МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп’ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Карпец Дмитро Олександрович

Програмний засіб налаштування маршрутизаторів через веб-інтерфейс / Router setup utility via Web-interface

напрям підготовки: 123 Комп’ютерна інженерія
фахове спрямування - Комп’ютерна інженерія
Бакалаврська робота

Виконав: студент 4 курсу, групи КСМ-43/2
Карпец Д.О

Керівник
Мельник Г.М

ТЕРНОПІЛЬ - 2018

РЕЗЮМЕ

Дипломний проект містить 58 сторінок пояснюючої записки, 11 рисунків, 8 таблиць, 1 додаток.

Метою дипломного проекту є розроблення програмного засобу для налаштування маршрутизаторів через веб-інтерфейс.

Методи дослідження включають методи фізичної і логічної структуризації комп'ютерних мереж, методи структурного програмування, теорію графів, елементи математичної логіки.

Розроблено програмний засіб для налаштування маршрутизаторів через веб-інтерфейс. Розроблено узагальнену структуру та об'єктну модель представлення правил фільтрації мережевих протоколів. Розроблено узагальнений алгоритм фільтрації пакетів. При реалізації програмного забезпечення розроблено процедуру доступу і аутентифікації через Веб-інтерфейс маршрутизатора, процедуру витягнення поточних налаштувань через веб-інтерфейс, процедуру завантаження правил фільтрації у пристрій.

В якості інструментальних засобів розроблення обрано мову Python та бібліотеку розроблення графічних інтерфейсів Qt. Розроблено графічний інтерфейс програмного засобу і проведено його тестування.

Ключові слова: ПРОГРАМНИЙ ЗАСІБ, КОМП'ЮТЕРНА МЕРЕЖА, МАРШРУТИЗАЦІЯ, ВЕБ-ІНТЕРФЕЙС.

RESUME

Diploma project contains 58 pages of main text, 11 figures, 8 tables, 1 appendices.

The aim of the diploma project is the development of software tools for configuring routers through the web interface.

Research methods include methods of physical and logical structuring of computer networks, structured programming methods, graph theory, elements of mathematical logic.

Developed a software tool for configuring routers through the web interface. The generalized structure and object model representing filter rules network protocols. The generalized algorithm for packet filtering. When implementing software developed procedure for access and authentication through Web interface router process extraction current settings through a web interface, downloading of filtering rules in the device.

As development tools selected language Python and GUI development library Qt. Developed GUI software and its testing conducted.

Keywords: SOFTWARE TOOL, COMPUTER NETWORK, ROUTING, WEB-INTERFACE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ММЕ	–	Міжмережевий екран
ЛОМ	–	Локальна обчислювальна мережа
ДБЖ	–	Джерело безперебійного живлення
LAN	–	Local area network
VLAN	–	Virtual Local Area Network
VPN	–	Virtual Private Network
NE	–	Network Element
RTT	–	Round trip time
SLA	–	Service Level Agreement
SNMP	–	Simple network management protocol
MIB	–	Management Information Base
RMONMIB	–	The Remote Network MONitoring (RMON) MIB
TDR	–	Time Domain Reflectometry
RFC	–	Request for Comments
NVP	–	Nominal Velocity of Propagation

ВСТУП

В даний час комп'ютерні мережі стрімко ростуть й отримують великого розповсюдження. До кількості комп’ютерів при протоколі IPv4, варто до заданої кількості добавити поширені локальні та корпоративні мережі. Усі ці комп'ютери були з'єднані з ціллю ефективного обміну інформацією, тому користувачі комп'ютерів потребують продуктивного обміну при великій кількості інформації на далеку відстань.

Клієнти мережі потребують зростання рівню якості каналів передачі даних, з старих телефонних дротів ми прийшли до масового використання оптично-волоконних ліній, також використання супутникового зв'язку тощо. Проте велику роль при такій кількості з'єднаних в мережі комп'ютерів грає рівень якості протоколів, які в свою чергу здійснюють передачу даних серед серверів, протоколів маршрутизації, алгоритмів за допомогою яких вони побудовані.

Зважаючи, що 32-бітний розмір адреси в мережах буде поступово замінено 128-бітною, що приводить до зростання максимальної кількості вузлів у 4.294.967.296 раз, слід зазначити, що найважливішу роль буде грати збільшення якості самої маршрутизації пакетів даних серед серверними вузлами в Інтернеті.

Маршрутизація є задачею знаходження дороги серед комп’ютерним вузлом, який надсилає пакети даних та комп'ютерним вузлом одержувача, проте в пов’язаній моделі IP, для виконання поставленої мети, як правило відбувається пошук маршрутів до шлюзів серед комп’ютерних мереж. В час коли пакети з даними перебувають на індивідуальній мережі або можливо підмережі, проблеми маршрутизації вирішують за технологією, відповідній до інтерфейсу певної мережі. IP маршрутизація розпочинається, доті коли необхідно здійснити передачу інформації серед різних мереж з різними інтерфейсами. Якщо мережі отримувача та відправника зв’язані між собою,

тоді пакети даних передаються через зазначений шлюх, що об’єднує мережі. Якщо задані мережі не пов’язані шлюзом, тоді пакети даних мають проходити через мережі, які проходять між відправником і одержувачем інформації та шлюзами, які їх з’єднують. Коли пакети даних приходять на шлюз мережі отримувача, технологія маршрутизації цієї мережі задає напрям пакетам даних до отримувача.

Практичне значення. Розроблено програмний засіб, що дозволить розгортати типові правила фільтрації трафіку на маршрутизаторах за допомогою веб-інтерфейсу. Розроблено алгоритми аналізу стану та керування мережевим обладнанням через web інтерфейс.

Метою дипломного проекту є розроблення програмного засобу для налаштування маршрутизаторів через веб-інтерфейс. Для досягнення мети потрібно розв'язати такі задачі:

- дослідити процеси маршрутизації;
- створити загальні правила маршрутизації;
- проаналізувати програмне забезпечення автоматизації налаштування активного мережевого обладнання;
- описати мову сценаріїв програмування активного мережевого обладнання і формат зберігання конфігурацій;
- вибрати інструменти і розробити UML діаграму класів;
- реалізувати програмне забезпечення;
- провести тестування розроблених засобів.

1 МАРШРУТИЗАЦІЯ В КОМП'ЮТЕРНІЙ МЕРЕЖІ

1.1 Протоколи маршрутизації

Протокол маршрутизації - мережевий протокол, який використовується маршрутизаторами для визначення можливих маршрутів прямування даних в комп'ютерній мережі. Застосування протоколу маршрутизації дозволяє уникнути ручного введення всіх допустимих маршрутів, що, у свою чергу, знижує кількість помилок, забезпечує узгодженість дій усіх маршрутизаторів в мережі і полегшує працю адміністраторів.

Протоколи маршрутизації діляться на два види, що залежать від типів алгоритмів, на яких вони засновані (або можуть бути гібридними - поєднувати обидва підходи):

– дистанційно-векторні протоколи, засновані на Distance Vector Algorithm (DVA);

– протоколи стану каналу зв'язку, засновані на Link State Algorithm (LSA).

Так само протоколи маршрутизації діляться на два види залежно від сфери застосування:

– міждоменна маршрутизація;

– внутрішньодоменна маршрутизація.

RIP - абревіатура словосполучення Routing Information Protocol — одного із найрозповсюдженіших протоколів маршрутизації в невеликих комп'ютерних мережах, який дозволяє маршрутизаторам динамічно оновлювати маршрутну інформацію (напрямок і дальність в хопах), отримуючи її від сусідніх маршрутизаторів.

RIP - так званий протокол дистанційно-векторної маршрутизації, який оперує транзитними ділянками в якості метрики маршрутизації. Максимальна кількість хопів, в RIP - 15 (метрика 16 означає «нескінченно велику метрику»). Кожен RIP-маршрутизатор за замовчуванням віщає в мережу свою повну

таблицю маршрутизації раз в 30 секунд, досить сильно навантажуючи низькошвидкісні лінії зв'язку. RIP працює на 3 рівні (мережевий) стека TCP / IP, використовуючи UDP порт 520.

В сучасних мережевих середовищах RIP - не найкраще рішення для вибору в якості протоколу маршрутизації, так як його можливості поступаються більш сучасним протоколам, таким як EIGRP, OSPF. Обмеження на 15 хопов не дає застосовувати його у великих мережах. Перевага цього протоколу - простота конфігурування.

OSPF - протокол динамічної маршрутизації, заснований на технології відстеження стану каналу (link-state technology), що використовує для знаходження найкоротшого шляху Алгоритм Дейкстри.

Протокол OSPF був розроблений IETF в 1988 році. Протокол OSPF являє собою протокол внутрішнього шлюзу (Interior Gateway Protocol — IGP). Протокол OSPF поширює інформацію про доступні маршрути між маршрутизаторами однієї автономної системи.

Властивості OSPF:

- висока швидкість збіжності;
- підтримка мережних масок змінної довжини VLSM;
- відсутність обмежень досяжності;
- оптимальне використання пропускної здатності мережі;
- оптимальний вибір шляху маршрутизації.

Цей протокол є незапатентований тобто відкритий для громадськості протокол, таким же, як є протокол RIP. Але OSPF на відміну від RIP, має значно більшу швидкість збіжності (рекалькуляції таблиці маршрутизації), немає обмеження на довжину шляху 15-ма хопами, враховує пропускну здатність мережі при виборі маршруту. Все це робить OSPF потужним, масштабованим протоколом маршрутизації.

BGP - з 1994 року єдиний протокол маршрутизації між автономними системами в глобальній мережі Інтернет.

BGP є протоколом міждоменної маршрутизації та належить до класу дистанційно-векторних протоколів. Як протокол міждоменної маршрутизації використовується усіма інтернет-провайдерами, а також великими компаніями та організаціями, які мають власні публічні номери автономних систем (ASN) та користуються послугами більш ніж одного інтернет-провайдера.

Разом з тим немає ніяких обмежень на використання BGP в локальних мережах крім рекомендацій про приватні ASN, але використання BGP як протоколу внутрішньодоменної маршрутизації є недоцільним через значний час конвергенції (збіжності) у порівнянні з іншими протоколами маршрутизації, що закладено в його дизайні.

TP-Link - міжнародний виробник комп'ютерного та телекомунікаційного обладнання. Головний офіс і заводи компанії розташовані в Китаї, Шеньчжень.

Компанія заснована в 1996 році в місті Шеньчжень. Назва TP-Link є скороченням від «Twisted Pair» - вита пара, «link» - з'єднання. Згодом TP стали трактувати як «Trust and Performance».

У 2005 році вийшла на світовий ринок. У 2007 році були відкриті представництва компанії в Сінгапурі та Індії. У 2008 році відкриті офіси в США і Німеччині. У 2009 році відкрито офіс в Росії. У 2011 році відкриті офіси в Польщі і Україні. Компанія отримала високі місця між іншими виробниками мережевого устаткування, в тому числі маршрутизаторів, модемів, в цілому електроніки, та перше місце в Китаї серед постачальників мережевого обладнання на ринку SOHO.

Маршрутизатори TP-Link мають функцію між мережевого екрану SPI (Stateful Packet Inspection). Міжмережевий екран SPI - Функція фільтрації з урахуванням контенту (Stateful Packet Inspection) допомагає запобігти кібератакам, оскільки протягом сесії відслідковується більша кількість параметрів. Під час сесії проводиться перевірка трафіка на відповідність протоколу. При заводських настроюваннях міжмережевий екран включений. Якщо потрібно, щоб усі комп'ютери локальної мережі мали можливість обмінюватися інформацією із зовнішнім світом, його можна відключити.

MikroTik – латвійський виробник мережевого устаткування. Компанія розробляє і продає Ethernet та безпровідне мережеве обладнання, зокрема маршрутизатори, мережеві комутатори, точки доступу, а також програмне забезпечення - операційні системи та допоміжне ПЗ. Компанія була заснована в 1996 році з метою продажу обладнання на ринках, що розвиваються.

Обладнання Mikrotik – складається із заліза, під загальним найменуванням RouterBoard, і операційної системи RouterOS. є напівпрофесійним мережевим обладнанням. На поточний момент воно може забезпечити достатньо широкий функціонал від простого комутатора, до складних схем комутації та маршрутизації трафіку. Mikrotik RouterOS Level6 - це операційна система на основі Linux, яка перетворює звичайний комп'ютер або MikroTik RouterBOARD у потужний маршрутизатор. Вона має високу продуктивність і широким функціоналом. Система настроюється через Web-Інтерфейс, графічну утиліту Winbox, Telnet або SSH. Маршрутизатори Mikrotik мають складні вбудовані функції між мережевого екрану.

1.2 Засоби розгортання вбудованого програмного забезпечення

Основні пристрої налаштування якими автоматизується розроблюваними засобами це маршрутизатори та міжмережеві екрани.

Маршрутизатор – це спеціалізований мережний комп'ютер, що має як мінімум один мережний інтерфейс і пересилаючий пакети даних між різними сегментами мережі, що зв'язує різнорідні мережі різних архітектур, що приймає рішення про пересилання на підставі інформації про топологію мережі, і на основі певних правил, заданих адміністратором.

Часто маршрутизатор не обмежується простим пересиланням даних між інтерфейсами, а також виконує й інші функції: захищає локальну мережу від зовнішніх загроз, обмежує доступ користувачів локальної мережі до ресурсів інтернету, роздає IP-адреси, шифрує трафік і багато іншого.

Маршрутизатори працюють на мережевому рівні моделі OSI: можуть пересилати пакети з однієї мережі до іншої. Для того, щоб надіслати пакети в потрібному напрямку, маршрутизатор використовує таблицю маршрутизації, яка зберігається у його пам'яті. Таблиця маршрутизації може складатися засобами статичної або динамічної маршрутизації.

Крім того, маршрутизатори можуть здійснювати трансляцію адреси відправника й одержувача (NAT), фільтрацію транзитного потоку даних на основі певних правил з метою обмеження доступу, шифрування/дешифрування передаваних даних тощо.

Маршрутизатором може виступати як спеціалізований пристрій, так і звичайний комп'ютер, що виконує функції простого маршрутизатора.

Основні функції програмного забезпечення маршрутизатора:

– збір та ввід інформації про внутрішні та зовнішні сервіси мережі;
– генерація та зберігання уніфікованих правил фільтрації;
– налаштування та підключення маршрутизаторів;
– запис правил фільтрації у маршрутизатори через відповідні інтерфейси.

Для прикладу розглянемо RouterOS від MikroTik. RouterOS – це мережева операційна система на базі Linux. RouterOS призначена для установки на маршрутизатори Mikrotik RouterBoard. Також дана система може бути встановлена на ПК, перетворюючи його в маршрутизатор з функціями брандмауера, VPN-сервера/клієнта, QoS, точки доступу та іншими.

Операційна система має кілька рівнів ліцензій зі зростаючим числом функцій. Крім того, існує програмне забезпечення під назвою Winbox, яке надає графічний інтерфейс для налаштування RouterOS. Доступ до пристроїв під управлінням RouterOS можливий також через FTP, Telnet, і SSH. Існує також API, що дозволяє створювати спеціалізовані додатки для управління і моніторингу.

Особливості RouterOS в тому, що вона підтримує безліч сервісів і протоколів, які можуть бути використані середніми або великими провайдерами - таких, як OSPF, BGP, VPLS / MPLS. RouterOS - досить гнучка

система, і дуже добре підтримується Mikrotik, як в рамках форуму і надання різних Wiki-матеріалів, так і спеціалізованих прикладів конфігурацій.

RouterOS забезпечує підтримку практично всіх мережевих інтерфейсів на ядрі Linux. З бездротових чіпсетів підтримуються рішення на основі Atheros і Prism (станом RouterOS версії 3.x). Mikrotik також працює над модернізацією програмного забезпечення, яка забезпечить повну сумісність пристроїв і ПЗ Mikrotik з набирающими популярність мережевими технологіями, такими як IPv6.

RouterOS надає системному адміністратору графічний інтерфейс (WinBox) для наочної і швидкої настройки брандмауера, маршрутизації та управління QoS. В тому числі, в інтерфейсі WinBox практично повністю реалізована функціональність Linux-утиліт iptables, iproute2, управління трафіком і QoS.

Cisco IOS (Internetwork Operating System - міжмережевий Операційна Система) - програмне забезпечення, що використовується в маршрутизаторах і мережевих комутаторах Cisco. Cisco IOS є багатозадачною операційною системою, яка виконує функції мережевої організації, маршрутизації, комутації та передачі даних.

В Cisco IOS є специфічний інтерфейс командного рядка (command line interface, CLI), який був скопійований багатьма іншими мережевими продуктами. Інтерфейс IOS пропонує набір багатослівних команд, відповідно до обраного режиму і рівню привілеїв користувача. Global configuration mode надає можливість для зміни налаштувань системи і мережевих інтерфейсів.

Всім командам приписується певний рівень привілеїв від 0 до 15, і до них можуть звернутися тільки користувачі з відповідним рівнем привілеїв. Через командний інтерфейс можна визначити доступні команди для кожного рівня привілеїв.

Інші виробники мережевого устаткування також використовують якісь свої операційні системи для управлянням устаткуванням, або використовують вже якісь загально готові шаблони ПЗ. Відповідно для кінцевого користувача надаючи нові версії прошивки для певного устаткування.

Відповідні об’єкти які потрібно буде розробити:

– графічний інтерфейс та дані про сервіси і параметри, перевірка коректності;

– структура даних правил, запис/читання з БД;

– процедура перевірки конекту, опитування, створення списку АМО;

– процедури роботи з інтерфейсом пристроїв (аналіз структури веб сторінки, аналіз скриптової мови), запис правил, перевірка працездатності.

1.3 Автоматизація реєстрації та обробки подій

Інженерні системи центрів обробки даних складаються з безлічі взаємопов'язаного обладнання, тому при настанні будь-якого тривожної події буває важко визначити, де саме виникла проблема. Для прикладу візьмемо проблему в контурі живлення, між розподільним щитом і активним мережевим обладнанням. Система повинна локалізувати проблему, визначати рівень можливих наслідків і відображати інформацію про конкретну систему в вікні тривог. Екранна форма зі схемою системи показує відносини між взаємопов'язаним обладнанням і можливими наслідками неполадок в окремих компонентах.

Мережевий аналізатор централізовано фіксує подію в базі даних і сповіщає диспетчера про виникнення проблеми та необхідності її розв'язання. Далі система визначає рівень серйозності події та присвоює їй певний пріоритет.

Система виводить повідомлення про вихід відслідковуваних параметрів за встановлені раніше межі, а також повідомлення про критичний час напрацювання експлуатованого інженерного обладнання. Схема роботи зображена на рисунку 1.2.

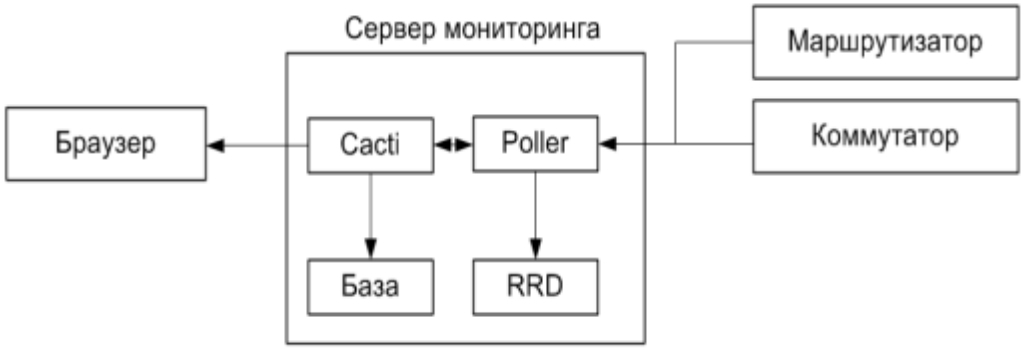

Рисунок 1.2 - Функціональна схема

Адміністрування і робота здійснюється через браузер. Для отримання нових графіків і шаблонів, ви входите в Cacti, як на сторінки Інтернету. Всі адміністративні дані будуть зберігатися в базі даних MySQL. На тому ж сервері, ви зможете знайти poller. Poller буде опитувати пристрої прописані в системі, будь то маршрутизатор, міні-АТС, сервер або додаток. Результати отримані з цих завдань зберігаються в RRD файлах. Система буде використовувати ці дані для створення графіків.

Система може бути розділена на три різні завдання, зображені на рисунку 1.3.

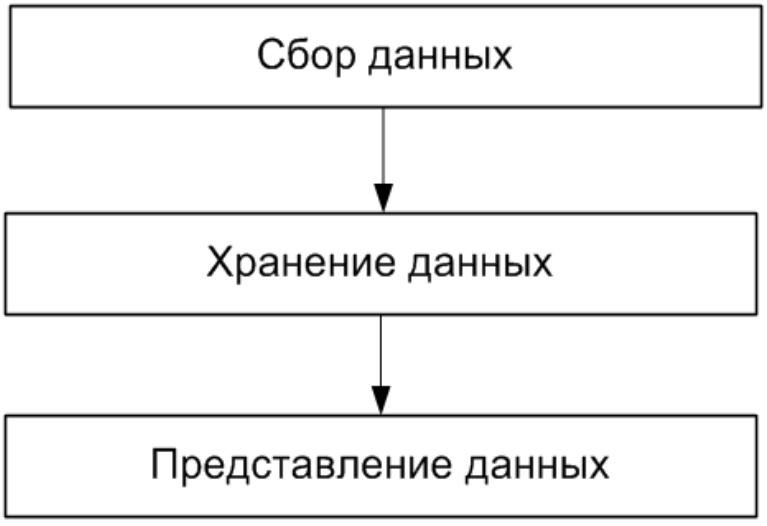

Рисунок 1.3 – Принципи роботи

Перше завдання полягає в отриманні даних. Система здійснює це за допомогою Poller. Poller запускається з планувальника операційної системи, наприклад, crontab для Unix ОС.

В існуючій IT-інсталяції, ви маєте справу з великою кількістю пристроїв різних типів, наприклад, серверів, мережевого обладнання, побутової техніки тощо. Для отримання даних від віддалених цілей/хостів, система буде в основному використовувати Simple Network Management Protocol SNMP. Таким чином, всі пристрої, які можуть використовувати SNMP, матимуть право перебувати під наглядом. Можна розширити можливості Cacti:

– скрипти викликають Data Input Method;

– скрипти викликають серверні скріпти для забезпечення більш високої продуктивності;

– SNMP запити до SNMP таблиць, що містить такі поняття, як інтерфейси і інші;

– скрипти запитів даних для отримання свідчень не використовуючи SNMP запити.

Є багато різних підходів при вирішенні цього завдання. Деякі з них можуть використовувати (SQL) бази даних, інші файли. Система використовує RRD для зберігання даних. RRD - скорочена назва Round Robin Database. RRD - система для зберігання і відображення зміни даних у часі (наприклад, пропускна здатність мережі, температура серверної, середнє навантаження на сервер). Вона зберігає дані дуже компактно і в той-же час досить, щоб можна було створювати інформативні графіки. Крім того, RRDtool буде виконувати певні завдання. Наприклад виконувати консолідацію для об'єднання необроблених даних і для зведених даних. Таким чином, застарілі дані стискаються для економії місця. RRDtool знає різні функції консолідації: AVERAGE, MAXIMUM, MINIMUM and LAST.

Однією з найпопулярніших особливостей RRDTool є вбудований графічний функціонал. Це буває корисно в поєднанні з деякими часто використовуваними веб-серверами. Так як це дає можливість отримати доступ

до графіків з будь-якого браузера з будь-якої платформи. Графік може бути намальований дуже по-різному. На графіку можливо зображення одного або кілька елементів на одному графіку. Підтримується автомасштабування і логоривфмування осі у. Ви можете нагромадити елементи один на інший і додати досить зручний опис, що позначає такі характеристики, як мінімальне, середнє, максимальне значення і багато іншого. Система об'єднує весь цей функціонал разом. Основою є PHP - це широко використовуваний універсальна мова сценаріїв, який особливо підходить для веб-розробки і може бути легко вбудований в HTML. Система реалізує функціонал Poller і використовує RRDTool для зберігання даних і створення графіків. Вся адміністративна інформація зберігається в базі даних MySQL.

1.4 Постановка задач бакалаврської роботи

Метою бакалаврської роботи є розроблення програмного засобу для налаштування маршрутизаторів. Для досягнення мети потрібно розв'язати такі задачі:

– проаналізувати технології маршрутизації;

– сформувати типові шаблони налаштувань засобів маршрутизації;

– проаналізувати програмне забезпечення автоматизації налаштування активного мережевого обладнання;

– описати мову сценаріїв активного мережевого обладнання і формат зберігання конфігурацій;

– вибрати інструменти і розробити UML діаграму класів;

– реалізувати програмне забезпечення;

– провести тестування розроблених засобів.

2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАСОБУ

2.1 Основні принципи та методи маршрутизації

Маршрутизація (Routing) – процес визначення маршруту прямування інформації між мережами. Маршрутизатор (або роутер) приймає рішення, що базується на IP-адресі отримувача пакету. Для того, щоб переслати пакет далі, всі пристрої на шляху слідування використовують IP-адресу отримувача. Для прийняття правильного рішення маршрутизатор має знати напрямки і маршрути до віддалених мереж.

Перші маршрутизатори представляли собою спеціалізоване ПЗ, яке обробляє вхідні IP-пакети специфічним чином. Це ПЗ працювало на комп'ютерах, у яких було кілька мережевих інтерфейсів, що входять до складу різних мереж (між якими здійснюється маршрутизація). Надалі з'явилися маршрутизатори в формі спеціалізованих пристроїв. Комп'ютери з маршрутизуючим ПЗ називають програмні маршрутизатори, технічне устаткування - апаратні маршрутизатори.

В сучасних апаратних маршрутизаторах для побудови таблиць маршрутизації використовується спеціалізоване ПО («прошивка»), для обробки ж IP-пакетів використовується комутаційна матриця (або інша технологія апаратної комутації), розширена фільтрами адрес в заголовку IP-пакета.

Є два типи маршрутизації:

- Статична маршрутизація – маршрути задаються вручну адміністратором;
- Динамічна маршрутизація – маршрути обчислюються автоматично за допомогою протоколів динамічної маршрутизації – RIP, OSPF, EIGRP, IS-IS, BGP, HSRP та ін, які отримують інформацію про топологію і стан каналів зв'язку від інших маршрутизаторів у мережі.

Оскільки статичні маршрути конфігуруються вручну, будь-які зміни мережної топології вимагають участі адміністратора для додавання і видалення

статичних маршрутів відповідно до змін. У великих мережах підтримка таблиць маршрутизації вручну може вимагати величезних витрат часу адміністратора. У невеликих мережах це робити легше. Статична маршрутизація не має можливості масштабування, яку має динамічна маршрутизація через додаткові вимоги до налаштування і втручання адміністратора. Але і у великих мережах часто конфігуруються статичні маршрути для спеціальних цілей у комбінації з протоколами динамічної маршрутизації, оскільки статична маршрутизація є стабільнішою і вимагає мінімум апаратних ресурсів маршрутизатора для обслуговування таблиці.

Динамічні маршрути виставляються іншим чином. Після того, як адміністратор активізував і налаштував динамічну маршрутизацію за одним з протоколів, інформація про маршрути оновлюється автоматично в процесі маршрутизації після кожного отримання з мережі нової інформації про маршрути. Маршрутизатори обмінюються повідомленнями про зміни у топології мережі в процесі динамічної маршрутизації.

Протокол маршрутизації може працювати тільки з пакетами, що належать до одного з маршрутизованих протоколів, наприклад, IP, IPX або Xerox Network System, AppleTalk. Маршрутизовані протоколи визначають формат пакетів (заголовків), найважливішою інформацією з яких для маршрутизації є адреса призначення. Протоколи, які не підтримують маршрутизацію, можуть передаватися між мережами за допомогою тунелів. Подібні можливості зазвичай надають програмні маршрутизатори і деякі моделі апаратних маршрутизаторів.

Маршрутизація в мережі Інтернет заснована на протоколах TCP/IP. Передача інформації здійснюється за допомогою IP-пакетів, заголовок кожного IP-пакета містить IP-адресу одержувача і відправника пакету. Кожен пакет обробляється маршрутизатором відповідно до його таблиці маршрутизації. Таблиця, в свою чергу, містить інформацію, комп'ютера з якою адресою направляти пакети з тим чи іншим діапазоном адрес. Наприклад, всі пакети певного діапазону можуть направлятися іншому маршрутизатору, який «відповідає» за цей сегмент.

У ряді випадків маршрутизатор може перетворювати заголовок пакета, замінюючи адресу відправника та/або одержувача пакета. Зокрема, це відбувається при взаємодії локальної мережі (що має свої адреси) з глобальною мережею Інтернет. В цьому випадку локальна мережа може бути видна ззовні по одному глобальному IP-адресу. Для того, щоб маршрутизатор міг направляти пакети з одним глобальним адресом тим чи іншим одержувачам в локальній мережі, використовується таблиця NAT, де крім IP-адрес вказуються порти, що ідентифікують додатки, що встановлюють з'єднання. При цьому номери портів вказані не в заголовку IP-пакета, а в заголовку сегмента TCP або UDP (сегменти інкапсулюються в поле даних IP-пакетів). Це дозволяє здійснювати взаємно-однозначну ідентифікацію одержувача і відправника в тих випадках, коли за одним глобальним адресом знаходиться безліч комп'ютерів локальних мереж. Приклад структури таблиці NAT наведено в таблиці 2.1.

Таблиця 2.1 – Приклад таблиці NAT

Глобальний адрес	Локальний адрес
208.164.201.225:1445	192.168.1.15:1445
208.164.201.225:1446	192.168.1.26:1445

В комп’ютерних мережах є декілька методів розсилки, розглянемо самі головні, які зображені на рисунку 2.1. В теорії комп'ютерних мереж anycast (відправка даних кому завгодно») - метод розсилки пакетів, що дозволяє пристрою посилати дані найближчому з групи одержувачів. Реалізовано, зокрема, в протоколі IPv6.

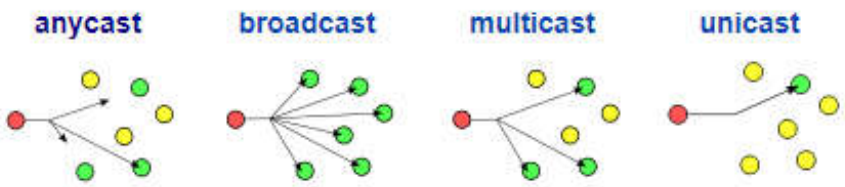

Рисунок 2.1 – Методи розсилки

У протоколі IP, anycast реалізований шляхом публікації однакового маршруту з різних точок мережі через протокол BGP. Одним з основних критеріїв вибору маршруту в BGP є AS-path - набір (список) номерів автономних систем, через які повинен пройти пакет; вибирається маршрут з найкоротшим списком AS-path. При отриманні анонса маршрутів з двох і більше точок, буде обраний самий короткий. Через особливості топології мережі або її політики найближчий вузол не обов'язково буде географічно найближчим. В даний час anycast використовується в мережі Internet для зменшення часу відгуку і для балансування навантаження кореневих DNS-серверів.

Широкомовний канал, широкомовлення (Broadcasting) - метод передачі даних в комп'ютерних мережах, при якому певний потік даних (кожен переданий пакет в разі пакетної передачі) призначений для прийому усіма учасниками мережі.

В TCP / IP широкомовлення (broadcast) можливо тільки в межах одного сегмента мережі (L2 або L3). Однак пакети даних можуть бути послані через меж сегмента, в який буде здійснено широкомовлення (наприклад, передача пакета на широкомовний IP-адрес через маршрутизатор за межами мережі). Навантаження на мережу в разі широкомовлення не відрізняється від звичайної передачі даних одному адресату, оскільки пакети даних не розмножуються (на відміну від unicast).

Прикладом широкомовлення є визначення MAC-адреси, певного IP-адресу (наприклад, за допомогою протоколу ARP). В цьому випадку відправляється широкомовний пакет із запитом, який досягає всіх підключених до даного L2-домену мережі пристроїв. Пристрій з необхідною IP-адресою відправляє у відповідь пакет, що містить необхідний MAC-адрес. Широкомовною IP адресою є остання адреса в підмережі.

Мультимовлення, багато-адресне мовлення (групова передача) - форма широкомовлення, при якій адресою призначення мережевого пакету є мультікастна група (один до багатьох). Існує мультимовлення на канальному, мережевому і прикладному рівнях.

Ряд додатків, наприклад, дистанційне навчання, розсилка пошти, радіо, відео, відеоконференцзв'язок, підтримують мультимовлення. У одно адресній мережі з кожним одержувачем встановлюється індивідуальне з'єднання навіть при споживанні одного ресурсу по загальному маршруту. У багато адресній розсилці джерело посилає єдиний екземпляр даних по загальному маршруту тим одержувачам, хто підписався на розсилку. Перевага цього підходу: додавання нових користувачів не тягне за собою необхідність збільшення пропускної здатності мережі по загальному маршруту до споживачів послуги. Відповідно, знижується навантаження і на проміжне обладнання.

В теорії комп'ютерних мереж unicast або односпрямована (одностороння) передача даних має на увазі під собою передачу пакетів єдиному адресату. Дана схема пакетної маршрутизації даних є прямим протиставленням широкомовної схеми маршрутизації.

2.2 Структура правил фільтрації пакетів

Вікно налаштування правил фільтрації трафіка для маршрутизатора TP-Link TL-WR841N / TL-WR841ND зображено на рисунку 2.2. Кожне правило має 5 структурних елементів.

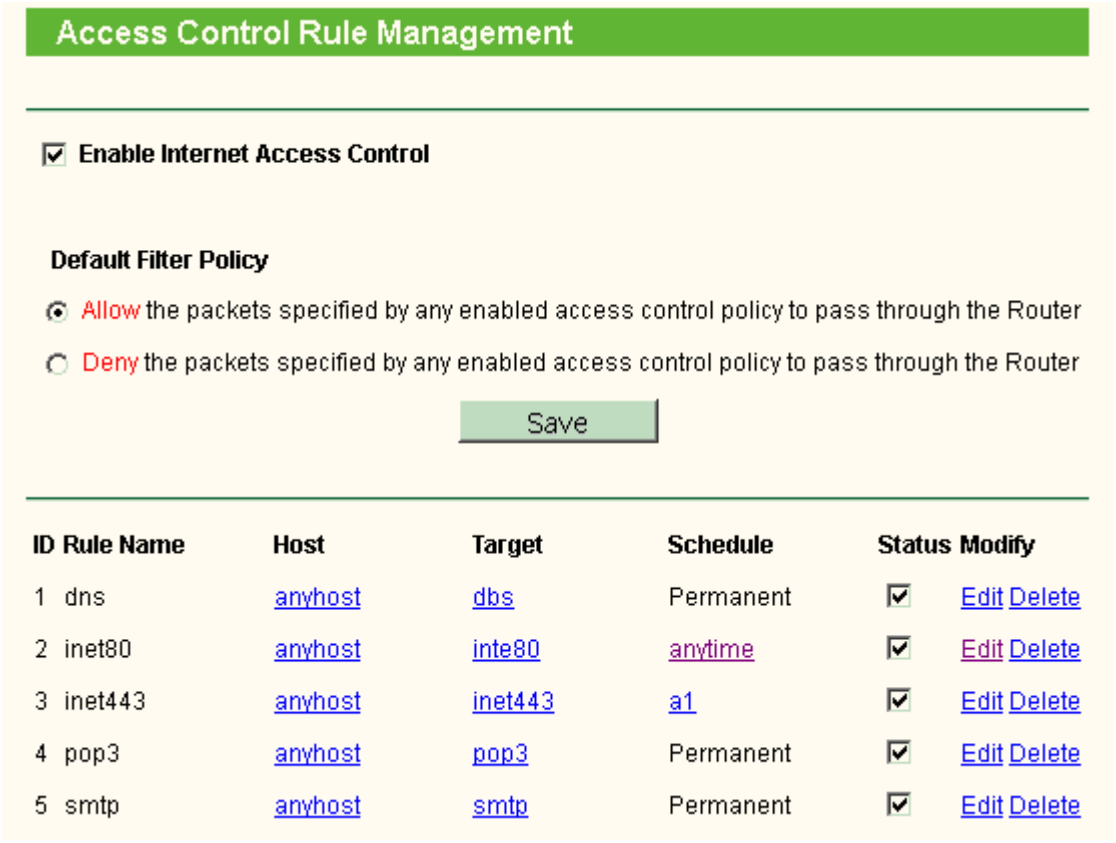

Рисунок 2.2 – Таблиця правил фільтрації трафіку

Для кожного правила контролю доступу (рисунок 2.3) в таблиці можна налаштувати наступні параметри:

– Ім'я правила (Name) - Дане поле містить ім'я правила, яке повинне бути унікальними;

– Вузол (Host) - Тут відображається вузол мережі, у відношенні якого діє зазначене правило;

– Ціль (Target) - Ціль, задана відповідним правилом;

– Розклад (Schedule) - Розклад, заданий відповідним правилом;

– Стан (Status) – Включено означає, що правило буде застосовуватися, виключено (знята галочка) – правило не застосовується.

Міжмережевий екран Mikrotik. Розглянемо основні поняття: ланцюжок (chain), стан з'єднання (connection state), умова, дія (action).

Ланцюжок (chain). При фільтрації трафік, залежно від свого призначення попадає в один з ланцюжків (chain) обробки трафіка. У фільтрі визначено три

основні ланцюжки:

- input вхідний трафік призначений для маршрутизатора. Наприклад, коли ми підключаємося до маршрутизатора за допомогою застосування winbox, трафік саме попадає в цей ланцюжок;
- output Вихідний трафік. Трафік, створюваний самим маршрутизатором. Наприклад, якщо ми виконуємо команду ping безпосередньо із самого маршрутизатора, трафік потрапить у цей ланцюжок;
- forward трафік, що йде через маршрутизатор. Наприклад, якщо комп'ютер з локальної мережі, установив з'єднання із зовнішнім сайтом, даний трафік попадає в ланцюжок forward.

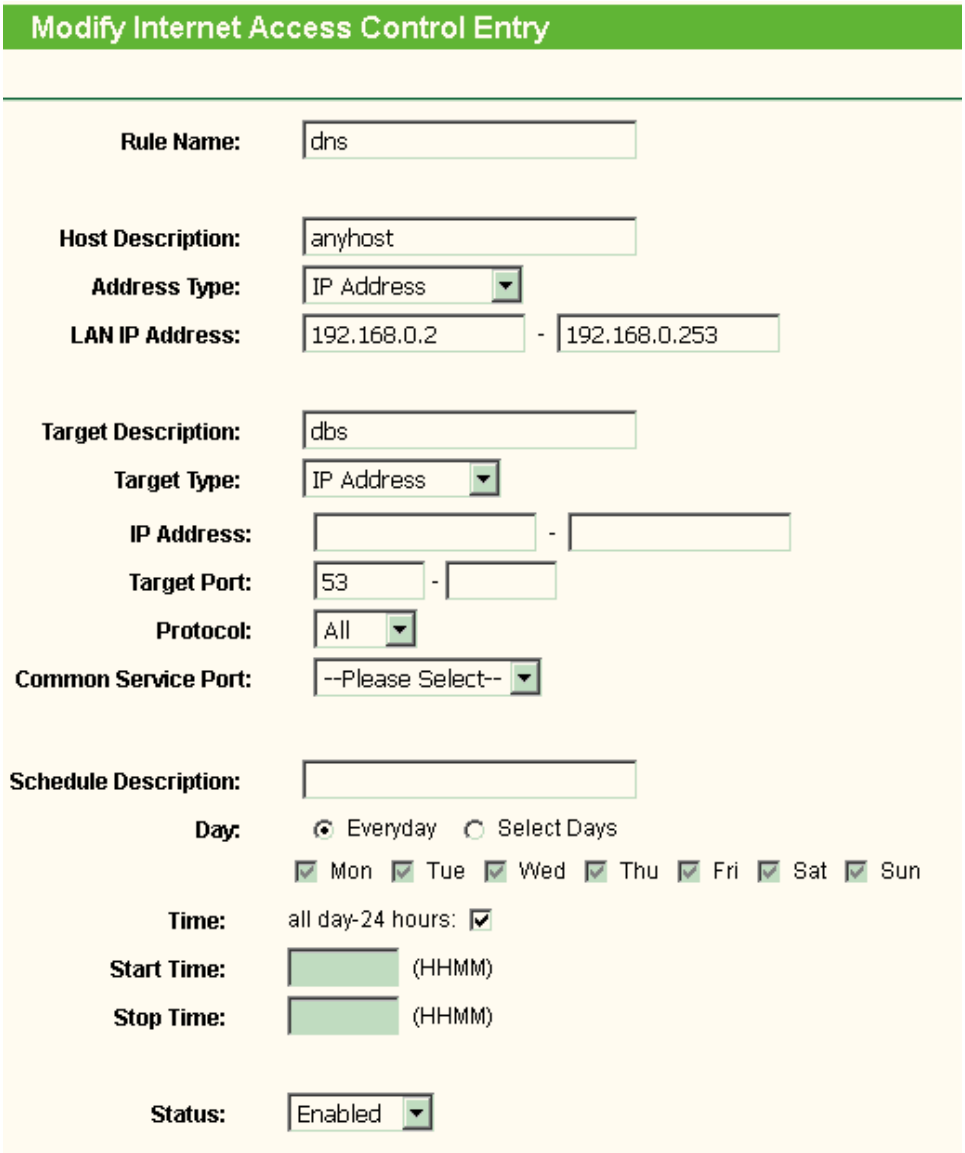

Рисунок 2.3 – Параметри одного правила фільтрації

У такий спосіб ми бачимо, що для захисту самого маршрутизатора необхідно використовувати ланцюжок input, а для захисту й фільтрації трафіка між мережами необхідно використовувати ланцюжок forward.

Крім того, адміністратор має можливість створювати свої власні ланцюжки обробки трафіка, до яких можна звертатися з основних ланцюжків. Буде розглянуто далі.

Стан з'єднання (connection state). Кожне з мережних з'єднань Mikrotik відноситься до одного з 4 станів:

- New – Нове з'єднання. Пакет, що відкриває нове з'єднання, ніяк не пов'язане із уже наявними мережними з'єднаннями, оброблюваними в цей момент маршрутизатором;
- Established – Існуюче з'єднання. Пакет відноситься до вже встановленого з'єднання, оброблюваного в цей момент маршрутизатором;
- Related – Зв'язане з'єднання. Пакет, який пов'язаний з існуючим з'єднанням, але не є його частиною. Наприклад, пакет, який починає з'єднання передачі даних в FTP-Сесії (він буде пов'язаний з керуючим з'єднанням FTP), або пакет ICMP, що містить помилку, що відправляється у відповідь на інше з'єднання;
- Invalid – Маршрутизатор не може співвіднести пакет з жодним з перерахованих вище станів з'єднання.

Виходячи з вищевикладеного, ми бачимо, що гарним варіантом настроювання фільтрації пакетів буде наступний набір умов:

- Обробляти нові з'єднання (connection state = new), приймаючи рішення про пропуск або блокування трафіка;
- Завжди пропускати з'єднання в стані established і related, тому що розв'язок про пропуск цього трафіка було прийнято на етапі обробки нового з'єднання;
- Завжди блокувати трафік, для якого стан з'єднання рівний invalid, тому що цей трафік не ставиться до жодного із з'єднань і фактично є

паразитним.

На рисунку 2.4 показано копію екрану діалогового вікна RouterOS для створення і редагування правил фільтрації трафіку.

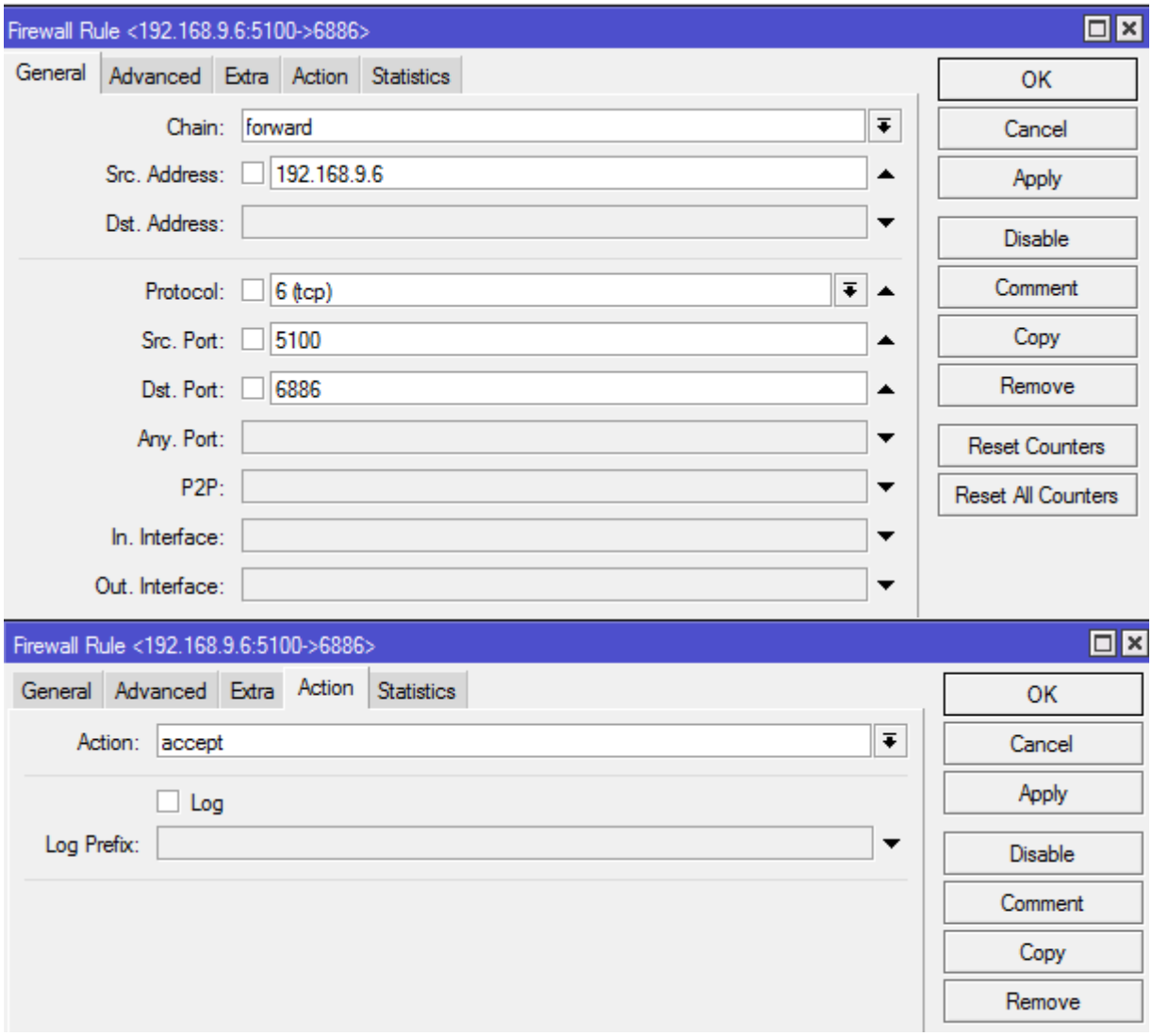

Рисунок 2.4 – Правило фільтрації для маршрутизатора Mikrotik

2.3 Структура програмного засобу та об’єктні моделі

Як бачимо структура правил маршрутизаторів різного класу дуже різна. Узагальнено можна сказати що правило має структуру ЯКЩО умова то ДІЯ.

Розроблена об’єктна модель представлення правил фільтрації (рисунок 2.5). Клас Rule представляє узагальнене правило фільтрації. Класи TP LINK SPI Rule та Mikrotik Firewall NAT Rule представляють правила фільтрації із атрибутами та методами відповідними до програмної моделі відповідних пристроїв.

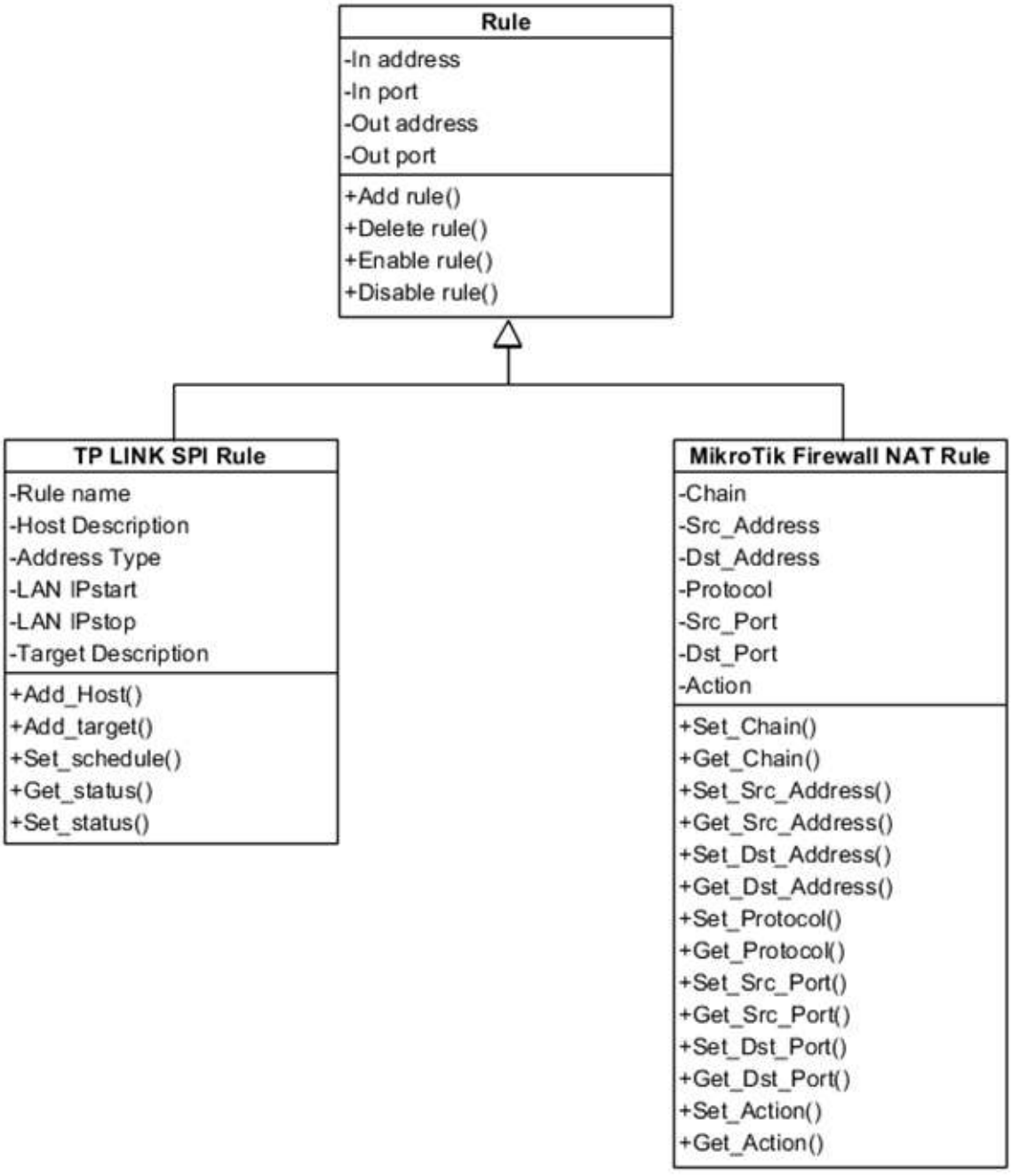

Рисунок 2.5 – Діаграма класів опису правила фільтрації

Атрибути класу Rule:

– In addres, Out address – адреси відправника, отримувача;

– In port, Out port – порти відправника, отримувача.

Методи класу Rule:

– Add rule(), Delete rule() додати видалити правило;

– Enable rule(), Disable rule() застосувати, відключити правило.

Атрибути класу TP LINK SPI Rule :

– Rule name – назва правила;

– Host Description – назву вузла;

– Address Type – тип адреси;

– LAN IPstart, LAN IPstop – діапазон IP адрес;

– Target description – опис цілі.

Методи класу:

– Add_Host() – додати вузол;

– Add_target() - додати ціль;

– Set_schedule() – додати правило запуску правила по часу;

– Get_status(), Set_status() – змінити статус.

Атрибути класу Microtik Firewall NAT Rule:

– Chain – ланцюжок;

– Src_Address, Dst_Address - адреси відправника, отримувача;

– Protocol – протокол;

– Src_Port, Dst_Port – порти відправника, отримувача;

– Action – дія з пакетом.

Методи класу Microtik Firewall NAT Rule:

– Set_Chain(), Get_Chain() – змінити ланцюжок;

– Set_Src_Address(), Get_Src_Address(), Set_Dst_Address(), Get_Dst_Address() – змінити адреси відправника, отримувача;

– Set_Protocol(), Get_Protocol() – змінити проткол;

– Set_Src_Port(), Get_Src_Port(), Set_Dst_Port(), Get_Dst_Port() – змінити порти відправника, отримувача;

– Set_Action(), Get_Action() – змінити дію виконувану над пакетом.

Операційна система пристроїв Мікротік має мову сценаріїв для виконання автоматизації рутинних операцій по налаштуванню. Перелічимо основні етапи типового налаштування:

– блокуємо всіх із чорного списку;
– фільтруємо корисний ICMP;
– блокуємо Bogon мережі;
– блокуємо DNS запити на зовнішній інтерфейс;
– захист від брутфорса сервіса SSH;
– захист від сканера портів;
– дозволяємо вже встановлені підключення й зв'язані connection-state=established;
– дозволяємо зовнішні підключення для власних потреб;
– дозволяємо вже встановлені підключення й зв'язані chain=forward connection-state=established.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ

3.1 Вимоги до функцій програмного засобу

Функція сканування мережі дозволить легко виявити хости в мережі і додати їх у список моніторингу з уже налаштованими за замовчуванням перевірками. Функція сканування мережі пропонує 2 способи пошуку хостів в мережі:

– сканування діапазону IP-адрес;

– імпорт з мережевого оточення.

Перший спосіб дозволяє виявити максимальну кількість пристроїв, і має наступні переваги:

– висока швидкість сканування діапазону вибору параметрів сканування і налаштувань мережі;

– дозволяє визначати різні види пристроїв, а саме принтери (локальні і мережеві), комутатори, хаби, сервери, сервери баз даних, роутери, WiFi точки доступу і т.д;

– застосовує відразу кілька ефективних способів пошуку пристроїв в мережі (ICMP-пінг, сканування списку TCP-портів, ARP-запити);

– дозволяє отримувати інформацію з пристроїв по SNMP;

– автоматично отримує багато іншої інформації про знайдені хости (IP, MAC-адреси, виробника мережевого адаптера, DNS-ім'я, тип ОС, підключені принтери, опис);

– дозволяє сканувати відразу кілька діапазонів IP-адрес.

Наступний спосіб це - імпорт з мережевого оточення, він працює трохи швидше, але не всі мережеві пристрої можуть бути знайдені (тільки комп'ютери і деякі сервера).

Якщо функції сканування мережі виявили не всі хости, ми можемо додати їх у список моніторингу вручну. Крім цього, використовуючи можливості програми, можна розбити знайдені хости на групи, створивши і перетягнувши в них хости курсором миші.

Кожному знайденому функцією сканування мережі хосту автоматично призначається специфічна перевірка, тип якої залежить від способу, за яким був знайдений цей хост (ICMP-пінг, TCP-порт, ARP-пінг). Можна залишити ці перевірки, змінивши при необхідності параметри за замовчуванням і додати додаткові перевірки. Всі зміни автоматично зберігаються в файлі і вступають в силу.

Всі функції програми доступні через контекстне меню, панель інструментів, головне меню, і будуть описані далі. Додатковою функцією програми є моніторинг стану пристроїв мережі. В основу її роботи покладено принцип періодичного запуску заданих для кожного хоста зі списку перевірок. Для кожного хоста можна задати кілька перевірок різного типу, про результати яких користувач може бути сповіщений налаштованою сигналізацією. Перевірка хостів проводиться паралельно декількома потоками. Кількість одночасно працюючих потоків можна налаштувати. Існує можливість завдання інтервалу перевірки для кожної перевірки - проміжок часу, протягом якого перевірка виконуватися не буде. Це дозволяє знизити трафік перевірочних пакетів в мережі під час роботи програми. Кількість типів перевірок не обмежена, в наступних версіях програми регулярно з'являтимуться нові типи. На даний момент доступні наступні перевірки: TCP-порт, ICMP-пінг, DNS, ARP, SNMP, порт на комутаторі, FTP, HTTP, NetBIOS, стан служби, існування процесу, існування папки, наявність файлу, розмір файла, дисковий простір, зовнішнє додаток (код завершення), Java-script, Visual Basic script, MS SQL Server, MySQL, Сервер БД (ODBC).

Перевірка "TCP-порт" дозволяє відстежувати доступність або недоступність будь-яких TCP-портів як локального, так і віддалених хостів. За допомогою цієї перевірки, зокрема, можна відстежувати роботу мережевих додатків. Наприклад, якщо «впав» сервер баз даних, програма може про це повідомити, виявивши заданий порт закритим.

Безумовно, по одній тільки ознаці відкритості порту неможливо визначити, як працює той чи інший сервіс. Для цього необхідні більш специфічні перевірки, які програма також уміє виконувати. Однак, в ряді

випадків, перевірка "TCP-порт" може виявитися навіть дуже корисною. Наприклад, для оперативного виявлення «троянів», «черв'яків» і інших небезпечних програм. Знаючи список TCP-портів, які часто використовуються такими програмами (списки можна знайти в Інтернеті), можна налаштувати кілька перевірок і моніторити ці порти. У разі якщо програма виявить, що будь-який із заданих портів відкрився, вона сповістить про це і необхідно буде терміново вживати додаткових заходів з діагностики хоста на предмери виявлення потенційно небезпечного ПЗ.

Інші перевірки для моніторингу сервісів, що працюють на основі протоколів TCP / IP:

– моніторинг FTP-серверів;

– моніторинг веб-серверів (HTTP), включаючи моніторинг контенту веб-сторінок.

У перерахованих перевірках, крім підключення до порту, програма перевіряє функціонування безпосередньо FTP- і HTTP-серверів.

За допомогою перевірки "Порт комутатора" програма дозволяє здійснювати моніторинг порту комутатора на предмет підключення певного хоста до цього порту.

Всі хости, підключені до комутатора, реєструються на ньому в спеціальних таблицях MAC-адрес і таблиці портів (таблиці маршрутизації). У цих таблицях ставиться відповідність фізичних адрес (MAC) хостів портів комутатора. Перевірка, заснована на протоколі SNMP, допомагає вчасно реагувати на перереєстрацію хостів з одного порту комутатора на інший. Для роботи перевірки необхідно задати список IP-адрес всіх комутаторів в мережі, їх Community (пароль для доступу до інформації, для читання) і вихідні значення номера порту і IP-адреси комутатора, на якому повинен бути прописаний хост. Перевірка періодично отримує з комутатора інформацію про реєстрацію хоста і порівнює з вихідними параметрами. Якщо хост, наприклад, виявляється на іншому комутаторі або на іншому порту - програма сигналізує про це.

Недоліки додатка.

Фактично, на сервер (іноді на окремий) ставиться якась програма, яка постійно опитує обладнання, а в SCOM перекидаються вже готові для візуалізації результати моніторингу. Через такий принцип роботи конфігурувати пакет доводиться не з інтерфейсу SCOM, а з інтерфейсу самого пакету, як правило, заплутаного і не інтуїтивного.

Пакети вміють збирати величезну кількість параметрів, але при цьому не вміють збирати саме ту інформацію, яка особливо цікава.

При грамотно налаштованих оповіщеннях в консоль управління SCOM'ом навіть заглядати не потрібно, все ясно з тих, хто прийшов на e-mail листів, ну або sms на телефон, кому як зручно.

В мережевому адмініструванні багато що залежить від якості мережі і від вимог, які пред'являються до цієї самої якості. Якщо залишити всі настройки пакета за замовчуванням, то адміністратори ризикують опинитися, або заваленими SMS-ками, або отримувати SMS раз в півроку при тому, що кінцеві користувачі будуть незадоволені якістю роботи мережі. Звичайно, кожен користувач пакета може налаштувати пороги під себе, але іноді порогів недостатньо, потрібно міняти набагато глибші механізми.

3.2 Вибір структур даних для реалізації правил фільтрації

Розглянемо структуру правила фільтрації пакетів. Брандмауери з пакетними фільтрами приймають рішення щодо того, пропускати пакет або відкинути, переглядаючи IP-адреси, прапорці або номери TCP портів у заголовку цього пакета. IP-адреса й номер порту - це інформація мережного й транспортного рівнів відповідно, але пакетні фільтри використовують і інформацію прикладного рівня, тому що всі стандартні сервіси в TCP/IP асоціюються з певним номером порту.

Для опису правил проходження пакетів складаються таблиці типу із стрічками представленими на рисунку 3.1.

дія	тип пакета	адреса джерела	порт джерела	адреса призначення	порт призначення	Прапорці

Рисунок 3.1 – Структура правила фільтрації

Поле "дія" може приймати значення пропустити або відкинути. Тип пакета - TCP, UDP або ICMP. Прапорці - прапорці із заголовка IP-пакета. Поля "порт джерела" і "порт призначення" мають сенс тільки для TCP і UDP пакетів.

Розглянемо узагальнену структуру фільтру пакетів на прикладі модуля iptables операційної системи Linux.

Пакети, що перехоплюються, зіставляються з набором правил, представлених у вигляді таблиць із регулярною структурою (ip_tables). Правила описують умови відповідності й указують подальші дії над пакетом, що попадають під задані умови: фільтрація, трансляція адрес (NAT) та інші перетворення.

Основні можливості:

- підтримка протоколів Ipv4 і Ipv6;
- Фільтрація мережного рівня (без контролю стану — stateless);
- Контроль стану (фільтрація сеансового рівня — stateful);
- Усі види трансляції адрес і портів (NAT/NAPT);
- Кілька шарів API для розширень сторонніх розроблювачів.

Цей брандмауер може використовуватися для контролю трафіка як на маршрутизаторах з декількома мережними інтерфейсам, так і на персональних робочих станціях, це зображено на рисунку 3.2, алгоритм обробки пакетів буде однаковим в обох випадках.

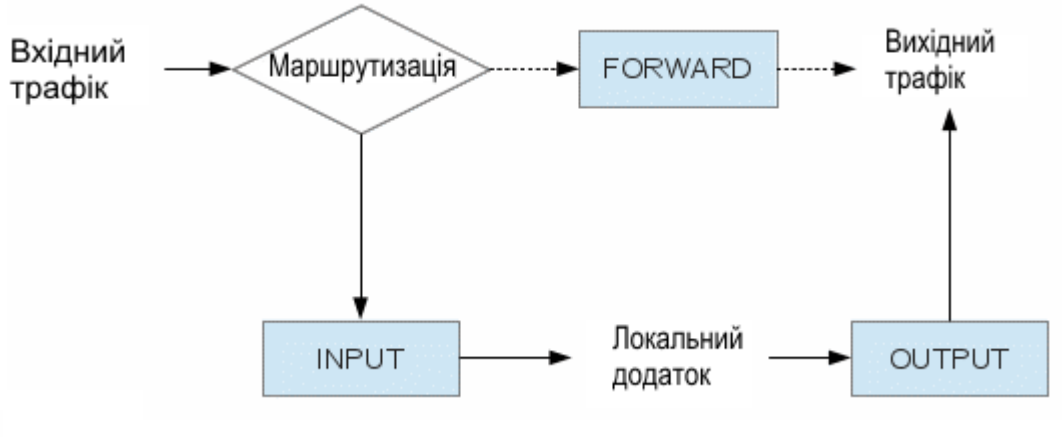

Рисунок 3.2 - Концепція обробки пакетів файєрволом netfilter

Алгоритм роботи наступний:

– для кожного вхідного пакета, захопленого файєрволом, визначається адресат і подальший маршрут;

– якщо адресатом є локальний процес, то пакет надходить у ланцюжок INPUT, де він перевіряється на предмет відповідності правилам цього ланцюжка. До пакета застосовується перше підходяще правило. Якщо пакет не відповідає жодному правилу ланцюжка, то він відкидається;

– якщо в системі кілька мережних інтерфейсів і настроєна переадресація, а пакет призначений для іншого мережного інтерфейсу, то він передається для обробки в ланцюжок FORWARD. Якщо переадресація не використовується або система не знає як обробити прийнятий пакет, то він відкидається;

– локальний процес може ініціювати відправлення пакетів; вихідні пакети надходять у ланцюжок OUTPUT.

Правила, ланцюжки й таблиці - це назви основних «будівельних блоків» netfilter/iptables:

– правило - структура, що описує критерії відповідності, застосовувана дія й лічильника. Якщо пакет відповідає критерію, до нього застосовується зазначена дія, і цей факт фіксується лічильником;

– ланцюжок - упорядкований набір правил, послідовно застосовуваних до пакета. На рис. 1 кольором відзначені ланцюжки INPUT, FORWARD і OUTPUT. Вони, а також ланцюжки PREROUTING і POSTROUTING, є

базовими й створюються при ініціалізації таблиць. Крім базових, iptables дозволяє створювати *користувацькі* ланцюжки;

– таблиця - сукупність базових і користувацьких ланцюжків, об'єднаних загальним функціональним призначенням.

Зв'язок між перерахованими елементами проілюстрована на рисунку 3.3.

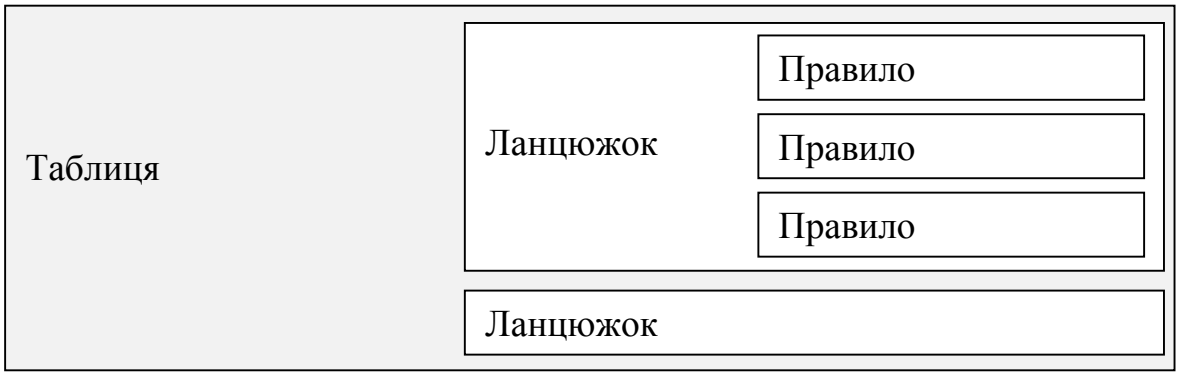

Рисунок 3.3 - Правила, ланцюжки й таблиці міжмережевого екрана Linux

У поточній версії міжмережевого екрана (iptables v1.4.16.3) підтримуються 4 вбудовані таблиці: Filter, NAT, Mangle, Raw.

Таблиця Filter містить ланцюжки правил фільтрації пакетів. Пакети можуть пропускатися далі, або відкидатися (дії ACCEPT і DROP відповідно), залежно від їхнього вмісту. Ця таблиця використовується за замовчуванням і представляє 3 ланцюжки:

– INPUT – правила фільтрації вхідних пакетів;

– OUTPUT – фільтр вихідного трафіка, котрий згенеровано локально;

– FORWARD – фільтрація маршрутизованого транзитного трафіка.

Таблиця NAT використовується для перетворення мережевих адрес (Network Address Translation). Через цю таблицю проходить тільки перший пакет з потоку. Перетворення адрес автоматично застосовується до всіх наступних пакетів:

– PREROUTING – Модифікація пакета перед маршрутизацією, виконувана відразу після того, як він попадає в систему. Звичайно

використовується для DNAT (destination NAT, перетворення адреси одержувача);

– POSTROUTING – Правила, застосовувані після маршрутизації. Модифікація пакетів виконується перед виходом їх із системи й, як правило, використовується для зміни (приховання) джерела, т.зв. source NAT (SNAT);

– OUTPUT – Правила трансляції адрес у пакетів, згенерованих локальними процесами.

Для розгортання набору правил фільтрації потрібно розробити:

– процедуру доступу і аутентифікації через веб-інтефейс маршрутизатора;

– процедуру витягнення поточних налаштувань через веб-інтерфейс;

– процедуру завантаження правил фільтрації у пристрій.

При запуску програмного засобу відбувається початкова ініціалізація бази даних правил. Далі програма переходить в інтерактивний режим і користувач може здійснити внесення і налаштування нових пристроїв. Із кожного пристрою користувач може завантажити правила фільтрації, модифікувати їх вручну або скористатися типовими шаблонами, розробленими на основі політик безпеки.

3.3 Розроблення екранних форм графічного інтерфейсу

Розробка форм програми. Вимоги: здійснення валідації, сортування, фільтрації. Розроблюваний мною програмний засіб відповідно до вимог працює в діалоговому режимі. Форми програми які відповідають за виконання певних функції, реалізовані за допомогою вкладок віджета QTabWidget. Звісно самим важливим елементом в програмі це є – таблиці, які показують перелік маршрутизаторів, а також правила фільтрацій. Поглянемо детальніше з чого саме складена таблиця QTableView, на рисунку 3.4.

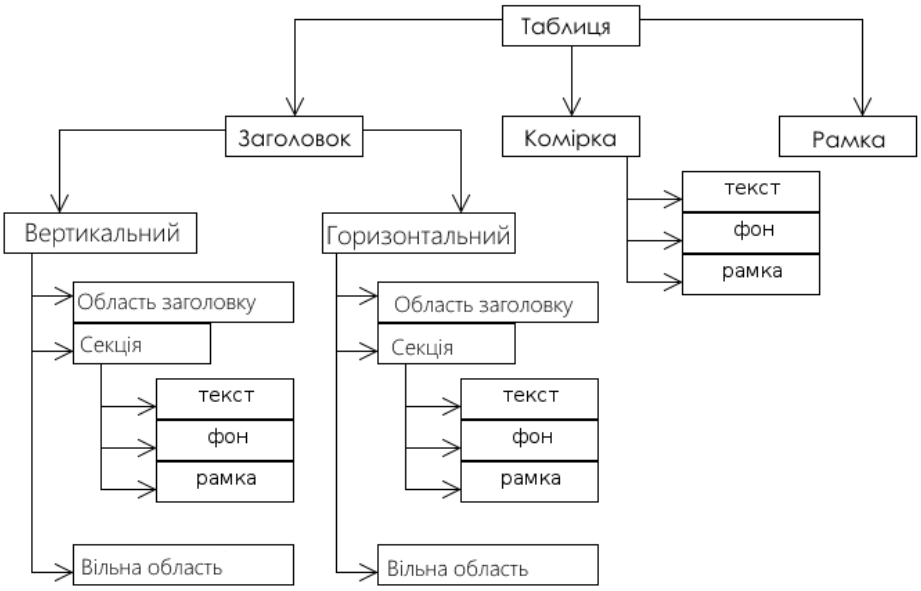

Рисунок 3.4 – Структура таблиці QTableView

Детальніше розглянеом заголовки. При використанні Qt, поділять два типа заголовків: горизонтальний (верхній), та вертикальний (лівий). В визначені «Область заголовка» розуміється вся область, в якій незабаром буде відображатия заголовок. Секція – це певна комірка в заголовку. Вільна область – це зона заголовка, яка не містить секцій (це відбувається, коли загальний розмір всіх секцій менший висоти самої таблиці).

Можна зосередитися на трьох режимах роботи системи:

– ввід, збереження і загрузка списку переліку маршрутизаторів з інформацією для аутентифікації;
– здійснення зчитування настройок пристроїв через веб-інтерфейс;
– здійснення загрузки настройок пристроїв через веб-інтерфейс.

Інформація, яка необхідна для додавання пристрою: назва пристрою, IP-адреса, порт, логін та пароль. Користувач при відкриті програми, здійснює додавання пристроїв в базу і задає інформацію для аутентифікації. На рисунку 3.5 зображено саму першу вкладку початкового вікна системи.

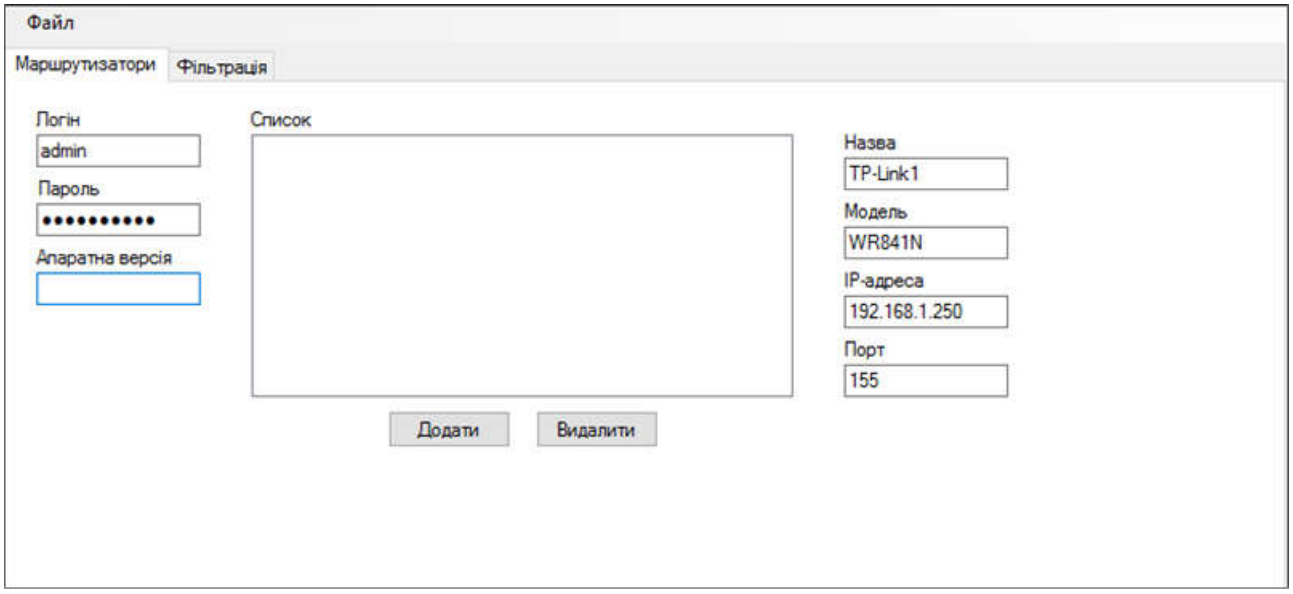

Рисунок 3.5 – Екран списку пристроїв

Після етапу з'єднання з необхідним пристроєм, користувач системи може виконати зчитування правила фільтрації з даного маршрутизатора. Відповідно до індивідуальних правил безпеки, користувач за необхідністю може відредагувати правила або створити нові. На рисунку 3.6 зображено екран другої вкладки програми, на якому можна здійснити додавання нового правила фільтрації та запису його на маршрутизатор, або наприклад зчитування вже виконуваних правил, здійснення їх редагування та зберігання.

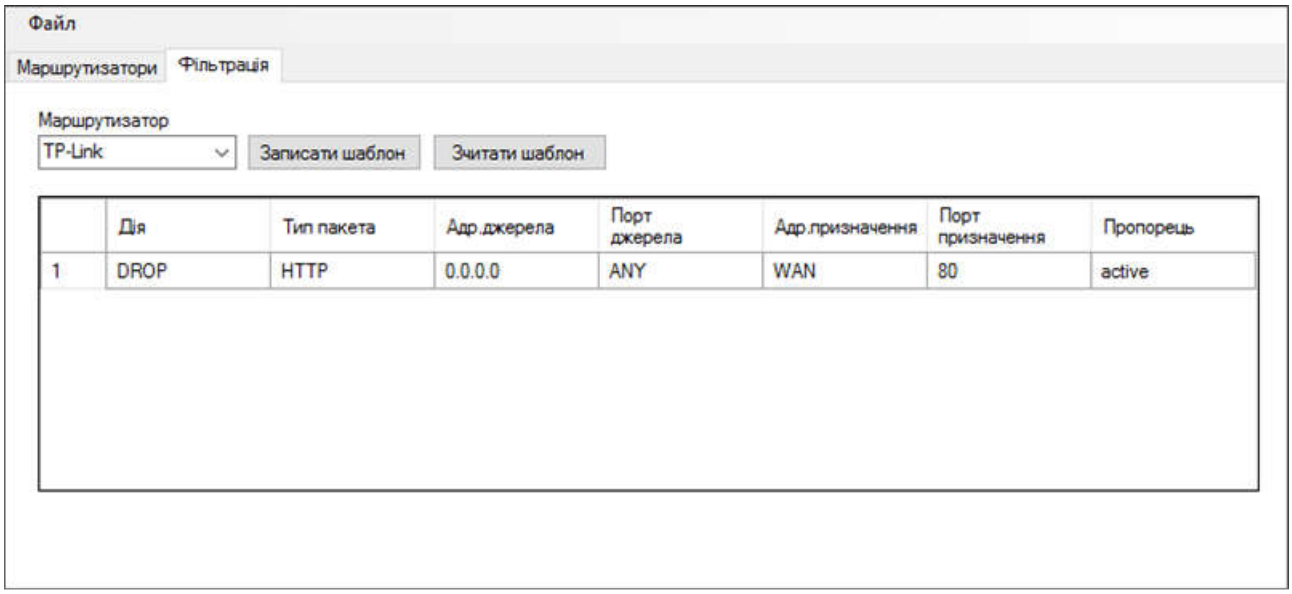

Рисунок 3.6 – Екран правил фільтрації

Тепер я опишу, як відбувається ситуація коли користувач виконує настройку безпеки. Користувач програми, нажимає на кнопку зчитати шаблон і отримує перелік правил фільтрації із маршрутизатора. Якщо необхідно, користувач редагує певні правила. Після завершення редагування всіх правил і перевірки на помилки, користувач нажавши на кнопку записати шаблон, здійснює запис конфігурації правил в маршрутизатор. При виконанні всіх цих дій, виконується автоматизована перевірка на помилки та авторизація на маршрутизатор.

Програмний код даного додатку наведений у додатку А. А в цьому розділі я описав вибір мови та середовище програмування, описав структуру графічного інтерфейсу програми та навів усі кроки створення програмного додатку відповідно до умов технічного завдання.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

4.1 Розрахунок витрат на розробку програмного забезпечення

В даному розділі дипломного проекту проводиться економічне обґрунтування доцільності розробки програмного забезпечення. Зокрема, здійснюється розрахунок витрат на розробку програмного забезпечення, експлуатаційних витрат, ціни споживання проектного рішення. В заключній частині визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблене програмне забезпечення призначене для керування маршрутизаторами через веб-інтерфейс, що призводить до ефективного управління правилами фільтрації пакетів.

Витрати на розробку і впровадження програмних засобів (К) включають:

$$K = K_1, + K_2,$$

де K_1 - витрати на розробку програмних засобів, грн;

K_2 - витрати на відлагодження і дослідну експлуатацію програмного рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ($B_{ОП}$);
- витрати на відрахування у спеціальні державні фонди ($Вф$);
- витрати на покупні вироби ($Пв$);
- витрати на придбання спец-обладнання для проведення експериментальних робіт ($Об$);
- накладні витрати ($Н$);
- інші витрати ($Ів$).

Розрахунок витрат на оплату праці.

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП

обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу.

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, ст.викладач	5000
2	Консультант техніко-економічного розділу, доцент	6027
3	Студент	1100

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij} ,$$

де n_{ij} – чисельність розробників *i*-ої спеціальності *j*-го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником *i*-ої спеціальності *j*-го тарифного розряду, год;

C_{ij} – годинна ставка працівника *i*-ої спеціальності *j*-го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^{0}(1+h)}{РЧ_{i}},$$

де C_{ij} – основна місячна заробітна плата розробника *i*-ої спеціальності *j*-го тарифного розряду, *грн*.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати 0,47 (при умові наявності доплат);

$РЧ_i$ - місячний фонд робочого часу працівника *i*-ої спеціальності *j*-го тарифного розряду, год. (приймаємо 168 год.).

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

$$B_{Ф} = \frac{20,5}{100} \cdot 2100,84 = 430,67 \text{ грн.}$$

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, ст. викладач	20,5	29,76	610,08
2	Консультант техніко-економічного розділу, доцент	2	52,74	105,48
3	Студент	144	9,62	1385,28
Разом				2100,84

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них. Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 4,5 грн. Середній щоденний час роботи на комп'ютері – 2 години.

Таблиця 4.3- Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	уп	50	2	100	10	110
2	Ручка кулькова	шт	6	2	12	1,2	13,2
3	Олівець простий	шт	3,5	1	3,5	0,35	3,85
4	Флешка USB	шт	168	1	168	16,8	184,8
5	Зошит, 96 арк	шт	11	1	11	1,1	12,1
6	Тонер для принтера	уп	50	1	50	5	55
Разом							378,95

Розрахунок витрат на використання комп’ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп’ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп’ютер	Час використання комп’ютера, год.	Витрати на використання комп’ютера, грн.
1	Проведення досліджень та оформлення їх результатів	75	337,5
2	Оформлення техніко-економічного розділу	3	13,5
4	Оформлення ДП	16	72
Разом		94	423

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1{,}5 * 2100{,}84 = 3151{,}26 \text{ грн.}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 0{,}1 * 2100{,}84 = 210{,}08 \text{ грн.}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{ОП} + B_Ф + B_{ПВ} + H + I$$

$$K_1 = 2100{,}84 + 430{,}67 + 378{,}95 + 3151{,}26 + 210{,}08 = 6271{,}8 \text{ грн.}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від},$$

де $S_{м.г.}$ - вартість однієї машино-години роботи ПК, грн./год;

$t_{від}$ - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,32 грн. Тому

$$К_2 = 2{,}5 \cdot 70 = 175 \text{ грн.}$$

Таблиця 4.5 - Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, *грн*.
1	Витрати на оплату праці	2100,84
2	Відрахування у спеціальні державні фонди	430,67
3	Витрати на куповані вироби	378,95
4	Накладні витрати	3151,26
5	Інші витрати	210,08
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	175
	Разом	6446,8

Визначення експлуатаційних витрат. Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\text{П}} = E_{1\text{П}} + E_{2\text{П}},$$

де E_n - одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

E_{1n} - вартість підготовки даних для експлуатації ПЗ (аналогу), грн.;

E_{2n} - вартість роботи комп'ютера для виконання проектного рішення (аналогу), грн.

Річні експлуатаційні витрати $B_{еп}$ визначаються за формулою:

$$B_{E\text{П}} = E_{\text{П}} * N_{\text{П}},$$

де N_n - періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1П} = \sum_{l=1}^{n} n_i t_i c_i \ ,$$

де i - категорії працівників, які приймають участь у підготовці даних (i=1,2,...n);

n_i - кількість працівників і-ої категорії, *осіб*.;

t_i - трудомісткість роботи співробітників і-ої категорії по підготовці даних, год.;

c_i - середнього годинна ставка працівника і-ої категорії з врахуванням додаткової заробітної плати.

Середньо годинна ставка працівниказнаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m},$$

де c_i^0 - основна місячна заробітна плата працівника і-ої категорії, грн.;

b - коефіцієнт, який враховує додаткову заробітну плату 0,57;

m - кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: с° = 1200 грн. Тоді:

$$c_1 = \frac{1200(1+0,57)}{22*8} = 10,7 \text{ грн/год.}$$

Трудомісткість підготовки даних для проектного рішення складає 1 год., для аналога 1,7 год.

Таблиця 4.6 - Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати , грн.
	Проектне рішення		
1	1	10,7	10,7
	Аналог		
1	1,7	10,7	18,19

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ},$$

де t - витрати машинного часу для реалізації проектного рішення (аналогу), год.;

$S_{МГ}$ - вартість однієї години роботи комп'ютера, грн./год.

$E_п$ =10,7 + 2,5 = 13,2 грн.; $E_а$ = 18,19 + 4,25 = 22,44 грн.

$B_{еп}$ = 13,2 * 255 = 3366 грн.; $B_{еа}$ =2 2,44 * 255 = 5722,2 грн.

$E_{2п}$ = 1 * 2,5 = 2,5 грн.; $E_{2а}$ = 1,7 * 2,5 = 4,25 грн.

4.2 Розрахунок ціни споживання проектного рішення

Ціна споживання - це витрати на придбання і експлуатацію проектного рішення за весь строк його служби:

$$Ц_{С(П)} = Ц_П + B_{(E)NPV},$$

де $Ц_п$ - ціна придбання проектного рішення, *грн.*

$$Ц_П = К(1 + \frac{П_Р}{100}) + К_0 + К_к ,$$

де $К$ - кошторисна вартість;

$Пр$ - рентабельність;

$К_о$ - витрати на прив'язку та освоєння проектного рішення на конкретному об'єкті, грн.;

$К_к$ - витрати на доукомплектування технічних засобів на об'єкті, грн..

$$Ц_Д = 6446,8 \cdot (1 + 0,3) = 8380,84 \text{ (грн.)}.$$

Вартість витрат на експлуатацію проектного рішення (за весь час його експлуатації), грн.:

$$B_{епрv} = \sum_{t=0}^{T} \frac{B_{еП}}{(1+R)^t} ,$$

де $B_{еп}$ - річні експлуатаційні витрати, грн.;

T - строк служби проектного рішення, років;

R - річна ставка проценту банку.

$$B_{епрv} = \sum_{t=1}^{5} \frac{3366}{(1+0,08)^t} = 15583,3 \text{ грн.}$$

$$B_{епрv} = \sum_{t=1}^{5} \frac{5722,2}{(1+0,08)^t} = 26491,65 \text{ грн.}$$

Тоді ціна споживання проектного рішення дорівнюватиме:

$$Ц_{сп} = 8380,84 + 15583,3 = 23964,14 \text{ грн.}$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{са} = 3500{,}0 + 26491{,}65 = 29991{,}65 \text{ грн.}$$

4.3 Визначення показників економічної ефективності

Економічний ефект в сфері проектування рішення:

$$E_{ПР} = Ц_{П} - Ц_{А}$$

$$E_{ПР} = 8380{,}84 - 3500{,}0 = 4880{,}84 \text{ грн.}$$

Річний економічний ефект в сфері експлуатації:

$$E_{КС} = B_{ЕА} - B_{ЕП}$$

$$E_{кс} = 5722{,}2 - 3366 = 2356{,}2 \text{ грн.}$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{екс} = \sum_{t=1}^{T} E_{екс}(1+R)^{T-t}$$

$$\Delta E_{екс} = \sum_{t=1}^{5} 2356{,}2(1+0{,}08)^{5-t} = 12723{,}48 \text{ грн.}$$

Сумарний ефект складає:

$$E = E_{пр} + \Delta E_{екс} = 4880{,}84 + 12723{,}48 = 17604{,}32 \text{ грн.}$$

В таблиці 4.7 зведені всі загальні показники економічної ефективності даного програмного засобу.

Таблиця 4.7 - Показники економічної ефективності проектного рішення

№	Найменування	Одиниці вимірювання	Значення показників	
			Базовий варіант	Новий варіант
1	Капітальні вкладення	грн.	-	3350,44
2	Ціна придбання	грн.	3500,0	8380,84
3	Річні експлуатаційні витрати	грн.	26491,65	15583,3
4	Ціна споживання	грн.	29991,65	23964,14
5	Економічний ефект в сфері проектування	грн.	-	4880,84
6	Економічний ефект в сфері експлуатації	грн.	-	2356,2
7	Додатковий ефект в сфері експлуатації	грн.	-	12723,48
8	Сумарний ефект	грн.	17604,32	

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Здійснено порівняння з існуючим аналогом, і цим показано, що дане проектне рішення має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, гнучкість, зручність. Згідно проведеного економічного обґрунтування дане проектне рішення є конкурентноздатним. Крім того, отримано сумарний економічний ефект у розмірі 17604,32 грн. і тому розробка і впровадження цього проектного рішення є економічно доцільними.

ВИСНОВКИ

1. Досліджено виконання маршрутизації пакетів в комп'ютерних мережах. Розглянуто основні технології та види маршрутизації маршрутизації. Проаналізовано перелік протоколів які виконуються на мережевому рівні моделі TCP/IP та виділено переваги і недоліки кожного з них.

2. Досліджено засоби розгортання вбудованого програмного забезпечення пристроїв. Розглянуто програмні застосування від різних виробників мережевого устаткування та на їх основі виділено головні функції програмного забезпечення для маршрутизаторів. Описано об’єкти та процедури розроблюваного особистого програмного засобу.

3. Визначено основні принципи виконання маршрутизації в програмних засобах маршрутизаторів, від існуючих на ринку виробників. Проаналізовано структуру і основні принципи правил фільтрації пакетів в програмних засобах. Описано структуру особистого програмного засобу та представлено його об’єктну модель.

4. Вибрано структуру даних для реалізації правил фільтрації у програмному засобі. Описано формат проходження пакетів при маршрутизації та складено алгоритм роботи засобу. Здійснено розроблення екранних форм графічного інтерфейсу програмного засобу. Описано етапи зчитування, редагування, зберігання правил фільтрації та запису їх на маршрутизатор. Виконано автоматичну перевірку коректності та авторизації на пристрій.

5. Для виконання мети розробки засобу, було обрано мову Python. Також для здійснення розробки графічного інтерфейсу програми, використовувалася бібліотека Qt. Відповідно було розроблено сам графічний інтерфейс. А також здійснено тестування програмного засобу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Развертывание приложений, служб и компонентов - [Електронний ресурс] Режим доступу - https://msdn.microsoft.com/uk-ua/library/wtzawcsz.aspx – Назва з екране.
2. TP Link. Руководство пользователя TL-WR841N TL-WR841ND. Беспроводной маршрутизатор серии N со скоростью передачи данных до 300 Мбит/с - [Електронний ресурс] Режим доступу - http://sevstar.net/uploads/5613cd8d92c41.pdf
3. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы. 4 изд. / В.Г. Олифер, Н.А. Олифер – СПб: Питер, 2010. – 944 с.
4. McCabe J. Network Analysis, Architecture, and Design. Third edition. / James D. McCabe - Morgan Kaufmann, 2007 – 495 p.
5. Яковина В.С. Основи безпеки комп’ютерних мереж: Навчальний посібник / За ред. Д.В. Федасюка. – Львів: НВФ "Українські технології", 2008. – 396 с.
6. Семенов А. Б. Структурированные кабельные системы. 4-е изд./ Семенов А. Б., Стрижаков С. К., Сунчелей И. Р. - М.: ДМК-Пресс, 2002. - 640 с.
7. Виденье отказоустойчивой, надежной, масштабируемой сети передачи данных - [Електронний ресурс] Режим доступу - http://habrahabr.ru/ blogs/personal/93629/
8. Хелеби С. Принципы маршрутизации в Internet, 2-е издание./ Хелеби С. Мак-Ферсон Д. Пер. с англ. - М.: "Вильямс", 2001. - 448 с.
9. Документація з настройки обладнання фірми Cisco. [Електронний ресурс]: Режим доступу http://www.cisco.com
10. Чекмарев А. Windows 7. Руководство администратора. - Спб.: БХВ-Петербург, 2010 – 896 с.
11. Визерспун Д. Освой самостоятельно LINUX за 24 часа, 3-е издание : - М.: Издательский дом "Вильямс", 2001.- 352 с.

12. Жуматий С.А. Программная среда поддержки эффективного выполнения задач на параллельных вычислительных системах [Текст] / С.А. Жуматий – М.:МГУ им М.В. Ломоносова, 2005. - 95 с.
13. Корнеев В. В. Параллельные вычислительные системы [Текст] / В. В. Корнеев - М.: Нолидж, 1999. - 320 с.
14. Трофимов С. А. CASE-технологии: Практическая работа в Rational Rose [Текст] / С. А. Трофимов - М.: БИНОМ, 2002. - 288 с.
15. Урман С. Oracle 9i. Программирование на языке PL/SQL [Текст] / С. Урман - М.:Лори, 2004. - 544 с.
16. Штайнер Г. HTML/XML/CSS. Справочник [Текст] / Г. Штайнер - М: Лаборатория базовых знаний, 2001. – 512 с.
17. Хахаев И. А. - Практикум по алгоритмизации и программированию на Python - Альт Линукс, 2010 – 126 с.
18. Лутц М. Программирование на Python. Том 2, 4-е издание – Символ-Плюс, 2011 – 992 с.
19. Н.А. Прохоренок PyQt. Создание оконных приложений на Python 3 – 2011 – 243 с.
20. Гифт Н. Python в системном администрировании - O'Reilly, 2009 – 511 с.
21. Лутц М. - Изучаем Python - O'Reilly, 2011 – 1280 с.
22. Головатый А. Django. Подробное руководство / Головатый А., Каплан-Мосс Дж. – Символ,2010 – 552 с.
23. Бизли Д. Python. Подробный справочник – Символ,2010 – 500 с.
24. Саммерфилд М. Программирование на Python 3 – Символ,2009 -608 с.
25. Сузи Р.А. Язык программирования Python - Бином-пресс – 300 с.
26. Уэсли Дж. Чан Python. Создание приложений - Вильямс, 2016 – 816 с.
27. Свейгарт Э. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих - Вильямс, 2016 – 592 с.
28. Шоттс У. Командная строка Linux. Полное руководство – Питер, 2017 -480 с.
29. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения – Питер, 2017 – 496 с.

30. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. - Тернопіль: ТНЕУ, 2013.– 65с.
31. Паздрій І.Р. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» – Тернопіль: Економічна думка 2014.- 36 с.