ЛАБОРАТОРНА РОБОТА 1
БАЗОВЕ НАЛАШТУВАННЯ МАРШРУТИЗАТОРА ЗА ДОПОМОГОЮ ІНТЕРФЕЙСУ КОМАНДНОГО РЯДКА

Мета роботи

- Використання інтерфейсу командного рядка для виконання базових налаштувань маршрутизатора.
- Перевірка конфігурацій і підключення.

Короткі теоретичні відомості

Невелика компанія розширила свій офіс за рахунок додаткового приміщення в іншій будівлі. Ви повинні налаштувати маршрутизатори таким чином, щоб забезпечити трафік між двома мережами.

Крок 1. Налаштування імен вузлів

а.

- Задайте ім'я вузла для маршрутизатора головного офісу MainOffice.
- В інтерфейсі командного рядка введіть наступні команди.

Router>**enable**
Router#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#**hostname MainOffice**
MainOffice(config)# *Use the key sequence***cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
MainOffice#**copy running-config startup-config**

б.

- Задайте ім'я вузла на маршрутизаторі Rmt_Site1.
- Виберіть маршрутизатор Rmt_Site1.
- В інтерфейсі командного рядка введіть наступні команди:

Router>**enable**
Router#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#**hostname Rmt_Site1**
Rmt_Site1(config)# *Use the key sequence***cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
Rmt_Site1#**copy running-config startup-config**

Крок 2. Налаштування інтерфейсів маршрутизатора

а. Налаштуйте послідовний інтерфейс маршрутизатора MainOffice.

- Виберіть маршрутизатор MainOffice.
- В інтерфейсі командного рядка введіть наступні команди.

MainOffice#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
MainOffice(config)#**interface serial0/1/0**
MainOffice(config-if)#**ip address 192.168.1.1 255.255.255.252**
MainOffice(config-if)#**clock rate 64000**
MainOffice(config-if)#**no shutdown**
MainOffice(config-if)#**exit**

б. Налаштуйте інтерфейс FastEthernet на маршрутизаторі MainOffice.

- В інтерфейсі командного рядка введіть наступні команди.

MainOffice(config)#**interface fastethernet0/0**
MainOffice(config-if)#**ip address 192.168.2.1 255.255.255.0**
MainOffice(config-if)#**no shutdown**
MainOffice(config-if)# *Use the key sequence***cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
MainOffice#**copy running-config startup-config**

в. Налаштуйте послідовний інтерфейс на маршрутизаторі Rmt_Site1.

- Виберіть маршрутизатор Rmt_Site1.
- В інтерфейсі командного рядка введіть наступні команди.

Rmt_Site1#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Rmt_Site1(config)#**interface serial0/1/0**
Rmt_Site1(config-if)#**ip address 192.168.1.2 255.255.255.252**
Rmt_Site1(config-if)#**no shutdown**
Rmt_Site1(config-if)#**exit**

г. Настройте інтерфейс FastEthernet на маршрутизаторі Rmt_Site1.

- В інтерфейсі командного рядка введіть наступні команди.

Rmt_Site1(config)#**interface fastethernet0/0**
Rmt_Site1(config-if)#**ip address 192.168.3.1 255.255.255.0**
Rmt_Site1(config-if)#**no shutdown**
Rmt_Site1(config-if)# *Use the key sequence***cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
Rmt_Site1#**copy running-config startup-config**

Крок 3. Налаштування протоколу маршрутизації RIP
а. Налаштуйте протокол RIPv2 на маршрутизаторі MainOffice.

- Виберіть маршрутизатор MainOffice.
- В інтерфейсі командного рядка введіть наступні команди.

MainOffice#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
MainOffice(config)#**router rip**
MainOffice(config-router)#**version 2**
MainOffice(config-router)#**network 192.168.1.0**
MainOffice(config-router)#**network 192.168.2.0**
MainOffice(config-router)# *Use the key sequence* **cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
MainOffice#**copy running-config startup-config**

б. Налаштуйте протокол RIPv2 на маршрутизаторі Rmt_Site1.

- Виберіть маршрутизатор Rmt_Site1.
- В інтерфейсі командного рядка введіть наступні команди.

Rmt_Site1#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Rmt_Site1(config)#**router rip**
Rmt_Site1(config-router)#**version 2**
Rmt_Site1(config-router)#**network 192.168.1.0**
Rmt_Site1(config-router)#**network 192.168.3.0**
Rmt_Site1(config-router)# *Use the key sequence* **cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
Rmt_Site1#**copy running-config startup-config.**

Крок 4. Налаштування паролів привілейованого режиму, консолі і віртуального терміналу

а. Виберіть маршрутизатор MainOffice.

- Перейдіть в режим глобальної конфігурації.
- Введіть в інтерфейсі командної строки секретний пароль привілейованого режиму, пароль консолі і пароль telnet, використовуючи наступні команди:

MainOffice#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
MainOffice(config)#**enable secret cisco123**
MainOffice(config)#**line console 0**
MainOffice(config-line)#**password class**
MainOffice(config-line)#**login**
MainOffice(config-line)#**exit**
MainOffice(config)#**line vty 0 4**
MainOffice(config-line)#**password class**
MainOffice(config-line)#**login**
MainOffice(config-line)# *Use the key sequence* **cntl + z** *here*
%SYS-5-CONFIG_I: Configured from console by console
MainOffice#**copy running-config startup-config**

б. Виберіть маршрутизатор Rmt_Site1.

- Перейдіть в режим глобальної конфігурації.
- Введіть в інтерфейсі командної строки секретний пароль привілейованого режиму, пароль консолі і пароль telnet, використовуючи наступні команди:

Rmt_Site1#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Rmt_Site1(config)#**enable secret cisco123**
Rmt_Site1(config)#**line console 0**
Rmt_Site1(config-line)#**password class**
Rmt_Site1(config-line)#**login**
Rmt_Site1(config-line)#**exit**
Rmt_Site1(config)#**line vty 0 4**
Rmt_Site1(config-line)#**password class**
Rmt_Site1(config-line)#**login**
Rmt_Site1(config-line)# *Use the key sequence* **cntl + z** *here*

%SYS-5-CONFIG_I: Configured from console by console
Rmt_Site1#**copy running-config startup-config**

Крок 5. Перевірка конфігурацій і можливості підключення

а. Виведіть поточну конфігурацію маршрутизатора MainOffice за допомогою команди show running-config..
б. Знайдіть ім'я вузла, паролі, IP-адресу та конфігурації протоколу маршрутизації.
в. Виведіть поточну конфігурацію маршрутизатора Rmt_Site1 за допомогою команди show running-config.
г. Знайдіть ім'я вузла, паролі, IP-адресу та конфігурації протоколу маршрутизації.
д. Виконайте луна-тестування PC1 з командного строкіPC0**:**

PC>**ping 192.168.3.3**
Простежте мережевий шлях від PC0 до PC1 за допомогою командного рядка на PC0:
PC>**tracert 192.168.3.3**
Натисніть кнопку Check Results (Перевірити результати).

Контрольні питання

а. Які команди використовуються для входу в режим конфігурації інтерфейсу FastEthernet 0/0, якщо ви починаєте роботу в режимі користувача EXEC?
б. Для настройки будь інтерфейсів необхідно використовувати команду "clock rate"? (DCE або DTE)

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008
2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.
3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.
4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.
5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.
6. CISCO Internetworking technology overview – Cisco, 1999.

ЛАБОРАТОРНА РОБОТА 2
ЗАСТОСУВАННЯ ПРИНЦИПІВ, ЗАКЛАДЕНИХ У ТАБЛИЦІ МАРШРУТИЗАЦІЇ

Мета роботи: Розібрати на практиці три важливих принципи маршрутизації.

1. Маршрутизатор приймає рішення на основі інформації, наявної в таблиці маршрутизації.
2. Якщо один маршрутизатор має повну таблицю маршрутизації, це ще не означає, що всі інші маршрутизатори володіють такою ж інформацією.
3. Інформація про маршрут з однієї мережі в іншу не містить відомостей про зворотному шляху (або шляхи повернення).

Короткі теоретичні відомості

CIDR - безкласова міждоменна маршрутизація

Навіть при ефективному виділення IP-адрес проблема розростання мережі зберігається. Якщо маршрутизатор знаходиться на межі мережі будь-якої організації (наприклад, університету), він повинен зберігати інформацію про всі підмережі, щоб знати, по якій лінії слід передавати пакети для цієї мережі. Якщо адреса призначення знаходиться за межами даної організації, він може використовувати просте правило за замовчуванням: відправляти пакети по лінії, що з'єднує цю організацію з іншою мережею Інтернет. Решта адреси призначення, очевидно, знаходяться поблизу.

Маршрутизатори інтернет-провайдерів і магістралей не можуть дозволити собі таку розкіш. Вони повинні знати шлях до будь-якої мережі, тому для них не може існувати просте правила за замовчуванням. Про такі магістральні маршрутизатори кажуть, що вони знаходяться у вільній від замовчувань зоні (default-free zone) мережі Інтернет. Ніхто не знає точно, скільки всього мереж підключено до Інтернету, але очевидно, що їх багато - можливо, близько мільйона. З них можна скласти дуже велику таблицю. Може бути, і не дуже велику з точки зору комп'ютерних стандартів, але уявіть собі, що маршрутизатор повинен проглядати її при відправці кожного пакету, а за секунду він відправляє мільйони таких пакетів. Для обробки пакетів з такою швидкістю потрібні спеціалізовані апаратні засоби і швидкодіюча пам'ять; звичайний комп'ютер для цього не підійде.

Різні алгоритми маршрутизації вимагають, щоб кожен маршрутизатор про-

менівался інформацією про доступні йому адресах з іншими маршрутизаторами. Чим більше розмір таблиці, тим більше даних необхідно передавати і обробляти. З ростом розміру таблиці час обробки зростає як мінімум лінійно. чим більше даних доводиться передавати, тим вище ймовірність втрати (в кращому випадку часової) частини інформації по дорозі, що може призвести до нестабільності роботи алгоритмів вибору маршрутів.

Проблема таблиць маршрутизаторів може бути вирішена за допомогою збільшення числа рівнів ієрархії, як це відбувається в телефонних мережах. Наприклад, якби кожен IP-адреса містив поля країни, штату або провінції, міста, мережі і номери хоста. У такому випадку, кожному маршрутизатору потрібно буде знати, як дістатися до кожної країни, до кожного штату або провінції своєї країни, кожного міста своєї провінції або штату і до кожної мережі свого міста. На жаль, такий підхід вимагає істотно більше 32 біт для адреси, а адресне поле буде використовуватися неефективно (для князівства Ліхтенштейн буде виділено стільки ж розрядів, скільки для Сполучених Штатів).

На щастя, спосіб зменшити розмір таблиць маршрутизації все ж існує.

Застосуємо той же принцип, що і при розбитті на підмережі: маршрутизатор може дізнаватися про розташування IP-адрес по префіксам різної довжини. Але замість того щоб розділяти мережу на підмережі, ми об'єднаємо кілька коротких префіксів в один довгий. Цей процес називається агрегацією маршруту (route aggregation).

Довгий префікс, отриманий в результаті, іноді називають суперсетами (supernet), на противагу подсетям з поділом блоків адрес.

При агрегації IP-адреси містяться в префіксах різної довжини. Один і той ж IP-адреса може розглядатися одним маршрутизатором як частина блоку / 22 (містить 210 адрес), а іншим - як частина більш великого блоку / 20 (містить 212 адрес). Це залежить від того, якою інформацією володіє маршрутизатор. Такий метод працює і для розбиття на підмережі і називається CIDR (Classless InterDomain Routing - безкласова міждоменна маршрутизація). Остання на сьогоднішній день версія описана в RFC 4632 (Fuller і Li, 2006). Назва ілюструє відміну від адрес, які кодують ієрархію за допомогою класів, про яку ми незабаром поговоримо.

Щоб краще зрозуміти алгоритм маршрутизації, розглянемо приклад. Припустимо, у нас є блок з 8192 адрес, починаючи з 194.24.0.0. Припустимо також, що Кембріджському університету потрібно 2048 адрес і йому виділяються адреси від 194.24.0.0 до 194.24.7.255, а також маска 255.255.248.0. Це буде префікс / 21. Потім Оксфордський університет запрошувати 4096 адрес. Так як

блок з 4096 адрес повинен розташовуватися на межі, кратній 4096, то йому не можуть бути виділені адреси, що починаються з 194.24.8.0. Замість цього він отримує адреси від 194.24.16.0 до 194.24.31.255 разом з маскою 255.255.240.0. Нарешті, Единбурзький університет просить виділити йому 1024 адреси і отримує адреси від 194.24.8.0 до 194.24.11.255 і маску 255.255.252.0_ Всі ці присвоєні адреси і маски зведені в табл. 1.

Таб.1 - Набір присвоєних IP-адрес

Університет	Перша адреса	Остання адреса	Кількість	Форма запису
Кембридж	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Эдинбург	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Свободно)	194.24.12.0	194.24.15.255	1024	194.24.12.0/22
Оксфорд	194.24.16.0	194.24.16.255	4096	194.24.16.0/20

Після цього всім маршрутизаторам, які перебувають у вільній від замовчувань зоні, повідомляються IP-адреси трьох нових мереж. Маршрутизатори, що знаходяться поруч з цими університетами (наприклад, в Лондоні - рис. 1), можливо, захочуть відправляти пакети на ці префікси за різними походить лініях. Тоді вони запишуть ці адреси у свої таблиці маршрутизації.

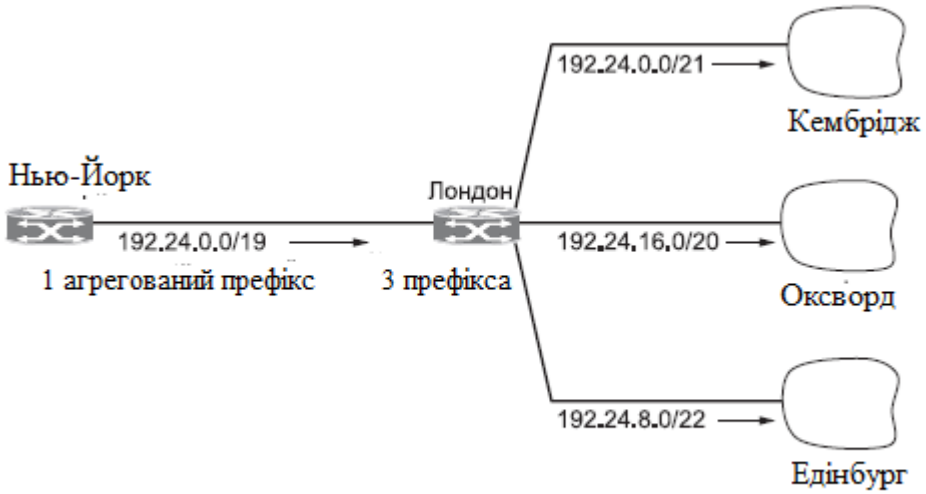


Рис.1 - агрегація IP - префіксів

Тепер подивимося на цю трійцю університетів з точки зору віддаленого маршрутизатора в Нью-Йорку. Всі IP-адреси, що відносяться до цих трьох префіксам, повинні відправлятися з Нью-Йорка (або зі США) в Лондон.

Процес маршрутизації в Лондоні дізнається про це і об'єднує три префікса в одну агрегированную запис 194.24.0.0/19 і передає її в Нью-Йорк. Цей префікс містить 8 Кбайт адрес і об'єднує три університети плюс 1024 вільних адреси.

Агрегація дозволила об'єднати три префікса в один, завдяки чому зменшилася кількість префіксів, про які повинен знати маршрутизатор в Нью-Йорку, і кількість записів в його таблиці маршрутизації.

Якщо агрегація увімкнена, вона проводиться автоматично. Цей процес залежить від того, де які префікси розташовані, а не від адміністратора, який виділяє мережам адреси. В Інтернеті агрегація використовується дуже активно, знижуючи розмір таблиць маршрутизації приблизно до 200 тисяч префіксів.

Далі все стає ще цікавіше: префікси можуть перетинатися. Згідно з правилом, пакети передаються в напрямку самого спеціалізованого блоку, або найдовшого збігається префікса (longest matching prefix), в якому знаходиться найменше IP-адрес. Поведінка маршрутизатора в Нью-Йорку (рис. 2) показує, наскільки гнучкою є такий тип маршрутизації. для відправки пакетів в наші три університету Нью-Йоркський маршрутизатор використовує один агрегований префікс. Але що робити, якщо той блок адрес, який раніше був вільним, тепер належить мережі в Сан-Франциско? Наприклад, Нью-Йоркський маршрутизатор може зберігати чотири префікса: один для Сан-Франциско і три для Лондона. Маршрутизація по найдовшому збігається префіксу дозволяє обійтися двома (рис. 2). Один загальний префікс використовується для того, щоб направляти трафік, призначений для всього блоку, в Лондон. Ще один специфічний префікс дозволяє направляти його частина в Сан-Франциско. Відповідно до правила найдовшого збігається префікса, пакети, призначені для IP-адрес в Сан-Франциско, будуть передані по вихідної лінії, що веде в Сан-Франциско. Пакети, призначені для більш великої мережі, будуть спрямовані в Лондон.

По суті CIDR працює таким чином. Коли прибуває пакет, необхідно визначити, чи відноситься дана адреса до даного префікса; для цього проглядається таблиця маршрутизації. Може виявитися, що за значенням підійде кілька записів. У цьому випадку використовується найдовший префікс. Тобто якщо знайдено збіг для маски / 20 і / 24, то для вибору вихідної лінії буде використовуватися запис, відповідна / 24. Однак цей процес був би трудомістким, якби таблиця маршрутизації проглядалася запис за записом. Замість цього був розроблений складний алгоритм для прискорення процесу пошуку адреси в таблиці (Ruiz-Sanchez та ін., 2001). У маршрутизаторах, що припускають комерційне використання, застосовуються спеціальні чіпи VLSI, в які дані алгоритми вбудовані апаратно.

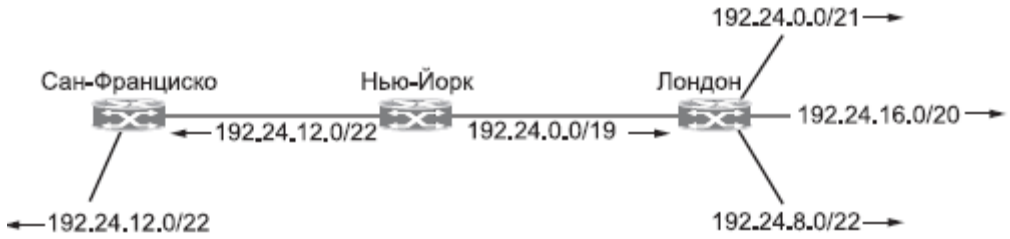


Рис.2 - Маршрутизація по самому довгому співпадаючому префіксу на Нью-Йоркському маршрутизаторі

Завдання на лабораторну роботу

Попередні знання / підготовка

Пакети прямують через мережу послідовно від одного маршрутизатора до іншого. Кожен маршрутизатор приймає незалежне рішення про передачу на підставі наявної у нього інформації про шляхи до одержувача. Хоча пакети можуть успішно досягати мережі одержувача, зворотний шлях може бути невідомий маршрутизатора одержувача. У таких випадках маршрутизатор не може направити трафік назад до відправника. Ця ситуація відома під назвою "чорна діра" маршрутизації.

Таблиця адрес

ПРИСТРІЙ	ІНТЕРФЕЙС	IP-адреса	Маска підмережі	Основний шлюз
R1	Fa0/0	192.168.1.1	255.255.255.0	-
	Fa0/1	192.168.2.1	255.255.255.0	-
R2	Fa0/0	192.168.2.2	255.255.255.0	-
	S0/0/0	192.168.7.1	255.255.255.0	-
	S0/0/1	192.168.3.1	255.255.255.0	-
R3	Fa0/0	192.168.4.1	255.255.255.0	-
	S0/0/0	192.168.5.1	255.255.255.0	-
	S0/0/1	192.168.3.2	255.255.255.0	-
R4	Fa0/0	192.168.6.1	255.255.255.0	-
	S0/0/0	192.168.7.2	255.255.255.0	-
	S0/0/1	192.168.5.2	255.255.255.0	-
PC1	NIC	192.168.1.10	255.255.255.0	192.168.1.1
PC2	NIC	192.168.1.20	255.255.255.0	192.168.1.1
PC3	NIC	192.168.4.10	255.255.255.0	192.168.4.1
PC4	NIC	192.168.6.10	255.255.255.0	192.168.6.1

Крок 1. Визначте, чому PC1 не може успішно відправити ехо-запит PC3.

а. Надішліть ехо-запит з PC1 на PC3. Зверніть увагу, що луна-запит невдалий.

б. Використовуйте команду show ip route, щоб перевірити таблицю маршрутизації на R1 з метою виявлення проблеми.

в. Чи бачите ви шлях до 192.168.4.0 в таблиці маршрутизації?

г. Введіть статичний маршрут на R1 для мережі одержувача 192.168.4.0.

R1 # configure terminal

R1 (config) #ip route 192.168.4.0 255.255.255.0 192.168.2.2

R1 (config) #end

д. Використовуйте команду show ip route для перевірки таблиці маршрутизації на R1. Чи тепер в цій таблиці шлях до 192.168.4.0?

е. У командному рядку PC1 відправте ехо-запит на 192.168.4.10. Зверніть увагу, що ехо-запит невдалий.

Крок 2. Перегляньте ехо-запит з PC1 на PC3 в режимі моделювання.

а. Перейдіть з режиму реального часу в режим моделювання. Виберіть вкладку Simulation, що знаходиться за вкладкою реального часу в нижньому правому куті.

б. Відфільтруйте трафік так, щоб було видно тільки пакети ICMP. У режимі моделювання натисніть кнопку Edit Filters. Виберіть прапорець Show All / Noneдля скидання всіх прапорців і потім виберіть ICMP.

в. Виберіть пристрої відправник і одержувач для моделювання. Над значком режиму моделювання є два значка у вигляді конвертів. Виберіть конверт Add Simple PDU (P). Призначте PC1 відправником трафіку ICMP, клацнувши PC1 в робочій області. Призначте PC3 вузлом-одержувачем.

г. Увімкніть режим моделювання, натиснувши кнопку Auto Capture / Play. При цьому запускаються ехо-запити між PC з використанням ICMP.

д. Зверніть увагу, що R1 відсилає ICMP-трафік на R3. R3 відсилає ICMP-трафік на PC3. PC3 відповідає відправкою ICMP-трафіку назад на R3. Однак R3 відмовляється приймати пакети. Чому exo-запити на R3 невдалі?

е. Вийдіть з режиму Simulation, клацнувши значок режиму Realtime.

Крок 3. Виправте помилку маршрутизації на R3.

а. Оскільки R3 не повертає ICMP-трафік PC1, перевірте таблицю маршрутизації на R3.

б. Чи бачите ви в таблиці маршрутизації шлях для 192.168.1.0?

в. Введіть статичний маршрут на R3 для мережі одержувача 192.168.1.0.

R3 # configure terminal

R3 (config) #ip route 192.168.1.0 255.255.255.0 Serial 0/0/1

R3 (config) #end

г. Використовуйте команду show ip route для перевірки таблиці маршрутизації на R3. Чи тепер в цій таблиці шлях до 192.168.1.0?

д. У командному рядку PC1 відправте exo-запит на 192.168.4.10. Exo-тестування має пройти успішно. В іншому випадку ще раз перевірте виконані вами операції, знайдіть і усуньте помилку.

Крок 4. Перегляньте exo-запит з PC1 на PC3 в режимі моделювання.

а. Створіть новий сценарій для цього другого моделювання, встановивши прапорець New під Scenario 0. При цьому меню, що випадає зміниться на Scenario 1.

б. Відфільтруйте трафік так, щоб було видно тільки пакети ICMP. Відфільтруйте трафік так, щоб було видно тільки пакети ICMP. У режимі моделювання натисніть кнопку Edit Filters. Виберіть прапорець Show All / None для скидання всіх прапорців і потім виберітеICMP.

в. Виберіть пристрої відправник і одержувач для моделювання. Виберіть конверт Add Simple PDU (P). Призначте PC1 відправником ICMP-трафіку і PC3 - вузлом-одержувачем.

г. Увімкніть режим моделювання, натиснувши кнопку Auto Capture / Play. При цьому запускаються exo-запити між PC з використанням ICMP.

д. Зверніть увагу, що R1 відсилає в R3 трафік ICMP. R3 відсилає ICMP-трафік в PC3. PC3 відповідає відправкою ICMP-трафіку назад на R3. R3 повертає ICMP-трафік в R1. R1 направляє відповідь в PC1. Помилки маршрутизації усунені.

е. Вийдіть з режиму моделювання, клацнувши значок режиму Realtime.

ж. Виберіть вкладку Check Results (Перевірити результати), щоб переконатися в правильному виконанні вправи.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008
2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.
3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.
4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.
5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.
6. CISCO Internetworking technology overview – Cisco, 1999.

ЛАБОРАТОРНА РОБОТА 3
НАЛАШТУВАННЯ МАРШРУТИЗАТОРУ ЗА ЗАМОВЧУВАННЯМ

Мета роботи: отримати практичні навички формування маршрутів у мережі та налагодження маршрутизаторів.

Короткі теоретичні відомості

Щоб направити пакети адресовані в мережі, які явно не вказані в таблиці маршрутизації використовується маршрут за замовчуванням (Default route). Маршрути за замовчуванням доступні в топологиях де не бажано вивчення більш специфічних мереж, як у випадку, кінцевих тупикових мереж (stub network) або коли кількість системних ресурсів обмежена, наприклад немає достатньо оперативної пам'яті, щоб прийняти всі маршрути які існують в світі. Даний документ пояснює як налаштувати маршрут за замовчуванням або шлюз останнього доступу за допомогою IP команд

Щоб направити пакети адресовані в мережі, які явно не вказані в таблиці маршрутизації використовується маршрут за замовчуванням (Default route). Маршрути за замовчуванням доступні в топологиях де не бажано вивчення більш специфічних мереж, як у випадку, кінцевих тупикових мереж (stub network) або коли кількість системних ресурсів обмежена, наприклад немає достатньо оперативної пам'яті, щоб прийняти всі маршрути які існують в світі.

Даний документ пояснює як налаштувати маршрут за замовчуванням або шлюз останнього доступу за допомогою наступних IP команд:

• ip default-gateway
• ip default-network
• ip route 0.0.0.0 0.0.0.0

ip default-gateway

Команда ip default-gateway відрізняється від двох інших команд. Вона повинна використовуватися тільки у випадках коли ip routing вимкнений на Cisco маршрутизаторе. Наприклад, якщо маршрутизатор це хост у світі IP, ви можете

використовувати дану команду, щоб визначити маршрут за замовчуванням для нього. Ви можете також використовувати цю комнади, коли ваш маршрутизатор перебувати в режимі завантаження (boot mode) для того, щоб завантажити IOS по протоколу TFTP. У режимі boot у роутера вимкнений ip routing.

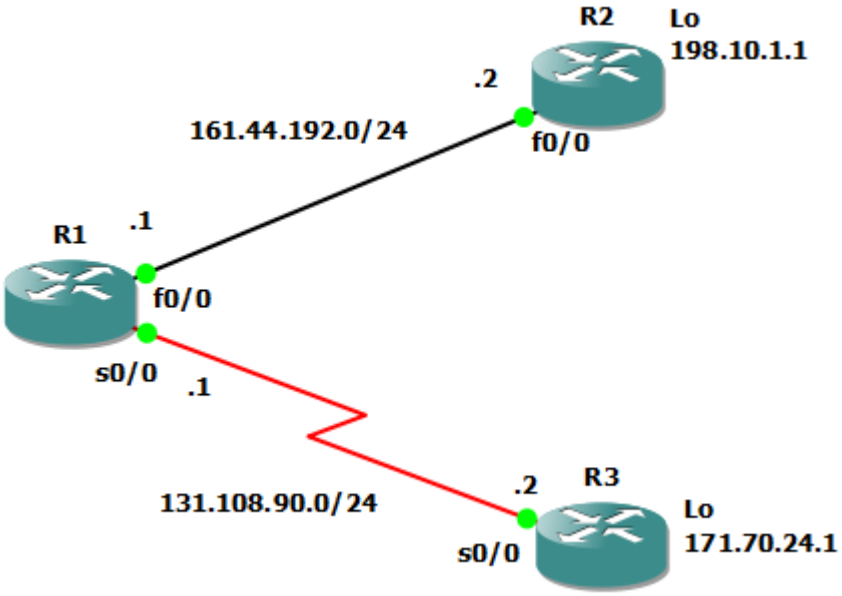
Наступний приклад визначає роутер з IP адресою 172.16.15.4 як маршрут за замовчуванням

```
ip default-gateway 172.16.15.4
```

ip default-network

На відміну від команди ip default-gateway, ви можете використовувати команду ip default-network коли на роутері увімкнена маршрутизація. Коли ви налаштовуєте ip default-network маршрутизатор розглядає маршрути на таку мережу як шлюз останнього вибору. Для кожної мережі налаштованої за допомогою команди ip default-network, якщо у роутера є маршрут на таку мережу, цей маршрут позначається як кандидат в маршрут за замовчуванням.

Нижченаведена мережева діаграма показує таблицю маршрутизації взяту з роутера R1



Зауважимо, що статичний маршрут на мережу 198.10.1.0 йде через 161.44.192.2 і що шлюз за замовчуванням не встановлений.

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     161.44.0.0/24 is subnetted, 1 subnets
C       161.44.192.0 is directly connected, FastEthernet0/0
     131.108.0.0/24 is subnetted, 1 subnets
C       131.108.90.0 is directly connected, Serial0/0
S    198.10.1.0/24 [1/0] via 161.44.192.2
R1#
R1#
R1#
```

Якщо ми сконфігуріруемо команду ip default-network 198.10.1.0, таблиця маршрутизації змінитися на таку

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 161.44.192.2 to network 198.10.1.0

     161.44.0.0/24 is subnetted, 1 subnets
C       161.44.192.0 is directly connected, FastEthernet0/0
     131.108.0.0/24 is subnetted, 1 subnets
C       131.108.90.0 is directly connected, Serial0/0
S*   198.10.1.0/24 [1/0] via 161.44.192.2
R1#
R1#
```

Тепер шлюз останнього вибору встановлений на 161.44.192.2. Цей результат незалежний від будь-якого протоколу маршрутизації, як показано у висновку команди show ip route вище. Ви можете додати інший кандидат в маршрут за замовчуванням, налаштувавши інший примірник ip default-network

```
2513(config)#ip route 171.70.24.0 255.255.255.0 131.108.90.2
2513(config)#ip default-network 171.70.24.0
```

```
R1#
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 161.44.192.2 to network 198.10.1.0

     171.70.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       171.70.0.0/16 [1/0] via 171.70.24.0
S       171.70.24.0/24 [1/0] via 131.108.90.2
     161.44.0.0/24 is subnetted, 1 subnets
C       161.44.192.0 is directly connected, FastEthernet0/0
     131.108.0.0/24 is subnetted, 1 subnets
C       131.108.90.0 is directly connected, Serial0/0
S*   198.10.1.0/24 [1/0] via 161.44.192.2
R1#
```

Однак, після того як була введена команда ip default-network у виведенні вище, мережа не була позначена як мережа за замовчуванням. Наступний розділ пояснює чому.

Позначення мережі за замовчуванням

Пам'ятайте, що команда ip default-network є класової (classfull). Це означає, що маршрутизатор має маршрут на підмережа зазначену в цій команді, він інсталює маршрут на основну мережу. З цього місця ніяка мережа не позначається як мережу за замовчуванням. Команду ip default-network необхідно ввести ще раз, використовуючи основну мережу, для того, щоб позначити кандидата в маршрут за замовчуванням.

```
R1(config)#ip default-network 171.70.0.0
```

```
R1#
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 171.70.24.0 to network 171.70.0.0

 *   171.70.0.0/16 is variably subnetted, 2 subnets, 2 masks
S*      171.70.0.0/16 [1/0] via 171.70.24.0
S       171.70.24.0/24 [1/0] via 131.108.90.2
     161.44.0.0/24 is subnetted, 1 subnets
C       161.44.192.0 is directly connected, FastEthernet0/0
     131.108.0.0/24 is subnetted, 1 subnets
C       131.108.90.0 is directly connected, Serial0/0
S*   198.10.1.0/24 [1/0] via 161.44.192.2
R1#
R1#
R1#
```

Якщо ж первинний статичний маршрут вказував на основну мережу, то немає необхідності двічі вводити це команду.

Тут все ще немає працюючих IP протоколів. У відсутність працюючих динамічних протоколів ви можете налаштувати маршрутизатор, щоб він вибирав маршрут за замовчуванням з ряду кандидатів, грунтуючись на тому, чи має таблиця маршрутизації маршрут на мережі відмінні від 0.0.0.0/0. Ви можете зробити так, щоб маршрутизатор вибирав маршрут за замовчуванням на певну мережу перевіряючи свою роутінговую таблицю, а, що не конфігуруємо статично маршрути на зазначені next-hop.
Якщо ви втратили маршрут на певну мережу, маршрутизатор вибере інший кандидат в маршрут за замовчуванням. У наступному прикладі ми видаляємо статичний маршрут:

```
R1(config)#no ip route 171.70.24.0 255.255.255.0 131.108.90.2
```

Після видалення маршруту на мережу за замовчуванням, таблиця маршрутизації виглядає так:

```
R1#
R1#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 161.44.192.2 to network 198.10.1.0

     161.44.0.0/24 is subnetted, 1 subnets
C       161.44.192.0 is directly connected, FastEthernet0/0
     131.108.0.0/24 is subnetted, 1 subnets
C       131.108.90.0 is directly connected, Serial0/0
S*   198.10.1.0/24 [1/0] via 161.44.192.2
R1#
R1#
```

ip route 0.0.0.0 0.0.0.0

Створення статичного маршруту на мережу 0.0.0.0 це інший спосіб встановити шлюз останнього вибору для маршрутизатора. Однак, на маршрутизаторі потрібно включити ip routing.

Зауважимо, що протокол маршрутизації IGRP не розуміє маршрут 0.0.0.0. Тому він не може мовити маршрут за замовчуванням створений за допомогою команди ip route 0.0.0.0 0.0.0.0. Натомість використовуйте команду ip default-network.
EIGRP може мовити маршрут на мережу 0.0.0.0, але статичний маршрут бути вприснуто в протокол маршрутизації командою redistribute.

Маршрут за замовчуванням створений командою ip route 0.0.0.0 0.0.0.0 не поширюється протоколами маршрутизації OSPF і IS-IS. Додатково, маршрут за замовчуванням не може бути вприснуто в OSPF або IS-IS використовуючи команду redistribute. Для генерації маршруту за замовчуванням в роутінговом домені OSPF або IS-IS використовуйте команду default-information originate.
У наступному прикладі ми налаштовуємо шлюз останнього вибору:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 131.108.90.2
```

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 131.108.90.2 to network 0.0.0.0

     161.44.0.0/24 is subnetted, 1 subnets
C       161.44.192.0 is directly connected, FastEthernet0/0
     131.108.0.0/24 is subnetted, 1 subnets
C       131.108.90.0 is directly connected, Serial0/0
S    198.10.1.0/24 [1/0] via 161.44.192.2
S*   0.0.0.0/0 [1/0] via 131.108.90.2
R1#
```

Якщо ви налаштовуєте кілька мереж як кандидати в маршрути за замовчуванням, використовуючи команду ip default-network, мережа яка має найменшу адміністративну дистанцію вибирається як мережа для шлюзу останнього вибору. Якщо мережі мають одну і ту ж саму адміністративну дистанцію, тоді перша за списком мережу в таблиці маршрутизації (show ip route) вибирається як мережа для шлюзу останнього вибору.

Якщо ви використовуєте обидві команди ip default-network і ip route 0.0.0.0 0.0.0.0 для конфігурації кандидатів у мережі за замовчуванням, і мережа використовувана в команді ip default-network відома статично, то мережа визначена в команді ip default-network мати перевагу і вибирається як мережа для шлюзу останнього вибору.

Якщо ж мережа використовувана в команді ip default-network доставляється протоколом маршрутизації, то команда ip route 0.0.0.0 0.0.0.0, яка має меншу адміністративну дистанцію, має перевагу.

Якщо ви використовуєте кілька команд ip route 0.0.0.0 0.0.0.0 для конфігурації маршрутів за замовчуванням, трафік буде балансуватися по декількох маршрутах.

Завдання на лабораторну роботу
Початкові дані

Необхідно налаштувати маршрут за замовчуванням для клієнтського маршрутизатора Cisco 1841. При налаштуванні маршруту за замовчуванням використовується IP-адреса глобальної мережі на маршрутизаторі постачальника послуг Інтернету Cisco 1841. Це маршрутизатор наступного переходу після клієнтського маршрутизатора Cisco 1841.

Крок 1. Перевірка доступності IP-адреси локальної мережі маршрутизатора ISP з клієнтського маршрутизатора.

а. Для підключення до клієнтського маршрутизатора Cisco 1841 скористайтеся програмним забезпеченням емуляції терміналу на комп'ютері CustomerPC. В якості пароля консолі використовуйте cisco123.

б. С допомогою команди ping перевірте, чи доступний IP-адреса локальної мережі 209.165.201.1 на маршрутизаторі ISP з маршрутизатора CustomerRouter:

```
CustomerRouter> ping 209.165.201.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Крок 2. Налаштування маршруту за замовчуванням.

а. Увійдіть в привілейований режим EXEC за допомогою пароля cisco. Запит CustomerRouter # вказує, що ви перебуваєте в привілейованому режимі EXEC.

б. Увійдіть в режим глобальної налаштування. Запит CustomerRouter (config) # вказує, що ви перебуваєте в режимі глобальної налаштування.

в. Налаштуйте маршрут за замовчуванням, використовуючи IP-адреса ISP WAN в якості IP-адреси наступного переходу:

```
CustomerRouter (config) #ip route 0.0.0.0 0.0.0.0 209.165.200.226
CustomerRouter (config) #end
```

Крок 3. Перевірка настройки маршруту за замовчуванням.

а. За допомогою команди show ip route перевірте настройку маршруту за замовчуванням. Ось частина її вихідних даних:

```
CustomerRouter # show ip route
Codes: C - connected, S - static, ...
```

<вихідні дані опущені>

Gateway of last resort is 209.165.200.226 to network 0.0.0.0

C 192.168.1.0/24 is directly connected, FastEthernet0 / 0
209.165.200.0/27 is subnetted, 1 subnets
C 209.165.200.224 is directly connected, Serial0 / 1/0
S * 0.0.0.0/0 [1/0] via 209.165.200.226

б. Для перевірки підключення до IP-адресою локальної мережі з маршрутизатора ISP скористайтеся командою ping:

CustomerRouter # ping 209.165.201.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min / avg / max = 22/25/34 ms

Крок 4. Збереження конфігурації.

а. У привілейованому режимі EXEC збережіть поточну конфігурацію в якості початкової:

CustomerRouter # copy run start

б. Для перевірки зробленої роботи натисніть кнопку Check Results (Перевірити результати) у нижній частині вікна інструкцій.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008
2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.
3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.
4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.

5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.

6. CISCO Internetworking technology overview – Cisco, 1999.

ЛАБОРАТОРНА РОБОТА 4
НАЛАШТУВАННЯ ПРОТОКОЛУ RIP

Мета роботи: отримати практичні навички використання протоколів маршрутизації.

Короткі теоретичні відомості
Маршрутизація по вектору відстаней

Комп'ютерні мережі зазвичай використовують динамічні алгоритми маршрутизації, які є більш складними, ніж заливка, але в той же час і більш ефективними, оскільки вони дозволяють знаходити найкоротші шляхи для поточної топології.

Найбільшою популярністю користуються два динамічних методу: маршрутизація по вектору відстаней і маршрутизація з урахуванням стану каналів. У цьому розділі ми вивчимо перший, в наступному - другий метод.

Алгоритми маршрутизації по вектору відстаней (distance vector routing) працюють, спираючись на таблиці (тобто вектори), підтримувані усіма маршрутизаторами і містять відомості про найкоротших відомих шляхах до кожного з можливих адресатів і про те, яке з'єднання слід при цьому використовувати. Для поновлення даних цих таблиць проводиться обмін інформацією з сусідніми маршрутизаторами. В результаті маршрутизатор знає найкращий спосіб дістатися до будь-якої адреси призначення.

Алгоритм маршрутизації по вектору відстаней іноді називають по іменах його творців розподіленим алгоритмом Беллмана-Форда (Bellman-Ford) (Bellman, 1957; Ford і Filkerson, 1962). Цей алгоритм спочатку застосовувався в мережі ARPANET і в Інтернеті був відомий під назвою RIP.

При маршрутизації по вектору відстаней таблиці, з якими працюють і які оновлюються маршрутизаторами, містять записи про кожний маршрутизатор мережі. Кожен запис складається з двох частин: бажаний номер лінії для даного одержувача і передбачувана відстань або час проходження пакета до цього одержувача. В якості міри відстані можна використовувати число транзитних ділянок або інші одиниці виміру (про які ми говорили під час обговорення довжини найкоротшого шляху).

Передбачається, що маршрутизаторам відомо відстань до кожного з сусідів. Якщо в якості одиниці вимірювання використовується число транзитних ділянок, то відстань дорівнює одному транзитному ділянці. Якщо ж дистанція вимірюється часом затримки поширення, то маршрутизатор може виміряти його за допомогою спеціального пакету ECHO (exo), в який одержувач поміщає час отримання і який відправляє назад якомога швидше.

Припустимо, що в якості одиниці вимірювання використовується час затримки і цей параметр щодо кожного з сусідів відомий маршрутизатора. Через кожні T мс всі маршрутизатори посилають своїм сусідам список з приблизними затримками для кожного одержувача. Вони, зрозуміло, також отримують подібний список від усіх своїх сусідів. Припустимо, одна з таких таблиць прийшла від сусіда X і в ній вказується, що час поширення від маршрутизатора X до маршрутизатора i одно Xi. Якщо маршрутизатор знає, що час пересилки до маршрутизатора X одно m, тоді затримка при передачі пакета маршрутизатору i через маршрутизатор X складе Xi + m. Виконавши такі розрахунки для всіх своїх сусідів, маршрутизатор може вибрати найкращі шляхи і помістити відповідні записи в нову таблицю. Зверніть увагу, що стара таблиця в розрахунках не використовується.

Процес оновлення таблиці проілюстровано на рис. 1. На рис. 1, а показана мережу. Перші чотири стовпці на рис. 1, б показують вектори затримок, отримані маршрутизатором J від своїх сусідів. Маршрутизатор A вважає, що час пересилки від нього до маршрутизатора B дорівнює 12 мс, 25 мс до маршрутизатора C, 40 мс до D і т. Д. Припустимо, що маршрутизатор J виміряв або оцінив затримки до своїх сусідів A, I, H і K як 8, 10, 12 і 6 мс відповідно.

Тепер розглянемо, як J розраховує свій новий маршрут до маршрутизатору G. Він знає, що затримка до A становить 8 мс і при цьому A думає, що від нього до G дані дійдуть за 18 мс. Таким чином, J знає, що якщо він стане відправляти пакети для G через A, то затримка складе 26 мс. Аналогічно, він обчислює значення затримок для маршрутів від нього до G, що проходять через інших його сусідів (I, H і K), і отримує відповідно 41 (31 + 10), 18 (6 + 12) і 37 (31 + 6). Кращим значенням є 18, тому саме воно поміщається в таблицю в запис для одержувача G. Разом з числом 18 туди ж поміщається позначення лінії, по якій проходить найбільший короткий маршрут до G, тобто H. Даний метод повторюється для всіх інших

адресатів, і при цьому виходить нова таблиця, показана у вигляді правого стовпчика на малюнку.

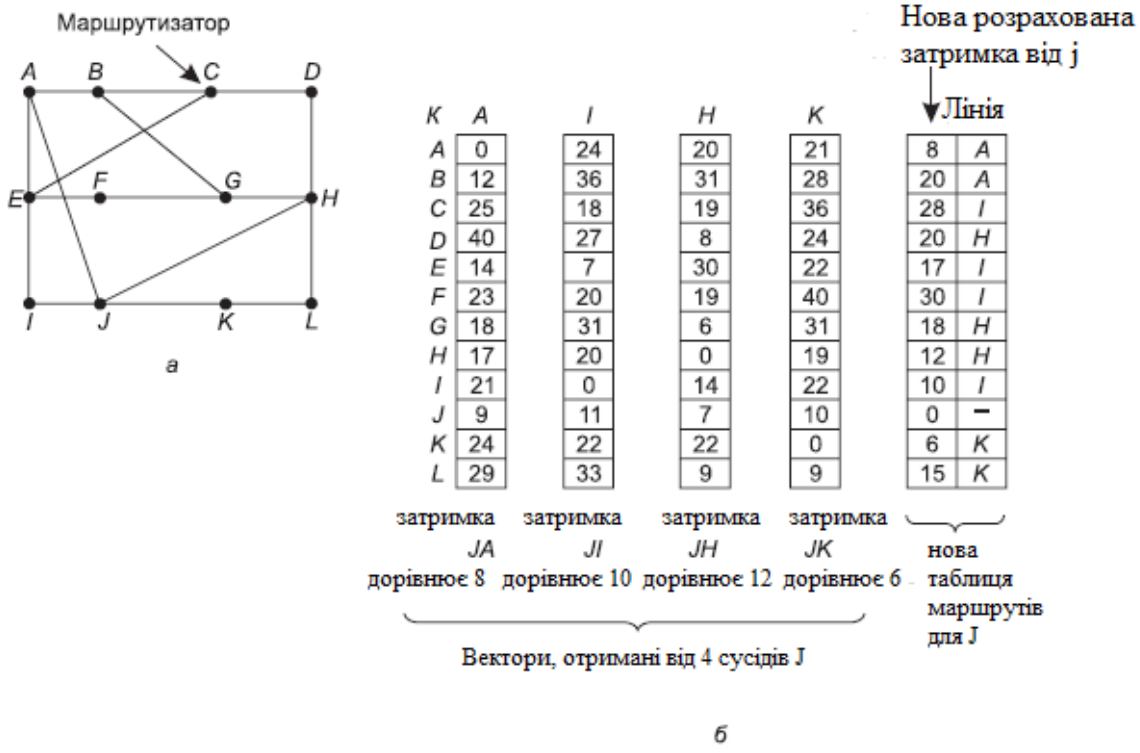


Рис.1 - мережа(а); отримані від A, I, H та K вектори та нова таблиця маршрутизації для J(б)

Проблема рахунки до нескінченності

Встановлення маршрутів, відповідних найкоротшим шляхам, в мережі називається конвергенцією (convergence). Алгоритм маршрутизації по вектору відстаней - простий метод, що дозволяє маршрутизаторам спільно обчислювати найкоротші шляхи. Однак на практиці він володіє серйозним недоліком: хоча правильна відповідь зрештою і знаходиться, процес його пошуку може зайняти досить багато часу.

Зокрема, такий алгоритм швидко реагує на гарні новини і дуже ліниво - на погані. Розглянемо маршрутизатор, для якого відстань до маршрутизатора X досить велике. Якщо при черговому обміні векторами його сусід A повідомить йому, що від нього до маршрутизатора X зовсім недалеко, наш маршрутизатор просто перемкнеться для передач маршрутизатора X на лінію, що проходить через цього соседа.Такім чином, хороша новина поширилася всього за один обмін інформацією. Щоб побачити, як швидко поширюються хороші звістки, розглянемо лінійну мережу з п'яти вузлів, показану на рис. 2, в якій мірою відстані служить кількість транзитних ділянок. Припустимо, що спочатку маршрутизатор A вимкнений, і всі інші

маршрутизатори про це знають. Тобто вони вважають, що відстань до A дорінює нескінченності.

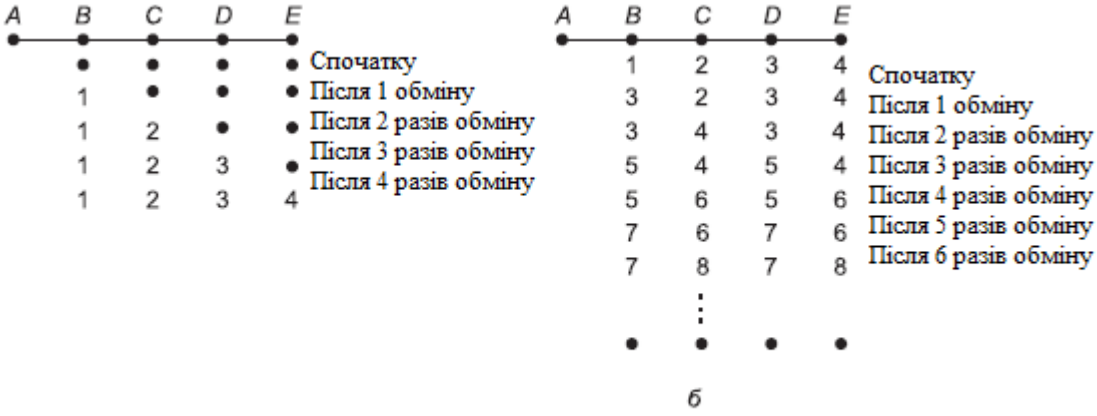


Рис.2 - проблема лічення до безкінечності

Коли в мережі з'являється A, решта маршрутизатори дізнаються про це за допомогою обміну векторами. Для простоти будемо припускати, що десь в мережі є гігантський гонг, в який періодично вдаряють, щоб ініціювати одночасний обмін векторами. Після першого обміну B дізнається, що у його сусіда зліва нульова затримка при зв'язку з A, а B позначає в своїй таблиці маршрутів, що A знаходиться зліва на відстані одного транзитного ділянки. Всі інші маршрутизатори в цей момент ще вважають, що A вимкнений. Значення затримок для A в таблицях на цей момент показані у другому рядку на рис. 2. а. При наступному обміні інформацією C дізнається, що у B є шлях до A довжиною 1, тому він оновлює свою таблицю, вказуючи довжину шляху до A, рівну 2, але D і E про це ще не знають. Таким чином, хороші вести поширюються зі швидкістю один транзитний ділянку за один обмін векторами. Якщо найдовший шлях в мережі складається з N транзитних ділянок, то через N обмінів всі маршрутизатори підмережі будуть знати про включені маршрутизаторах і заробили лініях.

Тепер розглянемо ситуацію на рис. 2. б, в якій всі зв'язки і маршрутизатори спочатку знаходяться у включеному стані. Маршрутизатори B, C, D і E знаходяться на відстані 1, 2, 3 і 4 транзитних ділянок від A відповідно. Раптово або A відключається, або відбувається обрив лінії між A і B (що з точки зору B одне і те ж).

При першому обміні пакетами B не чує відповіді від A. На щастя, C каже: «Не хвилюйся. У мене є шлях до A довжиною 2». B навряд чи здогадується, що шлях від C до A проходить через B. B може тільки припускати, що у C близько 10 вихідних зв'язків з незалежними шляхами

до А, найкоротша з яких має довжину 2. Тож тепер В думає, що може зв'язатися з А через С по шляху довжиною 3. При цьому першому обміні маршрутизатори D і E їх не оновлюють свою інформацію про А. При другому обміні векторами С зауважує, що у всіх його сусідів є шлях до А довжиною 3. Він вибирає один з них випадковим чином і встановлює свого відстань до А рівним 4, як показано в третьому рядку на рис. 2. б. Результати наступних обмінів векторами також показані на цьому малюнку. Тепер має бути зрозуміло, чому погані новини повільно поширюються - жоден маршрутизатор не може встановити значення відстані, більш ніж на одиницю перевищує мінімальне значення цієї відстані, що зберігається в його сусідів. Таким чином, всі маршрутизатори будуть до нескінченності збільшувати значення відстані до вимкненого маршрутизатора. Кількість необхідних для завершення цього процесу обмінів векторами можна обмежити, якщо встановити значення цієї «безкінечності» рівним довжині найдовшого шляху плюс 1.

Не дивно, що ця проблема називається рахунком до безкінечності (count-toinfinity).

Було зроблено багато спроб вирішити її, наприклад можна заборонити маршрутизатора повідомляти про свої найкоротших шляхах сусідам, від яких вони отримали цю інформацію, за допомогою правила розколотого горизонту внесеного в RFC 1058. Однак на практиці всі ці евристичні правила з красивими назвами виявилися абсолютно марними. Суть проблеми полягає в тому, що коли X повідомляє Y про те, що у нього є якийсь шлях, у Y немає ніякої можливості дізнатися, чи входить він сам в цей шлях.

Завдання на лабораторну роботу

Попередні знання / підготовка

Для вивчення маршрутизації по протоколу RIP створена проста Маршрутизовані мережу. На даному занятті ви сконфігуріруете протокол RIP для використання в мережі і налаштуєте мережеві пристрої, що беруть участь в обміні даними по мережі.

Крок 1. Налаштування маршрутизатора SVC01 і включення маршрутизації по протоколу RIP.

а. В інтерфейсі командного рядка налаштуйте інтерфейс Fast Ethernet 0/0, використовуючи IP-адреса 10.0.0.254 / 8.

б. Налаштуйте інтерфейс serial 0/0/0, використовуючи перший відповідний IP-адреса в мережі 192.168.1.0 / 24 для підключення до маршрутизатора RTR01. Введіть частоту синхронізації: 64000.

в. Налаштуйте інтерфейс Serial 0/0/0, використовуючи перший відповідний IP-адреса в мережі 192.168.2.0 / 24 з тактовою частотою 64000.

г. За допомогою команди no shutdown включіть налаштовані інтерфейси.

д. Налаштуйте маршрутизацію по протоколу RIP для сповіщення мереж про налаштованих інтерфейсах.

е. Налаштуйте кінцеві пристрої.

• Налаштуйте сервер Server0, використовуючи перший відповідний IP-адреса в мережі 10.0.0.0 / 8. Налаштуйте відповідний шлюз і маску підмережі.

• Налаштуйте принтер Printer0, використовуючи другий відповідний IP-адреса в мережі 10.0.0.0 / 8. Налаштуйте відповідний шлюз і маску підмережі.

Крок 2. Налаштування маршрутизатора RTR01 і включення маршрутизації по протоколу RIP.

а. Налаштуйте інтерфейс Fast Ethernet 0/0, використовуючи перший відповідний IP-адреса в мережі 192.168.0.0 / 24 для підключення до маршрутизатора RTR02.

б. Налаштуйте інтерфейс serial 0/0/0, використовуючи другий відповідний IP-адреса в мережі 192.168.1.0 / 24 для підключення до маршрутизатора SVC01.

в. Налаштуйте для інтерфейсу Fast Ethernet 0/1 IP-адреса 172.16.254.254 / 16.

г. Додайте всі налаштовані інтерфейси за допомогою команди no shutdown.

д. Налаштуйте маршрутизацію по протоколу RIP для сповіщення мереж про налаштованих інтерфейсах.

е. Налаштуйте кінцеві пристрої.

• Для PC0 використовується перший відповідний IP-адреса в мережі 172.16.0.0 / 16.

• Для PC1 використовується другий відповідний IP-адреса в мережі 172.16.0.0 / 16.

• Налаштуйте відповідний шлюз і маску підмережі для кожного з комп'ютерів.

Крок 3. Налаштування маршрутизатора RTR02 і включення маршрутизації по протоколу RIP.

а. Налаштуйте інтерфейс Fast Ethernet 0/0, використовуючи другий відповідний IP-адреса в мережі 192.168.0.0 / 24 для підключення до маршрутизатора RTR01.

б. Налаштуйте інтерфейс serial 0/0/0, використовуючи другий відповідний IP-адреса в мережі 192.168.2.0 / 24 для підключення до маршрутизатора SVC01.

в. Налаштуйте для інтерфейсу Fast Ethernet 0/1 IP-адреса 172.17.254.254 / 16.

г. Додайте всі налаштовані інтерфейси за допомогою команди "no shutdown".

д. Налаштуйте маршрутизацію по протоколу RIP для сповіщення мереж про налаштованих інтерфейсах.

е. Налаштуйте кінцеві пристрої.

• Для PC2 використовується перший відповідний IP-адреса в мережі 172.17.0.0 / 16.

• Для PC3 використовується другий відповідний IP-адреса в мережі 172.17.0.0 / 16.

• Налаштуйте відповідний шлюз і маску підмережі для кожного з комп'ютерів.

Крок 4. Перевірка конфігурації протоколу RIP на кожному маршрутизаторі.

а. Переконайтеся в тому, що маршрутизація RIP повністю конвергируют, за допомогою команд show ip protocols ishow ip route інтерфейсу командного рядка кожного з маршрутизаторів. Команда show ip protocolsпредназначена для відображення списку мереж, в які відбувається відправлення оновлень, і адрес сусідніх маршрутизаторів, що використовують RIP. Команда show ip route відображає список всіх

відомих локальному маршрутизатора маршрутів, у тому числі маршрутів RIP, які позначені символом "R".

б. Тепер, кожне з пристроїв, задіяних у даній вправі, повинно успішно виконувати ехо-запити до решти пристроям.

в. Для перевірки зробленої роботи натисніть кнопку Check Results (Перевірити результати) у нижній частині вікна інструкцій.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008
2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.
3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.
4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.
5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.
6. CISCO Internetworking technology overview – Cisco, 1999.

ЛАБОРАТОРНА РОБОТА 5
СЛУЖБА ІМЕН ДОМЕНІВ DNS

Мета роботи: отримати практичні навички в роботі зі службою домених імен

Короткі теоретичні відомості

Хоча програми теоретично можуть звертатися до веб-сторінок, поштових скриньках та інших ресурсів по мережевих адресах комп'ютерів (наприклад, IP), на яких зберігається дана інформація, користувачам важко запам'ятовувати такі адреси. Крім того, розміщення веб-сторінки компанії за адресою 128.111.24.41 означатиме, що в разі переїзду сервера компанії на нову машину, новий IP буде необхідно повідомити всім зацікавленим особам. Для відділення імен машин від їх адрес було вирішено використовувати зрозумілі імена високого рівня. Тому звернутися до веб-серверу компанії можна за адресою www.cs.washington.edu. Проте, так як мережа сама по собі розуміє тільки числові адреси, потрібен механізм перетворення імен в мережеві адреси. У наступних розділах ми вивчимо, як проводиться це відображення в Інтернеті.

Колись давно в часи мережі ARPANET відповідність між текстовими та чис-ловимі адресами просто записувалося в файлі hosts.txt, в якому перераховувалися всі імена комп'ютерів і їх IP-адреси. Щоночі всі хости отримували цей файл з сайту, на якому він зберігався. В мережі, що складається з декількох сотень великих машин, що працюють під управлінням системи з поділом часу, такий підхід виправдовував себе.

Однак ще за довго до того, як до мережі були підключені мільйони комп'ютерів, всім стало ясно, що цей спосіб не зможе працювати вічно. По-перше, розмір файлу рано чи пізно став би занадто великим. Однак, що ще важливіше, якщо управління іменами хостів не здійснює централізовано, неминуче виникнення конфліктів імен. Водночас уявити собі централізоване управління іменами всіх хостів гігантської міжнародної мережі досить складно. Для вирішення вищезгаданих проблем в 1983 році і була розроблена служба імен доменів (DNS, Domain Name System). Відтоді вона стала найважливішою частиною Інтернету.

Суть системи DNS полягає в ієрархічній схемі імен, заснованої на доменах, і розподіленій базі даних, що реалізує цю схему імен. В першу чергу ця система використовується для перетворення імен хостів в IP-адреси, але також може використовуватися і в інших цілях. Визначення

системи DNS дано в RFC 1034, 1035, 2181 і далі розроблено в багатьох інших.

В загальних рисах система DNS застосовується в такий спосіб. Для перетворення імені в IP-адресу прикладна програма звертається до бібліотечної процедурі, яка називається розпізнавачем (resolver), передаючи їй ім'я як параметр. Розпізнавач посилає запит, що містить ім'я, локальному DNS-серверу, який шукає ім'я і повертає відповідний IP-адресу розпізнавачів, який, в свою чергу, передає цю адресу прикладнії програмі. Запит і відповідь передають-ся як UDP-пакети. Маючи IP-адресу, програма може встановити TCP-з'єднання з адресатом або послати йому UDP-пакети.

Простір імен DNS

Управління великим і постійно змінюючимся набором імен являє собою нетривіальну задачу. В поштовій системі на листах потрібно вказувати (явно або неявно) країну, штат або область, місто, вулицю, номер будинку, квартиру і прізвище одержувача. Завдяки використанню такої ієрархічної схеми не виникає плутанини між Марвіном Андерсоном, що живуть на Мейн-стріт в Уайт-Плейнс, штат Нью-Йорк, і Марвіном Андерсоном з Мейн-стріт в Остіні, штат Техас. Система DNS працює аналогічно.

Для Інтернету основа ієрархії іменування розроблена організацією під на-званням ICANN (Internet Corporation for Assigned Names and Numbers - інтернет-корпорація з присвоєння імен та адрес). ICANN була створена для цих цілей в 1998 році, так як Інтернет розвинувся у всесвітній економічний концерн. Інтернет концептуально розділений на більш ніж 250 доменів верхнього рівня (top-level domains). Доменами називають в Інтернеті безліч хостів, об'єднаних в логічну групу. Кожен домен верхнього рівня підрозділяється на піддомени (subdomains), які, в свою чергу, також можуть складатися з інших доменів і т. Д. Всі ці домени можна розглядати у вигляді дерева, показаного на рис.1. Листям дерева є домени, не розділяються на піддомени (але складаються з хостів, звичайно). Такий кінцевий домен може складатися з одного хоста або може пред-ставлять компанію і містити в собі тисячі хостів.

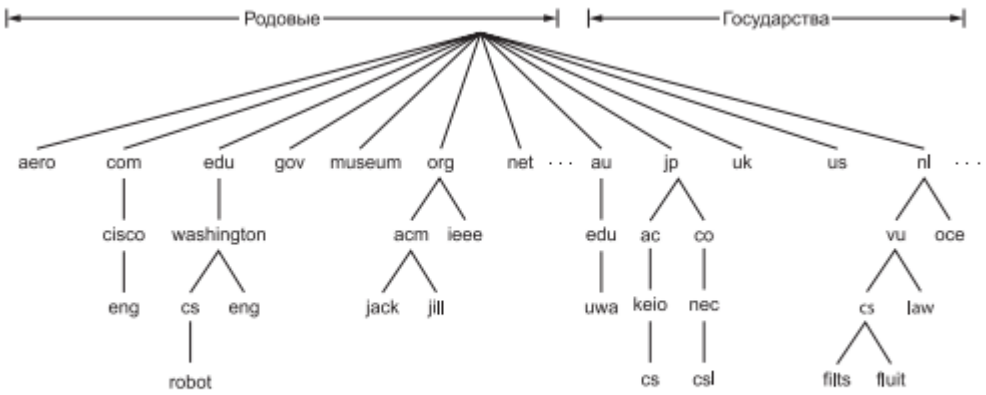


Рис.1 - Частина доменних імен Інтернет простору

Домени верхнього рівня розділяються на дві групи: родові домени і домени держав. В майбутньому будуть додаватися нові базові домени вищого рівня.
За кожною державою відповідно до міжнародного стандарту ISO3166 закріплений один домен держави. Інтернаціоналізовані доменні імена країн, в яких використовується алфавіт, відмінний від латинського, були введені в 2010 році. Ці домени дозволяють іменувати хости, використовуючи арабські, кириличні, китайські та інші писемності.

Зарезервувати домен другого рівня, такий як імя_компанii.com, просто. Домени вищого рівня управляються реєстраторами (registrars), призначеними ICANN. Для того щоб отримати ім'я, потрібно просто звернутися до відповідного (в даному випадку com) і перевірити, чи доступне бажане ім'я і не є воно чиєїсь торговою маркою. Якщо все гаразд, замовник реєструється і за невелику щорічну абонентську плату отримує домен другого рівня.

Однак у міру комерціалізації та інтернаціоналізації Інтернету з'являється все більше спірних питань, особливо щодо іменування доменів. Ці суперечки захоплюють і саму ICANN. Деякі домени є самоорганизующимися, на інших існують обмеження і не всякий може отримати ім'я (як показано в табл. 7.1). Але які обмеження доречні? Взяти хоча б домен pro. Він призначений для кваліфікованих фахівців. Але хто є фахівцем, а хто ні? Зрозуміло, що доктори і адвокати - це професіонали, суперечці немає. А що робити з вільними фотографами, вчителями музики, заклинателями, водопровідниками, перукарями, татуювальник, найманцями і повіями? Чи мають право кваліфіковані представники всіх цих та багатьох інших професій отримувати домени pro? Хто повинен це визначати?

Крім того, на іменах можна заробляти. Так країна Тувалу здала в оренду права на свій домен tv за $ 50 млн завдяки тому, що код країни відмінно підходить для реклами телевізійних сайтів. Практично всі

загальновживані англійські слова використовуються як імена піддоменів com, разом з найбільш частими помилками. Спробуйте набрати якесь слово, яке стосується домашнього господарства, тварин, рослин, частин тіла т. д. У самої практики реєстрації доменних імен з метою їх подальшого продажу зацікавленій стороні навіть є назва - кіберсквоттінг (cybersquatting). Багато компаній, які виявилися не достатньо спритними в цьому питанні, виявили, що найочевидніші доменні імена вже зайняті, коли почалася ера Інтернету і вони спробували зареєструватися. У загальному і цілому, якщо не були порушені права на товарний знак і не було почато шахрайських дій, щодо імен працює правило «першим Запитив - першим отримав». Проте політика щодо вирішення спорів з приводу імен все ще не до кінця розроблена.

Ім'я кожного домена, подібно повному шляху до файлу в файлової системі, складається з шляху від цього домену до (безіменній) вершини дерева. Компоненти шляху поділуляють точками. Так, домен технічного відділу корпорації Cisco може виглядати як eng.cisco.com, а не так, як це прийнято в стилі UNIX (/ com / cisco / eng). Слід відзначити, що через таку ієрархічної системи найменування eng.cisco.com не конфліктує з потенційним використанням імені eng в домені eng.washington.edu, де він може позначати факультет англійської мови Вашингтонського університету. Імена доменів можуть бути абсолютними і відносними. Абсолютна ім'я домена завжди закінчується крапкою (наприклад, eng.cisco.com.), Тоді як відносне ім'я - ні. Для того щоб можна було єдиним чином визначити істинні значення відносних імен, вони повинні інтерпретуватися в деякому контек-сте. У кожному разі іменований домен означає певний вузол дерева і всі вузли під ним.

Імена доменів нечутливі до зміни регістра символів. Так, наприклад, edu, Edu і EDU означають одне і те ж. Довжина імен компонентів може досягати 63 символів, а довжина повного шляху не повинна перевершувати 255 символів.

В принципі, нові домени можуть додаватися в дерево з використанням як родового домена, так і домена, що позначає країну. Наприклад, cs.wasington.edu можна без проблем помістити в домен us під ім'ям cs.wasington.wa.us. На практиці, однак, майже всі організації в США поміщаються під родовими доменами, тоді як майже всі організації за межами США розташовуються під доменами своїх держав. Не існує будь-яких правил, що забороняють реєстрацію під кількома до-менами верхнього рівня. Великі компанії часто саме так і чинять (наприклад, sony.com, sony.net і sony.nl).

Кожен домен управляє розподілом доменів, розташованих під ним. Наприклад, в Японії домени ac.jp і co.jp відповідають американським

доменах edu і com. В Голландії подібне відмінність не використовується, і всі домени організацій поміщаються прямо під доменом nl. Як приклад наведемо імена доменів факультетів обчислювальної техніки («комп'ютерних наук» - comuter science) трьох університетів.

1. cs.wasington.edu (Вашингтонський університет, США)
2. cs.vu.nl (університет Врійе, Нідерланди)
3. cs.keio.ac.jp (університет Кейо, Японія)

Для створення нового домену потрібен дозвіл домену, в який він буде включений. Наприклад, якщо у Вашингтонському університеті утворилася група VLSI, яка хоче зареєструвати домен vlsi.cs.wasington.edu, їй потрібно дозвіл від того, хто керує доменом cs.wasington.edu. Аналогічно, якщо створюється новий університет, наприклад, університет Північної Дакоти, він повинен попросити менеджера домена edu привласнити їх домену ім'я unsd.edu (якщо воно ще не зайнято). Таким чином, вдається уникнути конфлікту імен, а кожен домен відстежує стан всіх своїх піддоменів. Після того як домен створений і зареєстрований, в ньому можуть створюватися піддомени, наприклад cs.unsd.edu, для чого вже не потрібно дозволу вищестоящих доменів.

Структура доменів відображає не фізична будова мережі, а логічне поділ між організаціями та їх внутрішніми підрозділами. Так, якщо факультети обчислювальної техніки та електротехніки розташовуються в одній будівлі і користуються однією загальною локальною мережею, вони проте можуть мати різні домени. І навпаки, якщо, скажімо, факультет обчислювальної техніки розташовується в двох різних корпусах університету з різними локальними мережами, логічно всі хости обох будівель зазвичай належать до одного й того ж домену.

У кожного домена, незалежно від того, чи є він одиноким хостом або доменом верхнього рівня, може бути набір асоційованих з ним записів ресурсів (resource records). Ці записи є базою даних DNS. Для одинокого хоста запис ресурсів найчастіше представляє собою просто його IP-адресу, але існує також багато інших записів ресурсів. Коли распознаватель передає ім'я домена DNS-серверу, те, що він отримує назад, являє собою записи ресурсів, асоційоване з його ім'ям. Таким чином, істинне призначення системи DNS полягає в перетворенні доменних імен в записи ресурсів.

Запис ресурсу складається з п'яти частин. Хоча для ефективності вони часто пере-кодуються в двійкову форму, в більшості описів записи ресурсів представ-лени у вигляді ASCII-тексту, по одному рядку на запис ресурсу. Ми будемо використовувати наступний формат:

Поле Domain_name (ім'я домена) позначає домен, до якого відноситься поточна запис. Зазвичай для кожного домена існує декілька записів ресурсів, і кожна копія бази даних зберігає інформацію про декілька доменах. Поле імені домена є первинним ключем пошуку, використовуваним для виконання запитів. Порядок записів в базі даних значення не має. У відповідь на запит про домен повертаються всі записи необхідного класу.

Поле Time_to_live (час життя) вказує, наскільки стабільно стан записі. Рідко мінливим даними присвоюється високе значення цього поля, наприклад, 86 400 (число секунд в добі). Непостійна інформація позначається невеликим значенням, наприклад, 60 (1 хвилина).

Третім полем кожного запису є поле Class (клас). Для інформації Інтернета значення цього поля завжди дорівнює IN. Для іншої інформації застосовуються інші коди, однак на практиці вони зустрічаються рідко.

Поле Type (тип) означає тип DNS-записи. Їх існує досить багато. Важливі типи записів перераховані в табл. 2.

Запис SOA (Start Of Authority - початкова точка повноважень) повідомляє ім'я первинного джерела інформації про зону сервера імен (описаного нижче), адреса електронної пошти його адміністратора, унікальний порядковий номер, різні прапори і тайм-аути.

Найважливішою є запис A (Address - адреса). Вона містить 32-розрядну IPv4-адрес уінтерфейсу для хоста. У відповідного запису AAAA («quad A» - «чотири A») є 128-розрядна IPv6-адреса. У кожного хоста в Інтернеті повинен бути щонайменше одна IP-адреса, щоб інші машини могли з ним спілкуватися. На деяких хостах може бути одночасно встановлено декілька мережевих з*єднань. В цьому випадку їм потрібно за дві або більше записи типу A або AAAA. Відповідно, DNS може видавати кілька адрес на одне ім'я.

Запис MX є стандартною. У ній вказується ім'я хоста, готового приймати пошту для зазначеного домену. Справа в тому, що не кожна машина може займатися прийомом пошти. Якщо хто-небудь хоче надіслати листа на адресу, наприклад bill @microsoft.com, то що відправляє хосту потрібно буде спочатку знайти поштовий сервер на microsoft.com. Запис MX може допомогти в цих пошуках.

Таб.2. – Основні типи записів ресурсів DNS

Тип	Зміст	Значення
SOA	Початковий запис зони	Параметри для цієї зони
A	IPv4 адреса хосту	Ціле число, 32 двійкових розряди

AAAA	IPv6 адреса хосту	Ціле число, 128 двійкових розрядів
MX	Обмін поштою	Пріорітет, з яким домен бажає приймати електронну пошту
NS	Сервер імен	ім’я серверу для цього домену
CNAME	Канонічне ім’я	ім’я домену
PTR	Вказівник	Псевдоніми IP-адреси
SPF	Правило відправки пошти	Правила відправки пошти, закодовані в текстовому вигляді
SRV	Сервіс	Хост даного сервісу
TXT	Текст	Не інтерпретує мий ASCII - текст

Ще один важливий тип запису - це NS. Запис NS містить інформацію про сервер імені для домену або піддомена. Це хост, на якому міститься копія бази даних для домена. Він використовується в процесі пошуку імені, тому ми коротенько опишемо цей процес.

Запіс CNAME дозволяють створювати псевдоніми. Уявімо собі, що людина, знайомий в загальних рисах з формуванням імен в Інтернеті, хоче послати повідомлення користувачеві paul на відділенні обчислювальної техніки Массачусетського технологічного інституту (MIT). Він може спробувати вгадати потрібний йому адресу, склавши рядок paul@cs.mit.edu. Однак ця адреса працювати не буде, так як домен відділення обчислювальної техніки Массачусетського технологічного інституту насправді називається csail.mit.edu. Таким чином, для зручності тих, хто цього не знає, MIT може створити запис CNAME, що дозволяє звертатися до потрібного домену за обома іменах. Такий запис буде мати наступний вигляд:

Як і CNAME, запис PTR вказує на інше ім'я. Однак на відміну від запису CNAME, що є, власне, Макровизначення (тоесть механізмом заміни одного рядка інший), PTR являє собою звичайний тип даних DNS, інтерпретація якого залежить від контексту. На практиці запис PTR майже завжди використовується для асоціації імені з IP-адресою, що дозволяє за IP-адресою знаходити ім'я відповідної машини. Це називається зворотним пошуком (reverse lookups).

Запис SRV - це новий тип, що дозволяє визначати хост для шуканого сервісу в домені. Наприклад, веб-сервер для cs.wasington.edu може бути

визначений як cockatoo.cs.wasington.edu. Даний запис є розширеним варіантом записи MX, яка виконує ту ж задачу в рамках поштових сервісів.

SPF - також новий тип запису. Він дозволяє домену закодувати інформацію про те, які машини будуть відсилати з нього листи в іншу частину Інтернету. Це допомагає приймаючим машинам перевіряти, чи припустима дана пошта. Якщо пошта приходить з машини, яка називається dodgy, а доменні записи говорять про те, що пошта з домена буде надсилатися тільки машиною під назвою smtp, великі шанси того, що дані повідомлення є спамом.

Останні в списку, TXT-записи спочатку призначалися для того, щоб дозволити доменах ідентифікувати себе довільним чином. Сьогодні з їх допомогою зазвичай кодується інформація, призначена для зчитування машиною, звичайно це SPF-інформація.

Нарешті, останнє поле записи ресурса- це полеValue (значення) - може бути числом, ім'ям домена або текстової ASCII-рядком. Сенс поля залежить від типу запису. Короткий опис поля Value для кожного з основних типів записів дано в табл. 2.

Приклад інформації, що зберігається в базі даних DNS домену, наведено в лістингу 1. У ньому показана частина (гіпотетичної) бази даних домену cs.vu.nl, представленого також у вигляді вузла дерева доменів на рис. 1. У базі даних міститься сім типів записів ресурсів

Лістинг 1. Частина можливої бази даних домена cs.vu.nl

```
: Официальная информация для cs.vu.nl
cs.vu.nl.       86400   IN  SOA  star boss (9527.7200.7200.241920.86400)
cs.vu.nl.       86400   IN  MX   1 zephyr
cs.vu.nl.       86400   IN  MX   2 top
cs.vu.nl.       86400   IN  NS   star

star            86400   IN  A    130.37.56.205
zephyr          86400   IN  A    130.37.20.10
top             86400   IN  A    130.37.20.11
www             86400   IN  CNAME  star.cs.vu.nl
ftp             86400   IN  CNAME  zephyr.cs.vu.nl

flits           86400   IN  A    130.37.16.112
flits           86400   IN  A    192.31.231.165
flits           86400   IN  MX   1 flits
flits           86400   IN  MX   2 zephyr
flits           86400   IN  MX   3 top

rowboat             IN  A    130.37.56.201
                    IN  MX   1 rowboat
                    IN  MX   2 zephyr

little-sister       IN  A    130.37.62.23

laserjet            IN  A    192.31.231.216
```

В першому не закоментованому рядку лістінга 1 дається основна інформація про домен, яка в подальшому нас цікавити не буде. Наступні два рядки визначають два хоста, з якими слід зв'язатися в першу чергу при спробі доставити електронну пошту, надіслану за адресою person@cs.vu.nl. Хост на ім'я zephyr (спеціальна машина) слід опитати першим. У разі невдачі слід спробувати доставити лист машині по імені top. У наступному рядку визначений сервер імен для домену star.

Після порожніх рядків, доданих для зручності читання, йдуть рядки, що повідомляють IP-адреси для star, zephyr і top. Далі слід псевдонім www.cs.vu.nl, дозволяючий не звертатися до якоїсь конкретної машині. Створення цього псевдоніма дозволяє домену cs.vu.nl змінювати свій WWW-сервер, не змінюючи адреси, за якою користувачі зможуть продовжувати до нього звертатися. Те ж справедливо і для домену ftp.cs.vu.nl - FTP-сервера.

У секції, призначеної для машини flits, перераховані два IP-адреси і три можливі варіанти адреси для обробки пошти, надісланої на flits.cs.vu.nl. В першу чергу, природно, слід намагатися доставити лист самому комп'ютеру flits. Але якщо цей хост вимкнений, слід продовжувати спроби, звертаючись до хостів zephyr itop. Наступні три рядки містять типові записи для комп'ютера, в даному випадку для rowboat.cs.vu.nl. Зберігається в базі даних інформація містить IP-адресу, а також імена першого і другого хостів для доставки пошти. Слідом йде запис про машину, яка сама не здатна отримувати пошту. Останній рядок, ймовірно, описує лазерний принтер, підключений до Інтернету.

Сервери імен

Теоретично один сервер міг би містити всю базу даних DNS і відповідати на всі запити до неї. На практиці цей сервер виявився б настільки перевантаженим, що був би просто марним. Більш того, якби з ним коли-небудь що-небудь сталось, то весь Інтернет не працював би.

Щоб уникнути проблем, пов'язаних із зберіганням всієї інформації в одному місці, простір імен DNS розділене на непересекающіесязони (zones). Один можливий спосіб поділу простору імен, показаного на рис.1, на зони зображений на рис. 2. Кожна окреслена зона містить частину загального дерева доменів.

Розстановка меж зон цілком залежить від адміністратора зони. Це рішення ґрунтується на тому, скільки серверів імен вимагається в тій чи іншій зоні. Напри заходів, на рис. 2 у Вашингтонського університету є зона для washington.edu, яка керуєдоменом eng.washington.edu, але не доменом

cs.washington.edu, розташованим в окремій зоні зі своїми серверами імен. Подібне рішення може бути ухвалене, коли факультет англійської мови не хоче управляти власним сервером імен, але цього хоче факультет обчислювальної техніки.

Кожна зона також асоціюється з одним або більше сервером імен. Це хости, на яких знаходиться база даних для зони. Зазвичай у зони є один основний сервер імен, який отримує інформацію з файлу на своєму диску, і один або більше другорядних серверів імен, які отримують інформацію з основного сервера імен. Для підвищення надійності деякі сервери імен можуть бути розташовані поза зоною.

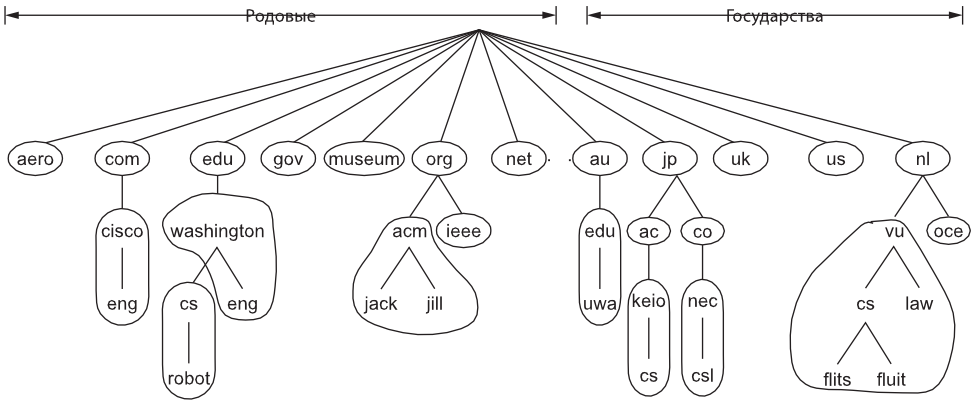


Рис. 3 Приклад пошуку розпізнувачем ім’я віддаленого хосту в десяти кроках

Процес пошуку адреси по імені називається дозволом імен (name resolution). Распознаватель звертається із запитом дозволу імені домена до локального серверу імен. Якщо шуканий домен належить до сфери відповідальності даного сервера імен, як, наприклад, домен top.cs.vu.nl підпадає під юрисдикцію домена cs.vu.nl, тоді даний DNS-сервер сам відповідає розпізнавачів на його запит, передаючи йому авторитетну запис (authoritative record) ресурсу. Авторитетної називають запис, одержувану від офіційного джерела, котра береже даний запис і керуючого її станом. Тому такий запис завжди вважається вірною, на відміну від Кешована записів (cached records), які можуть застарівати.

Однак що відбувається, якщо домен віддалений, як, наприклад, у випадку, коли flits.cs.vu.nl намагається знайти IP-адресу для robot.cs.washington.edu у Вашингтонському університеті? В цьому випадку, якщо в кеші немає інформації про запитуваній домені, доступному локально, сервер імен посилає віддалений запит. Пояснимо даний процес на прикладі, показаному на рис. 3. На першому кроці (позначений «1») надсилається запит локального сервера імен. Цей запит містить ім'я шуканого домену, тип (A) і клас (IN).

На наступному кроці посилається запит на один з кореневих серверів імен (root name servers), що знаходяться на вершині ієрархії. На цих серверах імен зберігається інформація про кожного домені вищого рівня. Щоб зв'язатися з кореневим сервером, на кожному сервері імен повинна бути інформація про один або більше кореневих серверах імен. Зазвичай ця інформація представлена в файлі системної конфігурації, який завантажується в кеш DNS, коли запускається сервер DNS. Він є просто списком записів NS і відповідних записів A.

Існує 13 кореневих серверів DNS, які називаються нехитро - від a-root-servers.net до m.root-servers.net. Кожен кореневий сервер логічно міг би бути окремим комп'ютером. Однак так як весь Інтернет залежить від кореневих серверів, вони є потужними машинами, а інформація, що зберігається на них, неодноразово дублюється. Більшість серверів розташоване в різних географічних точках, і доступ до них здійснюється за допомогою адресації будь-якого пристрою з групи, при цьому пакет доставляється на найближчий адресу (ми описали адресацію будь-якого пристрою групи у п'ятому розділі). Дублювання інформації підвищує надійність і продуктивність

Малоймовірно, щоб цей кореневий сервер імен знав адресу машини в Вашингтонськім університеті. Швидше за все, він навіть не знає адреси сервера імен самого університету, однак він повинен знати сервер імен домену edu, на якому розташований cs.washington.edu. Він повертає ім'я та IP-адреса для частини відповіді на третьому кроці. Далі локальний сервер імен продовжує цей складний шлях. Він направляє запит серверу імен edu (a.edu-servers.net), який видає ім'я сервера Вашингтонського університету. Цей процес проілюстрований шагамі 4 і 5. Тепер ми вже підійшли ближче. Локальний сервер імен відсилає запит на сервер імен Вашингтонського університету (крок 6). Якщо шукане ім'я домена знаходиться на факультеті англійської мови, буде отримана відповідь, так як зона університету цей факультет охоплює. Але факультет обчислювальної техніки вирішив запустити власний сервер імен. Запит повертає ім'я та IP-адреса сервера імен факультету обчислювальної техніки

Вашингтонського університету (крок 7).

Нарешті, локальний сервер імен запитує сервер імен факультету обчисли-котельної техніки Вашингтонського університету (крок 8). Цей сервер відповідає за до-мен cs.washington.edu, так що він повинен видати відповідь. У підсумку остаточний відповідь повертається (крок 9), і локальний сервер імен передає його на flits.cs.vu.nl (крок 10). Ім'я отримано.

Ви можете вивчити цей процес, використовуючи стандартні програми типу dig, які встановлені на більшості UNIX-систем. Наприклад,

надрукувавши ви відправите запит robot.cs.washington.edu на сервер імен a.edu-servers.net і отримаєте роздруківку результату. Так ви побачите інформацію, яку ми отримали на четвертому кроці в нашому прикладі, і дізнаєтеся ім'я та IP-адреси серверів імен Вашингтонського університету.

У цьому довгому сценарії є три технічні моменту, що вимагають пояснень. Коли хост flits. cs.vu.nl відсилає запит на локальний сервер імен, цей сервер виконує запит від імені flits, поки не отримає відповідь, яку можна буде повернути. Він не повертає часткових відповідей. Вони можуть бути корисними, але в запиті про них немає ні слова. Цей механізм називається рекурсивним запитом (recursive query).

З іншого боку, кореневий сервер імен (і кожний наступний) не продовжує рекурсивно запит локального сервера імен. Він повертає лише часткову відповідь і переходить до наступного запиту. Локальний сервер імен відповідає за продовження пошуку відповіді, спрямовуючи наступні запити. Цей механізм називається ітеративним запитом (iterative query).

В одному процесі пошуку імені можуть бути задіяні обидва механізму, як показано в цьому прикладі. Рекурсивні запити практично завжди здаються кращими, але багато серверів імен (особливо кореневі) їх не обробляють. Вони занадто завантажені. Ітеративні запити накладають вантаж обробки запиту на ту машину, яка їх породжує. Для локального сервера імен розумно підтримувати рекурсивні запити, щоб надавати сервіс хостам на своєму домені. Ці хости не обов'язково повинні бути налаштовані таким чином, щоб оббігати всі сервери імен, їм потрібна лише можливість звернутися до локального.

Друге, на чому варто загострити увагу, - це кешування. Всі відповіді, в тому числі всі повернуті часткові відповіді, зберігаються в кеші. Таким чином, якщо інший хост cs.vu.nl запрошувати robot.cs.washington.edu, відповідь буде вже відомий. Більш того, якщо хост запрошувати інший хост на тому ж домені, наприклад galah. cs.washington.edu, відповідь може бути відісланий безпосередньо на сервер імен, який відповідає за це ім'я. Подібним чином запити на інші домени на washington.edu можуть починатися безпосередньо з сервера імен washington.edu. Використання відповідей, збережених в кеші, серйозно скорочує кількість кроків в запиті і підвищує продуктивність. Сценарій, який ми накидали, насправді, є гіршим з можливих варіантів, так як в кеші немає корисної інформації.

Однак відповіді, збережені в кеші, не є авторитетними, так як зміни в домені cs.washington.edu не розповсюджуватимуться автоматично на всі кеші, в яких може зберігатися копія цієї інформації. З цієї причини записи кеша зазвичай довго не живуть. В кожного запису ресурсу присутня поле Time_to_live. Воно повідомляє віддалених серверів, наскільки довго слід

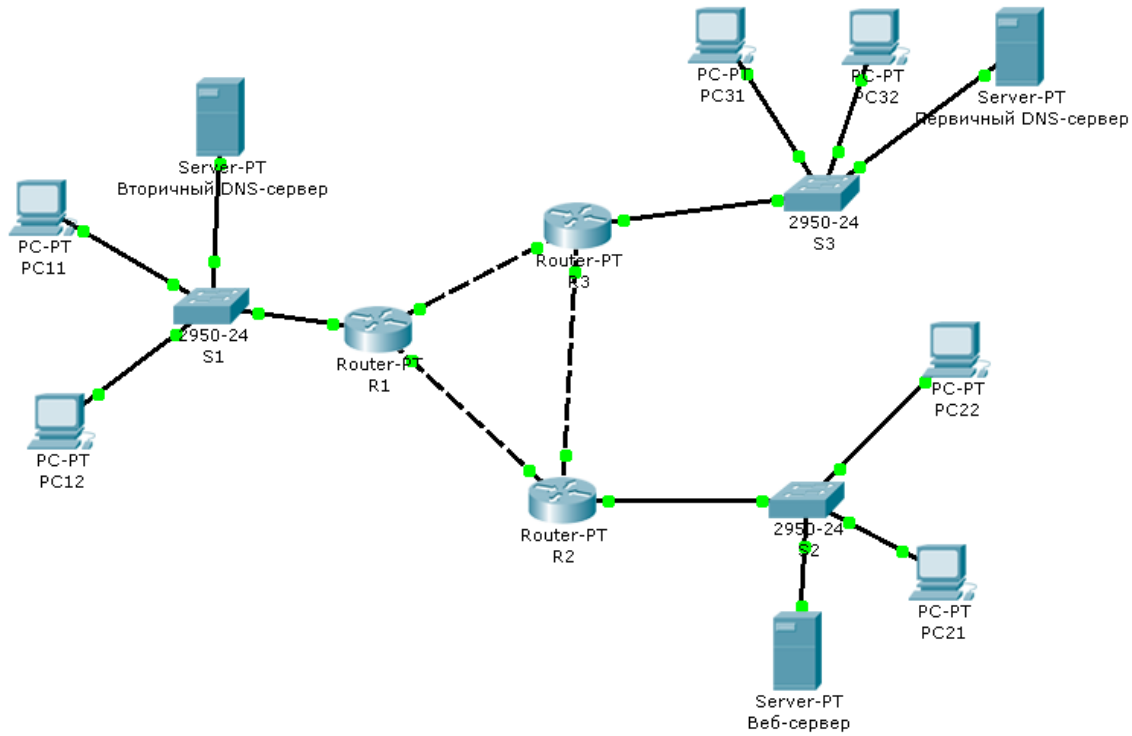
зберігати цей запис в кеші. Якщо яка-небудь машина зберігає постійна адреса роками, можливо, буде достатньо надійно зберігати цю інформацію в кеші протягом одного дня. Для більш непостійній інформації, ймовірно, більш обачно видаляти всі записи через кілька секунд або одну хвилину.

Завдання на лабораторну роботу
Налаштування DNS-сервера в середовищі Cisco Packet Tracer.

У лабораторній роботі необхідно забезпечити доступ до сайту з будь-якого з комп'ютерів мережі. Для роботи з сайтом не по IP, а по імені використовується служба DNS (яка знаходить відповідність між IP-адресою і DNS-ім'ям (якщо таке є) і навпаки).

З метою надійності використовується не тільки первинний, але і вторинний DNS-сервер, який зберігає копію бази даних первинного DNS-сервера.

1. Створити мережу за зразком.



2. Налаштуйте адресацію у відповідності з таблицею:

	Роутер в мережі	**Адреса мережі**	**Маска**
мережа №1	R1	192.168.1.0	255.255.255.128
мережа №2	R2	192.168.2.0	255.255.255.128
мережа №3	R3	192.168.3.0	255.255.255.128
мережа №4	R1,R2	10.0.0.0	255.0.0.0

мережа №5	R1,R3	20.0.0.0	255.0.0.0
мережа №6	R2,R3	30.0.0.0	255.0.0.0

(визначте клас, кожної з мереж і маску в префиксной формі).

3. Налаштуйте в мережі маршрутизацію по OSPF.

Налаштування DNS.

4. У налаштуваннях DNS-сервера створіть А-запис, в якому вказати відповідність між DNS-ім'ям і IP-адресою веб-сервера.

Наприклад: 192.168.1.1 → test.local

З комп'ютера зайдіть на вказаний сайт по DNS-імені.

Сайт повинен бути доступний з будь-якого комп'ютера мережі.

5. Налаштуйте для створеного домена алиас (друге ім'я) (test1.ru).

Для цього налаштуйте запис CNAME. Вкажіть два імені - ім'я вузла та ім'я вузла, яке також йому відповідає.

Часто зустрічаються ситуації, коли це потрібно:

Наприклад, будь-який сайт має адресу як з www, так і без www (yandex.ru і www.yandex.ru)

6. Налаштуйте запис SOA.

SOA-запис (Start Of Authority) - запис SOA містить ім'я первинного DNS-сервера (Primary Name Server), адреса, необхідний для встановлення технічних контактів (Hostmaster), серійний номер (Serial number) різні значення таймерів (Refresh, Retry, Expire, Minimum TTL)

Serial number

Serial number (серійний номер) - це номер версії файлу зони. Цей номер повинен бути позитивним цілим числом і збільшуватися кожного разу, коли в файл зони вносяться зміни. Збільшення серійного номера показує вторинним серверам, що зона змінена, і що їм необхідно оновити у себе зону.

У програмі Cisco Packet Tracer даний запис містить наступні поля:

- **DNS** – ім'я ресурсу.
- **Primary Name Server** – адреса первинного DNS-сервера для даного домена.
- **Minimum TTL** - визначає "час життя" негативних відповідей на запити про ресурси, що не існують в DNS. Допустимі значення: не менше 5 хвилин.

- **Retry Time** – показує, як довго вторинний сервер імен повинен чекати, перед тим як повторити спробу запиту первинного сервера (на предмет змін серійного номера даної зони), якщо попередня спроба виявилася невдалою.
- **Expire Time** – вказує верхнє обмеження по часу, протягом якого вторинний сервер може використовувати раніше отримані дані про зону до того як вони втратять силу через відсутність оновлення (наприклад, внаслідок відключення первинного сервера імен на тривалий час).
- **Refresh Time** - часовий параметр Refresh показує як часто вторинні сервери повинні запитувати первинний сервер, щоб дізнатися, чи не збільшився Чи Serial number (серійний номер) зони і, отже, чи не потрібно оновити її у себе.
- **Mail box** – поштова адреса відповідальної особи.

Для домену test1.ru налаштуйте всі зазначені вище параметри, у відповідність з таблицею:

Тип запису:	**Задаємо в Cisco Packet Tracer:**
Primary NameServer	IP-адресу вашого серверу
Minimum TTL	3600
Refresh Time	3600
Expire Time	86400
Mailbox	Ваш e-mail.
Minimum TTL	300

Примітка: з метою наочності - задані досить короткі інтервали часу. У реальності задають не менше години.

7. Налаштувати запис NS.

Запис NS (name server) вказує на DNS-сервер для даного домена.

Для стабільної роботи домену вказується не менше двох NS-записів.

У разі недоступності одного з DNS-серверів відбувається запит на інший DNS-сервер.

приклад:

Інформація про домен MAIL.RU

domain: MAIL.RU

nserver: ns1.mail.ru. 94.100.179.159

nserver: ns2.mail.ru. 94.100.186.189

nserver: ns3.mail.ru. 94.100.179.93

nserver: ns4.mail.ru. 94.100.178.100

nserver: ns5.mail.ru. 217.69.129.241

nserver: ns.mail.ru. 217.69.129.230

8. У попередніх завданнях були створені і налаштовані два DNS-сервера. Вимкніть один DNS-серверів і зайдіть після цього на веб-сайт.

Визначте і поясніть результат.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008

2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.

3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.

4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.

5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.

6. CISCO Internetworking technology overview – Cisco, 1999.

ЛАБОРАТОРНА РОБОТА 6
ПЛАНУВАННЯ МІЖМЕРЕЖЕВИХ ЕКРАНІВ

Мета роботи: Метою роботи є отримання навичок розміщення брандмауерів в підходящих місцях, що задовольняють вимогам безпеки.

Короткі теоретичні відомості

З швидким зростанням інтересу до Інтернету і операційній системі Windows NT безпеку мережі стала важливим завданням для багатьох компаній у всьому світі. Той факт, що інформація про злом і проникненні в корпоративні мережі та засоби, необхідні для цього, легко і широко доступні, ще більше посилює актуальність проблем безпеки. У зв'язку з цим адміністратори мережі часто витрачають набагато більше зусиль і часу на захист мереж, ніж на установку програм і адміністрування. Нові засоби, перевіряючі мережу на присутність слабких вузлів, типу Security Administrator Tool for Analyzing Networks (SATAN), безсумнівно, допомагають адміністраторам, але вони лише показують вразливі місця мережі, не забезпечуючи захисту. Давайте подивимося, від чого потрібно захищатися при роботі в мережі, щоб відповісти на питання: чому ви маєте потребу в firewall?

Проблеми з безпекою при з’єднанні з інтернетом

Коли ви підключаєте вашу приватну мережу до Інтернету, ви фізично з'єднуєтеся більш ніж з 50 тис. Невідомих мереж і всіма їх користувачами. Таке з'єднання відкриває вам шлях до багатьох корисних програмам і забезпечує величезні можливості поділу інформації, проте більшість приватних і корпоративних мереж містить інформацію, яка не повинна бути доступна іншим користувачам Інтернету. Крім того, не всі дії користувачів Інтернету є законними. Звідси випливають два основні питання:

Як захистити конфіденційну інформацію від тих, хто не має права доступу до неї?

1. Як захистити мережу та її ресурси від зловмисних користувачів і випадковостей, які відбуваються поза вашої мережі?

2. Будемо надалі називати атакою спробу доступу до прихованої інформації або просто проникнення в мережу, завжди маючи на увазі, що це не бажане для вас дію. Людину, яка виробляє таку дію, будемо називати зломщиком.

Захист конфіденціальної інформації

Конфіденційна інформація (як і будь-яка інформація в мережі) може або перебувати на носії інформації, або передаватися по мережі у вигляді пакетів. В обох станах інформація може стати предметом злому з боку як внутрішніх, так і зовнішніх користувачів. Ми обмежимося розглядом другого стану, коли інформація «в дорозі». Ось основні п'ять способів, які використовуються для отримання доступу до інформації:

- «винюхування» мережевих пакетів (network packet sniffers);
- містифікація пакетів IP (IP spoofing);
- взлом пароля (password attacks);
- перенаправлення пакетів зовні (distribution of sensitive information).
- використання проміжного комп'ютера (man-in-the-middle attacks)

При захисті інформації від таких атак ви хочете в першу чергу запобігти крадіжкам, руйнування, псування інформації або введення інформації ззовні. Стисло опишемо вищеперелічені способи злому.

«Винюхуванню» мережевих пакетів - це просто якийсь спосіб стеження за пакетами, що проходять по мережі. Зазвичай це програма, запущена на якомусь комп'ютері мережі.

Містифікація пакетів IP- це посилка пакетів з невірною інформацією про відправника, одержувача, номері порту або типі самого пакета. Наприклад, запис в пакет адреси відправника, що збігається з адресою машини внутрішньої мережі, може збільшити пріоритет обробки цього пакета.

Злом пароля зустрічається дещо частіше інших методів. Виробляється спроба будь-яким способом дізнатися пароль або отримати привілеї користувача root на який-небудь машині мережі.

Перенаправлення пакетів зовні є спробою направити назовні з вашої мережі інформацію, доступ до якої обмежено або заборонено.

Використання проміжного комп'ютера (man-in-the-middle attacks) - явне використання доступу до інформації вашої мережі. Приміром, це ситуація, коли користувач, що має офіційний доступ до інформації, намагається пересилати її в зовнішню мережу

Захист вашої мережі: підтримка цілісності мережі

При захисті інформації важливу роль відіграє підтримка цілісності мережі. Порушення налагодженої роботи мережі може привести до великих витрат часу на відновлення, а також відкрити нові можливості для зовнішніх атак, тобто стати першим етапом злому мережі. Ось методи, які для цього використовуються:

- «винюхування» мережевих пакетів (network packet sniffers);
- містифікація пакетів IP (IP spoofing);
- взлом пароля (password attacks);
- збій роботи (denial of service);
- атаки на рівні програм (application layer attacks).

Збій роботи - це не атака з метою злому. Це - спроба порушити або повністю блокувати роботу якогось вузла мережі, окремої програми або фізично знищити інформацію на носіях.

Атаки на рівні програм ставлять своєю метою одержання або знищення інформації за допомогою модифікації існуючих або установки нових, спеціально підготовлених програм. Аналогом таких атак є віруси.

Опису ідеї firewall як найбільш поширеного способу захисту мереж та інформації від злому присвячена інша частина статті.

Розвиток firewall

Очевидно, що необхідність захисту мереж породило цілий напрямок в комп'ютерній індустрії - технологію захисту мереж, яке в основному обертається навколо ідеї firewall.

Firewall - це точка поділу вашої мережі і тієї, до якої ви під'єднані. Цією точкою може бути комп'ютер, на якому запущено програмний firewall, або апаратно реалізований firewall. Firewall може бути простим, як звичайний маршрутизатор, фільтруючий пакети, або складним, що поєднує в собі функції багатозадачною маршрутизації, фільтрації пакетів і програмного proxy-сервера.

Перше покоління firewall (packet filtering firewalls), яке з'явилося в 1985 році, представляло собою перше покоління звичайних маршрутизаторів, що включають фільтрацію пакетів.

Друге покоління з'явилося в 1990-х і відомо як firewall ланцюгового рівня (circuit level firewalls).

Третє покоління - це firewall програмного рівня (application layer firewall).

Четверте покоління firewall засновано на динамічній фільтрації пакетів (dynamic packet filter firewalls), а першою його реалізацією була програма CheckPoint, випущена однойменною фірмою.

П'яте покоління firewall, яке з'явилося в 1996 році, базується на архітектурі kernel proxy. Зараз цей метод теж має як програмні, так і апаратні реалізації.

Лінія захисту (Security Perimeter)

Коли ви визначаєте тактику захисту мережі, ви повинні визначити спосіб охорони мережі та інформації, а також користувачів від пошкодження і втрати даних. Тактика захисту мережі заснована на управлінні рухом пакетів і контролем використання мережі. Ви повинні повністю описати мережу, встановити її «вузькі місця» і визначити дії, які робитимуться при порушенні захисту. При цьому ви точно обумовлюєте кордону, в яких діє ваша захист. Ці кордони і є мережа лінії захисту (perimeter networks).

Perimeter networks

Щоб встановити таку мережу захисту, ви повинні визначити мережу комп'ютерів, які потребують захисту, і визначити механізм їх захисту. Необхідним є умова, щоб firewall-сервер був шлюзом (gateway) між внутрішніми і зовнішніми мережами. Кожна мережа може містити мережі захисту всередині себе. Розрізняють зовнішні, середні і внутрішні мережі захисту. Зовнішня захист розділяє мережу, якою ви керуєте, і ту, якої ви не можете управляти. Середній рівень розділяє підмережі всередині вашої мережі або відокремлює доступні для зовнішніх користувачів комп'ютери від тих, до яких доступ заборонений. Внутрішній рівень дозволяє розділяти мережі всередині недоступних і зовні доступних частин вашої мережі.

Для подальшого викладу введемо три поняття: Trusted networks - мережі, які ви захищаєте і якими можете управляти (всередині perimeter network); Untrusted networks - мережі, якими ви не керуєте (зазвичай зовні), але з якими, проте, повинні обмінюватися інформацією; і Unknown networks - мережі, про які не можна сказати нічого певного, крім того, що вони існують.

Архітектура firewall

Firewall - це шлюз мережі, забезпечений правилами захисту. Він може бути апаратним або програмним. Відповідно до закладеними правилами обробляється кожен пакет, що проходить назовні або всередину мережі, причому процедура обробки може бути задана для кожного правила. Виробники програм і машин, що реалізують firewall-технології, забезпечують різні способи завдання правил і процедур. Зазвичай firewall створює контрольні записи, деталізують причину і обставини виникнення позаштатних ситуацій. Аналізуючи такі контрольні записи, адміністратори часто можуть виявити джерело атаки і способи її проведення, тим самим забезпечуючи себе додатковою інформацією про атаку.

Як працює фільтрація пакетів (packet filtering firewalls)

Кожен IP-пакет перевіряється на збіг закладеної в ньому інформації з допустимими правилами, записаними в firewall.

Параметри, які можуть перевірятися:

- фізичний інтерфейс руху пакета;
- адресу, з якої прийшов пакет (джерело);
- адресу, куди йде пакет (одержувач);
- тип пакета (TCP, UDP, ICMP);
- порт джерела;
- порт одержувача.

З цього переліку видно, що фільтрація пакетів не має справи з їх вмістом. Це дозволяє використовувати безпосередньо ядро операційної системи для завдання правил. По суті, створюються два списки: заперечення (deny) і дозвіл (permit). Всі пакети повинні пройти перевірку за всіма пунктами цього списку. Далі використовуються такі методи:

- якщо ніяке правило відповідності не знайдено, то видалити пакет з мережі;
- якщо відповідне правило знайдено в списку дозволів, то пропустити пакет;
- якщо відповідне правило знайдено в списку заперечень, то видалити пакет з мережі.

На додаток до цього firewall, заснований на фільтрації пакетів, може змінювати адреси джерел пакетів, що виходять назовні, щоб приховати тим самим топологію мережі (метод address translation).

Відзначимо переваги firewall, заснованого на фільтрації пакетів:

- фільтрація пакетів працює швидше інших firewall-технологій, бо використовується менша кількість перевірок;
- цей метод легко реалізуємо апаратно;
- одне-єдине правило може стати ключовим при захисті всієї мережі;
- фільтри не вимагають спеціальної конфігурації комп'ютера;
- метод address translation дозволяє приховати реальні адреси комп'ютерів в мережі.

Однак є й недоліки:

- немає перевірки вмісту пакетів, що не дає можливості, наприклад, контролювати, що передається по FTP. В цьому сенсі application layer і circuit level firewall набагато практичніше;
- немає інформації про те, який процес або програма працювали з цим пакетом, і відомостей про сесії роботи;
- робота ведеться з обмеженою інформацією пакета;

- в силу «низькорівневі» методу не враховується особливість переданих даних;
- слабо захищений сам комп'ютер, на якому запущено firewall, тобто предметом атаки може стати сам цей комп'ютер;
- немає можливості сигналізувати про позаштатних ситуаціях або виконувати при їх виникненні будь-які дії;

Firewall ланцюгового рівня (circuit level firewalls)

Оскільки при передачі великої порції інформації вона розбивається на маленькі пакети, цілий фрагмент складається з декількох пакетів (з ланцюга пакетів). Firewall ланцюгового рівня перевіряє цілісність всього ланцюга, а також те, що вона вся йде від одного джерела до одного одержувачу, і інформація про ланцюги всередині пакетів (а вона там є при використанні TCP) збігається з реально проходять пакетами. Причому ланцюг спочатку збирається на комп'ютері, де встановлений firewall, а потім вирушає одержувачу. Оскільки перший пакет ланцюга містить інформацію про всю ланцюга, то при попаданні першого пакета створюється таблиця, яка видаляється лише після повного проходження ланцюга.

Зміст таблиці таке:

- унікальний ідентифікатор сесії передачі, який використовується для контролю;
- стан сесії передачі: встановлено, передано або закрито;
- інформація про послідовність пакетів;
- адресу джерела ланцюга;
- адресу одержувача ланцюга;
- фізичний інтерфейс, використовуваний для отримання ланцюга;
- фізичний інтерфейс, використовуваний для відправлення ланцюга.

Ця інформація застосовується для перевірки допустимості передачі ланцюга. Правила перевірки, як і у випадку фільтрації пакетів, задаються у вигляді таблиць в ядрі.

Основні переваги firewall ланцюгового рівня:

- firewall ланцюгового рівня швидше програмного, так як виробляє менше перевірок;
- firewall ланцюгового рівня дозволяє легко захистити мережу, забороняючи з'єднання між певними адресами зовнішньої і внутрішньої мережі;
- можливо приховування внутрішньої топології мережі.

Недоліки firewall ланцюгового рівня:

- важко реалізувати цей алгоритм для не-TCP-протоколів;

- немає перевірки пакетів на програмному рівні;
- слабкі можливості запису інформації про нештатні ситуації, окрім інформації про сесії передачі;
- немає перевірки переданих даних;
- важко перевірити дозвіл або заперечення передачі пакетів.

Firewall програмного рівня

Крім цілісності ланцюгів, правильності адрес і портів, перевіряються також самі дані, передані в пакетах. Це дозволяє перевіряти цілісність даних і відстежувати передачу таких відомостей, як паролі. Разом з firewall програмного рівня використовується proxy-сервіс, який кешируєт інформацію для більш швидкої її обробки. При цьому виникають такі нові можливості, як, наприклад, фільтрація URL і встановлення автентичності користувачів. Всі з'єднання внутрішньої мережі з зовнішнім світом відбуваються через proxy, який є шлюзом. У proxy дві частини: сервер і клієнт. Сервер приймає запити, наприклад на telnet-з'єднання з внутрішньої мережі з зовнішньою, обробляє їх, тобто перевіряє на допустимість передачі даних, а клієнт працює з зовнішнім комп'ютером від імені реального клієнта. Природно, спочатку всі пакети проходять перевірку на нижніх рівнях.

Гідності proxy:

- розуміє і обробляє протоколи високого рівня типу HTTP і FTP;
- зберігає повну інформацію про сесії передачі даних як низького, так і високого рівня;
- можлива заборона доступу до деяких мережевих сервісів;
- є можливість управління пакетами даних;
- є приховування внутрішніх адрес і топології мережі, так як proxy є фільтром;
- залишається видимість прямого з'єднання мереж;
- proxy може перенаправляти запити мережевих сервісів на інші комп'ютери;
- є можливість кешування http-об'єктів, фільтрації URL і встановлення автентичності користувачів;
- можливе створення докладних звітних записів для адміністратора.
- Недоліки proxy:
- вимагає зміни мережевого стека на машині, де стоїть firewall;
- не можна напряму запустити мережеві сервіси на машині, де стоїть firewall, так як proxy перехоплює роботу портів;

- неминуче уповільнює роботу, тому всі дані обробляються двічі: «рідний» програмою і власне proxy;
- так як proxy повинен вміти працювати з даними будь-якої програми, то для кожної програми потрібен свій proxy;
- немає proxy для UDPi RPC;
- іноді необхідна спеціальна настройка клієнта для роботи з proxy;
- proxy не захищений від помилок в самій системі, а його робота сильно залежить від наявності останніх;
- коректність роботи proxy безпосередньо пов'язана з правильністю обробки мережевого стека;
- використання proxy може вимагати додаткових паролів, що незручно для користувачів.

Динамічна фільтрація пакетів (dynamic packet filter firewalls)

В основному цей рівень повторює попередній, за двома важливими винятками:

- можлива зміна правил обробки пакетів «на льоту»;
- включена підтримка UDP.

Рівень kernel proxy

Уровень kernel proxy возник достаточно недавно. Основная его идея — попытка поместить описанный выше алгоритм firewall программного уровня в ядро операционной системы, что избавляет компьютер от лишних затрат времени на передачу данных между ядром и программой proxy. Это повышает производительность и позволяет производить более полную проверку проходящей информации. Я не буду подробно останавливаться на этом методе, так как на данный момент разные фирмы реализуют его несколько по-разному и нет общепринятой концепции.

Коротко про реалізації firewall

Як вже було сказано, firewall може бути реалізований як програмно, так і апаратно. Апаратна реалізація являє собою якийсь спеціалізований комп'ютер, єдиною функцією якого є робота в якості firewall. Причому ця програма зашита в його залізо. Це дозволяє домагатися великої продуктивності. Однією з провідних фірм, що виробляють такі комп'ютери, є Cisco (серія Cisco Access Servers). Програмна реалізація - це просто програма, яка виконується на комп'ютері-шлюзі і виконує описані вище функції (наприклад, LanGuard, Cisco

IOS Software, Checkpoint). Очевидно, що для роботи такої програми необхідний досить потужний комп'ютер з великим об'ємом пам'яті, причому нерозумно сильно завантажувати цей комп'ютер іншою роботою. Для прикладу наведу короткий опис програми LanGuard, яка має один з найвищих рейтингів.

Програма написана під Windows NT і працює по четвертому, описаного вище рівню. Вона безкоштовна обмежений час, після чого дозволяє працювати, якщо мережа складається не більше ніж з п'яти комп'ютерів. Програма виконує наступні функції:

- захищає мережу від доступу зовнішніх користувачів, в тому числі і сам шлюз, дозволяючи розписувати правила по портам, додаткам і мережевим адресам;
- відстежує програми sniffer;
- відстежує кількість проходить по мережі інформації;
- відстежує використання Інтернет і конкретну зв'язок з сайтами, дозволяючи вводити правила, наприклад заборона використання будь-яких сайтів;
- відстежує віруси типу троянських коней;
- може створювати докладні звіти про використання мережі;
- видає попередження адміністратору.

Нижче на малюнках представлено декілька екранів, що наочно демонструють LanGuard.

На закінчення скажемо, що зараз firewall де-факто є стандартом захисту мереж. Більш того, він входить до складу багатьох операційних систем сімейства UNIX. Щорічно проводяться форуми, присвячені firewall, постійно удосконалюються наявні та створюються нові програми, що реалізують firewall. Зараз firewall існує для всіх відомих платформ. Більшість фірм сьогодні пропонують останній, п'ятий рівень firewall, що забезпечує роботу proxy на рівні ядра операційної системи.

Завдання на лабораторну роботу

Ви є техніком, що здійснює підтримку роботи мережі середнього підприємства. В процесі росту підприємства відкритий науково-дослідний відділ, що працює над новим, вельми секретним проектом. Існування проекту залежить від захисту даних, використовуванихнауково-дослідницькою групою.

Ви є техніком, що здійснює підтримку роботи мережі середнього підприємства. В процесі росту підприємства відкритий науково-дослідний відділ, що працює над новим, вельми секретним проектом. Існування проекту залежить від захисту даних, використовуванихнауково-дослідницькою групою.

Сценарій 1. Захист мережі від хакерів.

Так як в компанії підвищені вимоги до безпеки, рекомендується встановити міжмережевий екран для захисту мережі від хакерів, працюючих в Інтернеті. Дуже важливо обмежити доступ до внутрішньої мережі з Інтернету.

В міжмережевому екрані Firewall_1 попередньо налаштовані правила для забезпечення необхідної клієнту безпеки. Встановіть цей брандмауер в мережі клієнта і перевірте правильність його функціонування.

Крок 1. Заміна маршрутизатора Router_A брандмауером Firewall_1.

а. Демонтуйте маршрутизатор Router_A і замініть його брандмауером Firewall_1 Підключіть інтерфейс технології Fast Ethernet 0/0 брандмауера Firewall_1 до інтерфейсу Fast Ethernet 0/1 комутатора Switch_A.

б. Підключіть інтерфейс Fast Ethernet 0/1 брандмауера Firewall_1 до інтерфейсу Ethernet 6 хмари мережі ISP. (Використовуйте прямий кабель для обох сполук.)

в. Підтвердіть ім'я мережевого вузла для Firewall_1 - "Firewall_1".

г. На Firewall_1 налаштуйте IP-адресу глобальної мережі та маску підмережі для інтерфейсу

Fast Ethernet 0/1 209.165.200.225 і 255.255.255.224, відповідно.

д. На брандмауері Firewall_1 виберіть IP-адресу глобальної мережі та маску підмережі для інтерфейсу Fast Ethernet 0/1: 192.168.1.1 і 255.255.255.0.

Крок 2. Перевірка конфігурації брандмауера Firewall_1

Для перевірки настройки використовуйте команду **show run**. Нижче наводиться частина зразкового лістингу:

```
Firewall_1#show run
Building configuration...

hostname Firewall_1
!
interface FastEthernet0/0
ip address 192.168.1.1 255.255.255.0 ip nat inside
duplex auto speed auto
!
interface FastEthernet0/1
ip address 209.165.200.225 255.255.255.224 ip access-group 100 in
    ip nat outsidea. duplex auto
speed auto
```

!
interface Vlan1 no ip address shutdown
!
ip nat inside source list 1 interface FastEthernet0/0 overload ip classless
ip route 192.168.2.0 255.255.255.0 192.168.1.2 ip route 192.168.3.0 255.255.255.0 192.168.1.3
!
access-list 1 permit 192.168.0.0 0.0.255.255access-list 100 deny ip any host 209.165.200.225
<выходные данные опущены>
!
end

З комп'ютера ПК_В, відправте ехо-запит 209.165.200.225, щоб переконатися, що у внутрішнього комп'ютера мається доступ в Інтернет.

PC>**ping 209.165.200.225**

Pinging 209.165.200.225 with 32 bytes of data:

Reply from 209.165.200.225: bytes=32 time=107ms TTL=120б. Reply from 209.165.200.225: bytes=32 time=98ms TTL=120
Reply from 209.165.200.225: bytes=32 time=104ms TTL=120 Reply from 209.165.200.225: bytes=32 time=95ms TTL=120

Ping statistics for 209.165.200.225:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds:
Minimum = 95ms, Maximum = 107ms, Average = 101ms

В привілейованому режимі EXEC брандмауера Firewall_1 збережіть поточну конфігурацію в початкову за допомогою команди copy run start.

Сценарій 2. Захист мережі відділу досліджень і розробок

Тепер, коли вся мережа захищена від трафіку, що надходить з Інтернету, прийшов час захистити мережу відділу досліджень і розробок (підмережа Subnet C) від можливих проникнень з внутрішньої області мережі. Для проведення досліджень науково- дослідницькій групі необхідний доступ до серверів, розташованих в підмережі В, і до Інтернету. Комп'ютерам підмережі В повинно бути відмовлено в доступі до підмережінауково-дослідного відділу.

В міжмережевому екрані Firewall_2 попередньо налаштовані правила для забезпечення необхідної клієнту безпеки. Встановіть цей брандмауер в мережі клієнта. Перевірте правильність його функціонування.

Крок 1. Заміна маршрутизатора Router_C брандмауером Firewall_2.

а. Видаліть маршрутизатор Router_C і замініть його брандмауером Firewall_2.

б. Підключіть інтерфейс Fast Ethernet 0/1 брандмауера Firewall_2 до інтерфейсу Fast Ethernet 0/3 комутатора Switch_A. Підключіть інтерфейс Fast Ethernet 0/0 брандмауера Firewall_2 до інтерфейсу Fast Ethernet 0/1 комутатора Switch_C. (Використовуйте прямий кабель для обох сполук.)

в. Підтвердіть ім'я мережевого вузла для Firewall_2 - "Firewall_2".

г. На Firewall_2 налаштуйте IP-адресу глобальної мережі та маску підмережі для інтерфейсу Fast Ethernet 0/1: 192.168.1.3 і 255.255.255.0, відповідно.

д. На брандмауері Firewall_1 виберіть IP-адресу локальної мережі та маску підмережі для інтерфейсу FastEthernet 0/0: 192.168.3.1 і 255.255.255.0.

Крок 2. Проверка конфигурации брандмауэра Firewall_2

Для перевірки налаштувань використовуйте команду "show run". Далі представлена частина вихідних даних.

Firewall_2#**show run**Building configuration...
...
!
interface FastEthernet0/0
ip address 192.168.3.1 255.255.255.0 ip nat inside
duplex auto
speed auto
a. !
interface FastEthernet0/1
ip address 192.168.1.3 255.255.255.0 ip access-group 100 in
ip nat outside duplex auto speed auto
!
access-list 1 permit 192.168.3.0 0.0.0.255access-list 100 permit ip host 192.168.2.10 anyaccess-list 100 permit ip host 192.168.1.1 any
<вихідні дані >
!
end

За запитом команди на ПК_B використовуйте команду ping, щоб переконатися, що комп'ютери в підмережі Subnet B не можуть отримати доступ до комп'ютерів в підмережі Subnet C.

PC>**ping 192.168.3.10**

б.Pinging 192.168.3.10 with 32 bytes of data:

Request timed out. Request timed out. Request timed out. Request timed out.

Ping statistics for 192.168.3.10: Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)

За запитом команди на ПК_С використовуйте команду ping, щоб переконатися, що комп'ютери в підмережі Subnet C мають доступ до сервера в підмережі Subnet B.

PC>**ping 192.168.2.10**

Pinging 192.168.2.10 with 32 bytes of data:

Request timed out.

в.Reply from 192.168.2.10: bytes=32 time=164ms TTL=120 Reply from 192.168.2.10: bytes=32 time=184ms TTL=120 Reply from 192.168.2.10: bytes=32 time=142ms TTL=120

Ping statistics for 192.168.2.10:

Packets: Sent = 4, Received = 3, Lost = 1 (25% loss), Approximate round trip times in milli-seconds:
Minimum = 142ms, Maximum = 184ms, Average = 163ms

За запитом команди на ПК_С використовуйте команду ping, щоб переконатися, що комп'ютери в підмережі Subnet C мають доступ до Інтернету.

PC>**ping 209.165.200.225**

Pinging 209.165.200.225 with 32 bytes of data:

Reply from 209.165.200.225: bytes=32 time=97ms TTL=120 г.Reply from 209.165.200.225: bytes=32 time=118ms TTL=120 Reply from 209.165.200.225: bytes=32 time=100ms TTL=120 Reply from 209.165.200.225: bytes=32 time=110ms TTL=120

Ping statistics for 209.165.200.225:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds:
Minimum = 97ms, Maximum = 118ms, Average = 106ms

д. В привілейованому режимі EXEC брандмауера Firewall_2 збережіть поточну конфігурацію в початкову за допомогою команди copy run start.

е. Для перевірки зробленої роботи натисніть кнопку Check Results (Перевірити результати) в нижній частині вікна інструкцій.

Контрольні питання

а. Навіщо потрібно встановлювати брандмауер у внутрішній мережі?

б. Яким чином маршрутизатор, налаштований для використання довідки NAT, дозволяє захистити комп'ютерні системи, розташовані всередині маршрутизатора NAT?

Вивчіть розташування брандмауерів Firewall_1 і Firewall_2 в завершеною топології мережі. в.Які мережі можна вважати надійними і ненадійними для брандмауера Firewall_1? Які мережі вважаються надійними і ненадійними для брандмауера Firewall_2?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008
2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.
3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.
4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.
5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.
6. CISCO Internetworking technology overview – Cisco, 1999.

ЛАБОРАТОРНА РОБОТА 7
СПОСТЕРЕЖЕННЯ КОНВЕРГЕНЦІЇ МЕРЕЖІ

Мета роботи: отримати практичні навички моделювання та аналізу мережі.

Короткі теоретичні відомості

Найскладніша проблема в будь-якому протоколі динамічної маршрутизації - кільцеві маршрути. В дистанційно-векторних протоколах маршрутизації вона вирішується за рахунок використання безлічі різноманітних механізмів, час роботи яких, а отже, і час конвергенції мережі, може становити кілька хвилин після відмови якогось каналу. В протоколах з урахуванням стану каналів проблема кільцевих маршрутів вирішується за рахунок того, що кожен маршрутизатор зберігає у себе повну топологічну базу мережі, а також використовує складні математичні алгоритми для її обробки.

Протокол EIGRP виявляє кільцеві маршрути, і відбувається процес їх знищення за рахунок того, що кожен пристрій зберігає у себе деяку базову топологічну інформація, а завантаження центрального процесора (CPU) пристрою помітно знижується за рахунок того, що така топологічна інформація досить кратка. Коли маршрутизатор виявляє кілька маршрутів до однієї і тієї ж IP-підмережі, він поміщає тільки найкращий маршрут в таблицю маршрутизації. Топологічна інформація про мережу потрібна протоколу EIGRP для того ж, для чого вона потрібна протоколу маршрутизації OSPF: щоб після зміни топології відреагувати на зміну як можна швидше і використовувати новий маршрут, а також уникнути виникнення кільцевих (циклічних) маршрутів в мережі. Фактично протокол маршрутизації EIGRP зберігає записи про наступні транзитних маршрутизаторах і деякі деталі маршрутів до них, але не оперує топологічної інформації про мережу за найближчими транзитними маршрутизаторами. Менш докладна інформація про топологію мережі не вимагає чогось схожого на складний алгоритм SPF, тому конвергенція відбувається набагато швидше, вимагає менше ресурсів і ймовірність виникнення кільцевих маршрутів за рахунок цього зменшується.

Принцип работы EIGRP

Основними перевагами EIGRP є:

- низьке споживання мережевих ресурсів в режимі нормальної експлуатації (в умовах стабільної мережі передаються тільки пакети "hello")
- при виникненні змін по мережі передаються тільки зміни, що відбулися в маршрутної таблиці, а не вся таблиця цілком; це дозволяє зменшити навантаження на мережу, створювану протоколом маршрутизації
- малий час конвергенції в разі зміни в топології мережі (в окремих випадках збіжність забезпечується майже миттєво)
- протокол EIGRP є вдосконаленим протоколом дистанційній-векторної маршрутизації, в якому для розрахунку найкоротшого шляху до кінцевого адресою використовується алгоритм дифузного поновлення (Diffused Update Algorithm - DUAL).

Основні версії протоколу

Існують дві основні версії протоколу EIGRP - версія 0 і 1. У ранніх версіях програмного забезпечення Cisco IOS (аж до версій 10.3 (11), 11.0 (8) і 11.1 (3)) використовується більш рання версія протоколу EIGRP (з цієї причини деякі пояснення , що містяться в цьому документі, можуть бути не дійсні для цих версій). Ми настійно рекомендуємо використовувати останню версію EIGRP, оскільки ця версія містить безліч поліпшень, пов'язаних зі стабільністю і продуктивністю.

Основні принципи

При розрахунку найкращого шляху до кінцевого адресою типовий дистанційно-векторний протокол зберігає наступну інформацію: відстань (distance) (сумарна метрика або відстань, наприклад, лічильник переходів) і вектор (наступний перехід). Наприклад, на всіх маршрутизаторах мережі на рис. 1 виконується протокол маршрутної інформації (Routing Information Protocol - RIP). Маршрутизатор 2 вибирає шлях до мережі A, перевіряючи число переходів для кожного наявного шляху.

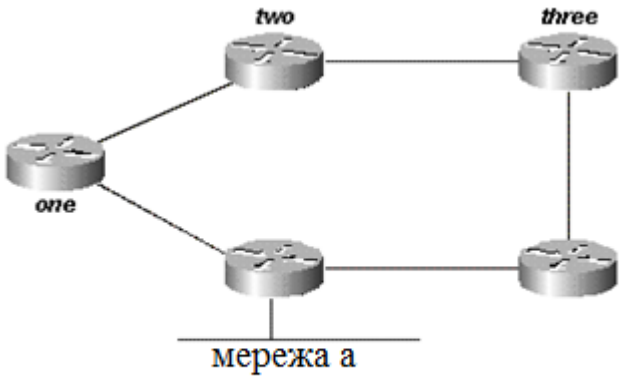


Рис.1

Оскільки довжина шляху, що проходить через Маршрутизатор 3, дорівнює трьом переходам, а довжина шляху через маршрутизатор 1 дорівнює двом переходам, маршрутизатор 2 вибирає шлях, що йде через Маршрутизатор 1, і відкидає інформацію, отриману від Маршрутизатора 3. Якщо маршрут, який пролягає між маршрутизатором 1 і мережею А, порушується, то маршрутизатор 2 втрачає зв'язок з цим пунктом призначення доти, поки цей маршрут не блокуватиме в таблиці маршрутизації по часу простою, яке дорівнює трьом періодам оновлення (90 секунд); а маршрутизатор 3 оголошує маршрут повторно (що відбувається в RIP кожні 30 секунд). Якщо не враховувати період утримання, то маршрутизатора 2 буде потрібно 90-120 секунд, щоб перевести шлях з маршрутизатора 1 на маршрутизатор 3.

Замість того, щоб розраховувати на повні регулярні оновлення для виконання повторної збіжності, EIGRP (замість відкидання даних) будує таблицю топології використовуючи для цього всі оголошення своїх сусідів і виконує збіжність або за допомогою пошуку підходящого беспетлевого маршруту в таблиці топології, або (якщо про такий маршруті нічого невідомо) за допомогою опитування своїх сусідів. Другий маршрутизатор зберігає інформацію, отриману від першого і третього маршрутизаторів. Він вибирає шлях через маршрутизатор 1 як кращий шлях ("наступник"), а шлях через маршрутизатор 3 - як шлях без петель (можливий наступник). Коли маршрут через маршрутизатор 1 стає недоступним, маршрутизатор 2 в пошуках можливого наступника звертається до таблиці топології і відразу ж починає використовувати маршрут через маршрутизатор 3.

З цих коротких пояснень очевидно, що EIGRP повинен забезпечити наступне:

- систему, при якій EIGRP пересилає оновлення, необхідні в даний момент (це досягається за допомогою виявлення та обслуговування сусідів)

- спосіб, що дозволяє визначити, який з маршрутів, отриманий маршрутизатором, є беспетлевого
- процедуру, що дозволяє видаляти неробочі маршрути з таблиць топології на всіх маршрутизаторах, що знаходяться в мережі
- процедуру опитування сусідів, яка дозволить знайти нові маршрути до кінцевого адресою замість втрачених старих маршрутів

Всі ці вимоги будуть розглянуті далі.

Виявлення та обслуговування сусідів

Для поширення маршрутної інформації по мережі в EIGRP використовується неперіодичне інкрементне оновлення маршрутів. Це означає, що EIGRP пересилає оновлення тільки для змінюються маршрутів і тільки в той момент, коли такі маршрути змінюються.

Основний недолік таких оновлень полягає в тому, що ви можете не впізнати, в який час маршрут, що проходить через сусідній маршрутизатор, став недоступним. Не можна блокувати за часом маршрути, які очікують отримання нової таблиці маршрутизації від сусіда. Для надійної пересилання змін до маршрутної таблиці в EIGRP використовується механізм взаємодії між сусідніми маршрутизаторами (два маршрутизатора стають "сусідами" в тому випадку, якщо кожен з них отримує пакети "hello" від свого сусіда).

EIGRP посилає пакети "hello" кожні 5 секунд (для каналів з високою пропускною здатністю) і кожні 60 секунд (для багатоточкових каналів з низькою пропускною здатністю).

- п'ятисекундними пакет "hello" використовується:
  - в мережевих середовищах (наприклад, Ethernet, Token Ring і FDDI)
  - в послідовних каналах "точка-точка" (наприклад, виділені лінії, що використовують протоколи PPP або HDLC; підлеглі інтерфейси Frame Relay типу "точка-точка" і підлеглий інтерфейс ATM
  - багатоточкові лінії з високою пропускною здатністю (вище, ніж у лінії T1) (наприклад, ISDN PRI і Frame Relay)
- Шестидесяти секундний пакет "hello ":
  - багатоточкові лінії з пропускною здатністю як у T1 або нижче - як у багатоточкових інтерфейсів Frame Relay, багатоточкових інтерфейсів ATM, комутованих віртуальних каналів ATM і базових інтерфейсів обміну ISDN

Частота, з якою EIGRP відправляє пакети "hello", називається hello-інтервалом; цей параметр можна налаштовувати для кожного інтерфейсу окремо за допомогою команди ip hello-interval eigrp. Час утримання - час, протягом якого маршрутизатор буде вважати сусіднє пристрій чинним, не отримуючи при цьому від нього пакет "hello". Час утримання зазвичай дорівнює трьом hello-інтервалам (15 і 180 секунд за замовчуванням). Час утримання налаштовується за допомогою команди ip hold-time eigrp.

Зверніть увагу на те, що при зміні hello-інтервалу час утримання не коригується автоматично (час утримання необхідно змінити вручну відповідно до новим значенням hello-інтервалу).

Два EIGRP-маршрутизатора можуть стати сусідами навіть в тому випадку, якщо таймер hello і таймер утримання не збігаються. Час утримання вказується в пакетах hello, тому кожен сусід повинен залишатися в робочому стані, навіть якщо інтервал hello і таймер утримання не збігаються.

Величину hello-інтервалу, встановлену на маршрутизаторі, неможливо визначити безпосередньо. Однак цю величину можна з'ясувати з лістингу команди show ip eigrp neighbor, виконаної на сусідньому маршрутизаторе.

Якщо лістинг команди show ip eigrp neighbor отримано з пристрою Cisco, то в цьому випадку для інтерпретації результатів можна скористатися утилітою Output Interpreter (тільки для зареєстрованих клієнтів). Дана утиліта відображає потенційні проблеми і пропонує способи їх вирішення. Для використання Output Interpreter необхідно включити JavaScript.router# **show ip eigrp neighbor**

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num	Type
1	10.1.1.2	Et1	13	12:00:53	12	300	0	620	
0	10.1.2.2	S0	174	12:00:56	17	200	0	645	

rp-2514aa# **show ip eigrp neighbor**

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num	Type
1	10.1.1.2	Et1	12	12:00:55	12	300	0	620	
0	10.1.2.2	S0	173	12:00:57	17	200	0	645	

```
rp-2514aa# show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address      Interface   Hold Uptime   SRTT   RTO  Q  Seq  Type
                               (sec)         (ms)       Cnt Num
1   10.1.1.2     Et1          11 12:00:56   12   300  0  620
0   10.1.2.2     S0          172 12:00:58   17   200  0  645
```

Значення, вказане в лістингу команди в стовпці "Hold", ніколи не повинно перевищувати значення часу утримання, а також не повинно бути менше різниці значень часу утримання та інтервалу між повідомленнями привітання (якщо, звичайно, пакети вітання не були втрачені). Якщо значення в стовпці "Hold" зазвичай варіюється від 10 до 15 секунд, то інтервал вітання становить 5 секунд, а час утримання - 15 секунд. Якщо значення в стовпці "Hold" зазвичай варіюється від 120 до 180 секунд, то інтервал вітання становить 60 секунд, а час утримання - 180 секунд. Якщо значення не збігаються зі значеннями таймера, заданими за умовчанням, тоді необхідно перевірити відповідний інтерфейс на сусідньому маршрутизаторе (таймер "hello" і таймер інтервалу вітання могли бути змінені вручну).

Примітка:

- EIGRP не створює однорангові зв'язку з використанням вторинних адрес. Джерело всього трафіку EIGRP - основний адресу інтерфейсу.
- При налаштуванні EIGRP по мережі Frame Relay з множинним доступом (багатоточковому з'єднанню і ін.) В операторахframe-relay map слід налаштувати ключове слово broadcast. Суміжність двох маршрутизаторів EIGRP не буде створена без ключового слова broadcast. Більш детально див. Розділ "Налаштування та усунення неполадок Frame Relay".
- Кількість сусідніх вузлів, підтримуваних EIGRP, не обмежена. Фактична кількість підтримуваних сусідніх вузлів залежить від характеристик пристрою:
  - обсяг пам'яті
  - обчислювальна потужність
  - обсяг переданої інформації (наприклад, кількість переданих маршрутів)
  - рівень складності топології
  - стабільність мережі

Побудова таблиці топології

Тепер ми знаємо, яким чином маршрутизатори спілкуються між собою. А ось про що вони "кажуть" ?. Звичайно - про таблиці топології. На відміну від протоколів RIP і IGRP, EIGRP не використовує таблицю маршрутизації (або перенаправлення) для зберігання всіх даних, необхідних для його роботи. Натомість EIGRP формує другу таблицю (таблицю топології), на основі якої здійснюється установка маршрутів в таблиці маршрутизації.

Примітка: В Cisco IOS починаючи з версій 12.0T і 12.1 протокол RIP веде свою власну базу даних, звідки маршрути надходять в таблицю маршрутизації.

Щоб дізнатися основний формат таблиці топології на EIGRP-маршрутизатора, виконайте команду show ip eigrp topology. У таблиці топології містяться відомості, необхідні для побудови набору відстаней і векторів для кожної досяжною мережі, включаючи:

- найменша пропускна здатність на маршруті, що йде до даної мережі (дані надаються висхідним сусідом
- сумарна затримка
- надійність маршруту
- завантаження маршруту
- мінімальний шлях максимального розміру переданого блоку даних (MTU)
- можлива відстань
- оголошене відстань
- джерело маршруту (зовнішні вузли маркуються)

Імовірне й фактичне відстані обговорюватимуться далі в цьому розділі.

Якщо лістинг команди show ip eigrp topology отримано з пристрою Cisco, то в цьому випадку для інтерпретації результатів можна скористатися утилітою Output Interpreter (тільки для зареєстрованих клієнтів). Дана утиліта відображає потенційні проблеми і пропонує способи їх вирішення. Для використання Output Interpreter необхідно включити JavaScript.

Метрики EIGRP

Для розрахунку метрик маршрутизації протокол EIGRP використовує мінімальну пропускну здатність маршруту до кінцевого адреси, а також сумарну затримку. Можна також налаштувати й інші метрики. Проте ми не рекомендуємо робити цього, оскільки в цьому випадку у вашій мережі можуть з'явитися петлі по маршрутизації. Метрики пропускної здатності і

затримки визначаються на основі значень, встановлених на інтерфейсах маршрутизаторів, які є частиною маршруту до мережі призначення.

На малюнку 2 показано, як маршрутизатор 1 обчислює найкращий шлях в мережу А.

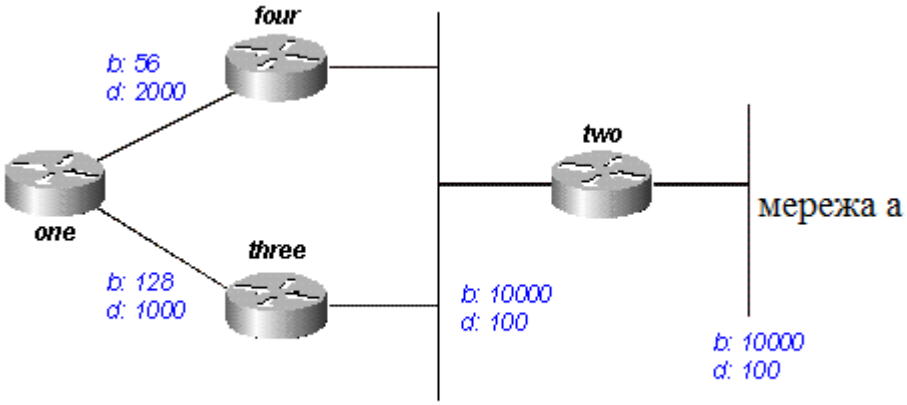


Рисунок 2

Розрахунок починається з двох оголошень, що стосуються шуканої мережі: перший маршрут проходить через маршрутизатор 4 (мінімальна пропускна здатність - 56 і сумарна затримка - 2200), а другий - через маршрутизатор 3 (мінімальна пропускна здатність - 128 і сумарна затримка - 1200). Маршрутизатор 1 вибирає маршрут з найменшим значенням метрики.

Тепер обчислимо метрику. EIGRP обчислює сумарну метрику шляхом зважування метрики пропускної здатності і метрики затримки. Для вимірювання пропускної здатності EIGRP використовує наступну формулу:

- пропуск. способн. = (10000000 / пропуск. Способн. (I)) * 256

де пропускна спроможність (i) - найменша пропускна спроможність усіх вихідних інтерфейсів, що входять до складу маршруту, що веде до мережі призначення (вимірюється в кілобітах).

Для обчислення затримки EIGRP використовує наступну формулу:

- затримка = затримка (i) * 256

де затримка (i) - сума затримок, які налаштовані на всіх інтерфейсах, що входять до складу маршруту до мережі призначення (вимірюється в десятках мікросекунд). У лістингу команд show ip eigrp topology і show interface затримка вказана в мікросекундах, тому для використання цих значень у формулі їх необхідно розділити на 10. У цьому документі використовується затримка, налаштована і показана на інтерфейсі.

Зазначені вище значення використовуються для визначення сумарної метрики для шуканої мережі:

- метрика = [K1 * пропускна здатність + (K2 * пропускна здатність) / (256 - навантаження) + K3 * затримка] * [K5 / (надійність + K4)]

Примітка: значення K слід використовувати тільки після ретельного планування. Неправильні значення K призводять до неможливості побудови зв'язків між сусідніми вузлами, в результаті чого у вашій мережі буде неможливо виконати збіжність.

Примітка: при K5 = 0 формула приймає наступний вигляд: метрика = [k1 * пропускна здатність + (k2 * пропускна здатність) / (256 - навантаження) + k3 * затримка].

За замовчуванням K має наступні значення:

- K1 = 1
- K2 = 0
- K3 = 1
- K4 = 0
- K5 = 0

Для функціонування за замовчуванням формулу формулу можна спростити наступним чином:

metric = bandwidth + delay

Маршрутизатори Cisco не виконують розрахунки з плаваючою комою, тому для правильного підрахунку метрики на кожному етапі калькуляції отриманий результат необхідно округляти в меншу сторону до цілого числа. В цьому прикладі загальна вартість маршруту через маршрутизатор 4 дорівнює:

minimum bandwidth = 56k

total delay = 100 + 100 + 2000 = 2200

[(10000000/56) + 2200] x 256 = (178571 + 2200) x 256 = 180771 x 256 = 46277376

А загальна вартість маршруту через маршрутизатор 3 дорівнює:

minimum bandwidth = 128k

total delay = 100 + 100 + 1000 = 1200

[(10000000/128) + 1200] x 256 = (78125 + 1200) x 256 = 79325 x 256 = 20307200

Таким чином, щоб досягти мережі A, маршрутизатор 1 вибирає маршрут через маршрутизатор 3.

Зверніть увагу на те, що значення пропускної здатності і затримки - це значення, налаштовані на інтерфейсі, через який маршрутизатор досягає наступного сегмента по шляху до мережі призначення. Наприклад, маршрутизатор 2 оголосив про мережу A і вказав затримку, налаштовану на його Ethernet-інтерфейсі. Потім маршрутизатор 4 додав до цього значення затримку, налаштовану на його Ethernet-інтерфейсі, а маршрутизатор 1 додав затримку, налаштовану на його послідовному порту.

Можлива відстань, оголошена відстань і можливий наступник

Можлива відстань - це найкраща метрика по маршруту в мережу призначення, включаючи метрику до сусіда, який оголосив маршрут. Оголошене відстань є сумарною метрикою шляху до мережі призначення, який оголошений висхідним сусідом. Можливий наступник - це маршрут, величина оголошеного відстані якого менше величини можливої відстані (поточний найкращий маршрут). На рис. 3 зображений цей процес:

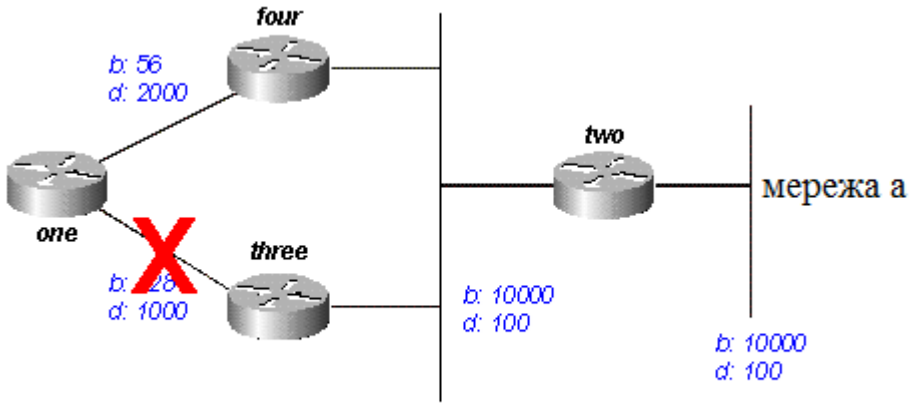


рисунок 3

Маршрутизатор 1 визначив, що в мережу A маються два маршрути: один маршрут проходить через маршрутизатор 3, а другий - через маршрутизатор номер 4.

- Вартість маршруту, що проходить через маршрутизатор 4, дорівнює 46277376, а оголошене відстань - 307 200.
- Вартість маршруту, що проходить через маршрутизатор 3, дорівнює 20307200, оголошене відстань дорівнює 307200.

Зверніть увагу, що в кожному випадку EIGRP відраховує оголошене відстань починаючи з маршрутизатора, що оголосив маршрут до шуканої мережі. Іншими словами, оголошене відстань від маршрутизатора 4 - це

метрика маршруту до мережі A, який починається з маршрутизатора 4, а оголошене відстань від маршрутизатора 3 - це метрика маршруту до мережі A починаючи з маршрутизатора 3. EIGRP вибирає маршрут через маршрутизатор 3 в якості найкращого і використовує метрику через маршрутизатор 3 як можливу відстань. Оскільки оголошене відстань до цієї мережі через маршрутизатор 4 менше можливої відстані, маршрутизатор 1 вважає шлях через маршрутизатор 4 можливим наступником.

Коли зв'язок між маршрутизаторами 1 і 3 порушується, маршрутизатор 1 перебирає всі відомі йому шляхи до мережі A, і визначає, що у нього є можливий наступник, доступний через маршрутизатор 4. Маршрутизатор 1 звертається до цього маршруту, використовуючи метрику через маршрутизатор 4 в якості нового можливої відстані. Збіжність виконується миттєво, а оновлення, що надходять до низхідним сусідам, є єдиним трафіком, який генерується протоколом маршрутизації.

Тепер давайте розглянемо більш складний випадок, показаний на рисунку 4.

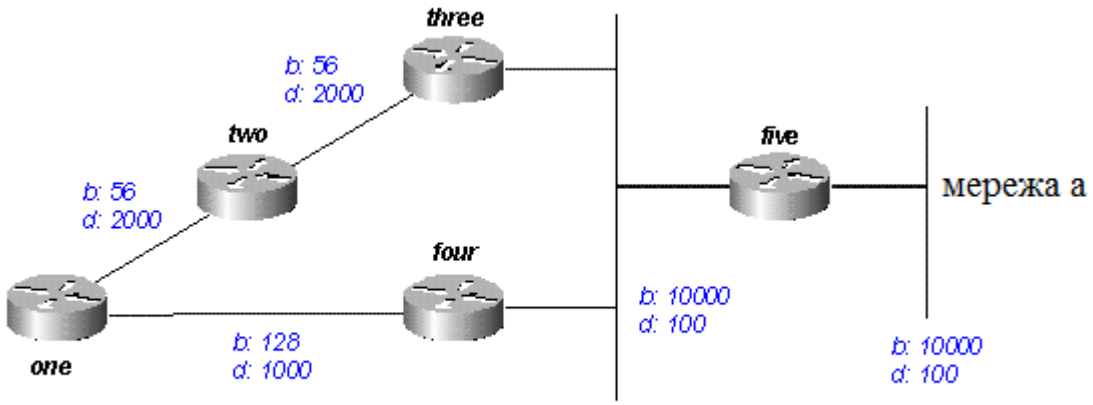


рисунок 4

Від маршрутизатора 1 в мережу A ведуть два маршрути: один маршрут проходить через маршрутизатор і має метрику 46789376, а другий через маршрутизатор 4 і має метрику 20307200. В якості маршруту в мережу A маршрутизатор 1 вибере найменшу з двох метрик, яка стане можливим відстанню. Тепер розглянемо маршрут, що йде через маршрутизатор 2. Нам необхідно з'ясувати, чи підходить цей маршрут в якості можливого наступника. Оголошене відстань від другого маршрутизатора одно 46277376. Це значення вище значення можливої відстані, тому цей шлях не є можливим наступником. Якби ми подивилися таблицю топології маршрутизатора 1 (за допомогою команди show ip eigrp topology), то ми б побачили тільки одну запис для мережі A - через маршрутизатор 4. (В дійсності в таблиці топології маршрутизатора 1 маються два записи, але можливим наступником може бути тільки одна з

них, інша ж запис в лістингу команди show ip eigrp topology відображено не буде; всі маршрути, які не є можливими наступниками, можна подивитися скориставшись командою show ip eigrp topology all-links).

Припустимо, що з'єднання між маршрутизаторами 3 і 4 було порушено. Виявивши, що єдиний шлях до мережі А втрачений, маршрутизатор 1 починає опитувати всіх своїх сусідів на предмет маршруту в мережу А. Оскільки маршрутизатор 2 може запропонувати маршрут в мережу А, він відгукнеться на запит, ініційований маршрутизатором 1. Оскільки найкращий маршрут, що йде через маршрутизатор 4, був втрачений, маршрутизатор 1 приймає маршрут, який йде в мережу А через маршрутизатор 2.

Визначення беспетлевого шляху

Яким чином протокол EIGRP використовує поняття можливої відстані, оголошеного відстані і можливого наступника для того, щоб визначити, чи є даний шлях дійсним і беспетлевого? На малюнку 4а показано, що маршрутизатор 3 виконує оцінку маршрутів в мережу А. Оскільки функція "розділений горизонт" відключена (наприклад, це необхідно при роботі з багатоточковими інтерфейсами Frame Relay), маршрутизатор 3 вкаже в мережу А три маршрути: через маршрутизатор 4, через маршрутизатор 2 (шлях проходить через маршрутизатор два, один, три і чотири) і через маршрутизатор 1 (шлях проходить через маршрутизатор один, два, три, чотири).

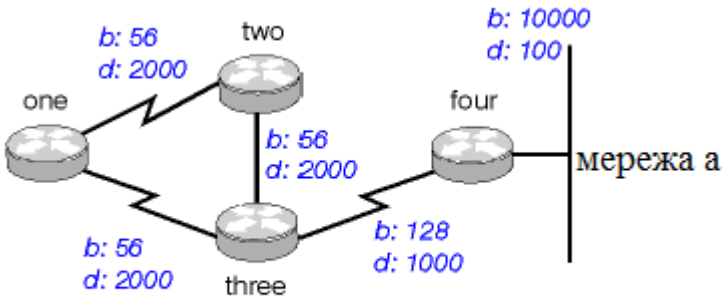


рисунок 4а

Якщо маршрутизатор прийме всі ці маршрути, то утворюється маршрутна петля. Маршрутизатор 3 вважає, що він зможе потрапити в мережу А через маршрутизатор 2, однак перш ніж потрапити в мережу А, шлях, що йде через маршрутизатор 2, проходить через маршрутизатор 3. Якщо з'єднання між маршрутизатором 4 і маршрутизатором 3 буде порушено, тоді маршрутизатор 3 буде вважати, що він зможе досягти мережі А використовуючи один з решти маршрутів, але через дію правил

для визначення можливих наступників він ніколи не використовуватиме ці маршрути в якості альтернативних. Щоб це зрозуміти, звернемося до метриці:

- сумарна метрика в мережу А через маршрутизатор 4: 20281600
- сумарна метрика в мережу А через маршрутизатор 3: 47019776
- сумарна метрика в мережу А через маршрутизатор 1: 47019776

Оскільки шлях через маршрутизатор 4 володіє найкращою метрикою, маршрутизатор 3 встановлює цей маршрут в таблиці перенаправлень і використовує метрику 20281600 в якості можливого відстані в мережу А. Потім маршрутизатор 3 обчислює оголошене відстань в мережу А для шляхів, що йдуть через маршрутизатори 2 і 1: 47019776 відповідає шляху через маршрутизатор 2, а 47019776 відповідає шляху через маршрутизатор 1. Оскільки значення цих метрик перевищує значення можливої відстані, маршрутизатор 3 не призначила жоден з цих маршрутів в якості можливих наступників, провідних в мережу А.

Припустимо, що зв'язок між маршрутизаторами 3 і 4 була порушена. Маршрутизатор 3 запрошувати альтернативний маршрут до мережі А у всіх сусідніх вузлів. Маршрутизатор 2 отримує запит і, оскільки цей запит надходить від наступника, виконує пошук всіх записів про можливих наступників в таблиці топології. Єдина така запис в таблиці топології належить маршрутизатора 1 (оголошене відстань цьому записі дорівнює значенню останньої відомої метриці через маршрутизатор 3). Оскільки оголошене відстань через маршрутизатор 1 більше ніж останнє відоме можливу відстань, маршрутизатор 2 позначає цей маршрут як недосяжний і починає опитувати своїх сусідів (в цьому випадку опитується тільки маршрутизатор 1) на предмет шляху до мережі 1.

Маршрутизатор 3 відправляє маршрутизатора 1 запит щодо мережі А. Маршрутизатор 1 виконує пошук в своїй таблиці топології і виявляє, що тільки другий єдиний маршрут до мережі А проходить через маршрутизатор 2 з оголошеним відстанню, рівним останнім відомим можливого віддалі через маршрутизатор 3. Оскільки оголошене відстань через маршрутизатор 2 знову-таки перевищує останнє відоме можливу відстань, цей маршрут не є можливим наступником. Маршрутизатор 1 позначає цей маршрут як недосяжний і запитує в єдиного, що залишився сусіда (маршрутизатор 2) шлях до мережі А.

Намагаючись знайти маршрут в мережу А маршрутизатор 3 опитав усіх своїх сусідів. В свою чергу маршрутизатори 1 і 2 позначили маршрут як недосяжний і опитали своїх сусідів, щоб знайти шлях в мережу А. При отриманні запиту від маршрутизатора 1 маршрутизатор 2 виконує пошук по

своїй таблиці топології, в ході якого виявляє, що пункт призначення позначений як недосяжний. Маршрутизатор 2 відповідає маршрутизатора 1, що мережа А недоступна. Коли маршрутизатор 1 отримує запит від маршрутизатора 2, він також посилає назад відповідь про те, що мережа А недоступна. Оскільки маршрутизатори 1 і 2 встановили, що мережа А чи не доступна, вони відповідають на початковий запит маршрутизатора 3. Збіжність закінчена, і всі маршрути переходять в пасивний стан.

Розщеплений горизонт і зворотний заборона

В попередньому прикладі ми обмовилися, що функція "розщеплений горизонт" була вимкнена (ми хотіли показати, яким чином EIGRP використовує можливу відстань і оголошене відстань, щоб визначити, чи є маршрут петлею). Проте в деяких випадках EIGRP використовує розщеплений горизонт, щоб запобігти виникнення петель маршрутизації. Перш ніж почати розглядати особливості використання розщепленого горизонту, необхідно пояснити, що це таке і як це працює. Правило розщепленого горизонту виглядає наступним чином:

- ніколи не оголошувати маршрут через інтерфейс, за допомогою якого маршрутизатор дізнався про маршрут.

Наведемо приклад. На малюнку 4а показано, що якщо маршрутизатор 1 підключений до маршрутизаторів 2 і 3 через єдиний многоточечний інтерфейс (наприклад, Frame Relay) і якщо при цьому маршрутизатор 1 дізнається про мережу А через маршрутизатор 2, то він не буде використовувати той же самий інтерфейс, щоб оголосити маршрутизатора 3 маршрут в мережу А. маршрутизатор 1 припускає, що маршрутизатор 3 отримає інформацію про мережі А безпосередньо від маршрутизатора 2.

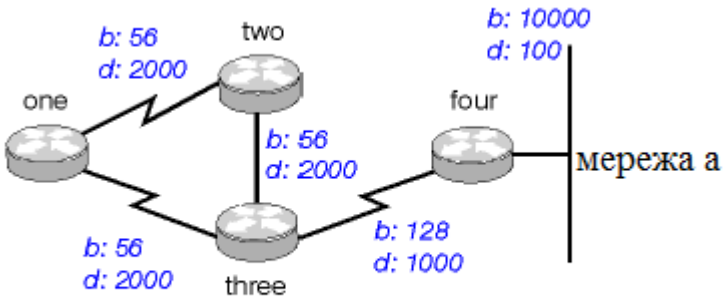


рисунок 4а

Зворотний заборона - це ще один спосіб запобігти появі петель. Правило зворотного заборони виглядає наступним чином:

- якщо інформація про маршрут надійшла через який-небудь інтерфейс, то цей маршрут необхідно оголосити недосяжним через той же самий інтерфейс.

Припустимо, що на маршрутизаторах, показаних на рисунку 4а, включена функція зворотного заборони .. Коли маршрутизатор 1 дізнається від маршрутизатора 2 про мережу А, він оголошує маршрутизаторам 2 і 3, що мережа А недосяжна через його канал. У разі якщо маршрутизатор 3 вказує якоїсь шлях в мережу А через маршрутизатор 1, він повинен видалити його, оскільки було оголошено про те, що мережа А недосяжна через цей шлях. EIGRP використовує два цих правила, щоб запобігти виникненню петель маршрутизації.

EIGRP використовує розщеплений горизонт і оголошує маршрут недосяжним в наступних випадках:

- якщо два маршрутизатора працюють в режимі ініціалізації (тобто виконують первинний обмін таблицями топології)
- якщо оголошується про зміну в таблиці топології
- при відправці запиту

Тепер розглянемо кожен з цих випадків.

режим ініціалізації

Коли два маршрутизатора вперше стають "сусідами", вони, перебуваючи в режимі ініціалізації, обмінюються таблицями топології. Кожен запис у таблиці, яку маршрутизатор прийняв в режимі ініціалізації, повторно оголошується (приймаючим маршрутизатором) новому сусідові, при цьому до цього запису застосовується максимальна метрика (це так званий "заборонений маршрут" - poison route, букв. Отруєний маршрут).

Зміни в таблиці топології

На рис. 5 в маршрутизаторі 1 використовується дисперсія для балансування трафіку, призначеного для мережі А на відрізку між двома послідовними каналами (канал в 56 Кбіт між маршрутизаторами 2 і 4 і канал в 128 Кбіт між маршрутизаторами 3 і 4). (Докладно про дисперсії см. Балансування навантаження, розділ, в якому обговорюється дисперсія).

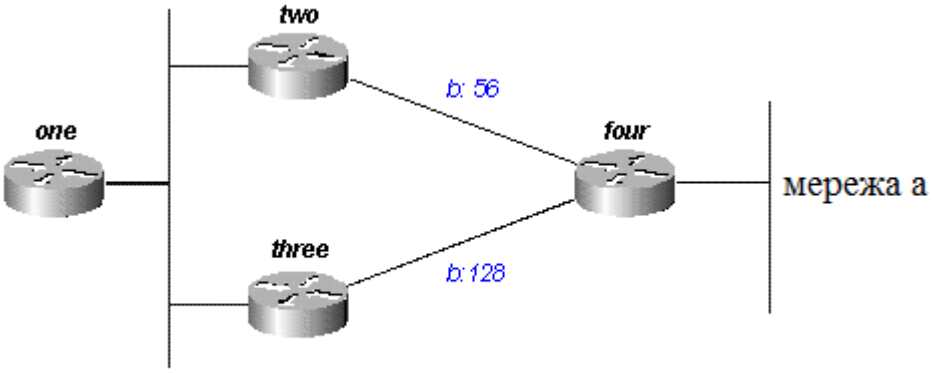


рисунок 5

Маршрутизатор 2 розглядає шлях через маршрутизатор 3 в якості можливого наступника. Якщо канал між маршрутизаторами 2 і 4 буде порушений, то маршрутизатор 2 виконає збіжність по маршруту, що проходить через маршрутизатор 3. Оскільки правило розщепленого горизонту свідчить, що не можна оголошувати маршрут через інтерфейс, за допомогою якого були отримані відомості про цей маршрут, маршрутизатор 2 зазвичай не передає оновлення. Однак це залишає маршрутизатор 1 з неприпустимою записом в таблиці топології. Коли маршрутизатор змінює свою таблицю топології таким чином, що при цьому відбувається зміна інтерфейсу, через який маршрутизатор з'єднується з мережею, то він відключає розщеплений горизонт, а виправлення міняють всі маршрути, які виходять із інтерфейсів, в зворотному напрямку. В цьому випадку маршрутизатор 2 відключає розщеплений горизонт для даного маршруту і оголошує мережу А недоступною. При отриманні цього оголошення Маршрутизатор 1 видаляє зі своєї таблиці маршрутизації свій маршрут до мережі А, що йде через маршрутизатор 2.

Запити.

Запити призводять до виникнення розщепленого горизонту тільки тоді, коли маршрутизатор отримує запит або оновлення від наступника, якого він використовує для одержувача в запиті. Розглянемо мережу, показану на рис. 6.

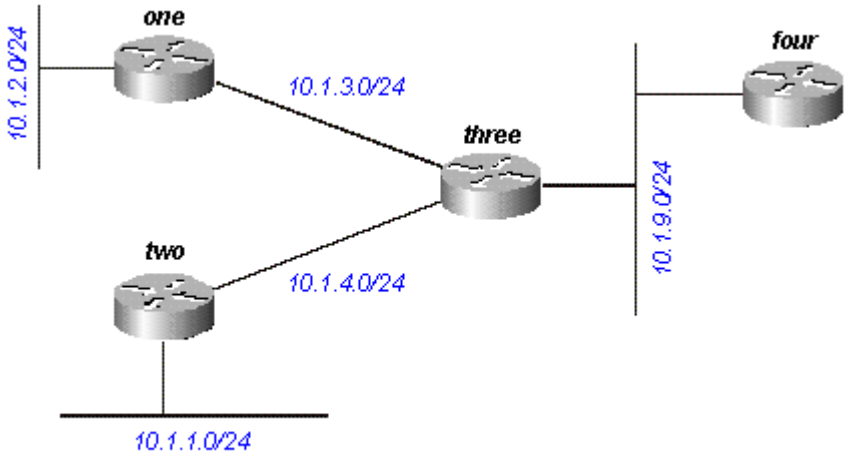


рисунок 6

Маршрутизатор 3 отримує від маршрутизатора 4 запит про мережу 10.1.2.0/24 (якої він досягає через маршрутизатор 1). Якщо у маршрутизатора 3 відсутній наступник для даного пункту призначення (з причини перемикання лінії або через інший тимчасової ситуації в мережі), цей маршрутизатор відправляє запит кожному зі своїх сусідів (в даному випадку це маршрутизатори 1, 2 і 4). Однак якщо маршрутизатор 3 отримає від маршрутизатора 1 запит або оновлення (наприклад, зміна метрики) для мережі 10.1.2.0/24, той він не відправлятиме запит назад маршрутизатора 1, оскільки останній є його наступником на маршруті до цієї мережі. Замість цього він відправить запити тільки маршрутизаторам 2 і 4.

Затримка в активних маршрутах

При певних обставинах може пройти велику кількість часу, перш ніж на запит буде отримана відповідь. "Мовчання" може бути настільки тривалим, що ініціював запит маршрутизатор може припинити очікування і скинути з'єднання з "мовчазним" маршрутизатором, при цьому відбувається фактичний перезапуск сеансу зв'язку з сусіднім вузлом. Така подія називається "затор на активному маршруті" (stuck in active - SIA). Найбільш прості прості випадки SIA виникають, якщо потрібно занадто багато часу для досягнення запитом іншого кінця мережі, а також для зворотного проходження відгуку. Наприклад, на рис. 7 маршрутизатор 1 записує велику число маршрутів SIA від маршрутизатора 2.

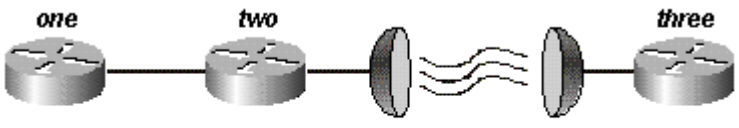


рисунок 7

Після проведення перевірок виявляється, що проблема зводиться до наявності затримки в супутниковому каналі між маршрутизаторами 2 і 3. Існують два можливих рішення проблеми цього типу. Перше - збільшити інтервал, протягом якого маршрутизатор буде очікувати відповіді на запит, перш ніж оголосити маршрут маршрутом SIA. Цю настройку можна змінити за допомогою команди timers active-time.

Однак оптимальне рішення проблеми полягає в тому, щоб перекомпонувати мережу таким чином, щоб зменшити діапазон запитів (щоб по супутниковому каналу передавалося мінімальна кількість запитів). Це питання розглядається в розділі Діапазон запитів. Проблема діапазону запитів не є поширеною причиною виникнення SIA маршрутів. Значно частіше деякі маршрутизатори в мережі не можуть відповісти на запит по одній з наступних причин:

- маршрутизатор сильно завантажений роботою і не може відповісти на запит (зазвичай це пов'язано з сильною завантаженням ЦП)
- у маршрутизатора є труднощі при роботі з пам'яттю, і він не може виділити пам'ять для обробки запиту або для формування пакета відгуку
- канал між двома маршрутизаторами не забезпечує достатньої якості (це значить, що по такому каналу і раніше можна передавати пакети в кількості, достатній для підтримання зв'язку "сусід-сусід", однак при цьому деякі запити або відгуки все-таки губляться при пересиланні від одного маршрутизатора до іншого)
- односпрямовані канали (тобто канали, в яких через збій трафік може передаватися тільки в одному напрямку)

Усунення SIA маршрутів

Процес усунення маршрутів SIA зазвичай складається з трьох етапів:

1. Спочатку необхідно встановити маршрути, про які система з разу в раз повідомляє як про маршрутах SIA.
2. Потім необхідно встановити маршрутизатор, який з разу в раз не відповідає на запити на даних маршрутах.
3. Потім необхідно визначити причину, по якій маршрутизатор не отримує запитів або не відповідає на них.

Перший крок виконується відносно просто. Якщо у вашій системі ведеться облік консольних повідомлень, то при швидкому перегляді журналу можна визначити, які маршрути найчастіше позначаються як маршрути SIA. Другий крок більш складний. Для збору цієї інформації необхідно використовувати команду show ip eigrp topology active:

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

A 10.2.4.0/24, 0 successors, FD is 512640000, Q
   1 replies, active 00:00:01, query-origin: Local origin
        via 10.1.2.2 (Infinity/Infinity), Serial1
   1 replies, active 00:00:01, query-origin: Local origin
        via 10.1.3.2 (Infinity/Infinity), r, Serial3
   Remaining replies:
        via 10.1.1.2, r, Serial0
```

Все сусідні вузли, для яких відображена буква R, ще тільки повинні відповісти (активний таймер показує час, протягом якого маршрут є активним). Зверніть увагу на те, що сусідні вузли можуть бути відсутні в розділі "Remaining replies" (очікувані відгуки). Ці вузли можуть з'явитися серед інших RDB. Особливу увагу необхідно приділяти маршрутам, які були активні протягом певного часу (зазвичай 2-3 хвилини) і відповідь від яких тільки очікується. Виконайте цю команду кілька разів, щоб з'ясувати, які сусіди не відповідають на запити або які інтерфейси мають багато пропущених запитів. Перевірте цей сусідній вузол, щоб дізнатися, чи знаходиться він постійно в режимі очікування відгуків від своїх сусідів. Повторюйте цю процедуру доти, поки не знайдете маршрутизатор, який з разу в раз не відповідає на запити. Можна також пошукати неполадки, що відносяться до з'єднання з цим сусідом, до пам'яті, а також неполадки, пов'язані із завантаженням ЦП та іншими проблемами, наявними у даного сусіда.

Якщо ви стикаєтеся з ситуацією, коли на вашу думку причиною неполадок є діапазон запитів, кращим рішенням проблеми буде зменшити діапазон запиту (а не збільшувати значення таймера SIA).

Перерозподіл

Даний розділ містить опис різних випадків перерозподілу. Будь ласка, пам'ятайте про те, що в прикладах, розглянутих нижче, вказаний мінімум, який необхідний для настройки перерозподілу. Перерозподіл пов'язано з потенційними проблемами, такими як: неоптимальна маршрутизація, освіта петель або повільна збіжність. Щоб уникнути подібних проблем, зверніться до розділу "Як уникнути проблем при перерозподілі" в документі "Перерозподіл протоколів маршрутизації".

Завдання на лабораторну роботу

1. підключення та налаштування з'єднань WAN;
2. настройка EIGRP для оголошення конкретних мереж;
3. дослідження конвергенції мережі з вікна інтерфейсу командного рядка при відключенні і повторному включенні інтерфейсу;
4. перевірка пакетів EIGRP в режимі моделіроваяв процесі конвергенції мережі.

Вихідні дані

Вам пропонується топологія з уже налаштованими вузлами HQ, Branch1, Branch2 і Branch3. В неї додано новий частково налаштований маршрутизатор (New_Branch). Маршрутизатор New_Branch потрібно підключити до HQ і Branch1, закінчити настройку нового маршрутизатора і перевірити конвергенцію мережі.

Крок 1. Підключення та настройка з'єднання WAN для маршрутизатора New_Branch

- Підключіть інтерфейс S0 / 0/0 маршрутизатора New_Branch до інтерфейсу S0 / 1/1 маршрутизатора HQ (DCE)
- Підключіть інтерфейс S0 / 0/1 маршрутизатора New_Branch до інтерфейсу S0 / 1/1 маршрутизатора Branch1 (DCE)
- Налаштуйте інтерфейс S0 / 0/0, використовуючи IP-адресу 172.16.3.218/30
- Налаштуйте інтерфейс S0 / 0/1, використовуючи IP-адресу 172.16.3.221/30

Крок 2. Налаштування EIGRP для оголошення конкретної мережі на маршрутизаторі New_Branch

- Налаштуйте на маршрутизаторі New_Branch протокол EIGRP і виберіть номер анонімної системи 3
- Оголосіть прямо підключення мережі

Крок 3. Спостереження за конвергенцією мережі в режимі реального часу

- У вікні CLI для маршрутизатора New_Branch можна спостерігати за конвергенцією в режимі реального часу. В процесі конвергенції буде видно, як EIGRP створює суміжності.

- Після завершення конвергенції відключіть інтерфейс S0 / 0/0 маршрутизатора New_Branch.
- Простежте за змінами мережі
- Знову включите інтерфейс S0/0/0

Крок 4. Спостереження за конвергенцією мережі в режимі моделювання

- Перейдіть в режим моделювання
- Налаштуйте фільтри списку подій так, щоб відображались тільки пакети EIGRP
- Відкрийте вікно CLI для маршрутизатора New_Branch
- Вимкніть інтерфейс S0/0/1
- Натисніть кнопку Auto Capture / Play, щоб почати моделювання
- Знову відкрийте вікно CLI і подивіться на результати
- Почекайте трохи і ще раз натисніть кнопку Auto Capture / Play, щоб призупинити моделювання
- Перевірте деякі пакети в списку Event List

Крок 5. Перевірка результату включення інтерфейсу

- Повторно запустіть моделювання, натиснувши кнопку Auto Capture / Play.
- Увімкніть інтерфейс S0 / 0/1 і простежте за ходом конвергенції за допомогою інтерфейсу командного рядка, списку подій і топології
- Зупиніть моделювання

Контрольні питання

1. Що відображалося в інтерфейсі командного рядка при конвергенції протоколу EIGRP?
2. Що сталося з пакетами EIGRP в нової локальної мережі при відключенні каналу WAN, яка зв'язує HQ і New_Branch?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кулаков Ю.О., Жуков І.А. Навчальний посібник «Комп’ютерні мережі», Київ 2008

2. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.

3. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.

4. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.

5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.:Питер, 1999.

6. CISCO Internetworking technology overview – Cisco, 1999.