УДК 004.051:004.7

DOI: 10.31673/2412-9070.2024.060456

С. В. ПРОКОПОВ, канд. техн. наук, доцент,
ORCID: 0009-0005-2251-9473
С. О. СЄРИХ, канд. техн. наук, доцент,
ORCID: 0009-0008-6033-3225
М. М. ГНІДЕНКО, аспірант;
ORCID: 0009-0008-4450-6472
О. В. ВИШНІВСЬКИЙ, аспірант,
ORCID: 0009-0008-0209-9549
Державний університет інформаційно-комунікаційних технологій, Київ

ПРОБЛЕМИ, ВИРІШЕННЯ ЯКИХ ВПЛИВАЮТЬ НА ФУНКЦІОНАЛЬНУ СТІЙКІСТЬ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ (SDNs)

SDN змінює майбутнє мереж і імпортує нові інновації. Попит на послуги та використання мережі швидко зростає. Незважаючи на драйвери зростання, програмно-визначені мережі стикаються у своєму розвитку зі значними проблеми. Маючи на увазі зростання кількість проблем, підвищується актуальність дослідницьких ініціатив, спрямованих на подоланнях викликів, які ініційовані цілою низкою проблем функціонування програмно-визначених мереж. Вирішення цих проблем може суттєво плинути на функціональну стійкість SDN. Операторам потрібна ефективна, гнучка, та масштабована мережа у будь яких умовах функціонування. Серед проблем програмно-визначених мереж можна виділити такі як необхідність забезпечення належної надійності, масштабованості, продуктивності, розміщення контролерів та безпеки. У дослідженні проведений аналіз причин виникнення цих проблем та запропоновані можливі шляхи їх вирішення, що суттєво вплине на функціональну стійкість програмно-визначених мереж. В той же час, запропоновані шляхи є не остаточним вирішенням проблем, а лише відкриттям поля для нових глибоких досліджень.

Ключові слова: Програмно-визначені мережі (SDNs), контролер SDN, надійність, масштабованість, продуктивність, розміщення контролерів, безпека.

Вступ

SDN дозволяє користувачам і операторам мережі створювати простішу, настроювану, програмовану та керовану мережу. На думку спільноти мережевих дослідників, SDN змінює майбутнє мереж і імпортує нові інновації. SDN застосовується у центрах обробки даних, безпроводових мережах, програмно визначеного радіо, мережах підприємств і кампусів. Попит на послуги та використання мережі швидко зростає. Незважаючи на те, що драйвери зростання, такі як відеотрафік, великі дані та мобільні пристрої, збільшують доходи, вони створюють значні проблеми. Оператори мобільного та телекомунікаційного зв'язку стикаються з перевантаженням спектру, переходом на Інтернет-протокол (IP) і збільшенням кількості користувачів мобільного зв'язку. Одночасно оператори центрів обробки даних стикаються з величезним зростанням кількості серверів і віртуальних машин, що збільшує трафік між серверами. Маючи на увазі зростання кількість проблем, підвищується актуальність дослідницьких ініціатив, спрямованих на подоланнях цих викликів, які ініційовані цілою низкою проблем функціонування програмно-визначених мереж. Вирішення цих проблем може суттєво плинути на функціональну стійкість SDN. Операторам потрібна ефективна, гнучка, та масштабована мережа у будь яких умовах функціонування.

Постановка задачі

Ця стаття присвячена дослідженню проблем застосування SDN мереж. З появою хмарних обчислень багато парадигм екосистем і бізнесу стикаються з потенційними змінами та можуть

жуть скасувати процеси обслуговування IT-інфраструктури. Вимоги до високої доступності спонукали телекомунікаційні мережі прийняти нові концепції хмарної моделі: програмно-визначена мережа (SDN). Але мережі SDN стикаються з різними проблеми, від масштабованості і продуктивності до надійності та безпеки. Тому актуальним є обговорення можливих рішень цих проблем.

Аналіз останніх досліджень

Впровадження SDN мереж у даний момент є актуальною темою. Тому існує велика кількість досліджень щодо концептуальних підходів їх розвитку та розгортання [1,2]. У той же час при застосуванні програмно-визначених мереж існує ціла низки проблем, виникнення яких пов'язано як з особливостями архітектури, так і зі зростанням навантаження у сучасних умовах, такого як відеотрафік, великі дані, хмара та мобільні пристрої. Публікації присвячені розгляду таких проблем програмно-визначених мереж як забезпечення надійності [3], масштабованості [4,5], продуктивності [6], розміщення контролерів [7] та безпеки [8].

Метою роботи є забезпечення функціональної стійкості програмно визначених мереж (SDNs) у різних умовах роботи. Варто зазначити, що запроваджені схеми дозволяють частково визначити шляхи вирішення окремих проблем при відсутності загального підходу. Тому необхідно комплексне дослідження як переліку проблем, вирішення яких впливають на функціональну стійкість програмно-визначених мереж (SDNs), так і нових варіантів та підходів до їх вирішення.

Основний матеріал досліджень

Незважаючи на те, що SDN є багатообіцяючим рішенням для IT та хмарних провайдерів і підприємств, воно стикається з певними проблемами, які можуть перешкоджати його функціональній стійкості та реалізації в хмарних і безпроводових мережах. На рис. 1 наведено список проблем, вирішення яких впливають на функціональну стійкість програмно-визначених мереж (SDNs).

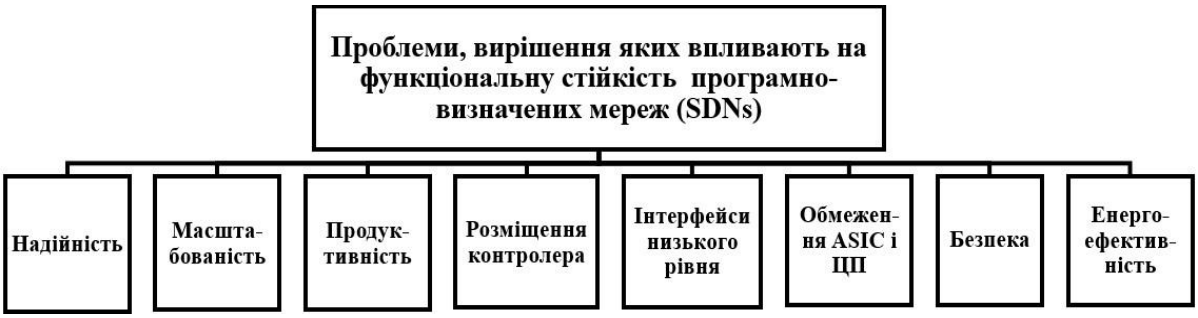

Рис. 1. Проблеми SDNs

Надійність. Контролер SDN повинен інтелектуально налаштовувати та перевіряти топології мережі, щоб запобігти помилкам і підвищити доступність мережі. Однак цей інтелект може бути пригнічений, що робить контролер схильним до єдиної точки відмови. У застарілих мережах, коли один або кілька мережевих пристроїв виходять з ладу, мережевий трафік направляється через альтернативні або сусідні вузли чи пристрої для підтримки безперервності потоку. Однак у архітектурі централізованого контролера (SDN) і за відсутності резервного контролера лише один центральний контролер відповідає за всю мережу. Якщо цей контролер виходить з ладу, вся мережа може зруйнуватися. Щоб вирішити цю проблему, IT-організації повинні зосередитися на використанні основних функцій контролера, які можуть підвищити надійність мережі. У разі збою зв'язку контролер SDN повинен мати можливість підтримувати багатошляхові рішення або швидке перенаправлення трафіку на активні канали.

Якщо контролер підтримує такі технології, як Virtual Router Redundancy Protocol (VRRP) і Multi-Chassis Link Aggregation Group (MC-LAG), це може сприяти підвищенню доступності мережі. У разі відмови контролера важливо, щоб контролер міг увімкнути кластеризацію двох

або більше контролерів SDN в режимі активного очікування; однак повинна підтримуватися синхронізація пам'яті між активним і резервним контролерами.

Можна показати, що централізована архітектура контролера реально може переривати мережевий трафік і потік потокових записів у разі відмови контролера. Можна запропонувати розподілену архітектуру, яка складається з набору менеджерів стійки (RM - Rack Managers), по одному на стійку, які діють як контролери. У цьому випадку, коли головний контролер виходить з ладу, потокові записи обробляються іншим резервним контролером (RM), доки головний контролер не відновиться. У разі несправності комутатора, встановлюється нові відображення (нові записи резервного потоку) у комутаторах ToR для кожного активного запису. Пакети в ToR будуть направлені до місць призначення альтернативними шляхами, зазначеними резервними потоковими записами.

Інше рішення, яке може протидіяти обмеженням надійності у централізованій архітектурі було запропоновано у [3]. Інтеграція між вільним багатошляховим доступом і перевантаженням керування базується на багатошляховому підході динамічного балансування навантаження, який запускає розподілений алгоритм у разі відмови контролера. Алгоритм оновлює комутатори з будь-якими змінами в «навантаженні шляху» на пов'язаних маршрутах у випадках заторів і дисбалансу навантаження.

Масштабованість. Відокремлення між площиною даних і керування відрізняє SDN від традиційної мережі. У SDN обидві площини можуть «розвиватися незалежно», поки їх з'єднують API, і це централізоване уявлення про мережу прискорює зміни в площині керування. Однак роз'єднання має свої недоліки. Окрім складності визначення стандартних API між обома площинами, можуть виникнути обмеження масштабованості. Деякі дослідники дійшли висновку, що коли мережа збільшує кількість комутаторів і кількість кінцевих хостів, контролер SDN може стати ключовим вузьким місцем.

У міру того, як пропускна здатність і кількість комутаторів і потоків збільшуються, більше запитів потраплятиме в чергу до контролера, який може не впоратися з ними всіма. Дослідження контролера SDN (NOX) показали, що він може обробляти до 30 тисяч запитів/с. Цього може бути достатньо для корпоративних і кампусних мереж, але це вузьке місце для мереж центрів обробки даних із високою швидкістю потоку. Крім того, великий центр обробки даних, що складається з 2 мільйонів віртуальних машин, може генерувати 20 мільйонів потоків на секунду. Однак поточні контролери можуть підтримувати приблизно 10^5 потоків за секунду в оптимальному випадку. Окрім перевантаження контролера, процес налаштування потоку може накласти обмеження на масштабованість мережі.

Налаштування потоку складається з чотирьох кроків:

- пакет надходить на комутатор і не відповідає жодному запису потоку;
- комутатор надсилає запит до контролера, щоб отримати інструкції щодо пересилання пакета;
- контролер надсилає новий запис потоку з новими правилами пересилання назад до комутатора;
- комутатор оновлює свої записи в таблиці потоків.

Продуктивність процесу налаштування залежить від ресурсів комутатора (ЦП, пам'ять тощо) і продуктивності контролера (програмного забезпечення). Час оновлення інформаційної бази пересилання комутатора (FIB - Forwarding Information Base) створює затримку в налаштуванні будь-якого нового потоку. Ранні тести контролерів і комутаторів SDN показали, що контролер міг відповісти на запит налаштування потоку протягом однієї мілісекунди, тоді як апаратні комутатори могли підтримувати кілька тисяч установок на секунду з затримкою менше 10 мс в найкращому випадку.

Затримки налаштування потоку можуть стати проблемою для масштабованості мережі. Крім того, накладні витрати на широкомовний мережевий трафік та швидке збільшення записів у таблиці потоків обмежують масштабованість SDN. Платформа SDN може призвести до обмеження видимості мережевого трафіку, що робить усунення несправностей майже неможливим. До SDN, мережева команда могла швидко виявити, наприклад, що резервне копію-

вання сповільнює роботу мережі. Тоді рішенням буде перенести резервне копіювання на менш завантажений час.

На жаль, за допомогою SDN видно лише джерело тунелю та кінцеву точку тунелю з трафіком User Datagram Protocol (UDP), але, що важливо, неможливо побачити, хто використовує тунель. Немає способу визначити, чи є проблема процесом реплікації, системою електронної пошти чи чимось іншим. Справжній найкращий кореспондент захищений від зору тунелями UDP, а це означає, що коли трафік сповільнюється та користувачі скаржаться, точно визначити проблемну зону в мережі є певним викликом. З цією втратою видимості, усунення несправностей ускладнюється, виникають обмеження масштабованості, а затримки у вирішенні можуть стати згубними для бізнесу. Щоб звести до мінімуму швидке збільшення записів потоку, контролер повинен використовувати перезапис заголовка в ядрі мережі. Потокові записи будуть на вхідному та вихідному комутаторах.

Покращена масштабованість мережі також може бути забезпечена шляхом увімкнення міграції віртуальних машин і віртуальних сховищ між сайтами, як у програмному проміжному програмному забезпеченні IaaS на основі OpenFlow і «CrossRoads», мережевій структурі на основі OpenFlow. Можливо також інше рішення проблеми масштабованості. Це розподілена архітектура керування потоками, яка може масштабуватися відповідно до вимог (велика кількість хостів, потоків і правил) великих мереж.

Життєздатне рішення проблем масштабованості пропонується у відмовостійкій архітектурі SDN «CORONET», яка масштабується до великих мереж завдяки механізму VLAN, встановленому в локальних комутаторах [4]. CORONET забезпечує швидке відновлення після збоїв комутатора або зв'язку, підтримує масштабовані мережі, використовує альтернативні методи багатошляхової маршрутизації, працює з будь-якою топологією мережі та використовує централізований контролер для пересилання пакетів. Він складається з модулів, відповідальних за виявлення топології, планування маршруту, призначення трафіку та розрахунок найкоротшого маршруту (алгоритм Дейкстри). Основною особливістю CORONET є використання мереж VLAN, які можуть спростити пересилання пакетів, мінімізувати кількість правил потоку та підтримувати властивості масштабованості.

В іншому рішенні, мікропотоками керують у площині даних, а більшими потоками — у контролері, що означає, що навантаження на контролер зменшиться, а масштабованість мережі буде максимально збільшена. Цей підхід мінімізує витрати на видимість контролера, пов'язану з кожним налаштуванням потоку, і зменшує вплив накладних витрат на планування потоку, таким чином підвищуючи продуктивність і масштабованість мережі.

Нарешті, [5] описує масштабовану структуру керування SDN, McNettle, яка виконується на багатоядерних серверах із спільною пам'яттю та базується на Nettle. Експерименти показали, що McNettle може обслуговувати 5000 комутаторів за допомогою одного контролера з 46 ядрами та може обробляти 14 мільйонів потоків за секунду із затримкою нижче 200 мкс для легких навантажень і 10 мс для навантажень, що складаються з до 5000 комутаторів.

Продуктивність. SDN — це метод, заснований на потоках і тому його продуктивність вимірюється на основі двох показників: часу налаштування потоку та кількості потоків за секунду, які контролер може обробляти. Існує два способи налаштування потоку: проактивний і реактивний. У проактивному режимі налаштування потоку відбувається до надходження пакета на комутатор і, отже, коли пакет надходить, комутатор уже знає, як з ним впоратися. Цей режим має незначну затримку та знімає обмеження на кількість потоків за секунду, які може обробляти контролер.

Загалом контролер SDN заповнює таблицю потокових записів максимальною кількістю можливих потоків. У реактивному режимі налаштування потоку виконується, коли пакет, що надходить на комутатор, не відповідає жодному із записів комутатора. Потім контролер вирішить, як обробити/обробити цей пакет і інструкції будуть кешовані на комутаторі. Як наслідок, реактивний час налаштування потоку є сумою часу обробки в контролері та часу для оновлення перемикача при зміні потоку. Таким чином, ініціювання потоку додає накладні витрати, які обмежують масштабованість мережі та вводять реактивну затримку налаштування потоку.

Іншими словами, нова настройка потоку вимагає від контролера погодити потік трафіку, що означає, що тепер кожен потік повинен проходити через контролер, який, у свою чергу, створює потік на комутаторі. Однак контролер — це програма, яка працює на серверній ОС через канал зі швидкістю 10 ГБ/с (із затримкою в десятки мілісекунд). Він відповідає за керування комутатором, який може перемикати 1,2 ТБ/с трафіку із середньою затримкою 1 мкс. Крім того, комутатор може обробляти 100 тис. потоків, при цьому в середньому 30 тис. втрачаються. Таким чином, контролеру можуть знадобитися десятки мілісекунд для встановлення потоку, тоді як життя потоку, що передає 10 МБ даних (типова веб-сторінка), становить 10 мс.

Були проведені різні експерименти з налаштування, щоб перевірити пропускну здатність і затримку різних контролерів. Змінювалась кількість комутаторів, кількість потоків і навантаження на контролер. На основі цих експериментів і моделювання можна дійти висновку, що додавання більшої кількості потоків за межі кількості комутаторів не покращує затримку, а обслуговування кількості комутаторів, яка перевищує кількість доступних ЦП, збільшує час відгуку контролера. Експерименти також показали, що час відповіді контролера коливається від 4 до 30 мс для різної кількості комутаторів з 4 потоками та 2^{12} запитами.

Робота з потоками 100K вимагає, щоб ASIC комутатора мали такий тип потоку. Сучасні ASIC не мають такої можливості і тому таблицю потоку потрібно використовувати як кеш. Підсумовуючи, швидкість налаштування потоку в кращому випадку анемічна на існуючому обладнанні і тому можлива лише обмежена кількість потоків за секунду. Лінійний пошук O(n) у великій нотації O(n) для програмних таблиць не може наблизитися до пошуку O(1) TCAM з апаратним прискоренням у комутаторі, що спричиняє падіння швидкості пересилання пакетів для великих розмірів таблиць із символами підстановки. Щоб подолати обмеження продуктивності, слід враховувати ключові фактори, які впливають на час налаштування потоку. Цими ключовими факторами є обробка та продуктивність введення/виведення контролера. Ранні тести показали, що продуктивність контролера можна значно збільшити за допомогою добре відомих методів оптимізації, таких як пакетне введення/виведення. Інше життєздатне рішення для полегшення проблеми з продуктивністю було запропоновано під назвою Maestro. Маестро використовував два основні параметри; «поріг пакетування вхідних даних» (IBT), настроюване порогове значення, яке визначає стадію для створення процесу потокового завдання для обробки запиту потоку і «порогове значення необроблених пакетів, що очікують» (PRT), яке визначає допустиму кількість незавершених Калібрування цих параметрів дозволить зменшити затримку та максимізувати пропускну здатність мережі відповідно до збільшення пропускної здатності PRT і IBT. Використовувати для пошуку оптимального діапазону значень PRT та IBT.

На основі результатів досліджень [6] і тестування продуктивності мережі SDN і звичайних мережевих архітектур, які були проведені, можна зробити висновок, що кращою топологією мережі на основі параметрів затримки є звичайна топологія мережі із середнім значенням затримки 0,09588 мс. Деякі пропозиції, які можна розглянути для майбутніх досліджень, полягають у тому, що тестування може включати більше однієї мережевої підмережі, використовуючи більше двох контролерів із більш складною топологією мережі.

Розміщення контролера. Проблема розміщення контролера впливає на кожен аспект відокремленої площини керування, від затримок налаштування потоку до надійності мережі, відмовостійкості та, нарешті, до показників продуктивності. Наприклад, глобальні мережі з великою затримкою поширення (WAN) обмежують доступність і час конвергенції. Це має практичні наслідки для проектування програмного забезпечення, впливаючи на те, чи можуть контролери реагувати на події в реальному часі, чи вони повинні заздалегідь надсилати дії пересилання до елементів пересилання. Ця проблема включає розміщення контролерів з урахуванням доступної топології мережі та кількості необхідних контролерів. Користувач визначає різні метрики, які контролюють розміщення контролера в мережі.

Випадкове розміщення для невеликого значення k у проблемі k-медіани, алгоритм аналізу кластеризації, призведе до середньої затримки між 1,4x і 1,7x більшою, ніж оптимальне розміщення. Пошук оптимального розміщення контролерів є гарячою темою для дослідження SDN,

особливо для глобальних розгортань SDN, оскільки вони вимагають кількох контролерів, а їх розміщення впливає на всі показники в мережі. Підвищення надійності є важливим, оскільки збої в мережі спричиняють розрив зв'язку між площиною керування та пересилання та можуть вивести з ладу деякі комутатори.

Проблема розміщення контролера з урахуванням надійності дуже важлива. Основну мету проблеми можна зрозуміти, використовуючи наступне питання: як розмістити задану кількість контролерів у певній фізичній мережі так, щоб попередньо визначена цільова функція була оптимізована. Можна розглядати проблему надійності як метрику розміщення, яка відображається відсотком дійсних шляхів керування. Можна розробити модель оптимізації, яка би максимізувала очікуваний відсоток дійсних шляхів керування. На цей відсоток впливає розташування контролера на одному з вузлів-кандидатів, кількість суміжностей між контролерами, доступна кількість контролерів і резервування комутаторів на контролері.

Будь-який збій, який відключає площину пересилання від контролера, може призвести до серйозного погіршення продуктивності. Грунтуючись на цьому спостереженні, можна описати проблему розміщення контролера з урахуванням стійкості (шляху) (захисту шляху). Стійкість з'єднання між контролером і комутатором розглядається як метрика розміщення, яка відображалася в здатності комутаторів захищати свої шляхи до контролера. Запропонована евристика спрямована на максимізацію можливості швидкого відновлення після відмови на основі розміщення контролера з урахуванням стійкості та маршрутизації трафіку керування в мережі. Ця евристика складається з двох алгоритмів для вибору найкращого розташування контролера та максимізації показника стійкості з'єднання.

Була розроблена проблему розміщення контролера з урахуванням затримки. Метою було не знайти оптимальне рішення для проблеми розміщення контролера з урахуванням затримки, а забезпечити початковий аналіз для подальшого вивчення формулювання фундаментальних проблем проектування. Таким чином, проблема була спрямована на мінімізацію середньої затримки розповсюдження на основі відповідного розміщення контролера. Мінімізація базувалася на оптимізаційній моделі, сформованій на основі задачі мінімальної k-медіани.

Відповідь на те, де та скільки контролерів розгортати, залежить від бажаних меж реакції, вибору(-ів) метрики та самої топології мережі [7]. Більшість мереж демонструють зменшення віддачі від кожного доданого контролера разом із компромісами між метриками. Дивно, але в багатьох мережах середнього розміру затримка від кожного вузла до окремого контролера може відповідати цілям щодо часу відповіді існуючих технологій.

Використання низькорівневих інтерфейсів між контролером і мережевим пристроєм. Незважаючи на те, що SDN спрощує керування мережею, розробляючи контрольні додатки з простими інтерфейсами для визначення мережевих політик високого рівня, базова структура SDN повинна перевести ці політики в конфігурації комутаторів низького рівня. Доступні сьогодні контролери забезпечують інтерфейс програмування, який підтримує низькорівневу, імперативну та керовану подіями модель. Інтерфейс реагує на мережеві події, такі як надходження пакетів і оновлення статусу з'єднання, встановлюючи та видаляючи окремі правила низькорівневої обробки пакетів, правила за правилом і комутатор за комутатором. У такій ситуації програмісти повинні постійно розглядати, чи вплине деінсталяція політик комутатора на інші майбутні події, які відстежує контролер. Крім того, вони повинні координувати кілька асинхронних подій на комутаторах для виконання навіть простих завдань.

Крім того, цей інтерфейс створює проблему поглинання часу та вимагає детального знання програмного модуля або апаратного пристрою, який виконує необхідні послуги. Багато дослідників розробляють різні мови програмування, які дозволяють програмісту описувати поведінку мережі за допомогою високорівневих абстракцій, залишаючи системі виконання та компілятору піклуватися про деталі реалізації. Застосовується FML, мову програмування високого рівня, що складається з операторів, які дозволяють або забороняють потоки, одночасно координуючи потоки через брандмауери та підтримуючи QoS. Однак це негнучка мова, оскільки вона не може перенаправляти або переміщувати потоки під час їх обробки.

Нарешті, існує Flog, мова керованого подіями логічного програмування. Введення логічного програмування в SDN корисно для обробки статистики мережі та поступових оновлень стану контролера. Головною особливістю, яка відрізняє Flog від інших мов, є комутатор навчання Ethernet. Процес навчання складається з моніторингу, групування та зберігання пакетів, які надходять на комутатор, а потім передачі цієї інформації до навчальної бази даних. Після цього генератор політики створює правила низького рівня, які переповнюють усі пакети, що надходять, а потім, на основі отриманої інформації, політика створює точне правило пересилання високого рівня.

Обмеження ASIC і ЦП. Хоча канал передачі даними керування в лінійній карті ASIC швидкий, канал передачі даних між ASIC і ЦП не використовується в частих операціях традиційного комутатора і тому він вважається повільним шляхом. Ethernet комутатор ProCurve 5406lz має пропускну здатність 300 ГБ/с, але виміряна пропускна здатність петлі між ASIC і ЦП становить 35 МБ/с. Зауважимо також, що ЦП з повільним CPU обмежує пропускну здатність між комутатором і контролером. Наприклад, пропускна здатність корисного навантаження налаштування потоку була виміряна між комутатором 5406lz і контролером OpenFlow і становить 10 МБ/с. Однак архітектура DIFANE використовує ці обмеження, розподіляючи правила узагальнення OpenFlow між різними комутаторами, щоб гарантувати, що рішення про пересилання відбуваються в площині даних.

Контроль каналу передачі даних між ASIC і ЦП не є традиційною операцією. OpenFlow визначає три лічильники для кожного запису таблиці потоків: кількість співпадань, кількість байтів пакетів у цих співпаданнях і тривалість потоку. Кожен лічильник визначено як 64 біти, і тому це додає 192 біти (24 байти) додаткової пам'яті на запис таблиці. Лічильники OpenFlow і логіка для їх підтримки додають значної складності та площі ASIC і створюють більше навантаження на ЦП. Якщо лічильники реалізовані в апаратному забезпеченні ASIC, може бути дуже важко змінити їхню функцію в міру розвитку протоколу SDN, оскільки це вимагатиме перепроектування ASIC або розгортання нового апаратного забезпечення комутатора. Крім того, перенесення локального лічильника з ASIC на контролер може різко обмежити продуктивність SDN.

Крім того, додавання підтримки SDN для створення ASIC означає пошук місця для структур, які зазвичай не зустрічаються в ASIC; лічильники байтів на потік, які використовує OpenFlow, можуть бути найбільшими такими структурами. Іншими словами, лічильники займають місце в області ASIC, усвідомлюючи, що ця область вважається дорогоцінною, оскільки розробка ASIC коштує багато грошей і часу. Однак, оскільки вартість ASIC комутаторів залежить від їх площі, існує верхня межа площі економічно ефективної ASIC. Оскільки область ASIC є цінною, це накладає обмеження на розмір структур пам'яті на кристалі, таких як TCAM, для підтримки записів таблиці потоків і лічильників для кожного входу. Однак будь-яка ділянка кремнію, виділена лічильникам, не буде доступною для пошукових таблиць.

Як відомо, комутатори мають центральний процесор для керування ASIC, але пропускна здатність між ними обмежена. Таким чином, зберігання лічильників у процесорі та DRAM замість ASIC спростить шлях від лічильників до контролера та мінімізує накладні витрати на контролер для доступу до цих лічильників. Інше можливе рішення, яке могло б усунути обмеження, це програмно-визначені лічильники (SDC), оскільки реалізація лічильників у програмному забезпеченні не вимагає перепроектування ASIC і може підтримувати більше інновацій. У запропонованому SDC ASIC не містить лічильників, але він генерує записи подій, які будуть додані до буфера. Щоразу, коли буферний блок заповнюється, ASIC переміщує його до ЦП. Центральний процесор витягує записи та оновлює свої лічильники, які зберігаються на приєднаній DRAM. SDC пропонує дві конструкції системи:

- комутатор SDC, у якому лічильник виведений із ASIC і замінений буферними блоками;
- комутатор SDC, у якому ЦП встановлено на ASIC.

Хоча друга конструкція вимагає додаткового простору ASIC, вона мінімізує пропускну здатність між площиною даних і ЦП, тому може розглядатися як ефективне рішення для контролю каналу передачі даних між ASIC і ЦП.

Безпека. На основі статистичних досліджень, проведених IT-організаціями, 12% респондентів у IT-бізнес-технологіях заявили, що SDN має проблеми з безпекою, а 31% респондентів у IT-бізнес-технологіях не визначилися, чи є SDN менш безпечною чи більш безпечною мережею. парадигми, ніж інші. Очевидно, IT-організації вважають, що SDN може створити певні проблеми з безпекою. Ризики безпеки SDN виникають через відсутність інтеграції з існуючими технологіями безпеки та неможливість перевірити кожен пакет. Крім того, вдосконалення інтелектуального програмного забезпечення контролера може збільшити вразливість контролера до хакерів і поверхонь атак. Якщо хакери отримають доступ до контролера, вони пошкодять кожен аспект мережі.

Підвищення безпеки SDN вимагає від контролера здатності підтримувати класи автентифікації та авторизації адміністраторів мережі [8]. Крім того, підвищення ефективності безпеки вимагає від адміністраторів можливості використовувати однакові політики для керування трафіком, щоб запобігти доступу до контрольного трафіку SDN. Додатковими рішеннями з урахуванням безпеки є впровадження інтелектуального списку контролю доступу (ACL) у контролері для фільтрації пакетів і повної ізоляції між орендарями, які спільно використовують інфраструктуру. Нарешті, контролер повинен мати можливість сповістити адміністраторів у разі будь-якої раптової атаки та обмежити зв'язок керування під час атаки.

SDN є багатообіцяючою технологією для комп'ютерних мереж і мереж центрів обробки даних, але їй все ще бракує політики стандартизації. Поточна архітектура SDN не містить стандартів для розуміння топології, затримки або втрат. Інші функції, які недоступні, включають виявлення циклу та можливість виправлення помилок у стані. SDN не підтримує горизонтальний зв'язок між мережевими вузлами для забезпечення співпраці між пристроями.

Висновки

Забезпечення функціональної стійкості програмно-визначених мереж вимагає вирішення цілої низки проблем, виникнення яких пов'язано як з особливостями архітектури, так і зі зростанням навантаження у сучасних умовах, такого як відеотрафік, великі дані, хмара та мобільні пристрої. У проведеному дослідженні проведений перелік проблем (надійність, масштабованість, продуктивність, розміщення контролера, інтерфейси низького рівня, обмеження ASIC і ЦП, безпека, енергоефективність), вирішення яких впливають на функціональну стійкість програмно-визначених мереж та запропоновані можливі шляхи їх вирішення. В той же час, запропоновані шляхи є не остаточним вирішенням проблем, а лише відкриттям поля для нових глибоких досліджень.

Список літератури

1. SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies, Thomas D. Nadeau, Ken Gray. Copyright © 2013. All rights reserved. Printed in the United States of America. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. – p. 382.

2. Гніденко М.П., Вишнівський В.В., Ільїн О.О. Побудова SDN мереж. – Навчальний посібник. – Київ: ДУТ, 2019. – 190 с.

3. Fang, S.,Yu, Y., Foh, C.H., Aung, K.M.M., "A Loss-Free Multipathing Solution for Data Center Network Using Software-Defined Networking Approach," IEEE Transactions on Magnetics, vol.49, no.6, pp.2723– 2730, June 2018.

4. Kim, H.J., Santos, J.R., Turner, Y., Schlansker, M., Tourrilhes, J., Feamster, N., "CORONET: Fault Tolerance for Software-Defined Networks,"Proceedings,2012 20th IEEE International Conference on Network Protocols (ICNP), pp.1–2, October 30–November 2, 2018.

5. Voellmy, A.,Wang, J.C., "Scalable Software-Defined Network Controllers," Proceedings, ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 289–290, 2019.

6. Raja Pahlevi Harahap, Deci Irmayani, Rahma Muti. Comparative analysis of software defined network performance and conventional based on latency parameters. Sinkron : Jurnal dan Penelitian Teknik Informatika Volume 7, Number 2, April 2022.

7. Brandon Heller, Rob Sherwood, Nick McKeown. The Controller Placement Problem. ACM SIGCOMM Computer Communication Review 42(4), September 2012.

8. Гніденко М.П., Прокопов С.В., Гніденко М.М. Підвищення безпеки програмно-визначених мереж (SDNs). / Київ: ДУТ. Наукові записки ДУТ – 2023. – №2(4). – с. 54-65.

S. Prokopov, S. Sierykh, M. Hnidenko, O. Vyshnivskyi

PROBLEMS WHOSE SOLUTIONS AFFECT THE FUNCTIONAL STABILITY OF SOFTWARE-DEFINED NETWORKS (SDNs)

SDN is changing the future of networking and bringing new innovations. Demand for services and network usage is growing rapidly. Despite the growth drivers, software-defined networks face significant challenges in their development. Bearing in mind the growing number of problems, the relevance of research initiatives aimed at overcoming challenges, which are initiated by a number of problems of the functioning of software-defined networks, increases. Solving these problems can significantly affect the functional sustainability of SDN. Operators need an efficient, flexible, and scalable network in any operating conditions. Among the problems of software-defined networks, such as the need to ensure adequate reliability, scalability, performance, placement of controllers and security can be highlighted. In terms of reliability, the SDN controller must be able to support multipath solutions or fast traffic forwarding to active links. The separation between the data plane and the control plane, which distinguishes SDN from traditional networking, affects the scalability and performance of software-defined networking. The controller placement issue affects every aspect of the isolated control plane, from flow setup latencies to network reliability, fault tolerance, and finally performance metrics. Improving SDN security requires the controller to be able to support authentication and authorization classes for network administrators. In addition, improved security performance requires administrators to be able to use the same traffic management policies to prevent access to SDN control traffic. The study analyzed the causes of these problems and suggested possible ways to solve them, which will significantly affect the functional stability of software-defined networks. At the same time, the proposed ways are not the final solution to the problems, but only the opening of the field for new in-depth research.

Keywords: Software-defined networks (SDNs), SDN controller, reliability, scalability, performance, placement of controllers, security.