

УДК 621.446 (045)

МОДЕЛЮВАННЯ РОБОТИ OVERLAY МЕРЕЖ SDN ТА ДОСЛІДЖЕННЯ ЇХ ОСНОВНИХ ХАРАКТЕРИСТИК

Р. С. Одарченко, канд. техн. наук. доц.; С. Ю. Даков, В. В. Поліщук, А. М. Тирсенко

Національний авіаційний університет

dacov@ya.ru

Проаналізовано можливості SDN-мережі та проведено їх порівняння із традиційною IP-мережею. Проведено порівняльний аналіз варіантів побудови сучасних комп'ютерних мереж, зокрема за технологіями OpenFlow, Overlay та традиційною IP. На віртуальних моделях показано ефективність етапів переходу від NAT до Overlay та OpenFlow. Пораховані, та представлені в графічному вигляді статистичні дані, що підтверджують ефективність та перспективність мереж SDN.

Ключові слова: SDN-мережі, SDN-контролер, Overlay, OpenFlow, IP-мережа, Open vSwitch, програмний комутатор,; фізична інфраструктура, протоколи тунелювання.

The paper analyzed the possibilities, SDN-network and traditional IP network. The comparison, OpenFlow, Overlay traditional IP-network. The virtual model shows the effectiveness of transition from NAT, to Overlay and OpenFlow. Numbered and presented in graphical form, statistical data.

Keywords: SDN-network; SDN-controller; Overlay; OpenFlow, IP-network; Open vSwitch, a software switch; physical infrastructure; tunneling protocols.

Вступ

Сьогодні комп'ютерні мережі (КМ) є необхідною частиною будь-якого підприємства, навчального закладу, державної організації тощо. Сучасні КМ не позбавлені недоліків, таких як: складність управління мережею, висока вартість мережевого обладнання, недостатньо ефективне використання каналу зв'язку через передавання великої кількості інформації для управління мережею замість корисного трафіку тощо. Всі ці недоліки сприяли еволюціонуванню до нової концепції SDN (Software Defined Networking — програмно-конфігурованих мереж).

Технологію SDN у сфері телекомунікацій можна віднести до перспективної. Якщо порівнювати її з традиційною мережею, яка на сьогодні ще залишається актуальною, у тому числі і для України, модель SDN має ряд переваг, серед яких розробники вирізняють такі: підвищення ефективності мережевого обладнання на 25–30 %; зниження на 30 % витрат на експлуатацію мереж; надання користувачам можливості програмно створювати нові сервіси та оперативно завантажувати їх в мережеве обладнання.

Для реалізації сучасних мереж SDN використовують підходи із використанням протоколів OpenFlow та Overlay. Проблема полягає в тому, щоб перейти повністю на OpenFlow, операторам доведеться повністю відмовитися від інструментів IP-мережі, що не завжди буде виправдано. Тому в процесі перехідного періоду, можна використовувати мережу Overlay. Overlay-мережа не залежить від інфраструктурного рівня архітектури SDN. Завдяки цьому архітектура SDN може бути розгорнута на будь-якій існуючій IP-мережі.

Постановка проблеми

Мета даної статті — дослідження основних можливостей мереж SDN, порівняння їх із традиційними IP-мережами, визначення основних переваг та розробка рекомендацій для переходу до нової архітектури.

Аналіз досліджень та публікацій

Основна тенденція розвитку бездротових систем LTE, це перехід від мереж минулого та існуючого покоління, до мереж наступного покоління [2; 10], де планується використання концепції SDN. Кожне покоління виходить з кращими параметрами, а традиційні архітектури дизайн мереж стають неефективні в динамічних середовищах [4].

У праці [6] відзначається, що SDN-мережі, мають переваги над традиційними мережами, але мають і численні недоліки, основним з яких є дуже складний та тривалий перехід до нової архітектури [1; 6; 7]. Тому, перш ніж зробити вибір, треба зрозуміти, у яких випадках це дійсно необхідно.

Проте дані порівняння традиційних IP-мереж та SDN відсутні у вітчизняній літературі.

Цілі

Дослідження архітектури Overlay-мереж SDN. Визначити, що потрібно для побудови ПКМ-мережі.

Проаналізувати переваги й недоліки методів архітектури SDN, визначитися в технічних можливостях OpenFlow та Overlay

Розробити модель Overlay-мережі з використанням програмних комутаторів Open vSwitch з допомогою ПЗ Mininet та MiniEdit.

Результати імітаційного моделювання, використати як вихідні дані для математичного моделювання у середовищі Mathcad.

Виклад основного матеріалу дослідження

Overlay мережі SDN — це програмний спосіб реалізації SDN-мережі, заснований на використанні програмних комутаторів (наприклад Open

vSwitch), що встановлюються на віртуальні машини (VM).

На програмних комутаторах налаштовується комутація між віртуальними портами і відповідність віртуальних та фізичних портів, а вже фізичні порти зв'язуються між собою тунелями (рис. 1) [1].



Рис. 1. Архітектура Overlay-мережі SDN

Як видно з рис. 1, Overlay-мережа SDN складається з таких компонентів: SDN-контролер, гіпервізор, програмний комутатор.

У такій архітектурі використовуються протоколи тунелювання та можливості фізичної інфраструктури. Наразі можливим є моделювання роботи даної мережі в програмному середовищі MiniNet [11]. Тому скористуємось його можливостями для визначення основних характеристик мережі, зокрема затримки пакетів.

Розрахунок середньої затримки пакета в мережах IP та SDN.

Для порівняння якості передачі даних в КМ використовують такі параметри: смугу пропускання, затримку при передачі пакета, джитер та відсоток втрати пакетів.

Затримка є найбільш важливим показником якості, тому для порівняння мереж SDN/OpenFlow, SDN/Overlay та IP корисно буде порівняти середню затримку пакета.

Для оцінки середньої затримки пакета в Overlay-мережі SDN скористаємось середовищем для моделювання мереж SDN — Mininet та графічним редактором для нього — MiniEdit.

Mininet — програма, яка забезпечує емуляцію роботи мережі SDN, дозволяє створити реалістичну віртуальну мережу, а також провести її налаштування та дослідження за допомогою командного рядка.

Створимо просту мережеву топологію в середовищі MiniEdit, що складається з двох хостів h1,

h2, чотирьох програмних комутаторів Open vSwitch s1, s2, s3, s4 та мережевого контролера c0. Побудована топологія зображена на рис. 2.

Для оцінки часу передавання пакету в такій мережі скористаємось утилітою ping. Запустимо мережу кнопкою “Start” в середовищі MiniEdit, після цього в програмі Mininet з хоста 1 пропінгуємо хост 2, командою “h1 ping h2”.

Результат вводу команди зображений на рис. 3.

Як видно, перший пакет передається найдовше. Це спричинено тим, що контролер ще не встановив правила маршрутизації для потоку пакетів.

При надходженні першого пакета до комутатора s1, комутатор звернеться до контролера. Контролер визначить правила маршрутизації для такого потоку пакетів і надішле їх всім комутаторам мережі. Далі всі комутатори будуть вже налаштовані і тому кожен наступний пакет передаватиметься значно швидше.

З рис. 3 видно, що перший пакет дійшов місця призначення за 25 мс. Можна зробити висновок, що середній час обробки пакету контролером становить 25 мс.

Для визначення середнього часу передачі пакету по вже сконфігурованій контролером мережі ще раз скористаємось утилітою ping.

Візьмемо більшу кількість пакетів (більшу вибірку) для отримання більш точних результатів. Результат пінгу 102 пакетами показаний на рис. 4.

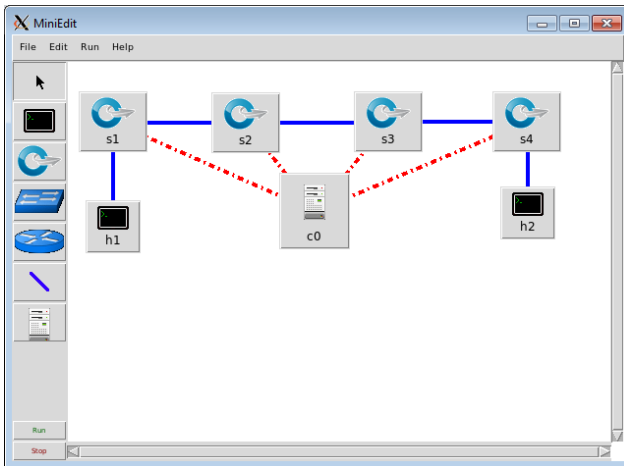


Рис. 2. Мережева топологія в середовищі MiniEdit

```

mininet@mininet-vm: ~
**** Starting 1 controllers
c0
**** Starting 4 switches
s1 s2 s3 s4
No NetFlow targets specified.
No sFlow targets specified.

NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent MiniEdit from quitting and will prevent you from starting the network again during this session.

*** Starting CLI:
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=25.0 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.00 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.083 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.083/6.561/25.076/10.696 ms
mininet> h1 ping h2

```

Рис. 3. Пінг від хоста 1 до хоста 2 у середовищі Mininet

```

mininet@mininet-vm: ~
64 bytes from 10.0.0.2: icmp_seq=59 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=61 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.109 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.157 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.198 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.160 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=101 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=102 ttl=64 time=0.082 ms
^C
--- 10.0.0.2 ping statistics ---
102 packets transmitted, 102 received, 0% packet loss, time 101152ms
rtt min/avg/max/mdev = 0.068/0.115/1.429/0.151 ms
mininet>

```

Рис. 4. Передача 102 пакетів

Як видно з рис. 4. середній час передачі пакету становить 0.115 мс.

На основі отриманих даних можливо здійснити математичне моделювання в середовищі Mathcad.

Визначимо вихідні дані для моделювання:

прийемо час передачі по каналу зв'язку за 0.002 мс (як задано в середовищі MiniEdit):

$$t_{link} := 0.002 \text{ мс} . \text{ Затримка в КЗ.}$$

Як було зазначено вище, час обробки контролером становить 25 мс:

$$t_{ctrl} := 25 \text{ мс} . \text{ Час обробки контролером.}$$

Вище було визначено середній час передачі по мережі з 4 комутаторами, який становить 0,115 мс. Звідси визначимо, скільки становить середній час обробки пакету одним комутатором. Врахуємо, що між хостами є 5 каналів зв'язку:

$$t_{ovs} := \frac{0,115 - 5t_{link}}{4} = 0,026 \text{ мс} . \text{ Час обробки}$$

пакету в OpenVSwitch-комутаторі.

Нехай OpenFlow-комутатор обробляє пакети з такою ж швидкістю, як і Open vSwitch. Час обробки пакету комутатором Open vSwitch включає

в себе також і час передачі від фізичного комутатора до віртуального і назад.

Отже, для визначення часу обробки OpenFlow віднімемо дві затримки у КЗ:

$t_{OFS} := t_{OVS} - 2t_{link} = 0,022 \text{ мс}$. Час обробки пакету в OpenFlow комутаторі.

Для визначення часу обробки пакету традиційним комутатором візьмемо близький до характеристик комутатора, що використовується середовищем Mininet. Прикладом такого комутатора є DES-1005A, який обробляє 14880 пакетів за секунду.

Час обробки пакету таким комутатором:

$$t_{switch} := \frac{1000}{1488} = 0,067 \text{ мс} . \text{ Час обробки пакету}$$

в IP-комутаторі.

У даній топології використовується 4 комутатори і 5 КЗ між хостами:

$n := 4$ Кількість комутаторів;

$m := n+1$ Кількість КЗ.

Обчислимо, скільки часу займає доставка пакетів для кожного типу мережі. Топології мереж IP, SDN/OpenFlow, SDN/Overlay зображені на рис. 5, 6, 7.



Рис. 5. Спрощена топологія IP-мережі

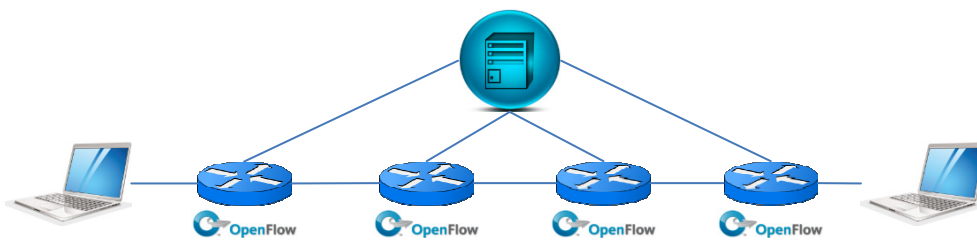


Рис. 6. Спрощена топологія SDN/OpenFlow-мережі

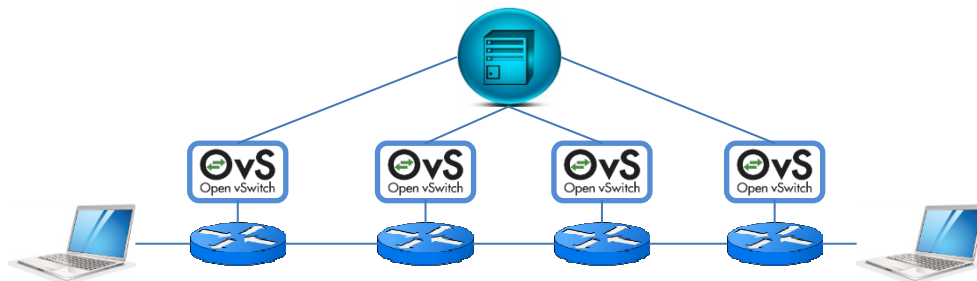


Рис. 7. Спрощена топологія SDN/Overlay-мережі

В IP-мережах, загальний час доставки пакетів становить суму затримки в кожному КЗ та часу обробки в кожному комутаторі:

$$d_{IP(p)} = mt_{link} + npt_{switch} . \text{ Для IP-комутаторів.}$$

В SDN/OpenFlow-мережах, для обрахунку загального часу доставки пакетів до суми затримки в кожному КЗ та часу обробки в кожному комутаторі додається також час доставки інформації

про перший пакет до контролера і назад, а також час прийняття рішення контролера:

$d_{SDN_OFS(p)} := mt_{link} + npt_{OFS} + 2t_{link} + t_{ctrl}$ для OpenFlow-комутаторів.

В SDN/Overlay-мережах, загальний час доставки пакетів обраховується так, як і для мереж SDN/OpenFlow, але замість часу обробки OpenFlow-комутатора використовуємо час обробки програмного комутатора Open vSwitch, який враховує час доставки пакета від фізичного до віртуального комутатора і назад:

$$d_{SDN_OVS(p)} := mt_{link} + npt_{OVS} + 2t_{link} + t_{ctrl}$$

для OpenVSwitch-комутаторів.

Для обрахунку середньої затримки пакета необхідно загальний час доставки пакетів поділити на кількість пакетів:

$$d_{average_IP(p)} := \frac{mt_{link} + npt_{switch}}{p} \text{ для IP-комутаторів;}$$

$$d_{average_SDN_OFS(p)} := \frac{mt_{link} + npt_{OFS} + 2t_{link} + t_{ctrl}}{p} \text{ для}$$

OpenFlow-комутаторів

$$d_{average_SDN_OVS(p)} := \frac{mt_{link} + npt_{OVS} + 2t_{link} + t_{ctrl}}{p} \text{ для}$$

OpenVSwitch-комутаторів

Побудуємо залежності загального часу доставки пакетів та середньої затримки від кількості пакетів у потоку даних. Графіки залежностей для трьох типів мереж зображено на рис. 9, 10. Графіки побудовані для кількості пакетів від 1 до 1000. Червона суцільна лінія — для IP-мережі, зелена пунктирна — для SDN/Overlay-мережі, лінія з синіх точок — для SDN/OpenFlow-мережі.

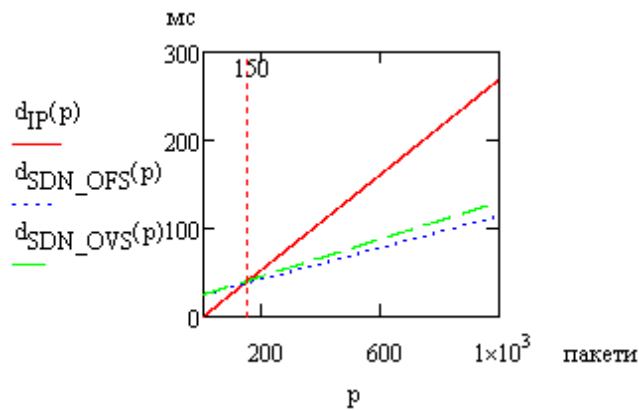


Рис. 9. Загальний час передачі даних від кількості пакетів (при $n = 4$)

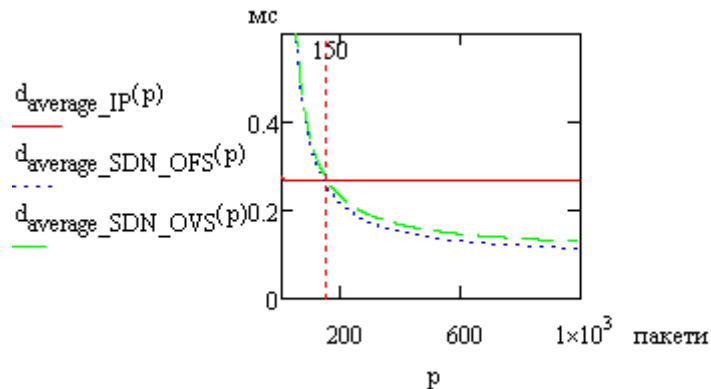


Рис. 10. Середня затримка пакета від кількості пакетів (при $n = 4$)

З графіків залежностей можна зробити висновок, що середня затримка в SDN мережі зменшується зі збільшенням кількості пакетів у потоці.

На середню затримку в традиційних мережах кількість пакетів в потоці не впливає.

Отже, підтверджена гіпотеза, що SDN ефективніше використовувати в мережах з великим об'ємом трафіку.

При цьому мережі з OpenFlow-комутаторами мають більшу ефективність, ніж мережі побудовані на Overlay, як і було зазначено в аналізі переваг підходів до реалізації ПКМ. Далі обрахуємо, як змінюється затримка в мережі залежно від кількості мережевого обладнання (комутаторів). Для чистоти обрахунків візьмемо кількість пакетів, за якою середня затримка приблизно однакова для всіх мереж.

Як видно з графіків, лінії перетинаються приблизно на позначці у 150 пакетів.

Отже зафіксуємо кількість пакетів:

$P:=150$.

Загальний час передачі даних залежно від кількості комутаторів виражено функціями:

$$d_{IP(n)} := (n+1)t_{link} + npt_{switch}$$

для IP-комутаторів;

$$d_{SDN_OFS(n)} := (n+1)t_{link} + npt_{OFS} + 2t_{link} + t_{ctrl}$$

для OpenFlow-комутаторів;

$$d_{SDN_OVS(n)} := (n+1)t_{link} + npt_{OFS} + 2t_{link} + t_{ctrl}$$

для OpenVSwitch-комутаторів.

Середня затримка пакета залежно від кількості комутаторів виражена функціями:

$$d_{average_IP(n)} := \frac{(n+1)t_{link} + npt_{switch}}{p}$$

для IP-комутаторів;

$$d_{average_SDN_OFS(n)} := \frac{(n+1)t_{link} + npt_{OFS} + 2t_{link} + t_{ctrl}}{p}$$

для OpenFlow-комутаторів;

$$d_{average_SDN_OVS(n)} := \frac{(n+1)t_{link} + npt_{OFS} + 2t_{link} + t_{ctrl}}{p}$$

для OpenVSwitch-комутаторів.

Побудуємо залежності загального часу доставки пакетів та середньої затримки від кількості комутаторів у мережі. Графіки залежностей для трьох типів мереж зображено на рис. 11, 12. Графіки побудовані для кількості комутаторів від 1 до 16.

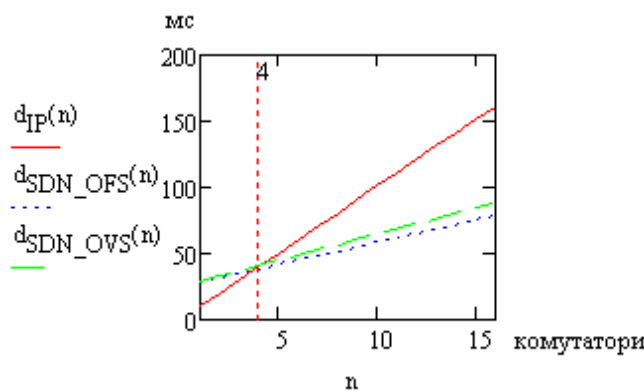


Рис. 11. Загальний час передачі даних від кількості комутаторів (при $p = 150$)

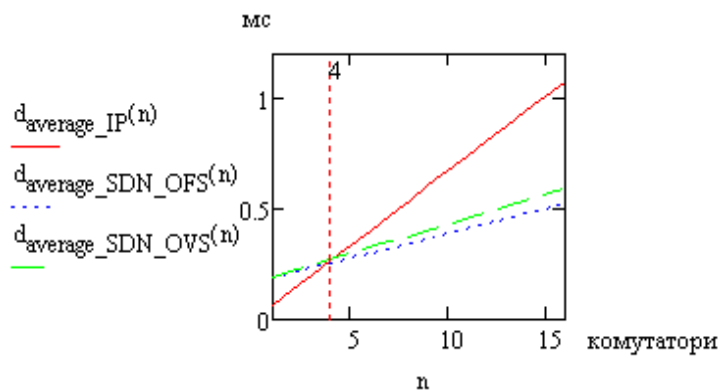


Рис. 12. Середня затримка пакета від кількості комутаторів (при $p = 150$)

Із графіків наведених вище зробимо висновок, що середня затримка в SDN-мережі збільшується зі збільшенням кількості комутаторів набагато повільніше ніж у традиційній мережі.

При 16 комутаторах маємо вигравш у затримці більш ніж у 2 рази. Отже, що SDN ефективніше використовувати в мережах з великою кількістю мережевого обладнання.

Мережі з OpenFlow-комутаторами мають більшу ефективність, ніж мережі побудовані на Overlay.

Висновки

Було проведено дослідження архітектури Overlay-мереж SDN. Визначено, що для побудови такої мережі необхідні такі елементи: SDN-контролер, гіпервізор, програмні комутатори, тунелі і протоколи тунелювання та будь-яка фізична інфраструктура традиційної IP-мережі.

Було проаналізовано переваги та недоліки двох методів реалізації SDN. Визначено, що Overlay-мережі можуть бути побудовані поверх будь-якої сучасної IP-мережі без заміни обладнання, крім того їх можливо будувати на оптич-

них мережах. Але порівняно з апаратним підходом такі мережі мають більшу затримку.

Досліджено можливості віртуального комутатора OpenvSwitch, до яких входить підтримка протоколів традиційних мереж VLAN, STP, NAT, мережних протоколів маршрутизації, протоколів тунелювання NVGRE, VXLAN, STT, протоколу OpenFlow для взаємодії з контролером, підтримка надання QoS та пріоритетизації трафіка та багато інших. Таким чином доведено, що Open vSwitch є оптимальним вибором для побудови Overlay-мереж.

Проведено моделювання роботи Overlay мережі з використанням програмних комутаторів OpenvSwitch за допомогою ПЗ Mininet та MiniEdit. Результати імітаційного моделювання використані як вихідні дані для математичного моделювання у середовищі Mathcad.

Проведено математичні розрахунки в ПЗ Mathcad, результатом яких є графіки залежностей середньої затримки від кількості пакетів у потоці та кількості комутаторів у мережі. Аналіз отриманих результатів доводить, що використання SDN мереж ефективніше за використання IP мереж при великих об'ємах трафіку та при великій кількості мережевого обладнання. Отже, доведено ефективність використання SDN для мереж провайдера ISP.

ЛІТЕРАТУРА

1. *Технологии* реализации программно-конфигурируемых сетей / Руслан Смелянский // Журнал сетевых решений/LAN. — 2014. — № 4.

2. *Скрипин В.* Глава НКРСИ: 4G в Украине появится не раньше 2016 г. [Електронний ресурс] — електронные текстовые данные — режим доступа: <http://itc.ua/news/glava-nkrsi-4g-v-ukraine-poyavitsya-ne-ranshe-2016-goda/>

3. *3G* / [Електронний ресурс] — електронні текстові дані — режим доступу: https://uk.wikipedia.org/wiki/3G#3G_D0.B2_D0.A3.D0.BA.D1.80.D0.B0.D1.97.D0.BD.D1.96

4. *Стандарты 2G*, технология GSM / [електронний ресурс] — електронные текстовые данные — режим доступа: <http://prointer.net.ua/blog/standarty-2g-tehnologiya-gsm/>

5. *Разработка* → Сетевые технологии SDN — Software Defined Networking / [Електронний ресурс] — електронные текстовые данные — режим доступа: <https://habrahabr.ru/company/muk/blog/251959/>

6. *Proactive Overlay versus Reactive Hop-by-Hop*, 2000515-002-EN Sept 2015.

7. *SDN: прорывная технология или маркетинговый пузырь?* / [Електронний ресурс] — електронные текстовые данные — режим доступа: <http://www.iksmedia.ru/articles/5002752-SDN-proryvnaya-texnologiya-ili-mark.html>

8. *SDN: кому и зачем это надо* / Александр Барсков / Журнал сетевых решений/LAN, 2012. — № 12.

9. *Mininet*. An Instant Virtual Network on your Laptop (or other PC) [Електронний ресурс]. — Режим доступа до ресурсу: <http://mininet.org/>.

10. *Тенденции* и перспективы развития мирового рынка инфраструктуры беспроводной связи (2G, 3G, LTE, WiMAX, Wi-Fi), 2012–2017 гг. / [Електронний ресурс] — електронные текстовые данные — режим доступа:

http://www.json.ru/poleznye_materialy/free_market_watches/analytics/tendencii_i_perspektivy_razvitiya_mirovogo_rynka_infrastruktury_bespровodnoj_svyazi_2g_3g_lte_wimax_wi-fi_2012-2017_gg/

11. *Production Quality*, Multilayer Open Virtual Switch [Електронний ресурс]. — Режим доступа до ресурсу: <http://openvswitch.org/>.

REFERENS

1. *Tehnologii* realizatsii programno — konfiguriruemiyh setey / Ruslan Smelyanskiy // Zhurnal setevyih resheniy/LAN. — 2014. — # 04.

2. *Скрипин В.* Глава НКРСИ: 4G в Украине поавится не рanshe 2016 goda [elektronniy resurs] — elektronni tekstovi dani — rezhim dostupu: <http://itc.ua/news/glava-nkrsi-4g-v-ukraine-poyavitsya-ne-ranshe-2016-goda/>

3. *3G* / [elektronniy resurs] — elektronni tekstovi dani — rezhim dostupu: https://uk.wikipedia.org/wiki/3G#3G_D0.B2_D0.A3.D0.BA.D1.80.D0.B0.D1.97.D0.BD.D1.96

4. *Standarty 2G*, tehnologiya GSM / [elektronniy resurs] — elektronni tekstovi dani — rezhim dostupu: <http://prointer.net.ua/blog/standarty-2g-tehnologiya-gsm/>

5. *Razrabotka* → Setevye tekhnolohyy SDN — Software Defined Networking / [elektronniy resurs] - elektronni tekstovi dani — rezhym dostupu: <https://habrahabr.ru/company/muk/blog/251959/>

6. *Proactive Overlay versus Reactive Hop-by-Hop*, 2000515-002-EN Sept 2015.

7. *SDN: proryvnaya tekhnolohiya yly marketingovyyi puzyr?* / [elektronniy resurs] - elektronni tekstovi dani — rezhym dostupu: <http://www.iksmedia.ru/articles/5002752-SDN-proryvnaya-texnologiya-ili-mark.html>

8. *SDN: кому y zchem это надо* / Aleksandr Barskov / Zhurnal setevykh resheniy/LAN, 2012. — № 12.

9. *Mininet*. An Instant Virtual Network on your Laptop (or other PC) [Elektronniy resurs] — Rezhym dostupu do resursu: <http://mininet.org/>.

10. *Tendentsyy y perspektivy razvytiya myrovoho rynka ynfrastruktury besprovodnoi svyazy* (2G, 3G, LTE, WiMAX, Wi-Fi), 2012-2017 hh. / [elektronniy resurs] - elektronni tekstovi dani — rezhym dostupu: http://www.json.ru/poleznye_materialy/free_market_watches/analytics/tendencii_i_perspektivy_razvitiya_mirovogo_rynka_infrastruktury_bespровodnoj_svyazi_2g_3g_lte_wimax_wi-fi_2012-2017_gg/

11. *Production Quality*, Multilayer Open Virtual Switch [Elektronniy resurs] — Rezhym dostupu do resursu: <http://openvswitch.org/>.

Стаття надійшла до редакції 18.08.2016