УДК 621.391
Г. О. ГРИНКЕВИЧ, канд. техн. наук, доцент;
К. О. ДОМРАЧЕВА, канд. техн. наук;
С. В. ШЕЛУДЬКО, студент;
Д. П. КОНОВАЛОВ, студент,
Державний університет телекомунікацій, Київ

ПІДХОДИ ДО ВИМІРЮВАННЯ ТА МОНІТОРИНГУ МЕРЕЖІ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПЕРЕВІРКИ КЕРУЮЧИХ ПОВІДОМЛЕНЬ OPENFLOW

У статті визначено методи моніторингу мережі, на основі яких запропоновано нові схеми виявлення несправностей для SDN і докладний розгляд головних принципів технології OpenFlow. Також представлено концепцію Software-Defined-Networking,

Ключові слова: інформаційні мережі; програмно-конфігуровані мережі; інформаційна безпека; SDN; Linux; Ethernet; OpenFlow.

Вступ

Успішне функціонування мережної інфраструктури потребує одночасного виконання безлічі різнорідних завдань: від маршрутизації та моніторингу передаваних даних до контролю доступу та розподілу навантаження між мережними елементами. Парадигма програмно-конфігурованих мереж **SDN** (*Software Defined Network*) покликана спростити процес передавання даних і управління мережною інфраструктурою на основі логічного поділу мережі з виокремленням ***рівня управління*** і ***рівня передавання даних***. Зрештою згідно з концепцією SDN маємо мережу, що являє собою розподілену систему, де один або кілька контролерів керують безліччю комутаторів, котрі відповідають за передавання пакетів між мережним устаткуванням. Як і в традиційних мережах передавання даних, регулювання обміну повідомленнями та узгоджена взаємодія між елементами мережі забезпечується мережними протоколами. Усі функціональні можливості, що гарантують виконання вимог, а також порядок взаємодії контролерів і комутаторів SDN, визначаються протоколом OpenFlow. Техніка захисту здатна відновити OpenFlow мережі протягом 50 мс. При цьому жодних дій від мережного контролера не знадобиться, оскільки перемикач може безпосередньо реагувати на небезпечну ситуацію. Техніка реставрації передбачає, що контролер має взаємодіяти з мережними пристроями. Це вимагає додаткового часу, а тому метод недостатньо зручний для великомасштабних мереж із багатьма підвузлами. Сучасний стан формування методів аналізу і синтезу програмно-конфігурованих мереж знаходить відображення у працях таких учених, як В. Б. Толубко, Л. Н. Беркман, С. В. Козелков, Д. В. Агеєв, О. І. Лисенко, В. І. Новіков, М. М. Климаш, А. П. Бондарчук, O. Sheyner, P. Ammann, X. Ou, L. Wang, A. Рой, H. Poolsappasit.

Основна частина

Застосування формальних методів у рамках завдання аналізу і верифікації протоколу OpenFlow дозволяє не лише чітко описати набір властивостей, якими має володіти протокол, а й перевірити їх здійсненність і відповідність вимогам. Перевагою формальних методів є те, що при узгодженні та взаємодії процесів, які відповідають коректній роботі протоколу OpenFlow, формалізми можуть знадобитися в подальших розробках як прототипи, а це дозволить істотно скоротити часові й матеріальні витрати на виконання таких розробок. Щоб виявити, скажімо, неробоче посилання своєчасно, протоколи управління можуть слугувати для моніторингу підімкнення. Наприклад, неактивний порт комутатора можна виявити через втрату сигналу відмови. Виявлення шляху між двома вузлами у формі ламаної можливе за допомогою двонапрямленого Forwarding Detection (BFD) на базі протоколу Hello, визначеного в RFC 5880. Як альтернативу можна використовувати Link Layer Discovery Protocol (LLDP), але це викликає високе навантаження на мережному контролері і обмежує масштабованість, оскільки такі повідомлення моніторингу повинні бути оброблені на високій частоті. У [1] пропонується розширення OpenFlow специфікації для розгортання децентралізованого моніторингу мережі, у тому числі для генерування пакетів і обробки на мережних пристроях. Ідеться про підхід до несправності на основі технології MPLS, коли увага концентрується на відновленні при збоях у межах 50 мс.

Для того щоб мінімізувати кількість таких повідомлень, автори [1] пропонують тільки інформування вимикачів про збої зв'язку. Необхідний алгоритм діє на мережних пристроях, що дозволяє забезпечити менший час відновлення порівняно з випадком, коли контролер діє за схемою повідомлених катіонів. Як і в попередній концепції,

вимагається наявність додаткових функцій, якими буде поповнено комутатор OpenFlow.

Вимірювання і моніторинг. Моніторинг мережі інформує оператора про поточний стан мережі, а також є основою для алгоритмів виявлення несправностей. Це вимагає наявності відповідних датчиків у мережі, наприклад комутаторів, які забезпечують статистичні дані про потік записів.

Інструментарій NetFuse призначено для виявлення і пом'якшення перевантаження, яке може виникнути з різних причин, наприклад через розподілену відмову в обслуговуванні (DDoS), через атаки або в результаті запланованого резервного копіювання. Аби виявити таку раптову несправність, NetFuse установлюють як додатковий шар між мережними пристроями та мережним контролером для обробки OpenFlow керуючих повідомлень (скажімо, PacketIn, FlowMod, FlowRemoved). Для того щоб констатувати перевантаження, багатовимірний алгоритм агрегації використовується з метою виявлення підозрілих потоків. Надалі вони керуються адаптивним механізмом управління, який модифікує правила контролю на комутаторах, щоб краще справлятися з мережним перевантаженням.

На основі OpenFlow керуючих повідомлень Packet-in та FlowRemoved, FlowSense для обчислення лінії зв'язку використовують програми, виконувані на верхній частині мережі контролера. Спрощується розгортання порівняно з підходами, які вимагають додаткового шару (наприклад, NetFuse). Оскільки вимірювання оновлюється тільки після отримання повідомлення FlowRemoved, то потік, який містить правило байдужих полів у поєднанні з тривалим терміном дії, призводить до сповільненого досягнення результату. Дія при реалізації активної політики установлення потоку повідомлення Packet може відбутися будь-коли, а це означає, що активне опитування знадобиться для виявлення набутого стану.

Робота OpenSketch розглядається програмно. При цьому маємо просту конструкцію для площини даних, які можуть бути реалізовані на апаратному забезпеченні, тоді як функції аналізу даних розташовано на площині управління. Згідно з ідеєю ескізів, які є компактними структурами даних, особливо гнучкі зіштовхування потоків виступають як OpenFlow специфікація. У поєднанні з контролером OpenSketch для ескізу бібліотеки можуть бути розроблені нові алгоритми вимірювання, що дозволяє реалізувати автоматизовану конфігурацію комутаторів відповідно до вимог додатка вимірювань.

Установлення додаткових потоків записів для цілей моніторингу розглядається у спеціальній схемі. Ця схема прийнятна, оскільки OpenFlow специфікації підтримують кілька етапів потоку записів. Мережний контролер, зчитуючи відповідні лічильники періодично, може виявити важливий Hitters через ієрархічний алгоритм. Перевага цього підходу полягає у використанні апаратного забезпечення і знижених витратах на комутатор.

Виявлення атак DDoS спирається на аналіз потоку статистики за допомогою нейронних мереж, із використанням підходів, реалізованих на платформі контролера NOx і таких, що включають у себе розглянуті далі кроки.

-> [Flow Collector] -> [Feature Extractor] -> [класифікатор] -> Alarm
|---------- <----------|
LoopDetection

Насамперед потік статистики з одного чи кількох комутаторів витягується через певні проміжки часу за допомогою потоку колектора. Далі функція ідентифікує відповідні функції C, які вказують на атаку DDoS. Наприклад, темпи зростання окремих притоків в одному напрямі є індикатором початку такої атаки. Кількість перемикачів, що беруть участь у моніторингу, може бути адаптована до нових топологій, але збільшення кількості комутаторів створює додаткові накладні витрати через зростання кількості керуючих повідомлень.

Пряме вимірювання лічильників, пов'язаних зі входом, необхідне для того, щоб побудувати матриці, які містять об'єм між пунктами відправлення та пунктами призначення пар у мережі. Проте запити збільшують навантаження на комутатор, а це небажано в мережі, що складається з кількох перемикачів і великої кількості притоків. OpenTM описує стратегії визначення того, якому комутатору для запиту уздовж потоку слід зменшити накладні витрати. Автори пропонують рівномірний розподіл для запитів потоку лічильників серед усіх комутаторів у мережі, включаючи інформацію про маршрутизацію від мережного контролера з цією метою.

Доцільно описати запити як P2P трафік, що може класифікуватися на основі особливостей мережного рівня. Це означає, що питання конфіденційності пов'язано з глибокою інспекцією невідомих пакетів (DPI) і немає потреби у спеціалізованому апаратному забезпеченні, бо аналіз можна здійснити на мережному контролері. Після визначення відповідних функцій мережного рівня, здатних відрізнити P2P від не-P2P трафіку, вони можуть бути реалізовані як додаток для контролера NOX і оцінені для загальнодоступного набору даних.

Налаштування, перевірка і тестування. Виявлення мережних несправностей є серйозною проблемою для кожного оператора, оскільки йдеться про людський фактор. Для того щоб прискорити виявлення несправності, необхідні нові методи

для SDN архітектур. Маючи на меті налагодження мереж, застосовують підхід, що включає в себе як мережний контролер (програмне забезпечення), так і мережні пристрої (апаратне забезпечення), котрі потребують відшукання основної причини помилки. Формальна перевірка програмного забезпечення контролера може визначити, чи гарантуються певні властивості коректності (наприклад, припинення циклів) і чи забезпечується SOFT [2].

Зауважимо, що тестування комп'ютерних мереж потрібне заради того, аби гарантувати, що всі мережні компоненти працюють, як і очікувалося. При цьому тестовий сценарій має бути якомога ближчий до операціональних ситуацій. Ця проблема вирішується за допомогою ATPG [3], що генерує тестові пакети, тоді як OFRewind [4] забезпечує можливість для повторного відтворення захоплених сценаріїв.

OpenFlow дозволяє використовувати стандартне обладнання, вимагаючи при цьому правильного виконання агентів на всіх мережних пристроях. ONF має визначити випробування із серією тестів, що особливо корисно для виробників задля уникнення помилок у програмному забезпеченні на етапі розробки нової моделі комутатора.

Аналогічний підхід реалізується за допомогою OFTest [3], що є частиною проекту Project Floodlight і дозволяє розробляти платформи SPECI тестових сценаріїв.

SOFT [5] використовує тести на сумісність, щоб забезпечити правильну реалізацію на всіх комутаторах у мережі. З огляду на символічне виконання всі можливі шляхи в програмі (firmware) визначають поведінку кожного мережного пристрою. Згодом здійснюється перехресна перевірка між мережними пристроями з використанням головного пристрою, що виявляє невідповідності між вхідними наборами. Це дає змогу мережним операторам знаходити помилки реалізації перед розгортанням обладнання і гарантує безпомилкову роботу мережних пристроїв (рис. 1).

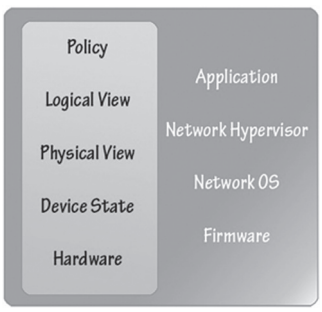


Рис. 1. Рівні стану та кодові рівні стека SDN у разі безпомилкової роботи в мережі

Зауважимо, що рівні стану можуть бути коректно відображені на будь-який інший рівень (еквівалентності). В іншому випадку помилка може бути локалізована між рівнями, які відрізняються один від одного і можуть бути виявлені в проміжному рівні коду.

Системний підхід до пошуку та усунення несправностей SDN висвітлено в [2], де пропонується, наприклад, потік, який відокремлює стек SDN на рівні стану і на кодових рівнях. Рівні стану являють собою мережу конфігурації, тоді як рівні стека описують відображення між двома рівнями стану. Методика дозволяє автоматичне визначення кодового рівня, де міститься несправність. Нормальне поводження мережі забезпечує еквівалентність між усіма рівнями стану таким чином, що кожний рівень може бути відображений на будь-який інший. У разі несправності мережі помилка може бути локалізована в код рівня, розташованого між рівнями стану, які не є еквівалентними. Після того як помилку було розміщено, її причину можна визначити за допомогою методів, про які вже йшлося.

Мережний налагоджувач ndb [5] допомагає операторові визначити причину збоїв програмного забезпечення (або помилки) у мережі. Кожний комутатор надсилає повідомлення, які спрацьовують, коли пакет з інформацію про збіг потоку вводу надходить на комутатор. Такі повідомлення збираються з усіх перемикачів у централізований колектор, який може бути використаний для визначення пакетів, що стосуються пакета точки зупинки. Наприклад, для того, щоб досліджувати помилку досяжності для пакета, надісланого від хоста *A* до хоста *B*, може бути використаний такий запит:

• OFRewind [3] — забезпечує запис і відтворення налагоджених засобів для SDN. Він вставляється як проксі-сервер між мережним контролером і мережними пристроями, дозволяючи перехоплення і видозміну повідомлень управління для запису або відтворення фабричного калібрування. Для того щоб зберігати трафік користувачів, сховища даних, керовані компонентом OFRewind, підмикаються до комутаторів. Різні режими, скажімо Replay, дозволяють різні тестові сценарії, аби отримати, наприклад, тільки керуючі повідомлення.

• VeriFlow [3] — забезпечує більш сфокусований вигляд на площині даних VERI. Це створює додатковий шар між мережним контролером і мережними пристроями, щоб мати змогу перевірити нові правила в режимі реального часу, перш ніж вони будуть розгорнуті в мережі. Правила мережі поділено на безліч класів еквівалентності (ЕКМ), де подібні дії переадресації розташовано в ЕКМ. Вони зберігаються в деревоподібних структурах даних, котрі описують шляхи пересилання пакетів у мережі. Нові правила аналізують VeriFlow шляхом виконання запитів на основі інваріантів

для того, аби виявити їх порушення. Такі інваріанти можуть бути встановлені через API та охоплювати широкий спектр умов, наприклад досяжності, петлі вільності й консистенції.

Anteater [6] також фокусується на площині даних аналізу. Вона являє собою зібрану топологію мережі та пересилання інформаційних баз (FIBs) із мережних пристроїв, що відіграють роль логічних функцій. Використовуючи цей метод, автори змогли виявити 23 помилки мережі. Перевага такого підходу полягає в тому, що реальне поводження системи можна проаналізувати за відсутності протоколів маршрутизації, завдяки чому процедура істотно спрощується. Окрім того, цей підхід дозволяє виявляти помилки програмного забезпечення маршрутизатора.

Компоненти системи OF Rewind наведено на рис. 2.

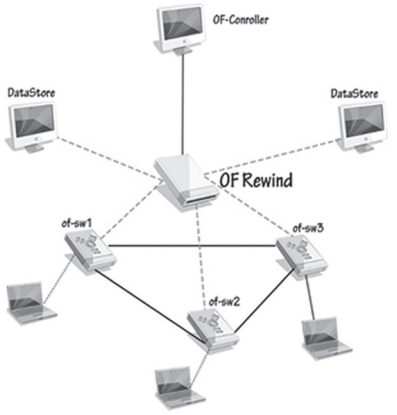
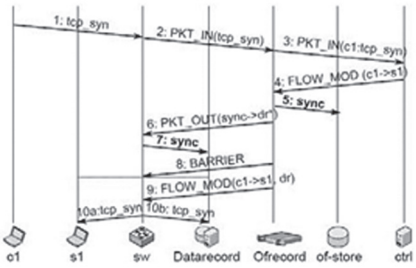
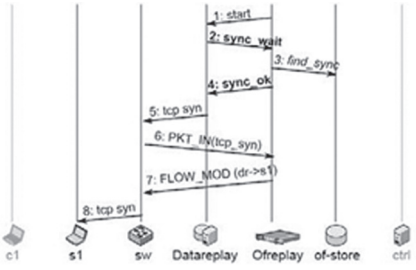


Рис. 2. Системні компоненти OFRewind

Запит TCP SYN, що відправляється від хоста c1 до S1 (рис. 3), може бути записаний за допомогою інструментарію Ofrecord (*а*) і знову прочитаний за допомогою інструментарію Ofreplay (*б*).



а



б

Рис. 3. Приклад використання Ofreplay, коли запит (5: TCP SYN) відправляється із системи зберігання Datareplay від c1

У разі просторового аналізу (HSA) [4] головна увага зосереджується на виявленні помилок, таких як збої досяжності, а також циклів пересилання, трафіку з ізоляцією, проблем витоку. Кожний заголовок пакета подається як точка у просторі, тоді як мережні пристрої та порти моделюються в ньому. Коли пакет перетинає мережу, вона трансформується з однієї ділянки до іншої. Поділ мережного трафіку здійснюється через стрибок, що подається як підмножина простору. Мережні пристрої, такі як комутатори й маршрутизатори, додатково можуть бути змодельовані за допомогою відповідних функцій передавання даних. У поєднанні із заголовком простору це дозволяє виявляти порушення у вигляді відповідних відмов. І хоча така структура здатна локалізувати джерело помилки (наприклад, непослідовні таблиці маршрутизації), вона не може визначити причину, з якої сталася помилка.

Випробування нових застосувань контролера потрібне OpenFlow для належної конфігурації контролера та перемикачів. Оскільки небагато осіб має доступ до таких ресурсів, то системи, до складу яких входять віртуальні машини (VM), є привабливою альтернативою. Через величезну витрату пам'яті, необхідної для створення великої мережі, масштабованість поширюється лише на кілька перемикачів. Більш легкий тип віртуалізації можливий з Mininet [5] мережним емулятором, який дозволяє здійснювати тестування на стандартному ноутбуці.

Для створення нової мережі наведена далі команда дає приклад тесту на віртуальній мережі з топологією дерева глибини 2, розгалуженням 8 і мережним контролером на основі NOX:

```
mn --switch ovsk --controller nox --topo tree,
depth=2,\\ fanout=8 --test pingAll
```

Усі номери, хости, комутатори й контролери емулюються, причому команди оболонки дозволяють використовувати основні засоби мережі (наприклад, пінг).

Mininet дає змогу розробникам виконувати й тестувати нові програми контролера в різних мережних топологіях. Обмеження цього підходу виникає через те, що продуктивність віртуалізації машини зазнає впливу високих навантажень. Дається взнаки швидкість передавання пакетів, а також складність O(n) порівняно з O(1) для таблиці пошуку в апаратному перемикачі на основі трійкової асоціативної пам'яті (TCAM). Завдяки своєму широкому використанню Mininet є найбільш поширеним емулятором для SDN та OpenFlow.

Мережа відлагодження необхідна й для того, щоб виявити помилки, коли збої програмного забезпечення не є очевидними. Метод [6] може бути використаний при тестуванні програм контро-

лера з метою виявлення порушень властивостей коректності, зумовлених помилками в програмному забезпеченні контролера. Перевірка моделі використовується для опису топології мережі та досліджує простір станів, який включає в себе контролер, перемикачі та користувачів. Для того щоб зменшити розмір вхідного простору, обробку подій на контролері виконують автоматично символічним двигуном, який генерує тестові входи і впускає пакети в мережу. Метод-прототип був оцінений на трьох реальних OpenFlow додатках, написаних у Python для контролера NOX. Типовим прикладом є помилка в застосуванні комутатора MAC-навчання, який має надсилати пакети конкретного пристрою навіть тоді, коли цей пристрій переміщується в нове місце в мережі. Якщо жорсткий тайм-аут нотифікації пропущено, то всі пакети раніше доставляються на колишнє місце і відповідні записи потоку не оновлюються з новими параметрами визначення місцезнаходженнянння. Цей інструментарій дозволяє виявляти такі конструкції помилок реалізації, що не є загальнодоступними.

Безпека. Архітектура SDN дозволяє реалізувати нові концепції безпеки, котрі неможливо було впровадити раніше. Наприклад, кожний мережний пристрій може бути налаштований так, щоб блокувати ті чи інші пакети, хоча для виявлення вторгнень традиційно потрібні дорогі апаратні вирішення. У [7] показано, як алгоритми виявлення аномалій можуть бути адаптовані до мереж OpenFlow на основі реалізації в мережному контролері. Водночас архітектура SDN відкриває новий простір для діяльності зловмисників і вимагає адекватних механізмів захисту. Наприклад, FortNOX [8] забезпечує механізм безпеки, який захищає потік установки від супротивників.

У [9] розглянуто виявлення аномалій OpenFlow і стверджується, що потік має бути переміщений від ядра до домашньої мережі (близько до користувача), щоб отримати кращі результати щодо виявлення. Автори адаптували чотири існуючі алгоритми для використання з OpenFlow, у тому числі виявлення скануючих інфекцій на хостах, що лімітує швидкість в умовах інфекції; виявлення аномалій із використанням максимальної ентропії та детектора аномального значення. Оскільки навантаження з обробки поширюється на домашніх користувачів, такий підхід знижує вимоги з обробки до постачальника послуг інтернету (ISP) і знижує витрати. Крім того, розгортання алгоритмів виявлення на мережному контролері не впливає на продуктивність пересилання пакетів на площині даних.

Зауважимо, що FRESCO [10] структура дозволяє розгортати служби безпеки для розімкненої Flow. Ця структура включає в себе мову сценаріїв, що сприяє розвитку служб безпеки на базі API та бібліотек, котрі включають у себе 16 багаторазових модулів. Саму структуру FRESCO реалізовано як додаток, побудований на контролері NOX.

Захист від атак IP-сканування описано в OpenFlow щодо випадкового вузла комутації [2]. Цей підхід випливає с того, що статичні IP-адреси вважаються легкою мішенню для зловмисників. Проте можна уникнути за допомогою проактивних методів, які змінюють IP-адреси господарів з плином часу (рухома мішень оборони (MTD)). У разі розімкненого потоку реальна IP-адреса, яку зберігає господар, замінюється віртуальною IP-адресою. Остання часто зазнавала перепризначення до мережних пристроїв за допомогою мережного контролера. Для цього потрібен механізм трансляції адрес, а також гарантії певних обмежень, таких як мутації непередбачуваності. Підхід оцінювали з використанням MiniNet проти зовнішнього мережного сканера (NMAP15) і атаки хробака.

FortNOX [4] як нове виконання ядра політики безпеки підвищує захист потоку під час процедури налаштування в OpenFlow. Це може бути використано суперником для того, аби взяти під своє керування мережу. Натомість FortNOX вимагає цифрового підпису як засіб авторизації. Окрім того, FortNOX виявляє, коли потік здатний обманути політику безпеки. Це можливо не тільки в разі перекриття діапазонів IP, а й тоді, коли правило встановлює новий заголовок пакета і пакет може досягти пункту призначення. В іншому випадку відбувається блокування.

Висновки

♦ Управління несправностями концентрується переважно на виявленні та усуненні переривань зв'язку, які можуть не тільки завдати шкоди даним, а й порушати зв'язок із контролером.

♦ Дослідження щодо області OpenFlow на основі SDN вже допомогли вирішити деякі питання до стадії оперативного розгортання самого проекту.

♦ Маючи на меті запропонувати нові схеми виявлення несправностей для SDN, слід передусім оволодіти методами моніторингу для цієї архітектури.

♦ Існує кілька підходів до вимірювання та моніторингу мережі, котрі ґрунтуються на вставлянні додаткових шарів між пристроями та контролером мережі. Це дозволяє забезпечити перевірку керуючих повідомлень OpenFlow, щоб ідентифікувати, наприклад, трафік перевантаження. Додатковий шар має, вочевидь, архітектуру, відмінну від оригінальної архітектури OpenFlow і вимагає додаткових апаратних засобів, а також аналізу потоку записів на мережі контролера. Усе це можна досягнути розгортанням алгоритмів моніторингу, що працюють на основі мережного контролера.

♦ Результати аналізу потоку можуть бути використані для виявлення характеристик трафіку та побудови повної картини мережі шляхом включення вхідних сигналів від усіх мережних пристроїв.

♦ Для створення програмного забезпечення необхідно налагодити контроль за пакетами в мережі, хоча формально перевірка може бути використана для визначення правильності потоку. Це необхідно для того, щоб генерувати докладні мережні знімки, а також інспектувати параметри функції за певних умов у мережі. Із погляду безпеки потенціал механізмів OpenFlow, реалізованих на нових схемах безпеки, не був добре вивчений.

♦ Аналізуючи пакети в повідомленні Packet на контролері мережі, доходимо висновку, що вони можуть бути використані для ідентифікації та визначення проникнення зв'язку, перш ніж потік налаштується. І хоча для безпеки мережі було досліджено нові вектори атак, централізований контролер мережі та канал управління залишаються частково незахищеними.

Список використаної літератури

1. ***Metaswitch*** *Networks, «PCE — an evolutionary approach to SDN». [Електронний ресурс] // Режим доступу:*
http://www.metaswitch.com/sites/default/files/metaswitch-white-paper-pce-an-evolutionary-approach-to-sdn.pdf (2012.)

2. ***Смелянский Р. Л.*** *Программно-конфигурируемые сети. Открытые системы. СУБД 9. 2012. С. 23–26.*

3. ***Захаров А. А., Попов Е. Ф., Фучко М. М.*** *Аспекты информационной безопасности архитектуры SDN [Електронний ресурс] // Режим доступу:*
http://vestnik.sibsutis.ru/uploads/1459328716_7845.pdf

4. ***Климаш. М. М.*** *Розробка методу балансування навантаження в SDN мережах на основі модифікованого протоколу STP [Електронний ресурс] / М. М. Климаш., М. І. Бешлей., Ю. Л. Дещинський., О. М. Панченко // Комп'ютерні технології друкарства. 2015. С. 146–155. Режим доступу:*
http://ctp.uad.lviv.ua/images//ktd/34_klymach.pdf

5. ***Шалимов А. В.*** *Технологии SDN/OpenFlow [Електронний ресурс] // Режим доступу:*
http://lvk.cs.msu.su/~sveta/SDN_OpenFlow_basics_lecture1.pdf

6. ***Casado M.*** *Fabric: a retrospective on evolving SDN / M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian // in Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN'12). New York, NY, USA: ACM. 2012. P. 85–90.*

7. ***Hoelzle U.*** *OpenFlow @ Google. [Електронний ресурс] // Режим доступу:*
http://opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf (2012.)

8. ***HP*** *Networking, «Software defined networks (SDN)». [Електронний ресурс] // Режим доступу:*
http://h17007.www1.hp.com/us/en/mobile/solutions/tech/sdn.html

9. ***Juniper*** *Networks, «OpenFlow Switch Application (OF-APP) for Juniper MXSeries Routers». [Електронний ресурс] // Режим доступу:*
https://developer.juniper.net/shared/jdn/docs/Programmable-Networks/OpenFLow APP JDN Overview.pdf

10. ***Gude N.*** *NOX: towards an operating system for networks / N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker //*

Рецензент: доктор техн. наук, **В. Ф. Заїка,** Державний університет телекомунікацій, Київ.

А. А. Гринкевич, К. А. Домрачева, С. В. Шелудько, Д. П. Коновалов

ПОДХОДЫ К ИЗМЕРЕНИЮ И МОНИТОРИНГУ СЕТИ ДЛЯ ОБЕСПЕЧЕНИЯ ПРОВЕРКИ УПРАВЛЯЮЩИХ СООБЩЕНИЙ OPENFLOW

В статье определены соответствующие методы мониторинга сети, на основе которых предложены новые схемы обнаружения неисправностей для SDN и подробно рассмотрены главные принципы технологии OpenFlow. Представлена также концепция Software-Defined-Networking.

Ключевые слова: информационные сети; программно-конфигурируемые сети; информационная безопасность; SDN; Linux; Ethernet; OpenFlow.

G. A. Grynkevych, K. O. Domracheva, S. V. Sheludko, D. P. Konovalov

APPROACHES TO MEASURING AND MONITORING THE NETWORK TO PROVIDE OPENFLOW CONTROLLED CHECKING

The article identifies the appropriate network monitoring methods to be explored to propose new SDN fault schemes and explains in detail the basic principles of OpenFlow. Detection of network failures is a serious problem for each operator and is based on the human factor. In order to reduce the time of failure detection, new methods for SDN architectures are needed. There are several approaches to network measurement and monitoring, which are based on the insertion of additional layers between devices and the network controller. This allows for the control of OpenFlow control messages, for example, to identify traffic overload. Since the extra layer differs from the original OpenFlow architecture and requires additional hardware, the analysis of the flow of records on the network controller - this can be achieved by deploying monitoring algorithms that operate on the basis of the network controller. Flow analyzes can be used to detect traffic characteristics and build a complete picture of the network by including incoming signals from all network devices. Also presented is the concept of Software-Defined-Networking.

Keywords: information network; software-configured network, information security; SDN; Linux; Ethernet; OpenFlow.