МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра Комп'ютерних систем та мереж

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Комп'ютерних систем та мереж

_____(Жуков І.А.)

«____» _____2022 р.

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"МАГІСТР»

Тема: Управління Інтернет-каналом за допомогою маршрутизаторів на платформі Mikrotik

Виконавець: Кузьменко А.Ю.

Керівник: Андрєєв О. В.

Нормоконтролер: Андрєєв О.В.

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж
Напрям (спеціальність) 123 "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних систем та
мереж

_____(Жуков І.А.)

« ____ » _____2022 р.

ЗАВДАННЯ

на виконання дипломного проекту студента

Кузьменко Антон Юрійович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): Управління Інтернет-каналом за допомогою маршрутизаторів на платформі *Mikrotik*
затверджена наказом ректора від "06" вересня 2022р. № 1266/ст
2. Термін виконання проекту (роботи): з 05 вересня 2022 р. по 30 листопада 2022 р.
3. Вихідні данні до проекту (роботи): Аналіз існуючих комп'ютерних мереж, а також засобів керування комп'ютерними мережами та управління Інтернет каналом користувачів цих мереж. Технології управління Інтернет каналом. Підбір апаратно-програмних засобів для забезпечення можливості керування Інтернет каналом.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): Дослідження мережі Інтерент та програмних основ її функціонування. Дослідження засобів керування Інтернет каналом, Вибір апаратно-програмних засобів. Приклад налаштувань засобів управління Інтернет каналом.
5. Перелік обов'язкового графічного матеріалу: презентація *PowerPoint.*

6. Календарний план-графік

№ п/п	Етапи виконання дипломного проекту(роботи)	Термін виконання етапів	Примітка
1.	Огляд літератури	05.09.22-11.09.22	виконано
2.	Розгляд вимог до керування Інтернет каналом.	16.09.22-19.09.22	виконано
3.	Аналіз Інтернет мереж	20.09.22-30.09.22	виконано
4.	Аналіз технологій передачі даних в мережі Інтернет	05.10.22-14.10.22	виконано
5.	Аналіз технологій управління Інтернет каналом	15.10.22-20.10.22	виконано
6.	Розробка мережі з можливістю керування доступом до мережі Інтернет	21.10.22-30.10.12	виконано
7.	Оформлення пояснювальної записки	31.10.22-06.11.22	виконано
8.	Виготовлення графічного матеріалу	07.11.22-10.11.22	виконано

7. Дата видачі завдання 05 вересня 2022 р.

Керівник дипломної роботи(проекту)_____ Андрєєв О.В.
(підпис)

Завдання прийняв до виконання_____ Кузьменко А.Ю.
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту спеціаліста «Управління Інтернет-каналом за допомогою маршрутизаторів на платформі *Mikrotik*», 98 с., 16 рисунків, 1 додаток, 20 літературних джерел.

МЕРЕЖА, ПРОТОКОЛ, ЕМУЛЯТОР, МОДЕЛЮВАННЯ, ВІРТУАЛЬНА МАШИНА.

Об'єкт дослідження – Доступ до мережі Інтернет комп'ютерів мережі побудованої в окремій організації. Предмет дослідження – керування Інтернет-каналом в комп'ютерній мережі організації.

Мета дипломного проекту – Побудова комп'ютерної мережі окремої організації з можливістю керування Інтернет каналом.

Метод дослідження – моделювання і аналіз комп'ютерної мережі організації з побудовою окремих відділів та розділення Інтернет-трафіку між ними.

Досліджено спеціалізоване програмне забезпечення для роботи в локальних комп'ютерних мережах, яке використовується при керуванні Інтернет каналом, та доступом до ресурсів в мережі. Під час використання мережі користувач має можливість користуватися всіма можливостями глобальної мережі, які не суперечать політиці організації щодо використання ресурсів.

Матеріали дипломного проекту рекомендується використовувати в практичній діяльності адміністраторам комп'ютерних систем та мереж. На даний час розробка впроваджена і використовується деякими приватними організаціями.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Кіберслекінг *(CyberSlacking)*- проблема так званого неділового використання Інтернету на робочому місці.

Інтерне́т (від англ. *Internet*) — всесвітня система взаємосполучених комп'ютерних мереж, що базуються на комплекті Інтернет-протоколів.

Провайдер - компанія, яка забезпечує вихід в Інтернет.

Шлюз (*gateway*) - це комп'ютер або система комп'ютерів зі спеціальним програмним забезпеченням, що дозволяє зв’язатися двом мережам з різними протоколами.

LAN (Local Area Network)- локальні обчислювальні мережі.

WAN (Wide Area Network)- глобальні обчислювальні мережі.

TCP (Transmission Control Protocol) – протокол що відповідає за те, як буде проходити інформація в всесвітній мережі.

IP (Internet Protocol) – протокол що відповідає за те, куди буде надсилатися інформація, тобто завідує адресацією пакетів.

WAP (*Wireless Access Protocol)* - протокол бездротового доступу.

Точка присутності *(Points of Presence, POP)*- точки, в яких мережа Інтернет-провайдера зв'язується з мережами інших Інтернет-провайдерів.

ISO – Міжнародна організація по стандартизації (*International Standartization Organization*)

MAN – регіональні мережі (*Metropolitan Area Networks*)

OSI – модель взаємодії відкритих систем (*Open Systems Interconnection*)

WAN – глобальна мережа (*Wide Area Network*)

ВСТУП

Що таке Інтернет? Наведемо декілька прикладів визначення слова Інтернет, які найбільш популярні в даний момент.

Інтернет (від *international* (міжнародний) і *net* (Мережа)) - це всесвітня, кооперативно керована сукупність комп'ютерних мереж різного рівня і підпорядкованості, рівноправні але обмінюються інформацією за допомогою базових протоколів *TCP / IP*. Інтернет включає локальні мережі, шлюзи, сервери та комп'ютери, розташовані по всьому світу. Структура Інтернету нагадує павутину, в вузлах якої знаходяться сервери, об'єднані між собою лініями зв'язку. Вузли, з'єднані високошвидкісними інформаційними каналами, і складають базис Інтернету.

З появою Інтернету життя людей істотно змінилося. Скільки усього може подарувати всесвітня павутина. Багато користувачів Інтернету, переважно молодь, вже не можуть собі уявити, що б вони робили без мережі. Зараз навіть існують бізнесмени, що заробляють гроші в Інтернеті. Але мережа окрім заробітку дарує безліч можливостей.

Зв'язок

Електронні листи вже давно витіснили звичайні поштові послання. Писати текст на комп'ютері зручніше, іноді швидше, а доставка листа відбувається значно швидше і надійніше. І це притому, що навіть немає необхідності платити гроші за відправлення послань.

Чат, аська і скайп - нові онлайн засобу зв'язку. За допомогою чату можна послати повідомлення адресата, що знаходиться в сотнях кілометрах від посилача. Аська дозволяє писати короткі повідомлення своїх друзів і знайомих, а за допомогою скайпа можна взагалі безкоштовно здійснювати дзвінки у будь-яку точку світу. Скайп - програма, що є цифровим диктофоном і телефоном в одному флаконі. Голосова інформація спочатку записується, потім стискається і кодується, а після цього пересилається за вказаною адресою, декодується і відтворюється.

Закачування інформації

Це одна з найпоширеніших причин, по якій люди хочуть користуватися Інтернетом. За допомогою інтернету можна легко викачати необхідний файл за невеликий проміжок часу, що залежить від швидкості з'єднання. Закачувати можливо текстові документи, фільми, музику, кліпи, стислі теки, програми і багато що інше.

Транслятор

Інтернет-транслятор. Він може ставати телевізором або радіоприймачем, передаючи інформацію. В результаті користувач через свій комп'ютер може дивитися серіали, фільми, слухати радіо.

Новий світ

Всесвітня павутина стає абсолютно іншим світом - віртуальним. А у новому світі - нова валюта. Так в Інтернеті існують електронні гроші, що є еквівалентами доларам, гривням і іншим світовим грошовим одиницям. Через мережу можна оплачувати комунальні рахунки, послуги стільникового зв'язку, купувати домени і так далі

Подарунок геймерам

Окрім усього перерахованого Інтернет ще дозволяє молодим людям грати по мережі в різні онлайн гри. Так, знаходячись в Росії, можна грати з американцями, японцями, французами і людьми іншої національності. Деякі компанії, розробляючи комп'ютерні ігри, створюють сервери - спеціальні місця для ігор, де збираються любителі ігор з усього світу.

Інтернет стає невід'ємною частиною нашого життя як зручний інструмент для роботи. Сфера Інтернет-послуг в останні роки розширюється неймовірно швидко, і тепер мережею користуються навіть ті компанії і окремі користувачі, у яких всього три роки тому персональний комп'ютер був межею мрій. Однак, як і будь-яка інша хороша річ, Інтернет втрачає всі свої переваги, якщо їм

починають зловживати. На заході проблема так званого неділового використання Інтернету на робочому місці стала настільки поширеною, що отримала спеціальну назву - кіберслекінг *(CyberSlacking)*, а десятки компаній, що розробляють програмне забезпечення, зайнялися системами управління доступом до Інтернет і випустили цілий ряд апаратних і програмних рішень для управління доступом в мережу.

Якої шкоди може завдати кіберслекінг сучасній компанії?

- Зловживання Інтернетом завжди означає зниження продуктивності праці службовців, частина робочого часу яких витрачається на перегляд розважальних ресурсів.
- Значно збільшується вхідний Інтернет-трафік, що призводить, з одного боку, до збільшення витрат компанії, а з іншого - до зниження продуктивності корпоративної мережі.

Таким чином, стає очевидно, що боротися зі зловживанням Internet потрібно. Залишилося лише вирішити, яким чином

Мета і завдання дипломного проекту

1. Аналітичний огляд існуючих методів управління Інтернет-каналом;
2. Розробка моделі комп'ютерної мережі офісу;
3. Розробити методику, встановити та налаштували апаратні засоби керування;
4. Розробка модернізованої моделі мережі офісу з впровадженою системою керування Інтернет-каналом.

Об'єкт проекту – комп'ютерна мережа в офісі компанії.

Предмет проекту – організація доступу в мережу Інтернет для декількох підрозділів компанії з розмежуванням мереж підрозділів в окремі сегменти.

Методи дослідження – моделювання і аналіз комп'ютерної мережі між персональними комп'ютерами та мережевому маршрутизаторі, який встановлений в серверній шафі.

РОЗДІЛ 1 ОСНОВНІ ВІДОМОСТІ ПРО МЕРЕЖУ ІНТЕРНЕТ

1.1. Загальне поняття про Інтернет

Інтерне́т (від англ. *Internet*) — всесвітня система взаємосполучених комп'ютерних мереж, що базуються на комплекті Інтернет-протоколів. Інтернет також називають мережею мереж. Інтернет складається з мільйонів локальних і глобальних приватних, публічних, академічних, ділових і урядових мереж, пов'язаних між собою з використанням різноманітних дротових, оптичних і бездротових технологій. Інтернет становить фізичну основу для розміщення величезної кількості інформаційних ресурсів і послуг, таких як взаємопов'язані гіпертекстові документи Всесвітньої павутини (*World Wide Web — WWW*) та електронна пошта.

В повсякденній мові слово Інтернет найчастіше вживається в значенні Всесвітньої павутини і доступної в ній інформації, а не у значенні самої фізичної мережі. Також вживаються терміни Всесвітня мережа, Глобальна мережа чи навіть одне слово Мережа, Інéт, Тенета, Міжмережжя, Інтернéтрі або Нéтрі. Все частіше Інтернет вживається і з малої літери, що можна пояснити паралелями з термінами «радіо», «телебачення», які пишуть з малої.

Історія Інтернету сягає досліджень 1960-х років, які проводилися на замовлення уряду США і мали на меті створення надійних розподілених комп'ютерних мереж, стійких до пошкоджень. Попередницею Інтернету стала мережа *ARPANET* (англ. *Advanced Research Projects Agency Network*), яка почавши функціонувати в кінці 1960-х, в кінці 1970-х об'єднувала близько 200 вузлів.

Урядове фінансування магістральної мережі Національного наукового

фонду США в 1980-х, а також приватне фінансування для інших комерційних магістральних мереж в усьому світі призвело до участі в розробці нових мережевих технологій і злиття багатьох мереж. Комерціалізація в 1990-х міжнародної мережі привела до її популяризації та впровадження в практично кожен аспект сучасного життя людини. З 2011 року більше 2,1 мільярда людей користуються послугами Інтернету.

Інтернет не має централізованого управління, правил використання чи доступу. Кожна складова мережа встановлює свої власні стандарти. Централізовано визначаються правила використання адресного простору Інтернет-протоколу та Системи доменних імен. Керує цим Інтернет корпорація з присвоєння імен та номерів (англ. *Internet Corporation for Assigned Names and Numbers*, або *ICANN*), міжнародна некомерційна організація з головним офісом у США. Технічне обґрунтування і стандартизацію основних протоколів (*IPv4* та *IPv6*) проводить *Internet Engineering Task Force (IETF)*, некомерційна організація, відкрите міжнародне співтовариство проектувальників, учених, мережевих операторів і постачальників послуг.

Мережа побудована на використанні протоколу *IP* і маршрутизації пакетів даних. В наш час Інтернет відіграє важливе значення у створенні інформаційного простору глобального суспільства, слугує фізичною основою доступу до веб-сайтів і багатьох систем (протоколів) передачі даних.

1.2. Структура мережі Інтернет

Образ Інтернету можливо представити як безкрайнього міста-магполіса, який складається з конгломерату окремих будинків, районів і кварталів, з'єднаних в одне ціле магістралями. Кожен такий окремий будинок - це самостійна автономна мережа, спільність яких за допомогою з'єднання каналами зв’язку і породжує Інтернет (рис. 1.1). А цементує основи Інтернету протокол *TCP / IP* – така мережева мова цього міста.

Становий хребет Інтернету складають його опорні мережі *(CoreBackbone Network)* провайдерів вищого рівня. Всі опорні мережі без обмежень обмінюються між собою Інтернет трафіком. Весь інший світ отримує доступ до

хребта Інтернету вже через провайдерів першого (транснаціонального) рівня, що мають вихід в різні країни. Слідом за провайдерами першого рівня розташовані мережеві провайдери другого рівня - національного і третього - регіонального, з'єднані між собою високошвидкісними каналами передачі даних, які надають доступ до Інтернету місцевим провайдерам - *Internet Service Provider*.

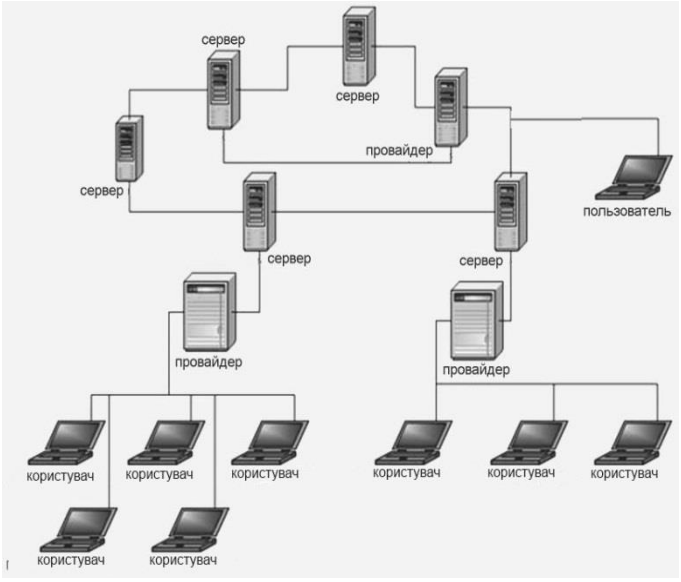

Рис. 1.1 Загальна схема глобальної мережі Інтернет

Провайдер - компанія, яка забезпечує вихід в Інтернет, тобто «постачальна» вас цією послугою. Саме провайдер на локальному рівні і забезпечує вихід в Інтернет індивідуальних користувачів. Кожен провайдер на своєму рівні вирішує всі організаційні, технічні та фінансові питання, представляючи в своїй особі перед вами всю всесвітню мережу.

Інтернет трафік - кількість переданої інформації,вимірюється в байтах.

Шлюз (*gateway*) - це комп'ютер або система комп'ютерів зі спеціальним програмним забезпеченням, що дозволяє зв’язатися двом мережам з різними протоколами. Найчастіше шлюзи зв’язують локальні обчислювальні мережі *LAN (Local Area Network)* з глобальною мережею *WAN (Wide Area Network).*

Маршрутизатор *(router)* - пристрій, який пов'язує мережі побудованих на основі одного протоколу, але різними типами мережевого обладнання.

Маршрутизатори зменшують трафік, пропускаючи в мережу тільки ті дані, які призначені саме для неї.

Протокол передачі даних - спеціальні набори правил, які забезпечують обмін інформацією як між окремими пристроями, так і між цілими мережами.

Комп'ютери, включені до світової мережі, мають абсолютно різну архітектуру і різне програмне забезпечення. Для забезпечення сумісності мереж були створені протоколи - спеціальні набори правил, забезпечуючи обмін інформацією як між окремими пристроями та процесами, так і між цілими мережами.

TCP (Transmission Control Protocol) відповідає за те, як буде проходити інформація з всесвітньої мережі. Він відповідає встановлення надійного з'єднання між комп'ютерами і пересилання даних, контролюючи оптимальний розмір пакетів даних, генеруючу повторну передачу пакету при збої.

IP (Internet Protocol) відповідає за те, куди буде надсилатися інформація, тобто завідує адресацією пакетів. Протоколом *TCP* виконується нарізка направлених файлів на пакети, кожен зі своїм точною адресою розміщення в структурі файлу. За місцем прибуття отримані фрагменти збираються в єдине ціле.

Головні особливості протоколів *TCP / IP*:

- Відкритість стандартів, що розробляються незалежно від програмного і апаратного забезпечення мережі;

- Незалежність від безпосередньої фізичної середовища передачі;

- Унікальність адресації;

- Стандартизованість протоколів високого рівня, які використовуються в сервісах.

Для того щоб інформація знайшла потрібну програму (адже на комп’ютері одночасно працює безліч різних програм різного призначення) існує система портів. Порт - спеціальний номер, який присвоюється кожному процесу на комп'ютері і який виконує роль адреси відправника та адреси одержувача на транспортному рівні.

Розглянемо структурні складові Інтернету, апаратне та програмне забезпечення. Розглянемо рис. 1.2.

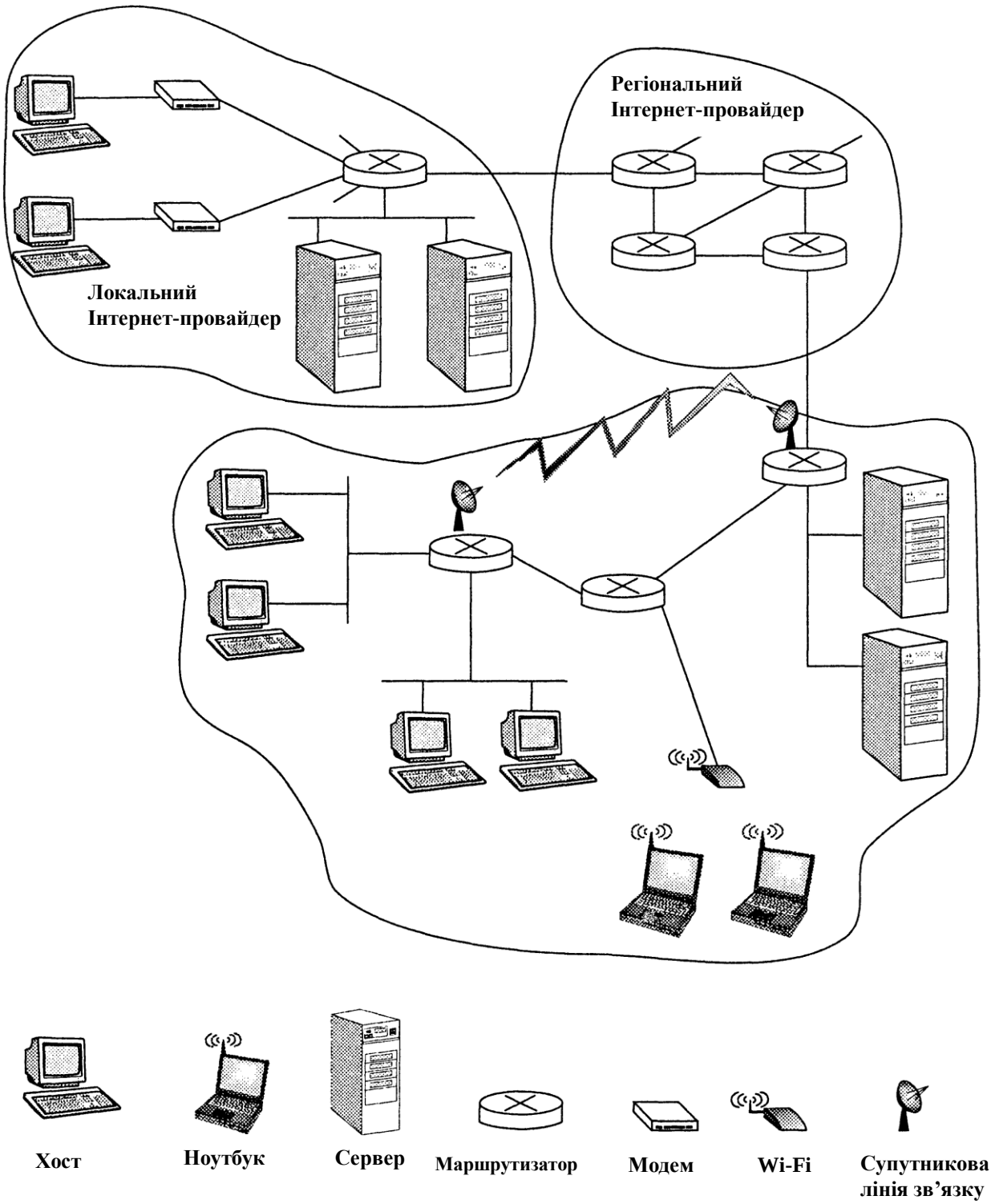

Рис. 1.2 Структурна схема Інтернету

Інтернет являє собою світову комп'ютерну мережу, яка зв'язує в єдине ціле мільйони обчислювальних пристроїв , розміщених в різних кутах земної кулі. Обчислювальні пристрої можуть бути настільними комп'ютерами, або так званими серверами, зберігаючи ми або передаючи інформацію, представлені в вигляді, наприклад, веб-сторінок або повідомлень електронної пошти. Останні

роки до Інтернету під'єднуються нетрадиційні кінцеві системи, такі як *PDA* (*Personal Digital Assistant* – персональний цифровий помічник), телевізори, мобільні комп'ютери, автомобілі і навіть холодильники. По оцінкам спеціалістів в 2002 році в Інтернеті нараховували від 100 до 500 мільйонів кінцевих пристроїв, так званих хостів.

Кінцеві пристрої зв'язані між собою лініями зв'язку. Існує велика кількість ліній зв'язку, які використовують різні типи фізичних носіїв: коаксіальні, мідні, оптоволоконні кабелі, лінії радіозв'язку ті інші. Лінія зв'язку визначає швидкість передачі даних, а максимальну швидкість передачі даних називають пропускною здатністю лінії та виміряють в бітах на секунду.

Хости далеко не завжди напряму з'єднані між собою єдиною фізичною лінією зв'язку. Навпроти, типовою є ситуація, коли зв'язок здійснюється за допомогою множини послідовних ліній, з'єднаних спеціальними комутуючими пристроями – маршрутизаторами. Маршрутизатор приймає порцію пакетів, передану по одному з його вхідних каналів зв'язку, а потім перенаправляє її в один зі своїх вихідних каналів зв'язку. В термінології комп'ютерних мереж передані порції даних називають пакетами. Послідовність каналів зв'язку та маршрутизаторів, через які проходить пакет в процесі передачі, називається маршрутом, або шляхом, пакета в мережі. Шлях пакета не відомий завчасно і визначається безпосередньо в процесі передачі пакета. В Інтернеті кожній парі кінцевих систем не надають виділений маршрут, тому що використовується технологія комутації пакетів, при цьому різні пари кінцевих систем можуть одночасно користуватися одним й тим самим маршрутом або його частиною. Перші мережі з комутацією пакетів, які були створені в далекі 70-ті роки, є «далекими родичами» сучасного Інтернету.

Доступ кінцевих систем до Інтернету здійснюється за допомогою постачальників послуг Інтернету, або Інтернет провайдерів (*Internet Service Provider, ISP*). Інтернет-провайдери поділяються на резидентних, університетських та корпоративних. Інтернет-провайдер надає мережу маршрутизаторів і канали зв'язку. Як правило, Інтернет-провайдери пропонують декілька способів підключення хостів до мережі: комутоване модемне з'єднання,

резидентне широко полосне підключення за допомогою кабельного модему або цифрової абонентської лінії *(Digital Subscriber Line, DSL)*, високошвидкісний доступ через локальну мережу *(Local Area Netwokr, LAN)*, а також безпровідний доступ. Крім того, Інтернет-провайдери здійснюють пряме підключення до мережі веб-сайтів. Для забезпечення зв'язку між віддаленими користувачами, а також для надання доступу до інформації, яка зберігається в Інтернеті, місцеві Інтернет-провайдери підключаються до Інтернет-провайдерів національної або інтернаціональної ланки. Останні використовують високошвидкісні маршрутизатори, з'єднані оптоволоконними кабелями. Кожний Інтернет-провайдер як нижньої, так верхньої ланки є адміністративною одиницею, яка передає дані по протоколу IP та дотримується угод про імена та адреси, прийнятих в Інтернеті.

Кінцеві системи, маршрутизатори та інші компоненти Інтернету використовують протоколи, які здійснюють управління прийомом та передачі інформації всередині Інтернету. Найбільш важливим протоколом в глобальній мережі є *TCP (Transmission Control Protocol* – протокол управління передачею*)* та *IP (Internet Protocol* – Інтернет-протокол*)*. Протокол *IP* визначає формат пакетів, які передаються між хостами та маршрутизаторами. Стек основних протоколів, які використовуються в Інтернеті, відомий під назвою *TCP/IP* .

Те що ми зазвичай називаємо словом «Інтернет», - це так званий «відкритий Інтернет». Крім загальнодоступного Інтернету існує багато закритих (приватних) комп'ютерних мереж, побудованих по тому ж принципу що і глобальна мережа. Як правило, приватні мережі призначені для використання всередині різних фірм та організацій. Вони не можуть обмінюватися інформацією з зовнішньою мережею, за виключенням повідомлень, які проходять через так звані брандмауери, контролюючі потік повідомлень, які входять чи виходять з мережі. Подібні мережі об'єднують терміном інтранет. Ця назва співзвучна назві «Інтернет» і відображає той факт що в зачинених мережах використовують такі ж самі хости, маршрутизатори, канали зв'язку та протоколи, канали зв'язку, що і в відкритому Інтернеті. З точки зору технологій і розвитку існування Інтернету забезпечується створенням, перевіркою та

впровадженням Інтернет-стандартів. Стандарти виробляє «проблемна група розробок для інтернету» *(Internet Engineering Task Force, IETF)*. Документи які створює група мають назву *RFC (Requests For Comments)*

1.3. Мережа інтернет з погляду обслуговування

З точки зору обслуговування Інтернет можливо поділити на декілька пунктів:

- Інтернет дозволяє розподіленим програмам, працюючих на кінцевих системах, здійснювати обмін даними між собою. В число таких програм входять віддалений робочий стіл, електронна пошта, засоби навігації в веб, засоби передачі відео та аудіо даних, Інтернет телефонія, мережеві комп'ютерні ігри, засоби обміну даними та інше. Маємо підкреслити що веб це не окрема комп'ютерна мережа, а одне з багатьох розподілених програмних засобів.
- Інтернет надає своїм розподіленим програмам два типи служб: надійну службу з встановленням логічного з'єднання та ненадійну службу без встановлення логічного з'єднання. Ці поняття означають наступне: надійна служба з встановленням логічного з'єднання гарантує, що дані які передаються відправником будуть доставлені отримувачу повністю (без втрат та пошкоджень) та в початковому порядку; ненадійна служба без встановлення логічного з'єднання не надає ніяких гарантій відносно доставки. Як правило розподілене програмне забезпечення підтримує один з двох типів передачі.
- На даний час Інтернет не дає гарантій відносно того, скільки часу знадобиться для передачі даних від відправника до отримувача. І, якщо не врахувати можливість підвищення пропускної здатності каналу доступу до вашого Інтернет-провайдера, ми можемо затребувати в Інтернеті більш високої якості обслуговування, якщо готові заплатити за це.

1.4. Сучасні мережеві протоколи

Любий рух інформації в Інтернеті між двома або більше пристроями підпорядковується протоколу. Так протоколи маршрутизаторів визначають шлях

пакета від відправника до отримувача; реалізовані апаратно протоколи мережевих карт двох фізично з'єднаних комп'ютерів контролюють потік бітів, які передаються по мережевому кабелю; протоколи контролю перезавантаження, які використовуються в кінцевих системах, потрібні для контролю частоти передачі пакеті. Інтернет повністю заснованих на протоколах.

В якості прикладу, одночасно простого і який наглядно ілюструє суть мережевого протоколу, розглянемо що відбувається в момент, коли ми відтворюємо запит до веб серверу. Графічно дану ситуацію ілюструє рис. 1.3. Спочатку комп'ютер відсилає серверу повідомлення із запитом на встановлення з'єднання і чекає на відповідь. Сервер приймає запит та відправляє повідомлення у відповідь, яке підтверджує встановлене з'єднання. Таким чином, знаючи, що тепер можливо запросити веб-документ, комп'ютер відсилає серверу ім'я ресурсу, а сервер повертає заданий ресурс користувачу.

Розглянувши приклади протоколів які демонструють дві найбільш значимими складових протоколу – повідомлення і дія, можу сформулювати наступне визначення: Протокол визначає формат та чергу повідомлень, якими обмінюються два або більше пристроїв, а також дії, які виконуються при передачі або прийомі повідомлень. Протоколи дуже широко використовуються взагалі в мережах, так і в мережі Інтернет.

1.5. Ядро комп'ютерних мереж

Предметом розгляду стане взаємодія між маршрутизаторами, точніше механізми передачі даних від одного хоста до іншого. Елементи структури мережі, які відносяться до ядра показані на рис. 1.3.

1.5.1. Комутації каналів та пакетів в мережі

Існує два фундаментальних підходу до організації ядра мережі: комутація каналів та комутація пакеті. При комутації каналів відбувається

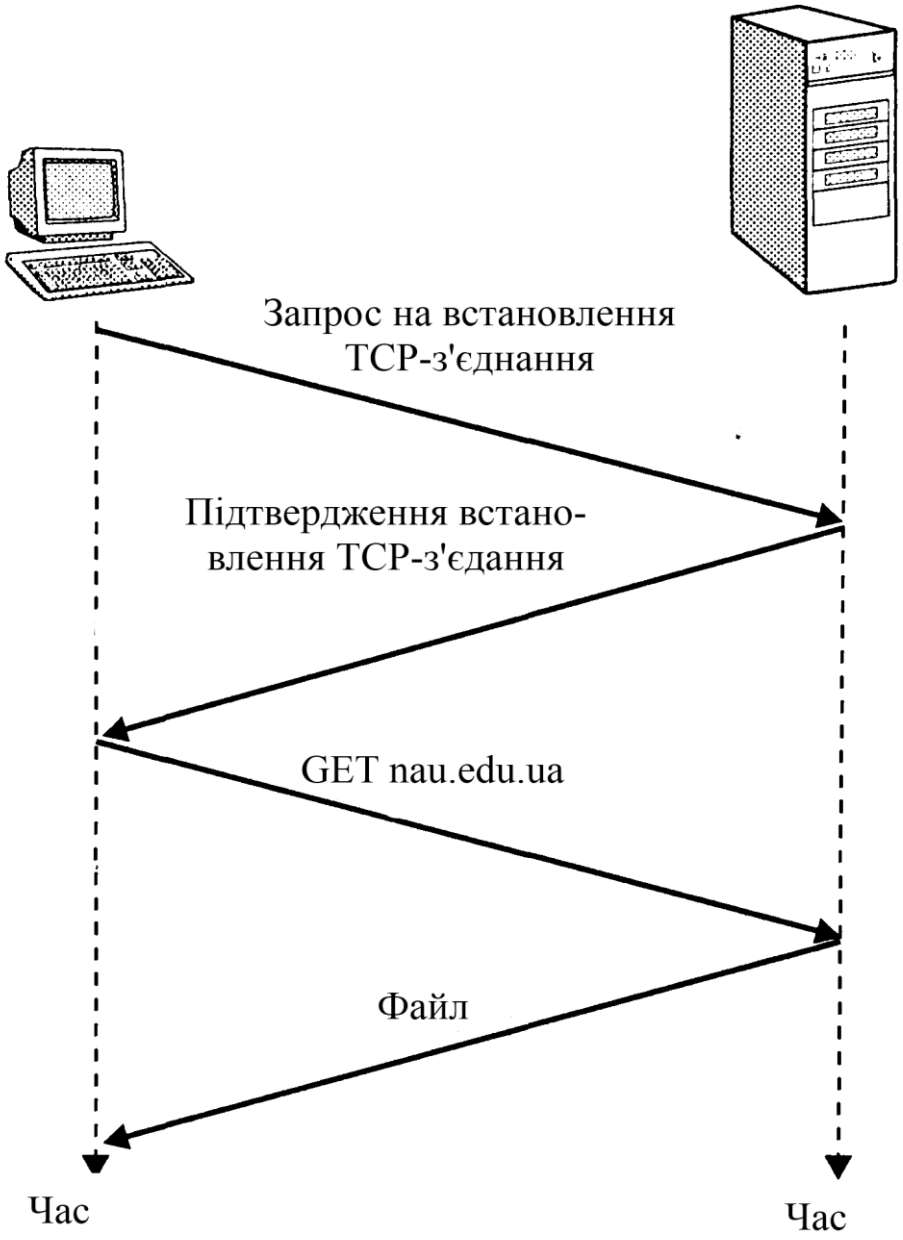

Рис. 1.3 Протокол спілкування між комп’ютерами

резервування на час сеансу зв’язку необхідних ресурсів на всьому мережевому шляху. При комутації пакетів ресурси запитуються при необхідності і виділяються за потрібністю. Іноді декілька повідомлень можуть намагатися використати лінію зв’язку одночасно, тому існує необхідність в організації черги повідомлень.

Сучасний Інтернет є типовою системою з комутацією пакетів. Як правило, пакет проходить через множину каналів, однак ніяких резервувань частотних полос при цьому не відбувається. У випадку перенавантаження каналу, пакет буде змушений чекати в черзі звільнення. Таким чином з точки зору швидкодії Інтернет намагається доставити пакет з максимальними зусиллями, час доставки

не гарантовано.

В мережах з комутацією каналів комутатори з'єднані меж собою лініями зв'язку. Кожна з ліній може підтримувати одночасно *n* каналів зв'язку. Хости напряму з'єднані з одним із комутаторів. Між парами хостів встановлюється виділене сковзне з'єднання. Таким чином, щоб хост А мав можливість передавати пакети хосту Б, потрібно зарезервувати одну полосу частот на кожній з ліній зв'язку, з'єднуючих хост А і Б.

Прихильники технології комутації пакетів завжди звертали увагу на серйозний недолік мереж з комутацією каналів, який заключається в тому, що виділені канали неможливо звільнити під час простою. Іншою причиною за яку комутація каналів викликає обґрунтовану критику, це необхідність в складному сигнальному обладнанні для управління комутаціями і виділенням частотних полос каналам зв'язку.

В сучасних комп'ютерних мережах відбувається автоматичне розбиття великих за об'ємом повідомлень на більш малі фрагменти, пакети. Пакет є одиницею передачі даних. При передачі пакет проходить через послідовність ліній зв'язку та комутаторів, звичайно названих маршрутизаторами. Передача пакета по лінії зв'язку здійснюється монопольно, з максимальною швидкістю, яку може забезпечити лінія зв'язку. Більшість маршрутів використовують механізм передачі з проміжним накопиченням. Це означає, що перед тим як почати передачу в вихідну лінію зв'язку, маршрутизатору потрібно завершити процес прийому пакета в буфер. Таким чином в маршрутизаторах виникає затримка накопичення, зумовлена необхідністю очікування закінчення прийому пакета.

Кожний маршрутизатор має множину вхідних і вихідних ліній зв'язку. Кожна вихідна лінія має буфер, який називають вихідним буфером. В буфері зберігаються пакети, призначені для передачі по лінії зв'язку. Буфери грають ключову роль в механізмі комутації пакетів. Якщо при закінченні прийому пакета виявляється що лінія зв'язку зайнята, то пакет ставиться в чергу в вихідному буфері. Таким чином крім затримки накопичення в буфері в маршрутизаторах присутня затримка очікування. Затримки очікування є

змінними показниками і залежать від завантаженості каналу. Оскільки розміри буферів обмежені, можлива ситуація коли місця в буфері буде недостатньо для розташування нового пакету. В такому випадку виникне втрата пакету – буде втрачений або новий пакет, або один за пакетів які знаходяться в черзі.

На рис. 1.4 приведена структура простої мережі з комутацією пакетів. Тут пакети представлені у вигляді тривимірних брусків. Ширина бруска відповідає довжині пакету. В даному прикладі всі пакети мають однакому довжину. Припустимо що хости А та В посилають пакету хосту Е, при цьому зв'язок хостів А та В з першим маршрутизатором здійснюється за допомогою ліній зв’язку *Ethernet*. Маршрутизатор направляє пакет на зовнішню лінію зв’язку. Якщо лінія перенавантажена, пакети очікують її звільнення в черзі. Подивимося, що відбувається при одночасній передачі пакетів хостами А і В. Очевидно, що ніякої синхронізації між хостами немає, і, відповідно, неможливо завчасно передбачити порядок передачі пакетів. Цю особливість називають стичним мультиплексором.

Противники комутації пакетів часто висувають тезис про те, що комутація пакетів не дозволяє організувати мережеве обслуговування в реальному часі (наприклад забезпечити передачу звука та відео), пояснюючи це непередбачуваними затримками в при передачі пакетів всередині мережі. Прихильники комутації пакетів помічають що дана технологія дає можливість більш ефективно організувати розподілення пропускної можливості лінії зв’язку, а також є більш простою, ефективною та дешевшою.

1.5.2. Сегментування повідомлень в мережі

В більшості сучасних мережах з комутацією пакетів передаючий хост розбиває довгі повідомлення, які генерує програма, на менші пакети. Ці пакети отримує адресат, з яких збирає вихідне повідомлення. Значною перевагою розбиття на пакети заключається в тому, що час передачі повідомлення, як правило, значно скорочується порівняно з передачею повідомлення цілим.

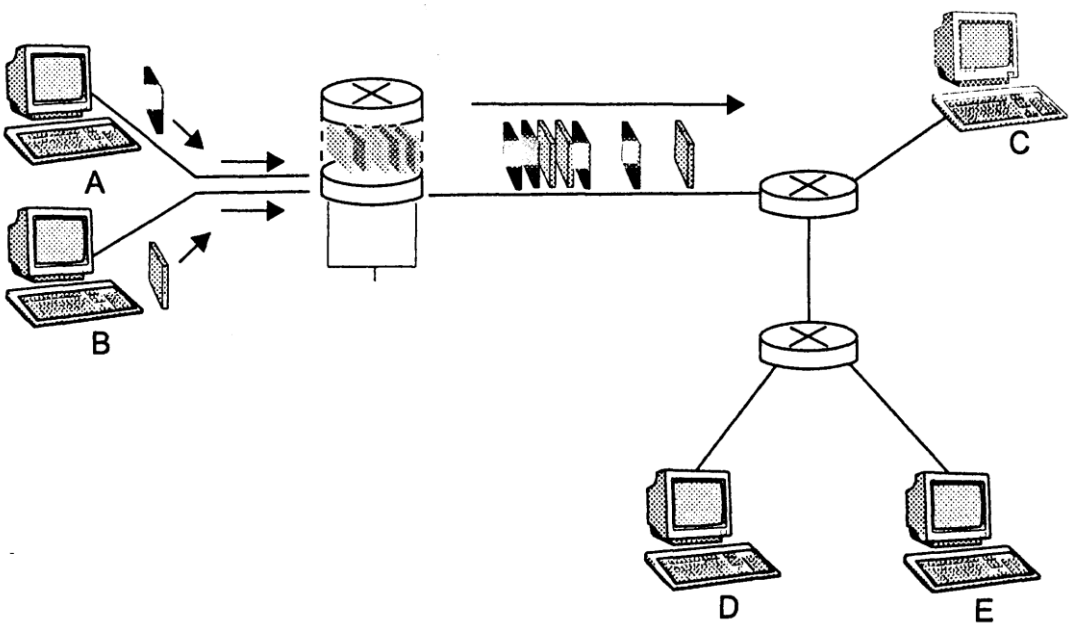

Рис. 1.4 Структура мережі з комутацією пакетів

1.5.3. Передача повідомлень в мережі

Існує два основних класи комп'ютерних мереж з комутацією пакетів: дейтаграмні мережі та мережі з віртуальним каналом. Ці два класи відрізняються між собою механізмом передачі пакетів в мережі. Мережі, в я яких передача здійснюється на основі аналізу адреси отримувача, називають дейтаграмними. Дейтаграмний спосіб передачі характерний для Інтернету. Якщо все ж таки в мережі використовується механізм передачі з віртуальним каналом, то кажуть мережі з віртуальними каналами. До останніх відносять мережі які підтримують протокол *X.25*, ретрансляцію кадрів, асинхронний режим передачі.

Віртуальний канал (*Virtual Channel, VC*) характеризується трьома складовими:

- Маршрутом, по якому передаються всі пакети від відправника до отримувача.
- Номерами віртуального каналу, по одному номеру на кожну лінію зв'язку, які створюють маршрут.
- Записами в таблицях трансляції номерів віртуального каналу, які є в кожному з комутаторів на маршруті.

Після того як з'єднання між отримувачем та відправником встановлено

(створено віртуальний канал), відправник може почати пересилку пакетів з відповідними номерами віртуального каналу. Оскілки кожна лінія зв'язку має свій номер віртуального каналу, кожний раз при проходженні пакету через комутатор, останній повинен автоматично змінювати для пакета значення номеру віртуального каналу. Новий номер віртуального каналу пакет отримує за допомогою таблиці трансляції номерів віртуального каналу.

Концепцію віртуального каналу ілюструє рис. 1.5. Припустимо що хост А запросив віртуальний канал з хостом *B*,і мережа встановила віртуальний канал з маршрутом *A-PS1-PS2-B*, назначивши лініям зв'язку номери 12, 22 та 32 відповідно. Таким чином, кожний пакет який відправляється через хост А, має номер 12, а пакети які відправляються за маршрутизаторів *PS1* та *PS2*, номери 22 та 32 відповідно.

В дейтаграмній мережі кожний пакет що передається включає в себе інформацію про адресу отримувача, який має ієрархічну структуру. Кожний раз при отриманні пакета комутатор аналізує фрагмент адреси пакета і направляє його на відповідну лінію зв'язку. Комутатор забезпечений таблицею маршрутизації, який зв'язує кінцеві адреси або її частини з лініями зв'язку. Після зчитування заголовка відбувається виділення адреси, який використовується в якості індексу таблиці маршрутизації.

Дейтаграмні мережі, на відміну від мереж з віртуальними каналами, не використовують інформацію про стан з'єднань в своїх комутаторах. Фактично люба мережа, побудована на дейтаграмній передачі, не контролює інформаційні потоки всередині себе, оскілки рішення про шлях слідування любого пакета приймається виключно на основі його адреси призначення і не залежить від з'єднання між хостами. Простота дейтаграмного механізму дає привід для критичних нарікань на адресу віртуальних каналів, відносно складності останніх. Прихильники віртуальних каналів відповідають на ці нарікання тим, що технологія забезпечує краще мережеве обслуговування програм.

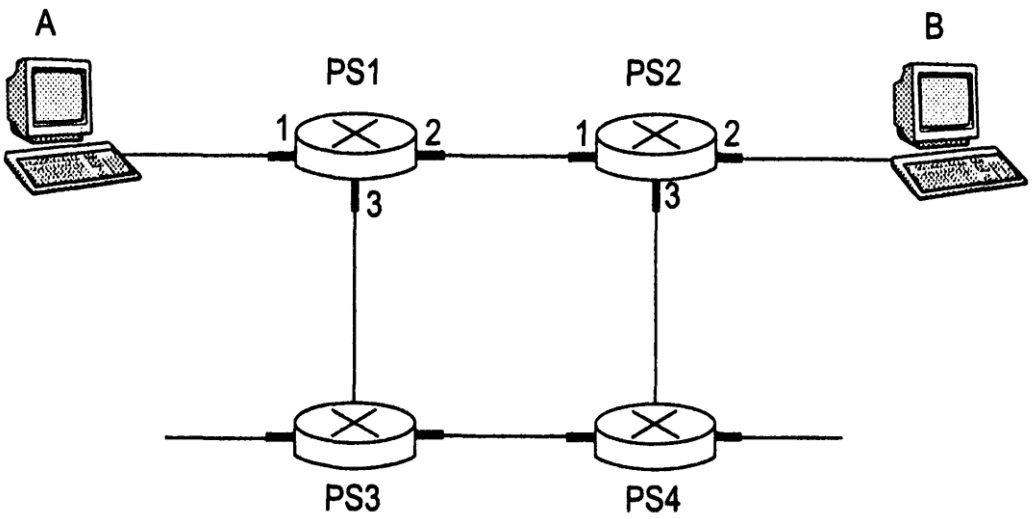

Рис. 1.5 Проста мережа з віртуальним каналом

1.6. Доступ до комп'ютерної мережі та її фізична середа

Доступ до мережі можливо умовно класифікувати наступним чином:

- Резидентний доступ використовується для підключення до мережі домашніх кінцевих систем.
- Корпоративний доступ використовується для підключення до мережі кінцевих систем, що належать приватним або державних організаціях.
- Мобільний доступ, що використовується для підключення різних портативних пристроїв.

Але, часто на практиці зустрічаються порушення приведеної класифікації, тому ці поняття відповідають тільки як найбільш типові випадки доступу.

1.6.1. Резидентний доступ

Спочатку, як правило, резидентний доступ здійснювався за допомогою модему та комутованої телефонної лінії через підключення до місцевого Інтернет-провайдера. Модем перетворював цифрові сигнали домашньої системи в аналогові сигнали, які передаються через телефонний кабель, який являє собою мідну виту пару. Сигнал приймається стороною Інтернет провайдера, де модем здійснює зворотне перетворення аналогових сигналів в цифрові та передає їх на вхід маршрутизатора. Таким чином типовий резидентний доступ до мережі забезпечується парою модемів, з'єднаних за допомогою комутованої лінії. На

сьогодні модеми дозволяють забезпечити швидкість передавання даних за швидкістю 56 Кбіт/с, що, в наші часи, вважається дуже малою.

Ця проблема отримала своє рішення у вигляді нових широкополосних засобів передачі інформації. Існують два основні засоби широкополосного доступу: цифрові абонентські лінії *(Digital Subscriber Line, DSL)* та оптоволоконно-коаксиалні кабелі *(Hybrid Fiber Coaxial Cable, HFC).*

*Я*к правило *DSL*-доступ забезпечується телефонними компаніями, іноді сумісно з Інтернет-провайдерами. *DSL*-технологія нагадує звичайних модемний доступ по телефонному кабелю, але дозволяє, за рахунок скорочення відстані від користувача до модему Інтернет-провайдера, значно підвищити швидкість передавання інформації. Звичайно швидкість між обміну між сторонам асиметричні, при цьому швидкість передачі у напрямку користувача значно вище, чим швидкість передачі в сторону Інтернет-провайдера.

1.6.2. Корпоративний доступ

*Я*к правило в державних або приватних організаціях доступ до Інтернету здійснюється за допомогою локальних мереж *(LAN)*, які з'єднують кінцеві системи з периферійним маршрутизатором. Серед усіх технологій локальних мереж, найбільш поширеною є технологія *Ethernet*. В ній для з'єднання кінцевих систем між собою та периферійним маршрутизатором використовується мідна вита пара. На сто годній цей спосіб підключення до Інтернету широко використовується для підключення домашніх систем.

1.6.3. Мобільний доступ

Прорив в області бездротових технологій, як і виникнення глобальної Мережі, привів до значних змін у сфері телекомунікацій. В 2000 році в Європі було більше власників мобільних телефонів, ніж власників машин або персональних комп'ютерів. В наш час існує два основні засоби безпровідного підключення до глобальної мережі. Бездротові локальні мережі дозволяють користувачам обмінюватися даними через базову станцію, часто звану точкою безпровідного доступу, перебуваючи на відстані десятків метрів від неї. Як

правило, базова станція має підключення до Інтернету за допомогою кабелю і здатна поєднати користувачів з глобальною мережею. У бездротових мережах з віддаленим доступом базова станція управляється постачальником телекомунікаційних послуг і забезпечує доступ користувачів на відстані до десятків кілометрів.

Бездротові локальні мережі, засновані на технології *IEEE 802.11b*, відомого як бездротова *Ethernet*-мережа *і Wi-Fi*, в даний час отримують масове поширення в різних навчальних, комерційних, розважальних організацій і навіть при домашньому користуванні. Подібні технології, реалізовані в будівлі, дозволяють користувачам задіяти електронну пошту або займатися подорожам по веб-сторінкам в будь-якій точці цієї будівлі.

Зараз спостерігається розширення кола домашніх користувачів, які разом з широкосмуговим доступом застосовують недорогі бездротові локальні мережі для створення потужних домашніх комп'ютерних мереж. Мережа включає в себе портативний та персональний комп'ютери, базову станцію, з якою пов'язаний портативний комп'ютер, кабельний модем, який здійснює підключення персонального комп'ютера до Інтернету, і, нарешті, маршрутизатор, що з'єднує базову станцію і персональний комп'ютер з кабельним модемом. Описана мережа дозволяє двом членам сім'ї мати широкосмуговий доступ в Інтернет, причому один з них може при цьому ходити з однієї кімнати в іншу.

Отримуючи доступ до Інтернету через бездротову локальну мережу, ви пов'язані необхідністю знаходитися на відстані не більше декількох десятків метрів від базової станції. Це допустимо для домашнього або корпоративного користування. Якщо ви знаходитеся в машині або за містом на відпочинку, на допомогу приходять технології мобільного телефонного зв'язку, які забезпечують доступ до глобальної Мережі на відстані десятків кілометрів від базової станції.

Технології *WAP* (*Wireless Access Protocol* - протокол бездротового доступу). *WAP*-телефони нагадують звичайні мобільні телефони, проте мають дещо збільшений екран і забезпечують низькошвидкісний доступ в Інтернет. Замість мови *HTML* в *WAP*-телефонах використовується мова розмітки *WML*

(*WAP Markup Language*), оптимізований під низькошвидкісний доступ і невеликий екран. Протокол *WAP* в Європі підтримується стандартом мобільного зв'язку *GSM*. У зв'язку з поширенням технології *GPRS* (*General Packet Radio Service* - основна служба радіотрансляції пакетів) очікується зростання популярності *WAP*.

Зараз телекомунікаційні компанії роблять великі інвестиції в безпровідні технології третього покоління (*Third Generation, 3G*), які дозволять здійснювати віддалений бездротовий доступ в Інтернет з комутацією пакетів на швидкостях не нижче 384 Кбіт/с. *3G* -системи забезпечать високошвидкісний доступ до веб-ресурсів і інтерактивного відео, а також телефонний зв'язок з більш високою якістю звуку, ніж у звичайних телефонних мережах.

1.7. Фізична середа передачі даних в комп'ютерній мережі

Передача між пристроями відбувається шляхом поширення електромагнітних хвиль або оптичних сигналів у фізичному середовищі. Фізична середа може приймати вельми різноманітні форми, причому на шляху прямування пакета ці форми можуть змінюватися. Прикладами фізичних середовищ є мідна вита пара, коаксіальний кабель, багатомодовий оптоволоконний кабель, територіальні та супутникові радіоканали.

Фізичні середовища можна розділити на два типи: провідні і безпровідні. Провідні середовища передачі припускають присутність твердотільного провідника і включають оптоволоконний кабель, мідну виту пару і коаксіальний кабель. У бездротовому середовищі передача здійснюється без участі твердих провідників; цей тип середовища використовується в безпровідних локальних мережах і при супутниковому зв'язку.

Мідна вита пара

Мідна вита пара є найдешевшим і найбільш популярним видом кабелів. Протягом більш ніж 100 років вита пара активно використовується в телефонних мережах. Можна сміливо стверджувати, що більше 99% всіх кабелів, з'єднуючих

абонентів з телефонними комутаторами, є мідними витими парами. Вита пара складається з двох ізольованих мідних дротів товщиною 1 мм, укладених в спіральну оболонку. Усередині оболонки дроти переплетені один з одним, щоб знизити рівень електричних перешкод, що виникають між парою провідників. Зазвичай перед поміщенням пар всередину кабелю їх забезпечують додатковими захисними екранами.

Неекранована вита пара *(Unshielded Twisted Pair, UTP)*, як правило, використовується в офісних локальних мережах, розташованих в одній будівлі.

З появою в 80-ті роки оптоволоконних ліній зв'язку багато фахівців прогнозували, що вони з часом повністю витіснять низькошвидкісну виту пару. Однак вита пара виявилася не настільки безперспективною. Неекрановану виту пара категорії 5 дозволяє отримати швидкість передачі даних 100 Мбіт / с на відстанях до сотні метрів. На менших відстанях можна добитися ще більшої швидкості. Цей тип кабелю ще довго може займати домінуюче положення в сфері локальних офісних мереж. Вита пара активно використовується для резидентного доступу в Інтернет.

Коаксіальний кабель

Коаксіальний кабель, як і вита пара, складається з двох мідних провідників, але ці провідники, на відміну від витої пари, розташовані не паралельно, а концентрично (коаксіально). Із застосуванням особливих видів ізоляції і екранування коаксіальний кабель дозволяє домогтися більш високих швидкостей передачі даних, ніж кручена пара. Коаксіальні кабелі поділяються на два види: з не модульованою передачею і з модульованою передачею. Коаксіальний кабель з не модульованою передачею має опір 50 Ом і товщину близько 1 см; до його фізичних переваг можна віднести легкість і гнучкість. Коаксіальний кабель з модульованою передачею має опір 75 Омі і має велику товщину, вагу і меншу гнучкість в порівнянні з кабелем з не модульованою передачею. Часто кабель з модульованим передачею використовують в системах кабельного телебачення. Лінії кабельного телебачення в поєднанні зі спеціальними кабельними модемами здатні забезпечити зв'язок користувачів з

Інтернетом на швидкості до 20 Мбіт/с і вище. При передачі по кабелю з модульованим передачею відбувається попередня модуляція (перенесення) аналогових сигналів в потрібну смугу частот.

Оптоволоконна середовище передачі являє собою тонкий і гнучкий кабель, всередині якого поширюються світлові імпульси, що несуть інформацію про передані біти. Навіть простий оптоволоконний кабель здатний передавати дані на величезних швидкостях в десятки і навіть сотні гігабіт на секунду. Оптоволоконні лінії не схильні електричним наведенням, мають дуже низький рівень ослаблення сигналу на одиницю довжини і мають значну стійкість до механічних впливів. Перераховані переваги зробили оптоволоконні лінії зв'язку вельми привабливою технологією для передачі інформації на великі відстані, особливо для міжнародних і міжконтинентальних комунікацій.

Оптоволоконне середовище

Активно використовується для передачі даних в Інтернеті. Однак висока вартість оптичних пристроїв (маршрутизаторів, приймачів і передатчиків) робить недоцільним (з економічних причин) застосування оптоволоконних ліній зв'язку для передачі на короткі відстані, наприклад, в локальних офісних мережах або для резидентного домашнього доступу.

Територіальні радіоканали

Радіоканали передають сигнали за допомогою електромагнітних хвиль радіодіапазону. Їх перевага полягає в тому, що для зв'язку не потрібно твердотільної провідника сигналів (отже, немає необхідності в його прокладці), тобто користувач може бути мобільним, є потенціал у збільшенні відстані передачі. Характеристики радіоканалу залежать від середовища передачі радіохвиль і відстані між кінцевими системами. До факторів середовища передачі відносять загасання сигналу внаслідок поширення в середовищі, проходження через поглинаючі предмети, взаємодії з відбитими електромагнітними хвилями, а також хвилями, що виходять від інших джерел випромінювання.

Супутникові радіоканали

Супутник зв'язку організовує взаємодію між двома або більше наземними прийомопередавачами. Він приймає сигнали одного частотного діапазону, проводить їх регенерацію за допомогою повторювача, а потім передає сигнали в іншому частотному діапазоні. Існують два типи супутників: геостаціонарні і низькоорбітальні.

Відмінною рисою геостаціонарних супутників є те, що вони не міняють свого положення щодо заданої точки земної поверхні. Це досягнуто шляхом поміщення супутника на орбіту, віддалену приблизно на 36000 км від земної поверхні. Значну відстань, яку потрібно долати сигналу, обумовлює велику затримку його поширення, 250 мс. Проте супутникові канали, за допомогою яких не складає ніяких труднощів досягти швидкостей передачі в сотні мегабіт в секунду, активно використовуються в телефонії та Інтернеті.

Низькоорбітальні супутники знаходяться значно ближче до земної поверхні, ніж геостаціонарні, і обертаються навколо неї, подібно місяцю. Для постійного покриття певних ділянок земної поверхні доводиться розміщувати на орбіті кілька супутників. В даний час є чимале число проектів низькоорбітальних систем зв'язку. Зокрема, в майбутньому передбачається використання подібних систем для передачі даних в Інтернеті.

1.8. Інтернет провайдери і магістралі Інтернету

У загальнодоступному Інтернеті мережі доступу з'єднуються з іншими мережами за допомогою ієрархії мереж Інтернет-провайдерів, зображеної на рис. 1.6. Знизу ієрархії перебувають мережі резидентних Інтернет-провайдерів, до яких звичайно підключаються кінцеві системи. Верхня частина ієрархії представлена мережами так званих Інтернет-провайдерів першої ланки. З одного боку, мережі цих Інтернет-провайдерів володіють типовими рисами комп'ютерних мереж - наявністю маршрутизаторів і зв'язків з іншими мережами. З іншого боку, мережі Інтернет-провайдерів першої ланки мають

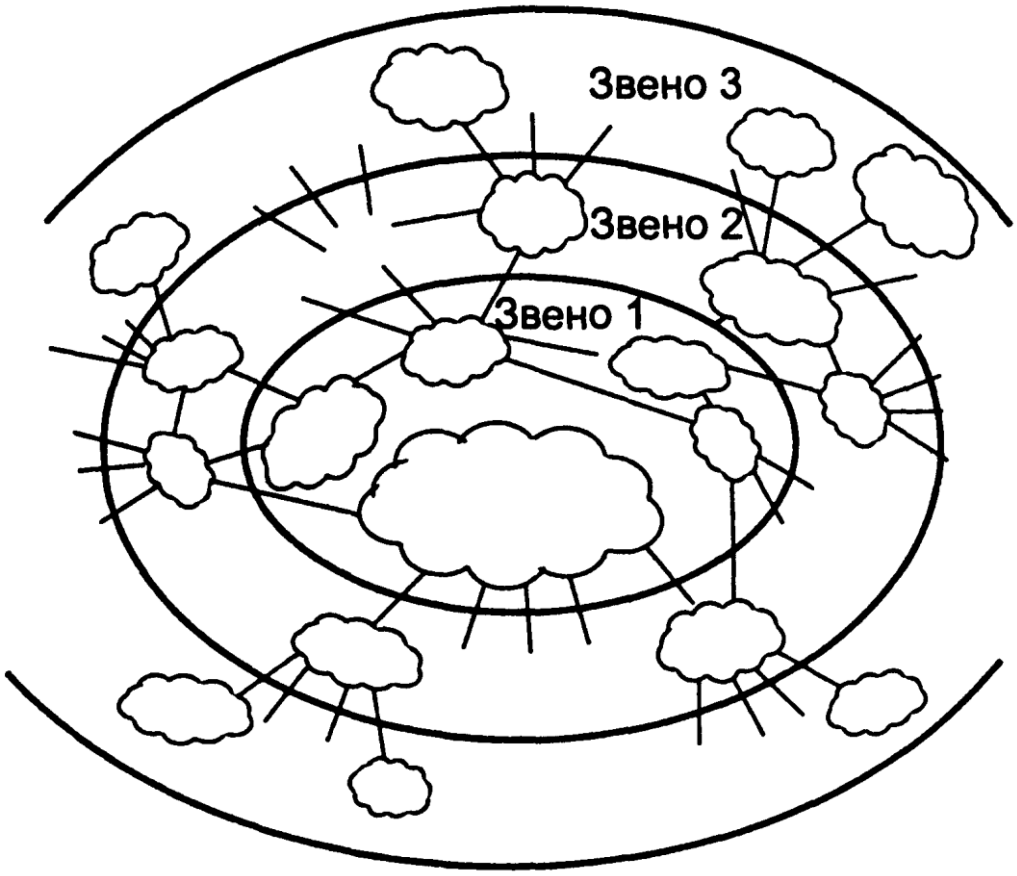

Рис. 1.6 Зв’язок між мережами Інтернет-провайдерів

свою специфіку. По-перше, їх лінії зв'язку зазвичай забезпечують швидкість передачі не нижче 622 Мбіт/с, а іноді 2,5-10 Гбіт/с. По-друге, маршрутизатори мереж Інтернет-провайдерів першої ланки повинні функціонувати з гранично високою швидкістю для того, щоб не викликати затримок пакетів. По-третє, всі мережі Інтернет-провайдерів першої ланки безпосередньо сполучені між собою. По-четверте, до кожної мережі Інтернет-провайдера першої ланки підключено велику кількість мереж Інтернет-провайдерів другої ланки та інших комп'ютерних мереж. На-кінець, по-п'яте, район мережевого охоплення Інтернет-провайдера першої ланки є міжнародним.

Мережі Інтернет-провайдерів першої ланки часто називають магістралями Інтернету. Мережі Інтернет-провайдерів другої ланки, як правило, мають регіональну територію охоплення і підключаються до декількох мереж Інтернет-провайдерів першої ланки. Кажуть, що Інтернет-провайдери другої ланки є споживачами послуг Інтернет-провайдерів першої ланки. Великі компанії та установи підключають свої корпоративні мережі безпосередньо до мереж Інтернет-провайдерів другої і навіть першої ланок і вважаються споживачами їх

послуг. Споживачі оплачують послуги своїх Інтернет-провайдерів.

Мережі Інтернет-провайдерів другої ланки також можуть з'єднуватися між собою і обмінюватися інформацією без участі Інтернет-провайдерів першої ланки. Нижче в ієрархії розташовані мережі Інтернет-провайдерів, які підключаються до мереж Інтернет-провайдерів другої ланки (до однієї або декількохим).

На останньому щаблі ієрархії перебувають мережі доступу. Заплутують попередню концепцію, полягає в тому, що деякі Інтернет-провайдери першої ланки одночасно можуть бути Інтернет-провайдерами другої ланки, до мереж яких приєднані мережі крупних організацій та Інтернет-провайдерів нижчих ланок.

Точки, в яких мережа Інтернет-провайдера зв'язується з мережами інших Інтернет-провайдерів (розташованих вище, нижче або на одному ієрархічному рівнів), називаються точками присутності *(Points of Presence, POP)*. Як правило, точка присутності являє собою одну або декілька груп маршрутизаторів мережі, до яких підключені маршрутизатори іншої мережі. У Інтернет-провайдерів першої ланки зазвичай є безліч точок присутності в різних географічних регіонах. Для підключення до Інтернет-провайдеру більш високої ланки споживач зазвичай орендує високошвидкісні лінії зв'язку у будь-якої телекомунікаційної компанії (що є «третьою стороною» в угоді) і з'єднує свої маршрутизатори з точкою присутності Інтернет-провайдера. Можливі з'єднання двох мереж одночасно в декількох точках присутності.

Крім сполучення через точки присутності, використовується також механізм з'єднання через точки доступу в мережу *(Network Access Points, NAP)*, кожна з яких може належати сторонній телекомунікаційній компанії або магістральному Інтернет-провайдеру і управлятися ними. Зазвичай подібна схема використовується при підключенні мереж Інтернет-провайдерів другої ланки один до одного і до мереж Інтернет-провайдерів першої ланки, а Інтернет-провайдери першої ланки частіше воліють сполучати свої мережі між собою через точки присутності. Оскільки в точках доступу необхідно забезпечувати комутацію і передачу величезних обсягів інформації в одиницю часу, їх

реалізація вельми непроста технологічно.

Топологія Інтернету є складною і складається з декількох десятків мереж Інтернет-провайдерів першого та другого ланок і сотень мереж менш великих Інтернет-провайдерів регіонального і локального масштабу. Останні підключаються до перших, які, в свою чергу, сполучені між собою за допомогою точок присутності або точок доступу.

1.9. Стек протоколів мережі Інтернету

Комунікаційна модель Інтернету складається з п'яти рівнів: фізичного, канального, мережевого, транспортного і прикладного. Замість термінів «одиниця обміну мережевого рівня »,« одиниця обміну канального рівня »і т. д. ми будемо використовувати спеціальні імена. Одиниці обміну канального рівня ми назвемо кадрами, одиниці обміну мережевого рівня - дейтаграммами, одиниці обміну транспортного рівня - сегментами, а одиниці обміну прикладного рівня - повідомленнями. Для одиниць обміну фізичного рівня зазвичай не передбачається спеціального імені. Комунікаційна модель Інтернету та одиниці обміну її рівнів зображені на рис. 1.7.

Підтримка протоколів може бути апаратною, програмною або змішаною. Протоколи прикладного рівня, такі як *HTTP* і *SMTP*, а також протоколи транспортного рівня практично завжди підтримуються програмно. Навпроти, протоколи фізичного і канального рівнів, тісно пов'язані з середовищем передачі даних, підтримується апаратно мережевою картою. Мережевий рівень, що знаходиться в центрі комунікаційної моделі, може підтримуватися як апаратно, так і програмно.

Далі дано характеристики кожного з п'яти рівнів комунікаційної моделі Інтернету.

Прикладний рівень

Прикладний рівень, як випливає з його назви, призначений для підтримки мережевих програм. Є безліч протоколів прикладного рівня, з яких найбільш важливими є *HTTP* (для подорожей по веб-сторінкам), *SMTP* (для електронної

пошти) і *FTP* (для обміну файлами). Розробка власного протоколу прикладного рівня не становить особливих труднощів.

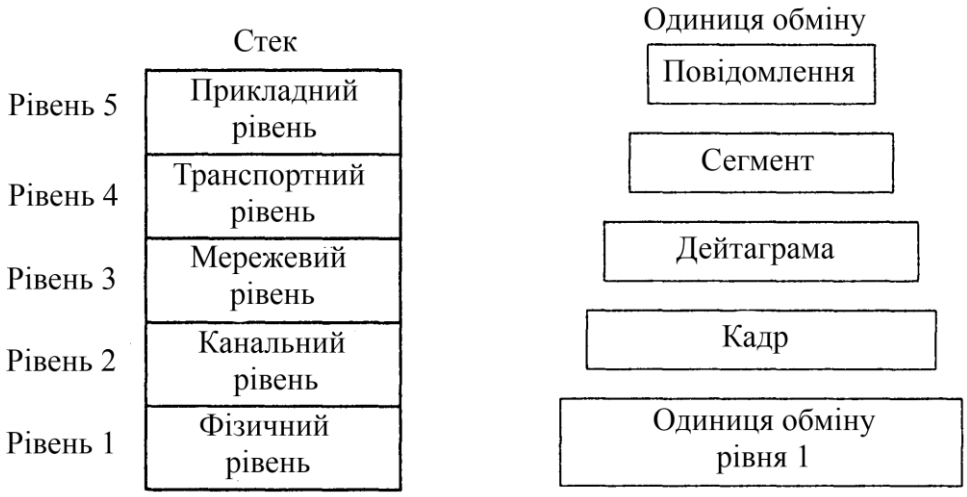

Рис. 1.7 Стек протоколів Інтернету та одиниці обміну різних рівнів

Транспортний рівень

Головна функція транспортного рівня полягає у передачі повідомлень прикладного рівня між клієнтом і сервером. В Інтернеті існують два транспортні протоколи: *TCP* і *UDP*. Протокол *TCP* забезпечує передачу з встановленням логічного з'єднання, тобто надійну передачу з контролем переповнення. Крім того, *TCP* виробляє розбиття довгих повідомлень на більш короткі і контролює перевантаження. Контроль перевантаження зводиться до примусового зниження швидкості передачі кінцевої системи при високому завантаженні мережі. Протокол *UDP* забезпечує передачу повідомлень без встановлення логічного з'єднання, тобто не надійний вид зв'язку, де допускаються спотворення і втрати даних.

Мережевий рівень

Мережевий рівень забезпечує передачу дейтаграм між двома хостами і базується на двох основних протоколах. Перший протокол визначає поля дейтаграми та інтерпретацію їх вмісту маршрутизаторами і кінцевими системами. Цей протокол є єдиним протоколом мережевого рівня в Інтернеті і називається *IP*. Другим протоколом є один з численних протоколів

маршрутизації, призначених для визначення шляхів дейтаграм від відправника до адресата. Число протоколів маршрутизації величезне. Інтернет є мережею мереж, а кожна мережа підтримує власний протокол маршрутизації, і визначається адміністратором мережі. Незважаючи на функціональні відмінності між протоколом *IP* і протоколами маршрутизації, а також на широке розмаїття останніх, їх зазвичай об'єднують під загальним іменем *IP*, підкреслюючи цим їх сполучну роль в організації глобальної мережі.

Протокол транспортного рівня (*TCP* або *UDP*) передає сегмент і адреса призначення протоколу *IP* мережного рівня, а протокол *IP* мережного рівня доставляє сегмент кінцевому хосту і передає його назад транспортному рівню.

Канальний рівень

Мережевий рівень забезпечує передачу пакета через серію маршрутизаторів між кінцевими системами. Для переміщення пакета (дейтаграми) від одного вузла до іншого мережевий рівень вдається до службам канального рівня. Таким чином, основна функція канального рівня полягає у передачі дейтаграм між вузлами на маршруті.

Канальний рівень використовує спеціальний протокол, орієнтований на використану лінію зв'язку. Іноді протоколи канального рівня забезпечують надійну передачу між вузлами. Відмінність надійної передачі на транспортному і канальному рівнях: протокол *TCP* забезпечує надійність на всьому шляху проходження повідомлення, а протокол канального рівня – лише між парою вузлів. До протоколів канального рівня відносяться *Ethernet* і *PPP*; іноді аналогічні функції несуть технології асинхронної передачі даних *(ATM)* і ретрансляції кадрів. Оскільки шлях від відправника до адресата звичайно складаються з ланцюжка різнорідних ліній зв'язку, передача дейтаграми може здійснюватися різними канальними протоколами.

Фізичний рівень

Якщо призначенням канального рівня є передача кадрів між сусідніми вузлами мережі, то фізичний рівень забезпечує передачу між вузлами окремих бітів інформації. Протоколи фізичного рівня також безпосередньо залежать від

використаної лінії зв'язку (мідної витої пари, одномодового оптоволокна і т. п.). Технологія *Ethernet* підтримує безліч протоколів фізичного рівня, призначених для підтримки кручений пари, коаксіального кабелю, оптоволоконного кабелю і деяких інших видів ліній. У кожній з ліній зв'язку механізми передачі біта різні.

1.10. Висновки до розділу

Ми розглянули програмні та апаратні компоненти, з яких складаються як комп'ютерні мережі взагалі, так і Інтернет зокрема. Ми почали з «периферії» комп’ютерних мереж, торкнувшись кінцевих систем та програм, а також транспортних послуг, що надаються програмам, запущеним на кінцевих системах.

Потім в центрі нашої уваги опинився самий нижній з точки зору комунікаційної моделі фізичний рівень передачі даних: ми розглянули основні середовища передачі і технології доступу до глобальної мережі. Ми також познайомилися зі структурою Інтернету, представили її як мережа мереж і дізналися про те, що ієрархія мереж Інтернет-провайдерів дозволила глобальній мережі з легкістю включати в себе нові сегменти.

Інша частина була присвячена основним аспектам функціонування комп'ютерних мереж. Спочатку ми розглянули основні причини затримок і втрат пакетів в процесі передачі. Ми ознайомилися з концепціями багаторівневої комунікаційної моделі та протоколами кожного з рівнів, складових архітектурну основу комп'ютерних мереж.

РОЗДІЛ 2 МЕТОДИ УПРАВЛІННЯ ІНТЕРНЕТ КАНАЛОМ

2.1. Класичні елементи системи управління трафіком в комп'ютерній мережі

Обмеження вихідного трафіку (*shaping*) - це механізм, за допомогою якого пакети затримуються перед передачею з тим, щоб швидкість передачі відповідала бажаної. Це один з найбільш часто використовуваних механізмів управління трафіком. Як побічний ефект, даний механізм дозволяє згладжувати вибухонебезпечне трафік.

Планування (*scheduling*) - це механізм, який дозволяє упорядковувати або змінювати порядок об'єкти між входом і виходом конкретної черги. Прикладами планувальника можуть послужити алгоритми *FIFO, SQF, WRR* та інші.

Класифікація (*classifying*) - механізм, що розділяє пакети для різної обробки, можливо в різні черги. У процесі прийому, маршрутизації і передачі пакетів, мережеве пристрій може по-різному їх класифікувати. Це може бути маркування, яке зазвичай відбувається на кордоні мережі з єдиним адмініструванням, або ж може виконуватися індивідуально на кожному проміжному вузлі.

Обмеження вхідного трафіку (*policing)* - черговий елемент системи якості обслуговування, що обмежує трафік. Цей механізм приймає пакети до певної швидкості, а над частиною трафіку перевищила заданий поріг виконується певна дія. Наприклад, можна знищувати трафік, або перекласифікувати його. Не дивлячись на те, що в даному випадку теж використовується концепція буфера токенів, він не підтримує можливість затримки пакетів на відміну від механізму обмеження вихідного трафіку

Знищення (*dropping*) - механізм, що знищує дані. Наприклад, він

використовується при переповненні буфера даних обмежувача вихідного трафіку

Знищення (*dropping*) - механізм, що знищує дані. Наприклад, він використовується при переповненні буфера даних обмежувача вихідного трафіку.

Маркування (*marking*) - механізм зміни пакета. Зверніть увагу, що це не fwmark. Цілі *MARK* і - *mark* утиліт *iptables* і *ipchains* відповідно, модифікують метадані пакета, а не сам пакет.

2.2. Вирішення проблеми безпечного використання ресурсів в мережі Інтернет

Проблему безпечного і продуктивного використання Інтернет ресурсів можна вирішити двома способами.

Перший - радикальне заборона використання Інтернету без необхідності. Якщо прийняти принцип «заборонено все, що явно не дозволено», користувачам дозволяється доступ тільки до строго певних сайтів. Другий спосіб - більш гнучкий, він дозволяє користувачам діяти за принципом «дозволено все, що не заборонено». У цьому випадку співробітник може вільно користуватися ресурсами Інтернету, проте його дії перебувають під контролем. Це означає, що якщо користувач виконає дії, що суперечать політиці безпеки, це буде виявлено і попереджено.

В даний час «радикальний» спосіб досі знаходить застосування. Він використовується, в першу чергу, організаціями, в котрих циркулює інформація з грифом «секретної ». До таких організацій належать різні науково-дослідні інститути,військові організації, державні органи і спеціальні служби. У таких «секретних» організаціях існують інструкції та документи, які суворо регламентують поведінку користувачів, пов'язані з отриманням інформації і її передачею за межі організації. А це значно полегшує діяльність контролюючих служб щодо забезпечення потрібного рівня захисту.

Інший приклад «радикального» способу - застосування в компаніях так званих Інтернет-кіосків, коли користувачам надається доступ до Інтернет

ресурсів через виділені термінали. Як правило, в цьому випадку дії користувачів суворо регламентується, а трафік, що проходить через цей термінал, контролюється спеціальними засобами.

Більшість же комерційних організацій і компаній віддають перевагу більш гнучкий спосіб регламентації спілкування із зовнішнім світом. Далі розглядається саме цей спосіб, оскільки саме при його застосуванні виникають суттєві проблеми. І полягають вони тому, що практично неможливо однозначно визначити, до якої інформації слід забороняти доступ.

Щоб забезпечити гнучкий контроль користування Інтернет ресурсів, необхідно ввести в компанії відповідну політику використання ресурсів. Ця політика може реалізовуватися як вручну, так і автоматично. Ручна реалізація означає, що в організації є спеціальний штат співробітників, які в режимі реального часу або по журналах маршрутизаторів, проксі-серверів або міжмережевих екранів ведуть моніторинг активності користувачів. Такий моніторинг є проблематичним, оскільки вимагає великих трудовитрат. Крім того, він вимагає не тільки відстеження активності користувача, але і категоризації сайтів, які відвідують користувачі, а провести роботу по категоризації сайтів силами співробітників *IT* - підрозділів окремої компанії практично неможливо.

Щоб уникнути описаних вище проблем і забезпечити гнучкий контроль використанням Інтернет ресурсів, компанія повинна дати адміністратору інструмент для реалізації політики використання ресурсів компанії. Цій меті служить так звана контент фільтрація. Її суть полягає в декомпозиції об'єктів інформаційного обміну на компоненти, аналізу вмісту цих компонентів, визначення відповідності їх параметрів прийнятій в компанії політиці використання Інтернет ресурсів та здійсненні визначених дій за результатами такого аналізу. У разі фільтрації веб-трафіку під об'єктами інформаційного обміну мають на увазі веб-запити, вміст веб-сторінок, файли які передаються за запитом користувача і т.д.

Об'єктивна потреба викликала безліч програмних продуктів, передбачених для контролю вмісту інформаційного обміну. Всі вони в тій чи іншій ступені

виконують покладену на них завдання. Однак необхідно мати на увазі, що ніяка автоматична система не дає 100% гарантії безпеки без діяльної участі людини в процесі фільтрації. Будь-яка технологія - тільки додатковий інструмент у руках пекло адміністратора. І від того, наскільки система адекватна завданням, які ставить перед собою адміністратор, залежатиме, чи вдасться знизити рівень ризиків, пов'язаних з використанням Інтернету.

2.3. Засоби контролю використання ресурсів в мережі Інтернет

У цьому розділі коротко описані наявні на ринку засоби фільтрації веб-трафіку. Ці засоби можна умовно розділити за типами і методам фільтрації. В даний час відомо три типи засобів, здатних в тій чи іншій мірі забезпечити контроль використання Інтернет ресурсів на корпоративному рівні.

До першого типу відносяться міжмережеві екрани, проксі-сервери, маршрутизатори і подібні їм засоби фільтрації. Другий тип - це сучасні антивірусні програми, які володіють базовими можливостями тематичної фільтрації. До третього типу відносяться спеціалізовані засоби, розроблені безпосередньо для контролю використання Інтернет ресурсів.

Кожен тип засобів контролю використання Інтернет ресурсів призначений для фільтрації на різних рівнях мережевої ієрархії. Засоби першого і другого типів напряму не призначені для контролю вмісту інформаційного обміну по каналах Інтернету. Так, міжмережеві екрани, проксі-сервери і маршрутизатори здійснюють фільтрацію трафіку на мережевому і транспортному рівнях. Спеціалізовані засоби контентної фільтрації здійснюють її на прикладному рівні. Якщо говорити про антивірусні програми, то з засобами третього типу їх об'єднує саме здатність здійснювати деякі базові функції тематичної фільтрації.

Загалом, засоби першого і другого типів можна вважати достатньо ефективними для компаній, в яких контроль вмісту інформаційного обміну не є першочерговим завданням (наприклад, там, де конфіденційна інформація не циркулює в корпоративної мережі). В організаціях, де активно використовується Інтернет і при цьому актуальне завдання контролю доступу користувачів до Інтернет ресурсів і захисту від витоку конфіденційної інформації, застосування

таких засобів недостатньо. Це обумовлено наступними недоліками засобів першого і другого типів:

- обмежена кількість параметрів, за яким можлива фільтрація трафіку;
- обмежені можливості по фільтрації тексту в запитах користувачів і завантажених сторінках;
- неможливість декомпозиції об'єктів інформаційного обміну, отже, відсутність більш глибокої фільтрації (наприклад, тип файлів визначається за декларативним, а значить, не обов'язково реальному розширення, при цьому відсутні засоби для визначення реального типу файлів за сигнатурою);
- відсутність необхідної гнучкості при реалізації політики використання Інтернет ресурсів;
- відсутність функціональності, яка забезпечує контентну фільтрацію. Наприклад, не всі міжмережеві екрани і проксі-сервери підтримують контроль переданих даних на рівні команд протоколів.

Спеціалізовані програмні засоби контролю використання Інтернет ресурсів призначені для перевірки даних що передаються на відповідність тих чи інших умов інформаційного обміну і виконання відповідних дій за підсумками перевірки. Програмне забезпечення цього типу можна розділити за місцем використання на клієнтські і серверні. Клієнтське програмне забезпечення працює на кожній робочій станції, де потрібно забезпечити контроль використання Інтернет ресурсів.

Застосування клієнтського програмного забезпечення в організаціях неефективно, виключаючи окремі специфічні випадки. Клієнтське програмне забезпечення постійно працює на комп'ютері користувача і ненадійно з точки зору безпеки, оскільки потенційно його конфігурація і налаштування можуть бути спотворені (підроблені) досвідченим користувачем або спеціалізованим шкідливим кодом. Крім того, таке програмне забезпечення вимагає великих трудовитрат на налагодження і супровід, оскільки ці операції здійснюються для кожної робочої станції окремо.

В корпоративному секторі найбільш ефективне застосування програмного забезпечення, яке функціонує на виділеному сервері, оскільки воно розроблено спеціально для використання в корпоративних мережах і найбільш повноцінно відповідає вимогам по функціональності і безпеки. До таких вимог відносяться:

- централізоване розміщення (забезпечує зручність встановлення, налаштування та впровадження);
- розміщення на сервері і обмеження доступу до налаштувань ПО (забезпечує безпеку, оскільки виключена можливість зміни налаштувань користувачами).
-

2.4. Класифікація корпоративних засобів контролю використання ресурсів мережі Інтернет

Далі розглядаються серверні (а значить, корпоративні) засоби фільтрації веб-трафіка. Говорячи про їхню класифікацію необхідно виділити основні ознаки, за яким вони розрізняються:

- використовувані методи фільтрації;
- місце розміщення в інфраструктурі мережі та спосіб впливу на трафік (технології *pass by* і *pass through*);
- можливість вибору режиму функціонування - *standalone* (автономні) і *integrated* (вбудовуються);
- глибина політики фільтрації;
- підхід до оповіщення про дії користучів;
- можливість генерації різних видів звітів за даними фільтрації.

Коли говорять про класифікацію по методам фільтрації, класи визначаються набором параметрів, за якими проводиться перевірка вмісту інформаційного обміну. Системи використовують різні набори перевірок в залежності від покладених на засоби фільтрації завдань.

Найбільш типові й поширені системи, в яких основним способом фільтрації веб-трафіку є перевірка адрес Інтернет-ресурсів (*URL*). Звичайно, ці системи застосовують і інші способи (фільтрація по командам протоколів,

іменами користувачів, *IP*? Адресами робочих станцій і типами файлів), проте саме результат перевірки адреси сайту служить підставою для рішення про блокування сайту. Такі системи, як правило, фільтрують тільки запити користувачів, не перевіряючи файли сайтів на наявність забороненого політикою безпеки вмісту.

Існують системи, в яких реалізований комплексний підхід до фільтрації трафіку. В них перевірки рівноцінні за значимістю, а склад набору перевірок визначається завданнями, покладеними на систему контролю. Відмінністю таких систем є те, що вони мають найбільш широкий набір перевірок, серед яких однією з найважливіших є перевірка вмісту тексту запитів і сайтів.

Існуючі системи розрізняються за місцем розміщення в корпоративній мережі. Їх можна розділити на дві великі групи - працюючих як проксі-сервери (технологія *pass through*) і працюють як аналізатори пакетів (*packet sniffer*), що перехоплюють пакети потрібних протоколів і перевіряють їх на наявність забороненого вмісту (технологія *pass by*).

Системи, що використовують технологію *pass through*, працюють як звичайні *HTTP*-проксі. Як правило, їх встановлюють між клієнтськими місцями і зовнішнім проксі-сервером або брандмауером. Клієнтські місця повинні бути налаштовані так, щоб вони використовували потрібний проксі-сервер.

До переваг систем, що використовують технологію *pass by*, можна віднести їх прозорості для користувача і можливість встановлення без переналаштування клієнтських місць. Недоліком є те, що вони не завжди можуть справлятися з великим потоком даних, які передаються по контрольованих протоколах.

Говорячи про режими функціонування системи, розрізняють автономні *(standalone)* і вбудовані рішення *(integrated)*. Автономні системи, що встановлюються в розрив і працюють незалежно від інших підсистем, більш надійні з точки зору якості фільтрації. В них можливість помилок при фільтрації знижена до мінімуму за рахунок повного контролю над переданими даними. Зазвичай такі рішення використовуються спільно із зовнішніми проксі-серверами з метою збільшення швидкості доступу за рахунок кешування данних.

Деякі компанії поставляють програмне забезпечення для контролю

використання Інтернет ресурсів у вигляді вбудованих модулів до існуючих проксі-серверів або міжмережевих екранів. Досить часто модулі фільтрації створюються для широко використовуваних проксі-сервера *Microsoft ISA*.

Крім того, деякі системи фільтрації поставляються у вигляді окремих серверів, а доступ до них здійснюється по протоколу *ICAP*, який є стандартом для контролю та модифікації запитів і відповідей, що проходять через проксі-сервер, що підтримує даний протокол. Досить часто у вигляді *ICAP*-серверів реалізуються антивірусні комплекси *(Symantec Antivirus, Dr.Web, TrendMicro)*. До переваг продуктів, реалізованих у вигляді таких серверів, можна віднести те, що вони можуть взаємодіяти з різними типами проксі-серверів і міжмережевих екранів, які реалізують функцію клієнта *ICAP*, а також те, що сервери можуть виконувати перевірку тільки певних типів даних. Недоліком цього підходу слід вважати те, що при використанні протоколу *ICAP* сервер фільтрації може не мати повного контролю над переданими даними.

2.5. Перевірка адрес Інтернет ресурсів

Один з основних способів фільтрації веб-трафіку - перевірка адрес Інтернет ресурсів. Мова йде про фільтрування по *URL*, який передбачає собою повний шлях до конкретного документу або розділу на сервері або комп'ютері, підключеному до Інтернету. Використовуючи даний спосіб фільтрації, можна заборонити доступ як до всього сайту або домену, так і до його частини або окремій сторінці. Рекомендується вибирати засоби фільтрації, які можуть забезпечити блокування доступу і в тому випадку, коли користувач замість повного *URL* вказує тільки *IP* адресу сервера або комп'ютера в мережі.

Системи контролю використання Інтернет ресурсів, які застосовують даний спосіб фільтрації, використовують спеціальний сервіс по категоризації сайтів.

2.6. Керування каналом мережі Інтернет на основі ОС *Linux*

Що може запропонувати нам *Linux*, ось далеко неповний список з того, що може запропонувати нам операційна система *Linux*:

- Управляти пропускною здатністю на окремих комп'ютерах.
- Управляти пропускною здатністю до окремих комп'ютерів.
- Допоможе "роздати" пропускну здатність по-справедливості.
- Захистити вашу мережу від *DDoS*-атак.
- Запобігти нападу з вашої мережі на сервери в Інтернет.
- Розпаралелити кілька серверів, з метою рівномірного розподілу навантаження.
- Обмежити доступ до ваших комп'ютерів.
- Обмежити доступ ваших користувачів до інших вузлів мережі.
- Виконувати маршрутизацію на основі *UID, MAC*-адрес, що виходять *IP*-адрес, номерів портів, типу обслуговування, часу доби і вмісту.

На сьогоднішній день ці додаткові можливості не отримали широкого розповсюдження. На те є ряд причин: хоча наявна документація досить докладна, вона майже не містить практичних рекомендацій. А питання управління трафіком взагалі не освітлені.

Більшість дистрибутивів *Linux*, як і більшість ОС *UNIX*, нині використовують досить древні утиліти *arp, ifconfig* і *route*. Поки що ці інструменти працюють досить адекватно, але іноді, на ядрах *Linux* версії 2.2 і вище, вони можуть поводитися досить несподівано. Мережева підсистема, в ядрах 2.2 і вище, була повністю переписана. Новий мережевий код дав збільшення продуктивності і більш високі експлуатаційні характеристики, що робить *Linux* ще привабливішим на ринку операційних систем.

Фактично, реалізація мережевої підсистеми в *Linux*, що виконує класифікацію, маршрутизацію і фільтрацію, виявилася навіть повнішою, ніж в спеціалізованих маршрутизаторах, міжмережевих екранах і інших облаштуваннях управління трафіком.

У міру появи нових розробок, вони «нашаровувалися» поверх існуючих реалізацій в існуючих операційних системах. Це постійне нашарування привело до того, що код, вирішальний завдання управління мережевим трафіком, часом проявляв дуже дивну поведінку.

Наново переписана, реалізація мережевої підсистеми дозволила досягти таких характеристик, які раніше були просто недоступні.

Linux має досить складну систему управління пропускною спроможністю, названою *Traffic Control* (Управління Трафіком). Вона підтримує різні методи класифікації, ділення по пріоритетах, спільного використання, і обмеження трафіку, що як входить, так і витікаю чого.

Якщо наш маршрутизатор обслуговує складну мережу, то треба задовольняти потреби різних людей, обслуговування яких, ймовірно, повинне відрізнятися. База політик маршрутизації дозволяє реалізувати це за допомогою набору таблиць маршрутизації.

2.6.1. Політики маршрутизації в мережі Інтернет

Якщо ви хочете використати цю можливість, переконаєтеся що ядро зібране з підтримкою «*IP : advanced router*» і «*IP: policy routing*».

Коли ядру необхідно вибрати маршрут, воно визначає відповідно до якої таблиці це треба робити. По-умовчанню, визначені три таблиці. Стара утиліта *route* змінює таблиці *main* і *local*, як і утиліта *ip*.

Якщо ми хочемо зробити щось цікаве, то треба задати правила, що використовують різні таблиці маршрутизації. Це дозволить нам перевизначити загальносистемну таблицю маршрутизації.

2.6.2. Маршрутизація через декілька каналів/провайдерів

Перше питання полягає в тому, як організувати маршрутизацію так, щоб відповіді на запити, що приходять через певного провайдера, скажемо провайдера 1, йшли через того ж провайдера.

Друге питання полягає у балансуванні навантаження між двома провайдерами. Це не складно, якщо у нас вже налагоджений роздільний доступ, описаний в попередньому розділі.

Замість вибору одного з провайдерів в якості маршруту за замовчуванням, ви настроюєте т.з. *multipath* маршрут. У стандартному ядрі це забезпечить балансування навантаження між двома провайдерами. Результатом команди

буде попеременний вибір маршруту за замовчуванням. Ви можете змінити параметр *weight*, так щоб один з провайдерів отримував велике навантаження.

Зверніть увагу, що балансування не буде ідеальним, оскільки вона ґрунтується на маршрутах, а маршрути кешуются. Це означає, що маршрути до часто відвідуваних сайтів не проходитимуть через різних провайдерів.

2.6.3.Дисципліни обробки черг для управління пропускною здатністю комп'ютерної мережі

Організація черги визначає спосіб відсилання даних. Важливо розуміти, що ми можемо керувати лише швидкістю передачі даних, що відправляються. У тому вигляді, в якому зараз існує *Internet*, ми не можемо контролювати обсяг вхідного трафіку. Це щось на зразок поштової скриньки. Немає ніякого способу впливати на те, який обсяг пошти приходить до вас, хіба що спілкуючись з кожним респондентом.

Продовжуючи нашу аналогію, це можна порівняти з викиданням половини вашої пошти в надії на те, що люди перестануть вам писати. Різниця лише в тому, що у випадку з *Internet* цей прийом спрацьовує

Якщо у вас є маршрутизатор і ви хочете обмежити швидкість завантаження у внутрішній мережі, вам потрібно це робити на внутрішньому інтерфейсі маршрутизатора, з якого дані передаються ваших комп'ютерів.

Крім того, ви повинні бути впевнені, що контролюєте вузьке місце з'єднання. Так, якщо у вас є 100-мегабітний мережева карта і маршрутизатор із з'єднанням в 256 Кбіт / сек, ви повинні переконатися, що не посилаєте даних більше, ніж ваш маршрутизатор може передати. Інакше канал буде контролювати маршрутизатор і саме він буде обмежувати доступну пропускну здатність. Нам потрібно, так би мовити, «володіти чергою» і бути самим повільним ланкою. На щастя це легко реалізується.

Прості безкласові дисципліни обробки черги

Як вже говорилося, дисципліни обробки черги визначають спосіб передачі

даних. Безкласові дисципліни, загалом, отримують дані, змінювють порядок, вносять затримку або знищують їх.

Вони можуть використовуватися для обмеження пропускної здатності інтерфейсу цілком, без будь-якого поділу за класами. Вкрай важливо, щоб ви зрозуміли призначення цього типу черг перед тим, як ми перейдемо до класових дисциплін!

Найбільш поширеною дисципліною є *pfifo_fast* - вона використовується за замовчуванням. Кожна з дисциплін має свої переваги і недоліки. Не всі з них досконально протестовані.

pfifo_fast – ця дисципліна працює, як видно з назви, за принципом «першим прийшов, першим пішов» *(First In, First Out)*. Це означає, що жоден пакет не отримує спеціальної обробки. Однак це не зовсім так. Дана чергу має три, так званих, «смуги». У кожній «смузі» пакети обробляються за принципом *FIFO*. Але смуга 1 не буде обслуговуватися до тих пір, поки є пакети в смузі 0. Аналогічно, поки є пакети в смузі 1, не обробляється смуга 2.

Ядро враховує значення поля пакета *Type of Service*, і направляє пакети з встановленим прапором мінімальна затримка в смугу 0.

Ми не можемо конфігурувати *pfifo_fast*, оскільки її параметри жорстко «зашиті».

Token Bucket Filter (TBF) проста дисципліна черги, яка передає надходять пакети зі швидкістю не перевищує адміністративно заданий поріг, але з можливістю перевищують його коротких сплесків.

TBF дуже точна дисципліна, при цьому вона не створює серйозних навантажень на мережу і процесор. Якщо вам потрібно просто обмежити швидкість на інтерфейсі, то це перший кандидат на користування. Реалізована *TBF* у вигляді буфера, який постійно заповнюють токенами з заданою швидкістю. Найбільш важливим параметром буфера є його розмір, що визначає кількість збережених токенів.

Кожен прибуваючий токен співставляється з одним пакетом даних з черги після чого видаляється. Зв'язавши цей алгоритм з двома потоками - токенів і даних, одержимо три можливих ситуації:

- Дані прибувають зі швидкістю рівною швидкості входять токенів. У цьому випадку кожен пакет має відповідний токен і проходить чергу без затримки.
- Дані прибувають зі швидкістю меншою швидкості надходження токенів. У цьому випадку лише частина існуючих токенів буде знищуватися, тому вони стануть накопичуватися до розміру буфера. Далі, накопичені токени можуть використовуватися при сплесках, для передачі даних зі швидкістю яка перевищує швидкість надходження токенів.
- Дані прибувають швидше, ніж токени. Це означає, що в буфері скоро не останеться токенів, що змусить дисципліну призупинити передачу даних. Ця ситуація називається «перевищенням». Якщо пакети продовжують надходити, вони починають знищуватися.

Остання ситуація дуже важлива, оскільки дозволяє адміністративно обмежувати доступну смугу пропускання.

Накопичені токени дозволяють пропускати короткі сплески, але при тривалому перевищенні пакети будуть затримуватися, а в крайньому випадку - знищуватися. Врахуйте, що в реальній реалізації дисципліни, токени відповідають байтам, а не пакетам.

Не дивлячись на те, що нам ймовірно нічого не доведеться змінювати, дисципліна *TBF* має певні параметри. У першу чергу це:

- *Limit* - це кількість байт, які можуть бути поміщені в чергу очікування токенів. Цю ж величину можна задати параметром *latency*, який визначає максимальний «вік» пакета в черзі *TBF*. В останньому випадку, до уваги береться розмір буфера, швидкість і, якщо задана, пікова швидкість *(peakrate)*.
- Розмір буфера в байтах. Максимальна кількість байт, для яких токени можуть бути доступні миттєво. В цілому, чим більше гранична швидкість, тим більше повинен бути розмір буфера. Наприклад, для обмеження на швидкості 10 Мбіт / с на платформі *Intel*, вам потрібен буфер розміром як мінімум 10 Кбайт, щоб досягти заявленої швидкості!

- Якщо буфер занадто малий, пакети можуть знищуватися. Це пов'язано з тим, що кожен тік таймера буде генеруватися більше токенів, ніж може помістится у вашому буфері.
- Пакет нульового розміру все одно використовує смугу пропускання. У мережах *ethernet*, будь-який пакет має розмір не менше 64 байт. *MPU* задає мінімальну кількість токенів для пакета.
- Обмеження швидкості. Якщо буфер заповнений токенами, то вхідні пакети будуть проходити чергу без всяких затримок.
- Якщо на момент надходження пакету є вільні токени, пакет пройде чергу без будь-яких затримок. Так би мовити, зі швидкістю світла. Можливо, це не зовсім те, чого ви хочете, особливо якщо ви використовуєте великий буфер.
- Параметр *peakrate* задає швидкість, з якою елемент може проходити чергу. Згідно теорії, це досягається організацією достатньої затримки між проходять пакетами.
- Очевидно, що максимальне значення *peakrate*, рівне 1 Мбіт / сек, накладало б сильне обмеження на область застосування цієї дисципліни. Однак, завдання великих значень *peakrate* можливо. Досягається це за рахунок проходження за один інтервал часу більше одного пакета даних.
- За замовчуванням, значення *mtu* одно одному пакету, тобто за раз проходить тільки один пакет.

***Stochastic Fairness Queueing* (*SFQ*)** - проста реалізація сімейства алгоритмів справедливою очередизації. Вона не так точна, як інші дисципліни, але вимагає менше розрахунків, і при цьому порівну розподіляє доступну смугу пропускання між сеансами.

Ключовим поняттям в *SFQ* є потік, який приблизно відповідає сеансу або потоку *TCP/UDP*. Трафік поділяється на достатню кількість черг типу *FIFO*, по одній на кожен потік. Після цього, всі черги обробляються в циклічному порядку, тим самим забезпечуючи кожному сеансу рівні шанси на передачу даних.

Завдяки цьому досягається дуже рівна поведінка, яка не дозволяє будь-

якому діалогу пригнічувати інші. *SFQ* називається «стохастичною» т.к. насправді для кожного сеансу чергу не формується, а трафік ділиться на обмежену кількість черг на основі хеш-алгоритму.

Через використання хешу, кілька сесій можуть потрапити в одну і ту ж чергу, що зменшує шанси на передачу кожного сеансу. Для того, щоб ця проблема не відчувалася, *SFQ* часто змінює алгоритм хешування, тому, якщо сесії і потраплять в одну чергу, тривати це буде лише кілька секунд.

Варто зауважити, що *SFQ* ефективний тільки якщо вихідний інтерфейс повністю завантажений. В іншому випадку ніякого позитивного ефекту спостерігатися не буде.

Зокрема, застосування *SFQ* на *Ethernet*-інтерфейсі до якого підключений кабельний модем або *DSL*-маршрутизатор абсолютно безглуздий без обмеження смуги пропускання!

SFQ в значній мірі самоконфігуріруощаяся:

- *Perturb* Інтервал зміни алгоритму хешування. Хорошим значенням є 10 секунд.
- *Quantum* кількість байт виведених з черги за один раз. За-замовчуванням дорівнює 1 пакету максимально можливого розміру *(MTU)*.
- *limit* загальна кількість пакетів, які можуть бути поміщені в чергу *SFQ* (наступні пакети будуть знищуватися).

Класові дисципліни обробки черг

Класові дисципліни широко використовуються у випадках, коли той або інший вид трафіку необхідно обробляти по різному. Прикладом класової дисципліни може служити *CBQ - Class Based Queueing* (дисципліна обробки черг на основі класів). Вона настільки широко відома, що багато хто ідентифікує поняття «Дисципліна Обробки Черг» з назвою *CBQ*, проте це далеко не так.

CBQ - один із старих алгоритмів і крім того - один з найскладніших. На жаль він може далеко не все. Це може виявитися несподіванкою для тих, хто свято вірить в те, що якщо яка-небудь досить складна технологія поширюється без документації, то це краща технологія з наявних варіантів.

Коли трафік передається на обробку класовій дисципліні, він має бути віднесений до одного з класів(класифікований). Визначення приналежності пакету до того або іншого класу виконується фільтрами. Дуже важливо розуміти, що саме фільтри викликаються з дисципліни, а не навпаки!

Фільтри, приєднані до дисципліни, повертають результат класифікації (грубо кажучи клас пакету), після чого пакет передається в чергу, що відповідає заданому класу. Кожен з класів, у свою чергу, може складатися з підкласів і мати свій набір фільтрів, для виконання точнішої класифікації своєї долі трафіку.

Крім того, у більшості випадків класові дисципліни виконують шейпінг (формування) трафіку, з метою переупорядковування пакетів(наприклад, за допомогою *SFQ*) і управління швидкістю їх передачі. Це безперечно необхідно у разі перенаправлення трафіку з високошвидкісного інтерфейсу (наприклад, *ethernet*) на повільний(наприклад, модем).

Кожен з інтерфейсів має одну витікаючу кореневу дисципліну. За-замовчуванням це нагадує раніше дисципліну - *pfifo_fast*. Кожній дисципліні і кожному класу призначається унікальний дескриптор, який який може використовуватися подальшими інструкціями для посилання на ці дисципліни і класи. Окрім витікаючої дисципліни, інтерфейс так само може мати і дисципліну, яка проводить управління трафіком.

Дескриптори дисциплін складаються з двох частин - старшого і молодшого номерів, у виді: <старший>:<молодший>. Кореневій дисципліні загальноприйнято привласнювати дескриптор '1:', що еквівалентно запису '1: 0'. Молодший номер в дескрипторі будь-якої дисципліни завжди '0'.

Старші номери дескрипторів класів завжди дублюють старший номер дескриптора свого «батька».

Дисципліна *PRIO* фактично ніяк не обмежує трафік, вона лише виконує його класифікацію на основі приєднаних до неї фільтрів. Ви можете розглядати дисципліну *PRIO* як потужнішу версію *pfifo_fast*, в якій кожна із смуг є окремим класом, а не простою чергою *FIFO*. Постановка пакету в чергу виконується дисципліною *PRIO* на основі фільтрів, заданих вами. За-замовчуванню створюються три класи. Ці класи містять звичайні дисципліни *FIFO*, але вони

можуть бути замінені дисциплінами будь-якого типу, які вам тільки доступні. Коли необхідно витягнути пакет з черги, то першим перевіряється клас :1. Кожен подальший клас перевіряється тільки у тому випадку, якщо в попередньому немає жодного пакету.

Ця «розкидати» трафік по пріоритетах, ґрунтуючись не лише на прапорах *TOS*. Ви можете так само додати інші дисципліни до зумовлених класів, що підвищить можливості управління трафіком, в порівнянні з *pfifo_fast*. Формально, дисципліна *PRIO* відноситься до розряду планувальників типу *Work - Conserving*.

Дисципліна CBQ. Одна з найскладніших дисциплін. Це найоб'ємніша, найнезрозуміліша і найзаплутаніша дисципліна організації черг. Це не тому, що автори алгоритму некомпетентні, а тому, що ідеологія цього алгоритму абсолютно не співпадає з ідеологією *Linux*.

Крім того, що ця дисципліна є класовою, вона так само може виконувати і шейпінг трафіку, правда саме ця її сторона є найслабкішим місцем. Якщо ви спробуєте обмежити 10 мегабітовий канал величиною в 1 мегабіт, то виявиться, що з'єднання просто простоюватиме 90% усього часу. Замість визначення об'єму трафіку, *CBQ* вимірює час в мікросекундах між запитами і на основі отриманого часу розраховується середня завантаженість каналу. Такий алгоритм роботи не завжди дає потрібні результати. Наприклад, що якщо мережевий інтерфейс не може забезпечити повне завантаження каналу на усю його можливу ширину, із-за неякісного драйвера? Як тоді правильно визначити час простою?

Проблема стає ще гостріше, якщо вам доводиться мати справу з такими речами, як *PPP* через *Ethernet* або *PPTP* через *TCP/IP*. Ефективна пропускна спроможність в даному випадку може бути визначена як пропускна.

Обмеження пропускної спроможності в *CBQ* виконується за рахунок визначення проміжку часу між проходженням сусідніх пакетів середнього розміру. В процесі роботи вимірюється ефективний час простою, як експоненціальне зважене середнє по ковзаючому вікну. До речі, *UNIX* розраховує величину *loadaverage*(середня величина навантаження) аналогічним чином.

Розрахунковий час простою віднімається із зваженого середнього, в результаті виходить величина *avgidle*. Повністю завантажений канал має величину *avgidle* рівну нулю -- проміжок часу між пакетами точно співпадає з розрахунковим. У разі перевищення заданого обмеження, величина *avgidle* стає негативною. Якщо перевищення досягає деякого порогу, *CBQ* призупиняє передачу. З іншого боку, після декількох годин простою, величина *avgidle* може вийти занадто великою і це приведе до того, що канал «відкриється» на усю ширину. Щоб цього не відбувалося, величина *avgidle* обмежується числом *maxidle*.

Hierarchical Token Bucket *CBQ* занадто складна і слабо оптимізована для більшості типових ситуацій. Його підхід точніше відповідає конфігураціям, коли необхідно розподілити задану смугу пропускання між різними видами трафіку на смуги гарантованої ширини, з можливістю запозичення.

HTB працює точно так, як і *CBQ*, але, на відміну від останньої, принцип роботи заснований не на обчисленні часу простою, а на визначенні об'єму трафіку, що повністю відповідає назві *Token Bucket Filter*.

Хоча конфігурація *HTB* -- завдання досить складне, проте конфігурації добре масштабуються. У випадку ж з *CBQ* процес конфігурації стає занадто складним навіть в найпростіших випадках.

***Intermediate queueing device*.** Облаштування *IMQ* не є дисципліною обробки черги, але тісно з ними пов'язано. У *Linux*, дисципліни обробки черг приєднуються до мережевих пристроїв і все, що поміщається в чергу пристрою, потрапляє спочатку в чергу дисципліни обробки черги. Із-за цього підходу існують два обмеження:

1. Обмеження пропускної спроможності повноцінно працює тільки для витікаючого трафіку(дисципліна обробки трафіку, що входить, існує, але її можливості мізерні в порівнянні з повнокласовими дисциплінами).
2. Дисципліна обробки черги обслуговує трафік тільки для одного інтерфейсу, немає ніякої можливості задати глобальні обмеження.

Облаштування *IMQ* намагається вирішити ці проблеми. За допомогою підсистеми фільтрації ОС *Linux* можна певні пакети направляти через цей

псевдо-интерфейс, до якого підключаються різні дисципліни обробки черг. Таким чином, можна управляти смугою пропускання, трафіку, що як входить, так і загального.

2.6.4. *GRE* та інші тунелі

У ОС *Linux* підтримуються 3 типи тунелів. Це тунелювання *IP* в *IP*, *GRE* тунелювання і тунелі не-ядерного рівня(як, наприклад, *PPTP*).

Тунелі можуть використовуватися для дуже незвичайних і цікавих речей. Також вони можуть посилити ситуацію, якщо вони конфігуровані неправильно. Не задавайте маршрут за умовчанням через тунель, якщо тільки ви точно не упевнені в тому, що робите. Ще, тунелювання збільшує навантаження на систему і мережу, тому що додаються додаткові *IP*- заголовки. Зазвичай, це 20 байт на пакет. Таким чином, якщо звичайний розмір пакету *(MTU)* в мережі дорівнює 1500 байтам, то при пересилці по тунелю, пакет може містити тільки 1480 байт. Це не обов'язково стає проблемою, але пам'ятаєте про необхідність правильного налаштування фрагментації пакетів, якщо ви сполучаєте великі мережі.

IPSEC: безпечна передача даних протоколами *IP* через Інтернет

IPSEC є безпечною версією протоколу *IP*. Поняття «безпеки», в даному випадку, означає можливість шифрування і аутентифікації. У чистому вигляді, з технічної точки зору, «безпека» означає тільки шифрування, проте, досить легко показати, що цього недостатньо - ви можете обмінюватися шифрованими даними, але не мати при цьому гарантій, що видалена сторона саме та, яку ви чекаєте.

Шифрування, в *IPSEC*, виконується протоколом *ESP*(*Encapsulating Security Payload* -- Інкапсульовані Захищені Дані), аутентифікація -- протоколом *AH*(*Authentication Header* -- Заголовок Аутентифікації). Ви можете конфігурувати їх обох, або один з них.

І *ESP*, і *AH* спираються на *Security Association*(захищений віртуальний канал, або контекст безпеки). *Security Association* - однонапрямлене логічне з'єднання (від відправника до одержувача) між двома системами, що

підтримують протокол *IPSec*, яке однозначно ідентифікується наступними трьома параметрами:

- індексом захищеного з'єднання (*Security Parameter Index, SPI* - 32-бітова константа, використовувана для ідентифікації різних *SA* с однаковими *IP*-адресою одержувача і протоколом безпеки);
- *IP*- адресою одержувача *IP*- пакетів *(IP Destination Address)*;
- протоколом безпеки (*Security Protocol - AH* або *ESP*).

Щоб забезпечити необхідний рівень безпеки, ми повинні б передавати відомості про конфігурацію по надійних каналах. Якби нам довелося настроювати видалений хост через *telnet*, то будь-яка третя особа запросто могла б отримати секретні відомості, і така конфігурація буде далеко не безпечна. Крім того, як тільки секретна інформація стає відомою кому-небудь, вона перестає бути секретною. Знання секретних відомостей дасть не так багато видаленому користувачеві, але ми маємо бути абсолютно упевнені в тому, що канали зв'язку з нашими партнерами дійсно надійно захищені. Ця упевненість вимагає великої кількості ключів, якщо у нас є 10 партнерів, то необхідно мати не менше 50 різних ключів.

Окрім проблеми, пов'язаної з необхідністю узгодження ключів, існує також необхідність в періодичній їх зміні. Якщо третя сторона зможе перехопити наш трафік, то рано чи пізно вона буде в змозі «розколоти» ключ. Це може бути відвернене за рахунок періодичної зміни ключів, але цей процес вже вимагає автоматизації.

Інша проблема полягає в тому, що при роботі з ключовою інформацією «вручну», як це описано вище, ми заздалегідь точно визначаємо алгоритми і використовувану довжину ключа, що у свою чергу вимагає тісної координації з видаленою стороною. Бажано було б мати можливість визначення ширшої політики призначення ключів, наприклад так: «Ми можемо використати алгоритми *3DES* і *Blowfish*.

Рішення цих проблем бере на себе Протокол Обміну Ключами – *IKE (Internet Key Exchange)*, що дозволяє обмінюватися згенерованими, автоматично і випадковим чином, ключами. Передача ключів здійснюється за допомогою

асиметричної технології кодування, відповідно до зумовлених алгоритмів.

У *Linux IPSEC 2.5*, реалізація цих можливостей виконана у вигляді демона *KAME 'racoon' IKE*.

Однак, *Internet*, в більшості своєму, заснований на протоколі *TCP / IP*, а в нього є кілька властивостей, які можуть нам допомогти. *TCP / IP* не може дізнатися пропускної здатності мережі між двома хостами, тому він починає передавати дані все швидше і швидше (це називається «повільний старт»). Коли пакети починають губитися через перевантаження передавальної середовища, передача гальмується. Насправді все трохи складніше і розумніше, але про це пізніше.

2.7. Інші системи керування для ОС *Windows*

2.7.1. *Lan2net Traffic Shaper*

У *Lan2net Traffic Shaper* реалізоване унікальне для Windows систем динамічне управління завантаженням каналу, побудоване на принципах диференційованого обслуговування *(differentiated services)*. Подібний механізм застосовується в рішеннях на базі *Linux*. Принцип диференційованого обслуговування полягає в застосуванні різної якості обслуговування *(quality of services)* для задоволення різних потреб користувачів каналу. При диференційованому обслуговуванні мережевий трафік розділяється на класи, і до кожного класу застосовується індивідуальне нормування параметрів трафіку. Це дозволяє найгнучкіше управляти завантаженням мережевого каналу. Для різних груп протоколів, *IP* адрес і тому подібне можна встановлювати пріоритети, задавати максимальну і мінімальну ширину каналу. В результаті, це дозволяє добитися того, що найбільш важливі сервіси або користувачі отримають максимальну ширину каналу і швидкість їх роботи з Інтернет не впаде, коли трафік з нижчим пріоритетом займатиме канал. Меню програми показано на рис. 2.1.

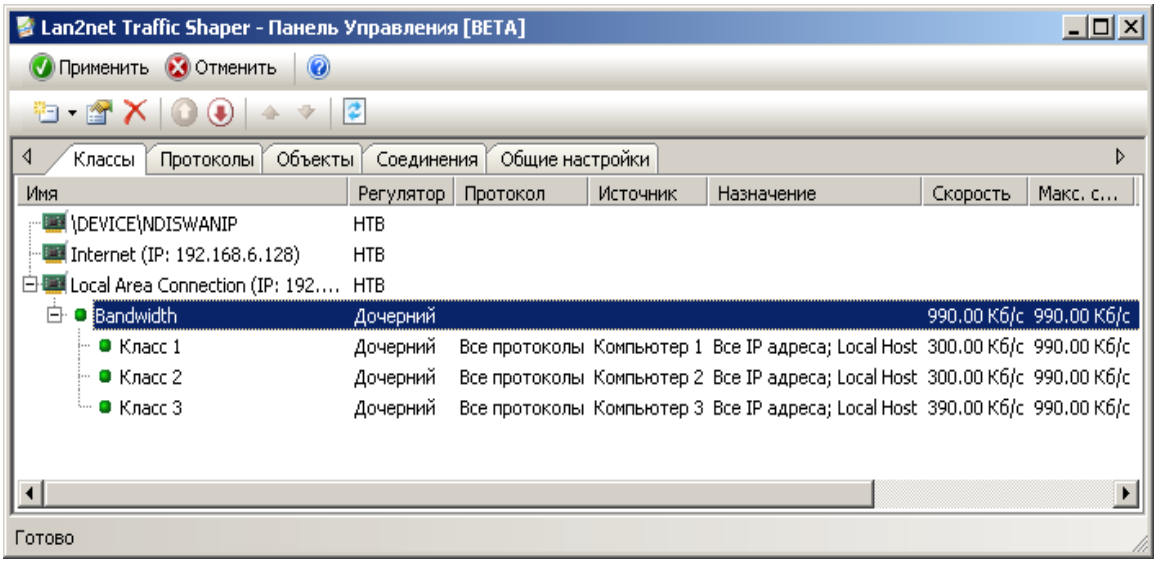

Рис. 2.1 Меню програми *Lan2net Traffic Shaper*

При диференційованому обслуговуванні мережевий трафік розділяється на класи, і до кожного класу застосовується індивідуальне нормування параметрів трафіку. Як показують дослідження, застосування подібної технології дозволяє поліпшити якість обслуговування користувачів навіть більшою мірою, чим придбання додаткової смуги пропускання у провайдера.

У загальному вигляді, якість обслуговування характеризується декількома, частенько взаємозв'язаними параметрами. Найбільш очевидний ключовий параметр тут, це - завантаження каналу, тобто, з якою швидкістю дані передаються по каналу. Для управління завантаженням каналу застосовуються різні схеми *(queueing disciplines)* або, більше простими словами, регулятори *(throttles).* Регулятори діляться на два типи: с розділенням трафіку на класи *(classful)* і без розділення *(classless)*. Регулятори без розділення трафіку на класи обмежують швидкість передачі усього трафіку того, що проходить через регулятор. Регулятори з розділенням трафіку на класи розділяють трафік, що проходить через них, на класи, і обмежують смугу пропускання для кожного класу індивідуально.

2.7.2. *UserGate Proxy & Firewall*

UserGate Proxy & Firewall — це комплексне рішення для організації загального доступу в Інтернет з локальної мережі, обліку трафіку і захисту корпоративної мережі від зовнішніх загроз. *UserGate* є ефективною

альтернативою дорогому програмному і апаратному забезпеченню і призначений для використання в компаніях малого і середнього бізнесу, вікно програми показано на рис. 2.2.

UserGate забезпечує комплексний захист локальної мережі, завдяки наявності двох вбудованих антивірусних модулів від провідних розробників антивірусних програм — Лабораторії Касперского і *Panda Security*. Антивірусні модулі проводять сканування усіх типів мережевого трафіку, включаючи поштовий, *HTTP* і *FTP* -трафік. На додаток до антивірусної перевірки в *UserGate* вбудований міжмережевий екран, що забезпечує надійний захист мережі від зовнішніх атак.

UserGate використовує комплексний підхід до забезпечення безпеки локальної мережі і сучасні методи боротьби з Інтернет-загрозами, такими, як віруси, шкідливі програми і хакерські атаки.

Функції інформаційної безпеки включають:

- Захист від вірусів
- Міжмережевий екран
- Розширений драйвер *NAT*
- Підтримка *VPN*- з'єднань

Захист від вірусів

Необхідність захисту локальної мережі від різних мережевих загроз, зокрема вірусів, Інтернет-черв'яків або троянов не можна недооцінювати, оскільки простий недогляд може мати непоправні наслідки для будь-якого бізнесу. Питання «Яке антивірусне застосування вибрати»? — що найчастіше задається на форумах, присвячених безпеці в Інтернет. Компанія *Entensys* співпрацює з двома світовими лідерами в області розробки антивірусного ПО — Лабораторією Касперского і *Panda Security* — з метою надати своїм користувачам вибір антивірусного рішення для використання у складі *UserGate*.

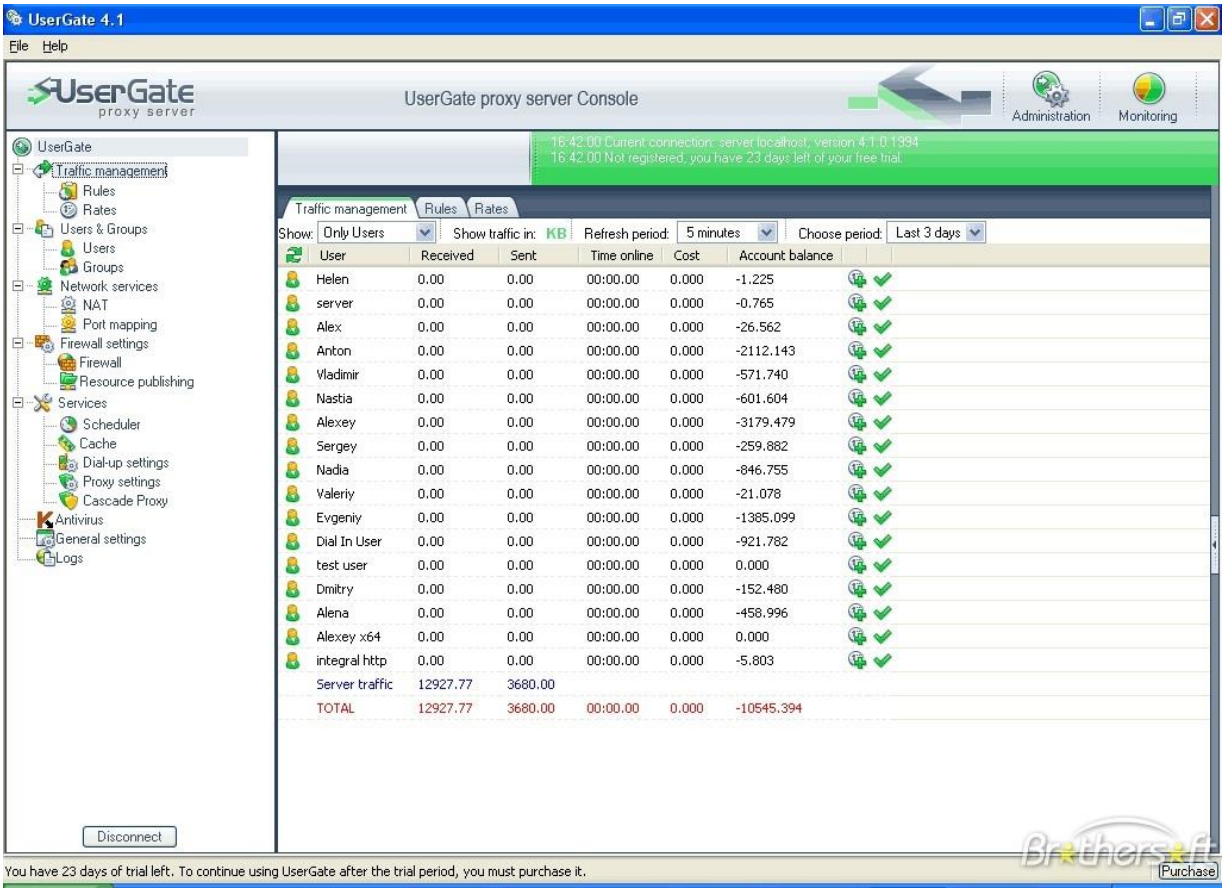

Рис. 2.2 Вікно програми *UserGate Proxy & Firewall*

Користувачі можуть за бажанням використати той або інший антивірусний модуль, або активувати обидва модулі для максимального захисту. При цьому можна комбінувати антивірусний захист *UserGate* із захистом файлової системи на локальних машинах за допомогою третього антивірусного рішення.

Міжмережевий екран

Міжмережевий екран (брандмауер) в *UserGate* дозволяє захистити локальну мережу від несанкціонованого доступу ззовні, одночасно надаючи можливість відкрити доступ до внутрішніх ресурсів, таким як поштовий, веб- або *VPN*- сервер в локальній мережі.

Розширений драйвер *NAT*

Версія *UserGate 5* містить новий, розширений варіант драйвера *NAT*. Функція маршрутизації тепер дозволяє адміністраторові створювати локальні підмережі і настроювати обмін пакетів між ними. Наявність підтримки протоколів *IP*- телефонії і публікація ресурсів дозволяють використати сучасні

способи комунікації і спільної роботи.

Підтримка *VPN*- з'єднань

UserGate підтримує передачу трафіку через протоколи *PPTP* і *L2TP* для з'єднання *VPN*- сервера з *VPN*- клієнтами локальної мережі. Крім того, можна використати публікацію мережевих ресурсів, щоб зробити *VPN*- сервер локальної мережі доступним видалено.

Контроль і статистика

За допомогою *UserGate* можна контролювати доступ в Інтернет окремих співробітників компанії і їх груп. Вбудований модуль *Entensys URL Filtering* дозволяє блокувати доступ до небажаних ресурсів як окремо, так і по категоріях сайтів. *UserGate* також дозволяє контролювати застосування, встановлені на клієнтських машинах, дозволяючи або забороняючи тому або іншому застосуванню вихід в Інтернет. Детальні статистичні звіти доступні як безпосередньо з програми, так і видалено за допомогою веб-браузеру.

UserGate дозволяє здійснювати повний контроль над використанням Інтернет-трафіку в компанії і надає адміністраторові детальну статистику. На основі даних статистики керівництво може визначати політику доступу в Інтернет в цій компанії, яка потім реалізується за допомогою гнучкої системи правил управління трафіком в *UserGate*.

Контроль доступу в Інтернет включає наступні функції:

- Користувачі і групи
- *URL*- фільтрація трафіку
- Контроль застосувань
- Швидкість і квотування
- Контроль трафіку і гнучка система звітів
- Модуль веб-статистики
- Білінгова система

Користувачі і групи

У основі роботи *UserGate* лежить поняття «користувач», яке визначається як комп'ютер або група комп'ютерів, об'єднаних загальною ознакою. В якості

такої ознаки може виступати *IP* - або *MAC*- адреса, пара «логін/пароль», обліковий запис в *Active Directory* або обліковий запис *Windows*. До усіх користувачів застосовуються правила розподілу трафіку, а також ведеться статистика і облік відвідування Інтернет-ресурсів.

Щоб спростити управління трафіком, адміністратор може об'єднати користувачів в групи за допомогою функції «Додати в групу». Інший спосіб угрупування користувачів полягає у використанні одного з декількох методів авторизації - наприклад, пари «логін/пароль». Адміністратор може вибрати будь-який спосіб, або обидва способи разом для ефективного управління безліччю користувачів, комп'ютерів або підмереж.

***URL*- фільтрація трафіку по категоріях сайтів**

Нецільове використання Інтернету на робочому місці є серйозною проблемою для працедавця і робить негативний вплив на продуктивність праці співробітників, безпеку локальної мережі, і збереження конфіденційних даних. Щоб уникнути потенційних загроз такого використання, невід'ємною частиною системи безпеки корпоративної мережі повинні стати механізми фільтрації відвідуваних веб-ресурсів.

Робота модуля фільтрації сайтів заснована на технології *Entensys URL Filtering,* яка також вилкористовується в *GateWall DNS Filter*. При цьому використовується категотризиційна база, в якій міститься 500 млн. сайтів в 82 категоріях. Адміністратор може забороняти доступ до окремих сайтів, до категорій сайтів, або до тих сайтів, адреси яких містять задані фрагменти слів. База спеціально адаптована для використання російськомовними користувачами і містить до 10 мільйонів російськомовних сайтів.

Контроль програм

Кількість програм, які так чи інакше використовують Інтернет-з'єднання для своєї роботи неухильно збільшується з кожним роком. Згідно з недавніми дослідженнями, застосування для миттєвого обміну повідомленнями(інтернет-пейджери) використовуються у більш ніж 80% організацій, і ця цифра постійно росте. Виникає необхідність контролю за діяльністю таких застосувань з метою захистити локальну мережу від зовнішніх загроз.

Контроль (фільтрація) програм — це технологія, що дозволяє обмежувати або блокувати трафік конкретних Інтернет- програм. Ця технологія має подвійне призначення: по-перше, вона дає можливість адміністраторові блокувати роботу певних Інтернет-застосувань (наприклад, *ICQ* або *MSN*), а по-друге — захищає локальну мережу від шкідливого ПО, яке може проникати в локальну мережу через такі застосування.

Обмеження трафіку і швидкості доступу

UserGate надає адміністраторові широкі можливості по контролю швидкості передачі даних між локальною мережею і Інтернетом. Встановити обмеження за швидкістю можна в модулях «Управління трафіком» і «Управління шириною каналу». Перший модуль призначений для налаштування обмежень швидкості по окремих користувачах і групах, тоді як другою дозволяє встановлювати обмеження швидкості для конкретного мережевого адаптера, протоколу (*TCP* або *UDP*), *IP*- адреси джерела або одержувача і порту.

Окрім швидкості, *UserGate* дозволяє також обмежувати об'єм трафіку і час перебування в мережі для користувачів і груп. При цьому, у розпорядженні адміністратора знаходиться широкий набір функцій, що дозволяє йому створювати правила, які задовольнятимуть будь-яким заданим вимогам. Наприклад, можна створити правило, яке стає активним при виконанні певних умов, таких як настання вказаного часу доби або використання певного протоколу.

Статистика відвідування Internet і система звітів

UserGate надає адміністраторові повну статистику про використання Інтернет в компанії по окремих користувачах і групах. Детальна статистика є основою для ухвалення рішень керівництвом про необхідність обмеження доступу до тих або інших ресурсів, або блокування роботи деяких Інтернет-застосувань.

Веб-статистика *UserGate*

Доступ до статистики *UserGate* можна отримати через Інтернет з будь-якої точки світу, використовуючи звичайний браузер. Інформація відображається не лише в табличному виді, але і у вигляді діаграм і графіків, що істотно полегшує сприйняття звітів. Об'єм доступної статистичної інформації залежить від рівня доступу користувача.

Білінгова система

Вбудована в *UserGate* білінгова система автоматично проводить розрахунок вартості роботи користувача в мережі Інтернет виходячи із заданої ціни, часу і/або об'єму трафіку. Можна встановлювати тарифи як для окремого користувача, так і для групи користувачів. Існує можливість перемикання тарифів залежно від часу доби, дня тижня або адреси сайту.

Організація доступу в Інтернет

За допомогою *UserGate* можете організувати доступ в Інтернет для співробітників вашої компанії через *NAT* або проксі-сервер, одночасно працювати через декілька Інтернет-провайдерів, а також оптимізувати споживання Інтернет-трафіку з тим, щоб уникнути навантаження на мережу і понизити витрати на трафік. Підтримка протоколів *IP*- телефонії дозволяє скористатися перевагами *VoIP*- рішень, щоб на їх основі створити сучасну комунікаційну інфраструктуру компанії.

Доступ в Інтернет

Забезпечення доступу в Інтернет і контроль трафіку є основним завданням придбання і установки проксі-сервера в компанії. За допомогою *UserGate* можна організувати доступ користувачів локальної мережі в мережу Інтернет як через *NAT*, так і через *HTTP-, FTP* - і інші типи проксі-серверів. Гнучкість і різноманіття функцій *UserGate* дозволяють адміністраторові мережі настроїти сервер так, щоб він відповідав найсерйознішим вимогам безпеки і продуктивності.

- Забезпечення доступу в Інтернет
- Проксі-сервери для різних протоколів
- Робота з декількома провайдерами
- Управління шириною каналу

- Кешування
- Підтримка *IP*- телефонії

Забезпечення доступу в Інтернет

UserGate дозволяє організувати доступ в Інтернет комп'ютерів у вашій локальній мережі, використовуючи будь-який тип Інтернет-підключення, такий як *DSL, ISDN*, кабельне підключення, комутований доступ або *WiFi*. *UserGate* служить проміжною ланкою між Інтернет і локальною мережею, і має як зовнішню *IP*- адресу для роботи в Інтернет, так і один або декілька локальних *IP*-адрес для роботи з комп'ютерами в локальній мережі.

Оскільки при використанні *UserGate* увесь Інтернет-трафік проходить тільки через сервер *UserGate*, такі завдання як управління трафіком, перегляд статистики закачувань і захист локальної мережі від зовнішніх загроз здійснюються централізований.

Проксі-сервери для різних протоколів

UserGate може служити проксі-сервером між локальною мережею і Інтернет. Функція проксі доступна таких протоколів, як *HTTP* (с підтримкою *HTTPs* і «*FTP* через *HTTP*»), *FTP, SOCKS, POP3, SMTP, SIP*, і *H.323*. При цьому, підтримується функція «прозорий проксі» при використанні якої немає необхідності в ручному налаштуванні браузерів користувачів. Крім того, ви можете вказати конкретний мережевий інтерфейс, на якому працюватиме проксі-сервер.

Використання *HTTP*- проксі може служити різним цілям, таким як прискорення відповідей на запити користувачів (кешування), і пов'язана з цим економія трафіку. Метою може бути також необхідність фільтрації Інтернет-трафіку і заборона доступу до деяких ресурсів.

Робота з декількома провайдерами

У *UserGate* підтримується одночасна робота з декількома Інтернет-провайдерами. Ця функція дозволяє надати доступ в Інтернет різним користувачам через різні провайдери, а також автоматично перемикати

користувачів на резервне з'єднання у разі, якщо з'єднання з основним Інтернет-провайдером не працює.

Управління шириною каналу

Із зростанням числа Інтернет-з програм, і, отже, об'єму трафіку виникає необхідність оптимізувати споживання трафіку. Управління шириною каналу в UserGate є функцією, покликаною вирішити це завдання.

Кешировання

Кешировання є однією з функцій *HTTP* - і *FTP*--проксі в *UserGate*, яка прискорює відкриття веб-сторінок або завантаження файлів, а також дозволяє істотно економити витрати на трафік, що входить. При використанні кешировання результати веб-запитів користувачів зберігаються на локальний диск комп'ютера, на якому встановлений *UserGate*.

Підтримка *IP*- телефонії

Підтримка протоколів *SIP* і *H.323* дозволяє використати проксі-сервер *UserGate* в якості *VoIP*- шлюзу як для програмних, так і для апаратних *IP*-телефонів. При використанні *SIP* проксі-сервера в моніторингу *UserGate* буде відображена уся інформація про поточний стан з'єднання(реєстрація, дзвінок, очікування та ін.), а також ім'я користувача, телефонний номер того, що дзвонить, час розмови і кількість переданих і отриманих байт. Ця ж інформація буде записана і у базу статистики *UserGate*.

Адміністрування мережі

UserGate включає *DHCP*- сервер для динамічного призначення *IP*- адрес в локальній мережі і функцію публікації ресурсів, яка дає можливість отримати доступ ззовні до ресурсів компанії усередині локальної мережі. Функція маршрутизації дозволяє передавати дані між двома локальними підмережами. І нарешті, сервер *UserGate* можна адмініструвати видалено, підключаючись до нього з будь-якої точки світу.

Мережеве адміністрування

За допомогою *UserGate* можна виконувати деякі рутинні операції, що дозволяє спростити мережеве адміністрування. Наприклад, вбудований *DHCP*-сервер автоматизує процес видачі *IP*- адрес комп'ютерам і іншим пристроям в

локальній мережі. Якщо комп'ютер з *UserGate* підключений до декількох локальних мереж, сервер *UserGate* можна настроїти як маршрутизатор (*router*), забезпечивши прозорий, двонаправлений зв'язок між локальними мережами. Публікація ресурсів дозволяє надати доступ до внутрішніх ресурсів компанії, наприклад до *Web, FTP, VPN* або до поштового сервера. І, нарешті, видалене адміністрування дає можливість видалено підключатися по локальній мережі або через Інтернет з будь-якого комп'ютера, на якому встановлена Консоль Адміністрування *UserGate*.

- *DHCP*- сервер
- Маршрутизація
- Публікація ресурсів
- Видалене адміністрування

DHCP- сервер

DHCP (англ. *Dynamic Host Configuration Protocol* — протокол динамічної конфігурації вузла) — це мережевий протокол, що дозволяє автоматично отримувати *IP*— адреси пристроям в локальній мережі у момент їх підключення, тим самим позбавляючи системного адміністратора від ручного налаштування.

Маршрутизація

Якщо комп'ютер з *UserGate* підключений до декількох локальних мереж, сервер *UserGate* можна настроїти як маршрутизатор *(router),* забезпечивши прозорий, двонаправлений зв'язок між локальними мережами.

Публікація ресурсів

Часто виникає необхідність надання доступу до внутрішніх ресурсів компанії ззовні. Як правило, до таких ресурсів відносяться *Web-, FTP-, VPN* - або поштовий сервер. Для того, щоб надати такий доступ за допомогою UserGate необхідно створити правило перенаправлення запитів на комп'ютер в локальній мережі, на якому запущена відповідна служба.

Видалене адміністрування

До сервера *UserGate* можна підключатися по локальній мережі або видалено через Інтернет з будь-якої точки світу. Для цього досить встановити на комп'ютер Консоль Адміністрування *UserGate*, і вказати в налаштуваннях з'єднання *IP*- адресу і порт сервера *UserGate*.

Можливість видаленого адміністрування сервера *UserGate* особливо корисна у разі, коли необхідно адмініструвати декілька серверів *UserGate* в різних місцях(наприклад, декількох Інтернет-кафе). При цьому, адміністрування здійснюється з однієї і тієї ж консолі — усі доступні сервера відображаються в списку «З'єднання», і можна видалено підключитися до будь-якого з них.

2.7.3. *Traffic Inspector*

Сертифіковане комплексне рішення для організації і контролю доступу в Інтернет. Не вимагає дорогого мережевого устаткування, забезпечує гнучку тарифікацію, надійний мережевий захист, розподіл завантаження, точний облік і статистику, економію трафіку і робочого часу. Вигляд програми показано на рис. 2.3.

Traffic Inspector спеціально створений для того, щоб об'єднати і доповнити усе різноманіття мережевих можливостей операційних систем *Microsoft Windows*, тому вам не доведеться робити спеціальних налаштувань. Все, що працювало раніше, працюватиме і після установки програми - знадобиться тільки авторизація користувачів.

Організація доступу в Інтернет. Контроль трафіку

Користувачі(у програмі вони називаються клієнти) можуть працювати як безпосередньо, через *NAT*, так і через проксі-сервер. Для кожного користувача створюється окремий обліковий запис(чи підвантажується з *Active Directory*), і усі його дії в мережі відображаються у вигляді простих і зрозумілих звітів.

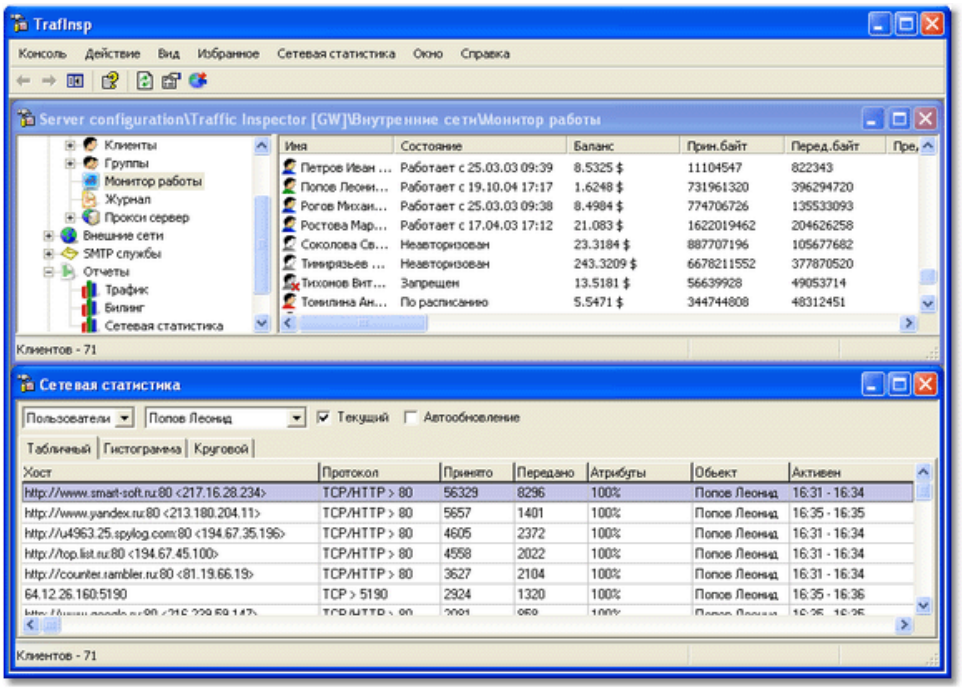

Рис. 2.3 Вигляд вікна програми *Traffic Inspector*

Облік трафіку. Система білінга (*billing*)

Підрахунок трафіку в програмі відбувається по кожному користувачеві з точністю до байта, причому ви самі визначаєте одиницю обліку, ліміти, кредити, блокування, фільтри і розклади. Система биллинга *Traffic Inspector* має сертифікат відповідності зв'язку, що гарантує виняткову точність розрахунків.

Проксі-сервер і економія

Використання проксі-сервера *Traffic Inspector* дозволяє кешувати часто використовувані інтернет-ресурси, а також блокувати банери, рекламні вставки, графіку, музику або відео і забороняти небажані сайти або їх розділи.

Безпека і захист мережі. *Firewall*

Захист мережі організований двома рівнями: мережевий екран забезпечує захист від зовнішніх атак, а система блокування і сповіщення при надмірній мережевій активності служить для внутрішнього контролю безпеки.

Фільтрація спаму

При використанні поштового шлюзу *Traffic Inspector* є можливість

застосувати систему блокування спаму на внутрішньому поштовому сервері.

Антивірусний захист

Окрім функції своєчасного виявлення зараження мережевими вірусами, для перевірки трафіку на проксі-сервері і поштовому шлюзі *Traffic Inspector* передбачені додаткові модулі антивірусного захисту.

Управління швидкістю і маршрутизацією

Traffic Inspector дозволяє задавати обмеження швидкості для користувачів або груп з динамічним розподілом навантаження, а система управління маршрутизацією *Advanced Routing* дає можливість направити трафік на різні канали доступу, у тому числі на супутник.

Видалений контроль і статистика

У програмі є ряд засобів для видаленого управління і моніторингу системи. Використовуючи *Traffic Inspector*, ви завжди будете в курсі справи про стан мережі, де б ви не знаходилися.

Traffic Inspector може бути використаний як в організаціях для безпечного і ефективного використання інтернет-підключення, так і в невеликих підприємствах, що роблять послуги з передачі даних : провайдерів, інтернет-кафе, готелях і хот-спотах. Щоб переконатися, що *Traffic Inspector* вам підходить, ми рекомендуємо безкоштовно перевірити в роботі його повнофункціональну версію.

2.8. Висновки до розділу

У цьому розділі ми ознайомились з технологіями і видами керування Інтернет каналом. Побачили що для кожної платформи існує немало засобів для управління Інтернет каналом. Видно що для операційних систем *Windows* немає безкоштовних та функціональних систем, всі системи розглянуть вище є комерційними. Тому в світі більш популярні рішення на базі операційних систем *Linux*. Таки системи є функціональними та безкоштовними в силу іншої схеми

заробітку їх авторів. Тому ми більш детально розглянули віртуальні технології які основані можливостях операційних систем *Linux*.

Але всі програмні засоби все-таки вимагають інвестувати в комп’ютерне обладнання та обслуговування операційних систем та систем керуванні трафіком. Тому ми маємо звернути увагу на програмно-апаратні рішення для організації управління Інтернет каналом.

РОЗДІЛ 3 РОЗРОБКА МОДЕРНІЗОВАНОЇ МОДЕЛІ МЕРЕЖІ ОФІСУ З ВПРОВАДЖЕНОЮ СИСТЕМОЮ КЕРУВАННЯ ІНТЕРНЕТ-КАНАЛОМ

3.1. Вибір засобів для реалізації системи керування Інтернет-каналом в комп'ютерній мережі організації

Серед засобів для реалізації даної системи найбільш популярні окремі сервери. На таких серверах зазвичай стоять повноцінні операційні системи, такі як *Linux* або *Windows*. На такі сервери приходять Інтернет-канали від провайдерів, далі сервери займаються розподіленням трафіку між користувачами та різними Інтернет-каналами.

Але такі системи є невідними з точки зору доцільності їх використання для організацій в яких працює порівняно невелика кількість робітників. Системи побудовані на повноцінних серверах потребують постійного обслуговування адміністраторами. Зазвичай повноцінні сервери мають велику надлишкову функціональність яка просто не використовуються, що викликає простій обладнання. Якщо вибирати обладнання для серверів, то системи побудовані на базі комплектуючих для клієнтських комп'ютерів не можуть гарантувати довговічність своєї роботи в режимі постійного включення, а спеціалізовані сервери можуть надто дорого коштувати. До всього, вартість серверної версій *Windows* теж викликає потребу повного обґрунтування її потреби для компанії. В такому випадку адміністратори організацій звертають увагу не на програмні а апартно-програмні рішення.

На допомогу приходять апаратні рішення *SOHO* (від *Small Office / Home Office* — малий/домашній офіс). Одним з таких рішень є апаратно програмна платформа *Mikrotik RouterBOARD. RouterBOARD* — апаратна платформа від

MikroTik, що є лінійкою маршрутизаторів під управлінням операційної системи *RouterOS*. Різні варіанти *RouterBOARD* платформ дозволяють вирішувати на їх основі різні варіанти мережевих завдань: від простої безпровідної точки доступу і керованого комутатора до потужного маршрутизатора з брандмауером і *QOS*.

Практично усі моделі *RouterBOARD* пристроїв живляться за допомогою *PoE* і мають, при цьому, роз'їм для підключення зовнішнього джерела живлення.

Моделі призначені для роботи з безпровідними технологіями мають слот(*miniPCI*) для підключення радіомодулів. Більшість моделей так само мають роз'єм для підключення до *COM*- порту ПК. У бюджетних моделях або залежно від конкретного призначення моделі, ті або інші елементи можуть бути відсутніми.

3.2. Операційна система *RouterOS*

Mikrotik RouterOS - це спеціалізована операційна система на ядрі *Linux*, призначена для побудови потужних багатофункціональних маршрутизаторів, файрволів, мостів, *vpn*- серверів, станцій і мостів *WI-FI*, хотспотов, управління якістю обслуговування і тому подібне на базі звичайних *PC*- машин з процесорами *x86* або власних апаратних маршрутизаторів *Mikrotik RouterBoard* на процесорах *PowerPC* і *Atheros*. На відміну від безлічі вільних відкритих проектів - це закрита система, що не дозволяє установку додаткового програмного забезпечення.

Перевагами *MikroTik* є можливість за невеликі гроші швидко розгорнути рішення професійного класу, з хорошою продуктивністю, великим функціоналом і великою зручністю роботи з системою, що поєднує переваги функціонального консольного інтерфейсу (ідеологія, аналогічна *Cisco*) і графічної утиліти для *Windows (Winbox)*.

MikroTik RouterOS вважається однією з кращих програмних реалізацій маршрутизатора 3-го рівня для *PC* і використовується у всьому світі. У *RouterOS* реалізовані багато функцій, існуючим тільки в ОС для дорогих маршрутизаторів

RouterOS підтримує багатоядерні та багатопроцесорні системи. Підтримує встановлення на *IDE, SATA* та *USB* пристрої збереження даних. Для встановлення операційної системи потрібно тільки 64 Мбайти вільного простору на диску. Також підтримує різноманітні мережеві інтерфейси, аключаючи 10Гбітні мережеві карти, 802.11 *a/b/g/n* бездротові карти та *3G* модеми.

RouterOS підтримує декілька методів налаштування - локально, послідовна консоль, *Telnet, SSH, GUI* застосування *Winbox*(тільки для *OS Windows*), простий аеб-інтерфейс, а так само *API* інтерфейс для створення власних застосувань. У разі відсутності доступу на рівні *IP RouterOS* підтримує доступ на *MAC* рівні, через *MAC Telnet* або *Winbox*. *RouterOS* має потужний, але в теж час легкий в освоєнні командний інтерфейс з підтримкою скриптів. З версії *RouterOS v4* з'являється підтримка скриптової мови *Lua*, яка розширює можливості по автоматизації і програмуванню маршрутизатора.

Firewall(брандмауер) здійснює контроль і фільтрацію мережевих пакетів, що проходять через нього, відповідно до заданих правил. *Network Address Translation(NAT)* дозволяє запобігти або обмежити звернення зовні до внутрішніх хостів, залишаючи можливість звернення зсередини назовні. *Firewall* надає можливість маркування пакетів для подальшої їх маршрутизації. Є можливість задати фільтрацію по *IP*- адресі, діапазону адрес, портам, діапазону портів, *IP* протоколу, *DSCP* і іншим параметрам, а також підтримуються статичні і динамічні списки адрес. *Layer7* фільтр дозволяє фільтрувати пакети на основі даних рівня програм. Іншими словами, він дозволяє розпізнавати пакети *HTTP, FTP, Gnucleus, Kazaa* і так далі, незалежно від порту. *Firewall RouterOS* підтримує *IPv6*.

RouterOS підтримує статичну маршрутизацію і безліч динамічних протоколів маршрутизації:

- Для *IPv4* це *RIP v1* і *v2, OSPF v2, BGP v4*.
- Для *IPv6* це *RIPng, OSPFv3* та *BGP*.

RouterOS також підтримує віртуальну маршрутизацію і пере направлення *(VRF)*. Можемо використати фільтр *Firewall*, щоб відмітити певні підключення

маркером маршрутизації, і потім відмічений трафік перенаправити на певного провайдера. *RouterOS* додана підтримка *MPLS* і *VRF*. *VRF*(*Virtual Routing and Forwarding*) - технологія дозволяє співіснувати декільком таблицям маршрутизації в межах одного маршрутизатора. Завдяки подібній віртуалізації можна зробити так, щоб різні клієнти користувалися одним і тим же адресним простором або різними протоколами маршрутизації. *MPLS*(*Multiprotocol Label Switching*) - механізм передачі даних, який емулює різні властивості мереж з комутацією каналів поверх мереж з комутацією пакетів. Він був розроблений з метою забезпечення універсальної служби передачі даних як для клієнтів мереж з комутацією каналів, так і мереж з комутацією пакетів. За допомогою *MPLS* можна передавати трафік самої різної природи, такий як *IP*- пакети, *ATM, SONET* і кадри *Ethernet*. У традиційній *IP* мережі пакети передаються від одного маршрутизатора іншому і кожен маршрутизатор читаючи заголовок пакету(адреса призначення) приймає рішення про те, по якому маршруту відправити пакет далі. У протоколі *MPLS* ніякого подальшого аналізу заголовків в маршрутизаторах шляхом дотримання не проводиться, а переадресація управляється виключно на основі міток.

VPN(*Virtual Private Network*) логічна мережа, що створюється поверх іншої мережі, наприклад Інтернет. Попри те, що комунікації здійснюються по публічних мережах з використанням небезпечних протоколів, за рахунок шифрування створюються закриті від сторонніх канали обміну інформацією. *VPN* дозволяє об'єднати, наприклад, декілька офісів організації в єдину мережу з використанням для зв'язку між ними непідконтрольних каналів. *RouterOS* підтримує різні методи *VPN* і тунельних протоколів: *Ipsec, OpenVPN, PPTP, PPPoE, L2TP, MLPPP, BCP, IPIP, EoIP, IPv6 over IPv4, VLAN - IEEE802.1q Virtual LAN support, Q - in - Q support, MPLS based VPNs*

RouterOS підтримуваний безліч безпровідних технологій, основна з них це точка доступу + клієнт. Основні підтримувані технології:

- *IEEE802.11a/b/g/n*
- *Nstreme і Nstreme2*

- *Client polling*
- *RTS/CTS*
- *Wireless Distribution System(WDS)*
- *Virtual AP*
- *WEP, WPA, WPA2*
- *Access control list*
- *Wireless client roaming*
- *WMM*
- *HWMP+ Wireless MESH protocol*
- *MME wireless routing protocol*

Шлюз *Hotspot* дозволяє надавати доступ до мережі Інтернет в громадських місцях використовуючи безпровідну мережу для пристроїв, що використовують технологію *Wi - Fi. RouterOS MikroTik* дозволяє створювати профілі для різних користувачів, обмежуючи за швидкістю доступу, об'єму трафіку, часу роботи. Основні особливості:

- *Plug - n - Play* доступ до мережі
- Локальна аутентифікація
- Управління користувачами
- Підтримка *RADIUS* для аутентифікації і управління користувачами
- Створення конфігурації для не інтерактивних пристроїв

Управління пропускною спроможністю це ряд механізмів по контролю за розподілом навантаження, затримками, своєчасністю доставки даних і надійністю. Якість обслуговування *(QoS)*, означає, що маршрутизатор може розставляти пріоритети мережевого трафіку. Основні можливості:

- обмежувати швидкість передачі даних для певних *IP* адрес, підмережі, протоколів, портів
- обмеження однорангової рухи
- пріоритетність деяких пакетною в порівнянні з іншими
- використання черги для прискорення веб-перегляду
- застосування обмежень на фіксовані тимчасові інтервали

- розподіл швидкості доступу залежно від навантаження на канал

RouterOS MikroTik дозволяє організувати кэшируючий проксі сервер для прискорення «браузинга» клієнтів оскільки об'єкти, що потрапили в кеш передаються по локальній мережі з набагато більшою швидкістю. Основні можливості:

- *HTTP* проксі сервер.
- прозорий проксі сервер.
- списки доступу по джерелу, призначенню, *URL*, запитам *(HTTP firewall).*
- завдання об'єктів, що підлягають кешуванню і об'єктів – виключень.
- використання багаторівневих проксі серверів.
- логування.
- підтримка *SOCKS proxy.*
- зберігання кеша на зовнішніх пристроях.

Для допомоги в адмініструванні в *RouterOS* були додані невеликі інструменти, такі як:

- *Ping, traceroute*
- *Bandwidth test, ping flood*
- *Packet sniffer, torch*
- *Telnet, SSH*
- *E - mail and SMS send tools*
- *Automated script execution tools*
- Вибірка вмісту з файлів
- Таблиця активних з'єднань
- *NTP Client and Server*
- *TFTP server*
- Підтримка динамічного *DNS*
- *SNMP* для побудови графіків і збору статистики
- *RADIUS* client and server

3.3. Платформа *RB2011UiAS-2HnD-IN* для *RouterOS*

Mikrotik RB2011UiAS-2HnD-IN (рис. 3.1) складається з 10-ти незалежних гігабитних мережевих портів, та одного порту SFP з можливістю перемкнути їх частину в режим апаратної обробки - наче ці порти підключені до звичайного світчу. Апаратну конфігурацію наведено в таблиці 1.

Рис. 3.1 Зовнішній вигляд маршрутизатора *RB2011UiAS-2HnD-IN*

Таблиця 3.1

Конфігурація маршрутизатора

Процесор	*Atheros AR9344 600MHz CPU*
Пам’ять	*128MB DDR SDRAM onboard memory*
Завантажувач	*RouterBOOT*
Збереження даних	*128MB onboard NAND storage chip*
Ethernet	11 незалежних 10/100/1000 *ethernet* портів
miniPCI	Не доступно
Доповнення	Кнопка перезавантаження
Серійний порт	Відсутній
Індикатори	*Power, NAND activity, 11 Ethernet LEDs*
Живлення	*Power over Ethernet: 9-28V DC on Ethernet port 1 (Only on pins 4,5,7,8. Passive PoE. Non 802.3af). Jack: 9-28V DC*

Розміри	25x230x90мм. Вага без упаковки та кабелів: 635g
Споживча потужність	До 6*W*
Робоча температура	-20C .. +50C
Операційна система	*MikroTik RouterOS v7*

3.4. Налаштування роутера

Найбільш популярними засобами налаштування роутерів є командна строка через термінальні програми та за допомогою спеціальної утиліти *Winbox*. При чому ця утиліта є найбільш зручною навіть для введення параметрів через консоль, так як в інтерфейсі утиліти доступна вбудована консоль.

Для заходу до основного потрібно запустити утиліту та ввести потрібні дані в вікно аутентифікації адміністратора, показано на рис. 3.2.

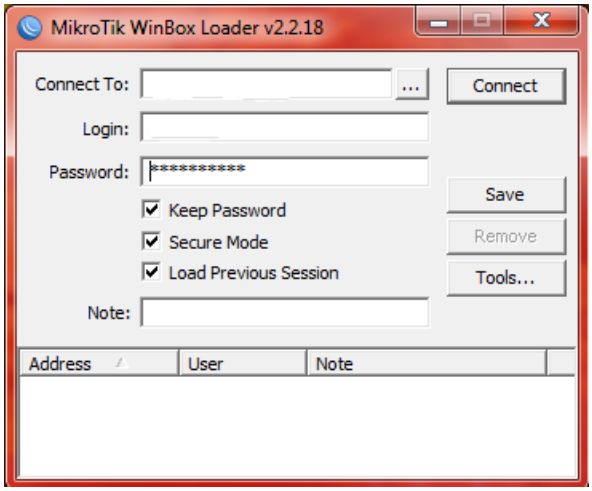

Рис. 3.2 Вікно аутентифікації адміністратора

В поле *Connect To* вводимо *IP* адресу або *MAC* адресу нашого роутера. Якщо розтер знаходиться у внутрішній мережі компанії, то можливо скористатися автопошуком пристрою натиснувши на кнопку з піктограмою «…» правіше від поля адреси пристрою. За замовчуванням *IP* адреса роутера є 192.168.88.1. Щоб мати можливість зайти через *IP* адресу, потрібно на комп'ютері адміністратора вручну вказати *IP* адресу в мережі 192.168.88.0, або

отримати цю адресу через протокол *DHCP*. В полі *Login* вводиться ім'я адміністратора, стандартний є «*admin*». В полі *Password* вписується пароль, адміністратора, за замовчування він відсутній.

Для зручності користування утилітою в подальшому можливо зберегти введений пароль, та запам'ятати минулу сесію. Для цього потрібно вибрати відповідні галочки нижче.

Для заходу в основне вікно утиліти, натискаємо кнопку *Connect*. Основне вікно утиліти показано на рис. 3.3.

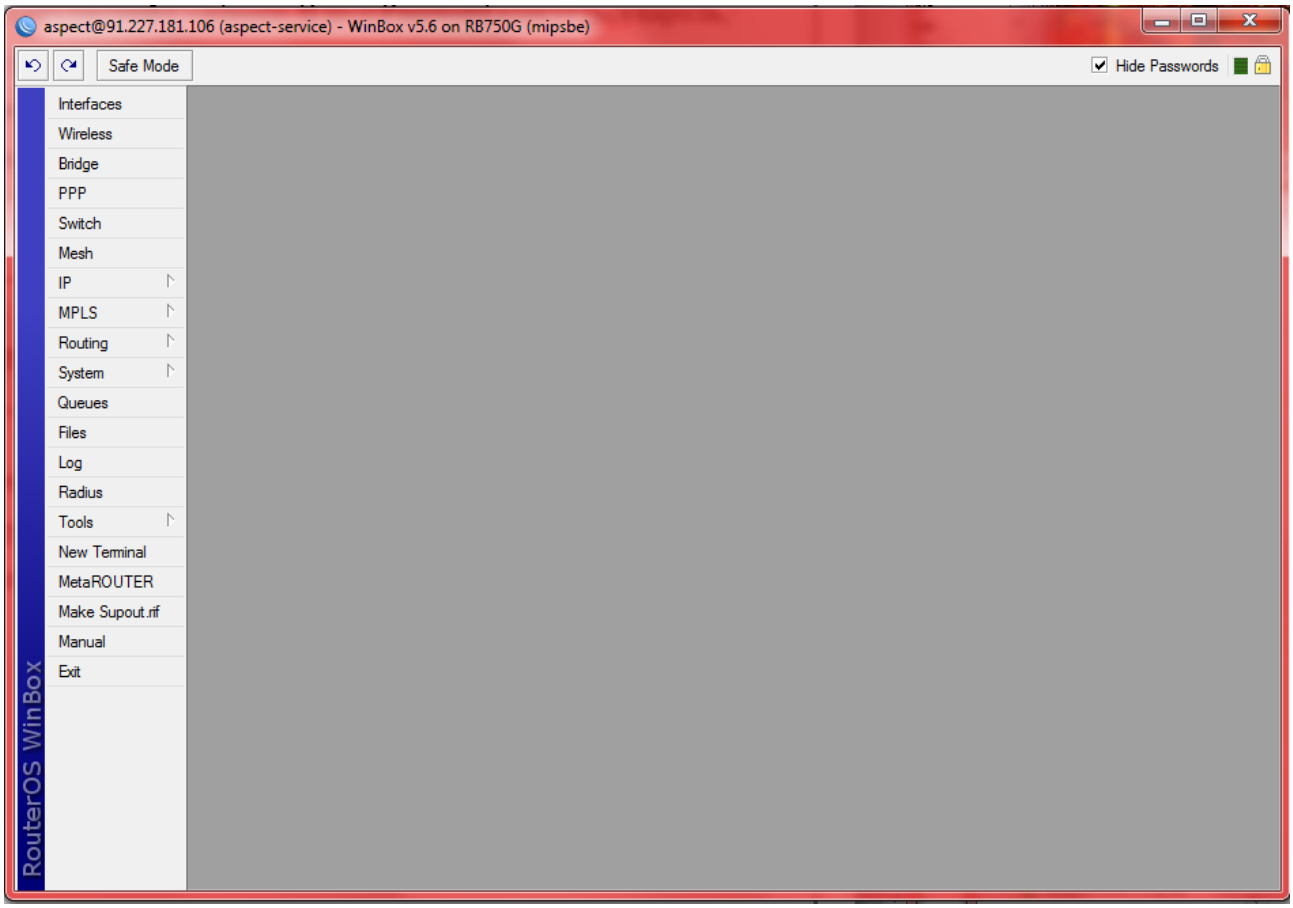

Рис. 3.3 Основне вікно утиліти *Winbox*

На вкладці *Interfaces* показується усі наші фізичні та віртуальні інтерфейси. Їх можливо перейменувати у відповідності до потреб компанії. В нашому випадку ми назвали 3 підрозділи іменами *aspect*, *buch*, *vigil*, а 2 інші інтерфейси віддані для вхідних ліній від провайдерів: *wan* та *wan2*. Перед назвою кожного інтерфейсу є буква його статусу: *D* – інтерфейс налаштований на автоматичне отримання *IP* адреси, *X* – інтерфейс вимкнено, *R* – інтерфейс активний, *S* – інтерфейс залежний від іншого інтерфейсу. Точніше про останній

пункт, в роутері всі інтерфейси є незалежні та працюють окремо один від одного, але, при, потребі можливо налаштувати так щоб він працював в парі, в залежності, від будь-якого іншого інтерфейсу. Це означає що на залежний інтерфейс будуть поширюватися налаштування головного інтерфейсу, при цьому роутер отримує функцію свіча на два інтерфейси.

Приступаємо до конфігурації в консолі, результат переглянемо в графічному інтерфейсі утиліти:

/ interface address set 1 name="wan" disabled=no
/ interface address set 1 name="aspect" disabled=no
/ interface address set 0 name="buch" disabled=no
/ interface address set 1 name="vigil" disabled=no
/ interface address set 0 name="wan2" disabled=no

Вікно зі списком інтерфейсів показано на рис. 3.4.

Тепер кожному з інтерфейсів потрібно задати *IP* інтерфейсів. Наша задача повністю розділити мережі різних відділів один від одного, тому для кожного з внутрішніх інтерфейсів вибираємо свою адресу мережі. *Aspect*: 192.168.3.1, *buch*: 192.168.1.1, *vigil*: 192.168.2.1. Маска підмережі для всіх інтерфейсів буде однаковою: 255.255.255.0, або /24. Інші *IP* адреси ми отримуємо від Інтернет провайдерів, нехай перший провайдер надає адресу 91.227.181.106/28, шлюз 91.227.181.96, *dns* сервер: 91.227.180.2. А другий провайдер 213.180.204.8/28, шлюз 213.180.204.1, *dns* сервер: 194.85.61.20, 193.232.130.14

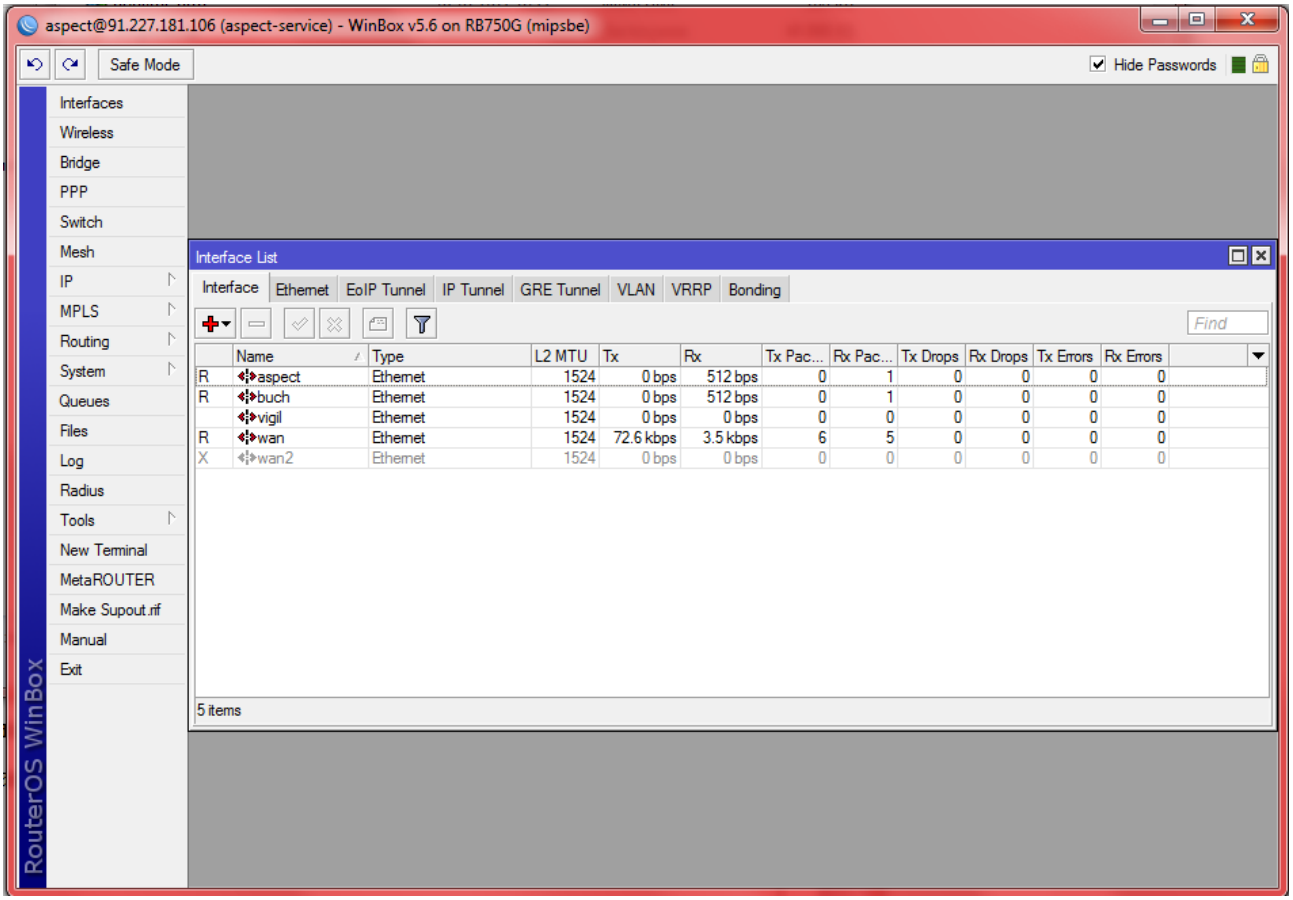

Рис. 3.4 Список інтерфейсів

Для налаштування Інтернет з’єднання прописуємо команди в консолі:

/ip address add interface=wan address=91.227.181.106/28

/ip route add gateway=91.227.181.96

/ip dns set servers="91.227.180.2"

/ip address add interface=wan2 address=213.180.204.8/28

/ip route add gateway=213.180.204.1

/ip dns set servers=194.85.61.20 193.232.130.14

Тепер налаштовуємо клієнтські інтерфейси:

/ip address add interface=aspect address=192.168.3.1/24

/ip address add interface=buch address=192.168.1.1/24

/ip address add interface=aspect address=192.168.2.1/24

Для того щоб наш роутер розпочав роздавати Інтернет через клієнтські інтерфейси, потрібно додати нові шлюзи:

/ip route add dst-address=0.0.0.0/0 gateway=91.227.181.106, 213.180.204.8

Якщо у першого провайдера швидкість доступу в два рази більша ніж у іншого, тоді 2/3 вихідних запросів потрібно віддати першому провайдеру, інші 1/3 на другого. Тоді прописуємо:

/ip route add dst-address=0.0.0.0/0 gateway=91.227.181.106, 91.227.181.106, 213.180.204.8

Для того щоб надати адміністратору чи користувачу доступ до свого комп'ютера або файлового серверу, необхідно налаштувати так званий проб рос портів через *NAT* на потрібну *IP* адресу. Наприклад, для можливості користування віддаленого робочого столу потрібно в консолі прописати правило для *NAT*:

/ip firewall nat add chain=dstnat dst-address=91.227.181.106 protocol=tcp dst-port=3389 action=dst-nat to-addresses=192.168.1.2 to-ports=3389

Вище ми прокинули порт 3389, який потрібен для функціонування віддаленого робочого столу. Тепер дописуємо правило в *Firewall*:

/ip firewall nat add action=accept chain=input dst-port=3389 protocol=tcp

Після додавання в нашу конфігурацію другого провайдера виникає проблема відповіді деяких сервісів по потрібному каналу, якщо звертатися до них через *NAT*. Наприклад, веб-сервер виділений через *NAT* на *IP* адресу першого провайдера отримує запит, то ймовірність отримання відповіді по тому ж каналу

вже не 100%.

Вирішити цю проблему допоможе механізм маркування пакетів. Таблиця Mangle призначена для операцій по класифікації і маркуванню пакетів та з'єднань, а також модифікації заголовків пакетів. Нас цікавить цепочка *prerouting*, яка дозволяє маркувати пакети до їх маршрутизації. Першим кроком створимо правило mangle:

/ip firewall mangle add chain=prerouting src-address=192.168.1.2 protocol=tcp src-port=3389 action=mark-routing new-routing-mark=wam

Пакети з локальної адреси 192.168.1.2 будуть маркуватися як *wan.* Тепер створимо маршрут:

/ip route add gateway=91.227.181.106 routing-mark=wan

Тепер все що відбувається з маркером *wan* проходить через шлюз першого провайдера.

Якщо кількість мережевих пристроїв в компанії постійно зростає, то настає момент коли стає не зручно налаштовувати кожний окремий пристрій. Незручність зв'язана з налаштуванням мережі на робочих місцях можливо прибрати за допомогою вбудованого у *RouterOS* серверу *DHCP*. Налаштування *DHCP* проходить в три етапи. Спочатку створюється пул *IP*-адрес які будуть роздаватися клієнтським комп'ютерам:

/ip pool add name=aspect ranges=192.168.3.2-192.168.3.254
/ip pool add name=buch ranges=192.168.1.2-192.168.1.254
/ip pool add name=vigil ranges=192.168.2.2-192.168.2.254

Потім створюємо екземпляр *DHCP*-сервера, який буде обслуговувати вказаний інтерфейс та роздавати *IP*-адреси з раніше створеного пула:

/ip dhcp-server add name=aspect interface=aspect address-pool=aspect disabled="no"

/ip dhcp-server add name buch interface= buch address-pool= buch disabled="no"

/ip dhcp-server add name= vigil interface= vigil address-pool= vigil disabled="no"

Нарешті визначаємо адреси шлюзу за замовчуванням і сервера *DNS*, який будуть використовувати в нашій мережі.

/ip dhcp-server network add address=192.168.1.0/24 gateway=192.168.1.1 dns-server=192.168.1.1

/ip dhcp-server network add address=192.168.2.0/24 gateway=192.168.2.1 dns-server=192.168.2.1

/ip dhcp-server network add address=192.168.3.0/24 gateway=192.168.3.1 dns-server=192.168.3.1

Готово. Далі нам потрібен свій *DNS* сервер. Власний сервер дозволить, по-перше, організувати адресацію хостів в нашій мережі по людино-подібним іменам замість *IP* адрес, по-друге, створити свій бар'єр на шляху доступу до нехороших сайтів.

Налаштування *DNS* серверу виконується наступною командою:

/ip dns set servers= 91.227.181.106, 213.180.204.8 allow-remote-requests=yes

З цього моменту наш маршрутизатор буде відповідати на *DNS* запити клієнтів. Відповіді він буде шукати у власній базі, а не знайшовши їх там, переадресує запит провайдерським серверам. Поповнити власну базу *DNS* можливо командою:

/ip dns static add address="192.168.3.100" name="printer.office"

Після чого можливо буде звертатися до хосту за ім’ям . Якщо в команді вказати фіктивний хост:

/ip dns static add address="127.0.0.1" name="vk.vom"

То за ім’ям хост буде пінгуватися, але це вже буде довшим інший хост, і любий інший доступ до нього, наприклад через браузер, буде неможливий. Починаючи з версії 3.0rc7 *RouterOS* дозволяє в якості *DNS* імені хоста використовувати регулярні вирази в нотації *POSIX basic*. Це значить що одною командою можливо заблокувати цілу групу хостів. Так команда:

*/ip dns static add address="127.0.0.1" name=".*video.*"*

Заблокує доступ до любого хосту, в імені якого в любому місці міститься слово *video*.

Одним з найбільших зловживань користувачів на робочих місцях – це користування децентралізованими піринговими мережами. Це так звані файлообмінні мережі, вони є найбільшими генераторами трафіку в мережі та здатні значно завантажувати канали.

Як відомо, всі пірингові клієнти працюють з портами вищими ніж 1000. Наприклад *eMule* порти - *TCP* 4672 та *UDP* 4672, *BitTorrent* - від 32459 до 65000. Простий пошук у веб-браузері рідко коли використовує порти вище 1000. Спочатку потрібно створити заборонне правило:

/ip firewall filter add chain=forward src-address=192.169.1.0/24 protocol=tcp dst-port=1000-65535 tcp-flags=syn connection-limit=1,32 connection-state=new action=drop

/ip firewall filter add chain=forward src-address=192.169.2.0/24 protocol=tcp

dst-port=1000-65535 tcp-flags=syn connection-limit=1,32 connection-state=new action=drop

/ip firewall filter add chain=forward src-address=192.169.3.0/24 protocol=tcp dst-port=1000-65535 tcp-flags=syn connection-limit=1,32 connection-state=new action=drop

Дане налаштування обмежує кількість одночасних налаштувань кількістю в 10 потоків. Цей захід різко знизить навантаження на канал.

Якщо нам потрібно поділити швидкість порівну між усіма користувачами, при чому, щоб швидкість не виділялася на клієнтів, які в даний момент не користуються Інтернетом, а віддавати усім іншим, ще щоб при великій кількості користувачів та вузькому каналі отримати деяку «буферність» каналу. Для цього потрібен скрипт написаний на внутрішній мові *RouterOS*.

Нехай адреси користувачів 192.168.2.100-192.168.2.102. Додаємо трьох користувачів в списки, при чому, тільки тих, хто реально існує. Не потрібно забивати цілий діапазон якщо ми його не використовуємо, так як кожний лишній запис дуже сильно відчується на швидкодії.

/ip firewall address-list add list="users" address=192.168.2.100
/ip firewall address-list add list="users" address=192.168.2.101
/ip firewall address-list add list="users" address=192.168.2.102

Щоб не добавляти списки вручну є невеликий цикл, який дозволить прискорити процес додавання діапазонів. Одноразовий запуск такого скрипту зробить аналогічні дії того що ми робили вище. Код скрипту показаний додатку А.

Другим кроком додаємо записи в */queue* та */ip firewall mangle*, знову ж щоб звільнити адміністратора від рутинної роботи з додаванням записів вручну, є невеликий скрипт. Цей скрипт необхідно запускати кожний раз, коли додається або видаляються записи в */ip firewall address-list list=users*, його основна задача

додати нові записи і видаляти непотрібні старі. Після закінчення роботи скрипту в системний лог буде записана інформація про кількість доданих або видалених записів. Цей скрипт показаний у додатку А.

Тепер запустимо скрипт обчислювальної та виконавчої частини схеми. Скрипт визначає, чи ввімкнена підтримка нічного часу та перевіряє даний час, згідно діапазону часу вибирає пропускну здатність каналу. Скрипт заміряє з якою швидкістю працює клієнт і якщо вона перевищує *ActiveThresholddown*, додає його як активного на прийом. Також збільшує лічильник активних на прийом користувачів. Далі виконує перевірку *ActiveThresholdup*, якщо користувач перевищив цю відмітку, то додає його як активну на віддачу. Також збільшує лічильник активних на віддачу.

Розраховує: *MaxRateDownload* ділить на кількість активних користувачів на прийом, виводить швидкість на користувача. *MaxRateUpload* ділить на кількість активних користувачів на віддачу, виводить швидкість на користувача. Далі встановлює ліміт всім користувачам в *Queue Tree* згідно розрахованої вище схемі. Далі розраховує значення в кілобітах та виводить в лог статистику.

Змінні в скрипті на початку:

MaxRateDownload –Ширина каналу на всіх користувачів (прийом) Біт/сек.

MaxRateUpload –Ширина каналу на всіх користувачів (віддачі) Біт/сек.

MaxRateDownloadNight - Ширина каналу на всіх користувачів в нічний час (прийом) Біт/сек.

MaxRateUploadNight - Ширина каналу на всіх користувачів в нічний час (віддача) Біт/сек.

ActiveThresholddown –Поріг при перевищенні якого користувач буде рахуватися активним (прийом) Біт/сек.

ActiveThresholdup — Поріг при перевищенні якого користувач буде рахуватися активним (віддача) Біт/сек.

usenighttime –може приймати значення «yes» та «no», дозволяє скрипту використовувати іншу ширину каналу. У відповідності з нічним тарифом.

nighttimestart –сповіщає скрипту початок дії нічного тарифу.

nighttimestop - сповіщає скрипту кінець дії нічного тарифу.

Скрипт представлений в додатку А.

Важливо помітити сироку в системному лозі *Performance Time*, дана строка дозволяє правильно встановити інтервал виконання скрипта в планувальнику. *Performance Time* відображає час виконання скрипту в секундах з точність до 1 сек. Даний час різко збільшується з кількість користувачів доданих в */ip firewall address-list list=«users»*. Інтервал виконання скрипта потрібно встановлювати виходячи з формули: *Performance Time* + 10-15 сек. При першому запуску скрипта встановлюємо інтервал рівний 1-2 хвилинам.

Параметри скрипта:

:local MaxRateDownload ("15000000");

:local MaxRateUpload ("15000000");

:local MaxRateDownloadNight ("20000000");

:local MaxRateUploadNight ("20000000");

Потрібно встановлювати чуть менше чим ширина каналу, щоб запобігти гальмувань при активності ще одного користувача, який до цього не використовував інтернет.

:local ActiveThresholddown ("15000");

:local ActiveThresholdup ("15000");

Ці значення встановлюються згідно своїм потребам, але виходячи з розрахунку: кількість користувачів помножене на *Thresholddown* не повинно перевищувати *MaxRateDownload* інакше користувачі будуть завжди не активними. Але й *users*ActiveThresholdup* не повинно перевищувати *MaxRateUpload*. Завжди потрібно залишати деякий запас.

Тепер розглянемо можливість розширення кількості наших відділів. В нашому роутері є 5 портів, два з яких ми використовуємо для отримання Інтернету від провайдерів а інші три для роздачі цього Інтернету користувачам. Але існує ймовірність організації ще одного відділу, якому потрібно виділити власну підмережу. Тоді виникає проблема недостатньої кількості портів в нашому роутері. Як вихід, можливо придбати новий роутер з більшою кількістю

портів, але краще скористатися можливостями *RouterOS* та додати *VLAN* інтерфейс який буде обслуговувати підмережу 192.168.4.0/24. Додаємо інтерфейс в терміналі:

/interface vlan add name=VLAN1 vlan-id=1 interface=ether3 disabled=no

Тепер додамо нашому новому інтерфейсу *ip* адресу:

/ip address add address=192.168.4.0/24 interface=VLAN1

Відтепер, після налаштування *NAT* так як і в інших портах, всі комп'ютери які подключені до мережевого порту №3, можуть отримувати Інтернет від цього порту знаходячись в різних під мережах.

Поговоримо про технологію *port knocking*. Вона дозволяє отримувати доступ до управління роутером, навіть якщо всі порти зачинені. Візьмемо базовий випадок, після звернення до визначених портів в певному порядку, нам відкриється доступ до 22 порту для керування по *ssh*. Допустимо ці порти будуть 44 порт *udp*, 8791 порт *tcp*, 62014 порт *tcp*, 28014 порт *udp* и 12761 *udp*. При «простукуванні» першого порту, адреса користувача буде заноситися в тимчасовий список. Після простукування наступного порту, перевіряється наявність адреси користувача в тимчасовому списку. Щоб не створювати велику кількість списків, обмежимо час життя списку в рамках 10-15 хвилин. Пишемо наступні команди в термінал:

/ip firewall filter

add chain=input protocol=udp dst-port=44 action=add-src-to-address-list address-list=knoc_stage1 address-list-timeout=10s disabled=no

add chain=input protocol=tcp dst-port=8791 src-address-list=knock_stage1 action=add-src-to-address-list address-list=knock_stage2 disabled=no

add chain=input protocol=tcp dst-port=8791 src-address-list=knock_stage2

action=add-src-to-address-list address-list=knock_stage3 address-list-timeout=10s disabled=no

/ip firewall filter>add chain=input protocol=tcp dst-port=62014 src-address-list=knock_stage3 action=add-src-to-address-list address-list=knoc_stage4 address-list-timeout=10s disabled=no

add chain=input protocol=udp dst-port=28014 src-address-list=knock_stage4 action=add-src-to-address-list address-list=knock_stage5 address-list-timeout=10s disabled=no

add chain=input protocol=udp dst-port=12671 src-address-list=knock_stage5 action=add-src-to-address-list address-list=knock_safe address-list-timeout=15m disabled=no

В останньому рядку ми додаємо *ip* в тимчасовий список на 15 хвилин. Далі додамо правило на корект до 22 порту для списку довірених і заборонити доступ для всіх інших.

add chain=input protocol=tcp dst-port=22 src-address-list=knock_safe action=accept disabled=no

add chain=input protocol=tcp dst-port=22 action=reject reject-with=tcp-reset disabled=no

Останнім правилом ми скидаємо всі з'єднання на 22 порт, бокування з *reject-with=tcp-reset* дозволяє огородитися від сканування портів, *nmap* просто не побачить цей порт. Щоб достукатися до роутера, потрібно використати утиліту *knock* для *Windows*, під *Linux* утиліту *knockd*.

knock 192.168.100.1 44:udp 8791:tcp 62014:tcp 28014:udp 12761:udp

Буває так що один з користувачів в компанії може підчепити Інтернет «червя», який починає відсилати спам з комп'ютера жертви. На це досить боляче реагують Інтернет провайдери, можуть навіть відключити компанію від

Інтернету. Тому потрібно блокувати інфікованих користувачів. Для початку блокуємо інфікованих користувачів:

/ ip firewall filter add chain=forward protocol=tcp dst -port=25 src-address-list=spammer action=drop

Додамо інфікованого користувача в список спамерів та заблокуємо йому доступ до SMTP трафіку на один день.

/ ip firewall filter add chain=forward protocol=tcp dst -port=25 connection -limit=30,32 limit=50,5 src-address-list=!spammer action=add-src-to-address-listaddress-list=spammer address -list-timeout=1d

Для моніторингу нашої мережі є можливість використовувати так звані пакети *netflow*. Але ці пакети потребують додаткового програмного забезпечення на комп'ютері адміністратора. *RouterOS* надає тільки статистичні дані, а їх аналізом займається стороння програма. Для *Windows* ситем найкраще підходить програма *NetFlow Analyzer*. Це веб-програма, написана на *Java*, в якості сервера виступає *Apache*, для зберігання даних використовується *MySQL*. Завдяки наглядності приборних панелів програми можливо спершого погляду оцінити отриману ситуацію, отримати дані про навантаження на різні участки мережі та трафік який генерує кожен користувач мережі.

Для отримання статистики потрібно налаштувати *RouterOS* на відправку пакетів на *IP* адресу адміністратора мережі. Це можливо зробити в консолі:

/ip traffic-flow set enabled=yes
/ip traffic-flow target add address=192.168.3.59:9996 version=9

або через утиліту *Winbox*, вікно налаштувань *NetFlow* показано на рис. 3.5.

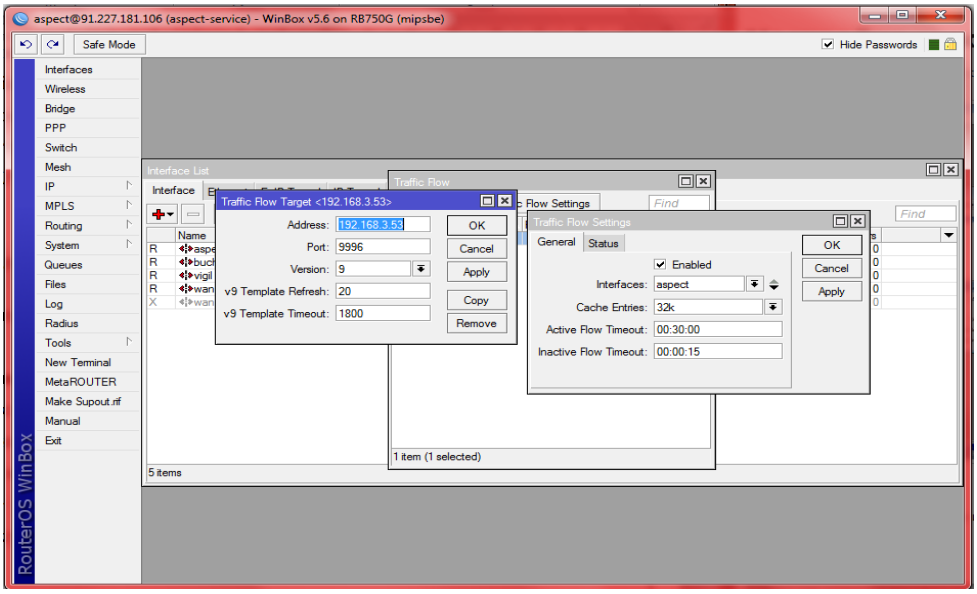

Рис. 3.5 Вікно налаштувань пакетів *NetFlow*

Тепер наш *Mikrotik* відсилає *NetFlow* пакети на комп'ютер 192.168.3.59 та порт 9996. Вікно програми *NetFlow Analyzer* показано на рис. 3.6.

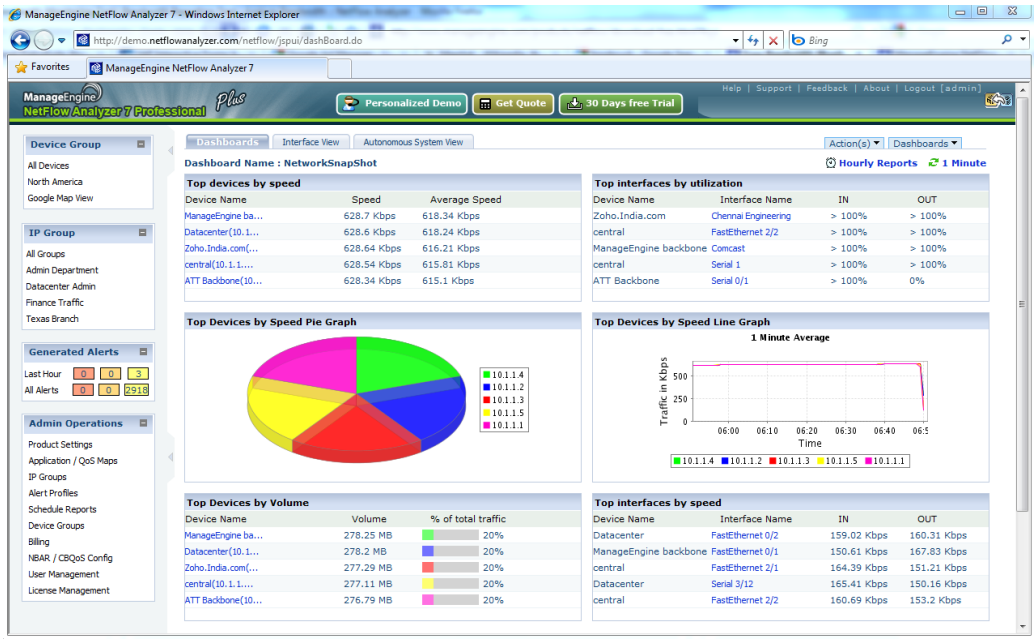

Рис. 3.6 Вікно програми *NetFlow Analyzer*

3.5 Висновки до розділу

Під час розробки модернізованої моделі комп'ютерної мережі офісу було використано можливості операційних систем на основі ядра *Linux*, які були реалізовані в операційній системі *RouterOS*. При чому розглянуто саме програмно-апаратне рішення на основі цієї системи.

Вкінці розробки отримано нову модернізовану модель комп'ютерної мережі офісу, що дозволить підвищити продуктивність праці в даній компанії.

ВИСНОВКИ

Дуже важливо оволодіти навиками розробки комп'ютерних мереж, операційними системами та їх мережними можливостями. Студенти, які в майбутньому стануть адміністраторами комп'ютерних мереж, повинні знати як вирішити задачі які може поставити майбутній роботодавець. На сьогодні небагато компаній можуть дозволити собі тримати окремі сервери та штат спеціалістів для підтримання своєї мережі. Тому потрібно знати як вирішити поставлені задачі максимально простими та фінансово вигідними способами.

При аналізі спеціалізованих книг та сайтів ми бачимо що мало уваги приділяється програмно-апаратним засобам керуванням Інтернет-каналом. В даній дипломній роботі доведено що більшість задач, які можуть стояти в порівняно невеликій організації, можливо вирішити не використовуючи дорогі сервери та спеціалізоване програмне забезпечення для них. Для розв'язання наукового завдання було виконано наступні дії:

1. Проаналізували існуючі Інтернет-технології, топології мереж та технології завдяки яким функціонує мережа Інтернет.
2. Проаналізували технології керування Інтернет-каналом для різних операційних систем. Роздивилися можливості використання засобів вбудованих в операційну систему так і окремі програмні засоби які розповсюджуються на комерційних умовах.
3. Вибрали вигідний програмно-апартний засіб керування Інтенет каналом, розглянули можливості вбудованого програмного забезпечення заснованого на ядрі операційної системи *Linux* з її перевагами:

- дозволяє проводити моніторинг таких мережевих сервісів як *SMTP, HTTP, SSH, POP3, IMAP* і багатьох інших;
- ядро системи засноване на ядрі ОС *Linux*, що забезпечує високу швидкодію при низькому навантаженні на устаткування програмно-апаратного комплексу;
- гнучка система конфігурації і широкі можливості налаштувань, існує

можливість розширення функціональності за рахунок підвищення рівня ліцензії та використання пакетів розширення;

- є можливість керувати нею віддалено за допомогою *web* інтерфейсу, з'єднання з яким може бути захищено *SSL* шифруванням, через *SSH* доступ, *telnet*. Також можливо використовувати фірмову утиліту *Winbox*, або використовувати *api* для написання своєї утиліти;

4. Розробили приклад налаштування маршрутизатора *RouterBOARD 750G* для порівняно невеликого офісу окремої компанії.

В процесі використання змодельованої комп'ютерної мережі користувачі зможуть отримати стабільний доступ до всесвітньої мережі. Оскільки ми використовуємо два різних Інтернет провайдерів, наші користувачі захищені від можливих проблем на магістралі провайдера. Відділи організації були розмежовані, що дає можливість не хвилюватися що документи які можуть містити комерційну таємницю можуть потрапити до користувачів мережі з іншого відділу, які за політикою безпеки організації не мали отримати доступ до них.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабич М.П. Атестаційні роботи магістрів і спеціалістів: Навчально-методичний посібник/ М.П. Бабич, І.А. Жуков – К.: НАУ, 2004. – 206 с.
2. Куроуз Д Компъютерные сети/ Куроуз ДЖ., Росс К. 2-е изд. – СПб.: Питер, 2004.- 765 е.:
3. Баловсяк Н.В. Интернет. Новые возможности. Трюки и эффекты./ Н.В. Баловсяк, О.М. Бойцев– СПб.: Питер, 2008. – 304 с.: ил.
4. Леонтьев В.А. Безопасность в сети Интернет/ В.А. Леонтьев – М.: ОЛМА Медиа Групп, 2008. – 256 с.
5. Гультяев А. К. Виртуальные машины: несколько компьютеров в одном/ А. К. Гультяев — СПб.: Питер 2006.— 224 с: ил.
6. Гуменюк В.А. Проблемы безопасности протоколов TCP/IP/ В.А. Гуменюк. - К.: НАУ, 2003. – с. 246-251.
7. Основи теорії мереж передавання та розподілу даних. Навч. посіб./ Жуков.І.А., Віноградов М.А., Дрововозов В.І., Халімон Н.Ф. - К.: Книжкове вид-во НАУ, 2006. - 272 с.
8. Жуков І.А. Комп'ютерні мережі та технології./ І.А. Жуков, В.О. Гуменюк, І.Є. Альтман– К.: НАУ, 2004. – 276 с.
9. Жуков І.А. Ідентифікація та прогнозування атак на комп'ютерні мережі на підставі тензорного нейромережного базиса // Проблеми інформатизації та управління"/ І.А. Жуков, Ю.М. Мінаєв, М.М. Гузій, – К.: НАУ, 2004.– Вип.9.
10. Закер К. - Компьютерные сети. Модернизация и поиск неисправностей в подлиннике,/ Закер К. 2003 год, 1008 стр.
11. Иванова Т.И. Корпоративные сети связи./ Т.И.Иванова—М.: Эко-Трендз, 2001.—282с.
12. Камалян А.К. Компьютерные сети и средства защиты информации: Учебное пособие/ А.К. Камалян, С.А. Кулев , К.Н. Назаренко К.Н. – Воронеж: ВГАУ, 2003.-119с.

13. Краковський В.Я. Computer Modeling and Simulation Textbook. Комп'ютерне моделювання. Підручник (англійською мовою)./ В.Я.Краковський К.: NAU, 2003. – 212 р., Київ: НАУ, 2003. –212 с.
14. Курносов А.П. Практикум по информатике/ А.П. Курносов Воронеж: ВГАУ, 2001.- 173 с.
15. Макарова Н.В. Информатика — М.: Финансы и статистика,/ Н.В. Макарова 1997. 768 с.
16. Майоров С.А. Введение в микроЭВМ./ С.А. Майоров, В.В. Кириллов, А.А. Приблуда – Л.: Машиностроение, 1988. – 304 с.
17. Олифер В.Г. Сетевые операционные системы/ В.Г. Олифер, Н.А. Олифер – СПб.: Питер, 2002. – 544 с.
18. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы./ В.Г. Олифер, Н.А. Олифер - СПб.: Питер, 2002.- 672 с.
19. Пономаренко Л.А. Инструментальные средства проектирования, имитационного моделирования и анализа компьютерных сетей./ Л.А. Пономаренко, В.И. Щелкунов, А.Я. Скляров – К.: Наукова думка, 2005. – 508 с.
20. Симонович С.В. Информатика. Базовый курс/ С.В. Симонович— СПб.: издательство "Питер", 2000. — 640 с.

ДОДАТОК А

1. Скрипт для додавання користувачів в списки:

```
#Settings
#########################################################
:local start ("100");
:local stop ("102");
:local net ("192.168.2.");
#########################################################

#########################################################
:global count ($start);
:for count from=$start to=$stop step=1 do={
/ip firewall address-list add list="users" address=( $net . $count);};
#########################################################
```

2. Скрипт для додавання записів в */queue* та */ip firewall mangle*.

```
#Settings
#########################################################
:local DownloadParent ("Download");
:local UploadParent ("Upload");
#########################################################
#Internal Var
#########################################################
:local i;
:local z;
:local userX;
:local enum (" ");
:local mark;
:local qrd;
:local qru;
:local mrd;
```

```
:local mru;
:local qrdadd;
:local qruadd;
:local mrdadd;
:local mruadd;
:set qrd (0);
:set qru (0);
:set mrd (0);
:set mru (0);
:set qrdadd (0);
:set qruadd (0);
:set mrdadd (0);
:set mruadd (0);
############################################################

############################################################
:log warning ("Rules Manager Started!");

:if ([/queue type find name="dshaper_down"] = "") do={ /queue type add
name="dshaper_down" kind="pcq" pcq-classifier=dst-address pcq-rate=0 pcq-
limit=50 pcq-total-limit=2000;};

:if ([/queue type find name="dshaper_up"] = "") do={ /queue type add
name="dshaper_up" kind="pcq" pcq-classifier=src-address pcq-rate=0 pcq-
limit=50 pcq-total-limit=2000;};

:if ([/queue tree find name=$DownloadParent] = "") do={ /queue tree add
name=$DownloadParent parent="global-out" queue="dshaper_down" priority=8;};
:if ([/queue tree find name=$UploadParent] = "") do={ /queue tree add
name=$UploadParent parent="global-out" queue="dshaper_up" priority=8;};
```

```
:foreach i in=[/ip firewall address-list find list="users"] do={ :set userX [/ip firewall address-list get $i address];

:if ([/queue tree find name=($userX . "_down")] = "") do={ /queue tree add name=($userX . "_down") parent=$DownloadParent queue="dshaper_down" packet-mark=($userX . "_down") priority=8; :set qrdadd ($qrdadd+1); };
:if ([/queue tree find name=($userX . "_up")] = "") do={ /queue tree add name=($userX . "_up") parent=$UploadParent queue="dshaper_up" packet-mark=($userX . "_up") priority=8; :set qruadd ($qruadd+1);};

:set enum (" ");
:set enum ([/ip firewall mangle find comment=($userX . "_up")]);
:if ($enum = "") do={ /ip firewall mangle add chain=forward src-address=$userX dst-address=0.0.0.0/0 action=mark-packet new-packet-mark=($userX . "_up") comment=($userX . "_up") disabled=no passthrough=yes; :set mruadd ($mruadd+1);
};

:set enum (" ");
:set enum ([/ip firewall mangle find comment=($userX . "_down")]);
:if ($enum = "") do={ /ip firewall mangle add chain=forward src-address=0.0.0.0/0 dst-address=$userX action=mark-packet new-packet-mark=($userX . "_down") comment=($userX . "_down") disabled=no passthrough=yes; :set mrdadd ($mrdadd+1);
};

};

:foreach z in=[/queue tree find parent=$DownloadParent] do={
```

:set mark [/queue tree get $z name];
:if ($mark !="") do={
:set mark ([:tostr $mark]);
:set mark ([:pick $mark 0 ([:len $mark]-5)]);
:if ([/ip firewall address-list find address=$mark] = "") do={/queue tree remove [/queue tree find name=($mark . "_down")]; :set qrd ($qrd+1); };};};

:foreach z in=[/queue tree find parent=$UploadParent] do={
:set mark [/queue tree get $z name];
:if ($mark !="") do={
:set mark ([:tostr $mark]);
:set mark ([:pick $mark 0 ([:len $mark]-3)]);
:if ([/ip firewall address-list find address=$mark] = "") do={/queue tree remove [/queue tree find name=($mark . "_up")]; :set qru ($qru+1); };};};

:foreach z in=[/ip firewall mangle find src-address="0.0.0.0/0" action="mark-packet" chain="forward"] do={
:set mark [/ ip firewall mangle get $z comment];
:if ($mark !="") do={
:set mark ([:tostr $mark]);
:set mark ([:pick $mark 0 ([:len $mark]-5)]);
:if ([/ip firewall address-list find address=$mark] = "") do={
:if ([/ip firewall mangle find comment=($mark . "_down")] != "") do={/ip firewall mangle remove [/ip firewall mangle find comment=($mark . "_down")]; :set mrd ($mrd+1); }}}}

:foreach z in=[/ip firewall mangle find dst-address="0.0.0.0/0" action="mark-packet" chain="forward"] do={
:set mark [/ ip firewall mangle get $z comment];
:if ($mark !="") do={

```
:set mark ([:tostr $mark]);
:set mark ([:pick $mark 0 ([:len $mark]-3)]);
:if ([/ip firewall address-list find address=$mark] = "") do={
:if ([/ip firewall mangle find comment=($mark . "_up")] != "") do={/ip firewall mangle remove [/ ip firewall mangle find comment=($mark . "_up")]; :set mru ($mru+1); }}}}
##########################################################

##########################################################
:log info ("----------------------------------------");
:log warning ("Rules Manager:");
:log info ("Queue Tree Download Records Added: " . $qrdadd);
:log info ("Queue Tree Upload Records Added: " . $qruadd);
:log info ("Mangle Download Records Added: " . $mrdadd);
:log info ("Mangle Upload Records Added: " . $mruadd);
:log info ("Queue Tree Download Records Deleted: " . $qrd);
:log info ("Queue Tree Upload Records Deleted: " . $qru);
:log info ("Mangle Download Records Deleted: " . $mrd);
:log info ("Mangle Upload Records Deleted: " . $mru);
:log info ("----------------------------------------");
##########################################################
```

3. Скрипт обчислювальної та виконавчої частини.

```
#Settings
##########################################################
:local MaxRateDownload ("15000000");
:local MaxRateUpload ("15000000");
:local MaxRateDownloadNight ("20000000");
:local MaxRateUploadNight ("20000000");

:local ActiveThresholddown ("15000");
```

```
:local ActiveThresholdup ("15000");

:local usenighttime ("yes");
:local nighttimestart ("02:00");
:local nighttimestop ("08:00");
############################################################

#Internal Var
############################################################
:local z;
:local i;
:local ii;
:local userX;
:local timedelay (0);
:local startmin;
:local startsec;
:local stopmin;
:local stopsec;
:local scripttimedelay (0);
:local scriptstartmin;
:local scriptstartsec;
:local scriptstopmin;
:local scriptstopsec;
:local userscount ("0");
:local userstmp ("");
:local firstdowntmp ("");
:local firstuptmp ("");
:local twodowntmp ("");
:local twouptmp ("");
:local activedownuserstmp ("");
```

```
:local activeupuserstmp ("");
:local activedowncount ("0");
:local activeupcount ("0");
############################################################

############################################################
:set scriptstartmin ([: pick [/system clock get time] 3 5]);
:set scriptstartsec ([: pick [/system clock get time] 6 8]);

:if ($usenighttime = "yes") do={
:set nighttimestart ([: pick $nighttimestart 0 2] . [: pick $nighttimestart 3 5]);
:set nighttimestop ([: pick $nighttimestop 0 2] . [: pick $nighttimestop 3 5]);
:local currenthours ([: pick [/system clock get time] 0 2]);
:local currenttime ([: pick [/system clock get time] 0 2] . [: pick [/system clock get time] 3 5] );
:local acttime ("day");
:local starttime ("day");
:if ($currenthours < 10) do={ :set acttime ("night"); };
:if ( [: pick $nighttimestart 0 2] < 10) do={ :set starttime ("night"); };
:local night ("no");

:if ($starttime = "night") do={
:if ( $currenttime > $nighttimestart && $currenttime < $nighttimestop) do={ :set night ("yes"); };
};
:if ($starttime = "day") do={
:if ( $acttime = "day") do={
:if ( $currenttime >= $nighttimestart) do={ :set night ("yes"); };
};
:if ( $acttime = "night") do={
```

```
:if ( $currenttime < $nighttimestop ) do={ :set night ("yes"); };
};};

:if ($night = "yes") do={
:set MaxRateDownload ($MaxRateDownloadNight);
:set MaxRateUpload ($MaxRateUploadNight);
};
};

:set ActiveThresholddown ($ActiveThresholddown / 8);
:set ActiveThresholdup ($ActiveThresholdup / 8);

:foreach i in=[/ip firewall address-list find list="users"] do={ :set userX [/ip firewall
address-list get $i address];
:set userscount ($userscount+1);
:set userstmp ($userstmp . $userX . ",");
};

:local users [:toarray $userstmp];

:set startmin ([: pick [/system clock get time] 3 5]);
:set startsec ([: pick [/system clock get time] 6 8]);

:global dcount ("1");
:for dcount from=1 to=$userscount step=1 do={
:set firstdowntmp ($firstdowntmp . [/ip firewall mangle get [/ip firewall mangle find
comment=[:pick $users ($dcount-1)] . "_down"] bytes] . ",");
:set firstuptmp ($firstuptmp . [/ip firewall mangle get [/ip firewall mangle find
comment=[:pick $users ($dcount-1)] . "_up"] bytes] . ",");
};
```

```
:set stopmin ([: pick [/system clock get time] 3 5]);
:set stopsec ([: pick [/system clock get time] 6 8]);

:global dcount ("1");
:for dcount from=1 to=$userscount step=1 do={
:set twodowntmp ($twodowntmp . [/ip firewall mangle get [/ip firewall mangle find
comment=[:pick $users ($dcount-1)] . "_down"] bytes] . ",");
:set twouptmp ($twouptmp . [/ip firewall mangle get [/ip firewall mangle find
comment=[:pick $users ($dcount-1)] . "_up"] bytes] . ",");
};

:if ( $stopmin > $startmin) do={
:set timedelay (($stopmin-$startmin) * 60);
};
:set timedelay (($timedelay+$stopsec)-$startsec);

:local firstdown [:toarray $firstdowntmp];
:local firstup [:toarray $firstuptmp];
:local twodown [:toarray $twodowntmp];
:local twoup [:toarray $twouptmp];

:global dcount ("1");
:for dcount from=1 to=$userscount step=1 do={

:if ( ($ActiveThresholddown * $timedelay) < ([:pick $twodown ($dcount-1)] - [:pick
$firstdown ($dcount-1)]) ) do={
:set activedownuserstmp ($activedownuserstmp . [:pick $users ($dcount-1)] . ",");
:set activedowncount ($activedowncount+1);
};
```

```
:if ( ($ActiveThresholdup * $timedelay) < ([:pick $twoup ($dcount-1)] - [:pick
$firstup ($dcount-1)]) ) do={
:set activeupuserstmp ($activeupuserstmp . [:pick $users ($dcount-1)] . ",");
:set activeupcount ($activeupcount+1);
};

};

:local activedownusers [:toarray $activedownuserstmp];
:local activeupusers [:toarray $activeupuserstmp];

:local maxlimitdown ("0");
:local maxlimitup ("0");

:if ( $activedowncount > 0 ) do={
:set maxlimitdown ($MaxRateDownload/$activedowncount);
:global dcount ("1");
:for dcount from=1 to=$activedowncount step=1 do={
:if ([/queue tree get [find name=[:pick $activedownusers ($dcount-1)] . "_down"]
max-limit] != $maxlimitdown) do={
/queue tree set [/queue tree find name=[:pick $activedownusers ($dcount-1)] .
"_down"] max-limit=$maxlimitdown; };
};
};

:if ( $activeupcount > 0 ) do={
:set maxlimitup ($MaxRateUpload/$activeupcount);
:global dcount ("1");
:for dcount from=1 to=$activeupcount step=1 do={
```

```
:if ([/queue tree get [find name=[:pick $activeupusers ($dcount-1)] . "_up"] max-limit] != $maxlimitup) do={
/queue tree set [/queue tree find name=[:pick $activeupusers ($dcount-1)] . "_up"] max-limit=$maxlimitup; };
};
};

:local kbsmaxdown ($MaxRateDownload/1000);
:local kbsmaxup ($MaxRateUpload /1000);

:if ( $maxlimitdown = 0 ) do={ :set maxlimitdown ($MaxRateDownload); };
:if ( $maxlimitup = 0 ) do={ :set maxlimitup ($MaxRateUpload); };
:local kbsmaxlimitdown ($maxlimitdown/1024);
:local kbsmaxlimitup ($maxlimitup/1024);

:set scriptstopmin ([: pick [/system clock get time] 3 5]);
:set scriptstopsec ([: pick [/system clock get time] 6 8]);

:if ( $scriptstopmin > $scriptstartmin) do={
:set scripttimedelay (($scriptstopmin-$scriptstartmin) * 60);
};
:set scripttimedelay (($scripttimedelay+$scriptstopsec)-$scriptstartsec);
##########################################################

##########################################################
:log info ("----------------------------------------");
:log warning ("Shaper:");
:log info ("MaxRate Download : " . $MaxRateDownload . " bps /" . $kbsmaxdown . " kbs / Upload : " . $MaxRateUpload . " bps /" . $kbsmaxup . " kbs");
:log info ("Active Users : Download : " . $activedowncount . " / Upload : " .
```

```
$activeupcount );
:log info ("User Speed Download : " . $maxlimitdown . " bps /" . $kbsmaxlimitdown .
" kbs / Upload : " . $maxlimitdown . " bps /" . $kbsmaxlimitup . " kbs" );
:log warning ("Performance Time: " . $scripttimedelay . " seconds.");
:log info ("----------------------------------------");
##########################################################
```