Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод автоматизованої побудови VPN-ланцюгів на платформі IaaS

(тема)

Виконав:
студент II курсу, групи СПм-19-1
Будько А.О.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ткачов В.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Будько Анні Олексіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Метод автоматизованої побудови VPN-ланцюгів на платформі IaaS

затверджена наказом по університету від " 30 " жовтня 2020 р. № 1486Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вхідні дані до роботи обчислювальна станція SR-905 – 4 шт.;
фрагмент комп'ютерної мережі кафедри ЕОМ;
програмне середовище Cisco Packet Tracert;
програмне середовище Network Simulator;
програмне середовище Graphical Network Simulator 3;
OpenVPN сервер.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ
1 Особливості побудови VPN-тунелів з використанням хмарних технологій
2 Розробка методу автоматизованої побудови VPN-ланцюгів на платформі IaaS
3 Модель динамічної маршрутизації в VPN-ланцюгах в умовах неповних даних IaaS-інфраструктури
4 Дослідження динаміки побудови маршрутів VPN-ланцюгів в умовах неповних даних IaaS-інфраструктури
Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційні матеріали. Слайди – 13 арк. ф. А4

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд технологій побудови VPN-тунелів з	03.11.20-09.11.20	
	використанням хмарних технологій		
2	Вибір та обгрунтування методики дослідження	10.11.20-17.11.20	
3	Вибір інструментальних засобів	18.11.20-23.11.20	
4	Проведення експериментів	24.11.20-01.12.20	
5	Оформлення матеріалів атестаційної роботи	02.12.20-07.12.20	
6	Подання атестаційної роботи керівникові та	08.12.20-09.12.20	
	її попередній захист		
7	Подання атестаційної роботи на рецензування	10.12.20-11.12.20	

Дата видачі завдання 02 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ткачов В.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 75 с., 12 рис., 2 табл., 2 дод., 17 джерел.

VPN, IAAS, ПЕРЕДАЧА ДАНИХ, ПРОГРАМНИЙ ЗАСІБ, РОЗПОДІЛ ПОТОКІВ ДАНИХ.

Метою атестаційної роботи є підвищення мережної безпеки шляхом розробки алгоритму автоматизованої побудови VPN-ланцюгів на платформі IaaS для побудови високозахищеної мережі на базі технології VPN.

У ході виконання атестаційної роботи проаналізовано особливості побудови VPN-тунелів з використанням хмарних технологій, розроблено новий метод автоматизованої побудови VPN-ланцюгів на платформі IaaS, створено модель динамічного VPN-тунелювання в умовах неповних даних IaaS та виконано дослідження динаміки побудови маршрутів VPN-ланцюгів. За результатами роботи зроблено висновки та рекомендації щодо застосування розробленого методу на рівні промислової експлуатації.

ABSTRACT

Master’s thesis: 75 pages, 12 figures, 2 tables, 2 appendices, 17 sources.

VPN, IAAS, DATA TRANSMISSION, SOFTWARE, DISTRIBUTION OF DATA FLOWS.

The major goal of this thesis is to increase network security by developing an algorithm for automated construction of VPN-circuits on the IaaS platform to build a highly secure network based on VPN technology.

In order to peculiarities of construction of VPN-tunnels with the use of cloud technologies were analyzed, a new method of automated construction of VPN-chains on IaaS platform was developed, a model of dynamic VPN-tunneling in conditions of incomplete IaaS data was created and the dynamics of VPN-chain routes construction was studied. Based on the results of the work, conclusions and recommendations were made on the application of the developed method at the level of industrial operation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОЕМ – область ефективної маршрутизації

ОКМ – оверлейні комп’ютерні мережі

IPSec – IP Security (англ., протокол для захисту даних, що передаються по протоколу IP)

L2TP – Layer 2 Tunneling Protocol (англ., протокол тунелювання другого рівня)

NDF – network delay fluctuations (англ., коливання мережної затримки)

PER – packet error (англ., коефіцієнт помилок в пакетах)

PLR – packet loss ratio (англ., коефіцієнт втрат пакетів)

PPP – Point-to-Point Protocol (англ., протокол точка-точка)

PPPoE – Point-to-point protocol over Ethernet (англ., протокол точка-точка в Ethernet)

PPTP – Point-to-Point Tunneling Protocol (англ., тунельний протокол точка-точка

SSL – Secure Sockets Layer (англ., шар захищених сокетів)

VPN – Virtual Private Network (англ., віртуальна приватна мережа)

WebRTC – Web Real Time Communications (англ., комунікація в реальному часі)

ВСТУП

За останні п’ять років значного поширення набули технології хмарних обчислень. Вони займають потужну нішу майже у всіх IT-процесах. Нерідко мають свій вплив і на критичні інфраструктури. Все це дозволяє виносити бізнес-майданчики саме у віртуальну площину, вибудовуючи термінальні мережі. Міжнародні бізнес-групи все частіше використовують в якості робочих станцій термінальні системи. При цьому сервери віртуальних машин часто розміщені в хмарній інфраструктурі, в той час як віддалені офіси змушені підключатися до них за допомогою терміналів. В даний час безпеку систем термінального доступу забезпечується за рахунок стійкості протоколів обміну даними між терміналом і хостом.

Для підвищення безпеки таких з'єднань, з недавнього часу використовується концепція VPN-ланцюгів. Її суть полягає у використанні сукупності проміжних VPN-серверів, взаємодію яких організовано за певним правилом з метою багаторазового «перепакування» трафіку між підмережами або за рахунок наскрізного тунелювання. При цьому важливими завданнями є швидкість побудови такої оверлейной структури і підтримка в межах допустимих значень показника затримки між кінцевими вузлами.

Тому на сьогодні актуальною є задача швидкої побудови VPN-ланцюгів з заданим рівнем безпеки і умовою мінімально допустимої мережевий затримки. Однак використання віртуальних інфраструктур, наприклад як IaaS дозволяє створювати платформонезалежні універсальні рішення.

В атестаційній роботі запропонований метод автоматизованої побудови динамічної мережі на базі VPN-ланцюгів з використанням IaaS-інфраструктури для забезпечення безпеки систем термінального доступу. В основі методу лежить рішення комбінаторних задач найкращого вибору.

Метод передбачає як планову перебудову маршруту в ланцюжку, так і

позапланову, що виникає через відмову VPN-сервера або перевищенні порогової мережевий затримки між терміналом і хостом.

У роботі також ставиться задача розробки моделі такої оверлейної інфраструктури шляхом рохробки принципів марщрутизації в статичних та динамічних сегментах мережі, визначення області ефективної маршрутизації, розрахунку показників навантаження IaaS-вузлів, оцінки динамічних показників, формування зовнішніх маршрутних схем точок входу та виходу в ланцюги.

Окремим питанням стоїть розробка алгоритмічного забезпечення маршрутизації в ланцюгах на основі неповних даних, зокрема схеми відновлення при відмові вузлів в IaaS-інфраструктурі, механізми оновлення зовнішніх маршрутних даних, врахування вимог до передачі службового трафіку.

У зв’язку з коронавірусними обмеженнями та унеможливленням проведення ряду прикладних експериментів, в атестаційній роботі поставлена задача провести числових експерименти для оцінки ефективності запропонованих рішень.

1 ОСОБЛИВОСТІ ПОБУДОВИ VPN-ТУНЕЛІВ З ВИКОРИСТАННЯМ ХМАРНИХ ТЕХНОЛОГІЙ

1.1 Аналіз сучасних технологій VPN-тунелювання

В сучасному IT-просторі існує велика кількість технологій побудови VPN-тунелів, які в загальному випадку створюють оверлейні комп'ютерні мережі (рисунок 1.1).

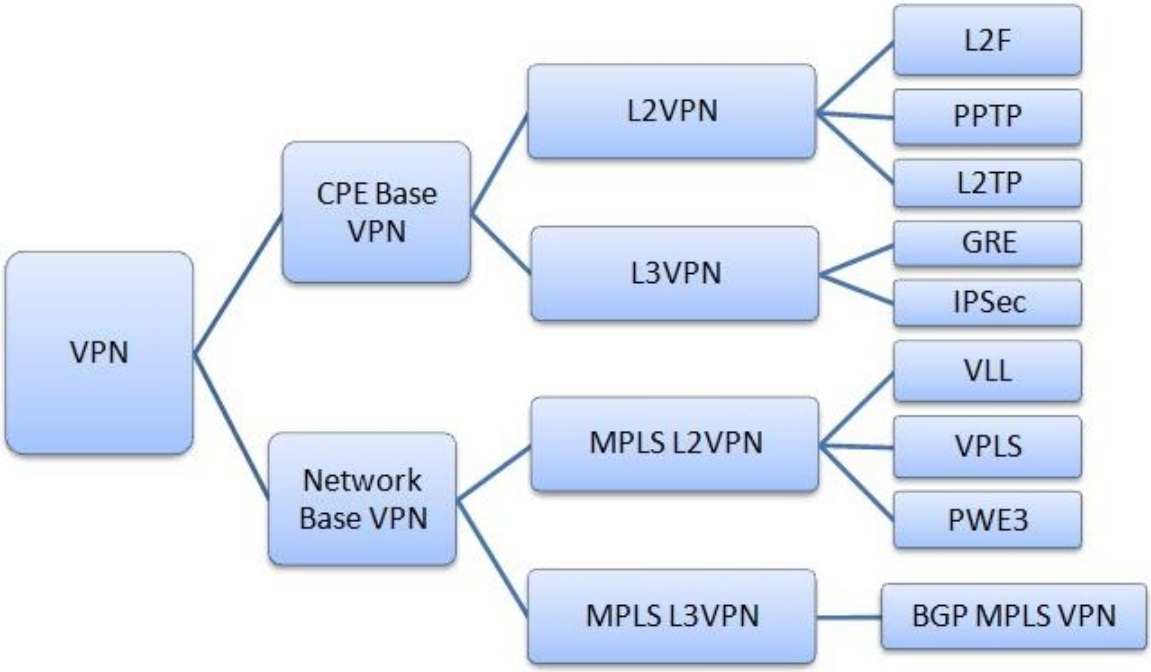

Рисунок 1.1 – Класифікація технологій побудови VPN-тунелів

VPN-тунелювання реалізується завдяки використанню специфічних мережних протоколів. Два основних протоколи, які використовуються для тунелювання VPN, – це PPTP і L2TP. Для протоколу PPTP і L2TP тунель схожий на сеанс. Обидві кінцеві точки тунелю повинні погоджуватися з

тунелем і повинні погоджувати змінні конфігурації, такі як призначення адреси, шифрування або параметри стиснення. У більшості випадків дані, що передаються всередині тунелю, відправляються з використанням протоколу на основі такої структури даних як дейтаграми. Протокол управління тунелем використовується як механізм створення, обслуговування і завершення тунелю. Після того, як тунель встановлено, дані можуть бути відправлені [1-5].

Два основних типи тунелювання VPN – це добровільне і примусове. При добровільному тунелюванні клієнтський комп'ютер може видати запит VPN для налаштування і створення добровільного тунелю. У цьому випадку комп'ютер користувача є кінцевою точкою тунелю і діє як клієнт тунелю. Добровільне тунелювання вимагає, щоб на клієнтському комп'ютері було попередньо встановлено відповідне програмне забезпечення і протоколи.

Якщо проаналізувати другий випадок, то при примусовому тунелюванні вузол віддаленого доступу з підтримкою VPN налаштовує і створює примусовий тунель. При використанні обов'язкового тунелю комп'ютер користувача не є кінцевою точкою тунелю. Іншим пристроєм, сервером віддаленого доступу, між комп'ютером користувача і тунельним сервером є кінцева точка тунелю і діє як клієнт тунелю.

Доволі цікавим напрямком побудови VPN-тунелів є використання у якості синергії інших технологій віртуалізації. Доцільним є провести аналіз цих технологій для розуміння роботи гібридних схем. До таких технологій відноситься, наприклад, використання проксі-вузлів. При використанні проксі-вузла запит від клієнтського комп’ютера до веб-ресурсу спочатку надходить на проксі-сервер, потім обробляється і вже після надходить на потрібний веб-сервер. Відповідь веб-сервера спочатку надходить на проксі-сервер і перенаправляється на ПК користувача [6-8].

Під час побудови віртуальної мережі замість одного проксі-сервера між комп’ютером користувача і віддаленим інформаційним ресурсом використовують мережу проксі-вузлів. Проксі-вузли класифікують за

протоколом, по якому вони передають трафік: HTTP/HTTPS, FTP, CGI, SOCKS або анонімайзер.

В даний час існує декілька готових рішень веб-проксі-вузлів, до яких відносяться: CGIProxy (на базі CGI-скриптів і OpenSSL), Glype Proxy (на базі PHP), PHProxy, Zelune (на базі PHP); Cohula (на базі Java).

Веб-проксі, який виступає в якості буфера між користувачем та інформаційними ресурсами мережі Інтернет, дозволяє частково зменшити задачу анонімного доступу до ресурсів або обійти обмеження локальної мережі користувача, однак він досить вразливий до виявлення та блокування як з боку адміністраторів локальної мережі користувача, так і з боку ресурсів мережі Інтернет.

Крім проксі-вузлів, існує інша гібридна технологія віртуалізації така як багатошарове, або вкладене, VPN-тунелювання. Таке VPN-тунелювання, на відміну, від традиційної технології, це процес, при якому приватні пакети даних відправляються від відправника до одержувача через існуючі тунелі анонімних мереж. При використанні такого VPN-тунелювання клієнтський комп'ютер фактично стає вузлом віддаленої (корпоративної) локальної мережі і стає невидимим у локальній мережі користувача; схема вузла служить шлюзом служб доменних імен (DNS) для клієнта і нічого не знає про локальну мережу користувача. Однак користувачі можуть визначити статичні маршрути на своїх комп'ютерах, щоб продовжити доступ до локальної мережі при одночасному підключенні до віддаленої віртуальної мережі.

Ще одна технологія побудови віртуальних мереж під назвою VLAN, дозволяє будувати віртуальні мережі з незалежної від фізичних пристроїв топології. Наприклад, можна об'єднати в одну віртуальну мережу відділ компанії, співробітники якого працюють у різних будинках і підключені до різних комутаторів, або є відділеними клієнтами та мають власні віртуальні тунелі, або навпаки, створити окремі мережі для пристроїв, підключених до одного комутатора, якщо це необхідно для забезпечення безпеки [9-11].

Цікаво, що спеціалізовані віртуальні комутатори дозволяють саме таким чином комутувати всі підключення по віртуальним підмережам VPN-сервера.

В основі технології VLAN лежить загальновизнаний стандарт IEEE 802.1Q. Він дозволяє додавати в мережний трафік інформацію про приналежність переданих даних до тієї чи іншої віртуальної мережі – теги VLAN. З їх допомогою комутатори і маршрутизатори, у тому числі і віртуальні, можуть виділити із загального потоку переданих по мережі кадрів ті, що відносяться до конкретного сегменту.

Ця технологія дає можливість організувати функціональний еквівалент декількох LAN-мереж без використання набору багатьох VPN-серверів, які знадобилися б для їх реалізації у такому вигляді. Все мережне обладнання замінюється віртуальним. Найбільш відомі виробники таких рішень ставлять акцент на те, щоб їх комунікаційне обладнання використовувало VLAN для розподілу потоків даних, бо це дозволяє уникати колізії і мінімізувати кількість DOS-атак.

На відміну від попередніх технологій існує новий рівень абстракції – технологія побудови віртуальної інфраструктури в хмарі (IaaS). Віртуальні приватні хмари є налаштовуваним пулом загальних обчислювальних ресурсів, виділених в рамках публічного хмарного середовища, забезпечуючи певний рівень ізоляції між різними користувачами, використовуючи ресурси. Такі хмари реалізують і віртуальну комутацію, маршрутизацію і брандмаунінг інфраструктури. Використання загальнодоступного хмарного середовища надає бізнесу такі переваги, як гнучкість, масштабованість та зниження IT-витрат [12-15].

Провайдер хмарних послуг повинен підтримувати всі процеси, що виконуються на серверах, в віртуальній мережі. Ізоляція між одним користувачем і всіма іншими користувачами тієї ж хмари (іншими користувачами, а також іншими користувачами загальнодоступної хмари) зазвичай досягається за рахунок виділення приватної IP-підмережі і

конструкції віртуального зв'язку (такий як VLAN або набір зашифрованих каналів зв'язку) на користувача. У хмарній інфраструктурі використовується описаний механізм, що забезпечує ізоляцію в хмарі, супроводжується функцією VPN, яка захищає за допомогою автентифікації і шифрування віддаленого доступу. Клієнт хмарної інфраструктури підключається через VPN-тунель до свого робочого місця, тому дані, що передаються в віртуального робочого місця, не видно іншим користувачам загальнодоступної хмари.

Повноцінна хмарна інфраструктура (IaaS) – передбачає надання розфасованої керованої інфраструктури у вигляді екземплярів віртуального сервера. Ці віртуальні екземпляри включають компоненти різного розміру або ємності, такі як пам'ять, оперативна пам'ять та обчислювальна потужність процесора, а також бажану операційну систему сервера (наприклад, Windows або Linux), засоби комунікації між віртуальними областями.

Ці послуги дозволяють організаціям розгортати і керувати своїми власними інфраструктурними рішеннями на віртуалізованих серверах, які розміщуються за межами фізичної інфраструктури організації, – в центрі обробки даних постачальника послуг, що дозволяє клієнтам зосередитися на управлінні робочими навантаженнями своїх додатків без необхідності підтримувати базову інфраструктуру сервера. Оскільки такі рішення розглядаються як обчислювальне, мережне та сховище IaaS, яке експлуатується постачальником послуг для однієї організації, що воно є підмножиною більшої хмарної інфраструктури (наприклад, загальнодоступної хмари).

Використання віртуальної приватної хмари як IT-рішення має багато переваг. Це економічно вигідне рішення, яке забезпечує підприємствам необхідну безпеку без дорогої інфраструктури. Переваги віртуальної приватної хмари наступні:

- масштабованість – дозволяє адміністратору IaaS додавати ресурси на

вимогу для розміщення додаткових користувачів;

- ефективність – покращена продуктивність є прямим результатом масштабованості та компонентів автоматизації – ресурси доступні на вимогу, а інфраструктура, необхідна для підтримки цих ресурсів, завжди доступна;

- автоматизація – видалення ручних процесів для динамічного забезпечення ресурсами віртуальних машин;

- безпека – вхідний та вихідний трафік хмарної інфраструктури залишається в корпоративному хмарному брандмауері. Хмарні адміністратори можуть встановлювати політики, щоб вказати, яким користувачам дозволено отримувати доступ до хмарних ресурсів.

- гонтроль – вся віртуальна приватна хмара дозволяє постачальнику послуг мати повний контроль над даними, що зберігаються у загальнодоступній хмарі, а також над вхідним та вихідним мережним трафіком.

1.2 Особливості застосування хмарних технологій при побудові VPN-тунелів

Типові мережі з застосуванням VPN-тунелів в основному зосереджуються на забезпеченні сильної анонімності за ціну меншої пропускної здатності, вищої затримки та погіршення зручності використання з обмеженою підтримкою маршрутизації. Вони також часто анонімують лише кілька конкретних програм.

Наприклад, оснує підхід до побудови анонімної мережі, в якому мережа складається з оверлейної мережі з великою кількістю VPN-тунелів, яка забезпечує анонімність усіх програм, що працюють поверх неї, і протоколів маршрутизації, які можна розглядати як анонімізовану версію маршрутизації векторних шляхів. Протокол зберігає високоефективні характеристики маршрутизації вектора шляхів, а також має додаткову перевагу, приховуючи топологію накладеної мережі (рисунок 1.2).

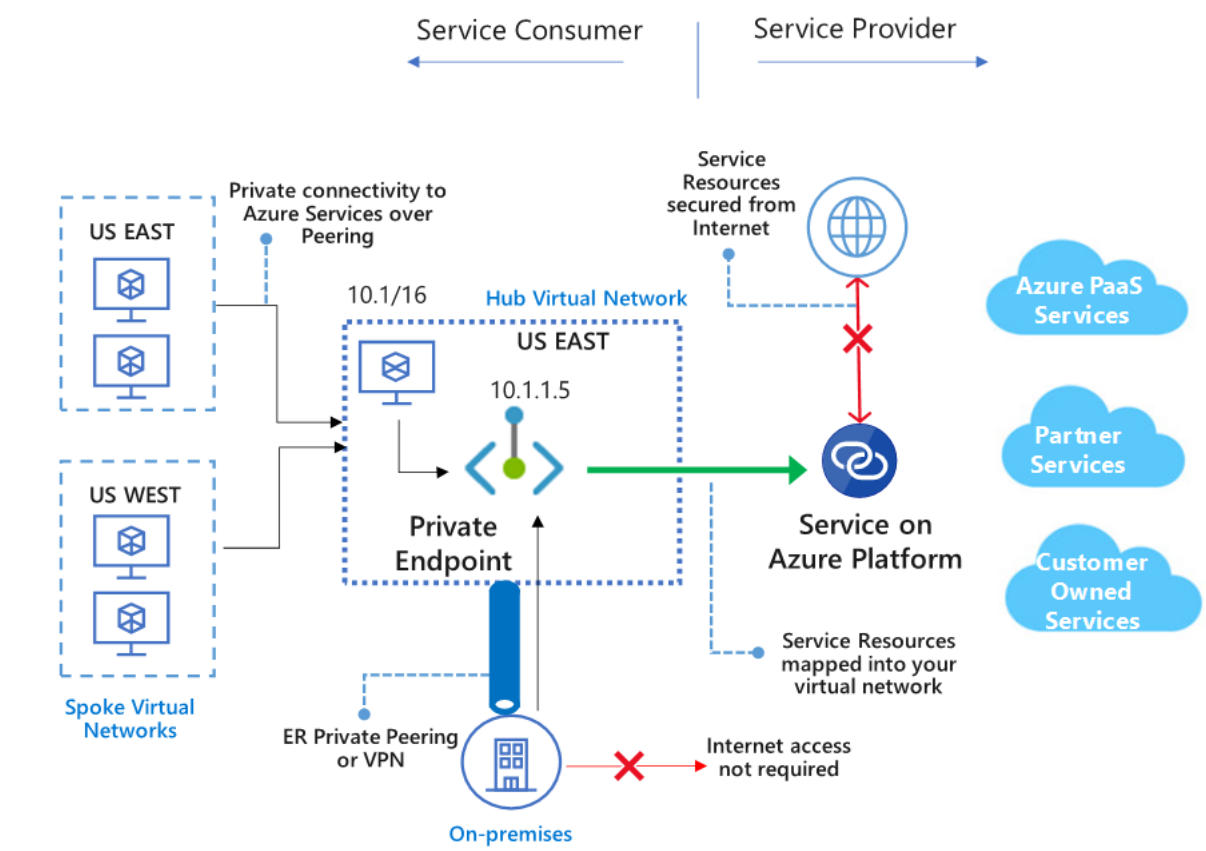

Рисунок 1.2 – Приклад оверлейної мережі з хмарними вузлами сервісів Azure

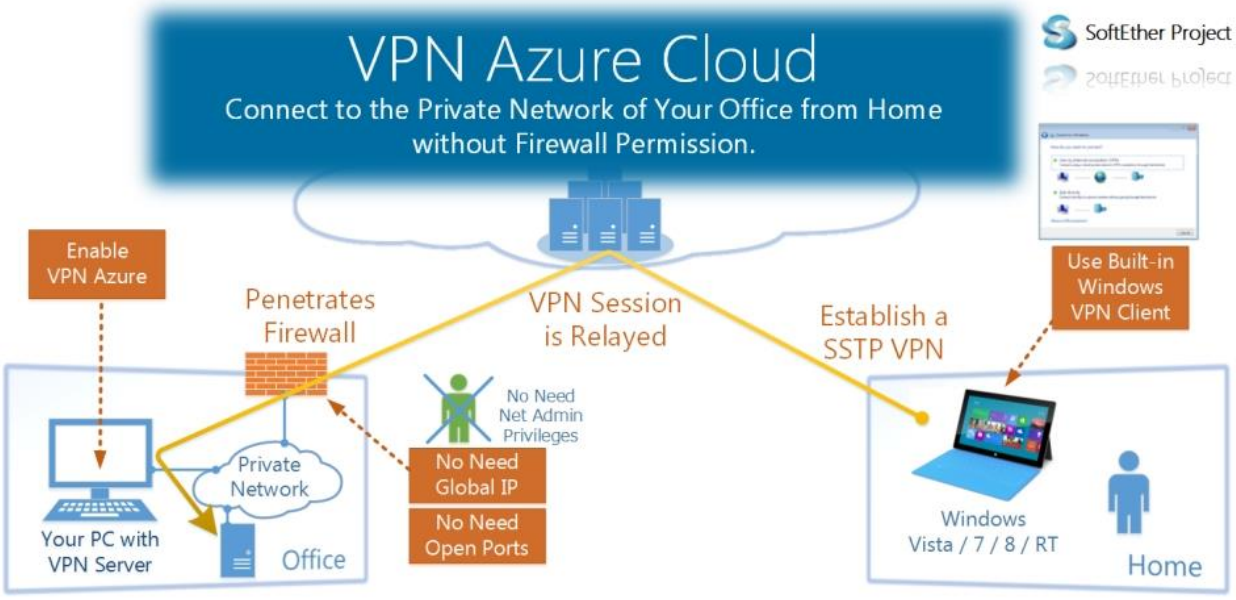

Рисунок 1.3 – Приклад схеми взаємодії компонентів оверлейної мережі з хмарними серісами

Оверлейна мережа запропонована в роботі [5] роз'єднує фактичні IP-адреси вузлів та віртуальні адреси, які вузли використовують у реальних додатках. Для цього використовуються специфічні віртуальні адреси, щоб анонімізувати хости та фізичну IP-адресу для ефективної маршрутизації. Віртуальні адреси також можуть бути динамічними для подальшого підвищення надійності вузлів (рисунок 1.3).

Традиційні протоколи маршрутизації передають інформацію про топологію мережі до вузлів, тоді як існуючі протоколи маршрутизації в віртуальних мережах на хмарних платформах не забезпечують автентифікацію для інформації про маршрутизацію. Шкідливий вузол може довільно зменшити значення вартості шляху, яке передається в анонімному повідомленні про оголошення маршруту, з метою негативного впливу на ефективність віртуальної маршрутизації або сприяння запуску різних атак, таких як підслуховування або атаки «людина посередині» (рисунок 1.4).

В рамках теорії функціонування тимчасових розподілених систем проблема маршрутизації в віртуальних каналах в структурованих оверлейних мережах на базі хмари залишається досить важкодоступною. Особливо незрозуміло, як оцінювати і покращувати анонімність відправника, тобто відсутність можливості відстеження тимчасових вузлів, які відправляють повідомлення іншим учасникам оверлейної мережі.

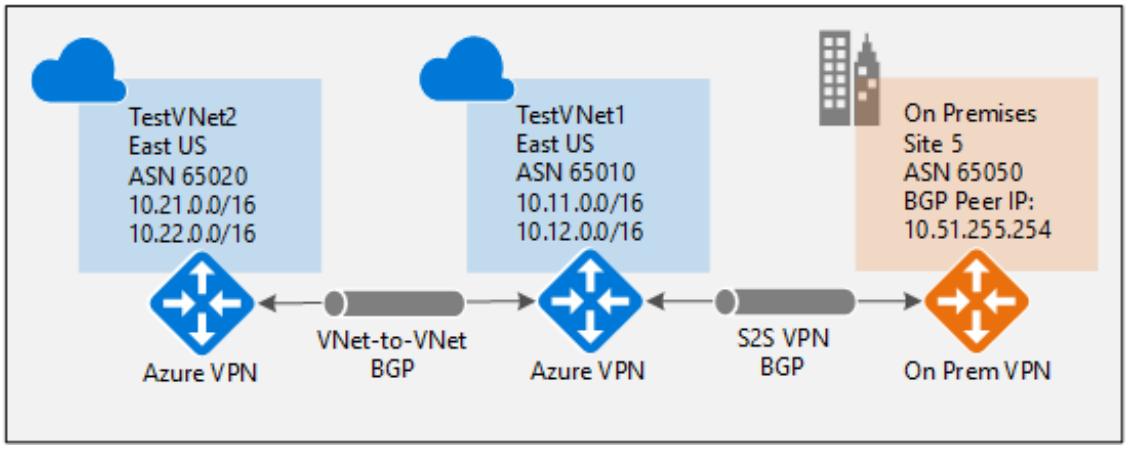

Рисунок 1.4 – Приклад маршутизації VPN-трафіку

В роботі [6], у структурованій віртуальній мережі на базі хмари, організованій у вигляді кільця, дослідники виявили, що метод, спочатку розроблений для анонімності одержувача, також покращує анонімність відправника. Метод заснований на використанні неточних записів в таблицях маршрутизації кожного бере участь однорангового вузла. Це має місце, якщо є необхідність побудувати VPN-інфраструктуру між різними вендорами хмарних послуг, як це показано на рисунку 1.5.

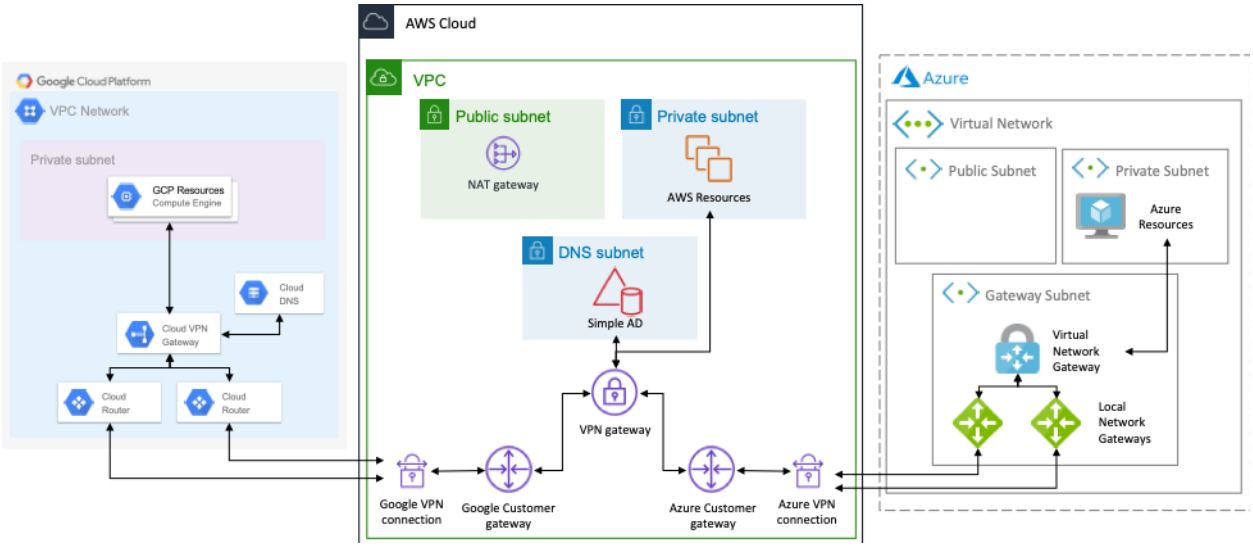

Рисунок 1.5 – Взаємодія між хмарами через VPN-тунелювання

Аналіз показує, що рівень надійності послуг, що надається відправнику, виміряна з точки зору середнього розміру набору показників надійності передачі даних в оверлейних мережах, трохи знижується, якщо однорангові вузли використовують неточну маршрутизацію; тим не менш, рівень надійності вимагає кращого поширення, при цьому хороші рівні надійності стають більш вірогідними за рахунок дуже високих і дуже низьких рівнів хмарної інфраструктури.

1.3 Огляд платформ для створення VPN-ланцюгів

VPN-ланцюг – це техніка, при якій кілька VPN-серверів прив’язуються до єдиної мережі, щоб поліпшити конфіденційність в Інтернеті під час перебування в Інтернеті.

Ідея послідовної передачі трафіку між серверами VPN проста. Таке з'єднання спочатку створюється між комп'ютером користувача і сервером VPN. Потім створюється другий зашифрований канал між першим і другим VPN-сервером. Таким чином, весь трафік шифрується двічі, послідовно. Існує велика кількість архітектурних рішень для побудови VPN-ланцюгів [15-17].

Каскадна схема побудови VPN-ланцюгів – найпопулярніша схема, що передбачає послідовне з'єднання між двома або більше VPN-вузлами, коли трафік йде від одного сервера до іншого (рисунок 1.6). Трафік йде від одного вузла до іншого і на кожному вузлі розшифровується і знову шифрується. Це створює теоретичний ризик перехоплення трафіку на будь-якому з серверів ланцюжка, якщо кіберзлочинець має до нього доступ. Переваги даної схеми – це простота використання і висока швидкість в порівнянні з наскрізною схемою. Достатньо отримати конфігураційний файл, додати його в VPN-клієнта і підключитися. До недоліків можна віднести те, що трафік розшифровується на кожному сервері і на кожному з серверів може бути перехоплений.

Один з провайдерів, що пропонує можливість створювати власні каскади VPN з чотирма серверами, – це Perfect Privacy.

На рисунку 1.6 видно, що IP-адреса користувача змінюється на кожному вузлі і повторно шифрується з використанням шифрування, наприклад, 256-бітного шифрування AES OpenVPN, перш ніж трафік вийде з каскаду VPN в звичайний Інтернет. З кожним вузлом новий VPN-сервер отримує тільки IP-адресу/розташування попереднього VPN-сервера, що додатково приховує і захищає справжню особистість користувача.

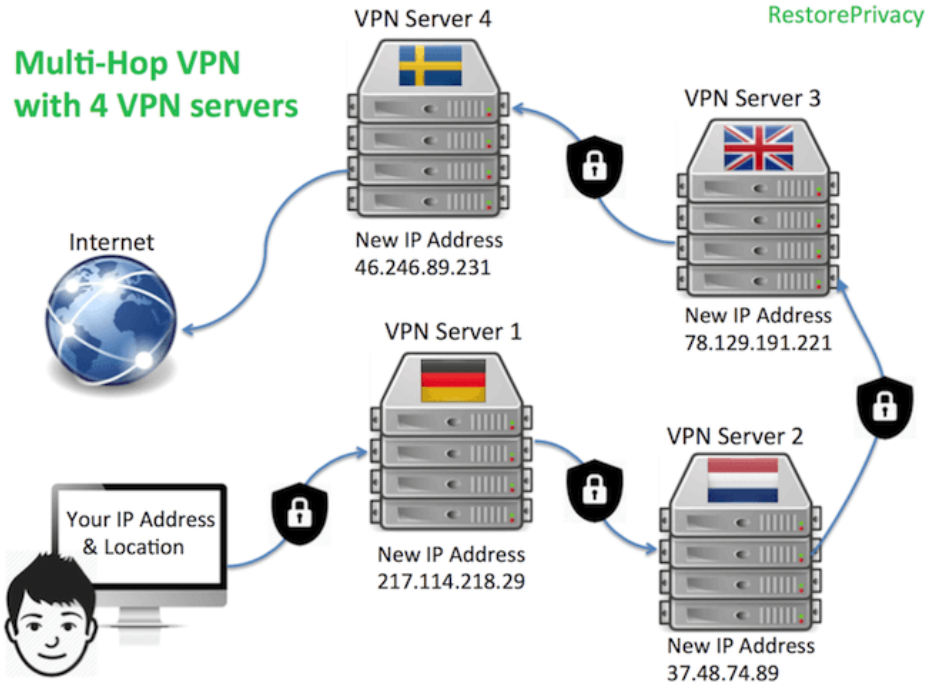

Рисунок 1.6 – Приклад каскадного з`єднання з чотирма VPN-вузлами

Також варто проаналізувати концепт Double-hop VPNs. Це унікальна функція деяких постачальників VPN. За допомогою конфігурації з двома VPN-вузлами, можна бачити вихідну IP-адресу так, що другий вузол може бачити вихідний трафік, але жоден вузол не матиме особистої IP-адреси і вихідного трафіку користувача. Концепт забезпечує гарну продуктивність, а також більш високий рівень безпеки та конфіденційності в порівнянні з налаштуванням з одним VPN-вузлом. Є кілька найбільш ефективних VPN-сервісів, що пропонують конфігурації подвійного переходу: NordVPN, VPN.ac та ProtonVPN.

Існує також підхід Self-configurable multi-hop VPN, який дозволяє індивідуально вибирати сервери в каскаді VPN. Такий VPN-сервіс пропонують наступні вендори послуг:

- Perfect Privacy (надає 4 VPN-вузли). Після тестування численних служб VPN за останні кілька років, Perfect Privacy залишається найкращою VPN для конфіденційності та безпеки. Серверні функції TrackStop і NeuroRouting забезпечують максимальний захист для всіх пристроїв –

незалежно від пристрою, який використовується: мобільний телефон Android з конфігурацією IPSec/L2TP, Mac OS з IKEv2, Windows або Linux. Також в цьому програмному продукті імплементована функція NeuroRouting – технологія динамічної побудови VPN-ланцюжка з механізмом машинного навчання;

- ZorroVPN (надає 4 VPN-вузли). Рішення пропонує гарний вибір вузлів у хмарному середовищі і хорошу продуктивність. Однак, окрім більш високої ціни, головним недоліком ZorroVPN є те, що сервіс не пропонує жодних спеціальних додатків VPN. Це викликає кілька проблем: потрібне використання сторонньої програми для підключення. Також необхідно вручну створити файл конфігурації вузла багатокористувацького VPN-сервера, а потім імпортувати файл у свою програму для VPN-підключення;

- OVPN (надає 2 VPN-вузли). Це служба VPN, яка пропонує конфігурації multi-hop VPN за допомогою додаткової функції. Ця функція коштує незначну суму на місяць на додаток до звичайної підписки на даний програмний продукт. Функція схожа на ProtonVPN та опцію Secure Core, яка дорожча за базовий рівень підписки. OVPN також підтримує IPV6;

- IVPN (надає 2 VPN-вузли). IVPN – це послуга VPN, що базується в Гібралтарі. Не підтримує IPv6, однак підтримує новий протокол WireGuard VPN. Ціни на IVPN вищі за середні, але це також повнофункціональний VPN сервіс із клієнтами для всіх основних операційних систем та пристроїв. Використовує хмарні технології.

Іншим варіантом є створення ланцюгів, використовуючи більше одного постачальника VPN одночасно. Такий підхід ще називають «VPN всередині VPN» або «вкладений ланцюжок» VPN.

Він підходить для захисту користувачів від мережі VPN, яка може бути скомпрометована, а також від сервера VPN, який може бути скомпрометований. Розглянемо різні способи для побудови ланцюгів з використанням більше одного постачальника VPN:

- схема «VPN 1 на маршрутизаторі – VPN 2 на комп’ютері/пристрої».

Це просте налаштування за допомогою VPN на маршрутизаторі, а потім використання іншої служби VPN на комп’ютері чи пристрої користувача, підключеному через VPN-маршрутизатор користувача. Вибір найближчих вузлів допоможе мінімізувати показники продуктивності за допомогою цього налаштування;

- схема «VPN 1 на комп'ютері (хост) – VPN 2 на віртуальній машині». Для реалізації такого підходу необхідно встановити VirtualBox, встановити та налаштувати операційну систему у віртуальній машині, наприклад Linux, а потім встановити та запустити VPN із віртуальної машини. Це налаштування також може підробити іншу операційну систему з головного комп’ютера;

- схема «VPN 1 на маршрутизаторі – VPN 2 на комп'ютері (хост) – VPN 2 на віртуальній машині». Також можна створити віртуальні машини у віртуальних машинах або віртуальні машини з ланцюжком ланцюжків (вкладена віртуалізація). Віртуальні машини – чудовий інструмент конфіденційності та безпеки, оскільки вони дозволяють створювати ізольоване середовище для різних цілей – також відоме як компартменталізація. У VirtualBox можна створити безліч різних віртуальних машин, використовуючи різні операційні системи.

Одним з головних недоліків використання ланцюжків VPN-серверів є значне падіння швидкості при їх використанні. Це пов'язано з шифруванням трафіку і географічною віддаленістю між серверами. Однак існує безліч модифікацій каскадного з’єднання, які компенсують втрату швидкості і при цьому забезпечують високу безпеку користувача.

Тут важливо також розглянути технологію NeuroRouting. По суті – це динамічна, «розумна», багатовузлова конфігурація VPN-ланцюга. Ця функція дозволяє використовувати всю мережу VPN-вузлів для динамічної маршрутизації всього трафіку. Концепція даної технології полягає у таких якостях, як:

- динамічність – інтернет-трафік динамічно маршрутизується через кілька переходів в мережі VPN-вузлів, щоб вибрати найбільш безпечний

маршрут. Шлях маршрутизації заснований на застосуванні TensorFlow, програмному забезпеченні з відкритим вихідним кодом для машинного навчання. Дані користувача залишаються в безпеці і зашифровані в мережі якомога довше перед виходом в Інтернет. Заснована на TensorFlow, оверлейна мережа постійно вивчає кращий і найбезпечніший маршрут для даного вузла

- «розумність» – кожен вузол, до якого користувач намагатиметься отримати доступ, матиме унікальний шлях. Доступ до декількох різних веб-сайтів дає безліч унікальних багатовузлових конфігурацій і різних IP-адрес одночасно. Кожна друга IP-адреса, яка транслюється на вузол, буде відповідати останньому переходу в ланцюжку серверів для даної URL. Функція активується на стороні сервера, тобто при кожному доступі до мережі VPN, незалежно від пристрою або програми, NeuroRouting буде активний. Це також означає, що він буде працювати на будь-якому пристрої – від маршрутизаторів до Mac OS і Android.

Розглянемо приклад реалізації NeuroRouting. На рисунку 1.7 показано, що користувач підключений до VPN-сервера в Рейк'явіку (Ісландія), з активним з'єднанням NeuroRouting.

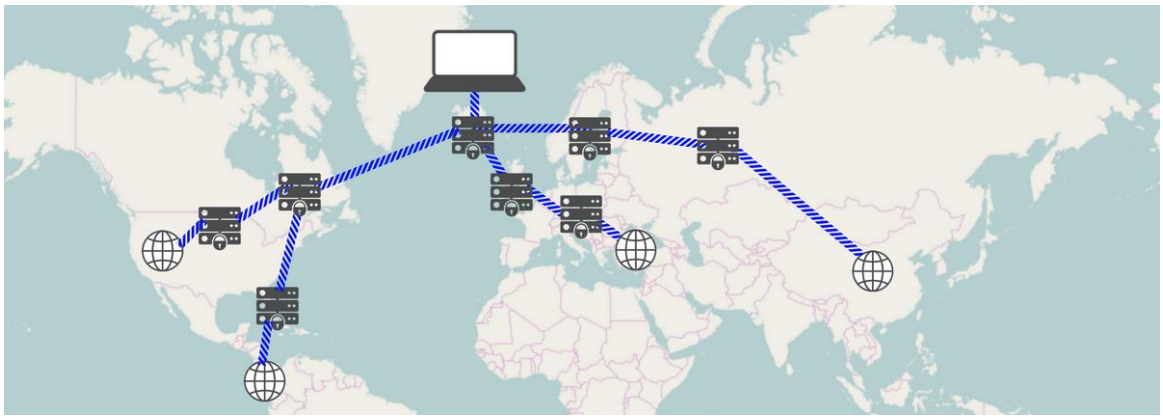

Рисунок 1.7 – Приклад використання технології NeuroRouting

Користувач одночасно відвідує чотири різних вузли, розташованих в США, Панамі, Болгарії і Китаї. Користувач буде транслювати чотири різних IP-адреси на кожен вузол одночасно, що відповідає останньому вузлу в ланцюжку перед вузлом, на який іде запит. Використання хмарних вузлів (віртуальних машин різних вендорів послуг з опціями віртуального маршрутизатора) збільшує заплутаність маршрутів, що підвищує безпеку користувача.

1.4 Постановка задачі

Таким чином, задачею атестаційної роботи є розробка алгоритмічного забезпечення та прототипу програмного засобу для автоматизованої побудови VPN-ланцюгів на платформі IaaS для забезпечення безпечної передачі даних в мережі Інтернет, які б усували основні недоліки існуючих підходів, зокрема затримку передач даних, коефіцієнт втрат тощо.

2 РОЗРОБКА МЕТОДУ АВТОМАТИЗОВАНОЇ ПОБУДОВИ VPN-ЛАНЦЮГІВ НА ПЛАТФОРМІ IAAS

2.1 Технологічні особливості процесу побудови VPN-ланцюгів

Історично склалося, що термінальний доступ дозволив більш раціонально розподіляти обчислювальні ресурси між користувачами перших дуже дорогих обчислювальних машин. З появою дешевих персональних комп'ютерів роль термінального доступу стала трохи знижуватися, так як склалася думка, що достатню продуктивність технологічного процесу можна отримати на робочому столі кожного користувача.

Однак, розвиток технологій віртуалізації і, як наслідок, – хмарних технологій, показало, що дешевизна персональних комп'ютерів не в змозі компенсувати щоденні витрати на супровід великої кількості робочих місць користувачів, що володіють нібито перевагами через можливість персоналізації налаштувань операційних систем і програмного забезпечення. Реально (у великих організаціях), наявність великої кількості «різношерстого» обладнання замість достоїнств створює додаткові складності системним адміністраторам, що обслуговують цю інфраструктуру. Питання забезпечення безпеки інфокомунікаційних систем в умовах ведення міжнародного бізнесу, також зажадали перегляду поглядів і повернення до термінального доступу, як більш уніфікованому і економічно виправданого.

Для поліпшення інформаційної безпеки (наприклад, для запобігання вилучення даних з робочих станцій) великі компанії все частіше і частіше використовують в якості терміналів тонкі клієнти. З'єднання між тонким клієнтом і термінальним сервером (в тому числі, хмарним сервером, IaaS-архітектурою або звичайним хостом) відбувається за певним протоколом (безпосередньо протокол залежить від вибору конкретного термінального

рішення: Microsoft, VMWare, Citrix і т.д.). Таким чином, в разі якщо робоча станція кінцевого користувача скомпрометована, то ризик зараження і пошкодження всієї системи все ж нижче, ніж в інфраструктурі з повноцінною робочою станцією. У цьому ланцюзі важливим є питання безпеки з'єднання між тонким клієнтом і віртуальною машиною або сервером. Наприклад, якщо це медична інформація (дані з високим рівнем конфіденційності), то вимоги до безпеки зростають на порядок, так як стоїть завдання виключення витоку конфіденційної медичної інформації та персональних даних пацієнтів, що регламентується законодавчими базами у більшості країн світу.

Перспективним напрямком підвищення безпеки в термінальних системах з тонкими клієнтами є використання VPN-технологій в хмарному середовищі. Одним з підходів, що активно розвивається, є підхід, коли може бути задіяно кілька VPN-серверів з побудованих у вигляді ланцюжка. Цьому в оглядовій частині було приділено значну увагу. Тому, аби не повторюватися, виокремимо основні моменти.

Відомим мінусом використання технології VPN є зниження часу передачі даних. А при побудові VPN-ланцюжків з декількома проміжними вузлами, складність завдання визначення оптимального маршруту не може бути вирішена за прийнятний час з причини високої швидкості зміни пропускних швидкостей між вузлами.

Технологія Multi-hop VPN chains є відносно новою, так як її розвиток пов'язаний з бурхливим розвитком хмарних технологій. Якщо раніше використання VPN-вузлів в якості елементів каскадних схем були не так поширеними, а безпеку (як і анонімізація) в мережі інтернет розглядалася з точки зору використання проксі-серверів, то розвиток PaaS і IaaS дозволило створювати віртуальні платформи для VPN-вузлів будь-яких масштабів .

В роботі [7] авторами пропонується прототип системи самоналаштовуваної VPN-мережі з використанням для маршрутизації та ізоляції трафіку – зв'язки технічних рішень OpenFlow і OpenVSwitch, VRF і OSPF. В роботі проаналізовано функціонал full-mesh мережі для клієнтів без

участі сервера при передачі трафіку (для тих випадків, в яких це можливо). Як недолік даного підходу важливо відзначити, що не реалізований алгоритм попереднього розрахунку топології в цілому. Крім того, реалізація full-mesh архітектури вимагає додаткових розрахунків оптимальних маршрутів для кожної пари віртуальних маршрутизаторів.

В огляді комерційних рішень [8] розглянуто ряд рішень для побудови ланцюгових схем з використанням. В атестаційній роботі уже була описана технологія NeuroRouting. NeuroRouting – технологія динамічного побудови VPN-ланцюжка з механізмом машинного навчання. Ця технологія дозволяє використовувати всю сукупність VPN-серверів для динамічної маршрутизації всього трафіку. Недоліком даного рішення є значний час побудови маршруту, сумарна мережева затримка в більшості випадків перевищує задану межу, при цьому користувач не може вносити зміни в ручному режимі. До всього іншого, алгоритм побудови маршруту заснований на виборі серверів з регіонального переваги.

В роботі [9] розглянуто варіант вирішення завдання вибору пропускних спроможностей каналів зв'язку транспортної мережі. Розроблений алгоритм вибору пропускних спроможностей гарантує мінімальні витрати на оренду каналів зв'язку з оптимальною пропускною спроможністю за умови виконання вимог до якості обслуговування.

З використанням множників Лагранжа розглянуто алгоритм вибору пропускних спроможностей каналів зв'язку для збалансованої за пріоритетами транспортної мережі зв'язку. Висока швидкодія даного алгоритму забезпечується за рахунок застосування алгебраїчних операцій над матрицями (додавання, множеннятощо). З використанням узагальненого методу множників Лагранжа було реалізовано порівняння умовних екстремумів функції витрат на оренду каналів зв'язку для одиночних активних проміжних вузлів, для всіх можливих пар, трійок і так далі аж до випадку, коли одночасно будуть активні всі проміжні вузли. Ефективність запропонованого підходу автори показують шляхом імітаційного

моделювання. При цьому в якості недоліків потрібно відзначити відсутність механізму мінімізації часу рішення задачі побудови оптимального маршруту з проміжних вузлів. Практичне відтворення даного підходу показує, що час побудови оптимального маршруту неприйнятно з урахуванням динаміки зміни пропускних спроможностей між вузлами.

2.2 Розробка алгоритмічного забезпечення

Для роботи і інструментарієм розробки алгоритму введемо такі поняття:

- термінал – вузол, з яким безпосередньо працює користувач, наприклад, тонкий клієнт;
- хост – вузол, до якого підключається термінал за допомогою різних мережевих протоколів через комп'ютерну мережу, наприклад, віртуальна машина на сервері віртуалізації;
- проміжний вузол – вузол, через який відбувається комутація терміналу з хостом за допомогою VPN-тунелювання.

Згідно з умовою задачі, необхідно організувати автоматизоване підключення терміналу до хосту через проміжні віртуальні вузли з метою забезпечення безпеки віддаленої роботи користувачів. Чим більше проміжних вузлів задіяно, тим нижча ймовірність перехоплення і розшифровки трафіку потенційним зловмисником. Однак сумарний час затримки передачі даних між терміналом і хостом залежить від технології віддаленого доступу і, як правило, не повинно перевищувати 50-75 мс.

Додатковими умовами є: час розрахунку маршруту з ланцюжка VPN-серверів не повинно перевищувати допустимого часу, необхідного на прилаштовуванням ланцюжка в разі відмови одного з проміжних VPN-серверів або зростання неприпустимого часу затримки передачі даних між терміналом і хостом; в разі неможливості побудови ланцюжків із заданою кількістю вузлів, ланцюжок може містити меншу кількість вузлів до моменту

часу, коли буде ініційовано перебудову ланцюжка в цілому; періодичність перебудови ланцюжка заздрості від кількості вузлів в ланцюжка; кількість вузлів в ланцюжку задається користувачем, сукупність проміжних вузлів може бути представлена у вигляді повнозв'язну графа.

З урахуванням всіх обмежень задачу можна представити у вигляді:

$$T_{(x\to y)} \to \min, при \begin{cases} T_{(x\to y)} \le T_{protocol} \\ T_{route} \le T_{task} \\ b = 1,2,n \end{cases}, \qquad (2.1)$$

де $T_{(x\to y)}$ – часова затримка при передачі даних між x-терміналом і y-хостом;

$T_{protocol}$ – максимально допустима часова затримка при передачі даних між x-терміналом і y-хостом, згідно обраного протоколу передачі даних;

T_{route} – часова затримка на побудову маршруту передачі даних між x-терміналом і y-хостом;

T_{task} – максимально допустима часова затримка при побудові маршруту передачі даних між x-терміналом і y-хостом;

b – кількість VPN-серверів, задіяних для побудови ланцюжка.

Часова затримка при передачі даних між x-терміналом і y-хостом визначається за формулою:

$$T_{(x\to y)} = t_{(x\to b_1)} + t_{(x\to b_2)} + ... + t_{(x\to b_n)} + t_{(b_n\to y)} + \sum t_{b_n} \qquad (2.2)$$

де $t_{(x\to b_n)}$ – часова затримка при передачі даних між x-терміналом і n-VPN-сервером;

$t_{(b_n\to y)}$ – часова затримка при передачі даних між n-VPN-сервером і y-хостом;

t_{b_n} – час, який витрачається на комунікаційні процеси на VPN-сервері.

Часова затримка на побудову маршруту передачі даних між х-терміналом і у-хостом визначається за формулою:

$$T_{route} = t_{route} + \sum t_{communication} \tag{2.3}$$

де $t_{communication}$ – час, що витрачається на авторизацію і встановлення маршруту на кожному з VPN-серверів;

t_{route} – час, що витрачається на побудову квазіоптимального маршруту передачі даних між усіма учасниками віртуальної мережі, визначається як:

$$t_{route} = \begin{cases} t_{(\Theta)} + t_{(\Theta_x+\Theta_y)} + t_{search\min}, \ if \ \ b=1; \\ t_{(\Theta_x)} + t_{sort\min\Theta_x} + t_{search\ group\ I}, \ if \ \ b=2; \\ t_{(\Theta)} + t_{sort\min\Theta} + t_{search\ group\ II}, \ if \ \ b=3; \\ t_{(\Theta)} + t_{sort\min\Theta} + (n-1)\times t'_{sort\min\Theta_{b_n}} + \sum t_{(\Theta_{b_n})} + t_{search\ group\ I}, \ якщо \\ \qquad b=4,6,...n,\ n=2k,\ k\in\mathbb{N}; \\ t_{(\Theta)} + t_{sort\min\Theta} + n\times t'_{sort\min\Theta_{b_n}} + \sum t_{(\Theta_{b_n})} + t_{search\ group\ II}, \ якщо \\ \qquad b=5,7,...n,\ n=2k+1,\ k\in\mathbb{N}; \end{cases} \text{, при} \tag{2.4}$$

$$t_{(\Theta)} = \begin{cases} t_{(\Theta_x)}, \ if \ t_{(\Theta_x)} \geq t_{(\Theta_y)}; \\ t_{(\Theta_y)}, \ if \ t_{(\Theta_y)} \geq t_{(\Theta_x)}; \end{cases} \tag{2.5}$$

де $t_{(\Theta)}$ – час, що витрачається на побудову матриці тимчасових затримок;

Θ_x – матриця тимчасових затримок, що витрачаються на передачу даних між х-терміналом і всіма VPN-серверами;

Θ_y – матриця тимчасових затримок, що витрачаються на передачу даних між у-хостом і всіма VPN-серверами;

$t_{(\Theta_x+\Theta_y)}$ – час, що витрачається на складання матриць;

$t_{search\min}$ – час, що витрачається на пошук мінімального значення затримки в складеної матриці;

$t_{sort\min}$ – час, що витрачається на сортування даних в матриці по зростанню;

$t_{search\ group\ I}$ – час, що витрачається на пошук першого-ліпшого VPN-сервера, який задовольняє умові:

$$T_{protocol} - t_{(x\to b_1)} - \sum t_{b_n} \geq t_{(b_{n-1}\to b_n)} + t_{(b_n\to y)} \tag{2.6}$$

$t_{search\ group\ II}$ – час, що витрачається на пошук першого-ліпшого VPN-сервера, який задовольняє умові:

$$T_{protocol} - t_{(x\to b_1)} - \sum t_{b_n} - t_{(b_n\to y)} \geq t_{(b_{n-2}\to b_{n-1})} + t_{(b_{n-1}\to b_n)} \tag{2.7}$$

У такому вигляді, при практичній реалізації розглянутого рішення, алгоритмічний вигляд матиме гіллясту структуру. Вектор рішення буде залежати від єдиного параметра, який задає користувач – кількості VPN-серверів.

Отже, розглянемо послідовність кроків для випадку, якщо користувачем обраний один VPN-сервер. Передбачається, що всі VPN-сервери свідомо мають уявлення про зв'язку термінал-хост.

Крок 1. Паралельно відбувається наповнення матриць Θ_x і Θ_y. Час, що витрачається на цю операцію буде визначено, виходячи з виразу (2.5).

Крок 2. Матриці Θ_x і Θ_y. складаються.

Крок 3. Пошук мінімального значення в складеної матриці.

Знайдений VPN-сервер буде обраний в якості рішення задачі.

Розглянемо послідовність кроків для випадку, якщо користувачем

вибрано два VPN-сервера.

Крок 1. Відбувається наповнення матриці Θ_x.

Крок 2. Сортування значень матриці Θ_x по зростанню.

Крок 3. В якості першого VPN-сервера береться той, час якого $t_{(x\to b_1)}$ мінімальний. Цей VPN-сервер виключається для подальшого пошуку другого VPN-сервера.

Крок 4. Виконується перевірка умови (2.6).

Крок 4.1. Перший знайдений, що задовольняє умові (2.6), VPN-сервер за час $t_{search\ group\ I}$ зупиняє алгоритм.

Крок 4.2. Якщо задовольняючий умові (2.6), VPN-сервер не знайдений, то повертаємося до кроку 3 і беремо $t_{(x\to b_2)}$. Цикл може повторюватися до $t_{(x\to b_n)}$.

Визначені VPN-сервери будуть обрані в якості рішення задачі.

Розглянемо послідовність кроків для випадку, якщо користувачем вибрано три VPN-сервери.

Крок 1. Паралельно відбувається наповнення матриць Θ_x і Θ_y. Час, що витрачається на цю операцію буде визначено, виходячи з виразу (2.5).

Крок 2. Паралельне сортування значень матриць Θ_x і Θ_y по зростанню.

Крок 3. В якості першого VPN-сервера береться той, час якого $t_{(x\to b_1)}$ чи $t_{(y\to b_3)}$ мінімальний. Якщо, наприклад, визначено першим VPN-сервер b_1, то він виключається для подальшого пошуку VPN-серверів b_2 і b_3.

Крок 4. Виконується перевірка умови (2.7).

Крок 4.1. Перший знайдений, що задовольняє умові (2.7), VPN-сервер за час $t_{search\ group\ II}$ зупиняє алгоритм.

Крок 4.2. Якщо задовольняє умові (2.7), VPN-сервер не знайдений, то:

Крок 4.2.1. Беремо значення $t_{(x\to b_2)}$ і виконуємо перевірку умови (2.7). Цикл може повторюватися до $t_{(x\to b_n)}$.

Крок 4.2.2. Якщо VPN-сервер не знайдений, то беремо значення і виконуємо перевірку умови (2.7). Цикл може повторюватися до $t_{(y \to b_n)}$.

Визначені таким чином VPN-сервери будуть обрані в якості рішення задачі. Якщо користувач вибирає 4 і більше VPN-серверів, то алгоритм має вигляд (2.4).

2.3 Числовий експеримент та оцінка ефективності

Розглянемо приклад побудови ланцюжка VPN з 6 VPN-серверами для організації термінальної системи на платформі протоколу RDP. Максимальний час затримки між терміналом і хостом не повинен перевищувати 50 мс. Припустимо, у нас є пул з 15 VPN-серверів. Матриця тимчасових затримок між VPN-серверами, терміналом і хостом представлена в таблиці 2.1.

Користувач вказав у вимозі побудову ланцюжка з 6 VPN-серверів. Маршрут буде побудований, виходячи з (2.4). Послідовність дій буде наступною.

Крок 1. У вектор-рядках $\underline{x} = (b_1,...,b_{15})$ и $\underline{y} = (b_1,...,b_{15})$ знаходяться мінімальні значення. Так як значення для y було знайдено як b_{14} швидше, то для x воно є недоступним. Результатом пошуку визначено такі зв'язки: $y \to b_{14}; x \to b_{15}$.

Крок 2. У вектор-рядках $\underline{b_{14}} = (b_1,...,b_{13})$ і $\underline{b_{15}} = (b_1,...,b_{13})$ знаходяться мінімальні значення. Так як значення для b_{14} було знайдено як b_3 швидше, то для b_{15} воно є недоступним. Результатом пошуку визначено такі зв'язки: $b_{14} \to b_3; b_{15} \to b_8$. Так як b_{14} і b_{15} мають по 2 плеча, вони виключаються з подальшого пошуку, тобто вектор-стовпці $\overline{b_{14}} = (b_1,...,b_{15})$ і $\overline{b_{15}} = (b_1,...,b_{15})$ стають недоступними.

Таблиця 2.1 – Матриця часових затримок між вузлами

Вузол	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
b_1	0	15	20	1	13	15	48	15	12	2	8	7	9	10	11
b_2	15	0	15	48	48	5	58	20	22	27	20	12	13	15	8
b_3	20	15	0	7	8	4	8	1	6	8	7	5	3	1	5
b_4	1	48	7	0	8	5	99	50	5	8	4	7	9	50	28
b_5	13	48	8	8	0	13	45	47	12	12	10	14	15	14	20
b_6	15	5	4	5	13	0	11	8	9	7	5	8	4	8	7
b_7	48	58	8	99	45	11	0	1	1	2	5	8	4	2	10
b_8	15	20	1	50	47	8	1	0	8	9	5	1	7	8	2
b_9	12	22	6	5	12	9	1	8	0	12	15	88	74	50	3
b_{10}	2	27	8	8	12	7	2	9	12	0	35	15	12	15	18
b_{11}	8	20	7	4	10	5	5	5	15	35	0	99	90	84	54
b_{12}	7	12	5	7	14	8	8	1	88	15	99	0	32	15	2
b_{13}	9	13	3	9	15	4	4	7	74	12	90	32	0	7	8
b_{14}	10	15	1	50	14	8	2	8	50	15	84	15	7	0	1
b_{15}	11	8	5	28	20	7	10	2	3	18	54	2	8	1	0
x	13	15	11	10	8	7	2	5	4	5	8	77	20	21	1
y	3	8	90	20	58	47	45	50	30	12	10	11	8	2	4

Крок 3. У вектор-рядку $\underline{b_8} = (b_1, b_2, b_4, b_5, b_6, b_7, b_9, b_{10}, b_{11}, b_{12}, b_{13})$, після сортування значень по зростанню, береться (з мінімальним значенням) елемент b_7. Так як b_8 має 2 плеча, він виключаються з подальшого пошуку, тобто вектор-стовпець $\overline{b_8} = (b_1, ..., b_{15})$ стає недоступними для пошуку в ньому значень на наступних кроках.

Крок 4. Так як кількість VPN-серверів, заданих користувачем кратне двом, то перевіряємо умову (2.6). Для цього складаємо вектор-рядки $\underline{b_3} = (b_1, b_2, b_4, b_5, b_6, b_9, b_{10}, b_{11}, b_{12}, b_{13})$ і $\underline{b_7} = (b_1, b_2, b_4, b_5, b_6, b_9, b_{10}, b_{11}, b_{12}, b_{13})$. Отримані значення перевіряємо в умови (2.6). Перше задовольняє значення є шуканим значенням для потрібного VPN-сервера. У нашому випадку це b_6:

$$T_{protocol} - t_{(x \to b_{15})} - t_{(b_{15} \to b_8)} - t_{(b_8 \to b_7)} - t_{(b_{14} \to y)} - t_{(b_3 \to b_{14})} \geq$$
$$\geq t_{(b_7 \to b_6)} + t_{(b_3 \to b_6)} \Rightarrow 43 > 15$$

Крок 5. Сумарний час затримки передачі даних між терміналом і хостом складе: 22 мс, що задовольняє максимальний поріг в 50 мс, згідно вимог, поставлених в задачі.

Матриця, що відображає роботу алгоритму представлена в таблиці 2.2.

Таблиця 2.2 – Результат роботи автоматизованої побудови VPN-ланцюга

Вузол	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
b_1	0	1	2	1	1	1	4	1	1	2	8	7	9	1	1
b_2	1	0	1	4	4	5	5	2	2	2	2	1	1	1	8
b_3	2	1	0	7	8	4	8	1	6	8	7	5	3	1	5
b_4	1	4	7	0	8	5	9	5	5	8	4	7	9	5	2
b_5	1	4	8	8	0	1	4	4	1	1	1	1	1	1	2
b_6	1	5	4	5	1	0	1	8	9	7	5	8	4	8	7
b_7	4	5	8	9	4	1	0	1	1	2	5	8	4	2	1
b_8	1	2	1	5	4	8	1	0	8	9	5	1	7	8	2
b_9	1	2	6	5	1	9	1	8	0	1	1	8	7	5	3
b_{10}	2	2	8	8	1	7	2	9	1	0	3	1	1	1	1
b_{11}	8	2	7	4	1	5	5	5	1	3	0	9	9	8	5
b_{12}	7	1	5	7	1	8	8	1	8	1	9	0	3	1	2
b_{13}	9	1	3	9	1	4	4	7	7	1	9	3	0	7	8
b_{14}	1	1	1	5	1	8	2	8	5	1	8	1	7	0	1
b_{15}	1	8	5	2	2	7	1	2	3	1	5	2	8	1	0
x	1	1	1	1	8	7	2	5	4	5	8	7	2	2	1
y	3	8	9	2	5	4	4	5	3	1	1	1	8	2	4

Зауваження: на всіх етапах додатково перевіряється різниця часу між пороговим і фактичним. У разі перевищення заданого часу, в відсортованих вектор-рядках необхідно вибирати такі + 1-значення *b*.

У разі неможливості побудови маршруту через відсутність допустимого рішення, алгоритм передбачає зниження кількості VPN-серверів в ланцюжку з наступним повідомленням користувача.

Для визначення ефективності роботи розробленого методу було

проведено ряд порівняльних експериментів. Програмний засіб, що реалізує логіку розробленого методу, порівнювався з програмним продуктом Perfect Privacy, які реалізують функцію NeuroRouting. Умовами проведення експерименту були: 4 VPN-сервера, інтервал часу між прилаштовуванням VPN-ланцюжка 1-4 години.

В результаті проведених експериментів отримано такі результати. На рисунку 2.1 показана робота програмного продукту Perfect Privacy.

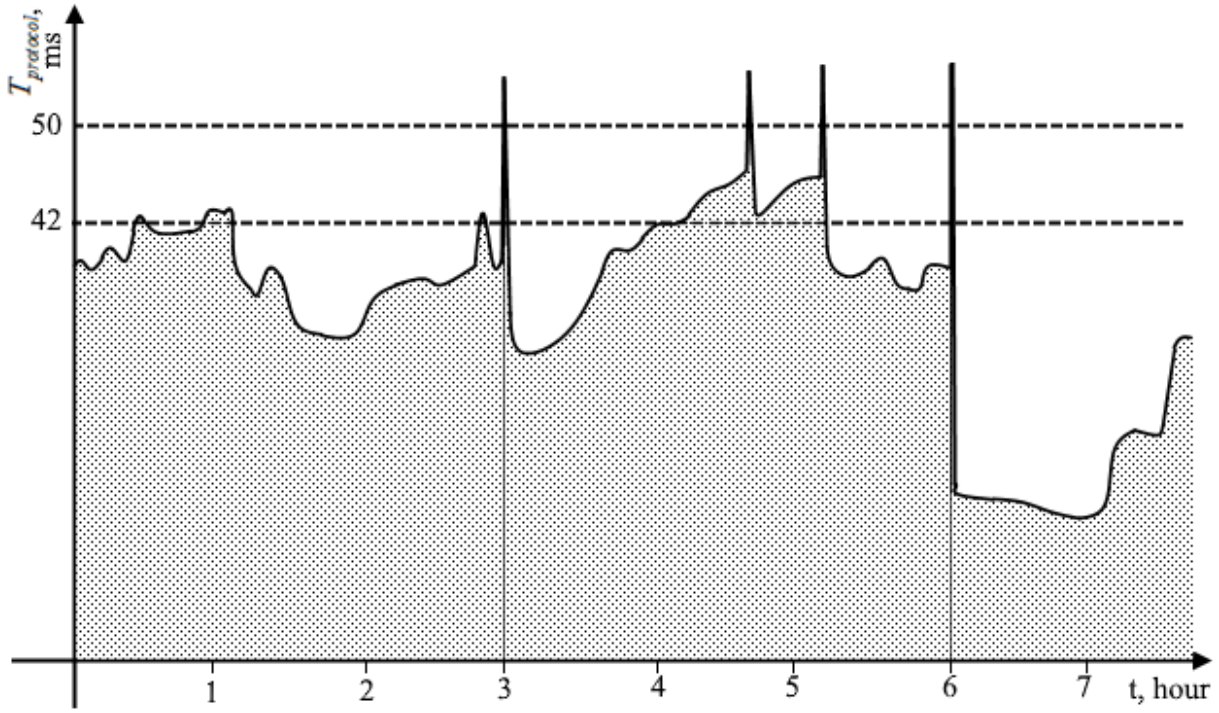

Рисунок 2.1 – Результат роботи Perfect Privacy

З рисунка 2.1 видно, що перебудова ланцюжка відбувалося 4 рази, з яких 2 планових та 2 позапланових в зв'язку з підвищенням часової затримки предпорогового значення в 42 мс. У зв'язку з тим, що користувач не може змінити частоту перебудови ланцюжка, безпека такого з'єднання знижується, так як використання одного і того ж VPN-тунелю протягом тривалого часу знижує безпеку з'єднання в цілому.

На рисунку 2.2 показана робота програми, що працює на основі розробленого методу (код фрагментів програми наведено у додатку Б).

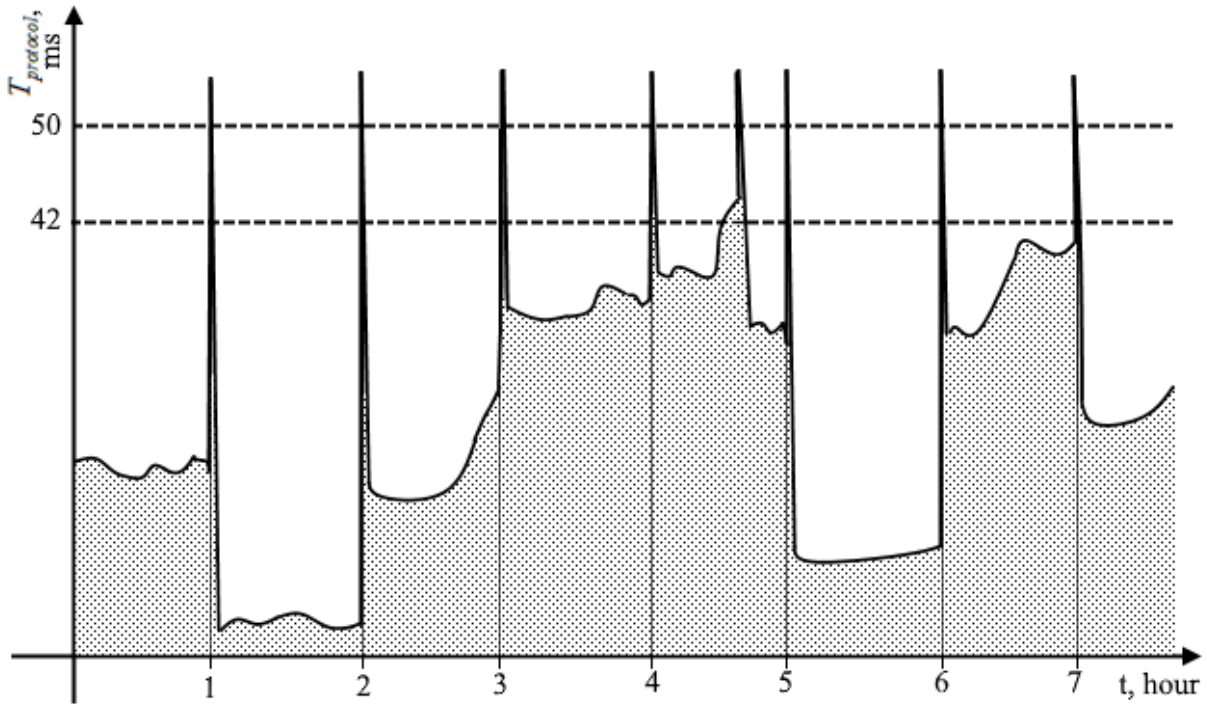

Рисунок 2.2 – Результат роботи програми з використанням запропонованого методу

З рисунка 2.2 видно, що перебудова ланцюжка відбувається щогодини, що підвищує безпеку з'єднання. Сумарна затримка на 15% нижче за рахунок вибору VPN-серверів з мінімальною затримкою. При цьому позапланова перебудова ланцюжка відбувалася 1 раз при перевищенні часової затримки предпорогового значення в 42 мс.

Це підтверджує ефективність запропонованого методу в порівнянні з існуючим – NeuroRouting.

3 МОДЕЛЬ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ В VPN-ЛАНЦЮГАХ В УМОВАХ НЕПОВНИХ ДАНИХ IAAS-ІНФРАСТРУКТУРИ

Для VPN-ланцюга з постійною топологією і навантаженням на канали зв'язку можна записати задачу маршрутизації в загальному вигляді з урахуванням зниження продуктивності каналів зв'язку при збільшенні навантаження.

Для орієнтованого графа $G = \langle V, E \rangle$, $V = \{\upsilon_i\}$, $E = \{e_j\}$ з пропускною спроможністю дуг $c(e) \forall e \in E$ і матриці вимог *D*, яка для кожної пари $(s,t) \in V \times V$ задає величину потоку *D(s, t)* з вершини *s*, яка називається джерелом в вершину *t*, яку називають стоком. Кожній парі *(s,t)* і дузі *e* можна зіставити змінну $f_e^{(s,t)} \geq 0$, що позначає обсяг трафіку з *s* в *t* по дузі *e*, тоді сумарний потік через ребро *e* визначиться як:

$$l(e) = \sum_{s,t \in V} f_e^{(s,t)} \tag{3.1}$$

Нехай φ_e – кусково-лінійна функція вартості використання ребра *e*, тоді спільне завдання маршрутизації є завдання на мережі з декількома сутностями:

$$\arg\min \varphi = \sum_{e \in E} \varphi_e \tag{3.2}$$

З заданими умовами приросту вартості у вигляді:

$$\varphi_e = \begin{cases} \xi_1^1 l(e) - \xi_1^2 c(e) \\ \xi_2^1 l(e) - \xi_2^2 c(e) \\ \dots \\ \xi_n^1 l(e) - \xi_n^2 c(e) \end{cases} \tag{3.3}$$

де n – число ділянок, на які розбивається VPN-структура φ_e.

Рівняння балансу потоків (3.3) можна переписати у вигляді:

$$\sum_{u:(u,v)\in E} f_{(u,v)}^{(s,t)} - \sum_{u:(v,u)\in E} f_{(v,u)}^{(s,t)} = \begin{cases} -D(s,t), \ якщо \ v = s, \\ D(s,t), \ якщо \ v = t, \quad v,s,t \in V \\ 0, \ в \ інших \ випадках. \end{cases} \tag{3.4}$$

У наведеному формулюванні задача вирішується алгоритмом SPF, причому основна увага зосереджена на виборі величин ξ_n^1 і ξ_n^2 обгрунтуванні цього вибору.

Як було показано раніше, в віртуальних мережах з динамікою: топологією і навантаженням на канали зв'язку та вузли, яка визначається матрицею вимог *D*, при збільшенні розмірів принципово неможливо здійснити глобальний пошук оптимальних рішень для кожної пари вузлів. Відсутність апріорно-надійних каналів зв'язку викликає потенційну можливість спотворення службової інформації.

Крім того, оскільки передача інформації в VPN-ланцюгу займає досить тривалий період, такий, що маючи «моментальний» маршрут, в вузлі джерелі існує ймовірність його руйнування раніше, ніж за нього буде переданий весь трафік відповідної вимоги до вузла призначення Таким чином, необхідно розглядати завдання створення VPN-ланцюга в автоматизованому вигляді, сформульоване в розділі 2 з урахуванням зміни в часі вихідного графа і множини відповідних вимог. В цьому випадку маршрут теж є функцією часу:

$$\rho(t)_{\langle i,j\rangle} = \rho(t, v_i, v_j, G(t), D(t)) \quad (3.5)$$

Коректність процедури вибору маршруту залежить від адекватності уявлення стану G в момент вибору і від швидкості його отримання. Чим точніше інформація про топологію і структуру вимог, тим правильніше буде рішення в момент його отримання, тобто коректність вихідних даних є необхідною умовою. Якщо знехтувати часом, що витрачається на вирішення задачы, то можна вважати, що в початковий момент маршрут, побудований на коректних вхідних даних, є оптимальним. Однак, при передачі трафіку по знайденому маршруту VPN-ланцюга можливі такі зміни стану G або D, які погіршать його початкові характеристики або зроблять непридатним.

Кожен вузол, який бере участь в маршрутизації, виконує збір інформації про стан мережевих компонентів – каналів і вузлів (по суті ребер і вузлів G). Внаслідок наявності затримок на доставку маршрутної інформації її точність зменшується в залежності від відстані. Оскільки розмір вихідного графа і, відповідно, матриці вимог є зовнішніми параметрами системи і не можуть бути змінені, то розумним є обмеження області збору точної інформації. За рахунок цього можна досягнути відразу два результату: по-перше, збільшується швидкість збору інформації про параметри і підвищується її точність, по-друге, знижується час отримання самого рішення з вихідних даних, оскільки час виконання завдання сильно залежить від розміру вхідного графа. Методи вирішення таких завдань, сильно залежать від параметрів розмірності вхідних множин, в даному випадку від стану графа мережі. Виникає питання: як обмежити область пошуку рішення для кожного вузла, забезпечити пошук маршрутів і передачу по ним призначеного для користувача трафіку в VPN-ланцюзі?

Вище було показано, що маршрут, який задовольняє умовам задачі, є функція часу, отже, його обчислення (за винятком тривіальних випадків) може бути виконано тільки з використанням розподіленої процедури, що виконується на вузлах, які він в себе включає. Це обумовлено тим, що

прогноз поведінки *G(t)* та *D(t)* не може бути отриманий. Для обчислення чергової ділянки маршруту необхідно точно знати стан оверлейної мережі в деякому околі вузла, оптимізуючого трафік усередині цієї області. Приблизних знань про зовнішній області (поза обраної околиці) досить для вибору вузла-посередника для подальшого побудови маршруту VPN-ланцюга. Диференційоване уявлення інформації про структуру мережі поблизу вузла, який будує маршрут, і інформації про віддалені ділянки мережі, дозволяє зробити якісний вибір вузла-посередника. Інформація та структура мережі біля цього вузла повинна бути найбільш точної і повної, а про її віддалені ділянки може бути наближеною.

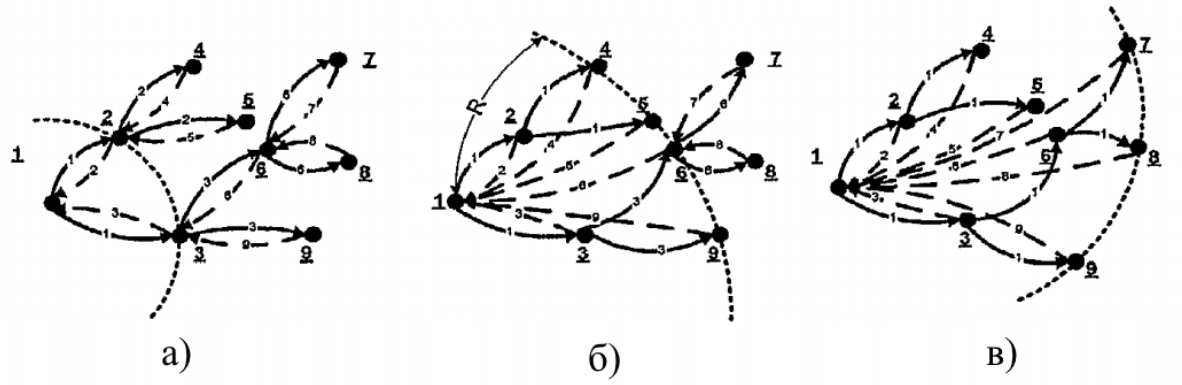

Рисунок 3.1. – Сценарії збору маршрутної інформації при побудові маршрутного уявлення VPN-ланцюга: а) вектор дистанції; б) стан каналу; в) область ефективної маршрутизації

На рисунку 3.1 наведені три варіанти збору маршрутної інформації та способу побудови маршруту. Випадок а) відповідає алгоритмам на основі вектора дистанції, представником яких є протокол RIP. Збір інформації здійснюється тільки за допомогою найближчих сусідів, в припущенні що вони володіють достовірною інформацією. Передача інформації користувачів проводиться аналогічно: визначається тільки найближчий сусід, який може передати дані по оптимальному маршруту. Даний метод маршрутизації простий в реалізації, практично не вимагає обчислювальних витрат, однак

має тенденцію до утворення петель.

Варіант б) відповідає алгоритмам на основі SPF. Тут збір інформації здійснюється з усіх вузлів мережі, кожним маршрутизатором (в ряді випадків для цього використовується багатоадресна розсилка повідомлень). Формування маршруту виконує кожен вузол самостійно тільки на основі тієї інформації якою володіє, однак при цьому вважається, що всі маршрутизатори мають однакові топологічними базами, матрицями вимог і завантаження каналів.

Таким чином, обмежуючи «сферу компетенції» кожного вузла, можна підвищити точність інформації про найближчі ділянках оверлейної мережі, навмисно знижуючи точність представлення віддалених ділянок. Алгоритм маршрутизації в оверлейних мережах, де у якості вузлів використовуються віртуальні елементи IaaS-інфраструктури, а сам процес відбувається в автоматизованому режимі, умовно можна називати алгоритмом маршрутизації з використанням неповних даних.

Отже, розглянемо поняття області ефективної маршрутизації. Під маршрутної інформацією розуміється набір вихідних даних для алгоритму маршрутизації, що включає в себе інформацію про топологію IaaS-інфраструктури, навантаження на канали і вузли, найкоротші маршрути і іншу інформацію, на підставі якої здійснюється управління трафіком. Склад цієї інформації визначається типом використовуваного алгоритму.

Розглянемо оверлейну мережу, побудовану на базі IaaS-інфраструктури, топологія якої описується графом $G(t) = \langle V, E \rangle$, причому множина вершин V відповідає множині вузлів мережі, а множина дуг E – симетричним каналам зв'язку, і зафіксуємо її в момент часу t_0.

Будемо називати $d(v_i, v_j)$ відстанню між вузлами, яке буде визначатися як найменша кількість проміжних переходів між ними. Нехай величина $d(v_i, v_j)$, $v_i, v_j \in V$ має такі властивості:

$$
\begin{aligned}
&d(v_i, v_j) = d(v_j, v_i), \forall v_i, v_j, \ \textit{якщо} \ \exists e_{ij} \in E; \\
&d(v_i, v_j) \geq 0 \ ma \ d(v_i, v_j) = 0 \Leftrightarrow v_i = v_j; \\
&\exists (v_i, v_j), (v_i, v_k), (v_j, v_k) \Rightarrow d(v_i, v_j) \leq d(v_k, v_i) + d(v_k, v_j)
\end{aligned}
\tag{3.6}
$$

В якості d можна вибирати будь-яку функцію, що володіє властивостями (3.6). Така функція є метрикою. Надалі, в якості d буде використовуватися найменша кількість проміжних переходів між вузлами в оверлейній мережі. Всі отримані результати, після незначних змін можуть бути поширені на випадки вибору інших метрик.

Для кожного $v_i \in V$ виберемо параметр $R > 0$, який визначає деякий зв'язний підграф $G(u_i, R) \subset G, G(v_i, R) = \langle V_i, E_i \rangle$. Значення праметра R, обране для вузла v_i, позначимо R_i. Елементами $G(u_i, R)$ будуть всі вузли і канали зв'язку, віддалені від v_i не більше, ніж на R_i, тобто відстань $d(v_i, v_j) \leq R_i \forall v_i \in G.$

Якщо вибирати величину R_i таким чином, щоб інформація про стан підграфа $G(u_i, R)$ могла бути отримана за досить короткий проміжок часу, то вузол v_i буде мати у своєму розпорядженні об'єктивну інформацію, необхідну для точного обчислення оптимальних шляхів між усіма парами вершин всередині $G(u_i, R)$.

Крім цього, обчислювальні витрати для невеликого підграфа будуть досить низькими, щоб забезпечити роботу в реальному часі ефективних алгоритмів пошуку шляхів. У число цих алгоритмів входить метод девіації потоків і інші методи, незатребувані в даний час з-за своєї обчислювальної складності.

Є ще одна передумова для обмеження області дії алгоритму оптимізації трафіку. Вона пов'язана зі зменшенням точності рішення. Якби оверлейну мережу, трафік якої піддається оптимізації була статичною, тобто, не змінювалися б не топологія, ні матриця вимог, то цього б не відбувалося. Інша справа – оверлейна мережа з динамічними параметрами, тут від

віддаленості компонентів від вузла, що виконує розрахунок залежить ймовірність точної оцінки їх стану. З відділеністю ймовірність знижується. Оцінка точності рішення в залежності від віддалення від вирішального вузла в атестаційній роботі не розглядається та є предметом подальших досліджень.

Отже, значення $G(u_i, R)$ називається околицею, або областю ефективної маршрутизації вузла v_i в оверлейній мережі IaaS-інфраструктури і має таке позначення: $R(v_i)$.

Слід зазначити, що вимога симетричності каналів зв'язку, сформульована раніше, не є необхідною для автоматизованої побудови VPN-ланцюга. Насправді, завжди можна вважати всі канали в оверлейній мережі односпрямовані з певними характеристиками, а двонаправлені канали зв'язку замінюються лише парою. Асиметричність ніяк не впливає на спосіб збору та передачі маршрутної інформації, оскільки вона стосується тільки одного напрямку руху: від конкретного віртуального маршрутизатора до інших вузлів, а не навпаки. Складність представляє тільки метод вимірювання відповідних метрик в асиметричній оверлейній мережі, однак, ця задача не розглядається в атестаційній роботі та є предметом подальших досліджень.

4 ДОСЛІДЖЕННЯ ДИНАМІКИ ПОБУДОВИ МАРШРУТІВ VPN-ЛАНЦЮГІВ В УМОВАХ НЕПОВНИХ ДАНИХ IAAS

Перед описом окремих процедур, виконуваних VPN-маршрутизатором в IaaS-інфраструктурі в процесі роботи, коротко опишемо схему алгоритму поставлених досліджень. У число основних функцій VPN-маршрутизатора входять наступні: початкова ініціалізація і побудова області ефективної маршрутизації, оновлення маршрутної інформації, підготовка до скидання і оповіщення сусідів, передача призначеного для користувача трафіку. Перехід алгоритму в той чи інший режим здійснюється при виникненні якої-небудь події, тобто алгоритм подієво-керований.

Початкова ініціалізація проводиться один раз при включенні маршрутизатора в мережу і запуску алгоритму. В цьому режимі початково є тільки інформація про наявні мережні інтерфейси запущеного VPN-вузла (по суті множина ребер інцидентних до вузла графа, який розглядається). На підставі заданої величини через безпосередніх сусідів проводиться побудова області ефективної маршрутизації.

В процесі роботи маршрутизатора при виникненні змін, пов'язаних з топологією або завантаженням каналів і вузлів, виконується процедура поновлення, яка полягає в коригуванні зовнішньої і внутрішньої маршрутної інформації, на підставі якої проводиться вибір маршрутів і оповіщення всіх вузлів, що належать, про цю подію. Кожному маршрутному запису зіставляється цілочисельний параметр, що характеризує її вік. У момент створення або поновлення запису цей параметр приймає значення 0. Через фіксовані проміжки часу (в моменти спрацьовування програмного таймера), вік запису збільшується. При досягненні деякого заданого порогу поновлення, відповідний маршрутний запис повинен бути оновлений або видалений з пам'яті вузла.

Механізм поновлення різний для зовнішньої і внутрішньої маршрутної інформації та детально розглядається в цьому розділі атестаційної роботи.

У тих випадках, коли необхідно виконати заплановане вимкнення VPN-маршрутизатора або виключення його з розподіленої процедури пошуку маршрутів, для забезпечення коригування маршрутної інформації на зацікавлених вузлах, до того, як виникне відмова і буде запущено оновлення щодо відмови, проводиться розсилка по області ефективної маршрутизації з оповіщенням про цю подію. Після цього VPN-маршрутизатор вимикається, а VPN-ланцюг перебудовується згідно алгоритму, який наведено у розділі 2.

Основним завданням будь-якого алгоритму маршрутизації є побудова маршрутів для передачі користувальницького трафіку. В даному випадку, маршрутизація здійснюється двома способами, в залежності від того, чи є трафік транзитним через область ефективної маршрутизації або VPN-вузол призначення їй належить цій області. У першому випадку проводиться вибір найкращого вузла, який повинен будувати наступну ділянку маршруту, а в другому – здійснюється передача даних.

Для передачі даних всередині по кожній ділянці маршруту використовується маршрутизація від джерела, що має на увазі повне визначення маршруту в джерелі. Кожен проміжний VPN-вузол ніяк не аналізує напрямок чергового переходу, а просто здійснює доставку наступного VPN-вузла або відправляє джерела повідомлення про помилку, якщо це неможливо.

Всі функції маршрутизатора можна розділити на дві категорії: функції пов'язані з передачею призначеного для користувача трафіку і функції побудови шляхів доставки. У число останніх входить підтримка актуальності інформації про структуру оверлейної мережі. Від того, наскільки дана інформація відповідає реальному стану оверлейної мережі, залежить ефективність маршрутизатора при передачі користувальницького трафіку. У ряді випадків (наприклад, відразу після включення) знань VPN-маршрутизатора недостатньо, щоб виконувати побудову шляхів. Отже,

виділимо основні режими функціонування VPN-маршрутизаторів і відповідні їм стани:

а) побудова $R(v_i)$. VPN-маршрутизатор v_i переходить в режим побудови області ефективної маршрутизації відразу після включення. Інші вузли не можуть йому делегувати побудова ділянки маршруту і не повинні використовувати v_i для транзитної передачі в своїх маршрутах. Після закінчення побудови $R(v_i)$ переходить в режим маршрутизації;

б) маршрутизація. В цьому режимі VPN-маршрутизатор виконує тільки вибір маршрутів і передачу призначеного для користувача трафіку по ним, на підставі тільки тієї інформації, якою він володіє. Після закінчення фіксованого тайм-ауту, виконується оновлення $R(v_i)$;

в) оновлення $R(v_i)$. В даному режимі проводиться обмін короткими HELLO-повідомленнями з усіма вузлами оверлейної мережі для перевірки їх працездатності та вимірювання продуктивності і завантаження каналів зв'язку. Призначений для користувача трафік передається через VPN-маршрутизатор і цьому VPN-маршрутизатору може делегуватися побудова ділянок маршрутів;

г) оновлення при відмові $R(v_i)$. VPN-маршрутизатор переходить в режим поновлення після виникнення збою при передачі користувальницького трафіку всередині $R(v_i)$. Даний режим подібний до режиму побудови $R(v_i)$. Інші вузли можуть використовувати в своїх маршрутах, якщо не мають інших альтернатив доставки користувальницького трафіку. Вузол v_i, при неможливості негайної передачі, повинен виконувати його буферизацію;

ґ) оновлення зовнішніх даних $R(v_i)$. Вузол v_i переходить в режим поновлення зовнішніх даних після закінчення фіксованого тайм-ауту. В цьому режимі формуються і відсилаються запити на оновлення до всіх прикордонним (граничним) маршрутизаторам $v_b \in {}^{(i)}V_b$ для яких є маршрутні

записи, які досягли порога поновлення. Вузол v_i бере участь в передачі користувальницького трафіку;

д) підготовка до скидання. В даному режимі вузлом v_i проводиться сповіщення вузлів, що належать $R(v_i)$ про відключення, передається користувальницький трафік, накопичений в буферах (якщо такий є). Інші маршрутизатори повинні скорегувати свої внутрішні шляхи доставки, виключивши ті, які містять v_i.

Перераховані режими роботи VPN-маршрутизаторів в VPN-ланцюгах відповідають наведеним другому розділі положенням алгоритму, за винятком доповнень, пов'язаних з побудовою та підтримкою цілісності уявлення.

Далі необхідно дослідити процес автоматизованої побудови VPN-ланцюгів. Початкова побудова $R(v_i)$ проводиться вузлом безпосередньо після виконання ініціалізації після включення v_i. По всім активним каналам надсилається запит на формування $R(v_i)$. Повідомлення повинно містити:

- GUID – унікальний ідентифікатор маршрутизатора;
- $I(v_i)$ – множина активних інтерфейсів;
- t_{msg} – часова мітка повідомлення;
- $Cfg(v_i)$ – конфігураційні параметри (в тому числі R_i);
- R_c – поточне значення метрики (в разі використання в якості метрики числа проміжних переходів – це номер переходу). Це поле інкременується кожним вузлом до тих пір, поки не буде досягнуто значення R_i. При $R_c < R_i$ вузол, який отримав повідомлення є внутрішнім для v_i, при $R_c < R_i$ – граничним.

Кожен вузол $v_j \in R(v_i)$, який отримав такий запит, в залежності від того є він для v_i граничним або внутрішнім, відправляє у відповідь повідомлення, що містить інформацію про нього.

У відповідь повідомлення повинно містити:

- *GUID* – унікальний ідентифікатор VPN-маршрутизатора;

- *isBound* – ознака граничності VPN-вузла, який відповів;
- $I(v_j)$ – множина безліч активних інтерфейсів v_j;
- t_{msg} – часова мітка повідомлення;
- $Cfg(v_j)$ – конфігураційні параметри v_j;
- $R(v_j)$ опис області ефективної маршрутизації v_j;
- $\tilde{M}_j$ маршрутна таблиця для вузлів не належать множині $R(v_j)$.

Кожен вузол v_j, який отримав повідомлення із запитом на формування $R(v_j)$, використовує наявну в ньому інформацію для поновлення своїх маршрутних таблиць. Належність вузла v_i області $R(v_j)$ визначається аналізом значень R_c і R_i.

У свою чергу, вузол v_i є ініціатором запиту на формування $R(v_i)$ на підставі відповідних повідомлень формує свої таблиці ${}^{(i)}\tilde{M}$ і ${}^{(i)}M$. Після закінчення заданого часу очікування відповідей, вважається що процедура побудови ОЕМ завершена і VPN-вузол переходить в режим маршрутизації.

Проаналізуємо процедуру перевірка стану $R(v_i)$. Для перевірки стану каналів і вузлів, а також збору інформації про їх завантаженості використовується віялове поширення тестового повідомлення *HELLO*, у відповідь на яке отримавший його вузол відповідає аналогічним повідомленням. Розсилка повідомлень є тригерною і ініціалізується після закінчення заданого інтервалу t_{HELLO}. Установка початкового значення таймера виконується з урахуванням характеристики динамічності $R(v_i)$.

Маршрутизація такого повідомлення виконується від джерела, тобто в повідомлення поміщається повний шлях його доставки на підставі інформації, що міститься в маршрутній таблиці ${}^{(i)}M$ вузла v_i, який його ініціював. Якщо який-небудь вузол $v_j \in R(v_i)$ виявляється недосяжний, то v_i отримує повідомлення про неможливість доставки по маршруту. Після його отримання v_i переходить в режим поновлення при відмові і переходить далі

для побудови VPN-ланцюга на наступному вузлі IaaS. Область поширення *HELLO*-повідомлень обмежена константою, яка залежить від структури мережі.

Граничне значення таймера оновлення можна змінювати в залежності від динаміки відповідної ділянки оверлейної мережі. Чим вище показник динамічності, тим менше граничне значення повинен мати цей таймер. Вибір порогового значення проводиться або експертом-адміністратором, який виконує конфігурацію мережі, або автоматично на основі вимірювання показника λ.

Проаналізуємо процес оновлення VPN-ланцюга при відмові вузлів. Процедура оновлення при відмові виконується всякий раз, коли маршрутизація від джерела всередині $R(v_i)$ була завершена невдало. Ініціює запуск даної процедури вузол v_i, який отримав повідомлення про неможливість доставки за вказаним ним маршрутом. Причиною можуть виступати: відмова каналу або вузла, надмірна завантаженість, зміна топології. У будь-якому випадку проводиться оновлення таблиць ${}^{(i)}\tilde{M}$ і ${}^{(i)}M$, що містять відомості про досяжні вузли і маршрути, а також негайна відправка інформації з новим поданням всім $v_j \in R(v_i)$ по побудованому дереву доставки.

Механізм обміну повідомленнями при даному оновленні подібний до того, який використовується для початкового побудови $R(v_i)$. Однак, слід зазначити, що в даному випадку вузол v_i продовжує передавати дані, якщо це можливо, або буферизує їх в пам'яті, до відновлення режиму маршрутизації.

На малюнку 4.1 приведена послідовність поширення інформації про відмову при передачі по маршруту всередині області ефективної маршрутизації. Штрихування означає область оверлейної мережі, в якій відомо про відмову.

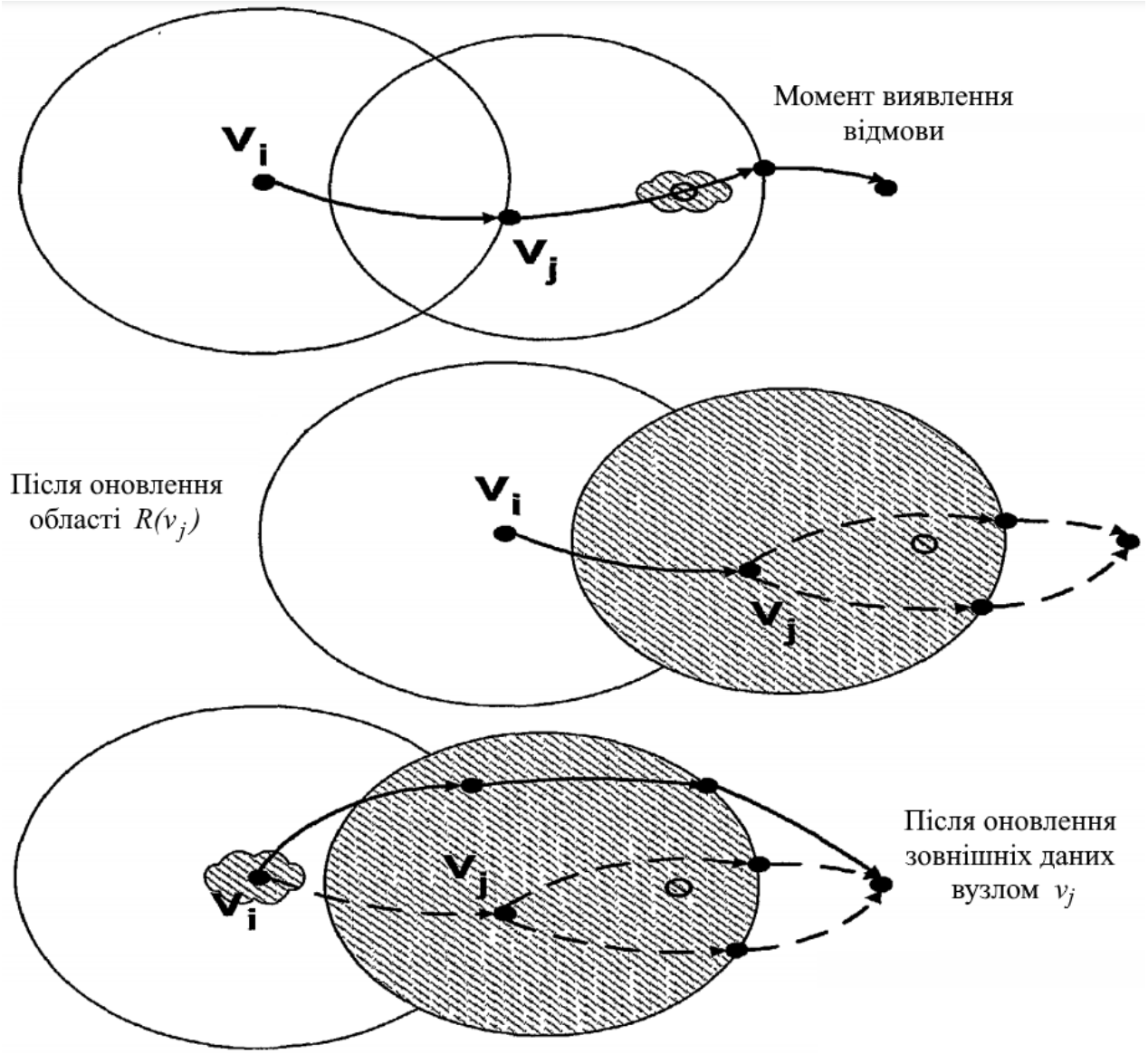

Рисунок 4.1 – Оновлення схеми після відмови

У першому випадку про це дізнається проміжний VPN-маршрутизатор, який бере участь тільки в передачі даних. Після повідомлення про неможливість передачі джерела відповідного маршруту (в даному випадку це v_i) запускається процедура поновлення при відмові і інформація поширюється в області ефективної маршрутизації цього вузла. Джерело повідомлення дізнається про зміну характеристик маршруту після чергового оновлення за таймером. На останньому зображенні рисунка 4.1 приведено відповідний стан вузлів. Слід відзначити, що тут розглядалися тільки

учасники конкретної передачі даних за певним маршрутом. У загальному випадку, всі вузли виконують поновлення за таймером і отримують відповідну інформацію.

Далі виконаємо аналіз процедури синхронізації часу. В історії розвитку комп'ютерних мереж існують приклади використання навігаційних супутникових систем в задачах адресації і маршрутизації в комп'ютерних мережах. Наприклад, в [10] описаний експеримент використання географічних координат і часу, отриманих за допомогою системи GPS, для присвоювання окремим вузлам мережних адрес і оптимізації процедур передачі користувальницького трафіку. В даний час вартість пристроїв прийому сигналів навігаційних систем порівнянна з серійними комплектуючими, а розміри настільки малі, що дозволяють виготовляти подібні пристрої у вигляді плат розширення персональних комп'ютерів. Зовсім інша ситуація з оверлейною мережею, де неможливо ідентифікувати, де знаходяться сегменти IaaS-інфраструктури. Тому тут застосовується дещо інший метод визначення умовних координат віртуального простору [11]. Він дозволяє використовувати єдиний час для ідентифікації виникнення подій.

Кожен запис в зовнішній і внутрішній маршрутній таблиці забезпечується тимчасовим ідентифікатором. Даний ідентифікатор, з одного боку дозволяє визначити інтервал часу, що пройшов з моменту створення маршрутної записи (її вік), а з іншого – може використовуватися для запобігання поширенню застарілої маршрутної інформації. У загальному випадку, кожний маршрутний запис в пам'яті VPN-маршрутизатора порівнюється з часовою міткою, що означає момент останньої успішної операції з її використанням. Як тільки значення тимчасової мітки досягає деякого граничного значення, маршрутний запис знищується як застарілий.

Далі в описується алгоритм поновлення зовнішніх маршрутних записів з використанням єдиного часу і аналізується його ефективність.

Первинне уявлення про зовнішні вузли і параметри доставки до них інформації формується при ініціалізації основного алгоритму, коли

здійснюється побудова $R(v_i)$. У таблицю з зовнішніми маршрутами імпортуються відомості, надані граничними VPN-маршрутизаторами. Слід зазначити, що для формування ${}^{(i)}\tilde{M}$ використовуються тільки граничні VPN-маршрутизатори.

Після побудови $R(v_i)$ і відповідних таблиць ${}^{(i)}\tilde{M}$ і ${}^{(i)}M$ проводиться установка інтервалу спрацьовування таймера ${}^{(i)}t_{upd}$, який використовується в такий спосіб: після закінчення інтервалу очікування до кожного граничного VPN-маршрутизатора $v_i \in {}^{(i)}V_b$ надсилається запит на оновлення зовнішніх маршрутних даних. Припустимо, що кількість змін, що відбулися на кожному v_i за інтервал часу ${}^{(i)}t_{upd}$ така, що інформація про них може бути поміщена всередину одного повідомлення. Тоді кількість згенерованих повідомлень при кожному оновленні зовнішніх даних складе $2 \bullet {}^{(i)}N_b$ де ${}^{(i)}N_b$ – число граничних VPN-маршрутизаторів для вузла v_i.

Формат запиту, що надсилається, повинен передбачати наявність поля, що містить час останнього оновлення зовнішніх маршрутних записів. Вузол, який отримав таке повідомлення, відправляє назад інформацію про всі вузли, час зміни яких лежить в проміжку між зазначеним часом в запиті і часом його отримання. Вузол v_i, який отримав відповідь на запит від граничного VPN-маршрутизатора, виключає з отриманого повідомлення всі записи, що описують шляхи до вузлів, які належать $R(v_i)$. Решта записів поміщається в маршрутну таблицю ${}^{(i)}\tilde{M}$. У тому випадку, коли в ${}^{(i)}\tilde{M}$ уже є записи, що відповідають тим, які були отримані при оновленні, – вони замінюються на нові.

Описаний механізм дозволяє реалізувати передачу тільки тих відомостей, які є дійсно новими, не передаючи для порівняння всю топологічну інформацію між вузлами, яка у IaaS-інфраструктурі досягає значного обсягу. Ключовим моментом в цьому алгоритмі є наявність загальних часових міток в єдиній часовій шкалі. Ще одним важливим

методом зменшення обсягів розсилки топологічних даних, є використання VPN-маршрутизаторів по-замовчуванню, що дозволяє, з одного боку обмежити область пошуку зовнішніх вузлів, а з іншого – гарантувати, що зовнішній трафік, спрямований до істотно віддалених вузлів призначень буде передаватися по захищеним каналам передачі даних (наприклад, реалізуючи концепт вкладених багатошарових VPN-тунелів).

Отже, далі доцільно розглянути процедуру підготовки до скидання. При плановому відключенні або перезавантаженні VPN-маршрутизатора необхідно сповістити вузли усередині $R(v_i)$, щоб знизити витрати на корекцію маршрутних даних на них. При цьому надсилається всім вузлам, що належить, повідомлення, що містить інформацію про VPN-маршрутизатор, його уявленні $R(v_i)$ і параметри функціонування. Вузли, що отримали таке повідомлення, коригують свої маршрутні таблиці. Якщо результати такого коригування зачіпають область ефективної маршрутизації отримав повідомлення вузла v_i, то дане повідомлення транслюється всім тим вузлам, які не присутні в поданні $R(v_i)$, але присутні в $R(v_j)$.

Далі розглянемо процедуру передачі даних в VPN-ланцюгу. Розглянемо можливі ситуації.

Ситуація 1 $v_{dst} \in R(v_i)$. Вибирається $\rho(v_i, v_{dst}) \in {}^{(i)}M$ і виконується маршрутизація від джерела, тобто v_i сам формує весь шлях і записує його всередину переданого повідомлення, а проміжні вузли не беруть участі в його побудові. Якщо v_{dst} є недосяжним по побудованому маршруту, то вузол, який це виявив, відправляє вузлу v_i повідомлення про помилку.

Таким чином:

- v_i має найкращий шлях до v_{dst};
- v_i має зворотний зв'язок, на підставі якого судить про адекватність своїх знань про область $R(v_i)$;
- жоден з вузлів $v_j \in R(v_i), v_i \neq v_j$ не несе навантаження по обчисленню

маршруту v_i від v_{dst}.

Ситуація 2 $v_{dst} \notin R(v_i)$. Для передачі корисних даних у зовнішню область виконуються наступні кроки.

- вибір з ${}^{(i)}\tilde{M}$ найкращого посередника $v_p \in {}^{(i)}V_b$ відповідно до раніше описаного критерію, для делегування йому подальшої побудови маршруту;

- вибір маршруту від v_i до v_p і інкапсуляція корисних даних в повідомлення, що відправляється вузлу v_p;

- передача сформованого повідомлення від вузла v_i до вузла v_p відповідає ситуації 1, яка описана вище.

Нехай для вимоги, яка надійшла в вузол v_i на передачу даних необхідно виконати маршрутизацію, використовуючи в якості посередників вузли v_j та v_k. Припустимо, що всередині $R(v_j)$ відбулася подія, яка зруйнувала маршрут, який з'єднує вузли v_j та v_k. Виявивший цей факт вузол, що знаходиться в межах $R(v_j)$ відправить вузлу v_j повідомлення про помилку. Вузол v_j, отримавши таке повідомлення, виконає буферизацію призначених для користувача даних і перейде в режим поновлення при відмові. Після завершення оновлення, маршрутизація призначених для користувача даних поновлюється: буде знайдений новий шлях до вузла v_k, або обраний інший граничний VPN-маршрутизатор-посередник. Вузол v_i дізнається про зміну параметрів маршруту тільки після виконання чергового оновлення зовнішніх маршрутних даних, надіславши запит вузлу v_j.

На рисунку 4.2 приведена схема передачі даних від вузла *S* до вузлів призначення D_i. З неї видно, що маршрут до вузла D_1, який знаходиться всередині області ефективної маршрутизації вузла-джерела, формується самим джерелом. У тому випадку, коли вузол призначення знаходиться поза ОЕМ (наприклад, вузол D_1), запускається розподілена процедура

формування маршруту і відповідні його ділянки формують вузли-посередники.

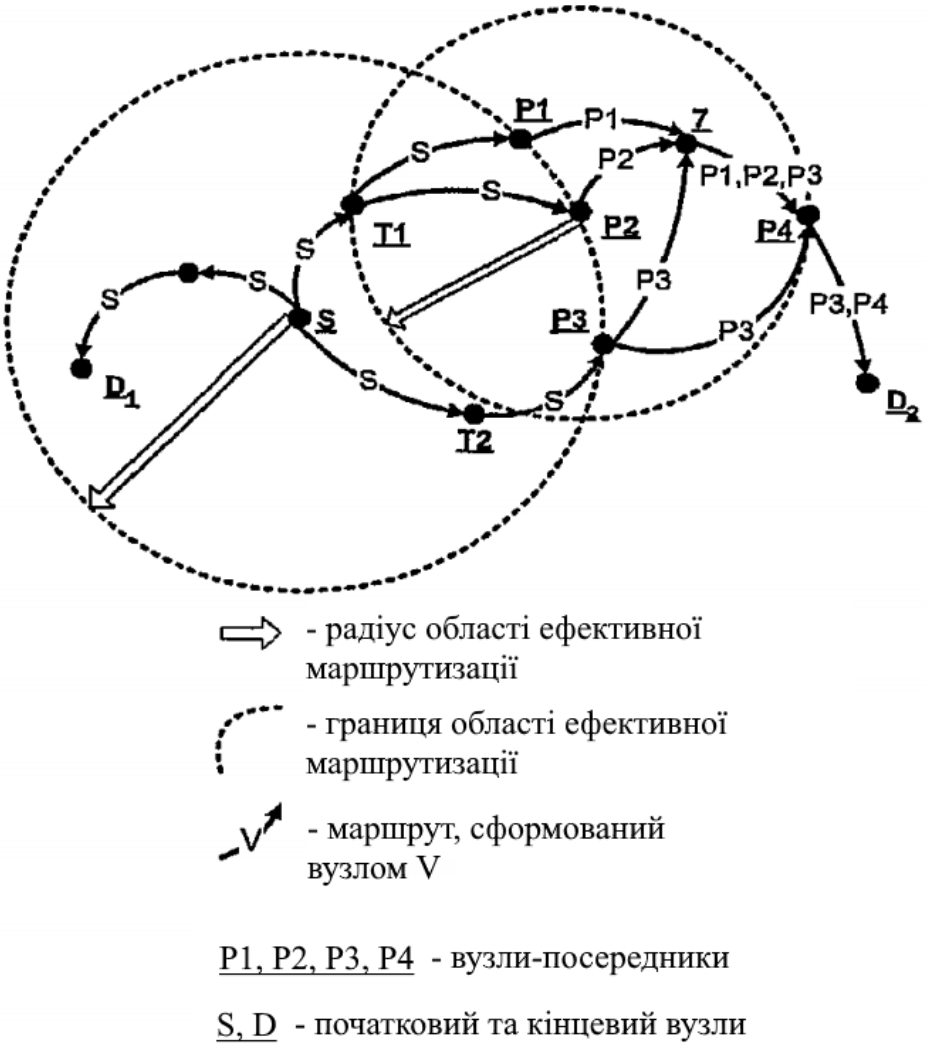

Рисунок 4.2 – Схема передачі даних

Таким чином, кожен VPN-маршрутизатор буде обізнаний про зміну. Слід зазначити, що швидкість поширення маршрутної інформації відповідає її значимості для працездатності алгоритму маршрутизації на вузлах. Так вузол v_j отримує інформацію про порушення зв'язності всередині негайно і відразу ж виконує оновлення. На відміну від вузла v_j вузол v_i дізнається про зміну параметрів маршрутів (які були викликані перерозподілом навантаження на канали зв'язку і вузли усередині $R(v_j)$), виконуючи чергове

оновлення таблиці ${}^{(i)}\tilde{M}$. Адекватність знань про топології істотніша, ніж знання про навантаження на канали і вузли. Описаний підхід дозволяє при передачі даних від джерела до одержувача прогнозувати ефективність подальшої передачі, як би «заглядаючи вперед». З іншого боку, оперативність поширення маршрутної інформації узгоджена з її значимістю для вузлів.

Важливим критерієм оцінки ефективності алгоритмів маршрутизації є час збіжності алгоритму маршрутизації. Як правило під цим розуміється час, необхідний для поширення інформації про одиничну зміну в структурі маршрутизованої мережі на базі IaaS-інфраструктури серед усіх вузлів і повної синхронізації маршрутних таблиць.

Для описуваного методу поширення повідомлень про зміни час збіжності всередині OEM кожного вузла не дуже важливе і реакція вузлів на зміни відбувається практично миттєво (в силу невеликих розмірів OEM). Цей ефект досягається за рахунок зниження часу збіжності по всій оверлейній мережі. Як було відзначено раніше, в віртуальних комп'ютерних мережах більш ефективним є досягнення динамічної рівноваги локальних областей.

І насамкінець, важливим питанням є аналіз показника обсягу службових даних.

Обсяг службового трафіку – один з найважливіших критеріїв оцінки алгоритмів маршрутизації. У розробленому алгоритмі весь службовий трафік можна розділити на дві категорії: обмін інформацією про області ефективної маршрутизації і обмін інформацією про віддалені ділянки оверлейної мережі. По суті, точність і оперативність доставки інформації другої категорії не потрібна, так як вона використовується лише для визначення найбільш вигідного напрямку подальшої передачі. Дійсно, при зміні стану деякої віддаленої підсистеми $R(v_i)$ часто, більшість маршрутів через $R(v_i)$ може бути відновлена за рахунок наявних ресурсів і загальний напрямок передачі користувальницького трафіку не зміниться.

Якщо ж втрачено будь-який «важливий» елемент в рамках всієї мережі, то це спричинить серію відмов при спробі передати призначений для користувача трафік через області, нездатні його передати. Ці відмови спричиняють за собою масові поновлення внутрішньої маршрутної інформації, і якщо, це можливо, буде відновлений новий режим передачі за втраченими напрямками, або, якщо це принципово недосяжно, даний напрямок буде виключено з маршрутних таблиць.

Отже, при такій схемі реалізації обміну маршрутною інформацією, необхідно мінімізувати зовнішній службовий трафік, так як він не може бути використаний для відновлення працездатності при відмовах, так як не несе інформації про зв'язність ділянок IaaS-фнраструктури. Поширення зовнішньої маршрутної інформації виконується за таймером, причому ініціалізує процес обміну вузол, зацікавлений в отриманні такої інформації.

Далі доцільно розглянути деякі характеристики внутрішнього службового трафіку між вузлами оверлейної мережі.

Вузол v_i, виявивши зміну стану $R(v_i)$, що викликане втратою працездатності деякого $v^* \in R(v_i)$, розсилає про це повідомлення кожному $v_j \in R(v_i), i \neq j$. Далі, в залежності від того, чи перебуває v^* в $R(v_j)$ виконується оновлення $R(v_j)$ і подальша розсилка повідомлення. Розсилка виконується тільки для тих повідомлень, які несуть інформацію про елементи області ефективної маршрутизації того вузла, який виконує його передачу.

Таким чином, підсумовуючи проведене дослідження отримано наступне твердження: радіус поширення повідомлення про зміну при відмові в області $R(v_i)$ ініційованого вузлом v_i, не перевищує подвоєного значення $R_{\max}$, де

$$R_{\max} = \max_{v_i \in V} R_j \qquad (4.1)$$

Нехай $v_m \in G$ вузол, який отримав повідомлення. Для $v_m \in R(v_i)$ виконання умови очевидне, оскільки $d(v_i, v_m) \le R_i$ за визначенням.

Нехай $v_m \notin R(v_i)$, тоді повідомлення про оновлення надходить (для доставки використовується маршрутизація «від джерела») від деякого вузла v_i, такого що $v*,\ v_m \in R(v_i)$. Отже, відстань визначається як:

$$\begin{cases} d(v*, v_j) \le R_j \\ d(v_j, v_m) \le R_j \end{cases}. \tag{4.2}$$

У свою чергу:

$$d(v_i, v*) \le R_i. \tag{4.3}$$

Враховуючи (3.6):

$$d(v_i, v_m) \le d(v_i, v_j) + d(v_j, v_m), \tag{4.4}$$

або з урахуванням (4.3):

$$d(v_i, v_m) \le 2 \times R_{\max}. \tag{4.5}$$

Нехай область, вузли якої задіяні в передачі повідомлення при відмові, матиме назву «зона відмови». Радіус зони відмови обмежений в такому випадку обмежений $2 \times R_{\max}$.

Оцінимо швидкість поширення повідомлення про зміну при відмові всередині $R(v_i)$. При виникненні збою всередині $R(v_i)$ вузол, який виконує маршрутизацію від джерела, відразу запускає механізм оновлення внутрішніх і зовнішніх маршрутних таблиць, причому відразу після цього сповіщає про це всі вузли, що знаходяться зоні відмови. Як показано вище

максимальний радіус поширення не перевищує $2 \times R_{\max}$ і, отже, час оповіщення не перевищує величини $2 \times R_{\max} \times t_{\max}$, де $t_{\max}$ – час передачі повідомлення по найбільш повільному каналу у всій оверлейній мережі. На цьому поширення повідомлення всередині зони відмови завершується.

Надходження нової інформації в зовнішню, по відношенню до зони відмови, область визначається частотою спрацьовування таймерів оновлень на вузлах VPN-ланцюга, котрі належать до цієї області. З цієї причини швидкість розповсюдження оновлень і, отже, обсяг відповідного службового трафіку поза зоною відмови не залежать від інтенсивності змін, а визначаються тільки значеннями відповідних таймерів на вузлах оверлейної мережі IaaS-інфраструктури, кількістю вузлів і розмірами області ефективної маршрутизації.

Таким чином, наведені вище механізми дозволяють виконувати автоматизовану перебудову VPN-ланцюгів в середовищі IaaS.

ВИСНОВКИ

В атестаційній роботі роботи поставлена і успішно вирішена задача розробки методу автоматизованої побудови VPN-ланцюгів на платформі IaaS. Зокрема:

- показано, що ефективність використання VPN-ланцюгів досягає максимального ефекту при 4-6 VPN-серверів в ланцюжку;

- частота перебудови ланцюжка в одну годину дозволяє істотно підвищити безпеку з'єднання між терміналом і хостом;

- складність запропонованого методу носить логарифмічний характер за рахунок швидких способів сортування та пошуку мінімальних тимчасових значень в матрицях;

- експериментально показано, що зниження часової затримки обміну даними між хостом і терміналом через ланцюжок VPN-серверів становить до 15% в порівнянні з відомими методами, заснованими на ідеї NeuroRouting.

Запропоновано підхід до подання маршрутної інформації в VPN-ланцюгів при застосуванні великих динамічних оверлейни мереж передачі даних, які побудовані в IaaS-інфраструктурі. Він має ряд переваг в порівнянні з іншими алгоритмами:

- трудоємність збору інформації про топологію і властивості оверлейної мережі, яка експоненціально залежить від кількості контрольованих мережних елементів, може бути значно знижена за рахунок керування розмірами області ефективної маршрутизації;

- якість маршрутної інформації, використовуваної для пошуку маршрутів у внутрішній області, підвищується, що дозволяє використовувати маршрутизацію від хоста з високою ймовірністю успішної доставки, наслідком чого стане зниження навантаження на проміжні вузли VPN-ланцюга;

- можуть бути використані незалежні розподілені процедури побудови маршрутів всередині всієї мережі.

При використанні різного представлення зовнішньої і внутрішньої маршрутної інформації необхідно визначати специфічні критерії вибору напрямку передачі трафіку в зовнішню область. Ці критерії повинні враховувати відстань до вузла-посередника у внутрішній області, стан та ефективність функціонування вузла-посередника, відстань до зовнішнього вузла від вузла-посередника, динаміку маршруту від вузла-посередника до вузла призначення. Задача вибору маршруту для зовнішньої області можна розглядати як задачу багатокритеріального вибору.

Для забезпечення автоматизованої побудови VPN-ланцюга, VPN-маршрутизатори повинні підтримувати наступні режими: побудова $R(v_i)$, перевірка стану елементів $R(v_i)$, оновлення при відмові $R(v_i)$, оновлення зовнішніх даних $R(v_i)$, підготовка до скидання $R(v_i)$. Для кожного з цих режимів роботи передбачена відповідна процедура в алгоритмах маршрутизації. Для мінімізації витрат на синхронізацію подій в системі необ-ходимо використовувати зовнішні джерела точного часу.

В роботі запропоновано алгоритм, який на основі відповідних критеріїв вибору маршрутів і процедур, дозволяє гарантувати швидку зходимість в локальних областях і обмеження поширення службової інформації в зовнішню область. Це дозволяє знизити витрати на передачі службового трафіку і підвищити реактивність VPN-маршрутизаторів за рахунок розподіленого обчислення маршруту.

Новизна даної атестаційної роботи полягає в тому, що вперше при побудові VPN-ланцюгів, з метою зниження складності задачі (і, як наслідок, часу її рішення) було використано сукупність комбінаторних підходів, що дають логарифмічну складність завдання і дозволяють за більш менший час побудувати VPN-ланцюг для підвищення рівня захищеності середовища обміну даними в глобавльній мережі між терміналом та хостом.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Vitalii Tkachov, Volodymyr Tokariev, Iryna Ilina, and Stanislav Partyka, "Modified Traveling Salesman Problem for a Group of Intelligent Mobile Objects and Method for Its Solving," International Journal of Electrical and Electronic Engineering & Telecommunications, Vol. 10, No. 1, pp. 1-7, January 2021. Doi: 10.18178/ijeetc.10.1.1-7.

2. Ткачов В.М. Разработка алгоритма мультиагентного управления группой мобильных «s-bot» / В.М. Ткачов, В.В. Токарєв, Г.І. Чурюмов // Реєстрація, зберігання і обробка даних. – Київ: Інститут проблем реєстрації інформації НАН України, 2019. – № 1. – Т. 21. – С. 46-56.

3. Ткачов В.М. Метод передачі даних в комп'ютерній мережі проміжного зберігання даних складної інформаційної системи / В.М. Ткачов // Системи управління, навігації та зв'язку. – Полтава: Полтавський національний технічний університет ім. Ю. Кондратюка, 2017. – № 3 (43). – С. 117-119.

4. Ткачов В.М. Проблема передачі даних типу Big Data у мобільній системі «мультикоптер-сенсорна система» / В.М. Ткачов, В.В. Токарєв, В.О. Радченко, В.О. Лебедєв // Системи управління, навігації та зв'язку. – Полтава: Полтавський національний технічний університет ім. Ю. Кондратюка, 2017. – № 2 (42). – С. 154-157.

5. Vitalii Tkachov, Anna Budko, Kateryna Hvozdetska and Daryna Hrebeniuk. Method of Building Dynamic Multi-hop VPN Chains for Ensuring Security of Terminal Access Systems // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T): Kharkiv 06-09 oct. 2020, Kharkiv.

6. Ткачов В.М. Аналіз методів забезпечення відмовостійкості оверлейних мереж / В.М. Ткачов, К.П. Гвоздецька // Проблеми інформатизації : тези доп. 8-ї міжнар. наук.-техн. конф., 26-27 листопада 2020

р., м. Черкаси, м. Харків, м. Баку, м. Бельсько-Бяла. Т. 1 / Черк. держ. технолог. ун-т [та ін.]. – Харків, 2020. – С. 44.

7. Tkachov, V., Bondarenko, M., Ulyanov, O., & Reznichenko, O. (2019, December). Overlay Network Infrastructure for Remote Control of Radio Astronomy Observatory. In 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT) (pp. 161-165).

8. Tkachov V.M. Architecture of Overlay Network with Nested VPN Tunneling / V. Tkachov, M. Bondarenko, M. Hunko // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: матеріали десятої міжнародної науково-технічної конференції. – Баку: ВА ЗС АР; Харків: ДП «ХНДІ ТМ»; Жиліна: УмЖ, 2020. – С. 36.

9. Tkachov V. Technology of Load Balancing in Anonymous Network Based on Proxy Nodes Cascade Platform / V. Tkachov, M. Hunko, M. Bondarenko, S. Artyomov // Четверта міжнародна науково-технічна конференція «Комп'ютерні та інформаційні системи і технології». Збірка наукових праць. Харків: ХНУРЕ. – 2020. – С. 82.

10. Ткачов В.М. Програмний кластер для паралельної обробки великих обсягів даних / В.М. Ткачов, Ю.А. Кривобоков, К.П. Гвоздецька // Міжнародна наукова інтернет-конференція "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 49)" / Збірник тез доповідей: випуск 19 (м. Тернопіль, 10 червня 2020 р.). – Тернопіль. – 2020. – 31-33 с.

11. Ткачов В.М. Застосування технології OpenVPN в рамках сервісу «Health Tracker» / В.М. Ткачов, А.О. Карасьов // 73-я науково-технічна конференція професорсько-викладацького складу, науковців, аспірантів та студентів ОНАЗ ім. О.С. Попова, 12-14 грудня 2018 року. – Одеса. – 2018. – С. 157-158.

12. Ткачев В.Н. Применение метода предотвращения коллизий при параллельной обработке данных в полинговых сетях контроля состояния сложных распределенных систем / В.Н. Ткачев, А,А. Коваленко, В.О.

Лебедев // Третя міжнародна науково-технічна конференція «Проблеми інформатизації» 12-13 листопада 2015 року. – Черкаси–Баку–Бельсько-Бяла–Полтава. – 46 с.

13. Коваленко А. А., Кучук Г. А. Оптимальное управление трафиком мультисервисной сети на основе методов последовательного улучшения решений //Системи озброєння і військова техніка. – 2016. – №. 3. – С. 59-63.

14. Кринкин К.В. Создание алгоритмов маршрутизации в динамических компьютерных сетях с использованием неполных данных / Дисс. … С.-Питербург, 2014 г. – 147 с.

15. Ткачов В.М. Питання кібербезпеки при впровадженні IAAS-рішень хмарних вендорів / В.М. Ткачов, С.О. Партика // Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі. – Матеріали першої міжнародної науково-практичної конференції. – Харків, НТУ «ХПІ». – 2016. – с. 38.

16. Tkachov V.M. Providing information security in systems of business process management in the IAAS-vendor environment / V.M. Tkachov, S.O. Partyka, V.O. Lebediev // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: матеріали шостої міжнародної науково-технічної конференції. – Полтава: ПНТУ; Баку: ВА ЗС АР; Кіровоград: КЛА НАУ; Харків: ДП «ХНДІ ТМ», 2016. – С. 25.

17. Ткачев В.Н. Метод передачи больших массивов данных с использованием SaaS-маршрутизаторов в оверлейных сетях / В.Н. Ткачев, В.О. Лебедев // Перша міжнародна науково-технічна конференція «Комп’ютерні та інформаційні системи і технології». Збірка наукових праць. Харків: ХНУРЕ. – 2017. – С. 31.