Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)
Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр
(назва освітнього ступеня)

на тему: Розробка системи моніторингу трафіку в ієрархічних комп'ютерних Мережах з використанням технологій MySQL, PHP, Bash

Виконав(ла): студент(ка) VI курсу, групи СПм-61
спеціальності 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

(підпис) Музиченко Є.Д (прізвище та ініціали)

Керівник (підпис) Стояном Ю.М (прізвище та ініціали)

Нормоконтроль (підпис) Стоянов Ю.М (прізвище та ініціали)

Завідувач кафедри (підпис) Петрик М.Р (прізвище та ініціали)

Рецензент (підпис) Стадник Н.Б (прізвище та ініціали)

Тернопіль
2022

Реферат

Кваліфікаційна робота магістра містить: 130 стр., 28 рис., 3 таблиць, 50 джерел.

ТРАФІК, СИСТЕМА МОНІТОРИНГУ, ІЄРАРХІЧНА КОМП'ЮТЕРНА МЕРЕЖА, NETFLOW, СЕНСОР, КОЛЕКТОР, ПРОГРАМА ВІЗУАЛІЗАЦІЇ.

У даній роботі була розглянута система моніторингу трафіку в ієрархічних комп'ютерних мережах; досліджено теоретичні основи, вивчено основні методи, моделі та алгоритми моніторингу трафіку, проаналізовано протоколи моніторингу для ієрархічних комп'ютерних мереж, зроблено аналіз ефективності різного програмного забезпечення, висунуто пропозиції щодо реалізації системи моніторингу трафіку в ієрархічних комп'ютерних мережах.

Різні системи були створені для моніторингу трафіку. Всі вони мають свої переваги і недоліки. Основним недоліком таких систем є те, що вони орієнтовані для загальних цілей, а вузькоспеціалізовані функції страждають. Оскільки ми маємо специфічне завдання, було розроблено систему моніторингу трафіку в ієрархічній комп'ютерній мережі за допомогою технології NetFlow. Ця система дозволяє здійснювати моніторинг, аналіз і збір трафіку в ієрархічній комп'ютерній мережі. Ми можемо будувати різні графіки, діаграми і таблиці статистики трафіку. Це може бути реалізовано в усіх галузях, де є ієрархічні комп'ютерні мережі, і де необхідно моніторити трафік. Створена система є дешевою, побудована на програмному забезпеченні з відкритим вихідним кодом. Отже, вона може бути використана для проектів з низьким бюджетом.

Результати дипломної роботи рекомендуються до використання в наукових дослідженнях та практичних роботах працівників комп'ютерних наук.

Abstract

TRAFFIC, MONITORING SYSTEM, HIERARCHICAL COMPUTER NETWORK, NETFLOW, SENSOR, COLLECTOR, VISUALIZING UNIT

In the work the traffic monitoring system for hierarchical computer networks was considered; investigated the study of the theoretical fundamentals, investigation of the known methods, models and algorithms of traffic monitoring, analysis of traffic monitoring protocols in hierarchical computer networks, consideration of traffic measurement effectiveness using different software; given the proposition for the realization of traffic monitoring system for hierarchical computer network.

Different systems were created for traffic monitoring. All of them have advantages and disadvantages. The main disadvantage of such systems is that they oriented on many purposes, so the narrowly specialized functions suffer. As we have very specific task, so the traffic monitoring system for the hierarchical computer network with the NetFlow technology was created. This system allows monitoring, analyzing and collecting traffic from the hierarchical computer network. We can build different graphs, diagrams and tables of traffic statistics. It can be implemented in all fields, where hierarchical computer network used and where traffic monitoring is needed. The created system is low cost, built on open source software. So it can be used for projects with low budget.

The results of the degree thesis are recommended to use in scientific researches and for practical works of the computer sciences workers.

ЗМІСТ

ГЛОСАРІЙ

ARP - протокол роздільної здатності адрес

ASIC - спеціальна інтегрована схема для застосунків

BOOTP - протокол Bootstrap

BPF - пакетний фільтр Berkeley

CLI - інтерфейс командного рядка

CPU – центральний процесор

DBI - інтерфейс бази даних

DHCP - протокол динамічної конфігурації хоста

DLL - динамічно завантажувані бібліотеки

DNS - служба доменних імен

DoS - відмова в обслуговуванні

EIGRP - розширений внутрішній протокол маршрутизації шлюзу

FTP - протокол передачі файлів

HTTP - протокол передачі гіпертексту

HTTPS - безпечний протокол передачі гіпертексту

ICMP - протокол керуючих повідомлень Інтернету

IDS - система виявлення вторгнень

IETF - інженерна робоча група Інтернету

IGMP - протокол керування групами Інтернету

IGRP - протокол маршрутизації внутрішнього шлюзу

IOS - мережева операційна система

IP - Інтернет-протокол

IPX - мережевий обмін пакетами

ISDN - цифрова мережа з інтегрованими послугами

JDBC - підключення до бази даних Java

LAN – Локальна мережа

MIB - інформаційна база управління

MAC - контроль доступу до медіа

NEPED - Мережевий детектор Ethernet Promiscuous

NIC - контролер мережевого інтерфейсу

NMS - станція керування мережею

NFR - мережевий бортовий самописець

OS – операційна система

PC – персональний комп’ютер

QoS - якість обслуговування

RAM – оперативна пам’ять

RFC - запит на коментарі

RMON - віддалений моніторинг мережі

SCTP - протокол передачі керування потоком

SNMP - простий протокол керування мережею

TCP - протокол керування передачею

UDP - протокол дейтаграм користувача

WAN – глобальна мережа

VPN - віртуальна приватна мережа

ВСТУП

Інтернет стає основною мережею зв'язку. Він дозволяє виконувати все більшу кількість видів діяльності, починаючи від перегляду веб-сторінок та обміну файлами до онлайн-ігор або IP-телефонії. Через зростаючу популярність Інтернету, системи моніторингу трафіку необхідно вбудовувати в мережу для контролю комунікації, щоб забезпечити QoS (якість обслуговування) або навіть уникнути порушень безпеки.

Система моніторингу трафіку – це мережевий аналітичний інструмент, який перевіряє використання локальної мережі та забезпечує відображення статистики вивантаження та завантаження. Основною метою системи є моніторинг (і підрахунок) IP-трафіку між локальною мережею (LAN) та Інтернетом.

Система моніторингу трафіку забезпечує облік і моніторинг трафіку в реальному часі. Він дуже динамічний, кожне нове підключення реєструється та відстежується, ви можете використовувати його для підрахунку корисного трафіку завантаження та вивантаження комп’ютера або розширити його для побудови системи обліку трафіку для всіх комп’ютерів у локальній мережі вашої компанії.

Актуальність**.**

Системи моніторингу дозволяють контролювати сотні і навіть тисячі параметрів, що стосуються роботи різних апаратних і прикладних підсистем. Крім того, вони забезпечують не тільки збір цих параметрів, але й виконують попередню статистичну обробку, полегшуючи наступний аналіз. На основі зібраних даних виявлені проблеми, що знижують загальну продуктивність системи, перспективний і сценарний аналіз. Зокрема, ми можемо оцінити завантаженість сервера за місяць, квартал чи рік, розрахувати параметри його роботи за рахунок збільшення кількості користувачів (запити, обсяг трафіку, тощо) або визначити завантаженість сервера підсистеми після оновлення. Аналіз може визначити, яка підсистема сервера найбільш чутлива до перезавантажень; модернізація яких підсистем дасть найбільший приріст продуктивності, яка динаміка і структура прикладеного навантаження. Таким чином, система моніторингу запобігає проблемам,

викликаним недостатньою продуктивністю, а також допомагає ретельно планувати розвиток дата-центру відповідно до розвитку бізнесу.

Мета і цілі дослідження**.** Основною метою даної роботи є створення системи моніторингу трафіку в ієрархічних комп’ютерних мережах. Предметом дослідження є ієрархічна комп'ютерна мережа. Об’єктом дослідження є система моніторингу трафіку.

Методи дослідження**.** Вибрані методи: вивчення теоретичних основ, дослідження відомих методів, моделей та алгоритмів моніторингу трафіку, аналіз протоколів моніторингу трафіку в ієрархічних комп’ютерних мережах, розгляд ефективності вимірювання трафіку за допомогою різного програмного забезпечення, реалізація системи моніторингу трафіку та створення конфігурації.

Наукова новизна отриманих результатів та їх практичне значення**.** Надзвичайна еволюція комп'ютерної техніки залежить від швидкої передачі даних. Втрати при передачі даних в комп'ютерних мережах призводять до зниження ефективності мережі. Тож ми маємо відстежувати та аналізувати трафік. Існує кілька технологій моніторингу трафіку. Існує кілька підходів до моніторингу мережевого трафіку, кожен із яких має різні сильні та слабкі сторони. На даний момент існує три основні варіанти моніторингу трафіку – RMON, NetFlow, sFlow.

Створено різні системи для моніторингу трафіку. Всі вони мають переваги і недоліки. Основним недоліком таких систем є те, що вони орієнтовані на багато цілей, тому страждають вузькоспеціалізовані функції. Оскільки у нас дуже конкретне завдання, була створена система моніторингу трафіку для ієрархічної комп’ютерної мережі за технологією NetFlow. Ця система дозволяє контролювати, аналізувати та збирати трафік з ієрархічної комп’ютерної мережі. Ми можемо будувати різні графіки, діаграми та таблиці статистики трафіку. Її можна реалізувати у всіх сферах, де використовується ієрархічна комп'ютерна мережа, і де необхідний моніторинг трафіку. Створена система має низьку вартість, побудована на відкритому програмному забезпеченні. Тому її можна використовувати для проектів з невеликим бюджетом.

1. ОГЛЯД ПРЕДМЕТНОЇ ОБАСТІ

1.1 Постановка завдань

Комп’ютерна мережа — це з’єднання або набір з’єднань, створених між двома чи більше комп’ютерами з метою обміну даними. Мережі будуються з різноманітних будівельних блоків: комп’ютерів, комутаторів, кабелів, роутерів, тощо. Для того, щоб класифікувати мережі на різні типи, нам потрібно враховувати такі фактори, як кількість елементів, розподіл об’єктів і способи з’єднання.

Найменша мережа — це пряме з’єднання двох комп’ютерів за допомогою кабелю. Однорангові системи використовуються в комп’ютерних робочих групах, де є невелика кількість систем, які не потребують центральної служби. Деякі комп'ютерні шини можна конфігурувати, тому вони вважаються малими мережами. Вони називаються персональними LANs (локальними мережами), або pLANs, і Bluetooth є прикладом цього типу мережі. Мережа, яка охоплює офіс, поверх або будівлю, називається локальною мережею або LAN. Локальні мережі можуть підтримувати кілька протоколів і підключати різні типи клієнтів. Локальну мережу, відокремлену з'єднувальним елементом, можна вважати окремою локальною мережею. Коли міст розділяє кілька локальних мереж, які географічно розподілені, він вважається глобальною мережею або WAN. Ми можемо аналізувати та класифікувати мережеві топології з точки зору теорії графів. Мережі можна формувати різноманітними способами, які передбачають формування ліній або ланцюгів, зірок або вузлів, кілець або деревних топологій. Різні топології пропонують різні можливості та мають різні вимоги. Процеси відображення топології мережі можуть виконуватися для фізичних або логічних елементів мережі або на основі того, як сигнали поширюються мережею.

Метою цього розділу є огляд існуючих рішень системи моніторингу трафіку в ієрархічних комп’ютерних мережах. Однією з основних ідей є дослідження того, що таке ієрархічна комп’ютерна мережа. Ми повинні розглянути її структуру, топологію та базову модель. З іншого боку, ми повинні шукати основні методи, моделі та алгоритми моніторингу трафіку в ієрархічних комп'ютерних мережах.

Завдяки цій інформації ми можемо робити аналіз протоколів моніторингу трафіку в ієрархічних комп’ютерних мережах.

Основними завданнями цього розділу є:

- дослідження ієрархічної комп'ютерної мережі, її структури, топології та базової моделі;

- дослідження основних методів, моделей та алгоритмів системи моніторингу трафіку в ієрархічних комп’ютерних мережах;

- проведення аналізу протоколів моніторингу трафіку в ієрархічній комп'ютерній мережі.

1.2 Ієрархічна комп’ютерна мережа як об’єкт моніторингу

1.2.1 Огляд ієрархічних комп'ютерних мереж

Однією з класифікацій комп’ютерних мереж є топологія, яку вони використовують. Топологія — це розподіл або розташування елементів мережі, як правило, як пристроїв, так і з’єднань. Оскільки все, що може отримати адресу, вважається мережевим елементом, ми можемо визначити логічний або віртуальний мережевий елемент у програмному забезпеченні, і вони обидва мають бути включені в будь-який топологічний опис. Мережа може бути описана з точки зору фізичної топології, яка описує зв'язок між пристроями або елементами; логічної топології, яка описує зв'язок або ієрархію між об'єктами в мережі; або гібридної топології, яка є поєднанням двох в єдину топологічну структуру. У дуже рідкісних випадках мережа може бути описана з точки зору топології сигналу. Логічна топологія може бути відображена, щоб вказати, як влаштовані вузли мережі та як взаємодіють один з одним. Фізична топологія визначала б мережу з точки зору фізичних з’єднань і фізичної структури мережі. Топологію сигналу можна побудувати, щоб показати, як конкретні типи сигналів переміщуються по мережі. Фізична та логічна топології можуть бути ідентичними, але часто вони абсолютно різні. Математичне дослідження пов’язаних систем є частиною теорії графів, і ця дисципліна може робити прогнози щодо кількості вузлів, необхідних для різних топологій, кількості зв’язків або розгалужень, тощо. Конкретна топологія, яка використовується будь-якою мережею, може бути однаковою незалежно від

швидкості мережі, протоколів, що використовуються для зв’язку, мережевого вузла або типів з’єднання. Топологія стосується лише відносного розташування елементів [1].

1.2.2 Топології фізичної мережі

Фізична топологія описує розташування пристроїв, які використовуються для реалізації мережі. Топологічні пристрої можуть бути або вузлами, або кінцевими точками, або вони можуть бути з'єднаннями чи посиланнями. Фізична топологія може приймати різні форми:

- Шиноподібна**.** Де вузли приєднуються до лінійної магістральної лінії.
- Зіркоподібна**.** Де кілька вузлів з’єднуються один з одним через один вузол.
- Кільцеподібна**.** Де вузли підключені до циклічної магістральної лінії.
- Сіткоподібна**.** Де вузли з’єднані з іншими вузлами безпосередньо (павутина).
- Деревоподібна**.** Де вузли в мережі поширюються назовні, як гілки дерева.

Багато мереж є комбінаціями цих типів. Можна розрахувати необхідну кількість підключень, яку матиме теоретична сітчаста мережа, коли кожен вузол з’єднаний з кожним іншим вузлом. З одноланковим зв’язком постійна топологія сітки «точка-точка» між вузлами є одночасно найпростішою з існуючих і найбільш непрактичною. Для обслуговування n кінцевих точок знадобиться 2 (n + 1) з’єднань, що для будь-якої великої мережі вимагатиме непідтримуваної інфраструктури постійних з’єднань. Більшість мереж «точка-точка», як і телефонні мережі, є комутованими, що усуває необхідність мати з’єднання «точка-точка» між кожним вузлом. Перемикання може здійснюватися апаратно за допомогою комутації каналів або шляхом зміни адресації в потоці даних, які посилаються на комутацію пакетів. Роберт Меткалф, який був одним із головних розробників технології Ethernet, описав цінність комутованих мереж з точки зору кількості користувачів. Закон Меткалфа стверджує, що корисність телекомунікаційної мережі пропорційна квадрату числа підключених до неї користувачів. Див. формулу:

$$N = n(n-1)/2,$$

де N – кількість унікальних з’єднань;

n – кількість вузлів.

Зі збільшенням кількості вузлів вона стає асимптотично пропорційною кривій для n2. Асимптота — це рівняння, яке наближається до певної функції або значення, коли одна з його змінних стає більшою. У наведеному вище прикладі, коли n стає великим, рівняння (n2-n)/2 буде домінованe n2, і ця крива матиме 1/2 розміру n2 [2].

1.2.3 Деревоподібні або ієрархічні мережі

Щоб задовольнити ділові та технічні цілі замовника щодо дизайну корпоративної мережі, нам може знадобитися топологія мережі, що складається з багатьох взаємопов’язаних компонентів. Це завдання стає легшим, якщо ми можемо розділити роботу та розробити дизайн за рівнями.

Експерти з проектування мережі розробили ієрархічну модель проектування мережі, щоб допомогти нам розробити топологію в окремих рівнях. Кожен рівень можна зосередити на певних функціях, що дозволяє нам вибрати правильні системи та характеристики для рівня. Наприклад, на рис. 1.1 високошвидкісні роутери WAN можуть передавати трафік через корпоративну магістраль WAN, середньошвидкісні роутери можуть з’єднувати будівлі в кожному кампусі, а комутатори можуть з’єднувати пристрої користувачів і сервери в будівлях.

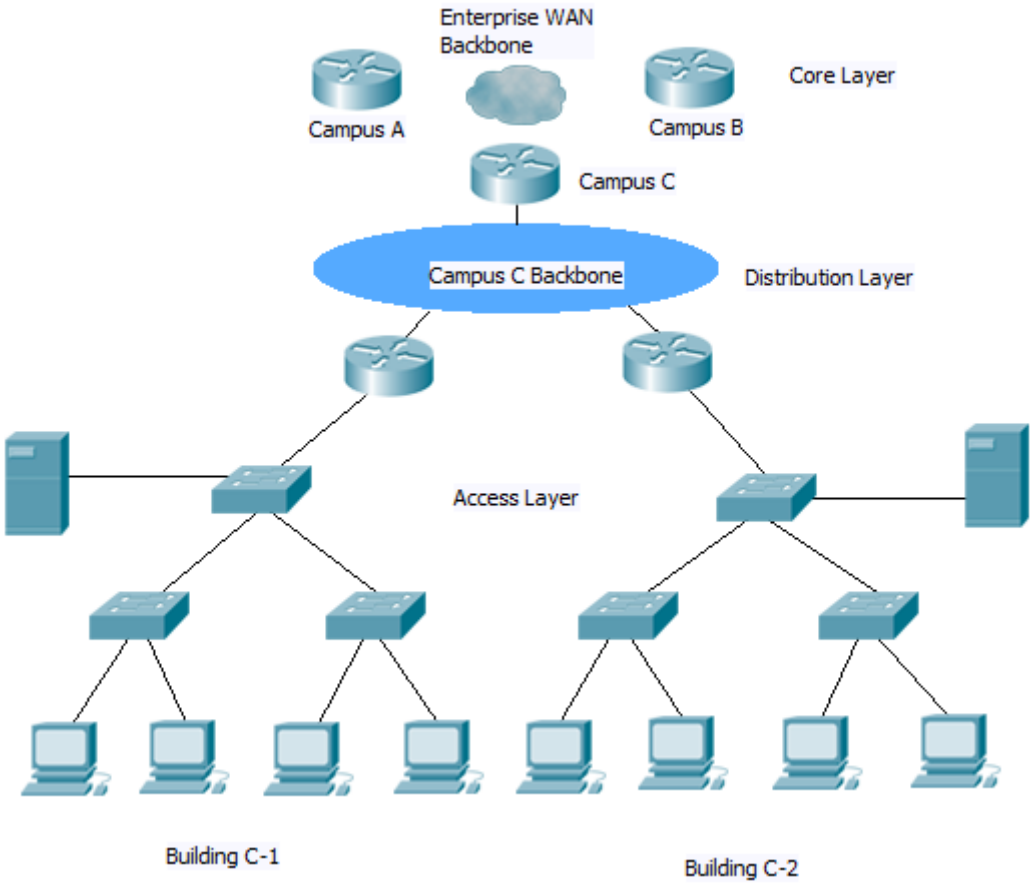

Рис. 1.1. Ієрархічна топологія

Деревоподібна мережа починається з найвищого рівня або кореневого рівня, де один вузол з'єднаний з вузлами другого рівня ієрархії. Кожен вузол другого рівня з'єднується з одним або кількома вузлами третього рівня, і кожен рівень розгортається далі. В ієрархії повинно бути принаймні три рівні, оскільки два рівні визначають зіркоподібну топологію. Кількість зв'язків у деревоподібній топології можна розрахувати за формулою:

$$L = n - 1,$$

де L — кількість зв'язків «точка-точка»;

n — кількість вузлів.

Кількість вузлів, приєднаних до батьківського вузла, розглядаються як віяло або коефіцієнт розгалуження. Деякі мережі застосовують симетричне розгалуження, і якщо це так, коефіцієнт розгалуження (f) має бути 2 або більше, оскільки коефіцієнт 1 визначає лише лінійну топологію. Хоча це називається деревоподібною мережею, її форму зазвичай малюють із коренем у верхній частині діаграми, що означає, що дерево перевернуте, як ми бачимо на рис. 1.2.

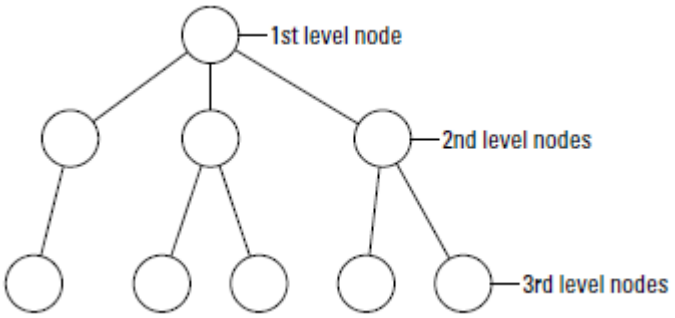

Рис. 1.2. Деревоподібна мережа

Більшість файлових систем, баз даних і систем каталогів використовують ієрархічну топологію. Це пояснюється тим, що алгоритми пошуку набагато ефективніші в ієрархії, ніж у лінійних або сітчастих топологіях. Це особливо стосується випадків, коли значення, що зберігаються на будь-якому вузлі, індексуються. Коли алгоритм пошуку спускається по дереву, перехід на наступний рівень виключає 1/f популяції дерева. Одним із недоліків ієрархічних топологій є те, що будь-який верхній елемент, пов'язаний з передачею даних між рівнями підсилюється, коли ми просуваємося вгору по ієрархії. Вузли на кожному рівні вище додають до верхнього елемента, що необхідний для обробки передачі даних [3].

Типова ієрархічна топологія:

- Серцевинний шар високоякісних роутерів і комутаторів, які оптимізовані для доступності та продуктивності.

- Рівень розподілу роутерів і комутаторів, які реалізують стратегії.

- Рівень доступу, який з'єднує користувачів через комутатори нижнього рівня та бездротові точки доступу.

1.2.4 Ефективність моделі проектування ієрархічної мережі

Мережі, які збільшуються не за планом, як правило, розвиваються в неструктурованому форматі. Доктор Пітер Велчер, автор статей про мережевий дизайн і технологію для Cisco World та інших видань, називає незаплановані мережі мережами fur-ball.

Велчер пояснює недоліки топології fur-ball, вказуючи на проблеми, які спричиняють занадто багато суміжностей ЦП (центрального процесора). Коли мережеві пристрої взаємодіють з багатьма іншими пристроями, робоче навантаження, необхідне для ЦП на пристроях, може бути обтяжливим. Наприклад, у великій плоскій (комутованій) мережі широкомовні пакети є обтяжливими. Широкомовний пакет перериває ЦП на кожному пристрої в широкомовному домені та вимагає часу на обробку на кожному пристрої, для якого встановлено розуміння протоколу для цієї трансляції. Це включає роутери, робочі станції та сервери.

Іншою потенційною проблемою з неієрархічними мережами, окрім широкомовних пакетів, є робоче навантаження ЦП, необхідне роутерам для зв'язку з багатьма іншими роутерами та обробки численних рекламних оголошень про маршрути. Методологія проектування ієрархічної мережі дозволяє розробити модульну топологію, яка обмежує кількість роутерів, що підключаються.

Використання ієрархічної моделі може допомогти нам мінімізувати витрати. Ми можемо придбати відповідні міжмережеві пристрої для кожного рівня ієрархії, таким чином уникаючи витрачання грошей на непотрібні функції для рівня. Крім того, модульний характер ієрархічної моделі проектування дає змогу точно планувати пропускну здатність на кожному рівні ієрархії, таким чином зменшуючи марну пропускну здатність. Відповідальність за управління мережею та системи керування мережею можуть бути розподілені між різними рівнями модульної мережевої архітектури для контролю витрат на управління.

Модульність дозволяє нам зберігати кожен елемент дизайну простим і зрозумілим. Простота зводить до мінімуму потребу в інтенсивному навчанні персоналу мережевих операцій і прискорює реалізацію проекту. Тестування дизайну мережі спрощується, оскільки на кожному рівні є чітка функціональність. Ізоляція несправностей покращена, оскільки мережеві технічні спеціалісти можуть легко розпізнавати точки переходу в мережі, щоб допомогти їм ізолювати можливі точки збою.

Ієрархічний дизайн полегшує зміни. Оскільки елементи мережі потребують змін, витрати на оновлення покладаються на невелику частину загальної мережі. У великих плоских або сітчастих мережевих архітектурах зміни, як правило,

впливають на велику кількість систем. Заміна одного пристрою може вплинути на численні мережі через складні взаємозв’язки.

1.2.5 Плоскі та ієрархічні топології

Плоска топологія мережі підходить для дуже малих мереж. З плоскою структурою мережі немає ієрархії. Кожен мережевий пристрій виконує, по суті, ту саму роботу, і мережа не поділена на рівні чи модулі. Плоску мережеву топологію легко спроектувати та реалізувати, її легко підтримувати, доки мережа залишається невеликою. Однак коли мережа розростається, плоска мережа небажана. Відсутність ієрархії ускладнює усунення несправностей. Замість того, щоб зосередити зусилля з усунення несправностей лише в одній зоні мережі, нам може знадобитися перевірити всю мережу.

1.2.6 Плоскі топології WAN

Глобальна мережа (WAN) для невеликої компанії може складатися з кількох сайтів, з’єднаних у кільце. Кожен сайт має роутер WAN, який з’єднується з двома іншими суміжними сайтами за допомогою з’єднань «точка-точка», як показано в частині А рис. 1.3. Поки глобальна мережа невелика (декілька сайтів), протоколи маршрутизації можуть швидко об’єднуватися, а зв’язок із будь-яким іншим сайтом може відновлюватися у разі збою зв’язку. Зв’язок відновлюється, якщо виходить з ладу лише одне з’єднання. Якщо більше ніж одне з’єднання не працює, деякі сайти ізольовані від інших.

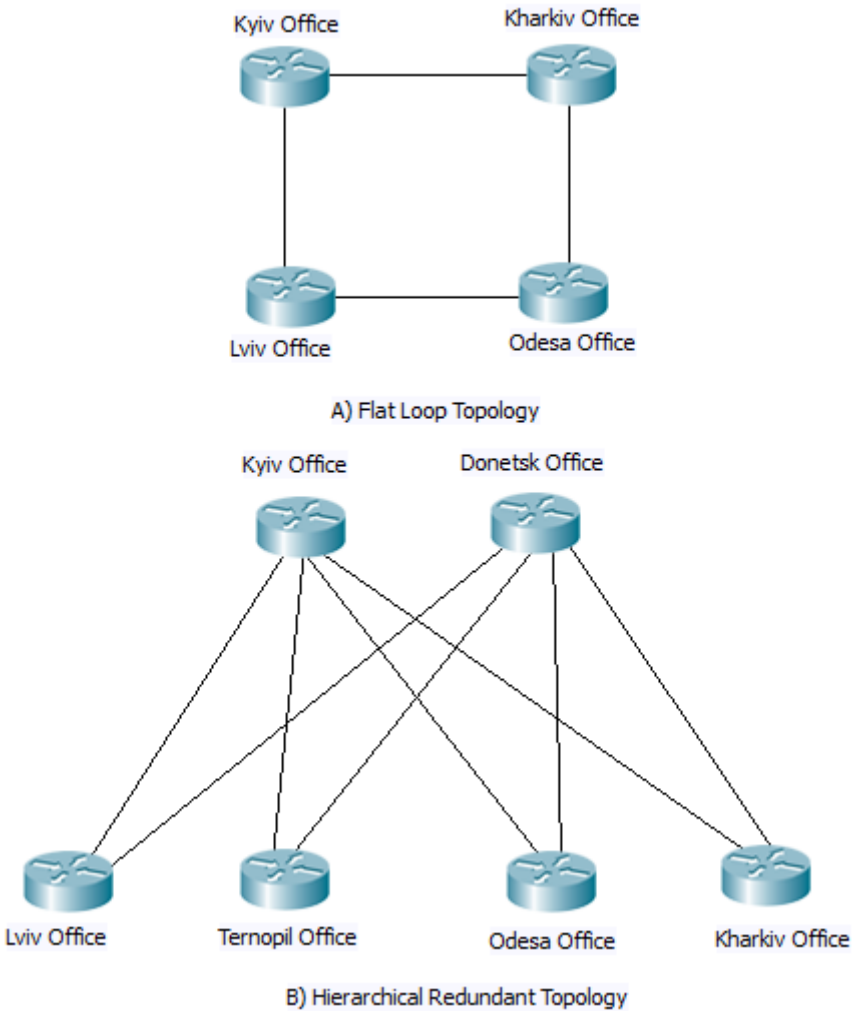

Рис. 1.3. А) Топологія плоского циклу. Б) Ієрархічна надлишкова топологія.

Однак, топологія плоского циклу зазвичай не рекомендується для мереж із багатьма сайтами. Топологія циклу може означати, що існує багато стрибків між маршрутизаторами на протилежних сторонах циклу, що призводить до значної затримки та більшої ймовірності збою. Якщо наш аналіз потоку трафіку показує, що маршрутизатори на протилежних сторонах топології циклу обмінюються великим трафіком, ми повинні рекомендувати ієрархічну топологію замість циклу. Щоб уникнути будь-якої єдиної точки збою, резервні маршрутизатори або комутатори можна розмістити на верхніх рівнях ієрархії, як показано в частині Б рис. 1.3.

Топологія плоского циклу, показана в частині А рис. 1.3, відповідає цілям щодо низької вартості та достатньо високої доступності. Ієрархічна надлишкова топологія, показана в частині Б рис. 1.3, відповідає цілям щодо масштабованості, високої доступності та низької затримки.

1.2.7 Плоскі топології LAN

На початку та в середині 1990-х типовою конструкцією локальної мережі були ПК (персональні комп'ютери) і сервери, приєднані до одного або кількох центрів у плоскій топології. Комп'ютери та сервери реалізували процес керування доступом до медіа файлів, наприклад, передачу маркерів або множинні доступи з визначенням несучої мережі з виявленням колізій (CSMA/CD), щоб контролювати доступ до спільної смуги пропускання. Усі пристрої були частиною однієї області пропускної здатності та мали здатність негативно впливати на затримку та пропускну здатність для інших пристроїв.

У наші дні розробники мереж зазвичай рекомендують підключати ПК і сервери до комутаторів канального рівня замість центрів. У цьому випадку мережа сегментується на домени з невеликою пропускною здатністю, щоб обмежена кількість пристроїв конкурувала за пропускну здатність у будь-який момент часу. Однак пристрої конкурують за обслуговування комутаційного обладнання та програмного забезпечення, тому важливо розуміти робочі характеристики потенційних комутаторів.

Пристрої, підключені до комутованої або мостової мережі, є частиною одного широкомовного домену. Перемикачі передають широкомовні кадри за межі усіх портів. З іншого боку, маршрутизатори сегментують мережі на окремі широкомовні домени. Один широкомовний домен має бути обмежений кількома сотнями пристроїв, щоб пристрої не були перевантажені завданням обробки широкомовного трафіку. Впроваджуючи ієрархію в проект мережі шляхом додавання маршрутизаторів, широкомовне випромінювання скорочується.

Завдяки ієрархічній структурі міжмережеві пристрої можуть бути розгорнуті для виконання роботи, з якою вони справляються найкраще. Маршрутизатори можуть додаватися до дизайну мережі кампусу, щоб ізолювати широкомовний трафік. Комутатори високого класу можна розгорнути, щоб максимізувати пропускну здатність для додатків із високим трафіком, а комутатори низького класу можна використовувати, коли потрібен простий і недорогий доступ. Максимізація загальної продуктивності шляхом модульного розподілу завдань, необхідних для міжмережевих пристроїв, є однією з багатьох переваг використання ієрархічної моделі проектування.

1.2.8 Сітчаста в порівнянні з ієрархічно-сітчастою топологією

Розробники мереж часто рекомендують сітчасту топологію для задоволення вимог доступності. У повносітчастій топології кожен маршрутизатор або комутатор підключений до іншого маршрутизатора або комутатора. Повносітчаста мережа забезпечує повне резервування та пропонує хорошу продуктивність, оскільки між будь-якими двома сайтами існує лише одноканальна затримка. Частковосітчаста мережа має менше з'єднань. Щоб отримати доступ до іншого маршрутизатора або комутатора в частковосітчастій мережі, може знадобитися обхід через проміжні ланки, як показано на рис. 1.4.

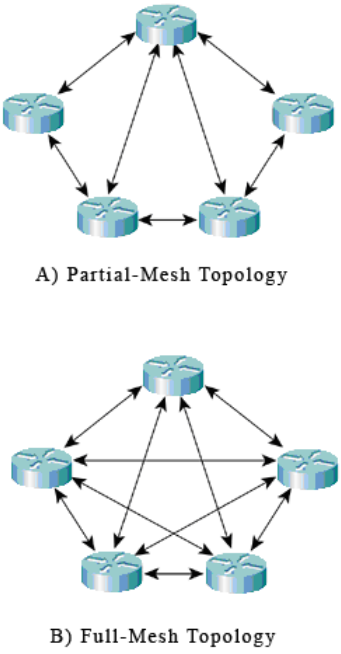

Рис. 1.4. А) Частковосітчаста топологія. Б) Повносітчаста топологія.

У повносітчастій топології кожен маршрутизатор або комутатор підключений до іншого маршрутизатора або комутатора. Кількість зв'язків у повносітчастій топології відображається за допомогою формули:

$$(N * (N - 1)) / 2,$$

де N - кількість маршрутизаторів або комутаторів.

Розділіть результат на два, щоб уникнути врахування маршрутизатора X до маршрутизатора Y і маршрутизатора Y до маршрутизатора X як два різні зв'язки.

Хоча сітчасті мережі характеризуються високою надійністю, у них є багато недоліків, якщо вони неретельно розроблені. Розгортання і підтримка сітчастих мереж може бути дороговартісною (Повносітчаста мережа є особливо дорогою). Сітчасті мережі також може бути важко оптимізувати, усунути неполадки та оновити, якщо вони не розроблені з використанням простої ієрархічної моделі. У неієрархічній сітчастій топології міжмережні пристрої не оптимізовані для певних функцій. Вирішувати проблеми мережі важко через відсутність модульності. Оновлення мережі є проблематичним, оскільки важко оновити лише одну частину мережі.

Сітчасті мережі мають обмеження масштабованості для груп маршрутизаторів, які транслюють оновлення маршрутизації або рекламу послуг. Зі збільшенням кількості суміжних центральних процесорів маршрутизатора збільшується обсяг пропускної здатності та ресурсів ЦП, призначених для обробки оновлень.

Правило полягає в тому, щоб трансляційний трафік становив менше 20 відсотків трафіку по кожному зв'язку. Це правило обмежує кількість суміжних маршрутизаторів, які можуть обмінюватися таблицями маршрутизації та рекламою послуг. Однак це обмеження не є проблемою, якщо дотримуватися вказівок щодо простого ієрархічного дизайну. Ієрархічний дизайн за своєю природою обмежує кількість суміжних маршрутизаторів.

З протоколами маршрутизації, такими як Знаходження найкоротшого часу (OSPF) і Розширений внутрішній протокол маршрутизації шлюзу (EIGRP), проблема полягає не в широкомовному/багатоадресному трафіку та ресурсах ЦП, які використовуються для повсякденної маршрутизації. Проблема полягає в обсязі роботи та пропускній здатності, що необхідні для відновлення маршрутизації після збою. Ми повинні бути обережними, щоб наша мережа не зросла в складну сітку лише тому, що вона все ще працює. Ймовірно, колись станеться збій, і тоді ми на практиці дізнаємось про недоліки, пов’язані зі складною сіткою маршрутизаторів.

Класична ієрархічна та резервна модель підприємства показана на рис. 1.5. Модель використовує частково сітчасту ієрархію, а не повносітчасту. На малюнку показано маршрутизовану корпоративну мережу, але цю топологію також можна використовувати для комутованої кампусної мережі.

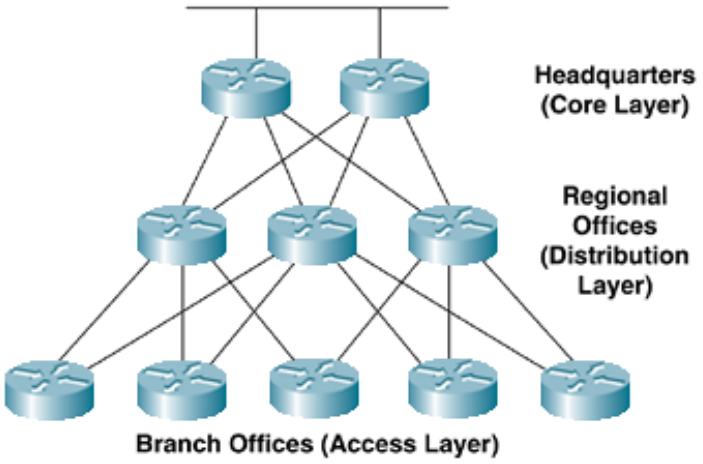

Рис. 1.5. Частковосітчаста ієрархічна модель

Для малих і середніх компаній ієрархічна модель часто реалізується як віялова топологія з невеликою сіткою або без неї. Корпоративна штаб-квартира або центр обробки даних утворюють хаб. Зв'язок на віддалені офіси та будинки дистанційної роботи утворюють спиці, як показано на рис. 1.6.

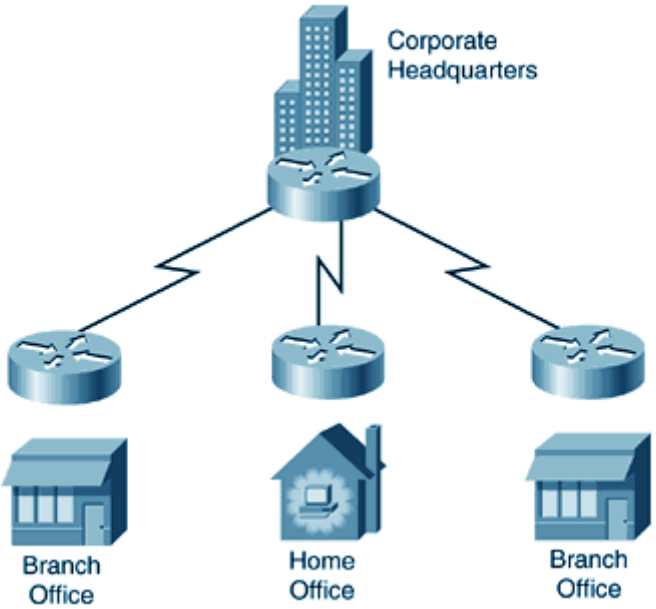

Рис. 1.6. Ієрархічна віялова топологія для середнього бізнесу

1.2.9 Класична трирівнева ієрархічна модель

Література, опублікована Cisco Systems, Inc. та іншими мережевими торговцями, розповідає про класичну трирівневу ієрархічну модель для топології проектування мережі. Трирівнева модель дозволяє агрегувати трафік і фільтрувати його на трьох послідовних шарах маршрутизації або комутації. Це робить трирівневу ієрархічну модель масштабованою до великих розмірів міжнародних мереж.

Хоча модель була розроблена в той час, коли маршрутизатори розмежовували рівні, модель може використовуватися як для комутованих мереж, так і для мереж з маршрутизацією. Трирівневі ієрархічні топології показані на рис. 1.2 і рис. 1.5.

Кожен рівень ієрархічної моделі виконує певну роль. Основний рівень забезпечує оптимальне транспортування між сайтами. Рівень розподілу з’єднує мережеві служби з рівнем доступу та реалізує політики щодо безпеки, завантаження трафіку та маршрутизації. У дизайні глобальної мережі рівень доступу складається з маршрутизаторів на межі мереж кампуса. У мережі кампусу рівень доступу забезпечує комутатори або центри для доступу кінцевих користувачів.

Основний рівень

Основний рівень трирівневої ієрархічної топології є високошвидкісною основою міжмережі. Оскільки основний рівень має вирішальне значення для взаємозв’язку, ми повинні створити основний рівень із надлишковими компонентами. Основний рівень має бути високонадійним і швидко адаптуватися до змін.

Налаштовуючи маршрутизатори на основному рівні, ми повинні використовувати функції маршрутизації, які оптимізують пропускну здатність пакетів. Нам слід уникати використання фільтрів пакетів або інших функцій, які сповільнюють маніпуляції з пакетами. Ми повинні оптимізувати ядро для низької затримки і хорошої керованості.

Ядро повинно мати обмежений і постійний діаметр. Маршрутизатори (або комутатори) рівня розподілу і клієнтські локальні мережі можуть бути додані до моделі без збільшення діаметра ядра. Обмеження діаметра ядра забезпечує передбачувану продуктивність і легкість усунення несправностей.

Для клієнтів, яким потрібно підключитися до інших підприємств через екстранет або Інтернет, топологія ядра повинна містити одне або кілька з'єднань до

зовнішніх мереж. Адміністратори корпоративних мереж повинні перешкоджати регіональним адміністраторам і адміністраторам філій планувати власні екстранети або підключення до Інтернету. Централізація цих функцій на основному рівні зменшує складність і потенційні проблеми з маршрутизацією, а також є важливою для мінімізації проблем безпеки.

Запровадження зв'язків між бізнес-партнерами у філії, де відбувається співпраця, може здатися логічним, але це означає, що ми маємо дозволити трафік партнера в філію, а не за її межі. З часом ми закінчимо мішанину розподілених списків контролю доступу та брандмауерів, що ускладнює здійснення політики. Це також значно підвищує витрати, якщо ми хочемо використовувати системи виявлення вторгнень (IDS) та інші технології безпеки.

Подібним чином деякі віддалені офіси з підключенням IPSec VPN відходять від розділеного доступу на віддалених сайтах, де користувачі мають локальний доступ до Інтернету на додачу до віддаленого доступу IPSec до головного офісу компанії. Незважаючи на витрати на пропускну здатність, змусити весь зовнішній доступ проходити через ядро мережі означає мати лише одну структуру безпеки для адміністрування, що є хорошим способом уникнути проблем безпеки.

Рівень розподілу

Рівень розподілу мережі є точкою розмежування між рівнями доступу та основними рівнями мережі. Рівень розподілу виконує багато ролей, включаючи контроль доступу до ресурсів з міркувань безпеки та контроль мережевого трафіку, який проходить через ядро з міркувань продуктивності. Рівень розподілу часто є рівнем, який розмежовує широкомовні домени (хоча це також можна зробити на рівні доступу). У мережевих конструкціях, які включають віртуальні локальні мережі (VLANs), рівень розподілу можна налаштувати для маршрутизації між VLANs.

Рівень розподілу дозволяє основному рівню підключати сайти, які працюють за різними протоколами, зберігаючи високу продуктивність. Щоб підтримувати високу продуктивність у ядрі, рівень розподілу може перерозподіляти між протоколами маршрутизації рівня доступу, що потребують інтенсивної пропускної здатності, і оптимізованими протоколами маршрутизації ядра. Наприклад, можливо, один сайт на рівні доступу все ще використовує старий протокол, наприклад IGRP.

Рівень розподілу може перерозподіляти між IGRP на рівні доступу та Enhanced IGRP на основному рівні.

Щоб покращити продуктивність протоколу маршрутизації, рівень розподілу може підсумовувати маршрути з рівня доступу. Для деяких мереж рівень розподілу пропонує маршрут за замовчуванням для доступу до маршрутизаторів рівня та запускає протоколи динамічної маршрутизації лише під час зв’язку з основними маршрутизаторами.

Щоб максимізувати ієрархію, модульність і продуктивність, рівень розподілу повинен приховувати детальну інформацію про топологію рівня доступу від основних маршрутизаторів. Рівень розподілу має підсумовувати численні призначення рівня доступу в кілька рекламних оголошень у ядро. Подібним чином рівень розподілу повинен приховувати детальну інформацію про топологію основного рівня від рівня доступу, підсумовуючи невеликий набір рекламних оголошень або лише один маршрут за замовчуванням, якщо це можливо. Рівень розподілу може забезпечити рівень доступу маршрутом до найближчого маршрутизатора рівня розподілу, який має доступ до ядра.

Рівень доступу

Рівень доступу надає користувачам локальних сегментів доступ до об'єднаної мережі. Рівень доступу може включати маршрутизатори, комутатори, мости, центри спільного доступу та бездротові точки доступу. Як згадувалося, комутатори часто реалізуються на рівні доступу в мережах кампусів, щоб розділити домени пропускної здатності, щоб задовольнити вимоги додатків, які потребують великої пропускної здатності або не можуть витримати змінну затримку, що характеризується спільною пропускною здатністю.

Для об'єднаних мереж, які включають невеликі філії та домашні офіси дистанційної роботи, рівень доступу може забезпечити доступ до корпоративної об'єднаної мережі за допомогою глобальних технологій, таких як ISDN, Frame Relay, орендовані цифрові лінії та аналогові модемні лінії. Ми можемо реалізувати функції маршрутизації, такі як маршрутизація набору номера за запитом (DDR) і статична маршрутизація, щоб контролювати використання пропускної здатності та мінімізувати витрати на віддалені канали рівня доступу. DDR зберігає зв'язок неактивним, за винятком випадків, коли потрібно надіслати певний трафік.

1.3 Методи, моделі та алгоритми моніторингу трафіку в ієрархічних комп’ютерних мережах

1.3.1. Системи моніторингу трафіку

Моніторинг трафіку є життєвоважливим елементом керування мережею та системою. Дуже мало відбувається на підприємстві без створення певного мережевого трафіку. Відстеження цього трафіку дає важливу інформацію про роботу корпоративних програм. Ця інформація має важливе значення для таких дій, як розподіл витрат, планування потужностей, аналіз якості обслуговування, виявлення несправностей, ізоляція та управління безпекою [4].

Колись моніторинг трафіку був простим завданням. У минулому велика кількість машин була підключена до спільної мережі. Спільна мережа дозволяє одному інструменту, підключеному до мережі, контролювати весь трафік, оскільки пакети, надіслані в одній частині мережі, приймаються в усіх інших частинах мережі.

Вимоги щодо збільшення пропускної здатності, зміни в моделях трафіку та швидке падіння цін на пристрої комутації пакетів і маршрутизації призвели до швидкого переходу від спільних мереж до мереж із високим ступенем сегментації.

Трафік більше не видно з однієї точки. Комутатор спрямовує пакети до певних портів на основі призначення пакетів. Щоб отримати повну картину мережевого трафіку, потрібно контролювати кожен порт комутатора. Використання зв’язків «точка-точка» ускладнює підключення приладів, а велика кількість інструментів, які знадобляться для моніторингу всіх портів комутатора, гарантує, що такий підхід не буде економічно ефективним. Крім того, самі комутатори та маршрутизатори мають складну внутрішню архітектуру, а потік пакетів усередині та через них стає важливим фактором продуктивності мережі.

Єдиний реалістичний спосіб моніторингу трафіку в комутованих мережах — це моніторинг трафіку всередині самих комутаторів. Окрім технічних складнощів завдання, існують також серйозні обмеження щодо ціни. Ринок комутаторів розвивається, і є дуже мало можливостей для збільшення вартості або впливу на

продуктивність цих пристроїв, особливо тому, що моніторинг є другорядним щодо основної функції комутації пристрою.

Існує кілька підходів до моніторингу мережевого трафіку, кожен із яких має різні сильні та слабкі сторони. На даний момент існує три основні варіанти моніторингу трафіку:

- RMON (Remote MONitor) — це стандарт IETF (Робочої групи Інтернет-інженерії), що визначає віддалений пристрій для моніторингу трафіку. Пристрій RMON відстежує та декодує кожен пакет у мережі, до якої він підключений, створює таблиці вимірювань, які пізніше можна завантажити програмою керування мережею [5].

- NetFlow. Маршрутизатори та комутатори Cisco, як частина їхньої системи моніторингу NetFlow, надсилають інформацію про завершені потоки трафіку до центрального колектора. Пристрій декодує кожен IP-пакет, веде таблиці активних потоків та пересилає записи потоків періодично або після їх завершення до програми керування мережею [6].

- sFlow. sFlow поєднує точні лічильники пакетів із статистичною вибіркою стану таблиць маршрутизації та мостів, які використовуються комутатором для пересилання випадково вибраних пакетів. Відібрана інформація негайно надсилається до центрального колектора для аналізу [7].

1.3.2 RMON як система моніторингу

Простий протокол керування мережею (SNMP) визначає як структуру, так і спеціальний протокол для обміну мережевою інформацією в обє'днаній мережі TCP/IP. Загальна модель, яку використовує SNMP, — це станція керування мережею (NMS), яка надсилає запити агентам SNMP, що працюють на керованих пристроях. Агенти SNMP також можуть ініціювати певні типи зв’язку, надсилаючи повідомлення-пастки, щоб повідомляти NMS про певні події.

Ця модель працює добре, тому SNMP став таким популярним. Однак одне фундаментальне обмеження протоколу та моделі, яку він використовує, полягає в тому, що він орієнтований на передачу мережевої інформації від агентів SNMP, які зазвичай є частиною постійних пристроїв TCP/IP, таких як хости та

маршрутизатори. Обсяг інформації, зібраної цими пристроями, зазвичай дещо обмежений, оскільки, очевидно, хости та маршрутизатори мають «справжню роботу», тобто виконувати роботу хостів і маршрутизаторів. Вони не можуть присвятити себе завданням управління мережею.

Таким чином, у ситуаціях, коли про мережу потрібно більше інформації, ніж збирають традиційні пристрої, адміністратори часто використовують спеціальні апаратні блоки, які називаються аналізаторами мережі, моніторами або зондами. Це спеціальні частинки обладнання, які підключені до мережі та використовуються суто для збору статистики та спостереження за подіями, які цікавлять або турбують адміністратора. Очевидно, було б дуже корисно, якби ці пристрої могли використовувати SNMP для отримання інформації, яку вони збирають, і могли створювати пастки, коли вони помічають щось важливе.

Для цього було створено специфікацію Віддаленого моніторингу мережі (RMON). RMON часто називають протоколом, а SNMP і RMON іноді називають «протоколами керування мережею TCP/IP». Однак RMON насправді взагалі не є окремим протоколом, він не визначає жодних операцій протоколу. RMON фактично є частиною SNMP і специфікації RMON [8].

Віддалений моніторинг (RMON) — це стандартна специфікація моніторингу, яка дозволяє різним мережевим моніторам і консольним системам обмінюватися даними моніторингу мережі. RMON надає мережевим адміністраторам більше свободи у виборі зондів і консолей для моніторингу мережі з функціями, які відповідають їхнім конкретним мережевим потребам.

Специфікація RMON визначає набір статистичних даних і функцій, якими можна обмінюватися між RMON-сумісними менеджерами консолі та мережевими зондами. Таким чином, RMON надає мережевим адміністраторам повну інформацію про діагностику несправностей мережі, планування та налаштування продуктивності.

RMON було визначено спільнотою користувачів за допомогою Робочої групи Інтернет-інженерії (IETF). Він став запропонованим стандартом у 1992 році як RFC 1271 (для Ethernet). Потім RMON став проектом стандарту в 1995 році як RFC 1757, ефективно замінивши RFC 1271. На рис. 1.7 показано зонд RMON, здатний

відстежувати сегмент Ethernet і передавати статистичну інформацію назад на RMON-сумісну консоль.

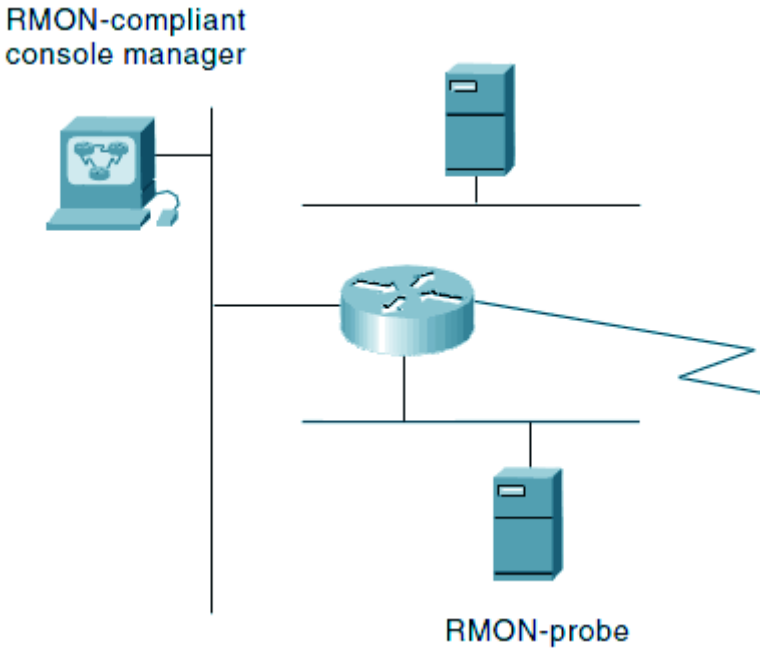

Рис. 1.7. Зонд RMON може надсилати статистичну інформацію на консоль RMON

RMON надає інформацію в дев’яти групах елементів моніторингу RMON, кожна з яких надає певні набори даних для задоволення загальних вимог до моніторингу мережі. Кожна група є необов’язковою, тому постачальникам не потрібно підтримувати всі групи в інформаційній базі керування (MIB). Для належної роботи деяких груп RMON потрібна підтримка інших груп RMON. У таблиці 1 узагальнено дев’ять груп моніторингу, визначених у RFC 1757 Ethernet RMON MIB [9].

Таблиця 1.1. Групи моніторингу RMON

Група RMON	Функція	Елементи
Статистика	Містить статистику, виміряну зондом для кожного контрольованого інтерфейсу на цьому пристрої.	Відкинуті пакети, надіслані пакети, надіслані байти (октети), широкомовні пакети, багатоадресні пакети, помилки

		CRC, ранти, гіганти, фрагменти, джеббери, зіткнення та лічильники для пакетів у діапазоні від 64 до 128, від 128 до 256, від 256 до 512, 512 до 1024 і від 1024 до 1518 байт.
Історія	Записує періодичні статистичні зразки з мережі та зберігає їх для подальшого пошуку.	Період відбору зразків, кількість зразків, відібрані зразки.
Сигналізація	Періодично бере статистичні зразки зі змінних у зонді та порівнює їх із попередньо налаштованими пороговими значеннями. Якщо змінна, яка відстежується, перетинає порогове значення, генерується подія.	Включає таблицю сигналізації і вимагає впровадження групи подій. Тип сигналізації, інтервал, початковий поріг, кінцевий поріг.
Хост	Містить статистику, пов’язану з кожним хостом, виявленим у мережі.	Адреса хоста, отримані та передані пакети і байти, а також широкомовні, багатоадресні та пакети помилок.
HostTopN	Готує таблиці, які описують хости, що очолюють список, упорядкований за однією з їхніх базових статистичних даних протягом інтервалу, визначеного станцією керування. Таким чином, ця статистика базується на швидкості.	Статистика, хост(и), періоди початку та зупинки вибірки, базова швидкість та тривалість.

Матриця	Зберігає статистику розмов між групами з двох адрес. Коли пристрій виявляє нову розмову, він створює новий запис у своїй таблиці.	Пари адрес джерела та призначення і пакети, байти та помилки для кожної пари.
Фільтри	Дозволяє порівнювати пакети за рівнянням фільтра. Ці відповідні пакети утворюють потік даних, який може бути захоплений або який може генерувати події.	Тип Bit-фільтра (маска або немаска), вираз фільтра (рівень бітів), умовний вираз (і чи ні) для інших фільтрів.
Перехоплення пакетів	Дозволяє перехоплювати пакети після того, як вони проходять через канал.	Розмір буфера для захоплених пакетів, повний статус (сигналізація) і кількість захоплених пакетів.
Події	Керує створенням і сповіщенням про події з цього пристрою.	Тип події, опис, час останнього надсилання події.

1.3.3 NetFlow як система моніторингу

NetFlow — це мережевий протокол, розроблений компанією Cisco Systems для роботи на обладнанні з підтримкою Cisco IOS для збору інформації про IP-трафік. Він є власністю, але підтримується платформами, відмінними від IOS, такими як маршрутизатори Juniper, Linux або FreeBSD і OpenBSD.

Маршрутизатори Cisco з увімкненою функцією Netflow генерують записи NetFlow; вони експортуються з маршрутизатора в протокол дейтаграм користувача (UDP) або протоколу передачі керування потоком (SCTP) і збираються за допомогою збирача netflow. Інші постачальники надають подібні функції для своїх маршрутизаторів, але з іншими назвами:

- Jflow або cflowd для Juniper Networks;

- NetStream для 3Com/H3C;

- NetStream для Huawei Technology;

- Cflowd для Alcatel-Lucent.

NetFlow — це інструмент, вбудований у програмне забезпечення Cisco IOS для характеристики роботи мережі. Візуалізація в мережі є незамінним інструментом для IT-фахівців. У відповідь на нові вимоги та тиск оператори мереж вважають критично важливим зрозуміти, як поводиться мережа, зокрема:

- Використання програми та мережі;

- Продуктивність мережі та використання мережевих ресурсів;

- Вплив змін на мережу;

- Аномалії мережі та вразливості безпеки;

- Питання довгострокової відповідності.

Cisco IOS NetFlow задовольняє ці потреби, створюючи середовище, де адміністратори мають інструменти для розуміння того, хто, що, коли, де та як рухає мережевий трафік. Коли поведінка мережі буде зрозуміла, бізнес-процеси покращаться, і буде доступний журнал аудиту того, як використовується мережа. Така підвищена обізнаність зменшує вразливість мережі, пов’язану з відключенням, і забезпечує ефективну роботу мережі. Удосконалення роботи мережі знижує витрати та призводить до підвищення доходів від бізнесу за рахунок кращого використання мережевої інфраструктури.

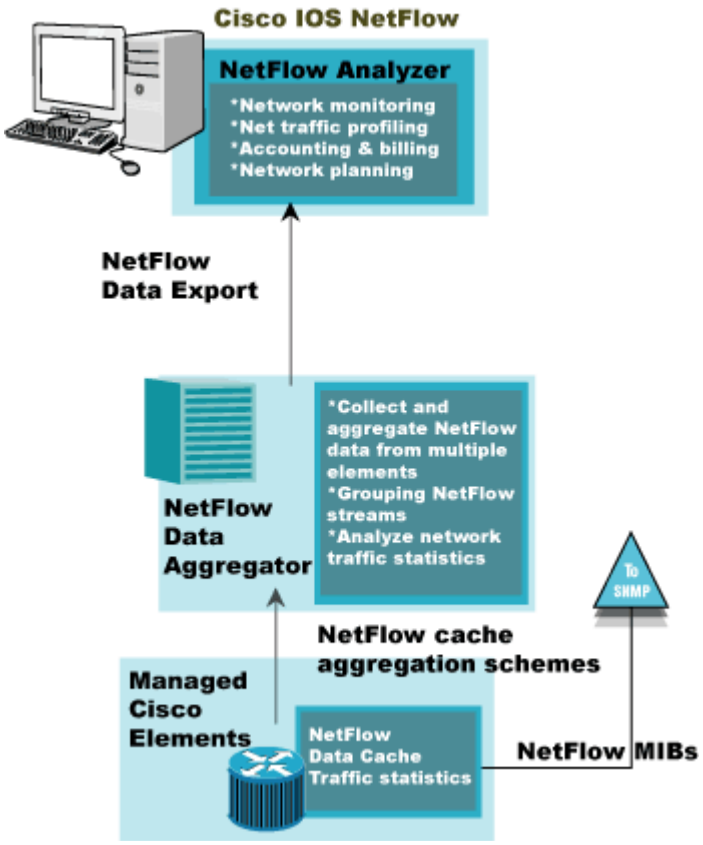

Рис. 1.8. NetFlow: протокол моніторингу та керування мережевим трафіком Cisco

Традиційно клієнти покладалися майже виключно на простий протокол керування мережею (SNMP) для моніторингу пропускної здатності. Незважаючи на те, що SNMP полегшує планування пропускної здатності, він дає недостатню характеристику додатків і шаблонів трафіку, що важливо, щоб зрозуміти, наскільки добре мережа підтримує бізнес. Більш детальне розуміння того, як використовується пропускна здатність, є надзвичайно важливим у сучасних IP-мережах. Лічильники інтерфейсу пакетів і байтів корисні, але розуміння того, які IP-адреси є джерелом і одержувачем трафіку та які програми генерують трафік, є неоціненним [11].

Здатність характеризувати IP-трафік і розуміти, як і куди він проходить, має вирішальне значення для доступності мережі, продуктивності та усунення несправностей. Моніторинг потоків IP-трафіку сприяє більш точному плануванню пропускної здатності та забезпечує належне використання ресурсів для підтримки цілей організації. Це допомагає IT-спеціалістам визначати, де застосувати якість обслуговування (QoS), оптимізувати використання ресурсів і відіграє важливу роль

у безпеці мережі для виявлення атак типу «Відмова в обслуговуванні» (DoS), мережевих черв'яків та інших небажаних мережевих подій.

NetFlow полегшує вирішення багатьох типових проблем, з якими стикаються IT-фахівці:

- Аналіз нових програми та їх вплив на мережу. Визначення нових мережевих навантажень програми, наприклад VoIP або додавання віддалених сайтів.

- Зменшення пікового трафіку глобальної мережі. Використання статистики NetFlow для вимірювання покращення трафіку глобальної мережі завдяки змінам політики додатків; розуміння, хто користується мережею та найбільших користувачів мережі.

- Усунення несправностей і розуміння проблемних точок мережі. Швидка діагностика низької продуктивності мережі, пропускної здатності і використання пропускної здатності за допомогою інтерфейсу командного рядка або інструментів звітування.

- Виявлення несанкціонованого трафіку глобальної мережі. Уникнення дорогих оновлень, визначаючи програми, що викликають перевантаження.

- Безпека та виявлення аномалій. NetFlow можна використовувати для виявлення аномалій і діагностики черв'яків разом із такими програмами, як Cisco CS-Mars.

- Перевірка параметрів QoS. Переконайтеся, що для кожного класу обслуговування (CoS) виділено відповідну пропускну здатність і що жоден CoS не має перевищення або недостатньої підписки.

Кожен пакет, який пересилається в межах маршрутизатора або комутатора, перевіряється на наявність набору атрибутів IP-пакету. Ці атрибути є ідентифікатором IP-пакета або відбитком пальця пакета та визначають, чи є пакет унікальним чи схожим на інші пакети. Традиційно IP-потік базується на наборі з 5 і до 7 атрибутів IP-пакетів.

Атрибути IP-пакетів, які використовує NetFlow:

- IP адреса джерела;

- IP адреса призначення;

- вихідний порт;

- порт призначення;

- тип протоколу рівня 3;
- клас обслуговування;
- інтерфейс маршрутизатора або комутатора.

Таблиця 1.2. Можливості постобробки системи NetFlow.

Функції пост обробки	Короткий опис
Схеми агрегації	Встановлює додаткові кеші агрегації з різними комбінаціями полів, що визначають, які традиційні потоки групуються разом і збираються, коли закінчується потік з основного кешу
Експорт в кілька місць призначень	Налаштовує ідентичні потоки даних NetFlow для надсилання на декілька хостів

1.3.4 sFlow як система моніторингу

Стандарт sFlow описує механізм для захоплення даних трафіку в комутованих або маршрутизованих мережах. Він використовує технологію вибірки для збору статистичних даних із пристрою, і з цієї причини застосовується до високошвидкісних мереж (зі швидкістю гігабіт або вище).

Агент sFlow — це реалізація механізму вибірки на апаратному забезпеченні (наприклад, комутатор). Колектор sFlow — це центральний сервер, який збирає дейтаграми

1.4 Аналіз протоколів моніторингу трафіку в ієрархічних комп'ютерних мережах.

1.4.1 Ресурси агента

Порівнюючи технології для використання у вбудованих програмах моніторингу, необхідно враховувати три основні ресурси, які споживатиме будь-яке вбудоване рішення моніторингу трафіку: процесор, пам’ять і пропускна здатність.

Ці ресурси є дорогими, і вбудоване рішення по вимірюванню має мінімізувати споживання ресурсів, щоб бути економічно ефективним. Наступні діаграми порівнюють RMON, NetFlow і sFlow у кожному з трьох ресурсів [15].

1.4.1.1 Центральний процесор

Обчислювальні вимоги до моніторингу трафіку значно впливають на вартість агента та масштабування. Обчислювально-інтенсивні методи моніторингу вимагають високопродуктивного центрального процесора мережевого керування в комутаторі або маршрутизаторі, що збільшує його вартість. Крім того, один процесор може не впоратися з вимогами моніторингу великої кількості портів, які містяться в багатьох комутаторах, тому для моніторингу можуть знадобитися додаткові ЦП. Порівняння навантаження ЦП, необхідного для моніторингу трафіку за допомогою кожної з різних технологій, показано на рис. 1.9.

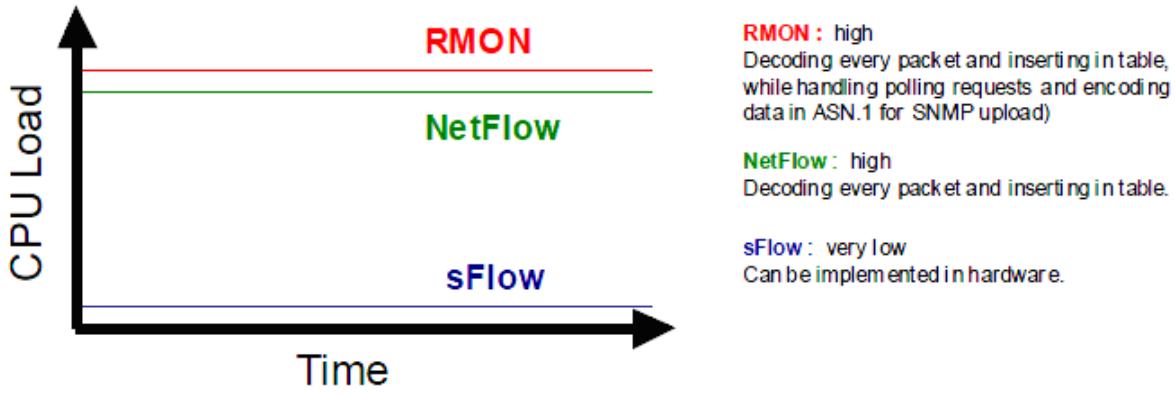

Рис. 1.9. Порівняння навантаження агента ЦП

І RMON, і агенти NetFlow намагаються побудувати матриці трафіку, декодуючи кожен пакет. З огляду на те, що канал зі швидкістю 100 Мбіт/с теоретично може перевищувати 200 000 пакетів за секунду, це створює найгірше навантаження, яке наситить усі ЦП, крім найшвидших на сьогодні. Навіть завантаження лише 10 000 пакетів на секунду означає, що пакет має оброблятися кожні 100 мікросекунд. (Якщо ЦП не встигає, пакети пропускаються. Таким чином, результати можуть бути систематично зміщеними.) [16].

Побудова матриці трафіку в пам’яті агента створює змінне, непередбачуване навантаження на ЦП, залежно від характеру моделей трафіку. І RMON, і NetFlow

страждають від цих змінних накладних витрат. У випадку RMON, центральний процесор також повинен увійти в діалог із станцією керування мережею, щоб завантажити результати за останній період. Упорядкування даних у форматі ASN.1 для завантаження SNMP спричиняє додаткові накладні витрати.

Навпаки, функцію вибірки агентів sFlow можна легко реалізувати апаратним забезпеченням. Таким чином, якщо мережа розривається зі швидкістю 200 000 пакетів на секунду, а частота дискретизації становить 1/1000, центральний процесор агента буде вимагати для обробки лише 200 пакетів в секунду. Оскільки не потрібно нічого робити, крім пересилання заголовка пакета на сервер, це навантаження справді дуже невелике.

1.4.1.2 Пам'ять

Обсяг пам'яті, необхідний для побудови вимірювань трафіку, впливає на вартість агента. Крім того, змінні вимоги до пам'яті викликають особливі проблеми у вирішенні питання скільки пам'яті необхідно включити в пристрій. Занадто мало пам’яті - і моніторингу трафіку не вистачатиме пам’яті, а дані будуть втрачені, занадто багато пам’яті додає агенту непотрібних витрат. Різні технології моніторингу з точки зору обсягу та мінливості їх вимог до пам'яті агента порівняно на рис. 1.10.

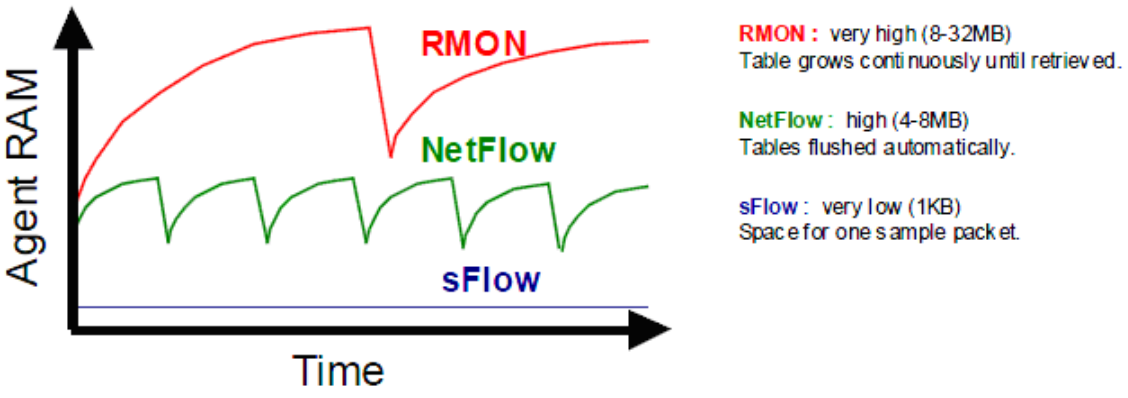

Рис. 1.10. Порівняння використання пам'яті агента

Агенти RMON і NetFlow створюють матриці трафіку в оперативній пам’яті агента. Розмір таблиць значною мірою залежить від схем трафіку. У найгіршому випадку кожен пакет може представляти новий потік, для якого адреси та

лічильники повинні зберігатися окремо. Перевага NetFlow полягає в тому, що його можна налаштувати на автоматичне очищення окремих потоків кожні 15 хвилин або близько того. Агент NetFlow також може очищати блоки потоків, щоб запобігти вичерпанню пам'яті. Однак агент RMON повинен зберігати всю матрицю трафіку в пам'яті та запустити нову на наступний період, поки він очікує на сервер, щоб отримати результати. Додаткові вимірювання RMON, наприклад, для найбільших розмовників, потребують ще більше пам'яті агента. Агенту sFlow достатньо лише оперативної пам'яті, щоб зберегти один пакет. Коли береться зразок, він негайно пересилається на сервер.

1.4.1.3 Пропускна здатність

Передача вимірювань від агентів моніторингу до центрального колектора для звітності та аналізу може споживати значну пропускну здатність. Пропускна здатність, яка використовується для моніторингу мережі, відбирається від мережевих програм.

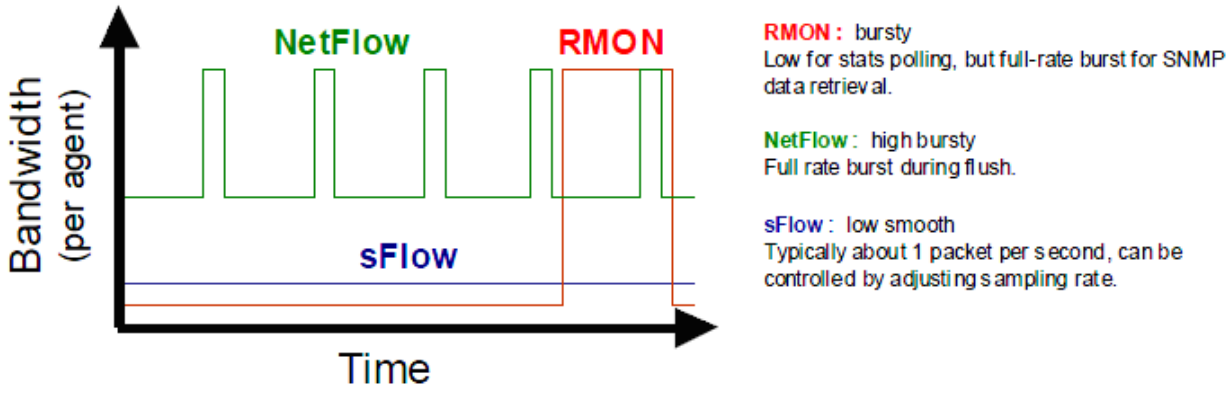

Рис. 1.11. Порівняння пропускної здатності

Також важлива інтенсивність передачі - високі пікові швидкості передачі можуть спричинити значні затори в мережі. Різні технології моніторингу з точки зору пропускної здатності мережі порівнюються на рис. 1.11. Навантаження на мережу від агента NetFlow відображається як потік великих пакетів UDP. Зазвичай вони містять потоки, про які відомо, що вони завершені або мають періодичне очищення, але коли таблиця потоків заповнюється, велика кількість потоків може очиститися одночасно, щоб звільнити простір. Агент на помірно завантаженому

каналі може генерувати в середньому близько 30 пакетів за секунду. Агент RMON збирає всю матрицю трафіку в пам'яті без додаткового очищення. Наприкінці періоду збору (наприклад, 1 година) виникає сплеск мережевої активності, оскільки сервер отримує його. Цей сплеск створює проблему планування для сервера: наприкінці години він хоче зібрати дані про трафік за останні години від кожного агента в мережі, бажано якнайшвидше, оскільки оперативна пам'ять агента може заповнюватися даними за наступні години. Це обмежує кількість агентів, якими можна керувати за допомогою одного сервера. Будь-які додаткові вимірювання RMON призводять до подальшого збільшення мережевого трафіку, оскільки сервер повинен налаштувати кожне вимірювання та отримувати його результати окремо. Процес вибірки агентів sFlow забезпечує постійний потік пакетів від усіх агентів до сервера. Просте налаштування частоти дискретизації, яку використовує кожен агент, може контролювати рівень трафіку.

1.4.2 Ресурси сервера

Агент моніторингу трафіку є лише невеликою частиною загального рішення для керування трафіком. Центральний сервер трафіку потрібен для конфігурації агентів, завантаження й архівування даних, а також надання даних через інтерфейс користувача. Ключовим фактором у визначенні вартості та масштабованості системи моніторингу трафіку є визначення ресурсів сервера, необхідних для роботи з одним агентом.

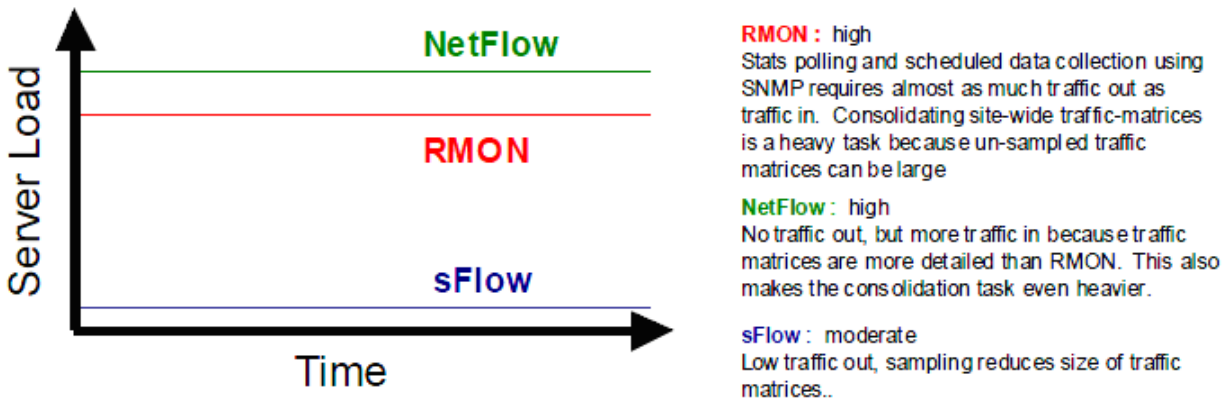

Рис. 1.12. Порівняння навантаження на сервер

Споживання ресурсів на сервері визначає, скількома агентами може керувати один сервер, і значно впливає на вартість і масштабованість загальної системи моніторингу трафіку. Кожна технологія з точки зору ресурсів сервера, необхідних для управління агентом, порівнюється на рис. 1.12. Сервери RMON і NetFlow страждають від великого розміру їхніх матриць трафіку агентів. Крім зусиль щодо отримання їх через мережу, ще існує завдання консолідації їх у матриці трафіку сайту (стежте за тим, щоб не рахувати один і той же потік більше одного разу, якщо його бачив більше ніж один агент). Для RMON запитування лічильників сегментів і збір даних про найбільших користувачів становить значне додаткове навантаження.

Розмір матриці трафіку, побудованої з вибіркових пакетів є значно менший. Зазвичай він складається з найбільш завантажених потоків, а також випадкового вибору менших потоків (багато потоків у мережі складаються лише з 4 або 5 пакетів). Тому завдання консолідації цих матриць трафіку набагато легше. Сервер також має перевагу в тому, що лічильники сегментів включені до зразків, і найбільших розмовників також можна обчислити з тих самих зразків.

1.4.3 Функції

Порівняння технологій виключно з точки зору масштабування та споживання ресурсів було б безглуздим. Важливо також порівняти функціональні можливості кожної системи. Наступні три області представляють ключові функціональні області, які має охоплювати система моніторингу трафіку.

1.4.3.1. Статистика сегментів у реальному часі

Відстеження та тенденції статистики сегментів для кожного посилання в мережі є основою ефективного керування мережею.

- RMON. Підтримується, але потрібно запитувати з сервера за допомогою SNMP. Це обмежує кількість агентів, яких можна контролювати.

- NetFlow**.** За межами специфікації. Може використовувати SNMP для отримання лічильників сегментів від більшості пристроїв.

- sFlow**.** Підтримується – доступний від кожного агента постійно без потреби опитування (лічильники інтерфейсів включені у зразки від агента).

Щоб сервер RMON збирав щохвилинну статистику сегментів, він повинен використовувати SNMP для опитування всіх агентів у мережі принаймні щохвилини. Синхронізувати всі ці вимірювання так, щоб вони починалися і закінчувалися на тій самій хвилинній межі, очевидно, неможливо. Опитування має бути розподілено в часі. Накладні витрати SNMP обмежують кількість лічильників, які можна запитати в одному пакеті. Додатковий трафік може бути значним. Більшість рішень RMON реагують, знижуючи порогове значення сигналізації на агента, і таким чином втрачають здатність з часом корелювати між сегментами. Специфікація NetFlow не стосується питання статистики сегментів. Зазвичай SNMP можна використовувати для отримання статистики сегментів. Проблеми з використанням SNMP для отримання статистики сегментів описані в абзаці про RMON.

За допомогою агента sFlow апаратні лічильники сегментів від кожного агента об’єднуються в потік вибірки з ефективним кодуванням. Таким чином, сервер може бачити актуальні лічильники з кожного сегмента без необхідності запитувати їх окремо. Повну статистику сегментів можна обчислити для кожного сегмента за кожну хвилину, синхронізовану з тією самою глобальною хвилинною межею.

Недостатньо знати, що сегмент перевантажений. Щоб щось з цим зробити, необхідно знати чому. Відповідь дають вимірювання найбільших користувачів.

- RMON. Підтримується як точкове дослідження - зазвичай 1 вимірювання на одному агенті за раз. Спричиняє накладні витрати на трафік, оскільки результати потрібно отримати за допомогою SNMP. Це обмежує кількість агентів, яких можна моніторити.

- NetFlow. Не підтримується.

- sFlow. Найкращі джерела, пункти призначення та пари для кожного протоколу, щохвилини на кожному агенті та порті – постійно. Немає накладних витрат на трафік, оскільки всі результати отримано з вхідних зразків.

За допомогою RMON для збору інформації про найбільших розмовників від агента потрібно налаштувати вимірювання та отримати результати. Робити це щохвилини для кожного сегмента мережі становить величезне завдання для сервера.

Крім того, обчислення найбільших розмовників у агенті RMON є дуже дорогим з точки зору як ЦП, так і оперативної пам’яті. У системі RMON найбільші розмовники реально можуть використовуватися лише як дослідження точок (можливо, як подальше вимірювання, коли лічильник сегментів перетинає порогове значення). NetFlow не підтримує вимірювання в режимі реального часу найбільших користувачів. Через затримку передачі записів потоку від агента до сервера сервер не може надавати точні дані в реальному часі. За допомогою sFlow найбільші користувачі обчислюються на сервері з того самого потоку зразків, який використовується для створення матриць трафіку. Обчислити найбільших користувачів із вибіркових даних набагато легше, ніж переглядати кожен пакет. Сервер може запропонувати найкращі джерела, найкращі пункти призначення та найкращі розмовні пари для кожного протоколу на кожну хвилину для кожного агента та порту. Можна додати додаткові вимірювання, такі як головні джерела багатоадресної IP-адреси, без збільшення навантаження на мережу чи агента. Сервер може навіть корелювати між тисячами сегментів і трендувати основні потоки в межах сайту чи підмережі.

1.4.3.2 Матриці трафіку

Матриці трафіку для всього сайту важливі для планування потужності, виставлення рахунків, оптимізації топології та звітності. Вони являють собою знімок пропонованого навантаження на мережу, який не залежить від того, як базова топологія насправді його несе. Таким чином, дані залишаються доречними та корисними, навіть якщо вузли переміщуються та топологія змінюється.

- RMON. великий набір (обмежений агентом).
- NetFlow. IP, ICMP, TCP, UDP (обмежений агентом).
- sFlow. Розширений пакет (не обмежений агентом, оскільки декодування виконується на сервері. Отже, нові матриці трафіку можна додавати будь-коли без оновлення агентів).

Стандарт RMON пропонує широкий набір матриць трафіку, хоча сумнівно, що агент RMON матиме достатньо оперативної пам’яті, щоб побудувати їх усі одночасно (навіть якби він мав, все одно існує проблема передачі даних на сервер

для консолідації). Іншим недоліком є те, що агенти повинні бути оновлені, щоб підтримувати нове декодування протоколу. NetFlow підтримує лише IP, ICMP, TCP і UDP. За допомогою sFlow декодування виконується на сервері, і немає додаткового навантаження на мережу чи агента, якщо обчислюється кілька матриць трафіку [17].

1.5 Висновки до розділу 1

Основна ідея цього розділу полягала в огляді рішень системи моніторингу трафіку в ієрархічних комп’ютерних мережах.

У першій частині було розглянуто основні завдання та мету першого розділу.

У другій частині розглядалася ієрархічна комп'ютерна мережа як об'єкт моніторингу. У цьому розділі описано, що таке ієрархічна комп’ютерна мережа, топології фізичної мережі, ефективність проектування ієрархічної мережі, плоска топологія у порівнянні з ієрархічною, плоска топологія глобальної мережі, плоска топологія локальної мережі, сітчаста в порівнянні з ієрархічно-сітчастою топологією, класична трирівнева ієрархічна модель. Використання ієрархічної моделі може допомогти нам мінімізувати витрати. Ми можемо придбати відповідні міжмережеві пристрої для кожного рівня ієрархії, таким чином уникаючи витрачання грошей на непотрібні функції для рівня. Крім того, модульний характер ієрархічної моделі проектування дає змогу точно планувати пропускну здатність на кожному рівні ієрархії, таким чином зменшуючи марну пропускну здатність. Відповідальність за керування мережею та системи керування мережею можуть бути розподілені між різними рівнями модульної мережевої архітектури для контролю витрат на управління.

У третій частині описані методи, моделі та алгоритми моніторингу трафіку в ієрархічних комп'ютерних мережах. Існує кілька підходів до моніторингу мережевого трафіку, кожен із яких має різні сильні та слабкі сторони. На даний момент існує три основні варіанти моніторингу трафіку – RMON, NetFlow, sFlow.

У четвертії частині описано аналіз протоколів моніторингу трафіку в ієрархічних комп'ютерних мережах. Результати цієї частини наведено в таблиці 1.3, яка підсумовує різні технології моніторингу з точки зору їх масштабованості та програм, для яких вони найкраще підходять.

Таблиця 1.3. Порівняння технологій моніторингу трафіку

	RMON	NetFlow	sFlow
Масштабування (максимальна кількість портів на сервер)	~100	~10	~50 000
Рекомендоване застосування	Віддалений аналізатор протоколів, направлений для усунення несправностей.	Виставлення рахунків і моніторинг безпеки помірних швидкісних з'єднань глобальної мережі на маршрутизаторах	Моніторинг всіх портів комутатора. Високошвидкісний моніторинг магістральних каналів. Управління та безпека перевантаження білінгу

Збираючи щохвилинну статистику сегментів і найбільших розмовників, а також консолідацію матриці трафіку по всьому сайту, важко зрозуміти, як будь-який сервер RMON може підтримувати більше 100 агентів одночасно. З огляду на високу вартість одиниці технології агента RMON, навіть це досить низьке число додається до непомірно дорогого рішення. Стандарт RMON визначає, по суті, віддалений аналізатор протоколів. Його здатність фільтрувати, перехоплювати та декодувати пакети робить його застосовуваним для тих випадків усунення несправностей, коли проблему можна зрозуміти, лише побачивши послідовності протоколів із міткою часу на проводі. Технологія NetFlow може бути корисною на з'єднаннях глобальної мережі, де важливо бачити кожен потік, можливо, для моніторингу безпеки або конкретного поетапного виставлення рахунків. Величезний обсяг згенерованих даних означає, що сервер не може керувати більш ніж 10 зв'язками. Техніка статистичної вибірки, яку використовує sFlow, добре масштабується для великої кількості агентів, десятки тисяч портів комутатора

можуть керуватися одним сервером. Нульова вартість агента та значні технічні переваги роблять його ідеальним для безперервного моніторингу трафіку на сайті (і в масштабах підприємства) та звітності.

2 РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ В ІЄРАРХІЧНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

2.1 Проектування

2.1.1 Архітектура RMON

RMON базується на архітектурі клієнт/сервер, де агент відіграє роль сервера, як показано на рис. 2.1. Кілька клієнтів або програм RMON можуть використовувати агента одночасно. Адміністратори налаштовують агента, щоб пропонувати вибрані перегляди різним членам команди керування. Агенти RMON використовують персонал, дозволяючи їм залишатися на централізованому сайті,

збираючи інформацію з широко розсіяних сегментів локальної мережі (LAN). Постійний моніторинг надає інформацію до виникнення проблем, дозволяючи віддаленому персоналу застосовувати проактивний підхід до управління. Проактивне управління покращує якість обслуговування та підвищує доступність, оскільки проблеми виникають рідше.

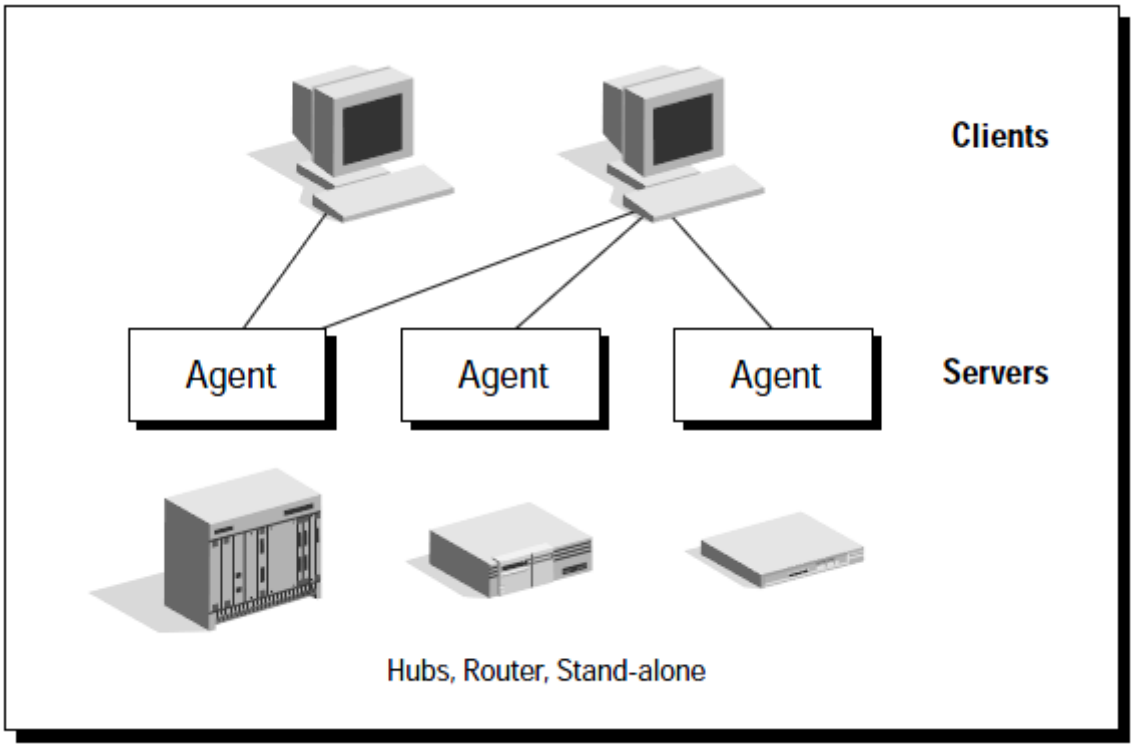

Рис. 2.1 Архітектура клієнт/сервер

Додаток RMON встановлює відповідні змінні MIB для визначення інтервалів вимірювання, порогових значень та інших робочих параметрів. Агент збирає та зберігає інформацію і передає її віддаленому клієнту за запитом. Агенти також надсилають пастку SNMP, коли виявляються певні умови, таким чином сповіщаючи віддаленого клієнта про ситуацію, яка потребує негайної уваги.

Агенти є універсальними інструментами, у наступних двох розділах ми розглянемо деякі способи їх групування та використання.

Варіанти групування

Адміністратори можуть розглянути різні варіанти групування: на вибір є автономні та вбудовані агенти. Автономні агенти є портативними та замкненими в апаратному пристрої, який зазвичай називають зондом. Деякі багатопортові агенти доступні для моніторингу набору сегментів, зменшуючи витрати на сегмент.

Вбудовані агенти розміщуються в таких пристроях, як центри, маршрутизатори та комутатори. Агенти в центрах можуть перемикатися між системними платами, дозволяючи одному агенту переміщатися по різних сегментах локальної мережі в центрі. Агенти в маршрутизаторах відстежують активність на інтерфейсах локальної мережі під дистанційним керуванням. Комутатори локальної мережі містять агенти RMON, які можуть переміщатися по портах комутатора. Необхідно перевірити продукти вбудованого агента, щоб визначити, чи призведуть вони до зниження продуктивності під час активного використання агента. Нова опція розміщує основні можливості RMON на мережевих інтерфейсних картах, вимикаючи завантаження мережевих пристроїв і зберігаючи ресурси.

Використання RMON

Підтримка високої доступності мережі та сталої якості обслуговування є основною роботою будь-якого адміністратора мережі. Віддалені агенти полегшують виконання цих завдань, надаючи швидшу відповідь на проблеми в режимі реального часу, оскільки час на дорогу не витрачається. Це безпосередньо означає більшу доступність і кращу якість обслуговування. Додаткова економія призводить до того, що співробітникам не потрібно постійно перебувати в сегменті локальної мережі, щоб пришвидшити реагування на збої. Крім того, агент уже на місці та має значні інтелектуальні можливості, що дозволяє автоматично ініціювати діагностичні вимірювання, коли в цьому сегменті локальної мережі виявляються певні умови.

Агент має велику цінність для тактичного управління — виявлення проблем, визначення причин і підтримання високої доступності та якості. Мережеві адміністратори завжди намагаються створити середовище проактивного керування; таке, у якому проблеми передбачають і виправляють до того, як вони матимуть суттєвий вплив на спільноту користувачів і на здатність організації виконувати свою підприємницьку діяльність. Постійний моніторинг дає мережевим адміністраторам нову можливість рухатися в цьому напрямку. Наприклад, зібрана інформація може бути використана для створення базових показників діяльності, що визначають нормальну поведінку будь-якого конкретного сегмента локальної мережі. Адміністратори отримують сповіщення, коли поведінка мережі виходить за межі нормального діапазону, і вони можуть запобігти проблемам до того, як їх

помітять користувачі. Охарактеризувати поведінку складних об'єднаних мереж важко. Наприклад, проблема між клієнтом в одному сегменті та сервером у кампусі включає в себе кілька сегментів локальної мережі з власним трафіком. Інформацію з набору агентів RMON2 можна корелювати для глибшого розуміння трафіку, що проходить між сегментами. Нові агенти надають інформацію про потоки трафіку між підмережами, розподіл трафіку протоколів і активність додатків. Здатності фільтрації агентів RMON2 дозволяють ще більш точні рівні деталізації. Наприклад, пакети можна відфільтрувати за такими критеріями, як тип протоколу або використання програми, забезпечуючи дуже детальний перегляд окремих розмов у сегменті локальної мережі.

2.1.2 Архітектура NetFlow

Мережевий потік NetFlow визначається як односпрямований потік пакетів між даним джерелом і джерелом призначення. Потік визначається комбінацією семи ключових полів, а інформація NetFlow стискається в базу даних під назвою кеш NetFlow. Сім полів потоку та кеш показані на рис. 2.2.

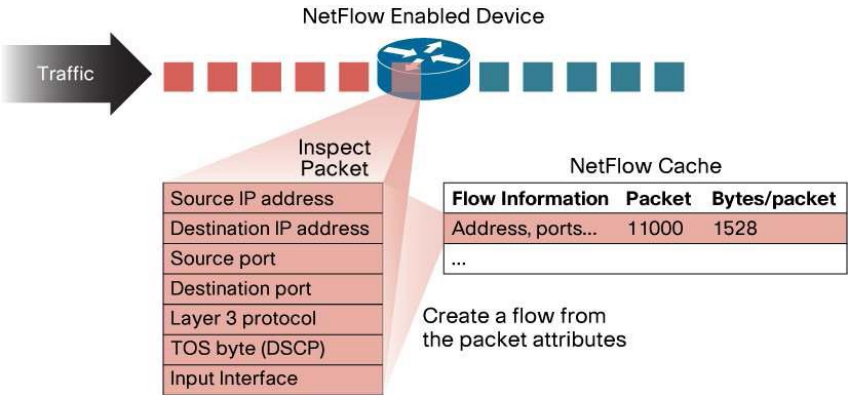

Рис. 2.2 NetFlow: структура протоколу моніторингу та керування мережевим трафіком Cisco

Записи NetFlow експортуються за допомогою деяких спеціально визначених схем Cisco. У наступній таблиці коротко описано функції постобробки. Користувач може конфігурувати ці функції, щоб налаштувати експорт даних NetFlow [12].

Система Netflow має три основні компоненти: *датчик, колектор і певну систему звітності*.

Датчик (також відомий як зонд) — це деймон, який слухає мережу та фіксує дані сеансу. Так само, як у випадку з Snort або будь-якою іншою системою IDS, колектору потрібно підключитися до центру, «дзеркального» порту комутатора або іншого пристрою, де він може бачити весь мережевий трафік. Якщо ми використовуємо брандмауер BSD або Linux, це чудове місце для запуску колектора NetFlow, увесь трафік має проходити через цей пристрій. Цей датчик об'єднує інформацію про сеанс і закидає її до колектора.

Колектор — це другий демон, який прослуховує через UDP-порт звіти від датчика та скидає їх у файл для подальшої оцінки. Різні колекціонери зберігають свої дані в різних форматах файлів.

Нарешті, система звітності зчитує файли, створені колектором, і створює звіти, зрозумілі людині. Система звітності повинна мати можливість читати формат, який використовує колектор. (рис. 2.3).

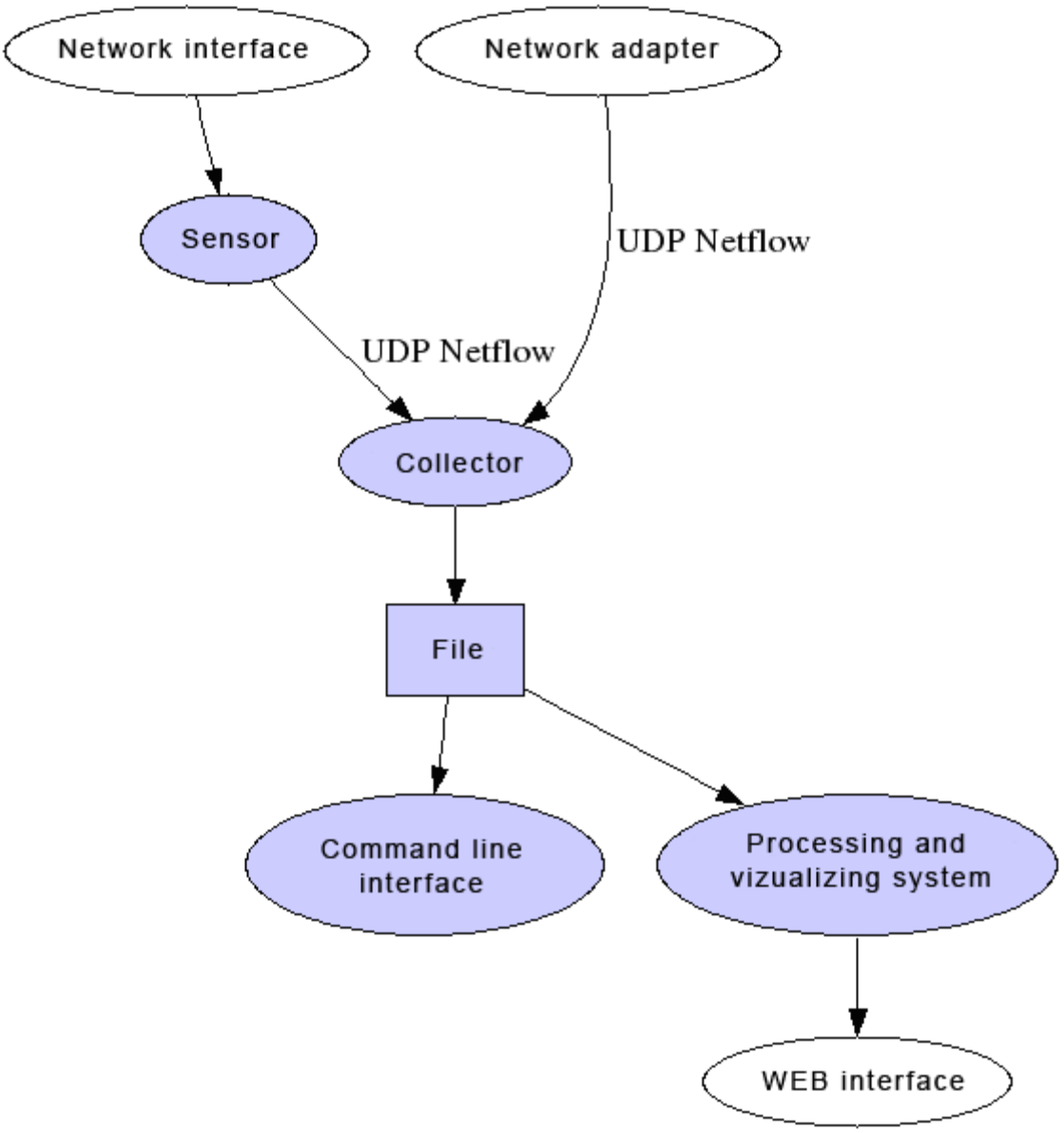

Рис. 2.3. Архітектура NetFlow

2.1.3 Архітектура sFlow

sFlow — це технологія вибірки від багатьох постачальників, вбудована в комутатори та маршрутизатори. Вона забезпечує можливість безперервного моніторингу потоків трафіку на рівні програми на швидкості з'єднання на всіх інтерфейсах одночасно [13].

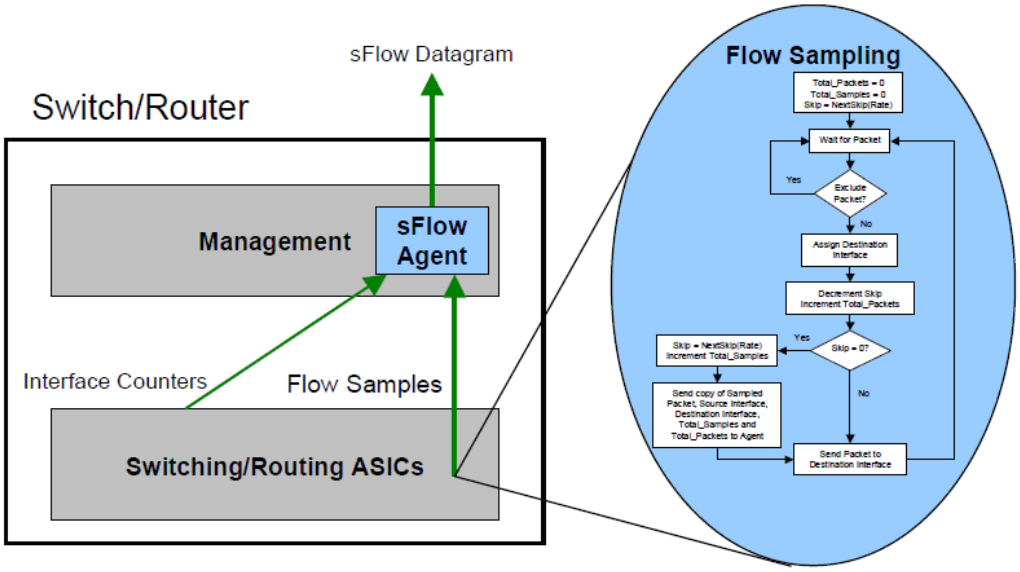

Рис. 2.4 Агент sFlow, вбудований у комутатор/маршрутизатор

Агент sFlow — це програмний процес, який виконується як частина програмного забезпечення для керування мережею на пристрої (див. рис. 2.4). Він поєднує лічильники інтерфейсів і зразки потоку в дейтаграми sFlow, які надсилаються через мережу до колектора sFlow. Вибірка пакетів зазвичай виконується ASIC комутацією/маршрутизацією, що забезпечує швидкість передачі даних. Також записується стан записів таблиці пересилання/маршрутизації, пов'язаних із кожним відібраним пакетом.

Агент sFlow обробляє дуже мало. Він просто пакує дані в дейтаграми sFlow, які миттєво надсилаються в мережу. Миттєве пересилання даних мінімізує вимоги до пам'яті та ЦП, пов'язані з агентом sFlow [14].

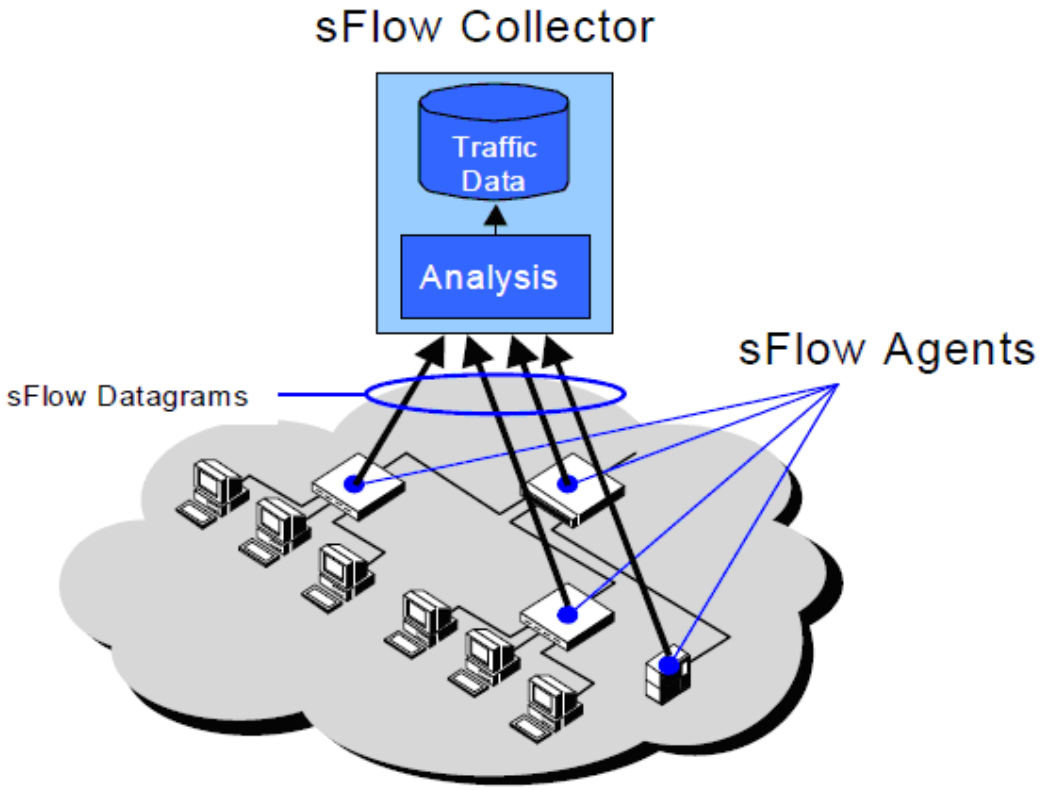

Рис. 2.5. Агенти та колектор sFlow

Основні елементи системи sFlow показані на рис. 2.5. Агенти sFlow по всій мережі безперервно надсилають потік дейтаграм sFlow до центрального колектора sFlow, де вони аналізуються для отримання детального перегляду потоків трафіку в реальному часі в усій мережі. sFlow моніторинг високошвидкісних, маршрутизованих і комутованих мереж має такі властивості:

- Точність**.** Оскільки вибірка є досить простою для виконання в апаратному забезпеченні, вона працює на швидкості проводу. Крім того, система sFlow розроблена таким чином, що можна визначити точність будь-якого вимірювання. Інші технології вимірювання потоку трафіку під час великих навантажень спричиняють похибки, які важко оцінити кількісно

- Деталізація**.** Повний заголовок пакета та інформація про комутацію/маршрутизацію дозволяють детально аналізувати потоки трафіку L2-L7.

- Масштабуваня**.** Система sFlow масштабується як за розміром, так і за швидкістю мережі, яку вона може моніторити. sFlow здатна моніторити мережі зі швидкістю 10 Гбіт/с, 100 Гбіт/с і вище. Тисячі пристроїв можна моніторити за допомогою одного колектора sFlow.

- Низька вартість**.** Агент sFlow дуже простий у застосуванні та додає незначні витрати на комутатор або маршрутизатор.

- Оперативність**.** Колектор sFlow завжди має з точністю до хвилини інформацію про трафік по всій мережі. Своєчасна інформація особливо важлива, якщо дані про трафік потрібні для забезпечення контролю в реальному часі. Наприклад, щоб керувати якістю обслуговування або захищатися від атак на відмову в обслуговуванні.

2.1.4 Ефективне вимірювання трафіку за допомогою програмного забезпечення ntop

Вимірювання трафіку стає дедалі складнішим через різноманітність типів трафіку та інтеграцію різних мережевих засобів. Хоча традиційні інструменти, такі як аналізатори та мережеві зонди, все ще корисні для вимірювання трафіку, вони часто обмежують свою сферу застосування аналізом мережевих пакетів і візуалізацією. Це означає, що люди-оператори повинні вручну виконувати певні дії, коли трапляються якісь аномальні умови руху, оскільки вони мають дуже обмежену підтримку (якщо вона взагалі є) із зазначених вище інструментів. У цій статті розглядається розробка та реалізація ntop, веб-програми для вимірювання та моніторингу трафіку з відкритим кодом. Це дозволяє користувачам відстежувати відповідну мережеву діяльність, включаючи характеристику трафіку, використання мережі, використання мережевого протоколу та виявлення перевантажень[21]. Розширюваність Ntop за допомогою компонентів програмного забезпечення, що динамічно завантажуються, відкриває його для розширень мережевими адміністраторами. Крім того, власні засоби виявлення недоліків безпеки дозволяють ntop виявляти потенційно небезпечні умови трафіку, а отже, динамічно та автономно адаптувати конфігурацію мережі для вирішення виявлених проблем.

2.1.5 Передумови та мотивація

Вимірювання мережевого трафіку вважалося необхідною діяльністю з перших днів створення мереж. Адміністратори повинні були стежити за мережевим трафіком з кількох причин, включаючи виявлення вузьких місць у мережі та планування розширення мережі. В останні кілька років ця діяльність стає все більш складною з кількох причин:

- Інтеграція кількох мережевих медіа все більше ускладнює виявлення проблем із зв'язком і вимагає мережевих аналізаторів і зондів, здатних не відставати від збільшення швидкості мережі.

- Взаємозв'язок існуючих автономних мереж, частково заснованих на IP, призводить до того, що протоколи, що не є IP, такі як NetBEUI, AppleTalk і IPX, безшумно поширюються в існуючі IP-мережі, заважаючи існуючим протоколам і знижуючи продуктивність мережі.

- Користувачі самостійно адмініструють свої ПК, що ще більше ускладнює роботу адміністратора мережі, оскільки він має обмежений контроль над такими хостами.

- Адміністратори мережі повинні постійно контролювати використання протоколів, які значною мірою використовують пропускну здатність мережі і, отже, значно впливають на загальну продуктивність.

UNIX традиційно надає інструменти для тестування основних проблем підключення, а також мережеві аналізатори, такі як tcpdump або snoop. Ці інструменти дуже потужні для відстеження проблем підключення до мережі та протоколу. Однак їм потрібні інструменти автономного аналізу, такі як tcpshow і tcptrace, для кращого аналізу та кореляції отриманих даних, а також ідентифікації мережевих потоків. Подібним чином, мережеві зонди, такі як агенти віддаленого моніторингу (RMON), є досить потужними, але, на жаль, потребують складних менеджерів простого протоколу керування мережею (SNMP), здатних правильно їх налаштувати та інструментувати, а також аналізувати зібрані дані. Через таку складність, а також вартість таких зондів, агенти RMON в основному використовуються виключно менеджерами мереж у великих установах. Інші інструменти для моніторингу мережі, такі як NeTraMet, пропонують розширені мови програмування для аналізу мережевих потоків і створення статистичних

записів подій. Незважаючи на те, що ці інструменти пропонують великі переваги з точки зору гнучкості та конфігурації користувача щодо агентів RMON, вони все одно вимагають менеджера SNMP для збору даних про трафік і зазвичай не надають адміністраторам засоби для ініціювання дій при виявленні певних моделей мережевого трафіку [22].

В Інтернеті є кілька інструментів, розроблених для виявлення недоліків мережевої безпеки та потенційних атак. Окрім деяких винятків, таких як мережевий бортовий самописець (NFR), ці інструменти зазвичай розроблені для виявлення атак на один хост (зазвичай той, де активовано інструмент) і не забезпечують захист мережі/підмережі. Інструменти вимірювання трафіку зазвичай не забезпечують підтримки безпеки; не дозволяють активних дій під час атак; вони просто повідомляють адміністраторів, коли атака вже сталася. Це пояснюється тим, що інструменти вимірювання класифікують мережевий трафік відповідно до певних статичних правил із визначеними пороговими значеннями. Ці порогові значення часто або не в змозі виразити складні шаблони трафіку (наприклад, атаки на безпеку), або достатньо гнучкі, щоб охопити всю підмережу без необхідності визначати однакові правила для всіх хостів підмережі.

Ntop — це веб-додаток для вимірювання та моніторингу трафіку. Його спочатку написали автори, щоб вирішити проблеми з продуктивністю магістральної мережі кампусу, оскільки доступні інструменти моніторингу трафіку не були задовільними з причин, перерахованих вище. Подібно до основного інструменту UNIX, який звітує про використання ЦП процесами, авторам потрібен був простий інструмент, здатний вимірювати мережевий трафік і повідомляти інформацію про захоплені пакети. Потім ntop перетворився на більш гнучкий, розширений і потужний інструмент, оскільки люди завантажували його через Інтернет і повідомляли про проблеми та давали пропозиції. Наступний розділ охоплює дизайн архітектури ntop, її компоненти та їх взаємодію, а також деталі реалізації та хитрощі, які використовуються для ефективної реалізації. Потім ми показуємо, як ntop можна ефективно використовувати як для вимірювання трафіку, так і для виявлення вторгнень у деяких реальних сценаріях. Нарешті, аналізуються та обговорюються деякі проблеми продуктивності.

2.1.6 Цілі проектування архітектури ntop

Ntop — це програма з відкритим вихідним кодом, написана мовою C, доступна безкоштовно за загальнодоступною ліцензією GNU. Це твердження не означає, що лише вихідний код ntop вільно доступний в Інтернеті, але також те, що багато вимог надійшли безпосередньо від перших користувачів ntop. Автори розробили першу версію ntop, а потім розмістили нові вимоги та розширення оригінальної архітектури, на яку сильно вплинула архітектура Webbin. Основні цілі дизайну Ntop включають:

- Портативність між платформами UNIX і non-UNIX (наприклад, Win32).

- Просте та ефективне ядро програми з низьким використанням ресурсів (як пам'яті, так і ЦП).

- Можливість моніторингу та керування мережею з віддаленого місця без необхідності запускати спеціальні клієнтські програми для аналізу інформації про трафік.

- Мінімальні вимоги (гола операційна система), але можливість використовувати функції платформи, якщо вони є (наприклад, потоки ядра).

- Можливість представлення даних як у символьному терміналі, так і у веб-браузері.

- Вихід аналізу трафіку багатий вмістом і легкий для читання.

- Розширення користувача за допомогою динамічно завантажуваних програмних компонентів (плагінів).

Ідея полягає в тому, щоб розробити просте та ефективне ядро програми, здатне виконувати загальні завдання, зокрема:

- Захоплення та демультиплексування пакетів незалежно від операційної системи (той самий код має працювати без змін на різних платформах) і типу інтерфейсу мережевої інтерфейсної карти (NIC), який використовується для захоплення пакетів.

- Базовий аналіз трафіку та характеристика протоколу.

- Специфікація та вимірювання мережевих потоків

- Вбудований HTTP-сервер для візуалізації даних трафіку без необхідності використання спеціальної клієнтської програми.

Дизайн Ntop поділяє філософію UNIX: програми не обов’язково мають бути великими та монолітними, але їх можна вигідно розділити на невеликі незалежні частини, які співпрацюють для досягнення спільної мети. Ядро відповідає за ефективне виконання основних завдань і надання можливостей розробникам плагінів, які можуть використовувати служби ядра. Це дозволяє розробникам знизити складність плагіна та зосередитися лише на функціональності плагіна, оскільки ядро виконає всі інші завдання. Використання плагінів дозволяє користувачам активувати лише ті з них, які потрібні, залежно від конкретної ситуації, у якій використовується ntop, таким чином уникаючи витрачання дорогоцінних циклів ЦП на плагіни, які надають інформацію, яка не є важливою в конкретному контексті. Крім того, він використовує складність плагінів, оскільки багато основних служб надаються ядром ntop, що спрощує реалізацію нових плагінів. Перш ніж аналізувати сценарії, у яких використовується ntop, варто ознайомитися з внутрішніми елементами програми, її компонентами та взаємодією.

2.1.7 Перехоплення пакетів

Перехоплення пакетів — це компонент ntop, який потенційно має більше проблем із портативністю, ніж інші компоненти. Фактично, на відміну від інших засобів, таких як потоки, не існує портативної бібліотеки для захоплення пакетів. UNIX бібліотека libpcap забезпечує портативний і уніфікований інтерфейс перехоплення пакетів, тоді як інші операційні системи надають власні засоби перехоплення. Завдяки гарному дизайну libpcap автори вирішили переносити libpcap навіть на платформи non-UNIX, вбудовуючи в нього власні засоби перехоплення платформи (наприклад, NDIS на Win32). Це дозволило вихідному коду ntop бути унікальним для різних платформ. Бібліотеки перехоплення пакетів часто мають невеликі внутрішні буфери, які не дозволяють програмам обробляти пакетний трафік. Щоб подолати цю проблему і, отже, зменшити втрату пакетів, ntop буферизує захоплені пакети. Це дозволяє аналізатору пакетів бути від’єднаним від захоплення пакетів і не втрачати пакети через пакетний трафік [23].

2.1.8 Аналізатор пакетів

Аналізатор пакетів обробляє по одному пакету. Заголовки пакетів аналізуються відповідно до мережевого інтерфейсу, що використовується. Інформація про хости зберігається у великій хеш-таблиці, записи якої містять кілька лічильників, які відстежують дані, надіслані/отримані хостом, відсортовані відповідно до підтримуваних мережевих протоколів. За потреби (наприклад, періодично або якщо не залишилося записів) ntop очищає таблицю хостів, щоб уникнути виснаження всієї доступної пам’яті та створення величезних таблиць, які знижують загальну продуктивність. Це гарантує, що використання пам’яті ntop не зростає нескінченно, а час обробки пакетів не збільшується лінійно з кількістю активних хостів. Кешування виконується в два етапи. Кешування першого рівня є напівпостійним і базується на GNU gdbm, тоді як кешування другого рівня реалізовано за допомогою бази даних стандартної мови запитів (SQL). Ntop локально кешує напівпостійну інформацію, таку як роздільна здатність адреси IP (відображення числової/символічної IP-адреси) та операційну систему віддаленого хоста (обчислено за допомогою інструменту nmap). Щоб зменшити кількість запитів служби доменних імен (DNS), ntop обробляє пакети відповідей DNS і кешує зіставлення для майбутнього використання. Мережеві події (наприклад, сеанси TCP), дані про продуктивність та інша актуальна інформація постійно зберігаються в базі даних. Зберігання відбувається або періодично, або щоразу, коли збирач сміття має видалити деякі дані. Ntop спілкується з базою даних за допомогою клієнтської програми, яка взаємодіє з базою даних за допомогою інтерфейсу бази даних Perl (DBI) або підключення до бази даних Java (JDBC), залежно від мови реалізації. Ця архітектура дозволяє відокремити ntop від конкретної бази даних і мати можливість спілкуватися з віддаленою базою даних (наприклад, основною базою даних компанії), маючи при цьому дуже просту і легку базу даних клієнта.

2.1.9 Мережеві потоки

Фільтрування пакетів базується на частині об'єкта libpcap - пакетному фільтрі Berkeley (BPF). BPF дозволяє точно визначати фільтри за допомогою простих англійських виразів, таких як ті, що приймаються tcpdump. Для кращої

продуктивності фільтри компілюються та оптимізуються перед їх збереженням у ntop. Мережевий потік — це потік пакетів, який відповідає визначеному користувачем правилу. Правила вказуються за допомогою виразів BPF під час запуску ntop. Подібно до потоків NeTraMet, мережеві потоки ntop можна використовувати для визначення трафіку, що представляє особливий інтерес. Ntop застосовує всі збережені фільтри потоку до кожного захопленого пакета. Коли пакет відповідає фільтру, лічильники потоку оновлюються. Зверніть увагу, що час обробки пакетів збільшується зі збільшенням кількості визначених потоків і складності пов’язаних фільтрів.

2.1.10 Підтримка HTTP

Незважаючи на те, що ntop може надавати інформацію про трафік у текстових терміналах, він дозволяє використовувати переваги веб-технологій. Насправді ядро ntop містить вбудований сервер HTTP/HTTPS, який пропонує користувачам перегляд більшої кількості інформації про трафік, ніж той, який пропонує інтерфейс терміналу (рис. 1). Сервер пропонує автентифікацію HTTP/HTTPS і дозволяє адміністраторам вказувати користувачів, які мають доступ до вибраних URL-адрес. Для більшої безпеки паролі користувачів зберігаються в базі даних у зашифрованому вигляді. Під час першого запуску ntop чутлива інформація про трафік захищена паролями за замовчуванням. Тоді адміністратори можуть повністю налаштувати ntop за допомогою веб-браузера [24].

2.1.11 Плагіни

Плагіни — це спільні бібліотеки (бібліотеки, що динамічно завантажуються, DLL, у термінології Windows) із чітко визначеною точкою входу, що зберігається у вказаному каталозі (плагіни/за замовчуванням). Під час запуску ntop перераховує збережені плагіни та завантажує їх послідовно в алфавітному порядку. Розробники

можуть використовувати плагіни для розширення ядра ntop, визначення користувальницьких переглядів інформації про трафік, яку збирає ntop, і впровадження розширених лічильників потоку трафіку, які виконують додаткові операції, окрім базового вимірювання трафіку.

2.1.12 Вимірювання мережевого трафіку за допомогою програмного забезпечення ntop

Ntop був розроблений, щоб надати нам простий, безкоштовний і портативний інструмент для вимірювання трафіку. Його розробка почалася через те, що ми не були задоволені існуючими інструментами моніторингу трафіку, як описано раніше. Ntop зосереджується на:

- Вимірюванні трафіку;
- Характеристиці та моніторингу трафіку;
- Виявленні порушень безпеки мережі;
- Оптимізації та плануванні мережі.

Наступні розділи детально охоплюють вищевказані області та показують, яку інформацію надає ntop мережевим адміністраторам, щоб допомогти їм визначити недоліки мережі, оптимізувати мережу та спланувати майбутні розширення. Оскільки ntop використовує деякі загальнодоступні інструменти/бібліотеки, наведена нижче таблиця допоможе читачеві зрозуміти, які послуги надають такі інструменти та бібліотеки та з якою метою вони використовуються ntop [25].

2.1.13 Вимірювання трафіку

На нашу думку, вимірювання трафіку складається з вимірювання відповідних дій трафіку. Ntop пов'язує кожен захоплений пакет із хостом відправника/одержувача. Таким чином, враховуючи хост (наприклад, його ім’я, НІК або IP-адресу), можна знайти всю пов’язану з ним діяльність трафіку, яку спостерігає ntop. Для кожного хоста ntop записує таку інформацію:

- Надіслані/отримані дані: загальний трафік (обсяг і пакети), згенерований або отриманий хостом, класифікований відповідно до мережевого протоколу (IP, IPX, AppleTalk тощо) і, якщо застосовується, IP протоколу (FTP, HTTP, NFS тощо).

- Багатоадресна IP-адреса: загальний обсяг групового трафіку (обсяг і пакети), згенерований або отриманий хостом.

- Історія сеансів TCP: список поточних активних сеансів TCP, встановлених/прийнятих хостом, і пов’язана статистика трафіку.

- Трафік UDP: загальний обсяг трафіку UDP (обсяг і пакети), відсортований за портом. Варто зазначити, що можна розпізнати просте сканування портів і сканування протоколів (наприклад, менеджер SNMP надіслав запити SNMP певному хосту), коли хост отримав пакети на вказаний порт, але не надіслав жодних даних.

- Використання служб TCP/UDP: список служб на основі IP (наприклад, відкритих і активних портів), наданих хостом, зі списком п’яти останніх хостів, які їх використовували.

- Тип операційної системи (ОС): хоча вгадати ОС хостів, на яких працює UNIX досить просто (наприклад, достатньо подивитися на банер, що відображається за допомогою команди telnet), в загальному, вгадати ОС є складним завданням. Ntop використовує nmap, який ідентифікує ОС, надсилаючи деякі фальшиві пакети та порівнюючи відповіді, якщо такі є, з базою даних відомих шаблонів.

- Відсоток використання пропускної здатності: фактичне, середнє та пікове використання пропускної здатності.

- Розподіл трафіку: локальний (підмережевий) трафік, локальний у порівнянні з віддаленим (за межами вказаної/локальної підмережі), віддалений у порівнянні з локальним.

- Розподіл IP-трафіку: трафік UDP у порівнянні з трафіком TCP; відносний розподіл IP-протоколів відповідно до назви хоста.

- Використання локальної мережі: статистика про відкриті сокети, надіслані/отримані дані та зв’язані вузли для кожного процесу, запущеного на хості, де активний ntop.

Крім того, ntop повідомляє глобальну статистику трафіку, включаючи:

- Розподіл трафіку: локальний (підмережевий) трафік, локальний у порівнянні з віддаленим (за межами вказаної/локальної підмережі), віддалений у порівнянні з локальним.

- Розподіл пакетів: загальна кількість пакетів, відсортованих за розміром пакета, одноадресний чи багатоадресний або широкомовний, а також IP-трафік у порівнянні з не-IP-трафіком.

- Використання пропускної здатності: фактичне, пікове та середнє використання пропускної здатності.

- Список активних сеансів TCP для кожного відомого хоста.

- Використання та розподіл протоколу: розподіл спостережуваного трафіку як за протоколом, так і за джерелом-адресатом (локальний чи віддалений).

- Матриця трафіку локальної підмережі: двовимірна матриця, де кожна комірка (X, Y) містить трафік, надісланий хостом X до хосту Y, де X і Y є хостами, які належать до локальної підмережі хосту, де запущено ntop.

- Мережеві потоки: статистика трафіку для кожного потоку, визначеного користувачем.

Поточна версія ntop постачається з кількома плагінами, які надають детальну статистику використання протоколу NFS/NetBIOS і відображають пропускну здатність мережі, яку використовують такі протоколи. Ntop відрізняється від багатьох інструментів моніторингу трафіку, оскільки він прозоро обробляє дані трафіку під час захоплення пакетів і надає інформацію про трафік у зручному для читання форматі. Інші інструменти або спочатку збирають дані, залишаючи аналіз трафіку додатковим програмам, або надають базову інформацію про отримані дані, змушуючи користувача писати макроси чи сценарії для вилучення інформації, яка його цікавить [26].

2.1.14 Характеристика та моніторинг трафіку

Моніторинг трафіку — це можливість ідентифікувати ситуації, коли мережевий трафік не відповідає визначеним правилам або перевищує певні порогові значення. Загалом, мережеві адміністратори вказують деякі правила, яким мають підкорятися всі хости.

Ntop надає підтримку для виявлення деяких проблем конфігурації мережі, зокрема:

- Використання повторюваних IP-адрес. Ідентифікація всіх маршрутизаторів підмережі, щоб можна було з'ясувати, чи помилково налаштований хост вважає, що він діє як маршрутизатор для локальної підмережі, чи хост використовує неправильну маску мережі для фактичної мережі.

- Ідентифікація локальних хостів, які встановили мережеву карту в безладному режимі (див. розділ «Виявлення шпигунів» нижче).

- Неправильна конфігурація програмного забезпечення: аналіз даних трафіку деяких протоколів дозволяє адміністраторам припустити, що на певному хості щось не так. Наприклад, використання ntop дозволило нам виявити встановлення несанкціонованого кешування DNS, який дуже часто заповнював свій кеш, і неправильно налаштований клієнт NTP, який запитував час кожні 5 секунд.

- Виявлення неправильного використання служби: щоб зменшити мережевий трафік, адміністратори вимагають, щоб користувачі використовували проксі-додатки (наприклад, HTTP/FTP-проксі) замість прямого підключення до віддаленого сайту. Правильна конфігурація ntop дозволяє адміністратору ідентифікувати хости/користувачів, які не використовують вказані проксі-сервери.

- Неправильне використання протоколу: ідентифікація комп'ютерів, які використовують непотрібні протоколи. Наприклад, ОС Windows за замовчуванням встановлює такі протоколи, як NetBEUI та IPX, тоді як більшість людей використовує лише TCP/IP.

- Надмірне використання пропускної здатності мережі: в організаціях, де підключення до Інтернету має обмежену пропускну здатність, важливо виявити хости/користувачів, які використовують більшу частину доступної пропускної здатності. Наприклад, це можна зробити шляхом відстеження значень трафіку для певних протоколів (наприклад, HTTP чи FTP) або ідентифікації хостів із встановленими з'єднаннями з віддаленими хостами.

Ntop ідентифікує маршрутизатори підмережі, перевіряючи IP-адресу цільової асоціації/адресу керування доступом до середовища (MAC) у кожному захопленому пакеті (а не лише в тих, що спрямовані на нелокальні IP-адреси). Маршрутизатори підмережі ідентифікуються за MAC-адресою призначення, тоді як хости з неправильно налаштованими масками мережі ідентифікуються, оскільки вони надсилають маршрутизатору ті пакети, які спрямовані на хости, що належать до локальної підмережі. Ідентифікація повторюваних IP-адрес і списку маршрутизаторів підмережі виконується плагіном arpWatch. Цей плагін відстежує пакети протоколу розпізнавання адрес (ARP), таким чином ідентифікуючи модифікацію IP-адреси (наприклад, хост змінює свою IP-адресу вручну або за допомогою протоколів, таких як DHCP чи BOOTP) і відбувається зіткнення MAC-адреси із тією самою IP-адресою (хост отримує кілька відповідей на один запит ARP). ArpWatch також використовується для виявлення підробки пакетів, як пояснюється нижче[27].

2.1.15 Діаграма варіантів використання

Тепер попрацюємо над діаграмою варіантів використання. Наша система передбачає декілька їх варіантів, а саме : запуск самої програми, налаштування системи моніторингу трафіку, відстежування мережевої діяльності користувачів та конфігурація функцій для налаштування експорту даних. Сама діаграма зображена на рисунку 2.6

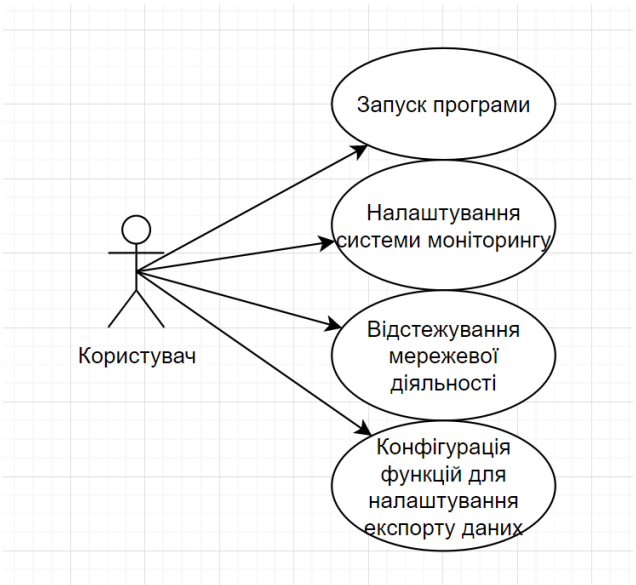

Рис. 2.6 Діаграма Варіантів Використання

2.1.16 Діаграма послідовності

Побудуємо діаграму послідовності для нашої системи. Система передбачає деяку кількість користувачів які користуються мережою, використовуючи трафік мережі, який після цього передається до роутера. Після цього наший колектор(Flow capture) збирає дані з роутера, які після цього передаються на сервер. Після проведенних операцій ми виводимо інформацію через WEB інтерфейс і маємо результат роботи нашої системи. Данна діаграма зображенна на рисунку 2.7

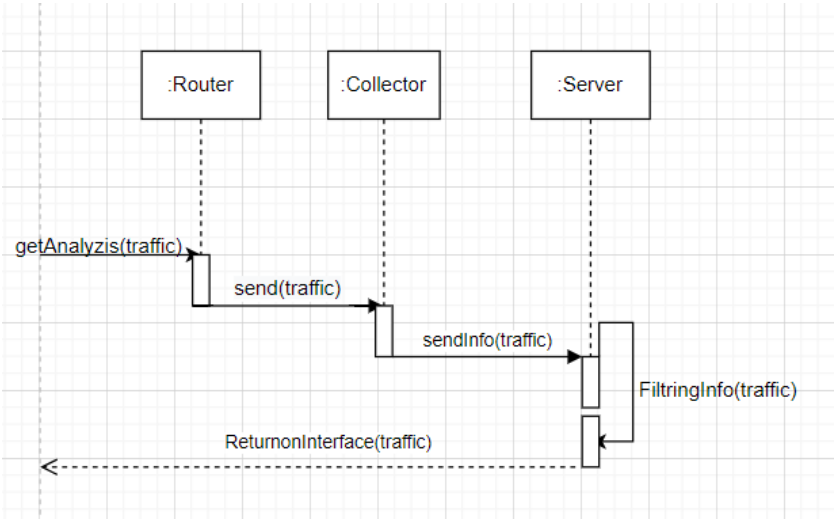

Рис. 2.7 Діаграма послідовності

2.1.17 Виявлення порушення безпеки мережі

У мережах більшість атак безпеки походять із самої мережі. З цієї причини ntop надає користувачам підтримку як для відстеження поточних атак, так і для виявлення потенційних прогалин у безпеці, зокрема:

- Виявлення сканування портів: класичне (надсилання пакета на кожен порт) і повільне (різновид сканування портів, де сканування портів відбувається дуже повільно, щоб ускладнити його виявлення) сканування портів (приховане сканування) можна легко виявити. Насправді ntop повідомляє назву останніх трьох хостів, які надіслали пакет на кожен порт, менший за 1024. Зараз ми додаємо підтримку негативного сканування портів (наприклад, сканування NULL і XMAS). Зауважте, що сканування портів виявляється не лише для хоста, на якому запущено ntop, але й для всіх хостів, для яких ntop може перехоплювати пакети (зазвичай всю підмережу). Це означає, що ntop забезпечує виявлення сканування портів

(під)мережі, тоді як дуже небагато ОС підтримують виявлення сканування портів, і то лише для хоста, на якому працює ОС [28].

- Виявлення підробки: Підробка відбувається, коли хост видає себе за інший хост з метою перехоплення пакетів. Загалом, для пакетів, які не походять із підмережі, де працює ntop, виявити підробку неможливо. Натомість підробку можна виявити принаймні для хостів, що належать до тієї самої підмережі хоста, де запущено ntop. Частина плагіна arpWatch ntop попереджає користувача, коли дві різні IP-адреси зіставляються з однією апаратною адресою. Зауважте, що виявлення підробки слід використовувати належним чином у мережах, де встановлено проксі-маршрутизатори ARP, або коли хост увімкнув підтримку декількох IP-адрес [29].

- Виявлення шпигуна: шпигун – це хост, чия мережева карта налаштована на безладний режим для захоплення пакетів незалежно від того, спрямовані вони на карту чи ні. Мережевий детектор Ethernet Promiscuous (neped) — це інструмент, який надсилає кожному хосту X локальної підмережі запит ARP, що містить IP-адресу X як цільову IP-адресу. На жаль, щойно описаний алгоритм працює не для всіх ОС; отже, нічого не можна сказати про хости, чиї NIC, очевидно, не налаштовані на безладний режим. Ntop періодично запускає neped і попереджає користувача про хости, чиї карти встановлені в безладному режимі [30].

- Виявлення троянських програм: зазвичай користувачі не виявляють троянських програм, таких як BO2K, доки не виникнуть серйозні проблеми. Оскільки такі додатки використовують добре відомі порти (наприклад, порт BO2K за замовчуванням — 3777), ntop може виявити їхню присутність, періодично перевіряючи, чи є якийсь мережевий трафік, що походить із призначених для цих портів.

- Відмова в обслуговуванні: Synflood — це здатність хоста надсилати пакети з установленим прапорцем SYN (прапорець SYN використовується для відкриття TCP-з’єднання) на відкриті порти цілі без подальшого встановлення з’єднання. Таким чином руйнівник заповнює всі слоти IP-стеку цілі, поки ціль не зможе приймати нові з’єднання. Хоча деякі ОС пропонують захист від synflood і, отже, на них ця проблема не впливає, ntop можна використовувати для виявлення руйнівників і повідомлення про проблему адміністратору мережі. Зауважте, що інші

види атак, зокрема smurf і fraggle, також виявляються шляхом аналізу трафіку, надісланого/отриманого кожним хостом [31].

Якщо виявлено порушення безпеки або неправильну конфігурацію/проблему мережі, ntop пропонує засоби для:

- Повідомлення про проблему адміністратору мережі.

- Розуміння того, де/як виникла атака, використовуючи інформацію про трафік, що зберігається в базі даних SQL.

- Виконання певних дій (якщо це можливо) для блокування атаки та, отже, обмеження її поширення на всю мережу. На даний момент ми розробляємо плагіни для сповіщення мережевих адміністраторів про проблему електронною поштою, перехопленнями SNMP або GSM SMS (система коротких повідомлень) за допомогою простих сценаріїв, які виконуються після виявлення добре відомої проблеми безпеки.

2.1.18 Оптимізація та планування мережі

Часто на продуктивність мережі впливає неоптимальна конфігурація деяких хостів і неефективне використання доступної пропусканої здатності. Зокрема, ntop дозволяє адміністраторам:

- Визначати непотрібні протоколи: іноді трафік генерується хостами/маршрутизаторами, які не налаштовані належним чином і намагаються з'єднатися з подібними собі за допомогою протоколів, які ніхто інший не використовує. Використовуючи ntop, ми часто ідентифікували трафік OSPF і протокол керування групами Інтернету (IGMP) у мережах, де ці протоколи не використовувалися. Крім того, такі протоколи, як IPX, коли використовуються лише одним хостом у мережі, генерують певний періодичний широкомовний трафік, який потім поширюється у всій підмережі.

- Визначення неоптимальної маршрутизації: плагін icmp-Watch ntop відповідає за обробку пакетів протоколу контрольних повідомлень Інтернету (ICMP). Можна ідентифікувати пристрої, які використовують неоптимальну маршрутизацію, просто відстежуючи повідомлення ICMP Redirect або періодично аналізуючи список маршрутизаторів підмережі.

- Характеристика та розподіл трафіку: Ntop дозволяє адміністраторам зрозуміти, як трафік розподіляється щодо протоколу та джерела (локальний чи віддалений трафік). Вивчення моделей трафіку допомагає адміністраторам зрозуміти, як мережа використовується як локально, так і з віддалених місць, і, отже, покращити, якщо це можливо, глобальну мережеву топологію та конфігурацію. Наприклад, використовуючи ntop, ми зрозуміли, що наш маршрутизатор мав визначити маршрут багатьом пакетам DNS просто тому, що DNS було розміщено не в найкращій підмережі.

Зменшення кількості протоколів, що використовуються: У деяких випадках два або більше протоколів використовуються з однією метою. Використання ntop дозволило нам побачити, що в нашій мережі кілька комп'ютерів Windows використовували як Net-BIOS, так і NetBIOS-over-IP, тоді як решта мережі використовувала лише NetBIOSover-IP. Змінивши конфігурацію мережі цих кількох хостів шляхом видалення Net-BIOS, кількість використовуваних протоколів було зменшено без втрати будь-якої існуючої функціональності [32].

- Розумніше використання пропускної здатності: пропускної здатності мережі ніколи не буває достатньо; отже, важливо намагатися уникати непотрібного спілкування. Вивчення того, як використовуються певні протоколи, допомагає адміністраторам визначити, куди додати програми, наприклад проксі-сервери, які дозволяють значно зменшити трафік шляхом кешування інформації.

В загальному, ntop поєднує функції, наявні в різних інструментах, які не завжди легко інтегрувати. Його унікальний інтерфейс користувача дозволяє адміністраторам негайно скористатися перевагами ntop без необхідності купувати та керувати клієнтськими програмами, необхідними для таких інструментів, як RMON або NeTraMet. Крім того, підтримка бази даних робить ntop придатним не лише для налагодження мережевих проблем, але й для тривалого моніторингу мережі та зворотного відстеження проблем [33].

2.1.19 Питання продуктивності

Продуктивність Ntop досить хороша, оскільки:

- продуктивність libpcap чудова.

- втрати пакетів (якщо такі є) дуже низькі, оскільки захоплені пакети буферизуються двічі, як всередині ядра, так і в ntop.

- потенційно тривалі дії (наприклад, визначення IP-адреси) реалізуються асинхронно.

- ntop породжує кілька потоків, які перешкоджають взаємодії користувача (наприклад, запити користувача HTTP) від перешкоджання збору даних.

- ntop широко використовує хеш-таблиці, індекси яких легко обчислити, але швидко під час пошуку інформації через природу мережевих адрес (наприклад, вони унікальні та вже мають 32/48-бітний числовий формат).

Користувачі широко тестували ntop на різних мережевих носіях на різних швидкостях. Загалом на продуктивність ntop значно впливають інші запущені процеси, оскільки деякі програми, що потребують ЦП, можуть займати всі цикли ЦП протягом кількох секунд, спричиняючи втрату пакетів. Якщо припустити, що ntop працює на середньо завантаженому хості, тести показали, що ntop може працювати з дуже низькими (якщо такі є) втратами пакетів на 100 Мб Ethernet.

2.1.20 Пропозиція щодо створення системи моніторингу в ієрархічних комп'ютерних мережах

Відповідно до всієї проаналізованої інформації ми створимо систему моніторингу трафіку, яка працюватиме в ієрархічних комп’ютерних мережах і аналізуватиме трафік за допомогою системи NetFlow та програмного забезпечення NTOP.

Основним завданням даної роботи є створення потужної системи аналізу трафіку в ієрархічній комп’ютерній мережі. Тому ми можемо періодично ловити та аналізувати мережеву netflow-статистику.

Очікуваний результат: за допомогою цього рішення можливо:

- збирати та аналізувати мережеву статистику на найпростіших пристроях та будувати графіки та таблиці;

- система відображає поточне навантаження на мережу та історію на графіках;

Структура реалізації: єдиний центр обміну даними, який є головним маршрутизатором, відправляє весь трафік мережі в один порт, потім з цього порту сервер отримує та аналізує дані. Вибір припав на ntop.

Я хотів би обійти цю проблему, обравши єдиний конкретний сценарій, який відразу підійде для 90% мережевих середовищ і який може працювати для більшості лише з незначними змінами. Припустимо, що:

- у нас є маршрутизатор Cisco, який підключений до колектора за допомогою, наприклад, порту 3000, і з ієрархічною мережею, яка моніториться. Моя демонстрація буде на Cisco 2620XM.

- у нас є хост, який виконує функції збору даних і сервера звітів. Це може бути той самий хост, що й ваш датчик, але збір даних і створення звітів збільшить навантаження на цю систему. Кваліфіковані зловмисники також зацікавлені в даних Netflow, тому вам слід розмістити колектор за брандмауером. Якщо у вас є веб-сервер, встановлений на колекторі, ви можете створювати гарні веб-звіти Netflow. Я використовую FreeBSD як систему збору та звітності, а також зміни назви пакетів відповідно до інших операційних систем.

2.2 Конструювання

У цьому розділі буде описано, як створити потужну ієрархічну систему комп'ютерного моніторингу. Спочатку ми повинні розробити основну модель ієрархічної комп'ютерної мережі. Для цього ми можемо використовувати програмне забезпечення Cisco Packet Tracer. Потім нам потрібно налаштувати систему моніторингу. Для цього можна використовувати таке програмне забезпечення:

- softflowd як датчик;
- flow-capture (з flow-tools) як колектор;
- flow-stat (з flow-tools) для аналізу в текстовому рядку;
- ntop для візуалізації та представлення в WEB.

Основними завданнями цього розділу є:

- моделювання ієрархічної комп'ютерної мережі;
- встановлення та налаштування датчика;
- встановлення та налаштування колектора;

- обробка та візуалізація даних NetFlow за допомогою програмного забезпечення ntop.

2.2.1 Створення основної топології

Для моделювання ієрархічної комп'ютерної мережі можна використовувати різне програмне забезпечення. Оскільки ми розглядаємо систему моніторингу NetFlow і ця система розроблена Cisco, ми можемо використовувати симулятор Cisco Packet Tracer. Packet Tracer — це потужна програма для моделювання мережі, яка дозволяє студентам експериментувати з поведінкою мережі та ставити запитання «що, якщо». Будучи невід'ємною частиною процесу навчання, Packet Tracer забезпечує симуляцію, візуалізацію, створення, оцінювання та можливості співпраці, а також полегшує викладання та вивчення складних технологічних концепцій. Packet Tracer доповнює фізичне обладнання в класі, дозволяючи студентам створювати мережу з майже необмеженою кількістю пристроїв, заохочуючи до практики, здійснення відкриттів та усунення несправностей. Навчальне середовище, засноване на моделюванні, допомагає учням розвивати навички 21 століття, такі як прийняття рішень, творче та критичне мислення, а також вирішення проблем. Для моделювання можна використати схему, яка зображена на рис. 2.8.

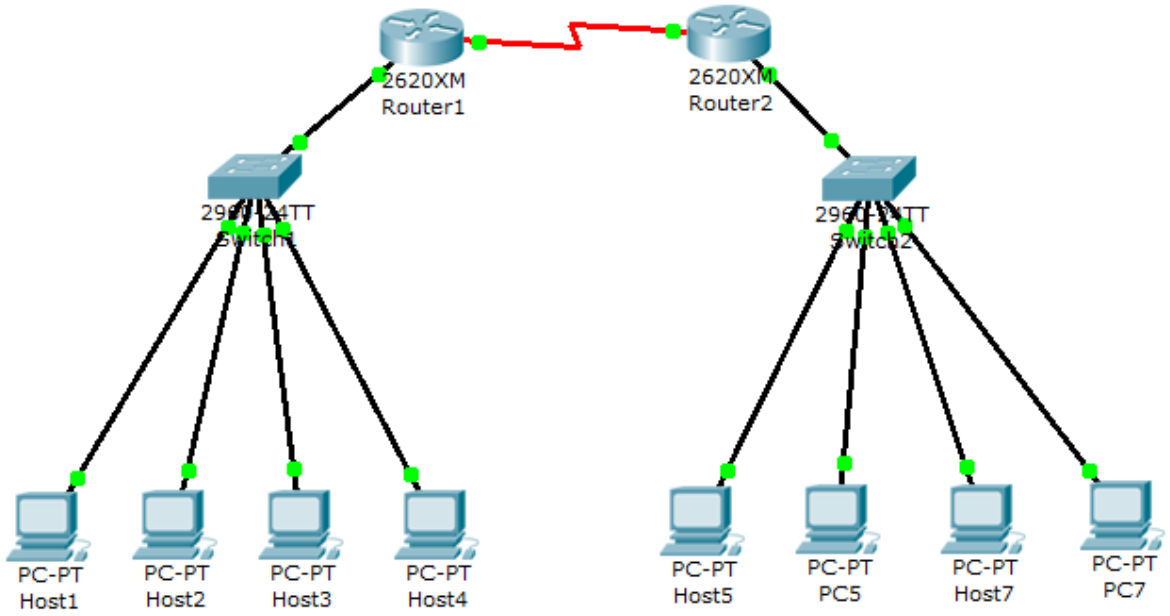

Рис. 2.8. Схема моделювання ієрархічної комп’ютерної мережі

2.2.2 Конфігурація хостів, маршрутизаторів, комутаторів

Для створення моделі мережі, яка зображена на рис. 3.1. нам потрібні деякі пристрої. Це:

- два маршрутизатори Cisco 2620XM з модулем NM-4A/S;
- два перемикачі 2960-24TT;
- 8 хостів (комп'ютерів).

Для налаштування хостів ми можемо використовувати таку конфігурацію:

Хост1: ip-адреса 192.168.0.1 маска 255.255.255.0 шлюз 192.168.0.100

Хост2: ip-адреса 192.168.0.2 маска 255.255.255.0 шлюз 192.168.0.100

Хост3: ip-адреса 192.168.0.3 маска 255.255.255.0 шлюз 192.168.0.100

Хост4: ip-адреса 192.168.0.4 маска 255.255.255.0 шлюз 192.168.0.100

Хост5: ip-адреса 192.168.1.1 маска 255.255.255.0 шлюз 192.168.1.100

Хост6: ip-адреса 192.168.1.2 маска 255.255.255.0 шлюз 192.168.1.100

Хост7: ip-адреса 192.168.1.3 маска 255.255.255.0 шлюз 192.168.1.100

Хост8: ip-адреса 192.168.1.4 маска 255.255.255.0 шлюз 192.168.1.100

Для налаштування маршрутизаторів, які використовують маршрутизацію RIP, ми можемо використовувати таку конфігурацію:

Маршрутизатор1: FastEthernet 0/0: ip-адреса 192.168.0.100 255.255.255.0

Serial1/0: ip-адреса 10.0.0.1 255.0.0.0

Маршрутизатор 2: FastEthernet 0/0: ip-адреса 192.168.1.100 255.255.255.0

Serial1/0: ip-адреса 10.0.0.2 255.0.0.0

Конфігурація Cisco IOS на маршрутизаторі 1 з протоколом маршрутизації RIP і конфігурація інтерфейсів наведені в Додатку А.

Конфігурація Cisco IOS на маршрутизаторі 2 з протоколом маршрутизації RIP і конфігурація інтерфейсів наведені в Додатку А.

Маршрутизатори з'єднані між собою за допомогою послідовного кабелю DTE. З'єднання між комутаторами та хостами, а також маршрутизаторами та комутаторами здійснюється за допомогою мідних прямих кабелів. Перемикачі без конфігурації.

2.2.3. Встановлення і налаштування датчика

Для створення ієрархічної системи моніторингу комп'ютерної мережі будемо використовувати операційну систему FreeBSD. Як програмне забезпечення для датчика ми можемо використовувати softflowd.

Для створення системи буде використано таке програмне забезпечення:

softflowd як датчик;

flow-capture (з flow-tools) як колектор;

flow-stat (з flow-tools) для аналізу в текстовому рядку;

ntop для візуалізації та представлення в WEB. (Рис. 2.7).

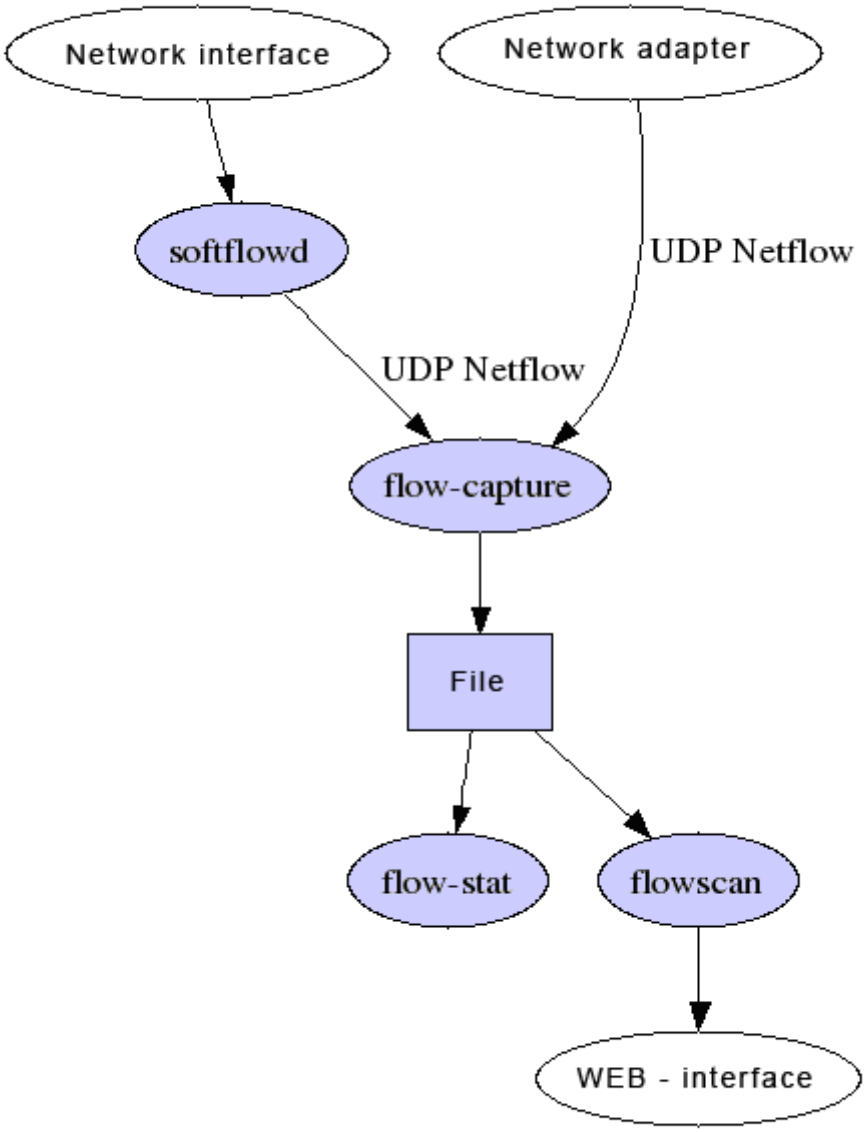

Рис. 2.9. Схема ієрархічної системи моніторингу комп'ютерної мережі.

2.2.4 Встановлення датчика і запуск програми

Найпростішим програмним забезпеченням для встановлення є датчик. По-перше, переконайтеся, що обладнання нашого датчика може прослуховувати весь мережевий трафік. Якщо це так, ми зможемо встановити програмне забезпечення датчика. Я рекомендую softflowd. Sortflowd запускається нестандартно за допомогою простого виконання інсталяції на BSD і Linux, вимагаючи лише libpcap.

(FreeBSD містить систему ng_netflow netflow, але оскільки я також маю датчики, що працюють на Linux, я не використовую її. Я віддаю перевагу використанню окремого програмного забезпечення в кожній операційній системі, коли це можливо.) Після встановлення softflowd нам потрібна лише назва інтерфейсу, який ми хочемо контролювати, та IP-адреса і порт UDP, де прослуховує наш колектор. Наприклад, щоб прослухати інтерфейс em0 і надіслати зібрані дані на 172.16.13.5, порт 3000, виконайте:

softflowd -i em0 -n 172.16.13.5:3000

Датчик негайно почне прослуховувати мережу та надсилати інформацію про сесію в колектор. Переконайтеся, що ця програма запускається під час завантаження.

2.2.5 Перевірка і завантаження скрипта

Softflowd містить програму управління softflowctl, яка дозволяє надавати команди запущеному softflowd. Щоб переконатися, що програмне забезпечення справді працює, перевірте статистику softflow після того, як softflow попрацює кілька хвилин.

статистика softflowctl
softflowd[40475]: Накопичена статистика:
Кількість активних потоків: 2298
Оброблено пакетів: 268086
Фрагменти: 0
Проігноровані пакети: 867 (867 не-IP, 0 закороткий)
Термін дії потоків минув: 3103 (0 вимушених)
Експортовано потоків: 6206 у 214 пакетах (0 помилок)

Важливим виходом тут є другий рядок, який повідомляє нам, скільки потоків активні на даний момент, і експортований рядок, який повідомляє нам, скільки потоків softflowd експортував до колектора.

Маршрутизатори Cisco можуть експортувати дані Netflow за рахунок дорогоцінного процесорного часу маршрутизатора. Якщо ми маємо складне налаштування маршрутизатора або якщо у нас маршрутизатор дуже низького рівня,

це може перевантажити наш маршрутизатор. Cisco була б дуже рада продати нам оновлення маршрутизатора, щоб ми могли належним чином експортувати Netflow, але загалом Unix-подібний блок є економічно ефективнішим. Багато інших пристроїв також підтримують Netflow.

Якщо у нас встановлено Ethereal або tcpdump, я рекомендую використовувати їх на цьому етапі, щоб підтвердити, що дані Netflow дійсно надходять у колектор. Якщо ні, підтвердьте, що softflowd запущено, і, можливо, спробуйте прапорець -D (налагодження), щоб побачити, чи є якісь проблеми з нашим налаштуванням.

Softflowd надсилає інформацію про потік до збирача лише після завершення потоку, наприклад, коли завершується сеанс FTP, коли веб-сторінку було доставлено тощо. Це означає, що в будь-який момент softflowd матиме кеш поточних підключень. Коли ми зупиняємо softflowd, запустіть softflowctl shutdown, щоб softflowd закінчив термін дії цих потоків і негайно надіслав їх до колектора. Просте вимкнення сервера, на якому працює softflowd, призведе до втрати активних, але неповних потоків. Ми все одно втратимо деяку інформацію, якщо перезавантажимо датчик, але ми можемо зробити цю втрату якомога меншою.

Для автоматичного softflow-завантаження ми повинні створити сценарій завантаження та додати його до системи. Сценарій потрібен, оскільки softflowd не може запускатися автоматично. Сценарій наведено в Додатку В.

Після створення сценарію нам потрібно:

- переконатися, що він знаходиться в каталозі /usr/local/etc/rc.d;

- перевірити, чи має розширення .sh;

- перевірити, що його можна виконати за допомогою команди:

chmod +x /usr/local/etc/rc.d/softflowd.sh.

Для запуску скрипта нам потрібно ввести такі змінні у файл /etc/rc.conf:

- softflowd_enable – запускати softflowd при завантаженні системи;

- softflowd_interfaces – які інтерфейси слухати;

- softflowd_netflow_host - IP-адреса колектора NetFlow;

- softflowd_netflow_port – порт, на якому слухає процес-колектор NetFlow.

Запуск і зупинка скрипта виконується такими командами:

/usr/local/etc/rc.d/softflowd.sh запустити

/usr/local/etc/rc.d/softflowd.sh зупинити

2.2.6 Встановлення колектора і запуск програми

Наш колектор збирає дані, експортовані датчиком, і зберігає їх на диску для довгострокового використання. Якщо можливо, інсталюйте колектор на веб-сервері; це зробить звітування набагато приємнішим і легшим. Я рекомендую flow-capture, дуже популярний колектор Netflow, який входить до пакету flow-tools. У FreeBSD інструменти flow-tools знаходяться в Ports tree за адресою /usr/ports/net-mgmt/flow-tools. Встановіть його за допомогою "встановити все". Не друкуйте "очистити". Можливо, нам доведеться відновлювати деякі компоненти вручну. З тієї ж причини не використовуйте попередньо скомпільований пакет інструментів flow-tools.

Створіть каталог для flow-capture, щоб зберігати його записи. Зазвичай я використовую /var/log/netflow, але скрізь, де є простір, це працює. У багатомегабітній мережі файли Netflow можуть заповнити кілька ГБ диска протягом кількох тижнів. Я також рекомендую створити збережений підкаталог у нашому каталозі журналу для використання системою звітності.

Тепер нам потрібен сценарій запуску, щоб flow-capture запускалося автоматично під час завантаження. Добре працює команда на кшталт наступної:

```
# /usr/local/bin/flow-capture -p /var/run/flow-capture.pid -n 287 \
-N 0 -w /var/log/netflows/ -S 5 0/0/3000
```

Більшу частину цього ми можемо використовувати без змін. Прапорець -w вказує flow-capture, куди розміщувати файли. Останній аргумент вказує flow-capture, який локальний IP прослуховувати, який віддалений IP прослуховувати та який порт UDP слід прослуховувати. У цьому випадку, 0/0/3000, колектор прослуховує всі локальні IP-адреси, для запитів з будь-якої віддаленої IP-адреси, на порту 3000. Якщо ми можемо отримувати випадковий Інтернет-трафік на колекторі, укажіть IP-адресу конкретного датчика в середньому значенні. (Мій колектор знаходиться за брандмауером, і будь-хто, хто може пройти повз брандмауер, також не матиме проблем з оманою flow-capture.) Flow-capture потребує аргументів -n 287, -N і -S 5 для взаємодії зі звітним пакетом, тому залиште їх у спокої.

Щойно ми запустимо flow-capture, файли потоку з’являться в каталозі журналу. Назви цих файлів походять від версії даних Netflow, які вони збирають, а також від дати й часу початку цих даних. Наприклад, назва файлу tmp-v05.2010-05-25.201001-0400 вказує на тимчасовий файл, що містить дані Netflow версії 5, зібрані 25 травня 2022 року, починаючи з 20:10:01. Кожні п’ять хвилин flow-capture переміщує тимчасовий файл у постійне розташування та створює новий тимчасовий файл. Постійні файли починаються з ft замість tmp, але в загальному імена точно такі ж.

Щоб підтвердити, що встановлення flow-capture справді щось збирає, подивіться, чи збільшується тимчасовий файл. Це має статися швидко, протягом кількох хвилин у завантаженій мережі.

2.2.7. Перевірка і завантаження скрипта

Щоб перевірити, чи збираються дані, заблокуйте нові файли в каталозі netflow і заблокуйте розмір тимчасового файлу. Він повинен збільшитися в розмірі.

Що стосується softflowd, для автоматичного завантаження колектора ми повинні створити сценарій завантаження та додати його до системи. Сценарій потрібен, оскільки перехоплення потоку не може запускатися автоматично. Сценарій наведено в Додатку Г.

Після створення скрипта нам потрібно:

- переконатися, що він знаходиться в каталозі /usr/local/etc/rc.d;

- перевірити, чи має розширення .sh;

- перевірити, що його можна виконати за допомогою команди:

chmod +x /usr/local/etc/rc.d/flowcapture.sh

Для запуску скрипта нам потрібно ввести такі змінні у файл /etc/rc.conf:

- flowcapture_enable – запустити flow-capture при завантаженні системи;

- flowcapture_port — порт, на якому NetFlow колектор повинен слухати (опціонально 3000);

- flowcapture_dir – назва каталогу, в якому збираються файли даних колектора (опціонально/var/netflow);

- flowcapture_flags – введіть прапорці для запуску flow-capture.

Запуск і зупинка скрипта виконується такими командами:

/usr/local/etc/rc.d/flowcapture.sh запуск

/usr/local/etc/rc.d/flowcapture.sh зупинка

2.2.8. Встановлення модуля Cflow.pm

Багато різних інструментів звітування Netflow використовують модуль Cflow.pm perl для читання файлів Netflow. Це включає бібліотеку та інструмент командного рядка для перегляду файлів потоку та керування ними. Найважче те, що кожен колектор має власний формат зберігання. Хоча початковою метою Cflow.pm була обробка файлів cflowd, Cflow.pm може підтримувати інші формати, якщо його правильно встановлено.

У цій частині більшість людей відмовляються від Netflow. Ретельно дотримуйтесь інструкцій. Не забудьте перевірити роботу після завершення встановлення Cflow.pm.

У деяких останніх версіях FreeBSD /usr/ports/net-mgmt/p5-Cflow автоматично визначає наявність бібліотек flow-tools. Cflow зв’язує цю бібліотеку як -lnsl, і якщо процес збирання не знайде її під час процесу налаштування, ми побачимо таке попередження:

Примітка (ймовірно, нешкідлива): бібліотеки для -lnsl не знайдено

Це попередження не є нешкідливим; це означає, що цей Cflow не працюватиме. Якщо ми не бачимо цього рядка, просто встановіть Cflow і перевірте, чи він працює. Cflow містить flowdumper(1), програму для читання файлів потоку в командному рядку. Перевірте найбільший файл потоку, який у нас є, щоб ми могли бути впевнені, що запис містить щось для перегляду.

#flowdumper -s ft-v05.2005-04-28.201501-0400 | більше

2005/04/28 19:14:01 172.16.30.247.80 -> 216.98.200.250.63647 6(SYN|ACK) 3 144

2005/04/28 19:14:01 216.98.200.250.63647 -> 172.16.30.247.80 6(SYN) 1 48

2005/04/28 19:14:01 172.16.30.247.80 -> 216.98.200.250.63648 6(SYN|ACK) 3 144

2005/04/28 19:14:01 216.98.200.250.63648 -> 172.16.30.247.80 6(SYN) 1 48

Кожен рядок є потоком. Це записує IP-адреси джерела та призначення різноманітних транзакцій TCP/IP. Ми можемо помітити, що цей конкретний фрагмент із чотирьох рядків насправді є лише двома сеансами TCP/IP. Перший рядок вказує, що трафік надходить з 172.16.30.247, порт 80, на хост 216.98.200.25. У наступному рядку показано трафік від другого хоста до першого.

Якщо інсталяція Cflow невдається, flowdumper покаже мовчання або помилку. Ми не можемо продовжити, доки принаймні не вирішимо цю помилку; ми не можемо продовжити, якщо хочемо, щоб наші інструменти звітування працювали. Видаліть поточний пакет p5-Cflow і створіть його іншим способом.

Пам’ятайте не очищати порт інструментів Flow. Поверніться до каталогу портів, cd до робочого підкаталогу і перейдіть до каталогу вихідного коду. У підкаталозі під назвою contrib є інший архів Cflow. Видобудьте його.

cd /usr/ports/net-mgmt/flow-tools/work/flow-tools-0.67/contrib

tar -xzvf Cflow-1.051.tar.gz

Cflow часто підбирає відповідну бібліотеку, коли встановлюється з цього місця під скомпільованим пакетом flow-tools. Це означає, що ми повинні мати вбудовані інструменти потоку в каталозі вище; ось чому я сказав не робити очищення. Просто дотримуйтеся звичайного процесу створення модуля Perl.

perl Makefile.PL

make

make install

Спробуйте flowdumper знову, і він повинен працювати.

Іноді у мене була навіть ця помилка. У такому випадку використовуйте грубу силу. Flow-tools встановлює libft.a під /usr/local/lib. Відредагуйте розділ Cflow.pm's Makefile.PL, де він перевіряє бібліотеку flow-tools (див. Додаток Д).

Якщо це не вдається, щось серйозно не так із встановленням Perl. Тепер запустіть make та make install. Зараз у нас є flowdumper з підтримкою flow-tools, який вказує на те, що модуль Cflow.pm Perl, який лежить в його основі, правильно працює з колектором.

Ймовірно, ми можемо легко уявити собі цілу низку сценаріїв Perl, які брали б ці дані та створювали гарні графіки і звіти про використання або визначали споживачів максимальної пропускної здатності. Однак інші люди вже зробили

важку роботу над цим. У моїй наступній статті ми розглянемо створення гарних зображень із даних Netflow.

2.2.9 Встановлення та налаштування ntop

Для візуалізації трафіку NetFlow ми можемо використовувати програмне забезпечення ntop. Щоб встановити його, ми можемо скористатися офіційним сайтом. Потім використовуйте традиційно «make && make install». Після цього ми повинні ініціалізувати програму.

Ініціалізація ntop:

ntop : Це ініціалізує ntop і попросить ввести ім’я користувача та пароль.

Ім'я користувача за замовчуванням: admin

Будь ласка, введіть пароль користувача адміністратора:

Введіть пароль ще раз:

Після встановлення пароля адміністратора ми отримаємо повідомлення в командному рядку приблизно на кшталт

"Четвер, 24 травня 2022 р., 23:05:10 Пароль користувача адміністратора встановлено"

Запустіть службу ntop:

service ntop start

2.2.10 Перевірка візуалізаційного програмного забезпечення

ntop можна керувати через веб-інтерфейс. Ми можемо ввести адресу сервера у веб-браузері:

http://ServerIP:3000 або https://ServerIP:3001.

Тепер ми можемо контролювати хости та керувати конфігурацією ntop за допомогою логіну адміністратора.

Плагіни

Наступні плагіни можна налаштувати для системи через веб-інтерфейс ntop:

- Host Last Seen: цей плагін створює звіт про час останнього перегляду пакетів з кожного конкретного хоста. Для запису додаткової інформації доступна база даних карток.

- icmpWatch: цей плагін створює звіт про пакети ICMP, які бачив ntop. Звіт містить рахунок кожного хоста, байта та кожного типу (відправлено/отримано).

- snmpPlugin: цей плагін використовується для моніторингу трафіку хоста за допомогою протоколу SNMP.

- Round Robin Database: цей плагін використовується для налаштування, активації та деактивації підтримки rrd ntop. Цей плагін також створює графіки даних rrd, доступних через посилання з різних звітів «Інформація про хост xxxxx».

- NetFlow: цей плагін використовується для налаштування, активації та деактивації підтримки NetFlow. ntop може як збирати, так і отримувати дані NetFlow V1/V5/V7/V9 і IPFIX (чернетки). Отримані дані потоку повідомляються як окремий «NIC» у звичайних звітах ntop.

- sFlow: цей плагін використовується для налаштування, активації та деактивації підтримки sFlow ntop. ntop може як збирати, так і отримувати дані sFlow.

- PDAPlugin: цей плагін створює мінімальний звіт ntop, придатний для відображення на pda.

2.3 Результати експерименту системи моніторингу трафіку в ієрархічних комп’ютерних мережах

Для тестування створеної ієрархічної комп’ютерної мережі можна використовувати пакети ICMP (ping) з мережі 192.168.0.0/24 на 192.168.1.0/24. Якщо пакети будуть проходити без втрат, то ієрархічна мережа працює нормально (Рис. 2.10.).

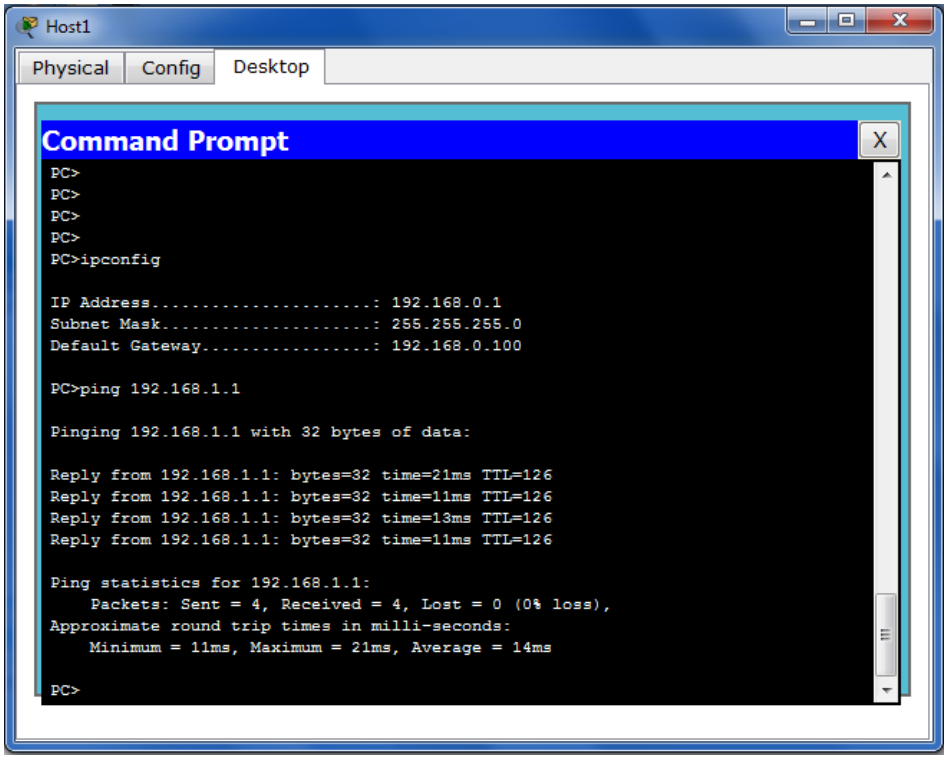

Рис. 2.10. Надсилання пакетів ICMP з Host1 (мережа 192.168.0.0) на Host2 (мережа 192.168.1.0).

Для тестування ієрархічної системи моніторингу комп’ютерної мережі можна використовувати програмне забезпечення ntop. ntop можна керувати через веб-інтерфейс. Ми можемо ввести адресу сервера у веб-браузері:

http://IP сервера:3000

https://ServerIP:3001

Тепер ми можемо контролювати ваші хости та керувати вашою конфігурацією ntop, вашим обліковим записом адміністратора. Наступний знімок екрана ntop на Рис. 2.11. показує частину «Глобальної статистики трафіку» для пристрою NetFlow.

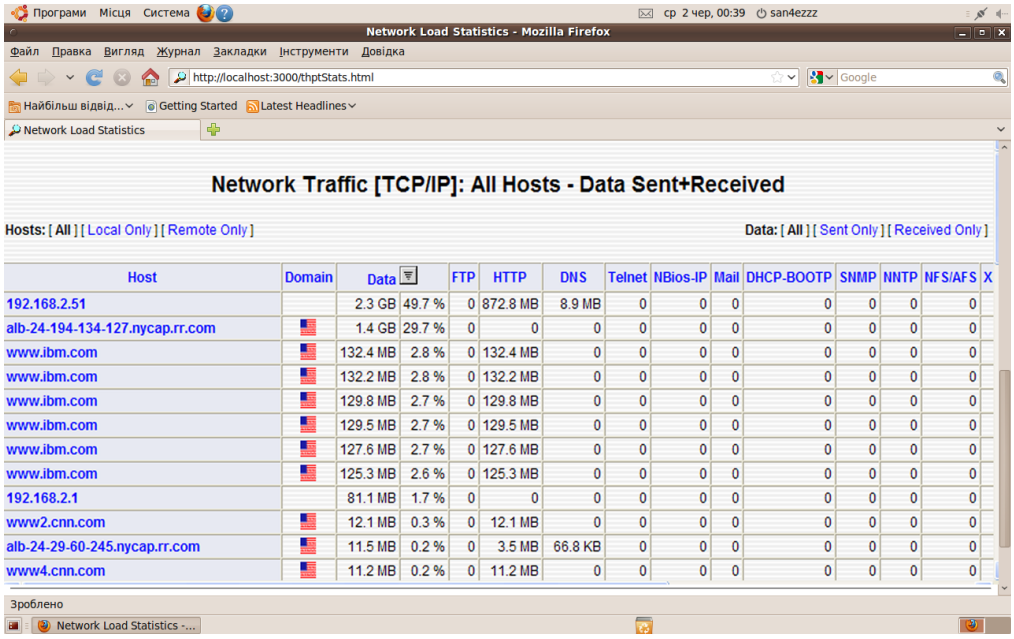

Рисунок 2.11. Статистика глобального трафіку Ntop NetFlow

Екран ntop, рис. 2.12., показує фактичний трафік NetFlow, отриманий із датчика NetFlow:

Рисунок 2.12. Трафік Ntop NetFlow

На наступних скріншотах ми бачимо результат таких функцій:

- інформацію про хост (Рис. 2.13);

- статистика завантаження мережі (Рис. 2.14);
- пропускна здатність мережі на всіх хостах (Рис. 2.15);
- мережева активність на локальних хостах (Рис. 2.16);
- мережевий трафік між локальними хостами (Рис. 2.17);
- мережевий трафік на всіх хостах (Рис. 2.18).

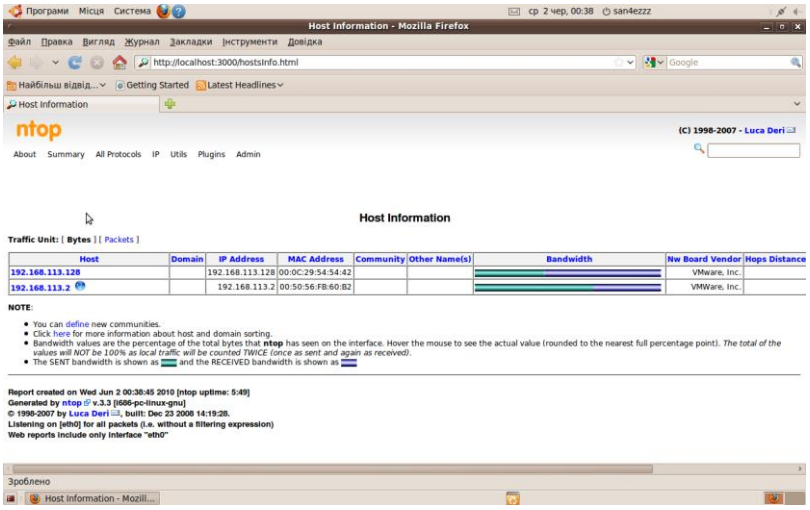

Рис. 2.13. Інформація про хост

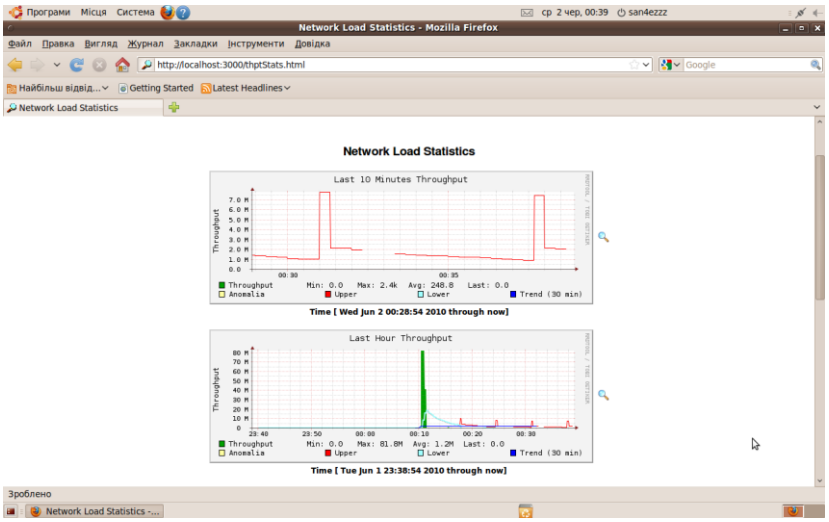

Рис. 2.14. Статистика завантаження мережі

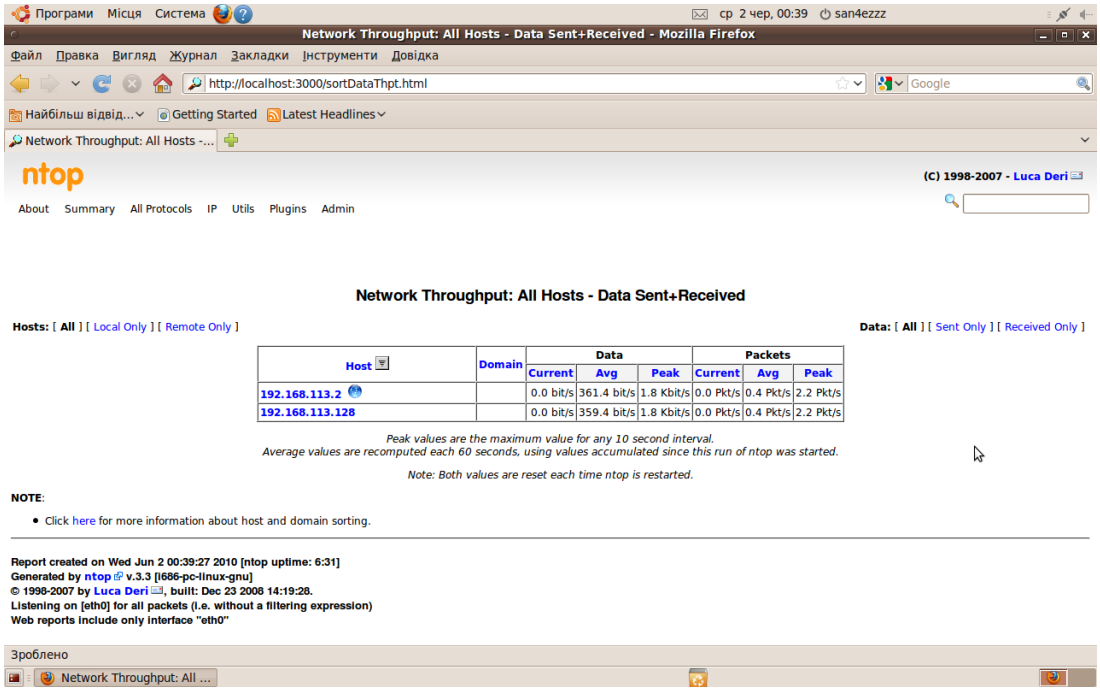

Рис. 2.15. Пропускна здатність мережі на всіх хостах

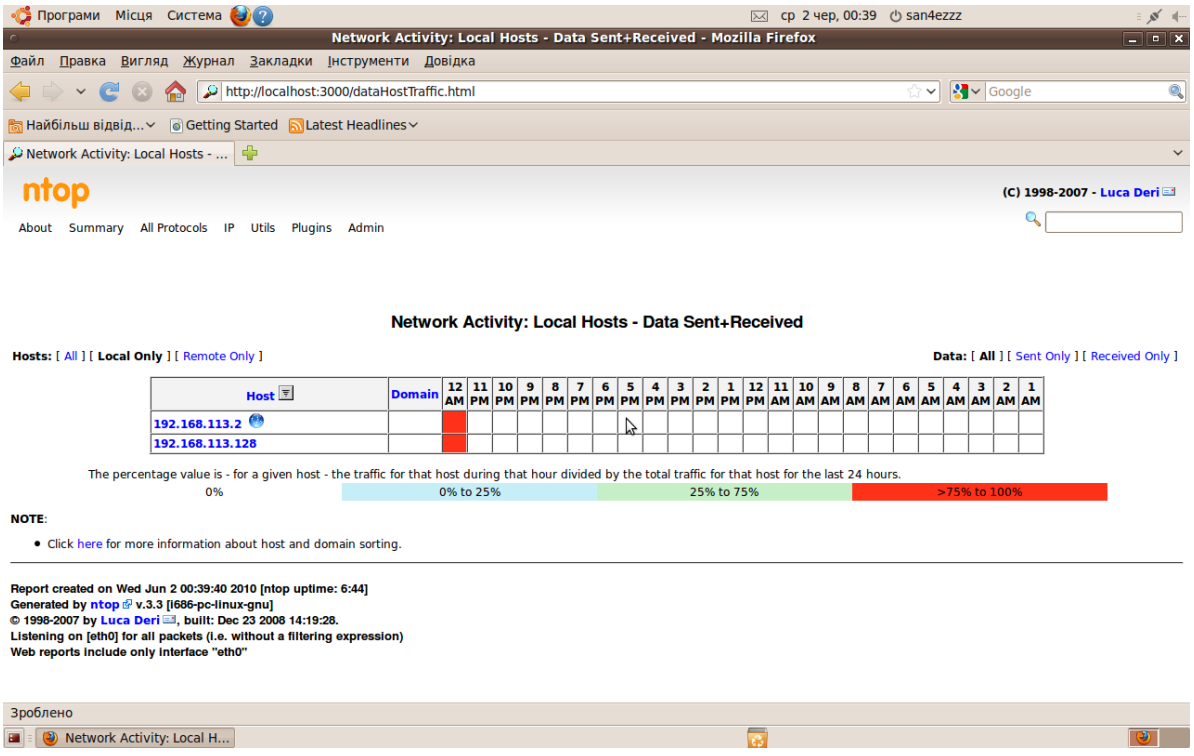

Рис. 2.16. Мережева активність на локальних хостах

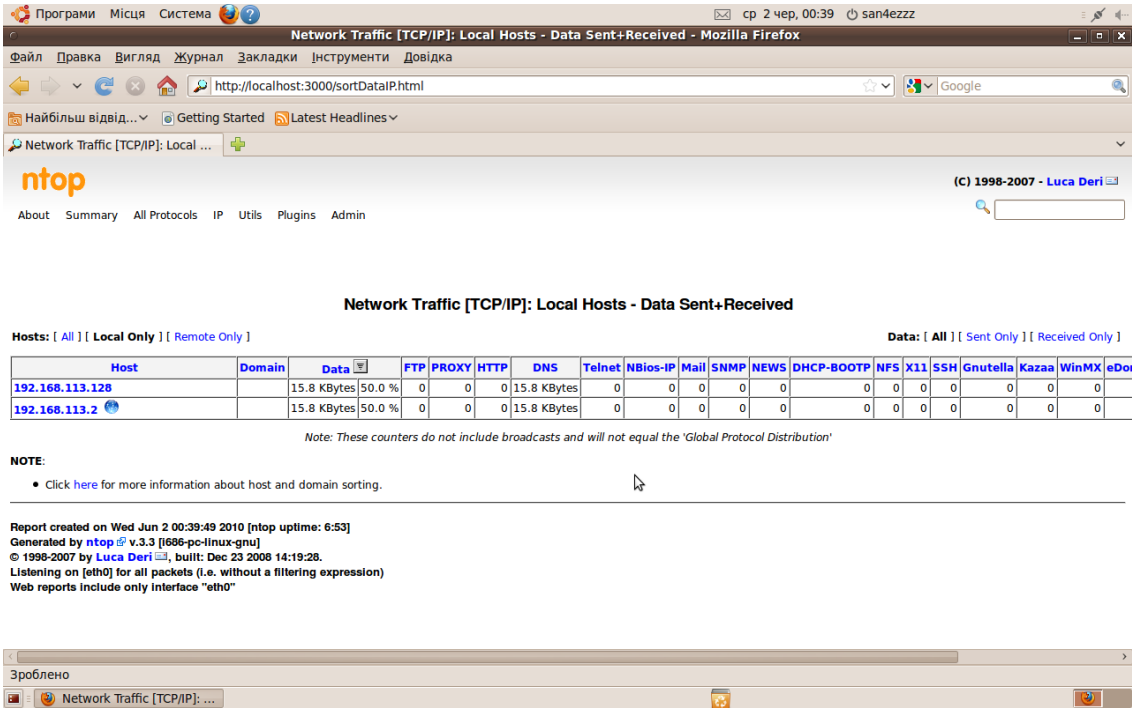

Рис. 2.17. Мережевий трафік між локальними хостами

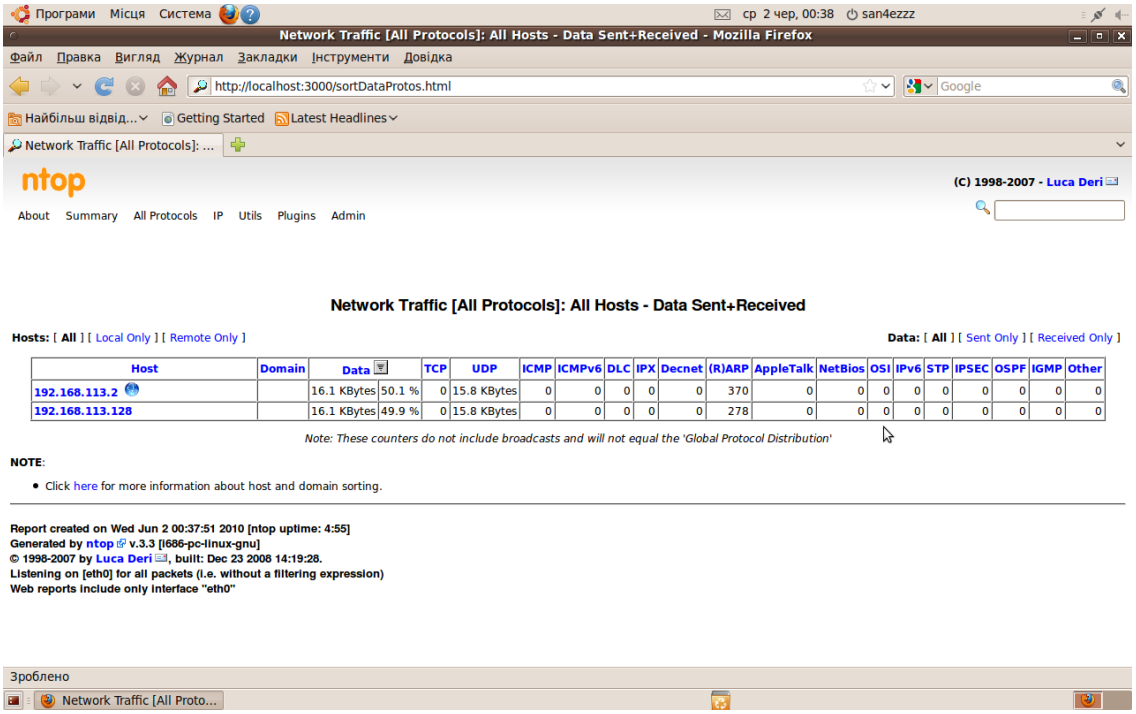

Рис. 2.18. Мережевий трафік на всіх хостах

2.4 Висновки до розділу 2

У цьому розділі ми розглянули проектування та конструювання системи моніторингу трафіку в ієрархічних комп’ютерних мережах. У розділі 2.3 ми маємо результати роботи системи.

У першому розділі було розглянуто основні завдання та мету другого розділу.

У третьому розділі було розглянуто тестування маршрутизації в ієрархічній комп’ютерній мережі з використанням пакетів ICMP. Пакети проходять, тому ми маємо стабільну роботу мережі. Потім ми зробили тест WEB-інтерфейсу, який показав статистику трафіку NetFlow.

3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

3.1 Охорона праці

Система моніторингу трафіку в ієрархічних мережах, як це вже зрозуміло з назви, буде використовувати комп'ютерні мережі та сервера, а отже нам треба відповідально віднестися до норм та правил безпеки їхнього використання. Тепер розглянемо цю тему більш детальніше.

Перелік нормативно-правових актів, що так чи інакше регулюють дане питання, є досить широким. Так, обов'язки роботодавця щодо забезпечення працівникам комфортних та безпечних умов для здійснення роботи, а також права працівників на такі умови передбачено частиною 2 ст. 2 та ч. 1 ст. 21 КЗпП, а також ст. 13 Закону України «Про охорону праці». Даний закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносин між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні.

Більшість актів у даній сфері становлять акти підзаконного рівня, а саме, численні правила, інструкції, державні санітарні правила і норми (ДСанПІН) тощо, якими врегульовуються окремі моменти щодо власне охорони праці в офісах, великих та малих підприємствах, та інших будівель які використовують комп'ютерну мережу, особливостей облаштування приміщень для роботи з нею та низки інших подібних вимог.

На сьогодні до основних підзаконних актів у даній сфері можна віднести:

— НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров`я працівників під час роботи з екранними пристроями

Вимоги поширюються на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми і видів діяльності та встановлюють мінімальні вимоги безпеки та захисту здоров'я під час здійснення роботи, пов'язаної з використанням екранних пристроїв незалежно від їхнього типу та моделі. Якщо аналізувати норми, прописані у даних документах, і спробувати

викласти їх більш-менш лаконічно й водночас вичерпною мовою зрозумілою більшості читачів, то їх можна звести до такого:

Під час облаштування робочого місця працівника з екранними пристроями необхідно обирати таке устаткування, яке не створює зайвого шуму та не виділяє надлишкового тепла. Рівні шуму на робочих місцях осіб, які працюють з екранними пристроями, мають відповідати вимогам Санітарних норм виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99.

Роботодавець зобов'язаний за необхідності проводити лабораторні дослідження умов праці працівників з метою виявлення шкідливих і небезпечних факторів виробничого середовища, важкості та напруженості трудового процесу (зокрема щодо виявлення ризиків, пов'язаних із погіршенням зору, порушенням фізичного стану, стресом) та вживати заходів щодо усунення виявлених ризиків відповідно до статті 13 Закону України „Про охорону праці".

Під час виконання робіт операторського типу, пов'язаних з нервово-емоційним напруженням, у приміщеннях під час роботи з екранними пристроями, на пультах і постах керування технологічними процесами та в інших приміщеннях мають дотримуватися оптимальні умови мікроклімату відповідно до вимог ДСН 3.3.6.042-99.

Вимоги щодо організації та обладнання робочих місць:

- Робочі місця працівників з екранними пристроями мають бути спроектовані так і мати такі розміри, щоб працівники мали простір для зміни робочого положення та рухів.
- Освітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном і навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98.
- Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.

У випадку виникнення аварійної ситуації оператор зобов'язаний:

- у всіх випадках виявлення пошкодження проводів електричного живлення, несправності заземлення та інших пошкодженнях електрообладнання, виникненні запаху гарі, диму - негайно вимкнути електричне живлення і повідомити про аварійну ситуацію свого безпосереднього керівника й чергового електрика;
- при попаданні людини під електричну напругу негайно звільнити її від дії струму шляхом вимкнення електричного живлення, до прибуття лікаря надати потерпілому долікарську медичну допомогу;
- при будь-яких випадках порушень роботи технічного обладнання або програмного забезпечення негайно викликати представника технічної служби з питань експлуатації обчислювальної техніки;
- у випадку виникнення різкого погіршення зору, виникнення головного болю, больових відчуттів у пальцях та кистях рук, посилення серцебиття - негайно припинити роботу з використанням ЕОМ, повідомити про те, що сталося, свого безпосереднього керівника й звернутися до медичної установи

3.2 Безпека в надзвичайних ситуаціях

Підвищення стійкості роботи підприємств приладо-будівної галузі, у воєнний час – одна із основних задач цивільної оборони України. Могутність країни базується на стійкій економіці. В сучасних умовах, коли науково-технічний прогрес у всіх сферах виробництва досяг небачених масштабів і привів до створення зброї масового ураження, в разі розгортання великомасштабної війни основні промислові центри і райони будуть головною ціллю для знищення зі сторони противника. Адже виведення економіки з ладу може призвести до того, що країна не зможе стояти на оборонні своїх кордонів та підтримувати життєдіяльність населення. На сьогодні, через бойові дії на сході України, проблема підвищення стійкості роботи підприємств приладо-будівної галузі стоїть як ніколи гостро.

Приладобудування - галузь машинобудування, що випускає засоби виміру, аналізу, обробки і представлення інформації, пристрої регулювання, автоматичні і автоматизовані системи управління; галузь науки і техніки, розробляюча засоби автоматизації і системи управління. Діяльність приладо-будівних підприємств забезпечується наявністю в їх розпорядженні необхідних ресурсів: людських, фінансових, матеріальних, енергетичних, за допомогою яких створюється продукція. Провідне місце в приладобудуванні за кількістю і різноманітністю приладів, що випускаються, займають засоби вимірювальної техніки. Створені методи і прилади виміру механічних, електричних, магнітних, теплових, оптичних, радіаційних і ін. величин. Вимірювальні прилади у поєднанні з регулюючими, обчислювальними і старанними пристроями складають технічну базу автоматизованих систем управління технологічними процесами (АСУТП).

Значне місце в приладобудуванні займає розробка і виробництво засобів випробувальної техніки. Прилади і машини випробування матеріалів і конструкцій на міцність для металургії машинобудування, індустрії будівельних матеріалів, гумотехнічної, легкої і інших галузей промисловості випускаються Івановським заводом випробувальних приладів, Армавірським заводом випробувальних машин і ін. підприємствами. На їх основі створюються автоматизовані, універсальні випробувальні установки, станції, полігони.

Сучасне приладобудування покликане забезпечувати народне господарство ефективними засобами і системами управління на основі широкого використання досягнень науки. Вивчаються процеси управління різними виробництвами, постачанням ресурсами, обслуговуванням адміністративно-господарською діяльністю, виявляються оптимальні вимоги до систем і засобів, визначаються економічно і технічно доцільні дороги їх реалізації, розробляються типові вирішення конкретних завдань управління при мінімізації номенклатури виробів приладобудування. Важливе значення має підвищення інформативності систем при одночасному скороченні кількості приватних відомостей, що представляються людині, що досягається за рахунок розширення аналітичній функції вимірювальних і обчислювальних пристроїв. Істотне підвищення автоматичності управління. Дослідження процесів документообігу в умовах дії АСОВІ дозволяє спростити і уніфікувати документообіг, вивільнити персонал від непродуктивної роботи, передаючи формування інформації відповідним пристроям. Дослідження технологічних процесів, різних режимів роботи устаткування і машин дає можливість ширше використовувати методи адаптації систем управління для здобуття найкращих техніко-економічних показників.

Під стійкістю роботи підприємств приладо-будівної галузі розуміють їх здатність за умов дії надзвичайних ситуацій виробляти продукцію в запланованих обсязі та номенклатурі, а при одержанні слабких чи середніх руйнувань чи порушенні постачання сировини відновлювати своє виробництво в мінімально короткі терміни. Щоб забезпечити нормальну роботу під час війни промислових об’єктів виробництва, скоротити можливі матеріальні втрати, необхідно ще в мирний час виконати великий комплекс різних заходів, які забезпечили б їхнє функціонування. Ці заходи спрямовані на зниження можливих втрат і руйнувань від сучасних засобів ураження і створення умов для нормальної роботи підприємств як у воєнний, так і в мирний час.

На стійкість роботи об’єктів виробництва впливають такі фактори:

- надійність захисту робітників від дії вражаючих факторів, що виникають під час надзвичайних ситуацій;

- здатність виробничого комплексу протистояти дії вражаючих факторів;
- надійність систем постачання об’єкта сировинною для виробництва певного виду продукції;
- захищеність об’єкта від дії вторинних вражаючих факторів.

При вирішенні проблеми підвищення стійкості роботи підприємств приладо-будівної галузі керуються єдиними принциповими положеннями:

- завчасне проведення заходів цивільного захисту, спрямованих на зниження можливих втрат та руйнувань у разі застосування збоку противника зброї масового ураження і на створення умов для швидкого відновлення виробництва після часткового руйнування;
- комплексний підхід в розробці і здійсненні заходів для всіх напрямків діяльності підприємства;
- узгодження цих заходів з територіальними і військовими органами управління.

Заходи з підвищення стійкості плануються з урахуванням місцевих умов, ступеня важливості об’єкта, його географічного положення, економічної доцільності проведення заходів. На мирний час планують, в основному, трудомісткі заходи, які потребують значних матеріальних витрат і часу, а на період загрози виникнення НС – такі заходи, які не потребують значних затрат часу чи проведення яких не є доцільним при нормальному функціонуванні. Також при проведені заходів з ЦЗ потрібно враховувати і внутрішні фактори, що впливають на стійкість: розмір виробництва, виду продукції, що випускається, чисельність працівників, рівень їх дисциплінованості і компетентності, особливості технології виробництва, системи постачання виробництва сировиною, технічною і питною водою, газо- та електроенергією.

З урахуванням розглянутих вище факторів виділяють такі основні шляхи і способи підвищення стійкості роботи підприємств приладо-будівної галузі:

- забезпечення надійного захисту робітників і службовців: укриття робітників і службовців, які продовжують роботу на об'єкті у воєнний час;

проведення евакуації робітників, службовців і членів їх сімей та забезпечення їх життєдіяльності; використання індивідуальних засобів захисту;

- захист основних виробничих фондів об'єкта від поразки: підвищення певною мірою опірності будівель, споруд впливу ударної хвилі, світлового випромінювання; укриття найбільш уразливого обладнання в захисних пристроях (шатрах, камерах, конусах і ін.); часткову зміну технології виробництва; вивезення в безпечні райони надлишків горючих речовин;

- забезпечення сталого постачання об'єкта всім необхідним для виробництва: підвищення надійності роботи транспорту; підготовка паливноенергетичного господарства до роботи у воєнний час;

- підвищення надійності та оперативності управління виробництвом: створення об'єктового і заміського пункту управління; прокладка підземних кабельних ліній зв'язку до всіх елементів об'єкта; створення оперативних змін управління для основного і заміського пунктів управління;

- підготовка до виконання робіт по відновленню об'єкта у воєнний час: планування відновлювальних робіт за кількома варіантами; підготовка ремонтних бригад; створення необхідного запасу матеріалів і обладнання, надійний його захист; створення страхового фонду технічної документації.

Кожен шлях містить кілька способів підвищення стійкості роботи підприємства, які, в свою чергу, містять кілька заходів ЦЗ або доповнюються ними. Наведені вище шляхи підвищення стійкості підприємств приладо-будівної галузі реалізуються за допомогою затверджених норм з ЦЗ прийнятих і обов’язкових до виконання для всіх об’єктів усіх галузей виробництва не залежно від форм власності і підпорядкування. Норми ЦЗ призначені для:

- зменшення рівня руйнувань основних фондів виробництва;
- підвищення стійкості роботи об’єкта і галузей виробництва;
- забезпечення умов для ліквідації наслідків надзвичайних ситуацій;

Контроль за виконанням вимог згаданих норм покладається на Управління та відділи з питань надзвичайних ситуацій.

ВИСНОВКИ

Метою даної роботи було створення системи моніторингу трафіку для ієрархічної комп'ютерної мережі. Основні результати цієї роботи описані нижче.

В якості об'єкта моніторингу розглядалася ієрархічна комп'ютерна мережа. Розглядалися ієрархічна комп’ютерна мережа, топології фізичної мережі, ефективність дизайну ієрархічної мережі, плоска в порівнянні з ієрархічною топологією, плоска топологія WAN, плоска топологія LAN, сітчаста в порівнянні з ієрархічною топологією, класична трирівнева ієрархічна модель. Використання ієрархічної моделі може допомогти нам мінімізувати витрати. Ми можемо придбати відповідні міжмережні пристрої для кожного рівня ієрархії, таким чином уникаючи витрачання грошей на непотрібні функції для рівня. Крім того, модульний характер ієрархічної моделі проектування дає змогу точно планувати пропускну здатність на кожному рівні ієрархії, таким чином зменшуючи марну пропускну здатність. Відповідальність за керування мережею та системи керування мережею можуть бути розподілені між різними рівнями модульної мережевої архітектури для контролю витрат на управління.

Описано методи, моделі та алгоритми моніторингу трафіку в ієрархічних комп’ютерних мережах. Існує кілька підходів до моніторингу мережевого трафіку, кожен із яких має різні сильні та слабкі сторони. На даний момент існує три основні варіанти моніторингу трафіку – RMON, NetFlow, sFlow.

Описано аналіз протоколів моніторингу трафіку в ієрархічних комп'ютерних мережах. Узагальнено різні технології моніторингу з точки зору їх масштабованості та програм, для яких вони найкраще підходять.

Збираючи щохвилинну статистику сегментів і найкращих учасників, а також консолідацію матриці трафіку на всьому сайті, важко зрозуміти, як будь-який сервер RMON може підтримувати більше 100 агентів одночасно. З огляду на високу вартість одиниці технології агента RMON, навіть це досить низьке число додає до непомірно дорогого рішення. Стандарт RMON визначає, по суті, віддалений аналізатор протоколів. Його здатність фільтрувати, перехоплювати та декодувати пакети робить його застосовним для тих випадків усунення несправностей, коли проблему можна зрозуміти, лише побачивши послідовності протоколів із міткою часу на проводі. Технологія NetFlow може бути корисною для WAN-з’єднань, де важливо бачити кожен потік, можливо, для моніторингу безпеки або конкретної

тарифікації потік-за-потоком. Величезний обсяг згенерованих даних означає, що сервер не може керувати більш ніж 10 посиланнями. Техніка статистичної вибірки, яку використовує sFlow, добре масштабується для великої кількості агентів, десятки тисяч портів комутатора можуть керуватися одним сервером. Нульова вартість агента та значні технічні переваги роблять його ідеальним для безперервного моніторингу трафіку на сайті (і в масштабах підприємства) та звітності.

NetFlow — це важлива технологія, доступна у вашому пристрої Cisco, яка допомагає вам бачити, як використовуються ваші мережеві активи та поведінку мережі. NetFlow допоможе зменшити витрати, надаючи вам контрольний слід, скоротить час на усунення несправностей і полегшить звіти для розуміння використання мережі. Це допоможе впроваджувати нові IP-додатки та виявляти вразливі місця безпеки. NetFlow дозволить вам зрозуміти, хто використовує мережу, призначення трафіку, коли мережа використовується та тип програм, які споживають пропускну здатність. Оскільки існує велика різноманітність програмного забезпечення для датчиків, фонової служби збирача та систем звітності, ви можете пробачити новачку за думку, що Netflow надто складний, щоб починати його налаштування. У цьому розділі зроблено спробу показати, як ntop можна використовувати для вимірювання та моніторингу трафіку. Такі функції, як вбудований HTTP-сервер, підтримка різних типів мережевих носіїв, низьке використання процесора, переносимість на різні платформи, зберігання інформації про трафік у базі даних SQL, розширюваність за допомогою програмних компонентів та інтеграція з багатьма мережевими інструментами роблять ntop придатним для тих, хто хоче аналізувати мережевий трафік без необхідності платити за дорогі інструменти, які часто мають обмежений обсяг і не мають багатьох функцій, які пропонує ntop. Як ntop, так і libpcap для Win32 поширюються за ліцензією GPL2 і їх можна завантажити безкоштовно.

Відповідно до вищезазначеної інформації була створена система моніторингу трафіку ієрархічної комп'ютерної мережі з використанням технології NetFlow. Для цього було використано таке програмне забезпечення:

- softflowd як датчик;
- flow-capture (з flow-tools) як колектор;
- flow-stat (з flow-tools) для аналізу в текстовому рядку;

- ntop для візуалізації та представлення в WEB.
- Cisco Packet Tracer для моделювання ієрархічної мережі;
- Операційна система FreeBSD.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Sosinsky B.* Networking Bible: Original edition. – Wiley, 2009.- P.8-16.
2. *Oppenheimer P.* Top-Down Network Design: 2nd Edition. - Cisco Press, 2004.- P.135-145.
3. *Charles M. Kozierok* The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference: 1st Edition. - No Starch Press, 2005.- P.83-88.
4. Introduction to Cisco IOS NetFlow. http://www.cisco.com
5. Remote Monitoring. http://www.cisco.com
6. Simple Network Management Protocol. http://www.cisco.com
7. *McConnell J.* RMON Methodology: 2nd Edition with updated information. - McConnell Consulting, Inc., 1997.- P.2-14.
8. *PAT 6578077 B1 US*. Traffic monitoring tool for band / G. Rakoshitz. - №08/999517. Filed 29. 12. 97; published 05. 05. 98 – P. 3.
9. *Deri, L. and Suin, S.* Effective Traffic Measurement using ntop. IEEE Communications Magazine, 38(5). - P.138-145.
10. *Deri, L.* NTOP User's Guide - Network Usage Monitor for Unix Systems. Centro Serra, University of Pisa, Italy. Available at ftp://ftp.unipi.it/pub/local/ntop/snapshots/NTOP.pdf.gz
11. *Столлингс В.* Современные компьютерные сети / Столлингс В. – СПб.: Питер, 2003. – 783 с. - (Серия "Классика computer science").
12. *Игнатов В.А.* Статистическая оптимизация качества функционирования электронных систем. / Игнатов В.А., Маньшин Г.Г., Трайнев В.А. М.: Энергия. – 1974. 264 с.
13. *Игнатов В.А.* Теория информации и передачи сигналов. / Игнатов В.А. 2-е издание. - М.: Сов. радио, 1990, 280 с.
14. *Остерлох Хизер* «Маршрутизация в IP-сетях. Принципы, протоколы, настройка»: Пер. с англ. -Спб.: ООО "ДиаСофтЮП", 2002. – 512 с.
15. *Олифер Н.А.* Средства анализа и оптимизации локальных сетей. / Н.А. Олифер, В.Г. Олифер. М.: Центр Информационных Технологий, 1998, 120 с.
16. Policing and Shaping Overview, QC: Cisco IOS Release 12.0 Quality of Service Solutions Configuration Guide. http://www.cisco.com
17. *Waldbusser, S.* Remote Network Monitoring Management Information Base, IETF STD 0059, May 2000
18. *Stallings W.* SNMP, SNMPv2, SNMPv2 and RMON 1 and 2, Third Edition, Addison Wesley, Sept. 1999.
19. *Даниліна Г.В.* Методи і алгоритми оптимального управління трафіком в обчислювальних мережах. / Даниліна Г.В. Гузій М.М., Ігнатов В.О., Милокум Я.В. – Проблеми інформатизації та управління. К.: НАУ, 2006. - Вип.17. С. 32-37.
20. *Даниліна Г.В. Гузій М.М., Жуков І А., Ігнатов В.О.* Моделювання перехідних режимів трафіку комп'ютерної мережі. / Даниліна Г.В., Гузій М.М., Жуков І А., Ігнатов В.О. Информационные технологии и безопасность: Сборник научных трудов. – К.: НАН Украины, Институт проблем регистрации информации, 2006, Вып.9. – С. 84 – 87.
21. *Гузій М.М.* Оптимізація управління трафіком обчислювальних мереж. / Гузій М.М., Даниліна Г.В., Ігнатов В.О. VIII Міжнародна науково-технічна конференція «АВІА-2007».– К.: НАУ, – Т. 1. – С. 13.21-13.24.

22. Метод динамічної маршрутизації з підтримкою якості обслуговування в мобільних комп'ютерних мережах Автореф. дис... канд. техн. наук: 05.13.13 / І.А. Клименко; Нац. авіац. ун-т. — К., 2006. — 20 с. — укр..
23. *Ignatov V.O.* Optimal traffic service in communications networks. / Ignatov V.O., Wu Zijuan. Электроника и системы управления. К.: НАУ, 2008.-Вип.18. С.54-59.
24. *Case J.* Simple Network Management Protocol (SNMP) / J. Case // RFC 1157 – 1990. – May.
25. *Waldbusser S.* Remote Network Monitoring Management Information Base / S. Waldbusser // RFC 1757. – 1995. – Feb.
26. NetFlow Services Export Version 9, RFC 3954
27. *Jeffery S. Chase,* "Server Switching: Yesterday and Tomorrow," Proc. The Second IEEE Workshop on Internet Applications, 2001
28. *Vegesna S.* Качество обслуживания в IP сетях. – М.: Издательский дом «Вильямс», 2003.- 560 с. : ил.
29. *Miloucheva I., Műller E., Anzaloni A.,* A practical appoach to forecast Quality of Service parameters considering outliers. 2003.
30. *Манн С. Крелл М.* Linux. Администрирование сетей TCP/IP. Пер. с англ. - М.: ООО "Бином Пресс", 2003. - 656 с.
31. *W.E.Leland,* M.S.Taqqu, W.Willinger, and D.V.Wilson. On the self-similar nature of Ethernet traffic (extended version).IEEE/ACM Transactions of Networking, 2(1):1-15,1994.
32. *Guziy M.M., Ignatov V.O., Wu Zijuan* Comparative analysis of modeling adeguacy of tne nonstationary traffic in telecommunication networks. Зб. наук. праць: Випуск 3 (21). - К.: НАУ, 2007. - 147 с.
33. *Al-Sharo Ya. M., Guziy N.N., Ignatov V.A.* Optimum diagnosing of computer networks. Зб. наук. праць: Випуск 2 (20). - К.: НАУ, 2007. – С. 125-128.
34. Як підготувати і захистити дисертацію на здобуття наукового ступеня. Методичні поради / Автор-упорядник Л.А. Пономаренко, доктор технічних наук, професор. — К.: Редакція «Бюлетеня Вищої атестаційної комісії України», Видавництво «Толока», 2001.- 80 с. — Бібліогр. - С. 80.
35. Довідник здобувача наукового ступеня. - Ю: Редакція «Бюлетеня Вищої атестаційної комісії України», 2000. – С. 64.

ДОДАТОК А

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ІМЕНІ ІВАНА ПУЛЮЯ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку кваліфікаційної роботи

«Розробка системи моніторингу трафіку в ієрархічних комп’ютерних мережах з використанням технологій MySQL, PHP, Bash»

Розробники:виконавець ст. гр. СПм-61

Музиченко Євген Дмитрович

_____

(підпис)

Керівник кваліфікаційної роботи

Стоянов Юрій Миколайович

_____

(підпис)

Тернопіль 2022

Зміст

1. ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану підготовки магістрів за спеціальністю 121 «Інженерія програмногозабезпечення».

Тема кваліфікаційної роботи: Розробка системи моніторингу трафіку в ієрархічних комп'ютерних мережах з використанням технологій MySQL, PHP, Bash».

Термін виконання: до «___»_____2022р.

2. ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Програмний продукт призначений для моніторингу та фільтруфання трафіку.
Програма буде корисною в багатьох підприємствах де використовується одна, або більше комп'ютерних мереж
Програма дозволить вам зрозуміти, хто використовує мережу, призначення трафіку, коли мережа використовується, та тип програм, які споживають пропускну здатність.

3. ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Функціональні вимоги

Система повинна передбачати одну роль:

користувач

Для користувачів система надає можливість моніторити, відфільтровувати трафік в ієрархічних комп'ютерних мережах серед великої кількості підключенних до неї клієнтів.

Кінцевий програмний продукт придатний для тих, хто хоче аналізувати мережевий трафік без необхідності платити за дорогі інструменти, які часто мають обмежений обсяг і не мають багатьох функцій, які пропонує ntop. Як ntop, так і libpcap для Win32 поширюються за ліцензією GPL2 і їх можна завантажити безкоштовно

3.2 Технічні вимоги

Вимоги до адміністративної частини: будь- яка операційна система на базі UNIX з веб сервером(наприклад Apache).

Вимоги до клієнтської частини: будь-яка операційна система з можливістю підключення до мережі та веб браузером.

3.3 Програмні вимоги

Розробка адміністративної частини: PHP, Bash

Розробка клієнтської частини: Python

Додаткові бібліотеки :

- softflowd як датчик;
- flow-capture (з flow-tools) як колектор;
- flow-stat (з flow-tools) для аналізу в текстовому рядку;
- ntop для візуалізації та представлення в WEB.
- Cisco Packet Tracer для моделювання ієрархічної мережі;
- Операційна система FreeBSD

4. ЕТАПИ РОЗРОБКИ

Розробка інформаційної системи проводиться в наступному порядку:

аналіз предметної області, аналіз конкурентів та основних алгоритмів програмної системи

вибір засобів розробки

розробка математичної моделі та складових програмного комплексу оформлення супровідної документації;

здача проекту.

Результати виконання кожного етапу проекту погоджуються з керівником проекту.

5. СУПРОВІДНА ДОКУМЕНТАЦІЯ

Для інформаційної системи повинні бути розроблені наступні документи: завдання

пояснювальна записка до кваліфікаційної роботи;

презентація проекту;

рецензія на проект;

диск з проектом.

Пояснювальна записка до проекту оформляється згідно діючих вимог до нормоконтролю проектів.

6. ПОРЯДОК ЗДАЧІ ПРОЕКТУ

Розроблена інформаційна системи повинна відповідати вимогами, що складаються з перерахованих у п.3.1 цього документу характеристик.

Для здачі проекту необхідно підготувати весь перелік документів зазначенийу п.5 цього документу.

Приймання проекту проводиться спеціально створеною комісією в термін зазначені в п.1 цього документу.

7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ

Назва етапу	Відмітка*
Аналіз предметної області	
Архітектура системи	
Проектування системи	
Використання системи	
Супровідна документація	

* відмітки про виконання етапу ставляться керівником проекту

ДОДАТОК Б

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7-8 грудня 2022 року

Тернопіль 2022

УДК 004.41

Є. Музиченко, Ю. Стоянов
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

СИСТЕМА МОНІТОРИНГУ ТРАФІКУ В ІЄРАРХІЧНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

UDC 004.41

E. Muzychenko, Y. Stoyanov

TRAFFIC MONITORING SYSTEM IN HIERARCHICAL COMPUTER NETWORKS

Ключові слова: трафік, система моніторингу, ієрархічна комп'ютерна мережа, сенсор, колектор.

Key words: traffic, monitoring system, hierarchical computer network, sensor, collector.

Система моніторингу трафіку – це мережевий аналітичний інструмент, який перевіряє використання локальної мережі та забезпечує відображення статистики вивантаження та завантаження. Основною метою системи є моніторинг (і підрахунок) IP-трафіку між локальною мережею (LAN) та Інтернетом.

Система моніторингу трафіку забезпечує облік і моніторинг трафіку в реальному часі. Він дуже динамічний, кожне нове підключення реєструється та відстежується, ви можете використовувати його для підрахунку корисного трафіку завантаження та вивантаження комп'ютера або розширити його для побудови системи обліку трафіку для всіх комп'ютерів у локальній мережі вашої компанії.

Системи моніторингу дозволяють контролювати сотні і навіть тисячі параметрів, що стосуються роботи різних апаратних і прикладних підсистем. Крім того, вони забезпечують не тільки збір цих параметрів, але й виконують попередню статистичну обробку, полегшуючи наступний аналіз. На основі зібраних даних виявлені проблеми, що знижують загальну продуктивність системи, перспективний і сценарний аналіз. Зокрема, ми можемо оцінити завантаженість сервера за місяць, квартал чи рік, розрахувати параметри його роботи за рахунок збільшення кількості користувачів (запити, обсяг трафіку, тощо) або визначити завантаженість сервера підсистеми після оновлення.

Література

1. Даниліна Г. В., Гузій М. М., Ігнатов В. О. Методи і алгоритми оптимального управління трафіком в обчислювальних мережах. Проблеми інформатизації та управління. К: НАУ, 2006. Вип. 17. С. 32–37.

Додаток В

Налаштування Cisco IOS на маршрутизаторі 1

```
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router1
!
!
!
!
!
!
!
!
!
!
ip name-server 0.0.0.0
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 192.168.0.100 255.255.255.0
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 10.0.0.1 255.0.0.0
 clock rate 128000
!
interface Serial1/1
 no ip address
 shutdown
!
interface Serial1/2
 no ip address
```

```
shutdown
!
interface Serial1/3
 no ip address
 shutdown
!
router rip
 network 10.0.0.0
 network 192.168.0.0
 network 192.168.1.0
!
ip classless
!
!
!
!
!
!
!
!
!
!
line con 0
line vty 0 4
 login
!
!
!
end

Interfaces configuration:
FastEthernet0/0 is up, line protocol is up (connected)
  Hardware is Lance, address is 00d0.ff6b.7261 (bia 00d0.ff6b.7261)
  Internet address is 192.168.0.100/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
```

```
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 21 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     16 packets output, 1112 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Serial1/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 10.0.0.1/8
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
  5 minute input rate 15 bits/sec, 0 packets/sec
  5 minute output rate 15 bits/sec, 0 packets/sec
     17 packets input, 884 bytes, 0 no buffer
     Received 17 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     17 packets output, 884 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Налаштування Cisco IOS на маршрутизаторі 2

```
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router2
!
!
!
!
!
!
!
!
!
!
ip name-server 0.0.0.0
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 192.168.1.100 255.255.255.0
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 10.0.0.2 255.0.0.0
!
interface Serial1/1
 no ip address
 shutdown
!
interface Serial1/2
 no ip address
 shutdown
!
interface Serial1/3
 no ip address
```

```
 shutdown
!
router rip
 network 10.0.0.0
 network 192.168.0.0
 network 192.168.1.0
!
ip classless
!
!
!
!
!
!
!
!
!
line con 0
line vty 0 4
 login
!
!
!
End
Interfaces configuration:
FastEthernet0/0 is up, line protocol is up (connected)
  Hardware is Lance, address is 0090.2b45.93ee (bia 0090.2b45.93ee)
  Internet address is 192.168.1.100/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 21 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
```

```
     13 packets output, 916 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Serial1/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 10.0.0.2/8
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
  5 minute input rate 15 bits/sec, 0 packets/sec
  5 minute output rate 15 bits/sec, 0 packets/sec
     13 packets input, 676 bytes, 0 no buffer
     Received 12 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     13 packets output, 676 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Сценарій завантаження скрипта softflowd

```
#!/bin/sh


# Set this rc.conf variables
#   * softflowd_enable          // enable softflowd?
#   * softflowd_interfaces      // interfaces to listen

#   * softflowd_netflow_host   // collector host
#   * softflowd_netflow_port   // collector port
# in rc.conf variables
# Example:
# softflowd_interfaces="em0 em1 xl0"
# softflowd_netflow_host=
# softflowd_netflow_port=


. /etc/rc.conf

if ! PREFIX=$(expr $0 : "\(/.*\)/etc/rc\.d/$(basename $0)\$"); then

    echo "$0: Cannot determine the PREFIX" >&2
    exit 1
fi

echo "$softflowd_enable" | grep -qix yes || exit

[ -z "$softflowd_interfaces" ] && exit

 [ -x ${PREFIX}/sbin/softflowd ] || exit
 SOFTFLOWD=${PREFIX}/sbin/softflowd
 SOFTFLOWCTL=${PREFIX}/sbin/softflowctl

case "$1" in

        start)
                for interface in ${softflowd_interfaces}
                do
                    ${SOFTFLOWD}            -i            ${interface}            -n
"$softflowd_netflow_host":"$softflowd_netflow_port"
                    echo -n softflowd[$interface]" "

                    softflowd_netflow_port="`expr $softflowd_netflow_port + 1`"
```

```
                done
                ;;
        stop)
                ${SOFTFLOWCTL} shutdown && echo -n ' softflowd'
                ;;
        *)
                echo "Usage: `basename $0` {start|stop}" >&2
                ;;
        esac

exit 0
```

Сценарій завантаження скрипту flow-capture

```
#!/bin/sh
# Set this rc.conf variables
#   * flowcapture_enable          // enable flow-capture?
#   * flowcapture_port            // port to listen to netflow data [optional]
#   * flowcapture_dir             // directory to place netflow statistics files to
[optional]
#   * flowcapture_flags           // override default specified in the script
#                                    and port and dir variables [optional]

flowcapture_port=8818
flowcapture_dir=/var/netflow
flowcapture_pid=/var/run/flow-capture.pid

. /etc/rc.conf

if ! PREFIX=$(expr $0 : "\(/.*\)/etc/rc\.d/$(basename $0)\$"); then
    echo "$0: Cannot determine the PREFIX" >&2
    exit 1

fi

echo "$flowcapture_enable" | grep -qix yes || exit

case "$1" in

        start)
                [ -x ${PREFIX}/bin/flow-capture ] || exit
                FLOWCAPTURE=${PREFIX}/bin/flow-capture

                port="$flowcapture_port"
                for dir in $flowcapture_dir
                do

                    [ -z "$flowcapture_flags" ] && \
                    flags="-p  ${flowcapture_pid}  -n  287  -N  0  -w  ${dir}  -S  5
0/0/${port}"
                    ${FLOWCAPTURE} ${flags}
                    port="`expr $port + 1`"
                done
                ;;
        stop)
                port="$flowcapture_port"
```

```
                        for dir in $flowcapture_dir
                        do
                            [ -e "${flowcapture_pid}.${port}" ] || { echo NetFlow collector
flow-capture is not running on port $port; exit ; }
                            kill "`cat ${flowcapture_pid}.${port}`" 2> /dev/null && echo -n
' flow-capture:'$port' '
                            port="`expr $port + 1`"

                        done
                        ;;
                *)
                        echo "Usage: `basename $0` {start|stop}" >&2
                        ;;
                esac

        exit 0
```

Редагування секції Makefile.PL з Cflow.pm, де він перевіряє наявність бібліотеки flow-інструментів

```
sub find_flow_tools {
          my($ver, $dir);
          my($libdir, $incdir);
          if (-f '../../lib/libft.a') {
             $dir = '../../lib';
             $incdir = "-I$dir -I$dir/..";
             $libdir = "-L$dir";
          }

       Edit the line that reads
             if (-f '../../lib/libft.a') {
       to read
             if (-f '/usr/local/lib/libft.a') {
```