папок ВМ під управлінням Fedora Linux для інших комп'ютерів мережі потрібно налаштувати сервер Samba. В даному описі ми не розглядаємо цієї можливості, оскільки різноманітні налаштування сервера Samba заслуговують бути темою окремої лабораторної роботи.

2.3. Windows XP Professional.

Заходимо в *Панель Управління => Сетевые подключения*. Клікаємо правою кнопкою на іконці мережевого під'єднання, обираємо *Свойства*.

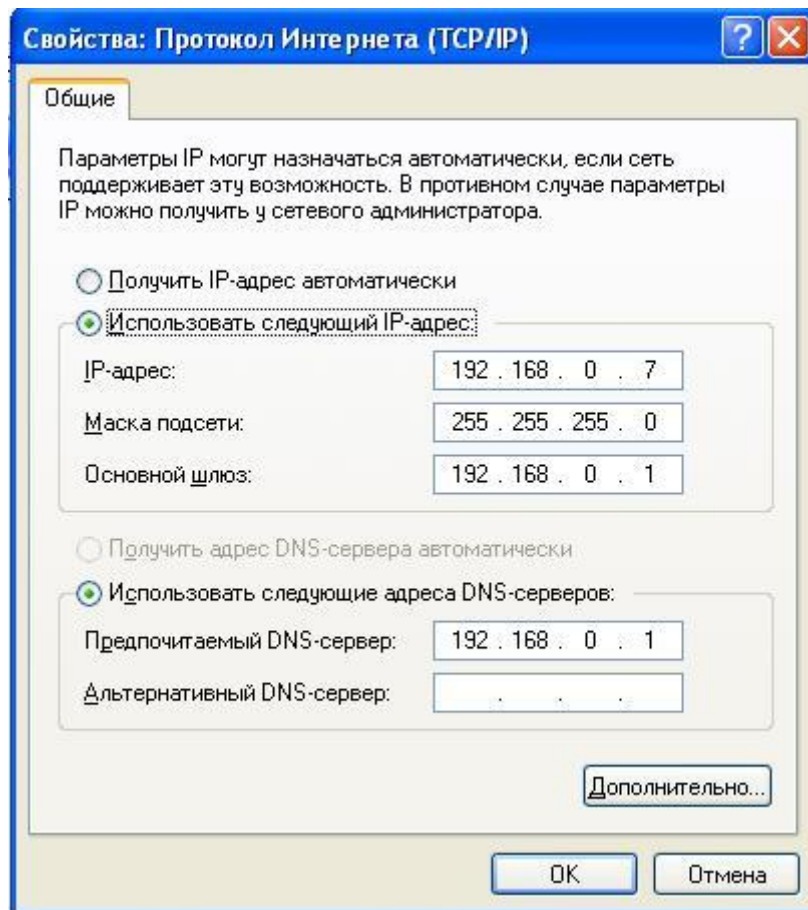
По-перше, на закладці *Дополнительно* вимикаємо "Брандмауер Windows" – це обов'язково. Якщо це необхідно, можна скористатися брандмауерами сторонніх розробників.

На закладці *Общие* обираємо *Протокол Интернета (TCP/IP)*, тиснемо *Свойства*.



У вікні, що з'явилося обираємо *Использовать следующий IP-адрес*. В полі *IP-адрес* вводимо адресу, котра ідентифікуватиме нашу ВМ у віртуальній мережі, наприклад 192.168.0.7. Заповнюємо поля таким чином:

<i>IP-адрес:</i>	192.168.0.7
<i>Маска подсети:</i>	255.255.255.0
<i>Основной шлюз:</i>	192.168.0.1
<i>Предпочитаемый DNS-сервер:</i>	192.168.0.1

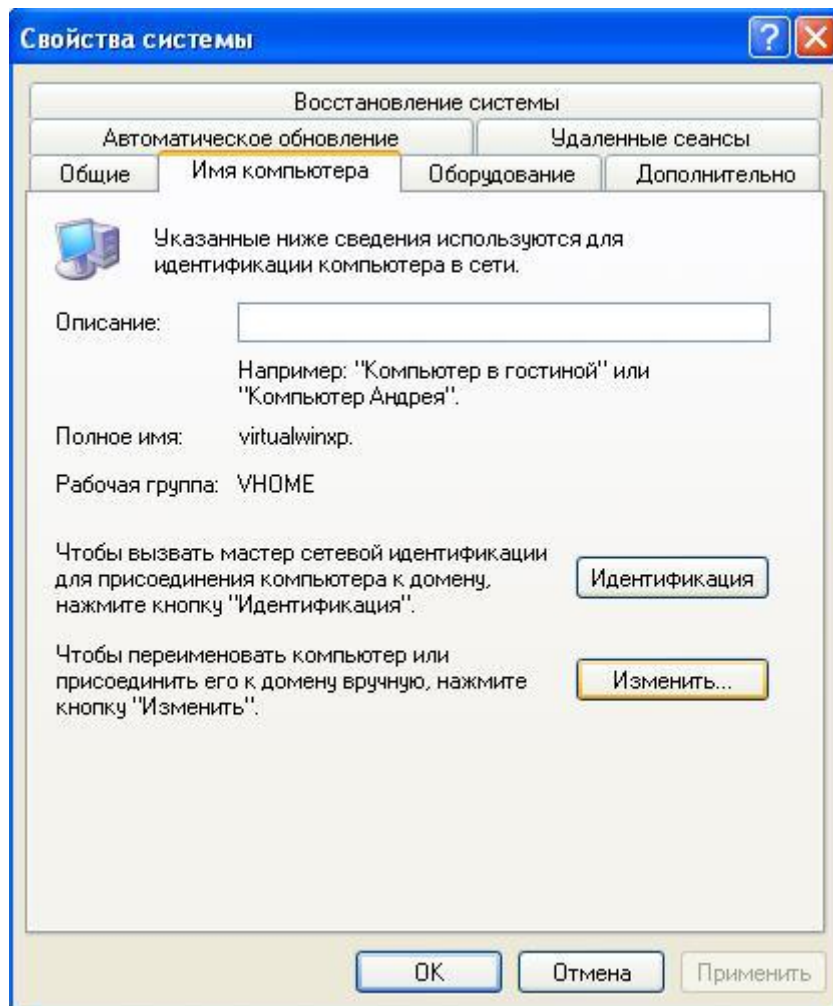


Тиснемо *ОК*. Знову *ОК*.

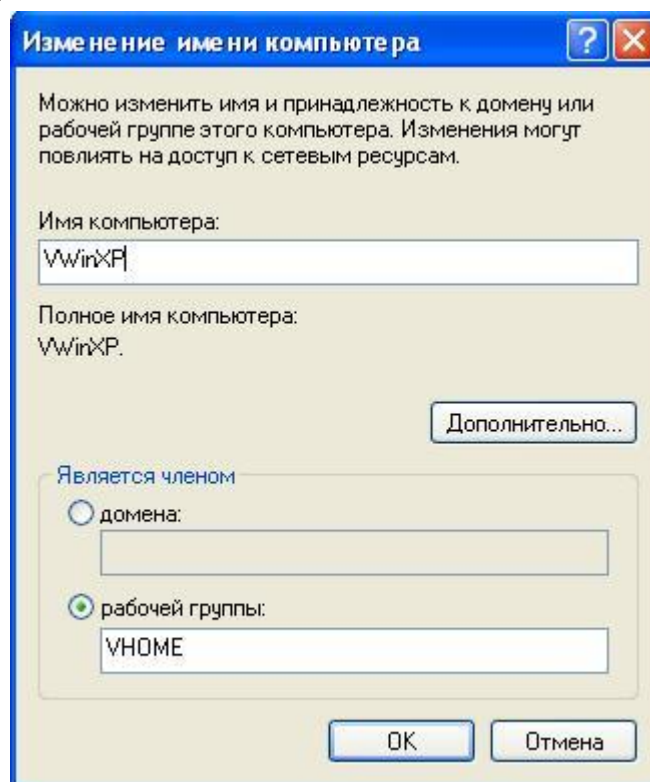
Тепер введемо налаштування ідентифікації ВМ в мережі.

Клацаємо правою кнопкою по значку *Мой компьютер*, далі *Свойства*.

На закладці *Имя компьютера* тиснемо *Изменить*.



У вікні вводимо майбутнє ім'я віртуального комп'ютера та робочу групу, до якої він буде належати.

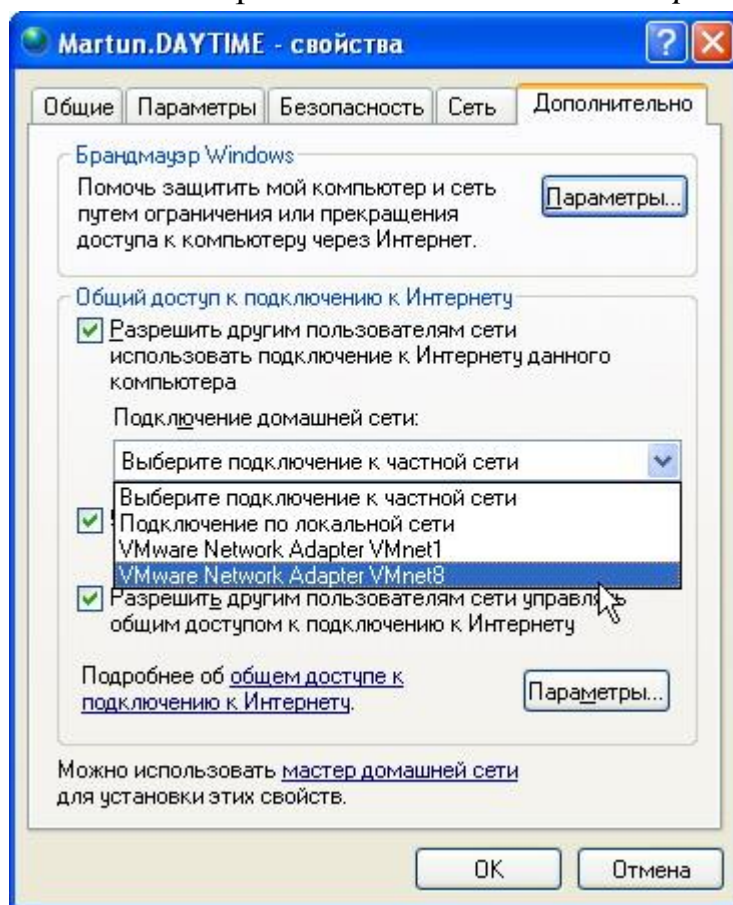


Зберігаємо зміни. Перезавантажуємо ОС. Мережу налаштовано.

3. Налаштування шлюзу в Інтернет.

Після того, як успішно налагоджено мережу між ВМ та хостовою ОС, залишилося ввести потрібні налаштування для того щоб ВМ могла виходити в Інтернет, використовуючи хост, як шлюз.

На хостовій ОС заходимо в *Панель управління => Сетевые подключения*. Обравши іконку під'єднання, через яке хост виходить у Інтернет вибираємо *Свойства*. На закладці *Дополнительно* ставимо галку, дозволяючи іншим користувачам мережі використовувати дане під'єднання. В списку *Подключение домашней сети* обираємо *VMware Network Adapter VMnet8*.



Тиснемо *OK*.

Вмикаємо віртуальну машину – інтернет повинен працювати з налаштуваннями броузера “по замовчуванню”.

4. Оформити звіт про виконання лабораторної роботи у якому подати:

- тему, мету та завдання лабораторної роботи;
- прізвище, ініціали та назву групи студента, що виконав роботу;
- опис налаштувань локальної мережі між хостовою ОС та віртуальною машиною. Коротко описати функції сервісу NAT;

- d) короткий опис налаштувань локальної мережі у різних операційних системах;
 - е) висновки.
5. Звіт оформити у вигляді файла з іменем:
“Прізвище”lan5.doc.

ЛАБОРАТОРНА РОБОТА № 6

Перевірка налаштувань стеку протоколів TCP/IP на ОС Windows та Linux

Мета роботи: Вивчити способи діагностики налаштувань стеку протоколів TCP/IP та працездатності мережі. Отримати дані про налаштування протоколів. Особливості налаштувань перевірити на прикладі ОС Windows та Linux.

Теоретичні відомості

Для організації передавання інформації у пакетних мережах створено стек протоколів **TCP/IP** (Transmission Control Protocol - протокол керування передачею / Internet Protocol – міжмережний протокол). Цей **стек протоколів** використовують для отримання доступу до ресурсів мережі Інтернет а також він дозволяє організувати мережу масштабу підприємства і зв'язувати комп'ютери, що працюють під керуванням різних операційних систем. Протокол підтримується майже всіма операційними системами і майже всі великі мережі базуються на TCP / IP.

Стан комп'ютера в мережі визначає його логічна IP-адреса. Реалізація **TCP/IP** дозволяє для вузла **TCP/IP** використовувати статичну IP-адресу або отримати IP-адресу автоматично за допомогою **DHCP-сервера** (Dynamic Host Configuration Protocol – протокол динамічної конфігурації хоста). Остання технологія характерна для локальних мереж.

За замовчуванням комп'ютери клієнтів, що працюють під управлінням ОС **Windows** або **Linux**, отримують інформацію про налаштування протоколу **TCP/IP** автоматично від служби **DHCP**. Однак навіть у тому випадку, якщо в мережі доступний **DHCP-сервер**, необхідно призначити статичну IP-адресу для окремих комп'ютерів в мережі.

Якщо служба **DHCP** недоступна, можна налаштувати **TCP/IP** для використання статичної IP-адреси.

Для кожної плати мережного адаптера в комп'ютері, що використовує **TCP/IP**, можна встановити IP-адресу, маску підмережі та шлюз за замовчуванням.

IP-адреса – це логічна 32-бітна адреса, яка ідентифікує TCP / IP вузол. Кожному мережному адаптеру в комп'ютері з запущеним протоколом TCP / IP співставляється унікальна IP-адреса, наприклад, 192.168.0.28. Адреса має дві частини: ID мережі, який ідентифікує всі вузли в одній фізичній мережі та ID вузла, який ідентифікує вузол у мережі. У цьому прикладі ID мережі - 192.168.0, і ID вузла - 28.

Маска підмережі. Велику мережу ділять на багато фізичних мереж (підмереж), з'єднаних маршрутизаторами. Адресний простір творять за допомогою механізму маскування. Маска підмережі закриває частину IP-адреси так, щоб TCP / IP міг відрізнити ID мережі від ID вузла. При з'єднанні вузлів TCP / IP, маска підмережі визначає, де знаходиться вузол одержувача: в локальній або віддаленій мережі. Для зв'язку в локальній мережі комп'ютери повинні мати однакову маску підмережі.

Шлюз за замовчуванням – це проміжний пристрій в локальній мережі, на якому зберігаються мережні ідентифікатори інших мереж підприємства або Інтернету. TCP / IP посилає пакети у віддалену мережу через шлюз за замовчуванням (якщо жоден інший маршрут не налаштований), який потім пересилає пакети іншим шлюзам, поки пакет не досягне шлюзу, пов'язаного з зазначеним адресатом.

Створено багато утиліт, які дозволяють швидко виконати діагностику IP-сполучення. Наприклад, користувачі Windows XP для діагностики мережного підключення можуть скористатися спеціальною програмою – майстром діагностики мережі. Шлях до цієї програми такий: (*Пуск > Все программы > Стандартные > Служебные > Сведения о системе > меню Сервис > Диагностика сети*). Однак більшість операцій легко може бути виконано з використанням команд самої операційної системи. Короткий опис можливостей найуживаніших утиліт подано нижче:

Ipconfig

Для відображення параметрів IP-протоколу використовуються утиліти *ipconfig* (Windows NT/2000/XP) і *winipcfg* (Windows 9x). Ця утиліта виводить на екран основні параметри налаштувань стеку протоколів TCP/IP: значення адреси, маски, шлюзу.

Ping

Команда використовується для перевірки протоколу TCP/IP і досяжності віддаленого комп'ютера. Вона виводить на екран час, за який пакети даних досягають заданого у її параметрах комп'ютера. За допомогою цієї утиліти і спеціально визначеної IP-адреси 127.0.0.1 можна здійснити перевірку проходження сигналу “самого на себе” навіть без наявності мережного під'єднання.

За замовчуванням команда посилає пакет довжиною 32 байти (розмір пакета може бути збільшений до 65 кбайт). Наступними за розміром тестового пакета відображаються час відгуку віддаленої системи та значення часу життя пакета (TTL). Параметр TTL фактично є відображенням числа маршрутизаторів, через які проходить пакет. Кожен маршрутизатор зменшує значення TTL на одиницю і при досягненні нульового значення пакет знищується. Такий механізм запроваджено для уникнення випадків зациклювання пакетів.

Tracert

Зв'язки між хостами у мережі забезпечують канали з різною продуктивністю, а часом взагалі можуть перериватися. Для показу шляху проходження сигналу до заданого хоста або встановлення причин поганого проходження використовують утиліту *tracert*. Час відгуку подається у мілісекундах.

Nslookup

Утиліта використовується для отримання інформації від DNS-сервера. За замовчуванням (після запуску без вказання параметрів) відбувається під'єднання до вказаного у налаштуваннях протоколу сервера DNS. Якщо у запиті вказувати необхідні імена, то можна отримати інформацію про дані DNS за цим іменем, знайти поштовий сервер, домен обслуговування, уточнити дані реєстрації та ін.

Хід роботи.

Завдання 1. Перевірити працездатність стека протоколів TCP / IP.

1. Запустіть віртуальну машину і завантажте ОС **Windows**.
2. Запустіть консоль (**Пуск / Програми / Стандартні / Командний рядок**).
3. У командному рядку введіть **cmd**.
4. У командному рядку новоствореного вікна введіть **ipconfig /all | More**.

5. При нормальній роботі комп'ютера отримаєте лістинг з інформацією про параметри мережного під'єднання.
6. Використовуючи наведену у лістингу інформацію, створіть у своїй папці текстовий документ і відобразіть такі дані:

- Ім'я комп'ютера;
- Основний DNS-суфікс;
- Опис DNS-суфікса для підключення;
- Фізичну адресу;
- DHCP (включений/виключений);
- Автоконфігурація (включена/виключена);
- IP-адреса автоконфігурації;
- Маска підмережі;
- Шлюз за замовчуванням.

8. Від'єднайте комп'ютер від мережі. Знову введіть команду *ipconfig /all*. Порівняйте лістинги, отримані тут і у пункті 2.

Завдання 2. Налаштуйте стек протоколів TCP / IP для використання статичної IP-адреси.

1. Відкрийте вікно **Мережні підключення** (*Пуск / Панель керування / Мережні підключення*).
2. Відкрийте **властивості підключення по локальній мережі**. Для цього можна скористатися контекстним меню.
3. У діалоговому вікні на вкладці **Загальні** відкрийте властивості **Протокол Інтернету TCP / IP**.
4. Клацніть перемикач *Використовувати Наступне IP-адресу* і введіть у відповідні поля дані: **IP_адреса; Маска підмережі; Основний шлюз; бажаний DNS**.
5. Застосуйте параметри кнопкою **ОК**.

6. Закрийте вікно властивостей підключення кнопкою **ОК** (якщо буде потрібно, то погодитися на перезавантаження комп'ютера).
7. Перевірте працездатність стека протоколів **TCP / IP**.

Завдання 3. Налаштуйте TCP / IP для автоматичного отримання IP-адреси.

1. Відкрийте вікно **Мережні підключення**.
2. Відкрийте властивості **Підключення по локальній мережі**.
3. Відкрийте властивості **Протокол Інтернету TCP / IP**.
4. Встановіть перемикач *Отримати IP-адресу автоматично*.
5. Закрийте діалогове вікно **Властивості: Протокол Інтернету TCP / IP** кнопкою **ОК**.
6. Застосуйте параметри кнопкою **ОК**.
7. Перевірте налаштування стека протоколів **TCP / IP**.
8. Отримайте іншу адресу для свого комп'ютера. Для цього:
 - запустіть консоль (командний рядок);
 - введіть команду для скидання призначених адрес - ;
 - введіть команду для отримання нової адреси ;
9. Перевірте працездатність стека протоколів **TCP / IP**.

Завдання 4. Перевірка правильності установки протоколу TCP/IP та функціонування каналу зв'язку.

1. Переконайтеся в працездатності стека **TCP / IP**, відправивши луна-запити на IP-адреси. Для цього скористайтеся командою **ping**, відправивши луна-запити на локальну адресу комп'ютера (*Loopback*) **ping 127.0.0.1** (на екрані повинні з'явитися повідомлення про отриману відповідь від вузла 127.0.0.1).
2. Перевірте логічний зв'язок між локальним комп'ютером та іншим комп'ютером мережі. Для цього відправте луна-запит за іншою IP-адресою, наприклад **172.21.5.1**.
3. Перевірте функціонування сервера імен Internet, виконавши команду

ping www.im.ua сервера.ua. Якщо система зможе розпізнати IP-адресу цього хоста, то це означатиме про правильне функціонування DNS-сервера.

4. Перевірте шлях проходження сигналу від вашого комп'ютера до віддаленого хоста за допомогою команди Tracert.
5. Проаналізуйте лістинги, отримані при виконанні попередніх пунктів завдання.

Завдання 5. Отримання інформації про налаштування DNS-сервера.

1. Виконайте команду *nslookup*.
2. Наберіть *server lnu.edu.ua* і натисніть *Enter*. Цією командою ми вказуємо адресу DNS-сервера, звідки хочемо отримати дані.
3. Наберіть *set type=all* і натисніть *Enter*.
4. Наберіть *im.ua* домену.ua і виконайте команду. Отримаємо дані про домен.
5. Проаналізуйте лістинги, отримані при виконанні попередніх пунктів завдання і подайте їх до звіту про виконану роботу.

Завдання 6. Оформити звіт про виконання лабораторної роботи у якому подати:

- f) тему, мету та завдання лабораторної роботи;
- g) прізвище, ініціали та назву групи студента, що виконав роботу;
- h) опис налаштувань локальної мережі між хостовою ОС та віртуальною машиною. Коротко описати функції сервісу NAT;
- i) короткий опис налаштувань локальної мережі у різних операційних системах;
- j) висновки.

Звіт оформити у вигляді файлу з іменем: "Прізвище"lan6.doc.

ЛАБОРАТОРНА РОБОТА №7

Тема: Адресування в IP – мережах.

Мета роботи: ознайомитись з адресацією в IP - мережах, навчитися розраховувати адреси мереж, підмереж, визначати необхідну кількість підмереж, визначати маску і адреси пристроїв для підмережі.

Короткі теоретичні відомості.

У мережах TCP / IP використовується три типи адрес: локальні адреси, звані MAC-адресами (апаратні), IP-адреси і символічні доменні адреси.

MAC-адресу призначають мережевим адаптерам і мережевим інтерфейсам маршрутизаторів виробниками обладнання і є універсальними (за функціональною ідентичністю), крім того, кожна адреса унікальна. Для всіх існуючих технологій локальних мереж MAC-адреса має формат 6 байт, наприклад,



IP-адреса є складовою частиною заголовка данограми, формується міжмережним протоколом IP (Internet Protocol), складається з чотирьох байт (для версії 4), призначається адміністратором під час конфігурації комп'ютерів і маршрутизаторів. IP-адреса складається з двох частин:



Номер мережі може бути обраний адміністратором довільно, або призначений за рекомендацією спеціального підрозділу Internet - Internic (Internet Network Information Center).

Номер вузла в протоколі IP призначається незалежно від локальної адреси вузла. Маршрутизатор входить відразу в кілька мереж, кожен порт маршрутизатора має свою власну IP-адресу. Кінцевий вузол також може входити в кілька IP-мереж. В цьому випадку комп'ютер повинен мати кілька IP-адрес по числу мережевих зв'язків.

Отже, IP-адреса характеризує не окремий комп'ютер або маршрутизатор, а одне мережеве з'єднання.

Символьні доменні імена будуються за ієрархічним принципом, розділяються крапкою. Будуються в наступному порядку: спочатку просте ім'я кінцевого вузла, потім ім'я групи вузлів (ім'я організації), потім ім'я більшої групи (піддомена) і так до імені домена найвищого рівня, наприклад, країна.

Класи IP-адрес

Всього існує 5 класів IP-адрес. Адреса складається з номера мережі і номера вузла в мережі. Яка частина адреси відноситься до номера мережі, а яка до номера вузла визначається першими значеннями бітів адреси. Значення цих бітів є також ознаками того, до якого класу належить та чи інша IP-адреса.

Відповідно до класів структура IP-адреси буде наступна:

Клас А

Номер біта	0	8	16	24	31
Адреса	0.....	
Мережева частина					

Клас В

Номер біта	0	8	16	24	31
Адреса	10.....	
Мережева частина					

Клас С

Номер біта	0	8	16	24	31
Адреса	110....	
Мережева частина					

Клас D

Номер біта	0	8	16	24	31
Адреса	1110...	

Клас Е

Номер біта	0	8	16	24	31
Адреса	1111...	

Якщо адреса починається з нуля, то мережу відноситься до класу А, номер мережі займає 1 байт, 3 байта інтерпретується як номер вузла в мережі.

Кількість вузлів в мережі 2^{24} .

Якщо перші два біта 10, то це клас В, по 2 байта на номер мережі і номер вузла, кількість вузлів 2^{16} .

Якщо адреса починається 110, то це мережа класу С, під номер мережі відводиться 24 біти, під номер вузла - 8.

Якщо адреса починається 1110, то це клас D - позначає особливі групові адреси.

Клас Е - адреси зарезервовані для майбутніх застосувань.

Особливі IP-адреси

У протоколі IP існує кілька домовленостей про особливу інтерпретацію IP-адрес:

- 1) якщо вся IP-адреса складається тільки з двійкових нулів, то вона позначає адресу того вузла, який згенерував цей пакет (використовується рідко);
- 2) якщо в полі номера мережі стоять лише нулі, то за замовчуванням вважається, що вузол призначення належить тій же самій мережі, що і вузол, який відправив пакет;
- 3) якщо всі виконавчі розряди IP-адреси рівні 1, то пакет повинен розсилатися всім вузлам, що знаходяться в тій же мережі, що й джерело цього пакета. Така розсилка називається обмеженим широкомовним повідомленням, відповідно *limited broadcast*.
- 4) якщо в номері вузла призначення стоять тільки одиниці, то пакет, що має таку адресу, розсилається всім вузлам мережі з заданим номером мережі. Така розсилка називається широкомовним повідомленням.

Особлива IP-адреса (якщо перший октет починається з 127) використовується для тестування програм і взаємодії процесів в межах однієї машини. Коли програма посилає дані за IP-адресою, наприклад, 127.0.0.1, то утворюється своєрідна петля. Дані не передаються по мережі, а повертаються модулем верхнього рівня, як тільки що прийняті. У літературі цю адресу називається *loop back*.

У протоколі IP немає поняття широкомовного адресування в тому сенсі, в якому воно використовується в протоколах канального рівня локальних мереж, коли дані повинні бути доставлені абсолютно до усіх вузлів. Як обмежена широкомовна IP-адреса, так і широкомовна IP-адреса мають межі поширення в Inter-мережі. Вони обмежені або мережею, до якої належить вузол-джерело пакета, або мережею, номер якої зазначено в адресі призначення. Тому поділ мережі за допомогою маршрутизатора на частини локалізує широкомовний шторм межами однієї зі складових загальної мережі. Основне призначення Multicast-адрес - поширення інформації за схемою один до багатьох. Комп'ютер, який хоче передавати одну інформацію багатьом абонентам за допомогою спеціального протоколу IGMP (Internet Group Management Protocol), повідомляє про створення нової мультикастної групи мережі з певною адресою. Маршрутизатори, що підтримують мультикаст, поширюють інформацію про створення нової групи в мережах, підключених до портів цього маршрутизатора.

Групова адресація призначена для економного поширення Internet аудіо- або відеопрограм, призначених відразу великій аудиторії слухачів або глядачів.

Використання масок в IP-адресації

Маска - це число, яке використовується в парі з IP-адресою і двійковий запис маски містить одиниці в тих розрядах, які повинні в IP-адресі інтерпретуватися як номер мережі. Для стандартних класів мереж маски мають таке значення:

Клас	Маска в двійковому вигляді	Маска в десятковому вигляді
A	11111111.11111111.11111111.00000000	255.255.255.0
B	11111111.11111111.00000000.00000000	255.255.0.0
C	11111111.00000000.00000000.00000000	255.0.0.0

Поле «номер мережі» в адресі називається мережевий префікс. Наприклад, 187.37.0.0/16, 16 - Мережевий префікс.

Приклади формування адрес

Завдання №1.

Організації виділено блок адрес 220.215.14.0/24. Розбити блок на 4 підмережі, найбільша з яких налічує 50 вузлів. Врахувати можливе зростання на 10%.

На першому етапі необхідну кількість підмереж заокругляємо у більшу сторону до найближчого степеня числа 2. Оскільки у даному прикладі число необхідних підмереж дорівнює 4, округляти не потрібно. Визначимо кількість бітів, потрібних для організації 4 підмереж. Для цього представимо 4 як степінь двійки: $4 = 2^2$. Степінь – і є кількість бітів, відведених для номера підмережі. Оскільки мережевий префікс блоку дорівнює 24, то розширений мережевий префікс дорівнюватиме $24 + 2 = 26$.

Мережний префікс				Підмережа	Вузол
0	8	16		24 25	31
220.215.14.0/26 <-->	10010000	10010000	00001110	0 0	000000
Розширений мережевий префікс					

Решта $32 - 26 = 6$ біт використовуватимуться для номера вузла. Перевіримо, скільки вузлів може бути задано 6-ма бітами: $2^6 - 2 = 62$ вузли. Чи достатньо це, враховуючи 10% зростання? 10% від 50 вузлів -- це 5 вузлів, а 55 вузлів менше можливих 62-х. Отже, два біти для номера підмережі нас влаштовують.

Наступний етап — знаходження підмереж. Для цього двійкове представлення номера підмережі, починаючи від нульового, підставляється в біти, відведені для номера підмережі.

Основна мережа	11011100	11010111	00001110	00	000000	220.215.14.0/24
Підмережа 0(00)	11011100	11010111	00001110	00	000000	220.215.14.0/26
Підмережа 1(01)	11011100	11010111	00001110	01	000000	220.215.14.64/26
Підмережа 2(10)	11011100	11010111	00001110	10	000000	220.215.14.128/26
Підмережа 3(11)	11011100	11010111	00001110	11	000000	220.215.14.192/26
Розширений мережевий префікс						

Для перевірки правдивості наших обчислень, працює просте **правило: десяткові номери підмереж повинні бути кратними номеру першої підмережі**. На цьому правилі можна побудувати й інше, яке спрощує розрахунок підмереж: досить обчислити адресу першої підмережі, а адреси наступних визначаються множенням першої адреси на відповідний номер підмережі. У прикладі ми легко можемо встановити адресу третьої підмережі, просто помноживши

$$64 * 3 = 192.$$

Як згадувалося, крім адреси підмережі, де всі біти вузлової частини рівні нулю, є ще одна службова адреса – ширококомовна. Особливістю ширококомовної адреси є те, що всі біти вузлової частини рівні одиниці. Розрахуємо ширококомовні адреси наших підмереж:

	підмережа
ШМА підмережі 0 (00)	11011100.11011100.00001110.00 111111 220.215.14.63/26
ШМА підмережі 0 (01)	11011100.11011100.00001110.01 111111 220.215.14.127/26
ШМА підмережі 0 (10)	11011100.11011100.00001110.10 111111 220.215.14.191/26
ШМА підмережі 0 (11)	11011100.11011100.00001110.11 111111 220.215.14.255/26
Розширений мережний префікс Вузлова частина = всі 1	

Легко помітити, що ширококомовною адресою є найбільша адреса підмережі. Тепер, отримавши адреси підмереж та його ширококомовні адреси, ми можемо побудувати таблицю використовуваних адрес:

№ підмережі	Найменша адреса підмережі	Найбільша адреса підмережі
0	220.215.14.1 - 220.215.14.62	
1	220.215.14.65 - 220.215.14.126	
2	220.215.14.129 - 220.215.14.190	
3	220.215.14.193 - 220.215.14.254	

Це і є розбивка, що задовольняє умові.

Завдання №2.

У першому прикладі підмережі були однакового розміру -- по 6 розрядів. Часто зручніше мати підмережі різного розміру. Припустимо, одна підмережа потрібна для задання адрес двох маршрутизаторів, пов'язаних за схемою "точка-точка". У цьому випадку використовується лише дві адреси.

Розглянемо тепер випадок, коли компанії виділено блок адрес 144.144.0.0/16. Потрібно розбити адресний простір на три частини, виділити адреси для двох пар маршрутизаторів і залишити певний резерв.

Розділимо мережу 144.144.0.0/16 на чотири рівні частини, виділивши два біти для номера підмережі:

Октет	W	X	Y	Z	
Підмережа 0(00)	10010000	10010000	00	000000	00000000 144.144.0.0/18
Підмережа 1(01)	10010000	10010000	01	000000	00000000 144.144.64.0/18
Підмережа 2(10)	10010000	10010000	10	000000	00000000 144.144.128.0/18
Підмережа 3(11)	10010000	10010000	11	000000	00000000 144.144.192.0/18

У середині третьої підмережі виділимо дві підмережі розміром у чотири адреси:

		Підмережа № 3				№ вузла
Підмережа 0(0)	10010000	10010000	11	000000	000000	00 144.144.192.0/30
Підмережа 1(1)	10010000	10010000	11	000000	000001	00 144.144.192.4/30
				Номер підмережі		

Отримані дві мережі використовуватимемо для адресації інтерфейсів маршрутизаторів. Адресний простір, що залишився, буде резервом, із якого виділятимемо адресні блоки за потребою. З решти адрес можна, наприклад, утворити 62 мережі розмірності класу C та ще декілька, трохи менших розмірів.

Завдання №3.

Дано адресу класу C: 200.35.1.0/24. В кожній підмережі передбачити адресний простір для 20 пристроїв. Визначити розширений мережевий префікс, максимальну кількість пристроїв, максимальне число підмереж, розписати адреси пристроїв в підмережі №6 і широкомовну адресу.

Рішення.

Знаходимо кількість біт для ідентифікації 20 пристроїв: $2^5 = 32$, значить, 5 біт, розширений мережевий префікс дорівнює $32 - 5 = 27$. Підмереж 8 ($2^{(27-24)} = 8$), вузлів - 30. В масці в двійковому поданні буде перших 27 одиниць:

255.	255.	255.	224	11111111.	11111111.	11111111.	11100000
------	------	------	-----	-----------	-----------	-----------	----------

#0	200.	35	1.	0/27	11001000.	00010011.	00000001.	00000000
#1	200.	35.	1.	32 /27	11001000.	00010011.	00000001.	00100000
#2	200.	35.	1.	64 /27	11001000.	00010011.	00000001.	01000000
#3	200.	35.	1.	96 /27	11001000.	00010011.	00000001.	01100000
⋮								⋮
#6	200.	35.	1.	192 /27	11001000.	00010011.	00000001.	11000000
#7	200.	35.	1.	224 /27	11001000.	00010011.	00000001.	11100000

Пристрої для шостої підмережі:

#6.1	200.	35.	1.	193 /27	10000100.	00101101.	00000001.	11000001
#6.2	200.	35.	1.	194 /27	10000100.	00101101.	00000001.	01100010
⋮								⋮
#6.30	200.	35.	1.	222 /27	10000100.	00101101.	00000001.	11011110
#6.III	200.	35.	1.	223 /27	10000100.	00101101.	00000001.	11011111

Завдання для самостійної роботи

Відповідно до варіанту для заданої IP адреси підрахувати адреси мереж, підмереж, визначати необхідну кількість підмереж, визначати маску і адреси пристроїв для підмережі. Оформити звіт.

Варіанти завдань

№ Варіанту	Блоки адрес	№ Варіанту	Блоки адрес
1	225.24.0.0/24	11	117.32.0.0/8
2	165.12.0.0/16	12	190.27.0.0/24
3	151.18.0.0/16	13	156.16.0.0/8
4	130.40.0.0/16	14	147.32.0.0/8
5	129.35.0.0/16	15	120.15.4.0/16
6	200.32.0.0/24	16	230.16.0.0/24
7	145.32.0.0/16	17	120.25.0.0/16
8	132.32.0.0/16	18	230.12.0.0/24
9	144.32.0.0/16	19	225.16.0.0/24
10	212.32.0.0/24	20	153.18.0.0/16

ЛАБОРАТОРНА РОБОТА № 8

Монтаж і налаштування бездротової мережі Wi-Fi

Мета роботи: Навчитись монтувати і налаштовувати локальну бездротову мережу між комп'ютерами за допомогою точки доступу з вмонтованим DHCP сервером. Вивчити особливості налаштувань в ОС Windows та Linux на прикладах Windows 7, Windows XP та Fedora Linux4 з використанням статичної та динамічної IP-адресації.

Теоретичні відомості

У сучасному світі все більше застосування знаходять бездротові мережі Wi-Fi, що дозволяють давати клієнтам доступ до ресурсів мереж, наприклад до Internet, з ноутбука або персонального комп'ютера, використовуючи в якості середовища передачі даних радіоканал, що не вимагає наявності спеціальних дротових з'єднань клієнтів з мережею, забезпечуючи таким чином їх мобільність

Переваги Wi-Fi

- Відсутність проводів.

Передача даних у мережі здійснюється через радіоканал. Можлива установка в місцях, де прокладка провідної мережі з тих чи інших причин неможлива або недоцільна, наприклад на виставках, залах для нарад.

- Мобільність, як робочих місць, так і самого офісу.

Так як бездротова мережа не прив'язана до проводів, Ви можете вільно змінювати місце розташування Ваших комп'ютерів в зоні покриття точки доступу, не турбуючись про порушення зв'язку. Мережа легко монтується / демонтується, при переїзді в інше приміщення Ви можете навіть забрати свою мережу з собою.

Недоліки Wi-Fi

- Відносно висока вартість обладнання

- Невелика дальність дії - 50-100 метрів

- Велика небезпека несанкціонованого підключення до мережі сторонніх користувачів

У запропонованій лабораторній роботі ми освоїмо створення найпростішої мережі Wi-Fi на прикладі підключення ноутбуків до точки доступу Wi-Fi з використанням статичної та динамічної IP-адресації.

Схема мережі має наступний вигляд:



Монтаж мережі.

1. Візьміть у викладача Wi-Fi-адаптер. Підключіть адаптер до USB-порту ноутбука № 2. (Див. схему мережі).
2. Увімкніть ноутбуки. Після завантаження операційної системи на ноутбуках, на обох адаптерах повинні загорітися сигнальні лампочки, що свідчать про встановлення радіообміну між адаптерами і точкою доступу.
3. Мережа зібрана, тепер для організації передавання повідомлень її необхідно налаштувати.

1. Налаштування мережі зі статичною адресою комп'ютера клієнта.

Налаштування мережі полягає в установленні протоколів ноутбука клієнта, які необхідні для його роботи, а так само включення і налаштування DHCP-сервера, який знаходиться в точці доступу.

Пам'ятаєте, що протокол - це спеціальна програма, за допомогою якої комп'ютери мережі обмінюються між собою даними за спеціальними правилами. У нашій мережі робочим протоколом буде протокол TCP / IP. Щоб комп'ютери могли обмінюватися між собою даними цей протокол повинен бути встановлений на всіх комп'ютерах, які знаходяться в мережі.

На ноутбучі-сервері протокол TCP / IP вже встановлено, нам залишилося встановити і налаштувати цей протокол на ноутбучі клієнта (див. схему мережі). Пам'ятайте, що всі пункти настройки повинні виконуватися в тій послідовності, в якій вони вказані. Не порушуйте послідовність налаштування.

На ноутбучі № 2 виконайте наступні дії:

1. Клацніть правою клавішею миші на значку «Моє мережне оточення», виберіть у меню «Властивості». Відкриється список мережевих підключень (рис.1.).

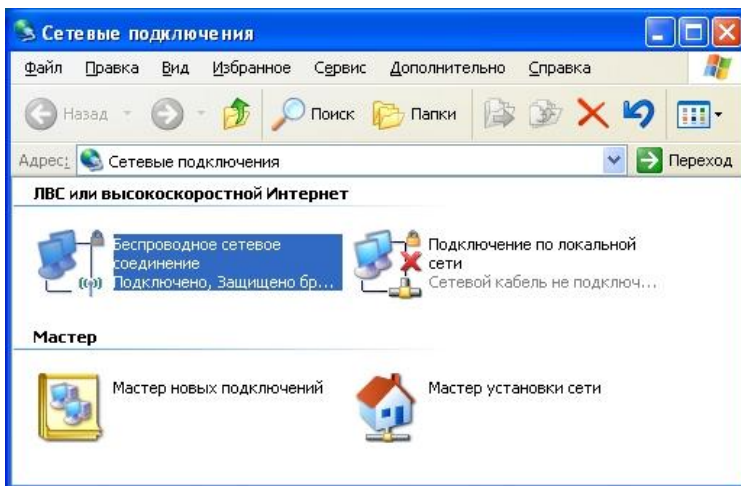


Рис.1.

2. Виберіть у списку «Бездротове мережеве з'єднання», клацніть по ньому правою клавішею миші та виберіть пункт «Властивості»). Відкриється вікно властивостей з'єднання (рис.2.).

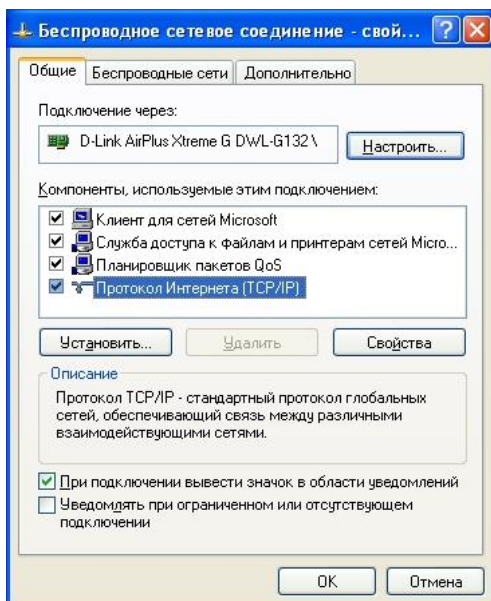


Рис.2.

3. У вікні виберіть «Протокол Інтернету (TCP / IP)», натисніть «Властивості». Відкриється вікно налаштувань протоколу (рис.3.). Активуйте ознаку «Використовувати наступний IP-адресу». Введіть в поля IP-адресу та Маску підмережі адреси установок, які зображені на рис.3.

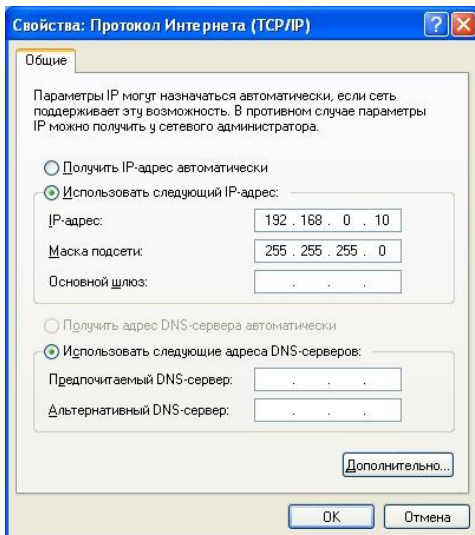


Рис.3.

Тут

192.168.0.10 - це IP-адреса комп'ютера в мережі.

255.255.255.0 - маска підмережі. Це спеціальний параметр, який разом з адресою однозначно визначає мережу, в якій знаходиться комп'ютер.

4. Після введення налаштувань, натисніть «ОК». Вікно «Властивості: Протокол Інтернету (TCP / IP)» закриється. У вікні «Бездротове мережеве з'єднання» (рис.2.) Натисніть "ОК".

Ми налаштували ноутбук клієнта для роботи з бездротовою мережею. Для ноутбука прописана статична IP-адреса, це означає що ми присвоїли ноутбуку виділену, постійну IP-адресу та інші настройки, які можна змінювати і призначати тільки вручну. Статична IP-адреса нам необхідна для того, щоб підключитися до точки доступу Wi-Fi і щоб інші комп'ютери в мережі могли з ним зв'язуватися.

Для того щоб почала функціонувати мережа Wi-Fi необхідно налаштувати точку доступу.


Встановлення точки доступу Wi-Fi і DHCP-сервера.

1. Завантажте оглядач Internet Explorer. Введіть в його адресному рядку адресу: <http://192.168.0.50/>. Це IP-адреса точки доступу Wi-Fi. За цією

адресою розташована система її конфігурації. Вхід в систему конфігурації захищений логіном і паролем і на екрані з'явиться вікно для введення цих даних.

Введіть Користувач - admin, Пароль - 12345678 і натисніть кнопку «ОК».

Відкриється головна сторінка системи конфігурації точки доступу Wi-Fi.


2. Клацніть по малюнку . Відкриється сторінка розширених налаштувань точки доступу.

3. Клацніть по малюнку . Відкриється сторінка для зміни налаштувань DHCP-сервера.

Встановіть наступні параметри DHCP, або змініть існуючі, якщо вони не збігаються із зазначеними:

1. Function Enable / Disable - Enabled
2. IP Assigned From - 192.168.0.51
3. The Range Of Pool (1-255) - 200
4. SubMask - 255.255.255.0
5. lease Time (60 - 31536000 sec) - 10000000
6. Status - ON



Клацніть по малюнку  щоб зберегти зроблені настройки. Точка доступу Wi-Fi піде на перезавантаження, яке займає приблизно півхвилини.

Запам'ятайте. Виконані вище настройки забезпечують виконання таких функцій:

Function Enable / Disable - Включає (Enabled) або відключає (Disabled) DHCP-сервер.

IP Assigned From - задає початковий IP-адресу, з якого починається діапазон IP-адрес, що виділяються динамічно користувачам (користувачі, які підключаються тимчасово).

The Range of Pool - задає кінець діапазону IP-адрес, кінцеве значення останньої цифри IP-адреси.

Таким чином у нашому прикладі ми поставили діапазон IP-адрес від 192.168.0.51 до 192.168.0.200 включно.

SubMask - маска підмережі. Це спеціальний параметр, який разом з адресою однозначно визначає мережу, в якій знаходиться комп'ютер.

Lease Time - час «життя» виділених користувачеві мережевих налаштувань.

При динамічній адресації налаштування існують певний час, після чого скидаються та програмне забезпечення користувача запитує нові налаштування. Тут задається час існування виділених користувачеві налаштувань (в секундах).

Status - спеціальний параметр, він ставиться в значення ON, якщо в мережі використовується спільно динамічна і статична адресації. В нашому випадку цей параметр встановлений в ON, оскільки на ноутбучі клієнта прописана статична, постійна адреса.

Перевірка роботи бездротової мережі.

Після того, як мережа налаштована, потрібно перевірити її роботу і переконатися, що комп'ютери можуть обмінюватися даними між собою.

Необхідно знати, що в мережі можуть існувати різні служби і сервіси, кожен з яких виконує свої завдання. У мережі, яку ми налаштували працюють дві служби: локальний WEB-сервер, призначений для розміщення HTML-сторінок в мережі, і Мережа Microsoft, за допомогою якої проводиться обмін файлами і спільна робота з клієнтами.

Спочатку перевіримо роботу WEB-сервера. WEB-сервер встановлений на ноутбучі сервер. Для того, щоб перевірити роботу WEB-сервера, запустіть на ноутбучі № 2 (комп'ютер Клієнт) оглядач Інтернету Internet Explorer і в його адресному рядку введіть <http://192.168.0.3/wifi/>

Якщо сторінка завантажиться, дійте відповідно до вказівок, написаними на цій сторінці.

Якщо сторінка не завантажилася, значить мережа налаштована неправильно.

Тоді зробіть наступне:

1. Перевірте ще раз налаштування протоколу TCP / IP ноутбука клієнта і переконайтеся що вони введені правильно.
2. Якщо помилка не зникає, покличете викладача.

Запам'ятайте. Статична IP-адресація має такі недоліки:

1. Для того, щоб дізнатися всі налаштування мережі, необхідно звернутися до адміністратора мережі, який повинен індивідуально виділити для кожного клієнта свою унікальну IP-адресу. Це незручно як для клієнта, так і для адміністратора.

2. При підключенні до будь-якої іншої бездротової мережі, настройки комп'ютера клієнта доводиться знову змінювати під нову мережу, дізнаючись їх у адміністратора.
 3. Якщо випадково ваші налаштування співпадуть з настройками іншого клієнта, ви не зможете підключитися до мережі.
- Усіх зазначених недоліків позбавлена динамічна IP-адресація.

2. Налаштування мережі з динамічною адресою комп'ютера клієнта.

Динамічна IP-адресація здійснюється за допомогою DHCP-сервера, який знаходиться в точці доступу. Розберемося що це таке.

DHCP-сервер використовує DHCP протокол (англ. Dynamic Host Configuration Protocol - протокол динамічної конфігурації вузла) - це мережевий протокол, що дозволяє комп'ютерам автоматично отримувати IP-адреси та інші параметри, необхідні для роботи в мережі TCP / IP. Для цього комп'ютер, який підключається до мережі, звертається до сервера, DHCP, який на час проведення сеансу роботи з мережею йому видає динамічну IP-адресу. Це дозволяє уникнути ручного настроювання комп'ютерів мережі, зменшує кількість помилок і дозволяє клієнтам швидко підключатися до мережі не витрачаючи час на налаштування протоколів зв'язку вручну.

Налаштування ноутбука на динамічну IP-адресацію.

1. Поверніться до початку лабораторної роботи, де ви здійснювали настройку мережі ноутбука № 2. (Розділ «Налаштування мережі»).
2. Повторіть кроки 1-3, тільки на 3-му кроці, де ви вводили статичну IP-адресу активуйте ознаку «Отримати IP-адресу автоматично». Це опція і включає динамічну IP-адресацію.
3. Натисніть «ОК», вікно «Властивості: Протокол Інтернету (TCP / IP)» закриється. У вікні «Бездротове мережеве з'єднання» (рис.2.) Натисніть "ОК".

Динамічна IP-адресація на ноутбуці налаштована!

Перевірка динамічної IP-адресації

1. Використовуючи процедуру «Безпечного вилучення пристрою» відключіть Wi-Fi адаптер від ноутбука клієнта. Вона виконується так само, як і при відключенні флеш-карт.
2. Видаліть адаптер з роз'єму USB.
3. Зачекайте кілька секунд і знову вставте адаптер в роз'єм USB. Відбудеться автоматичне підключення ноутбука клієнта до бездротової мережі Wi-Fi і ноутбуку будуть динамічно присвоєні IP-адреси та інші мережеві настройки.

Для того, щоб переконатися в тому, що мережеві настройки були динамічно присвоєні, зробіть наступне:

1. Відкрийте «Пуск / Стандартні / Командний рядок». З'явиться рядок для введення команд операційної системи.
2. Введіть у рядку команду:

ipconfig

і натисніть Enter

Ця команда відображає на екран налаштування протоколу TCP / IP вашого комп'ютера.

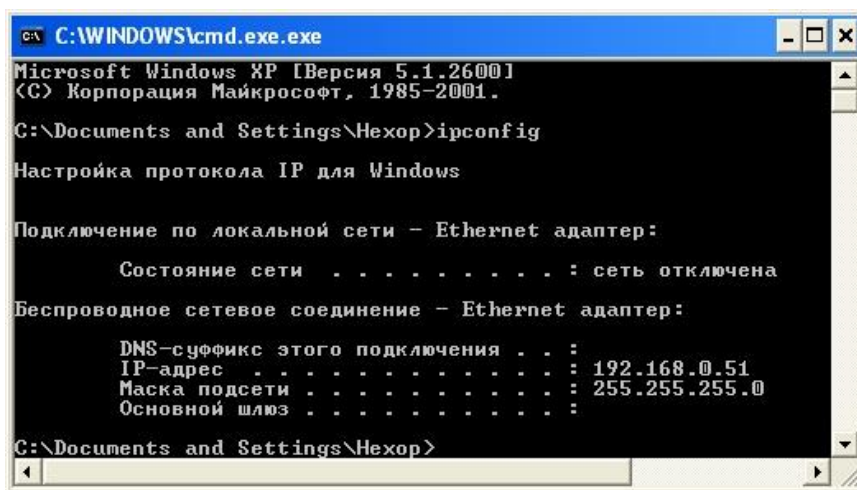


Рис .4.

Якщо зазначена командою IP-адреса комп'ютера знаходиться в діапазоні 192.168.0.51 - 192.168.0.200, значить динамічна IP-адресація працює нормально.

У випадку, якщо зазначена командою IP-адреса комп'ютера НЕ знаходиться в діапазоні 192.168.0.51 - 192.168.0.200), необхідно:

1. Виконати настройку мережі заново, встановивши статичну IP-адресу, потім, підключившись до точки доступу Wi-Fi перевірте, включений – чи ні DHCP-сервер і чи правильно - виставлені його параметри.
2. Якщо помилка не зникла - зверніться до викладача.

Перевірка роботи бездротової мережі.

Спочатку перевіримо роботу WEB-сервера. WEB-сервер встановлений на ноутбучі сервері. Для того, щоб перевірити роботу WEB-сервера, запустіть на ноутбучі клієнті оглядач Інтернету Internet Explorer і в його адресному рядку введіть <http://192.168.0.3/wifi/>

Якщо сторінка завантажиться, дійте відповідно до вказівок, написаними на цій сторінці

Якщо сторінка не завантажилася, значить мережа налаштована неправильно.

Тоді зробіть наступне:

3. Перевірте ще раз налаштування протоколу TCP / IP ноутбука № 2 і переконайтеся що вони введені правильно. IP-адреса повинна призначатися динамічно, увімкніть динамічну адресацію, якщо це не було зроблено.

4. Якщо помилка не зникає, покличете викладача.

Оформіть звіт з лабораторної роботи №8.

ЛАБОРАТОРНА РОБОТА №9

«Вивчення засобів моніторингу та аналізу мережевого трафіку.

Сніффер Wireshark »

МЕТА РОБОТИ:

1. Знати принципи аналізу мережевого трафіку.
2. Навчитися використовувати аналізатор (сніффер Wireshark).
3. Навчитися аналізувати мережевий трафік на прикладі протоколів ARP, IP і ICMP.

Теоретичні відомості.

Sniffer (від англ. To sniff - нюхати) – це мережевий аналізатор трафіку, програма або програмно-апаратний пристрій, призначений для перехоплення і подальшого аналізу, або тільки аналізу мережевого трафіку, призначеного для інших вузлів.

Перехоплення трафіку може здійснюватися:

- звичайним «прослуховуванням» мережевого інтерфейсу (метод ефективний при використанні в сегменті концентраторів (хабів) замість комутаторів (свічів), в іншому випадку метод малоефективний, оскільки на сніффер потрапляють лише окремі фрейми);
- підключенням сніффера в розрив каналу;
- відгалуженням (програмним або апаратним) трафіку і спрямуванням його копії на сніффер;
- через аналіз побічних електромагнітних випромінювань і відновлення таким чином трафіку, що прослуховується;
- через атаку на каналному (2 -й) або мережевому (3 -й) рівні, що приводить до перенаправлення трафіку жертви або всього трафіку

сегменту на сніффер з подальшим поверненням трафіку до належної адреси.

На початку 1990-х широко застосовувався хакерами для захоплення призначених для користувача логінів і паролів. Широке поширення хабів дозволяло захоплювати трафік без великих зусиль у великих сегментах мережі.

Сніффери застосовуються як в благих, так і в деструктивних цілях. Аналіз трафіку, що пройшов через сніффер, дозволяє:

- Відстежувати мережеву активність додатків.
- Налаштовувати протоколи мережевих додатків.
- Локалізувати несправність або помилку конфігурації.
- Виявити паразитний, вірусний і за кільцований трафік, наявність якого збільшує навантаження мережевого обладнання та каналів зв'язку.
- Виявити в мережі шкідливе і несанкціоноване ПО, наприклад, мережеві сканери, флудери, троянські програми, клієнти пірінгових мереж та інші.
- Перехопити будь-який незашифрований (а деколи і зашифрований) призначений для користувача трафік з метою впізнавання паролів та іншої інформації.

Поступово із інструментів, призначених тільки для діагностики, сніффери поступово перетворилися в засоби для досліджень і навчання. Наприклад, вони постійно використовуються для вивчення динаміки і взаємодій в мережах. Зокрема, вони дозволяють легко і наочно вивчати тонкощі мережевих протоколів. Спостерігаючи за даними, які посилає протокол, ви можете глибше зрозуміти його функціонування на практиці, а заодно побачити, коли деяка конкретна реалізація працює не у відповідності зі специфікацією.

На сьогоднішній момент існує досить велика кількість хороших реалізацій сніфферів. Деякі з них:

- Tcpdump ([Http: // www. Tcpdump. Org /](http://www.tcpdump.org/)) - консольний варіант сніффера. Портовано майже під усі найпоширеніші ОС;
 - Wireshark ([Http: // www. Wireshark. Org /](http://www.Wireshark.Org/)) до недавнього моменту був відомий під назвою Ethreal;
 - WinDump <http://www.winpcap.org/windump> ;
 - IP Sniffer
- та ін.

СНІФФЕР WIRESHARK

Програма Wireshark є однією з найбільш зручних реалізацій сніфферів. Перенесена на велику кількість платформ. Поширюється абсолютно безкоштовно. Зручно використовувати даний сніффер для вивчення і аналізу мережних протоколів.

Для початку розглянемо базовий принцип роботи сніффера на прикладі Wireshark.

Базовий принцип роботи сніфферів

На рис. 1 зображена схематично структура мережевої підсистеми ОС. Вся базова інфраструктура реалізована у вигляді драйверів і працює в режимі ядра. Призначені для користувача процеси і реалізації прикладних протоколів, зокрема інтерфейс сніффер працюють в режимі користувача.

На малюнку відображені два призначених для користувача процеси («мережевий процес 1» і «мережевий процес 2»).

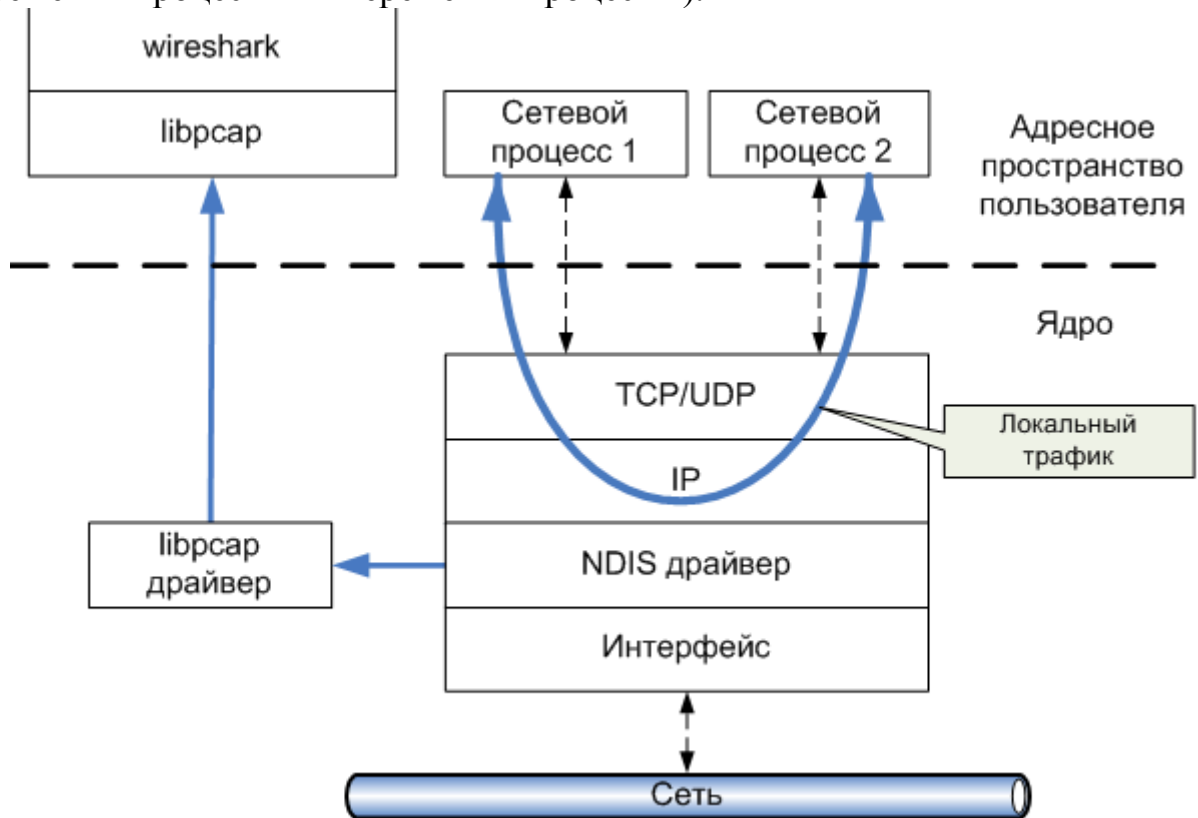


Рис.1. Принцип захоплення сніффером мережевого трафіку.

Основними компонентами сніффера є: драйвер для захоплення пакетів (libpcap драйвер), інтерфейсна бібліотека (libpcap) і інтерфейс користувача (Wireshark). Бібліотека libpcap (реалізація під ОС Windows носить назву WinPcap - <http://www.winpcap.org>) - Універсальна мережева бібліотека, самостійно реалізує велику кількість мережних протоколів і працює

безпосередньо з NDIS (Network Driver Interface Specification) драйверами мережеских пристроїв. За підсумками цієї бібліотеки реалізована велика кількість мережеских програм, зокрема сніффер Wireshark.

Сніффер використовує бібліотеку в режимі «захоплення» пакетів, тобто може отримувати копію всіх даних, що проходять через драйвер мережеского інтерфейсу. Зміни в самі дані не вносяться!

Основний нюанс використання сніффера полягає в тому, що він не дозволяє проводити аналіз локального трафіку, тому що він не проходить через драйвер мережеского пристрою (див. рис 1.). Тобто, якщо ви захочете проаналізувати сніффером трафік між 2-ми мережескими процесами на локальній машині (наприклад, ftp-сервер і ftp-клієнт), то на вас чекає розчарування. Однак, наприклад при використанні віртуальних машин, сніффер буде працювати без проблем, тому що віртуальні машини емулюють реальне середовище і мережескі адаптери, тому трафік йде через драйвера як і в нормальній ситуації при взаємодії з іншими фізичними мережескими машинами.

Також до недоліків більшості сніфферів варто віднести і той факт, що, дозволяючи аналізувати трафік, що проходить через мережеский інтерфейс, вони не можуть вказати, який саме додаток генерує або отримує його. Це пояснюється тим, що інформація про це зберігається на мережескому (наприклад, IP) рівні мережеского стека, а більшість сніфферів використовує власну реалізацію стека протоколів (наприклад, бібліотеку WinPcap), яка (як уже було показано) працює безпосередньо з драйверами пристроїв.

Також, сніффери вносять додаткове навантаження на процесор, тому що можуть обробляти досить об'ємний мережеский трафік, особливо для високошвидкісних з'єднань (Fast Ethernet, Gigabit Ethernet та ін.).

Використання програми Wireshark

Даний сніффер дозволяє в режимі реального часу захоплювати пакети з мережі, і аналізувати їх структуру. Також можна аналізувати структуру пакетів з файлу, що містить трафік, отриманий, наприклад, програмою «tcpdump» (unix / linux).

На рис. 2. зображено основне вікно програми Wireshark. У стандартному режимі вікно сніффера ділиться на 3 фрейми (панелі): список захоплених пакетів, «аналізатор» протоколів і вихідні дані пакетів. Розмір кожного фрейма можна змінювати на свій розсуд.

Розглянемо ці панелі докладніше.

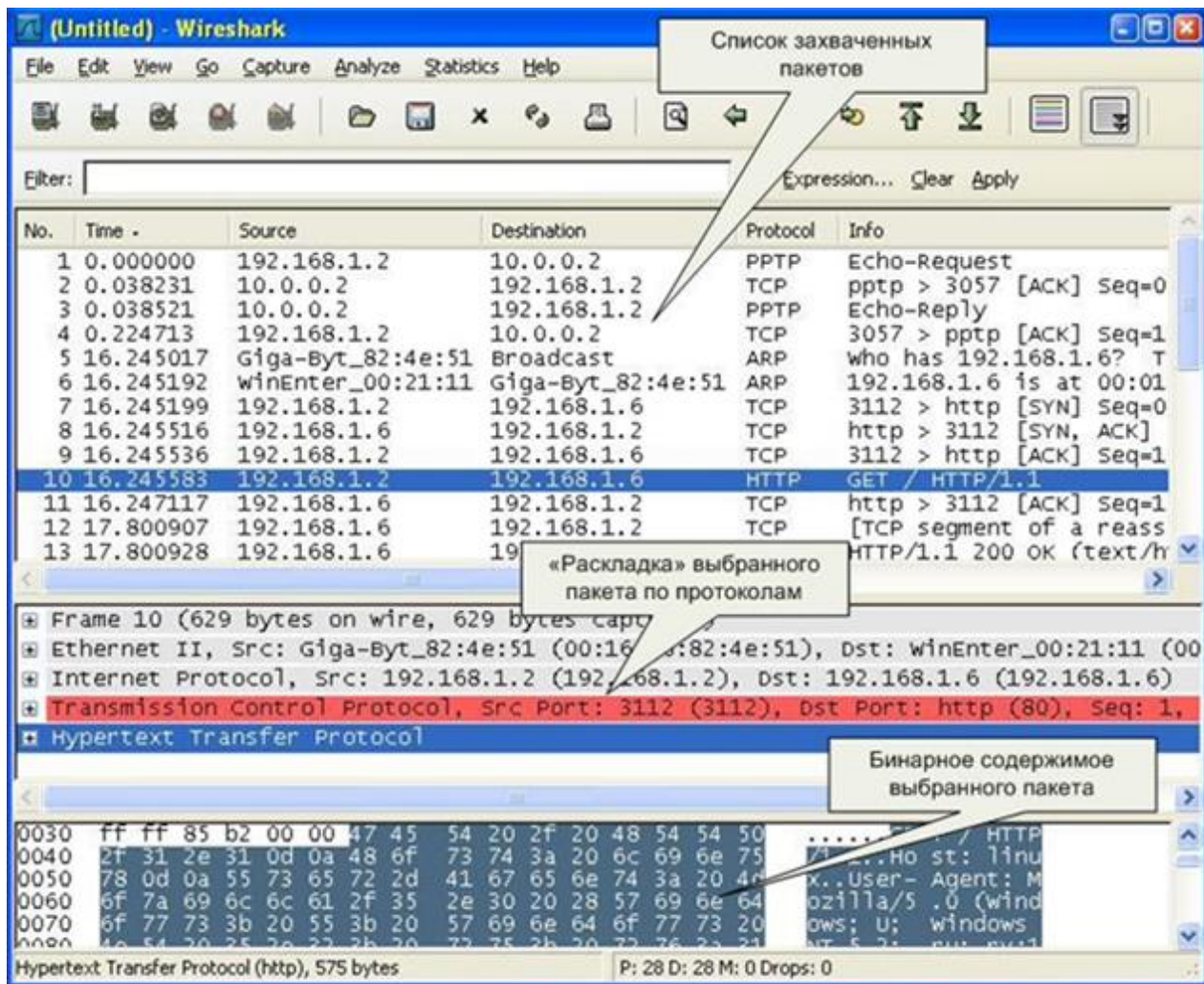


Рис.2. Основне вікно сніффера Wireshark.

Верхня панель містить список пакетів, захоплених з мережі. Список можна відсортувати за будь-якого поля (в прямому або зворотному порядку) - для цього натиснути на заголовок відповідного поля.

Кожен рядок містить наступні поля (за замовчуванням):

- порядковий номер пакета (No.);
- час надходження пакета (Time);
- джерело пакета (Source);
- пункт призначення (Destination);
- протокол (Protocol);
- інформаційне поле (Info).

Список відображуваних полів налаштовується в Edit/Perferencis/Columns. Для того, щоб зміни набули ефекту необхідно перезапустити програму, попередньо натиснувши кнопку Save.

При натисканні правої кнопки миші на тому чи іншому пакеті, з'явиться контекстне меню. Натисканням на середню кнопку миші можна позначати групу цікавлять нас пакетів.

Середня панель містить т.зв. «Дерево протоколів» для обраного у верхньому вікні пакета. У цій панелі в ієрархічному вигляді для обраного в верхньому вікні захопленого пакета відображається вкладеність протоколів відповідно до моделі взаємодії відкритих систем OSI. При натисканні на праву

кнопку миші викликається контекстне меню. При «розкритті» кожного з протоколу натисканням на значок «+» ліворуч, виводяться поля даних відповідних протоколів.

Нижня панель містить шістнадцяткове подання обраного пакета. При виборі того чи іншого поля в середній панелі автоматично буде підсвічуватися відповідна ділянка 16-го представлення.

ЗАХОПЛЕННЯ ПАКЕТІВ

Для початку захоплення пакетів необхідно задати параметри захоплення. Зокрема, вказати мережевий інтерфейс, з якого і буде здійснюватися захоплення пакетів. Ця дія є через меню як «Capture ->Options» або комбінації клавіш CTRL + K (див. рис. 3). Інтерфейс, що задається в полі «**Interface:**» можна вибрати з відповідного поля. У прикладі на рис. 3. показано, що доступні 3 інтерфейси: фізичний мережевий адаптер («Marvel ...»), та інтерфейси для віртуальних каналів, зокрема, встановленого VPN-з'єднання («WAN (PPP / SLIP) ...»). У більшості випадків підходить вибір інтерфейсу мережевого адаптера.

В якості додаткових параметрів захоплення можна вказати наступні:

- «**Capture Filter**» - фільтр захоплення (будемо розглядати далі). При натисканні на відповідну кнопку можна застосувати той чи інший фільтр відбору (з раніше збережених). Якщо таких немає, його можна вказати явно в рядку редагування.
- «**Update list of packets in real time**» - оновлення списку захоплених пакетів в режимі реального часу.
- «**Stop Capture**»- набір параметрів, що дозволяють задати те чи інше значення, при досягненні якого процес захоплення пакетів припиниться.
- «**Name Resolution**»- набір параметрів розпізнавання імен дозволяє визначити які із способів розпізнавання імен повинні використовуватися.
-

Для початку моніторингу мережевої активності потрібно натиснути «Start». Після вибору інтерфейсу, який нас цікавить, в подальшому можна починати і зупиняти захоплення пакетів через відповідні команди в меню «Capture».

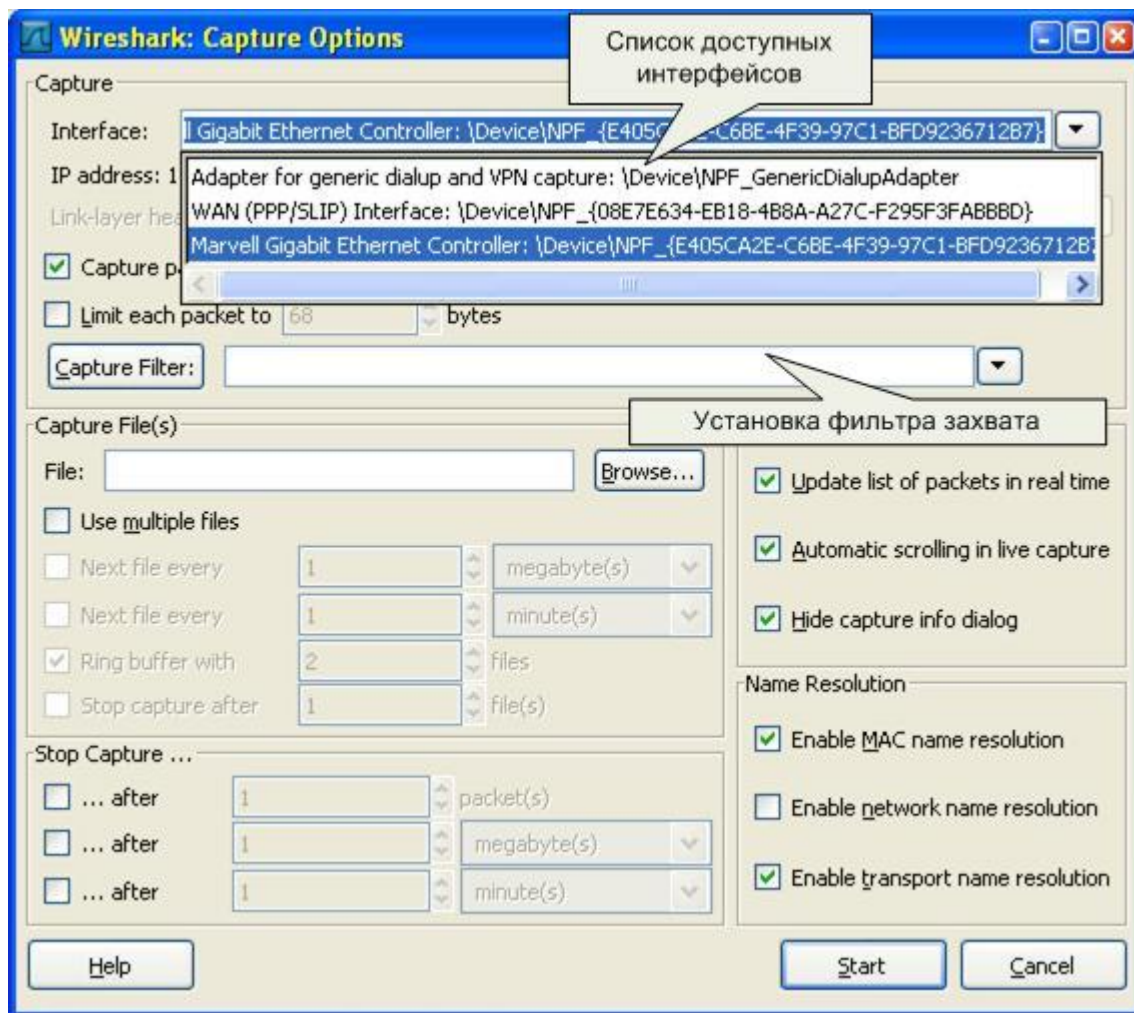


Рис. 3. Вибір інтерфейсу і параметрів захоплення пакетів

ФІЛЬТРАЦІЯ ПАКЕТІВ

Якщо запустити сніффер без додаткових налаштувань, він буде «захоплювати» всі пакети, що проходять через мережеві інтерфейси (див. Рис. 1). Взагалі, для загального ознайомлення з процесами, що відбуваються в мережі, дуже корисно поспостерігати активність мережевих протоколів в реальних умовах роботи системи в мережі. Поспостерігати все розмаїття протоколів, запитів, відповідей та ін. подій.

При цілеспрямованому використанні сніффера дуже часто необхідно вибірково відображати або захоплювати пакети за деякими заданими критеріями. Для цього слугують фільтри відображення і захоплення, відповідно.

Типи фільтрації трафіку

Існує два варіанти фільтрації пакетів: на етапі захоплення і на етапі відображення користувачеві. У першому випадку ефективність роботи

сніффера і споживані ним системні ресурси значно нижчі, ніж у другому випадку. Це пояснюється тим, що при досить інтенсивному трафіку мережі та і тривалому часу захоплення всі пакети повинні бити захоплені і збережені або в пам'ять, або на накопичувач на комп'ютері. Найпростіші підрахунки можу показати, що навіть для 100-мегабітної мережі системних ресурсів вистачить на нетривалий час. Фільтрація захоплення вже на момент отримання пакета набагато ефективніше, однак в такому випадку вона повинна бути реалізована на рівні самих драйверів захоплення. Даний факт, природно, ускладнює реалізацію сніффера. Wireshark підтримує обидва варіанти фільтрації. Розглянемо

Фільтри відображення

Фільтри відображення являють собою досить потужний засіб відображення трафіку. Фільтри задаються в рядку, що розташований вгорі основного екрану («Filter:»). Найпростіший фільтр відображення дозволяє відібрати пакети з того чи іншого протоколу. Для цього в рядку потрібно вказати назву протоколу (наприклад HTTP) і натиснути кнопку «Apply». Після цього у верхньому вікні залишаться пакети, що належать до цього протоколу. Кнопкою «Reset» дію фільтра скасовують.

Для роботи з фільтрами можна викликати вікно «Analyze/ Display Filters». Можна зберігати створені вирази під певними іменами для подальшого використання і т.д.

За допомогою логічних операцій (синтаксис мови Сі) можна складати логічні вирази. Логічна істина - 1, брехня - 0.

Список підтримуваних логічних операцій:

eq	==	рівність
ne	!=	не дорівнює
gt	>	більше ніж
Lt	<	менше ніж
ge	>=	більше рівне
Le	<=	менше рівне

Наприклад: tcp. Port == 80 (див. Рис. 4).

Filter: tcp.port==80					
Expression... Clear Apply					
No.	Time	Source	Destination	Protocol	Info
5	8.689788	192.168.1.2	192.168.1.1	TCP	4810 > http [SYN] Seq=0
6	8.691675	192.168.1.1	192.168.1.2	TCP	http > 4810 [SYN, ACK] S
7	8.691717	192.168.1.2	192.168.1.1	TCP	4810 > http [ACK] Seq=1
8	8.691877	192.168.1.2	192.168.1.1	HTTP	GET /html/js/alphaindex.
9	8.699198	192.168.1.1	192.168.1.2	TCP	http > 4810 [ACK] Seq=1
10	8.699230	192.168.1.1	192.168.1.2	TCP	[TCP segment of a reasse
11	8.699246	192.168.1.1	192.168.1.2	HTTP	HTTP/1.1 404 Not Found (
12	8.699266	192.168.1.2	192.168.1.1	TCP	4810 > http [ACK] Seq=46
13	8.699556	192.168.1.2	192.168.1.1	TCP	4810 > http [FIN, ACK] S
14	8.701682	192.168.1.1	192.168.1.2	TCP	http > 4810 [ACK] Seq=45

Рис. 4. Приклад задання простого фільтра відображення

Майстер побудови фільтрів відображення доступний через кнопку «Expression ...» (див. Рис. 5).

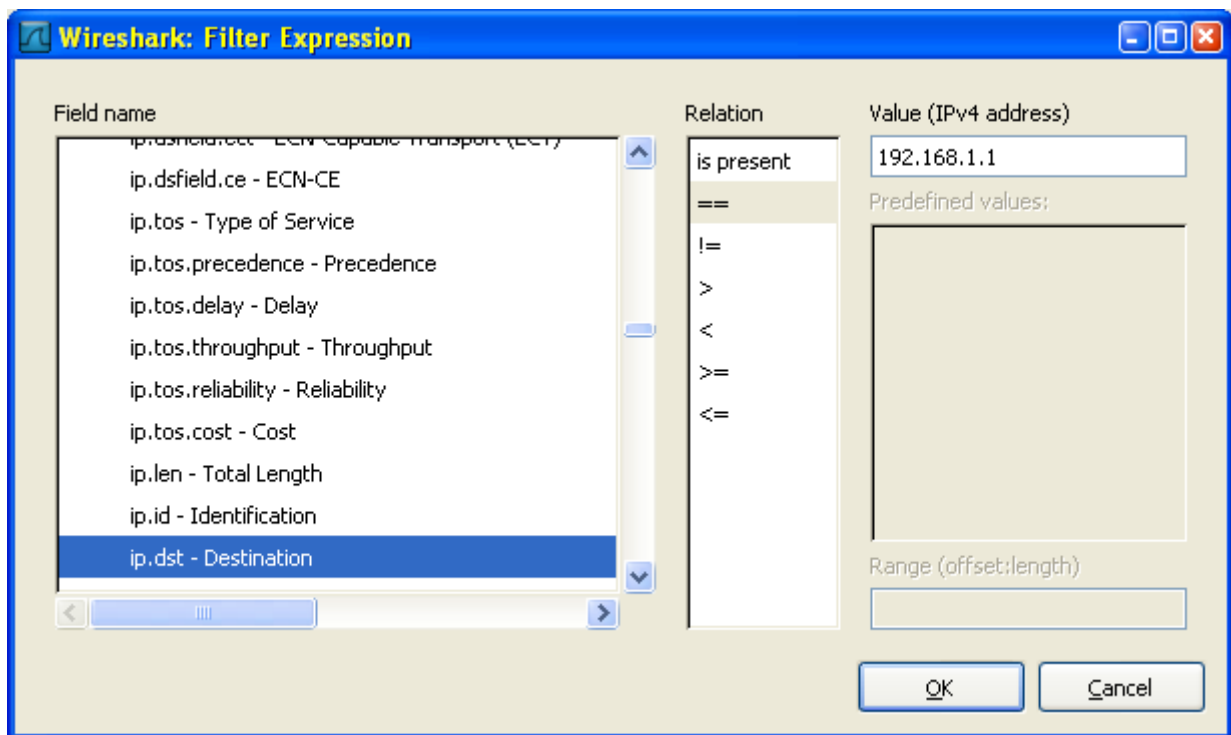


Рис. 5. Побудова фільтрів відображення

фільтри захоплення

За допомогою даних фільтрів можна захоплювати з мережі тільки ті пакети, які підходять під критерій відбору. Якщо не задано ніякого фільтра (за замовчуванням), то будуть захоплюватися всі пакети. В іншому випадку тільки пакети, для яких вказаний вираз буде істинним. Синтаксис фільтрів захоплення дещо відрізняється від синтаксису фільтрів відображення. Вираз складається з одного або більше примітивів розділених пробільними

символами. На рис. 6 наведено приклад фільтра для захоплення пакетів, адресованих на 80-й порт (http) вузла з ip - адресою 10.197.0.11.

Існує три різних типи примітивів: **type**, **dir**, **proto**.

Специфікатор **type** визначає тип параметра. Можливі параметри: **host**; **net**; **port**.

наприклад:

- host linux
- net 192.168.128
- port 80

Якщо не вказано жодного типу, передбачається що це параметр **host**.

Специфікатор **dir** визначає напрямок передачі. Можливі напрямки: **src**; **dst**; **src or dst**; **src and dst**.

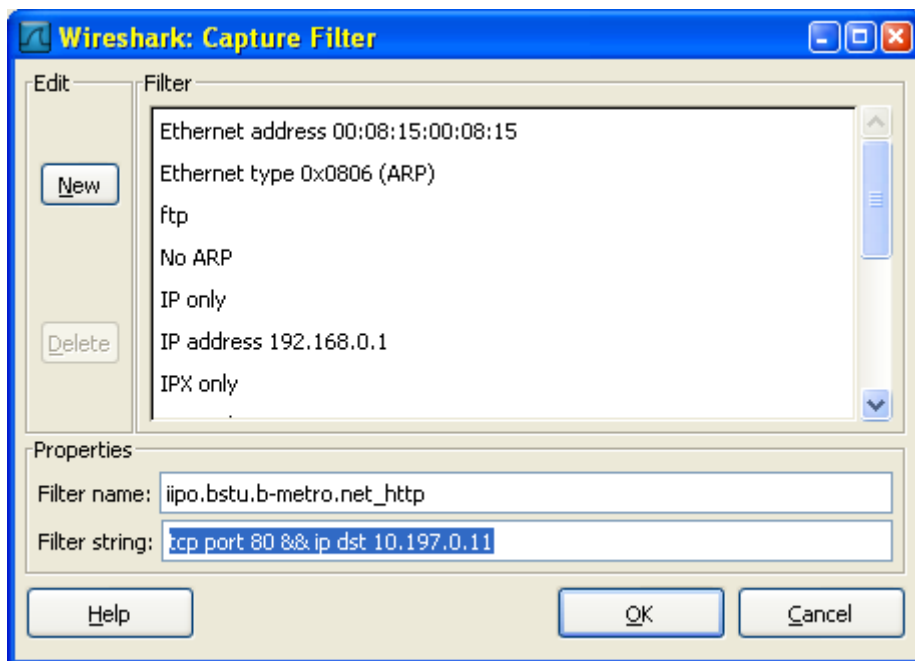


Рис. 6. Приклад фільтра захоплення

Якщо не визначено напрямок, то передбачається напрям «**src or dst**». Для протоколів типу point - to - point використовуються специфікатор inbound і outbound.

Специфікатор **proto** визначає тип протоколу, якому належить пакет.

Можливі протоколи: **ether**; **fddi**; **tr**; **ip / ipv 6**; **arp / rarp**; **decent**; **tcp**; **udp**.

наприклад:

- ether src linux
- arp net 192.168.128
- tcp port 80

Якщо протокол не визначений, то будуть захоплюватися пакети всіх протоколів. Тобто: «src linux» означає «(ip or arp or rarp) src linux»;

«net ctam » означає « (ip or arp or rarp) net ctam »; « port 53 » означає« (tcp or udp) port 53 ».

Також існує кілька спеціальних специфікаторів, які не потрапляють в описані вище випадки:

- *gateway*;
- *broadcast*;
- *less*;
- *greater*;
- *арифметичні вираження*.

Складні фільтри захоплення будуються з використанням логічних виразів.

Список операцій:

not	!	заперечення
and	&&	конкатенація (логічне І)
or		альтернатива (логічне АБО)

Приклади фільтрів захоплення

Нижче розглянуті деякі приклади побудови фільтрів захоплення.

- Захоплення всіх пакетів на мережевому інтерфейсі хоста 192.168.1.2: `host 192.168.1.2`
- Захоплення трафіку між хостом host 1 і хостами host 2 АБО host 3: `host host1 and (host2 or host3)`
- Захоплення всіх IP-пакетів між хостом host 1 і кожним хостом за винятком hostX: `ip host host 1 and not hostX`
- Захоплення пакетів ні згенерованих ні адресованих локальними хостами: `ip and not net localnet`
- Захоплення IP-пакетів розміром більше ніж 576 байт, що проходять через шлюз snup: `gateway snup and ip [2: 2]> 576`
- Захоплення всіх ICMP пакетів, за винятком пакетів ping: `icmp [0]! = 8 and icmp [0]! = 0`

Статистична обробка мережевого трафіку

сніффер Wireshark дозволяє виконувати різну статистичну обробку отриманих даних. Всі доступні операції знаходяться в меню «Statistics».

Загальна статистика - кількість отриманих / переданих пакетів, середня швидкість передачі і т.д. доступні через пункт «Statistics -> Summary».

Отримати інформацію із статистики оброблених протоколів в отриманих пакетах можна через пункт «Statistics -> Protocol Hierarchy».

Статистику за типом ір - пакетів, їх розміром і портом призначення можна отримати обравши підпункти меню «IP-address ...», «Packet length» і «Port type» відповідно.

Однією з найбільш цікавих можливостей є генерація діаграми взаємодії між вузлами, яка доступна пунктом меню «Flow Graph ...». В результаті можна спостерігати в досить наочній формі процес взаємодії на рівні протоколів. Наприклад, на рис. 7 приведена діаграма взаємодії при отриманні вузлом win 2003 статичної web -сторінки з сервера http://linux.

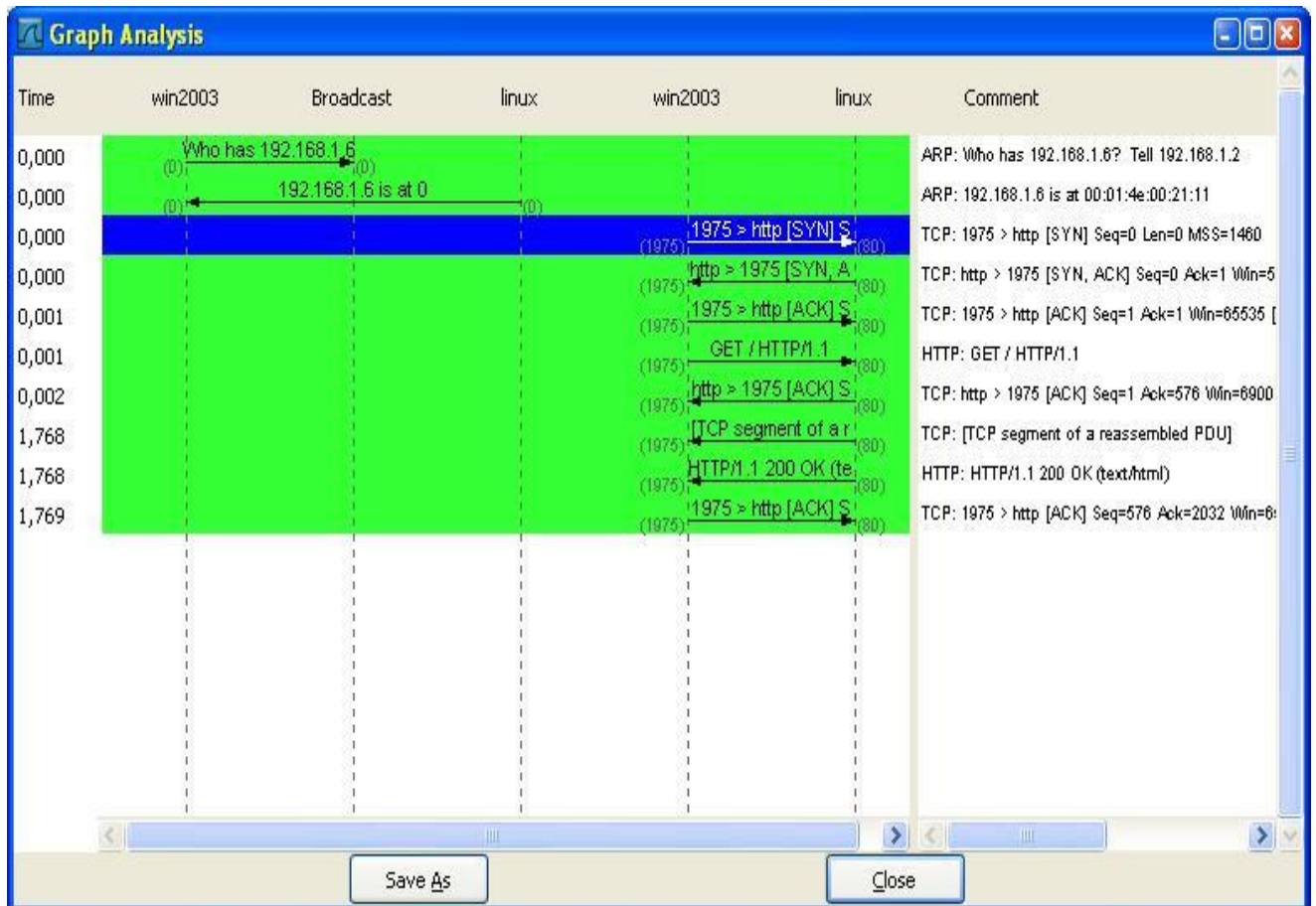


Рис. 7. Діаграма взаємодії

ВИСНОВОК

Програми-сніфери - це незамінний інструмент для вивчення того, що відбувається в мережі. Якщо знати, що насправді посиляється або приймається «по дротах», то важкі, на перший погляд, помилки вдається легко знайти і виправити. Сніффер є також важливий інструмент для досліджень динаміки мережі, а так само засіб навчання.

Приклад дослідження протоколів з використанням сніффера

Як приклад дослідження деякого протоколу з використанням сніффера розглянемо протокол ARP.

Протокол ARP

ARP (англ. Address Resolution Protocol - протокол дозволу адрес) - мережевий протокол, призначений для перетворення IP-адрес (адрес мережевого рівня) в MAC-адреси (адреси канального рівня) в мережах TCP / IP. Він визначений в **RFC 826**.

Даний протокол дуже поширений і надзвичайно важливий. Кожен вузол мережі має як мінімум дві адреси, фізичну адресу і логічну адресу. У мережі Ethernet для ідентифікації джерела і одержувача інформації використовуються обидві адреси. Інформація, що пересилається від одного комп'ютера іншому по мережі, містить в собі фізичну адресу відправника, IP-адресу відправника, фізичну адресу одержувача і IP-адресу одержувача. ARP -протокол забезпечує зв'язок між цими двома адресами. Існує чотири типи ARP-повідомлень: ARP-запит (ARP request), ARP-відповідь (ARP reply), RARP-запит (RARP - request) і RARP-відповідь (RARP - reply). Локальний хост за допомогою ARP-запиту запитує фізичну адресу хоста-одержувача. Відповідь (фізична адреса хоста-одержувача) приходить у вигляді ARP-Відповіді. Хост-одержувач, разом з відповіддю, шле також RARP-запит, адресований відправнику, для того, щоб перевірити його IP адресу. Після перевірки IP адреси відправника, починається передача пакетів даних.

Перед тим, як створити підключення до будь-якого пристрою в мережі, IP-протокол перевіряє свій ARP-кеш, щоб з'ясувати, чи не зареєстрована в ньому вже потрібна для підключення інформація про хості-одержувачі. Якщо такого запису в ARP-кеші немає, то виконується широкомовний ARP-запит. Цей запит для пристроїв в мережі має наступний сенс: «Хто-небудь знає фізичну адресу пристрою, що володіє такою IP-адресою?» Коли одержувач прийме цей пакет, то повинен буде відповісти: «Так, це моя IP-адреса. Моя фізична адреса наступна: ...» Після цього відправник оновить свій +ARP-кеш, і буде здатний передати інформацію одержувачу.

RARP (англ. Reverse Address Resolution Protocol - зворотний протокол перетворення адрес) - виконує зворотне відображення адрес, тобто перетворює апаратну адресу в IP-адресу.

Протокол застосовується під час завантаження вузла (наприклад, комп'ютера), коли він посилає групове повідомлення-запит зі своєю фізичною адресою. Сервер приймає це повідомлення і переглядає свої таблиці (або перенаправляє запит куди-небудь ще) в пошуках відповідної фізичної IP-адреси. Після виявлення знайдена адреса відсилається назад на вузол, що її запитував. Інші станції також можуть «чути» цей діалог і локально зберегти цю інформацію в своїх ARP-таблицях.

RARP дозволяє розділяти IP-адреси між хост-вузлами, що нечасто використовуються. Після використання якимось вузлом IP-адреси вона може бути вивільнена і виданий іншому вузлу. RARP є доповненням до ARP, і описаний в **RFC 903**.

Для перегляду ARP -кеша можна використовувати однойменну утиліту `arp` з параметром « - a ». Наприклад :

```
D: \> arp -a
```

```
Interface: 192.168.1.2 --- 0x10003
```

```
Internet Address Physical Address Type
```

```
192.168.1.1 00-15- e 9- b 6-67- 4 f    dynamic
```

```
192.168.1.6 00-01-4 e -00-21-11 dynamic
```

З даного результату команди `arp` видно, що в кеші на даний момент знаходиться 2 записи і видно відповідно `ip` - адреси машин і `MAC` - адреси їх мережевих адаптерів.

Записи в ARP -кеші можуть бути статичними і динамічними. Приклад, даний вище, описує динамічний запис кеша. Хост-відправник автоматично послав запит одержувачу, не повідомляючи при цьому користувача. Записи в ARP -кеш можна додавати вручну, створюючи статичні (static) записи кеша. Це можна зробити за допомогою команди:

```
arp -s < IP адреса> < MAC адреса>
```

Також можна видаляти з ARP -кеша. Це здійснюється шляхом наступного виклику :

```
arp -d < IP адреса>
```

Після того, як IP -адреса пройшла процедуру дозволу адреси, вона залишається в кеші протягом 2-х хвилин. Якщо протягом цих двох хвилин сталася повторна передача даних за цією адресою, то час зберігання запису в кеші продовжується ще на 2 хвилини. Ця процедура може повторюватися до тих пір, поки запис в кеші проіснує до 10 хвилин. Після цього запис буде видалено з кеша і буде відправлений повторний ARP -запит.

ARP спочатку був розроблений не тільки для IP протоколу, але в даний час в основному використовується для зіставлення IP- і MAC-адрес.

Подивимося ж на практиці як працює протокол ARP / RARP. Для цього скористаємося сніффером для захоплення мережевого трафіку.

Розглянемо приклад роботи протоколу ARP при зверненні до машини з адресою 192.168.1.5, виконавши запит з машини з адресою 192.168.1.2. Для успішного експерименту попередньо очистимо `arp` -кеш командою

```
arp -d 192.168.1.5
```

Для фільтрації ARP / RARP трафіку скористаємося фільтром захоплення. У нашому випадку це буде простий фільтр

```
arp or rarp
```

Далі запустимо захоплення трафіку командою « Start » і виконаємо звернення до заданої машини, наприклад, «пропінгувати» її :

```
D: \> ping 192.168.1.5
```

```
Pinging 192.168.1.5 with 32 bytes of data:
```

```
Reply from 192.168.1.5: bytes = 32 time <1ms TTL = 64
```

Reply from 192.168.1.5: bytes = 32 time <1ms TTL = 64
Reply from 192.168.1.5: bytes = 32 time <1ms TTL = 64
Reply from 192.168.1.5: bytes = 32 time <1ms TTL = 64
Ping statistics for 192.168.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Так як на момент початку роботи утиліти ping в arp - кеші не було інформації про MAC -адресу відповідного вузла, то спочатку система повинна виконати пошук цієї MAC - адреси, згенерувавши ARP - запит і відіславши його в мережу широкошовним пакетом. Після чого вона буде чекати відповіді від заданого вузла.

Подивимося, що ми отримаємо на практиці. Після зупинки сніффера ми повинні побачити результат схожий з тим, що відображений на рис. 8. У нашому випадку ми бачимо 2 захоплених пакети: ARP - запит і ARP - відповідь.

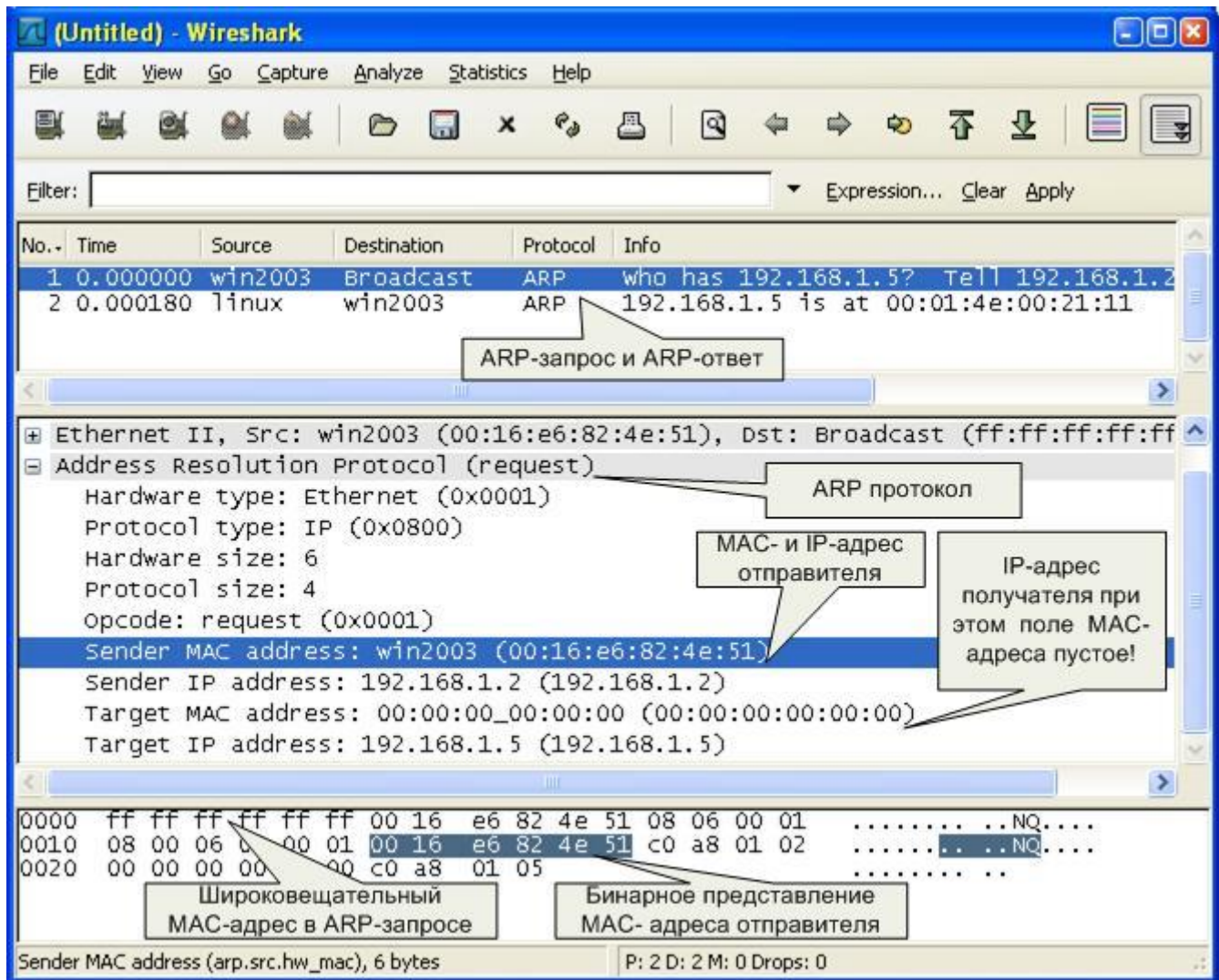


Рис. 8. Аналіз ARP-запиту

Проаналізуємо отримані пакети. Спочатку розглянемо ARP - запит (пакет №1). Виділивши пакет курсором, ми отримуємо його розкладку за протоколами (Ethernet + ARP) в середньому вікні. Wireshark дуже наочно «розкладає» заголовок протоколу по полях.

Ми можемо бачити, що в пакеті вказані MAC - і IP - адреси відправника («Sender MAC address» і «Sender IP address» відповідно). Це параметри машини, з якої виконується запит. В даному випадку запит направлений на отримання («Opcode: request» - запит) MAC -адреси машини, у якій IP -адреса («Protocol type: IP») 192.168.1.5 («Target IP address»). При цьому поле «Target MAC address» обнулене. Так як одержувача ARP - запиту на момент запиту ніхто не знає, Ethernet - пакет відправляється всім машинам в даному локальному сегменті, про що сигналізує MAC адреса Ethernet - пакету «ff:ff:ff:ff:ff:ff».

Примітка. Зверніть увагу, що пакет являє собою бінарну послідовність і сніффер виконує велику роботу по перетворенню полів з бінарного представлення в легкий для читання варіант.

Усі працюючі машини в мережі отримують пакет з ARP - запитом, аналізують його, а відповідь відсилає тільки та машина, чия IP - адреса відповідає IP - адресі в запиті. Таким чином, другий отриманий пакет є ARP - Відповіддю (див. Рис. 9). Це впливає з параметра поля « Opcode : reply ». Зверніть увагу, що даний пакет був відправлений саме тією машиною, чия MAC – адреса нас і цікавила («Sender IP address : 192.168.1.5»). При цьому поле « Sender MAC address » заповнене значенням «00 : 01 : 4 E : 00: 21: 11 ».

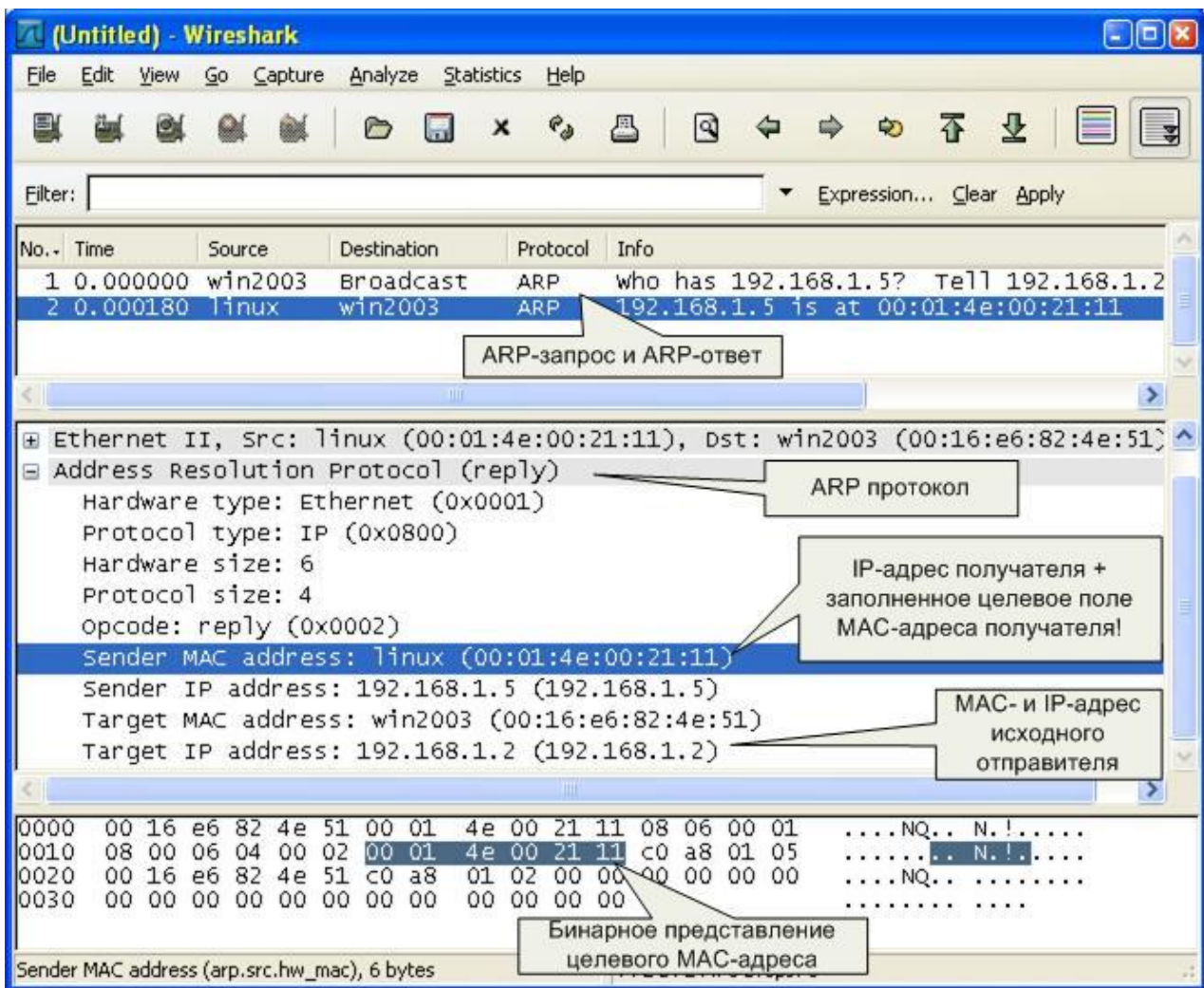


Рис. 9. Аналіз ARP-відповіді

Примітка. Зверніть увагу на поле « Info » в списку захоплених пакетів. Сніффер і тут спрощує аналіз мережевого трафіку, підказуючи призначення пакетів ☺

Тепер ми можемо повторно переглянути ARP - кеш і звірити дані в ньому з даними, які ми дізналися з аналізу пакетів ARP -запиту / відповіді :

D: \> arp -a

Interface: 192.168.1.2 --- 0x10003

Internet Address Physical Address Type

192.168.1.5 00-01-4e-00-21-11 dynamic

Варто також відзначити, що в реальних умовах в локальній мережі з великою кількістю машин arp / rarp трафік буває набагато більш інтенсивним.

ЗАВДАННЯ ДЛЯ ВИКОНАННЯ РОБОТИ

1. Вивчити інтерфейс програми Wireshark
2. Захопити 100 довільних пакетів. Визначити статистичні дані:
 - процентне співвідношення трафіку різних протоколів в мережі;
 - середню швидкість кадрів / сек;
 - середню швидкість байт / сек;
 - мінімальний, максимальний і середній розміри пакета;
 - ступінь використання смуги пропускання каналу (завантаження мережі).
3. Зафіксувати 20 IP - пакетів. Визначити статистичні дані:
 - процентне співвідношення трафіку різних протоколів стека tcp / ip в мережі;
 - середній, мінімальний, максимальний розміри пакета.
4. Виконати аналіз ARP-протоколу за прикладом з методичних вказівок.
5. На прикладі будь-якого IP - пакета вказати структури протоколів Ethernet і IP, відзначити поля заголовків і описати їх.
6. Проаналізувати і описати принцип роботи утиліти *ping*.
При цьому описати всі протоколи, використовувані утилітою. Описати всі поля протоколів. Скласти діаграму взаємодії машин при роботі утиліти *ping*.
7. Оформити звіт про виконання роботи.

КОНТРОЛЬНІ ПИТАННЯ

1. Які основні цілі моніторингу мережевого трафіку ?
2. Чим відрізняється моніторинг трафіку від фільтрації ?
3. Яке призначення класу програм-сніфферів ?
4. Які основні функції виконують сніффери ?
5. Навіщо використовуються фільтри відображення і фільтри захоплення сніффера Wireshark ? У чому їхня відмінність ?
6. Які базові функції статистичної обробки захоплених пакетів має сніффер Wireshark ?
7. Які завдання розрахований вирішувати протокол ARP ?

ЛАБОРАТОРНА РОБОТА № 10

Тема: «Створення мережевих додатків з використанням сокетів»

Теоретичні відомості:

Сокет - це один кінець двостороннього зв'язку між двома програмами, що працюють в мережі. Поєднуючи два сокети можна передавати дані між різними процесами (локальними і віддаленими). Реалізація сокетів здійснює інкапсуляцію протоколів мережевого і транспортного рівнів.

Існує два типи сокетів **потоківі і дейтаграмні**. *Потоковий сокет* - це сокет з встановленням з'єднання, що складається з потоку байтів, який може бути двонаправленим, тобто через кінцеву точку може передавати і отримувати дані. Потоковий сокет здійснює надійну передачу, підходить для передачі великих обсягів даних. Потоківі сокети використовують протокол TCP. Для цього типу сокетів шлях формується до початку передачі повідомлення. Сокет А запитує з'єднання з сокетом В, а сокет В або погоджується на встановлення з'єднання з сокетом А, або відкидає його.

Дейтаграмні сокети - сокети без встановлення з'єднання. Використовується протокол UDP.

Сокет складається з IP адреси машини і номера порту, що використовується додатком TCP. Оскільки IP адреси унікальні в Інтернеті, а номери портів унікальні на окремій машині, то номери сокетів унікальні в усьому Інтернеті. Ця характеристика дозволяє процесу спілкуватися через мережу з іншим процесом виключно на підставі номера порту.

Зазвичай додаток клієнт-сервер, що використовує сокети, складається з двох різних додатків: клієнта, який ініціює з'єднання з сервером, і сервера, що очікує запиту на з'єднання від клієнта. На стороні клієнта додаток повинен знати адресу і номер порту сервера. Відправляючи запит, клієнт намагається встановити з'єднання з сервером, якщо сервер запущений, сервер погоджується на з'єднання і створює новий сокет для встановлення взаємодії з клієнтом, який встановив з'єднання. Клієнт і сервер після цього можуть зчитувати і передавати повідомлення кожен зі свого сокета.

Завдання 1: Розглянемо приклад створення клієнт-серверного додатка. Клієнтська програма відправляє три числа і отримує обчислене серверним додатком значення суми чисел.

Створіть консольний додаток в середовищі *Visual Studio 2005*
File -> New -> Project->ConsoleApplication, назвіть *Klient*.

Додайте у розділ using.
`using System.Net;`
`using System.Net.Sockets;`

Введіть такий програмний код:

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.Net.Sockets;

namespace Klient
{
    class Program
    {
        static void Main(string[] args)
        {
            //создание сокета
            Socket s1 = new Socket(AddressFamily.InterNetwork,
                                    SocketType.Stream, ProtocolType.Tcp);
            IPAddress adr = Dns.Resolve("localhost").AddressList[0];
            // создание конечной точки с указанием параметров соединения
            IPEndPoint ipEnd = new IPEndPoint(adr, 8086);
            //соединение с сервером
            s1.Connect(ipEnd);
            Console.WriteLine("Соединение установлено.\n");
            Console.WriteLine("Введите три числа:\n");
            string str=null;
            int i=1;
            while (i <= 3)
            {
                Console.WriteLine("Введите {0} -е число:\n", i);
                str = Console.ReadLine();
                // буфер для отправляемых данных
                byte[] d = Encoding.ASCII.GetBytes(str);
                // отправка данных
                s1.Send(d);
                Console.WriteLine("{0} -е число отправлено!\n", i);
                i++;
            }
            Console.WriteLine("Все числа отправлены!\n");
            byte[] R = new byte[1024]; // буфер для полученных данных из сети
            //получение результата
            s1.Receive(R);
            Console.WriteLine("Вычисленный удаленно результат суммы равен {0}",
                               Encoding.ASCII.GetString(R));

            Console.ReadLine();
            s1.Close();
        }
    }
}

```

Створіть консольний додаток в середовищі Visual Studio 2005
File -> New -> Project->ConsoleApplication, назвіть Server1.
Введіть такий програмний код:

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.Net.Sockets;

namespace Server1
{
    class Program
    {
        static void Main(string[] args)
        {
            //создание сокета
            Socket Listener = new Socket(AddressFamily.InterNetwork,
                                         SocketType.Stream, ProtocolType.Tcp);
            IPAddress adr=Dns.Resolve("localhost").AddressList[0];
            //создание конечной точки с параметрами соединения
            IPEndPoint ipEnd = new IPEndPoint(adr, 8086);
            Listener.Bind(ipEnd);
            // прослушивание запросов о соединении
            Listener.Listen(10);
            Console.WriteLine("Ожидание соединения...");
            Socket s = Listener.Accept();
            // буфер для получаемых данных
            byte[] bufR = new byte[1024];
            int i=1;
            int sum=0;

            while (i <= 3)
            {
                //получение данных из сети
                s.Receive(bufR);
                // преобразование в строку из типа данных байт
                string data = Encoding.ASCII.GetString(bufR);
                Console.WriteLine("{0} -е число: {1}", i, data);
                sum += Convert.ToInt32(data);
                i++;
            }
            //буфер для отправляемых данных
            byte[] bufS = Encoding.ASCII.GetBytes(sum.ToString());
            //отправка результата
            s.Send(bufS);
            Console.WriteLine("Результат суммы равный {0}- отправлен", sum);
            Console.ReadLine();
            s.Close();
        }
    }
}

```

ЗАПУСТИТЬ СПОЧАТКУ СЕРВЕРНИЙ ДОДАТОК, А ПОТІМ КЛІЄНТСЬКИЙ ДОДАТОК.

У клієнтському додатку введіть числа і переконайтеся в правильності роботи мережевого додатка. Після тестування на локальному комп'ютері, скопіюйте мережевий додаток на різні комп'ютери, підключені до локальної мережі. У клієнтському додатку змініть "localhost" на ім'я комп'ютера, де знаходиться серверний додаток, який можна дізнатися за допомогою команди hostname. У серверному додатку аналогічно змініть "localhost" на ім'я комп'ютера - сервера.

Завдання 2: Реалізувати приклад створення клієнт -серверного додатка. Клієнтська частина представлена у вигляді Windows додатка, а серверна у вигляді консольного. Клієнтська програма відправляє кількість повідомлень, які буде відправлено на сервер, потім активуються елементи для введення повідомлень, В заголовку форми, відображаються результати відповіді серверного додатка про прийом і кількості прийнятих символів. Після закінчення відправки заданої кількості повідомлень, кнопка відправки стає неактивною.

Створіть Windows додаток до форми, що містить наступні елементи: .

Рисунок 1 – Конструктор форми

В режимі View Code введіть наступний програмний код

Додати:

```
using System.Net;
using System.Net.Sockets;

public partial class Form1 : Form
{
    int k, i = 0;
    Socket s;
    byte[] bytes = new byte[1024];
    public Form1()
    {
        InitializeComponent();

        textBox2.Visible = false;
        label2.Visible = false;
        button2.Visible = false;
        try
        {
            //устанавливаем удаленную конечную точку для сокета
            IPHostEntry ipHost = Dns.Resolve("127.0.0.1");
            IPAddress ipAdr = ipHost.AddressList[0];
            IPEndPoint ipEndPoint = new IPEndPoint(ipAdr, 11000);
            s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
            s.Connect(ipEndPoint);
            this.Text="Сокет соединен с " + s.RemoteEndPoint.ToString();
        }
        catch (Exception e)
        {
            MessageBox.Show("исключение", e.ToString());
        }
    }
}
```

Для першої кнопки, яка відправляє кількість повідомлень введіть такий код:

```
private void button1_Click(object sender, EventArgs e)
{
    k = Convert.ToInt16(textBox1.Text);
    byte[] msg1 = Encoding.ASCII.GetBytes(textBox1.Text);
    //отправляем данные через сокет
    s.Send(msg1);
    textBox1.Visible = false;
    textBox2.Visible = true;
    label2.Visible = true;
    button2.Visible = true;
    button1.Visible = false;
    label1.Visible = false;
}
```

Для другої кнопки, що відправляє повідомлення, введіть такий код:

```
private void button2_Click(object sender, EventArgs e)
{
    if (i < k)
    {
        string theMassege = textBox2.Text;
        byte[] msg = Encoding.ASCII.GetBytes(theMassege);
        //отправляем данные через сокет
        s.Send(msg);
        int bytesRec = s.Receive(bytes);
        this.Text = "сервер отвечает :" +
            Encoding.ASCII.GetString(bytes, 0, bytesRec).ToString();
        textBox2.Text = "";
        i++;
    }
    else button2.Enabled = false;
}
```

На наступному етапі створіть серверний консольний додаток, що містить наступний програмний код::

```

class Program
{
    static void Main(string[] args)
    {
        //устанавливаем для сокета локальную конечную точку
        IPHostEntry ipHost = Dns.Resolve("localhost");
        IPAddress ipAdr = ipHost.AddressList[0];
        IPEndPoint ipEndPoint = new IPEndPoint(ipAdr, 11000);
        //создаем сокет TCP/IP
        Socket sListener = new Socket(AddressFamily.InterNetwork,
            SocketType.Stream, ProtocolType.Tcp);
        try
        {
            sListener.Bind(ipEndPoint);
            sListener.Listen(10);
            //начинаем слушать соединения
            while (true)
            {
                Console.WriteLine("ожидание соединения с портом {0}", ipEndPoint);
                //программа приостанавливается, ожидая входящее соединение
                Socket handler = sListener.Accept();
                string data = null;
                //дождались клиента, который хочет соединиться

                while (true)
                {
                    byte[] bytes = new byte[1024]; // для получения текста сообщений
                    byte[] kol = new byte[10]; // для получения количества сообщений
                    int bRec = handler.Receive(kol);
                    int k = Convert.ToInt32(Encoding.ASCII.GetString(kol, 0, bRec));
                    Console.WriteLine("количество= {0}", k);

                    for (int i = 0; i < k; i++)
                    {
                        int bytesRec = handler.Receive(bytes);
                        data = Encoding.ASCII.GetString(bytes, 0, bytesRec);
                        Console.WriteLine("Полученные данные : {0}", data);
                        string theReply = "Спасибо за " + data.Length.ToString() + " символа";
                        byte[] msg = Encoding.ASCII.GetBytes(theReply);
                        handler.Send(msg);
                    }
                }

                handler.Shutdown(SocketShutdown.Both);
                handler.Close();
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.ToString());
        }
    }
}

```

Запустіть спочатку серверний додаток, потім клієнтський додаток.
Оформіть звіт про виконану роботу.

Клас Socket простору імен System.Net.Sockets

Властивість	Опис
AddressFamily	Надає сімейство адрес сокетів – значення з перелічення Socket.AddressFamily
Available	Повертає обсяг доступних для читання даних.
Blocking	Дає або встановлює значення чи знаходиться сокет в стадії блокування
Connected	Повертає значення, яке інформує чи з'єднаний сокет з віддаленим хостом
LocalEndPoint	Дає локальну кінцеву точку
ProtocolType	Дає тип протоколу сокета
RemoteEndPoint	Дає віддалену кінцеву точку
SocketType	Дає тип сокета

Методи класу Socket

Методи	Опис
Accept()	Створює новий сокет для обробки вхідного запиту на з'єднання
Bind()	Пов'язує сокет з локальною кінцевою точкою для очікування вхідних запитів на з'єднання
Close()	Закриває сокет
Connect()	Встановлює з'єднання з віддаленим хостом
Listen()	Поміщає сокет в режим прослуховування. Призначений тільки для серверних додатків
Receive()	Отримує дані від сполученого сокета
Select()	Перевіряє статус одного або декількох сокетів
Send()	Відправляє дані сполученому сокету
Poll()	Відправляє статус сокета
SetSocketOption	Встановлює опцію сокета
Shutdown()	Забороняє операції отримання і відправки на сокеті

Перерахування AddressFamily вказує схеми адресації для розв'язання адреси.

Параметр SocketType підтримує наступні параметри:

Dgram	Підтримує дейтаграми. Значення Dgram вимагає вказати Udp для типу протоколу і InterNetwork в параметрі адрес.
Raw	Підтримує доступ до базового транспортного протоколу.
Stream	Підтримує потокові сокети. Вимагає вказати TCP для типу протоколу і InterNetwork в параметрі адрес.

СПИСОК ЛІТЕРАТУРИ

1. VMware Workstation 5.0 User's Manual.
2. <http://www.vmware.com/support/ws5/doc/index.html> - VMware Workstation 5.0 Documentation.
3. http://www.vmweekly.com/articles/networking_in_vmware/1 - Networking in VMware. Andrei Baronov.
4. <http://www.vmggu.ru/articles/10/1/> - Типы файлов виртуальных машин компании VMware.
5. <http://www.vmggu.ru/articles/42/1/> - 10 сакральных вопросов и ответов по виртуальным машинам VMware.
6. <http://citforum.univ.kiev.ua/nets/hard/RJ45/> - РазВитой Ethernet или Обжимаем RJ45. Александр Севко.
7. <http://citforum.univ.kiev.ua/nets/hard/connector/> - Разъемы для оптики. В.Каток, И.Руденко, А.Ковтун.
8. <http://citforum.univ.kiev.ua/nets/optic/optic1.shtml> - Введение в технику волоконно-оптических сетей. Спирин А. А.
9. <http://www.intuit.ru/department/os/winadmin/5/> - Лекция: Подключения к рабочим группам.
10. Злобін Г.Г, Рикалюк Р.Є. Архітектура та апаратне забезпечення ПЕОМ: Навч.посіб. –К., 2006.

ЛАБОРАТОРНА РОБОТА № 11

Налаштування стеку протоколів TCP/IP для роботи з DHCP-сервером в VMware 5.0 на ОС Windows та Linux

Мета роботи: Навчитись налаштовувати локальну мережу між хостовою ОС та віртуальною машиною. Ознайомитись з функціями протоколу **DHCP** (автоматизація процесу призначення IP-адрес). Отримати відомості про налаштування TCP / IP для роботи з DHCP сервером. Вивчити особливості налаштувань в ОС Windows та Linux на прикладах Windows 7, та Fedora Linux4.

Теоретичні відомості

Протокол DHCP (автоматизація процесу призначення IP-адрес)

IP-адреси можуть призначатися адміністратором мережі вручну. Проте при великій кількості вузлів можуть виникати певні труднощі, щодо запам'ятовування та розподілу адрес. Ситуація ускладнюється ще тим, що багато користувачів не володіють достатніми знаннями для того, щоб конфігурувати свої комп'ютери для роботи в мережі і повинні тому покладатися на адміністраторів.

Протокол Dynamic Host Configuration Protocol (DHCP) був розроблений для вирішення ряду описаних проблем. Основним призначенням DHCP є динамічне призначення IP-адрес. Проте, окрім динамічного, DHCP може підтримувати і більш прості способи ручного і автоматичного статичного призначення адрес.

В ручній процедурі призначення адрес активну участь бере адміністратор, який надає DHCP-серверу інформацію про відповідність IP-адрес фізичним адресам або іншим ідентифікаторам клієнтів. Ці адреси повідомляються клієнтам у відповідь на їх запити до DHCP-серверу.

При автоматичному статичному способі DHCP-сервер привласнює IP-адресу (і, можливо, інші параметри конфігурації клієнта) з простору наявних IP-адрес без втручання оператора. Між ідентифікатором клієнта і його IP-адресою як і раніше, як і при ручному призначенні, існує постійна відповідність. Воно встановлюється у момент первинного призначення сервером DHCP IP-адреси клієнту. При всіх подальших запитах сервер повертає ту ж саму IP-адресу.

При динамічному розподілі адрес DHCP-сервер видає адресу клієнту на обмежений час, що дає можливість згодом повторно використовувати IP-адреси іншими комп'ютерами. Динамічний розподіл адрес дозволяє будувати IP-мережу, кількість вузлів в якій набагато перевищує кількість IP-адрес, що є у розпорядженні адміністратора.

DHCP забезпечує надійний і простий спосіб конфігурації мережі TCP/IP, гарантуючи відсутність конфліктів адрес за рахунок централізованого управління їх розподілом. Адміністратор управляє процесом призначення адрес за допомогою параметра "тривалості оренди" (lease duration), яка визначає, як довго комп'ютер може використовувати призначену IP-адресу, перш ніж знову надіслати запит щодо присвоєння йому адреси від серверу DHCP.

Прикладом роботи протоколу DHCP може служити ситуація, коли комп'ютер, що є клієнтом DHCP, видаляється з підмережі. При цьому призначена йому IP-адреса автоматично звільняється. Коли комп'ютер підключається до іншої підмережі, то йому автоматично призначається нова адреса. Ні користувач, ні мережний адміністратор не втручаються в цей процес. Ця властивість дуже важлива для мобільних користувачів.

Протокол DHCP використовує модель клієнт-сервер. Під час старту системи комп'ютер-клієнт DHCP, посилає повідомлення discover (досліджувати), яке ширококомовно(broadcast) поширюється по локальній мережі і передається всім DHCP-серверам приватної мережі. Кожний DHCP-сервер, що одержав це повідомлення, відповідає на нього повідомленням offer (пропозиція), яке містить IP-адресу і конфігураційну інформацію.

Комп'ютер-клієнт DHCP переходить в стан "вибір" і збирає конфігураційні пропозиції від DHCP-серверів. Потім він вибирає одну з цих пропозицій, переходить в стан "запит" і відправляє повідомлення request (запит) тому DHCP-серверу, чия пропозиція була вибрана.

Вибраний DHCP-сервер посилає повідомлення DHCP-acknowledgment (підтвердження), що містить ту ж IP-адресу, яка вже була послана раніше на стадії дослідження, а також параметр оренди для цієї адреси. Крім того, DHCP-сервер посилає параметри мережній конфігурації. Після того, як клієнт одержить це підтвердження, він переходить в стан "зв'язок", знаходячись в якому він може брати участь в роботі мережі TCP/IP. Комп'ютери-клієнти, які мають локальні диски, зберігають одержану адресу для використання при подальших стартах системи. При наближенні моменту закінчення терміну оренди адреси комп'ютер намагається відновити параметри "оренди" в DHCP-серверу, а якщо ця IP-адреса не може бути виділена знову, то йому повертається інша IP-адреса.

В протоколі DHCP описується декілька типів повідомлень, які використовуються для виявлення і вибору DHCP-серверів, для запитів інформації про конфігурацію, для продовження і дострокового припинення ліцензії на IP-адресу. Всі ці операції направлені на те, щоб звільнити адміністратора мережі від утомливих рутинних операцій по конфігуруванню мережі.

Проте використання DHCP несе в собі і деякі проблеми. По-перше, це проблема узгодження інформаційної адресної бази в службах DHCP і DNS. Як відомо, DNS служить для перетворення символьних імен в IP-адреси. Якщо

IP-адреси будуть динамічно змінюватися сервером DHCP, то ці зміни необхідно також динамічно вносити в базу даних серверу DNS. Хоча протокол динамічної взаємодії між службами DNS і DHCP вже реалізований деякими фірмами (так звана служба Dynamic DNS), стандарт на нього поки не прийнятий.

По-друге, нестабільність IP-адрес ускладнює процес управління мережею. Системи управління, засновані на протоколі SNMP, розроблені з розрахунком на статичність IP-адрес. Аналогічні проблеми виникають і при конфігуруванні фільтрів маршрутизаторів, які оперують з IP-адресами.

Нарешті, централізація процедури призначення адрес знижує надійність системи: при відмові DHCP-серверу всі його клієнти виявляються не в змозі одержати IP-адресу і іншу інформацію про конфігурацію. Наслідки такої відмови можуть бути зменшені шляхом використання в мережі декількох серверів DHCP, кожний з яких має свій унікальний набір IP-адрес.

Реалізація **TCP / IP** дозволяє вузлу **TCP / IP** використовувати статичні IP-адреси або отримати IP-адресу автоматично за допомогою **DHCP-сервера** (Dynamic Host Configuration Protocol-протокол динамічної конфігурації хоста).

Для простих мережевих конфігурацій, заснованих на локальних мережах (*LAN, Local Area Network*), він підтримує автоматичне призначення IP-адрес.

За промовчанням комп'ютери клієнтів, що працюють під управлінням ОС **Windows** або **Linux**, отримують інформацію про налаштування протоколу **TCP / IP** автоматично від служби **DHCP**.

Однак навіть у тому випадку, якщо в мережі доступний **DHCP-сервер**, необхідно призначити статичну IP-адресу для окремих комп'ютерів в мережі. Наприклад, комп'ютери з запущеною службою **DHCP** не можуть бути клієнтами **DHCP**, тому вони повинні мати статичну IP-адресу.

Якщо служба **DHCP** недоступна, можна встановити **TCP / IP** для використання статичної IP-адреси.

Для кожної плати мережного адаптера в комп'ютері, що використовує **TCP / IP**, можна встановити IP-адресу, маску підмережі та шлюз за замовчуванням.

Якщо сервер з запущеною службою **DHCP** доступний у мережі, він автоматично надає інформацію про параметри **TCP / IP** клієнтам **DHCP**.

Виконання роботи

Завдання . Створіть IP-калькулятор в табличному процесорі для полегшення формування маски підмережі.

1. Відкрийте табличний процесор і сформууйте таблицю за наступним шаблоном:

Зразок оформлення таблиці

Далі необхідно ввести в комірки **B5, J5, R5, Z5** формули для переведення двійкового представлення IP-адреси у точково десяткову нотацію за октетами.

2. Введіть у комірку **B5** формулу для перетворення 1-го октета IP-адреси в десяткову систему числення:

$$= I4 * 2^{i2} + H4 * 2^{H2} + G4 * 2^{G2} + F4 * 2^{F2} + E4 * 2^{E2} + D4 * 2^{D2} + C4 * 2^{C2} + B4 * 2^{B2}$$

3. Скопіюйте введену формулу в інші комірки (**J5, R5, Z5**).
4. Самостійно введіть у комірки **B5, J5, R5, Z5** формули для перетворення

	1-й октет								2-й октет								3-й октет								4-й октет							
біти	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	ідентифікатор (ID) мережі																								ID вузла							
IP-адреса	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
Десятковий запис	192								0								1								255							
Маска підмережі	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
Десятковий запис	255								255								255								248							

маски підмережі з двійкового подання в точково десяткову нотацію.

5. Збережіть файл у своїй папці з ім'ям *IPCALC*.

Продовження роботи трохи згодом.

НАВЧАЛЬНЕ ВИДАННЯ

Роман Євстахович Рикалюк

ЛАБОРАТОРНИЙ ПРАКТИКУМ з курсу «КОМП'ЮТЕРНІ МЕРЕЖІ»

Редактор

Підп. до друку **.**.05. Формат 60x84/16. Папір друк.№2. Друк на різогр.
Умовн.фарбовідб. 0.9.
Обл.-вид.арк. 1.0. Тираж 200 прим. Зам. **.

Видавничий центр Львівського національного університету імені Івана Франка
79000, Львів, вул.Дорошенка, 41