

Інфокомунікаційні системи

УДК 621.395

А.О. Гіблов, К.М. Руккас

Харківський національний університет імені В.Н. Каразіна, Харків

АНАЛІЗ ЕФЕКТИВНОСТІ АЛГОРИТМІВ КЕРУВАННЯ ПАМ'ЯТТЮ КОМУТАТОРІВ В ПРОГРАМНО-КОНФІГУРОВАНИХ МЕРЕЖАХ

У статті порівнюються чотири найбільш відомих алгоритми керування пам'яттю комутатора на предмет величини втрат пакетів. В результаті проведеної роботи, по-перше, визначено найбільш ефективний метод управління комутатором. По-друге, досягнуто раціональне використання оперативної пам'яті, що є найбільш важливим потенціалом комутатора.

Ключові слова: програмно-конфігуровані мережі, Clairvoyant, TCP-Based Heuristic, SmartTime, Cache-Based, втрат и пакетів, комутатор.

Вступ

Постановка задачі. Протягом останніх семи років ситуація з комп'ютерними мережами стрімко змінюється. Постійно зростають обсяги передачі даних. Кількість користувачів мережі, в тому числі через мобільні пристрої, зростає з кожним роком. У такій ситуації традиційні статичні мережі, основи яких закладалися в 70-х роках, не можуть повністю задовольнити сучасним вимогам. В обставинах, що склалися, програмно-конфігуровані мережі (Software-Defined Networks, SDN), що з'явилися в 2006 році, становлять великий інтерес, оскільки вони мають ті можливості, яких позбавлені традиційні мережі.

Оскільки програмно-конфігуровані мережі з'явилися порівняно недавно, багато аспектів ще досліджується. Активно розвиватися цей напрямок почав з 2011 року, і інтерес до цього типу мереж неухильно зростає. Різними авторами були запропоновані алгоритми керування пам'яттю комутатора. По кожному алгоритму авторами даються результати вимірювань заповнювання таблиці потоків комутатора і кількість втрачених пакетів. У той же час порівнянь ефективності алгоритмів створених різними авторами не проводилося.

Мета статті. Тому актуальним є питання порівняння алгоритмів керування пам'яттю комутатора, описаних в різних роботах. Алгоритми порівнювалися по середній кількості втрачених пакетів.

Аналіз останніх досліджень та публікацій. В [2] розглядаються алгоритми Clairvoyant і TCP-Based Heuristic, що зберігають кожне правило певний фіксований час. Також описується спосіб зберігання правил без використання тайм-аутів. В [3] описується алгоритм SmartTime, в якому час зберігання правила динамічно змінюється в залежності від інтенсивності отримання комутатором пакетів.

Викладення основного матеріалу

Основна ідея програмно-конфігурованих мереж – безпосередньо комутатори не зберігають ніяких налаштувань в енергонезалежній пам'яті. Замість цього кілька комутаторів зберігають адресу контролера (рис. 1). Правила обробки пакетів зберігаються централізовано на контролері, а комутатори отримують від них дані.

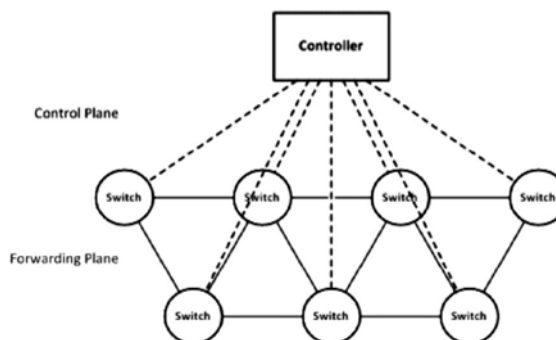


Рис.1. Загальна схема програмно-конфігурованих мереж

Загальна схема роботи таких мереж виглядає наступним чином. З кожного отриманого пакета зчитується заголовок. Далі в таблиці потоків шукається правило, для якого група полів відповідностей ідентична заголовку. Якщо правило знайдено, над пакетом виконуються дії, описані в правилі. Інструкції правил включають пересилку на певний порт комутатора, відправку на іншу таблицю потоків і зміна лічильників. Якщо в жодній таблиці не знайшлося відповідності, пакет направляється контролеру. Він формує відповідне правило для пакетів такого типу та встановлює на комутаторі правило в таблицю потоків.

Опис алгоритмів. *Алгоритм Clairvoyant.* Даний алгоритм описується в [2]. В алгоритмі для прийняття рішення про видалення потоку використовується

ся інформація про час приходу наступного пакета. Зважаючи на свою теоретичну спрямованість алгоритм присутній в даній роботі тільки для порівняння з іншими як імовірно найбільш оптимальний по співвідношенню кількості втрат/розмір таблиці потоків.

Алгоритм TCP-Based Heuristic. Даний алгоритм описується в [2]. Замість даних про час використовуються властивості TCP пакетів, а саме флаги завершення потоку даних транспортного протоколу. Частка TCP-трафіку в інтернеті становить близько 95%, так що націленість на цей протокол має право на існування.

Алгоритм SmartTime. Даний алгоритм описується в [3]. Відмінність від попередніх алгоритмів полягає в тому, що термін життя правил в таблиці потоків не фіксований. Якщо жоден потік ще не проходив через комутатор, термін існування правила невелике. З кожним наступним пакетом цей термін поступово збільшується до певної межі. В якості мінімального інтервалу пропонується 100 мілісекунд, максимального - 10 секунд.

Алгоритм Cache-Based Random. Даний алгоритм описаний в [2]. Даний спосіб керування маршрутизатором передбачає зберігання правил в таблиці потоків до тих пір, поки є місце для запису нових. Як тільки місце вичерпалося, випадковим чином вибирається правило для видалення і на його місце записується нове.

В рамках даної роботи було розроблене імітаційне середовище, в якому були реалізовані чотири розглянутих алгоритми. Всі алгоритми перебували в однакових умовах і отримували на вхід одні й ті ж дані. Порівняння проводилися на декількох наборах параметрів. Методика дослідження заснована на особливостях програмно-конфігурованих мереж і протоколу OpenFlow. Записи в таблиці маршрутизації зберігаються певний час. Найбільший період зберігання для досліджуваних алгоритмів становить десять секунд. По його закінченні або раніше за інших умов, специфічних для кожного алгоритму, запис видаляється. Якщо для пакета правило не знайдено, він відправляється до контролера, а пакет на комутаторі видаляється. Розроблена система запускає паралельно всі чотири алгоритми і підраховує кількість втрачених пакетів. Алгоритми порівнювались на трьох часових проміжках: 5, 15, 30 хвилин. Крім цього на кожному з проміжків експерименти проводилися в 3 варіантах:

- з десятима потоками, дев'ять з яких можуть одночасно зберігатися в пам'яті комутатора,
- з п'ятнадцятьма потоками, 12 з яких зберігаються в пам'яті,
- з двадцятьма потоками, 15 з яких зберігаються в пам'яті.

Система зберігає кількість втрачених пакетів за кожен 20-сек проміжок для кожного алгоритму. Тести відрізняються між собою кількістю потоків, що проходять через комутатор, і тривалістю. Умовно кожен тест можна розділити на три етапи. Перший -

етап наростання кількості потоків даних, на якому кількість потоків наростає. Він займає 1/5 від всього часу. На другому етапі кількість потоків максимальна, при цьому активні всі потоки. І на останньому етапі кількість потоків зменшується поки не досягне нуля.

Група експериментів з 10 потоками даних. Результати тесту на проміжку п'ять хвилин з десятима потоками даних показані на рис. 2. На етапі наростання кількості потоків тривалістю одну хвилину алгоритми Clairvoyant і SmartTime спочатку показують однаковий результат, але пізніше TCP-Based Heuristic показує кращий результат. Це пояснюється тим, що у цих алгоритмів дуже схожа схема видалення правил. SmartTime відразу ж відстає від усіх інших.

Для алгоритму SmartTime очевидна тенденція зменшення втрат через те, що різні потоки передаються з різною інтенсивністю. Це дає можливість збільшувати термін існування правила в комутаторі.

Середня кількість втрачених пакетів при найбільшому навантаженні складала: для алгоритму Clairvoyant - 21,4 пакетів, для алгоритму TCP-Based Heuristic - 17,7, для алгоритму SmartTime - 43 і для алгоритму Cache-Based 18,5 пакетів.

Також з графіка видно, що серед алгоритмів Clairvoyant, TCP-Based Heuristic і Cache-Based поліпшення показника для одного алгоритму не завжди означає поліпшення для інших.

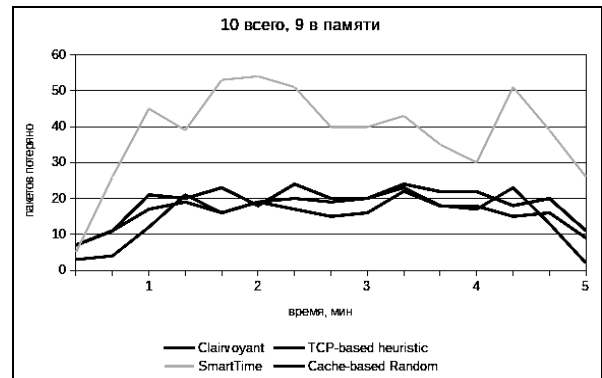


Рис. 2. Результати запуску системи на інтервалі 5 хвилин з 10 потоками, 9 з яких поміщається в пам'ять комутатора

У 2-му експерименті тривалість була збільшена з п'яти до п'ятнадцяти хвилин. Загальна кількість потоків була збільшена до п'ятнадцяти, а обсяг пам'яті комутатора збільшений до дванадцяти правил.

Для даного експерименту, результати якого показані на рис. 3, на етапі наростання кількості потоків свідчення схожі на попередній експеримент з інтервалом п'ять хвилин. За рахунок того, що час етапу став більше, а кількість потоків та сама, для алгоритмів Clairvoyant і TCP-Based Heuristic втрати збільшуються більш плавно в порівнянні з першим експериментом. Результати алгоритму SmartTime не змінилися. Для алгоритму Cache-Based втрати доходять до нуля через нерівномірне відкриття потоків. Більша їх частина з'являється на другій хвилині.

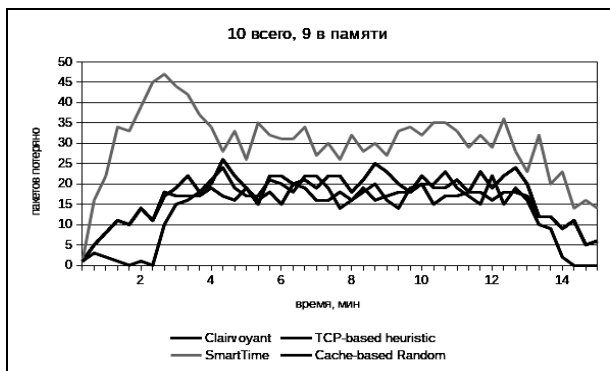


Рис.3. Результати запуску системи на інтервалі 15 хвилин з 10 потоками, 9 з яких поміщається в пам'ять комутатора

За результатами вимірювань при пікових навантаженнях, так само як і на п'ятихвилинному інтервалі, три алгоритми показали схожі результати. І лише результат алгоритму SmartTime виявився краще від попереднього.

Середня кількість втрачених пакетів при найбільшому навантаженні складала: для алгоритму Clairvoyant - 20,6 пакетів, для TCP-Based Heuristic - 17,3, для SmartTime - 31,6 і для Cache-Based 18,7 пакетів. На етапі завершення потоків результати ідентичні попереднього експерименту.

В останньому експерименті групи час роботи імітаційної системи склав 30 хвилин. Інтервали наростання потоків і їх завершення склали шість хвилин кожен. Результати, показані алгоритмами при піковому навантаженні, продовжують тенденцію двох перших експериментів. Алгоритм SmartTime показав середні втрати на цьому інтервалі такі ж, як і на інтервалі п'ятнадцять хвилин. Для трьох інших алгоритмів середні показники такі ж, як і в попередньому експерименті. Середня кількість втрачених пакетів при найбільшому навантаженні складала: для Clairvoyant - 21,1 пакетів, для TCP-Based Heuristic - 17,3, для SmartTime - 31 і для Cache-Based 18,4 пакетів. Результати експерименту показані на рис. 4.

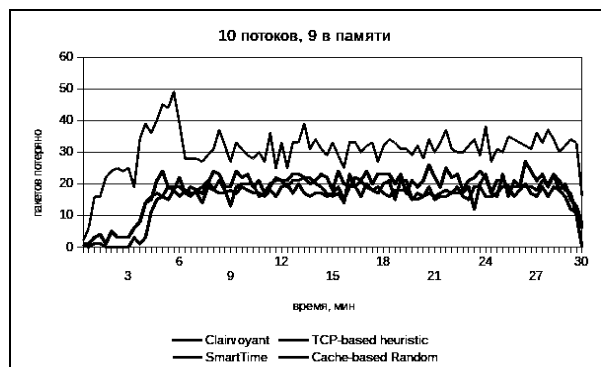


Рис. 4. Результати запуску системи на інтервалі 30 хвилин з 10 потоками, 9 з яких поміщається в пам'ять комутатора

Група експериментів з 15 потоками даних. При збільшенні кількості потоків з десяти до п'ятнадцяти з можливістю одночасно зберігати правила для дванад-

цяти з них результати, які продемонстрували алгоритми відрізняються (рис. 5). Причина цього - більша навантаження на комутатор і три потоки, які витісняються з пам'яті, замість одного. На етапі наростання кількості потоків ситуація така ж, як і на попередньому експерименті. Діапазон втрат на цьому етапі становить 5 - 80 пакетів за 20 секунд. На етапі з піковим навантаженням гірше всіх теж виявився алгоритм SmartTime. Тенденція скорочення кількості втрат тут більш наочна, ніж в попередньому експерименті.

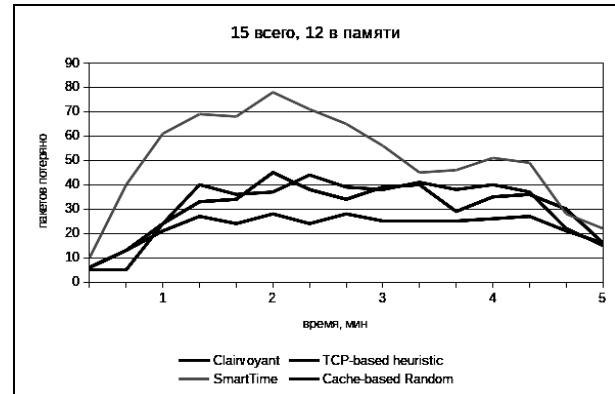


Рис. 5. Результати запуску системи на інтервалі 5 хвилин з 15 потоками, 12 з яких поміщається в пам'ять комутатора

Алгоритми Clairvoyant і Cache-Based показали схожі результати. Кількість втрачених пакетів варіюється від 25 до 45.

Алгоритм TCP-Based Heuristic показав кращий результат серед усіх алгоритмів в цьому експерименті. Завдяки ефективному управлінню правилами алгоритму потрібно менше вилучень правил при нестачі пам'яті.

Середня кількість втрачених пакетів при найбільшому навантаженні складала: для алгоритму Clairvoyant - 35,1 пакетів, для TCP-Based Heuristic - 25,3, для SmartTime - 61 і для Cache-Based 37,7 пакетів.

Етап завершення потоків все порівнювані алгоритми пройшли зі схожими показниками.

При навантаженні п'ятнадцять потоків всі алгоритми показали результат, схожий на відповідний експеримент на п'ятихвилинному інтервалі часу. Середні показники алгоритму SmartTime покращилися на 5 пакетів кожні 20 секунд. Вимірювання, отримані від інших алгоритмів, не змінилися за винятком етапу завершення потоків. На цьому етапі втрати на останній хвилині стали рівні нулю. Порівнюючи показники даного етапу з іншими експериментами ці відмінності списуються на імітаційну систему, тому що тривалість кожного потоку вибирається випадковим чином.

Середня кількість втрачених пакетів при найбільшому навантаженні складала: для алгоритму Clairvoyant - 36,6 пакетів, для TCP-Based Heuristic - 25,9, для SmartTime - 55,2 і для Cache-Based 36,5 пакетів. Результати експерименту показані на рис. 6.

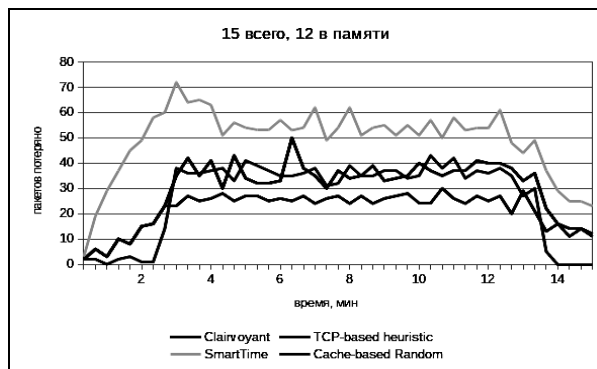


Рис. 6. Результати запуску системи на інтервалі 15 хвилин з 15 потоками, 12 з яких поміщається в пам'ять комутатора

У третьому експерименті даного набору, як і у відповідному з попереднього, результати підтверджують очікування. Всі алгоритми показують такі ж середні результати, а їх ранжування залишилося незмінним. Середня кількість втрачених пакетів при найбільшому навантаженні складала: для алгоритму Clairvoyant - 36,8 пакетів, для TCP-Based Heuristic - 21,7, для SmartTime - 40 і для Cache-Based 36,4 пакетів. Результати експерименту показані на рис. 7.

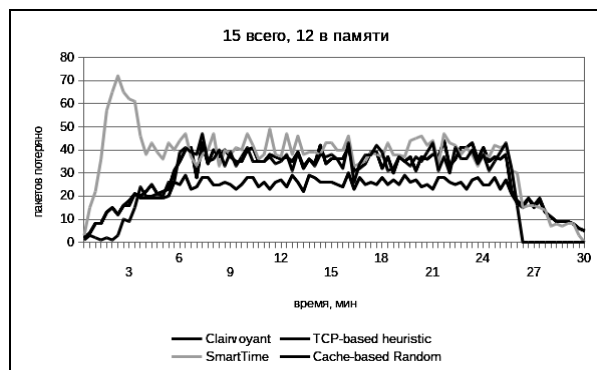


Рис. 7. Результати запуску системи на інтервалі 30 хвилин з 15 потоками, 12 з яких поміщається в пам'ять комутатора

Група експериментів з 20 потоками даних. У третій групі експериментів кількість потоків було збільшено до 20 з можливістю зберігання одночасно 15 правил. При загальному часу експерименту п'ять хвилин при піковому навантаженні видно, що при двадцяти потоках у алгоритму SmartTime збільшилися втрати пакетів. Якщо при менших навантаженнях скачок на початку основного етапу був нетривалий, то зараз, у зв'язку з ростом навантаження, таке збільшення спадає значно повільніше (рис. 8). Середня кількість втрачених пакетів при піковому навантаженні складала: для Clairvoyant - 53,6 пакетів, для TCP-Based Heuristic - 33,5, для SmartTime - 85,4, для Cache-Based - 51,6 пакетів.

На другому експерименті з набору є тільки одна відмінність. Це - періодичні стрибки втрат в зв'язку з тим, що на комутаторі збільшується кількість пакетів з різних потоків. У зв'язку з цим комутатора доводиться видаляти правила до закінчення їх терміну існування і

до завершення потоку. Середня кількість втрачених пакетів при піковому навантаженні складала: для Clairvoyant - 53,8 пакетів, для TCP-Based Heuristic - 33,8, для SmartTime - 73,8, для Cache-Based 51,8 пакетів. Результати всіх алгоритмів показані на рис. 9.

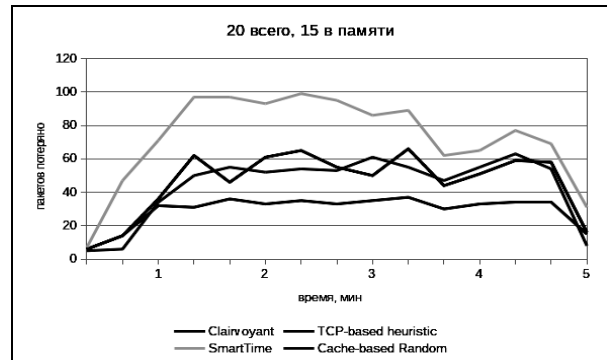


Рис. 8. Результати запуску системи на інтервалі 5 хвилин з 20 потоками, 15 з яких поміщається в пам'ять комутатора

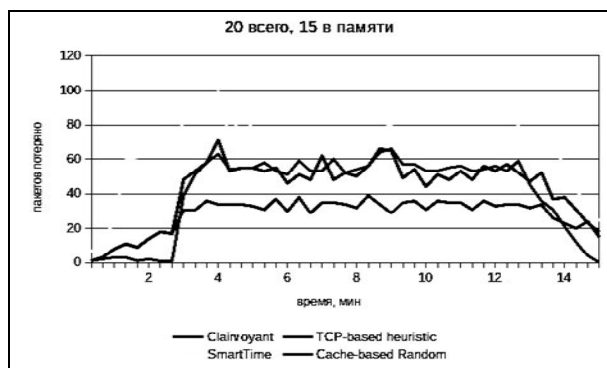


Рис. 9. Результати запуску системи на інтервалі 15 хвилин з 20 потоками, 15 з яких поміщається в пам'ять комутатора

В останньому експерименті групи з інтервалом часу 30 хвилин ситуація практично не змінилася. Як і в двох інших проміжках часу, з ростом навантаження на комутатор середня кількість втрачених пакетів зменшується. На початку пікового навантаження у алгоритму SmartTime немає стрибка втрат. Це пов'язано з тим, що з ростом кількості пакетів, що обробляються за одиницю часу ростуть і їх втрати. Середня кількість втрачених пакетів при піковому навантаженні складала: для Clairvoyant - 54,6 пакетів, для TCP-Based Heuristic - 34, для SmartTime - 79,1, для Cache-Based - 57,1 пакетів. Результати експерименту показані на рис. 10.

Аналіз отриманих результатів. Всі алгоритми показали різні результати. У кожного з них свої особливості, які вплинули на позицію того чи іншого алгоритму в підсумковому рейтингу.

Алгоритми поведуться по-різному при різній тривалості експерименту. У разі алгоритмів Clairvoyant, TCP-Based Heuristic і Cache-Based показали схожі середні втрати пакетів. Що стосується SmartTime, його середні втрати на великих проміжках часу менше, ніж на більш коротких.

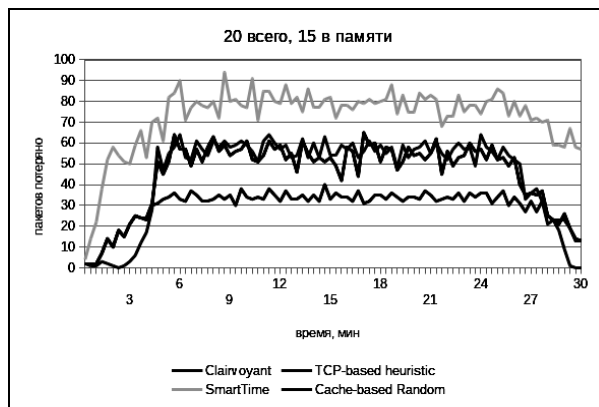


Рис. 10. Результаты запуска системы на интервале 30 минут с 20 потоками, 15 из которых помещаются в память коммутатора

За кількістю втрачених пакетів можна зробити висновки про ефективність алгоритмів. Найбільші втрати в кількісному співвідношенні, а також амплітуда втрат у алгоритму SmartTime. Головна причина - маленький початковий інтервал зберігання правила. Далі йдуть Clairvoyant і Cache-Based, що показали схожі результати. Краще за всіх виявився TCP-Based Heuristic.

У табл. 1 дані зведені результати середніх втрат пакетів у всіх дев'яти експериментах.

Таблиця 1

Середні втрати пакетів всіх алгоритмів

	Clairvoyant	TCP-Based Heuristic	Smart Time	Cache-Based
10 потоків, 9 у пам'яті				
5 хвилин	21,4	17,7	43	18,5
15 хвилин	20,6	17,3	31,6	18,7
30 хвилин	21,1	17,3	31	18
15 потоків, 12 в пам'яті				
5 хвилин	35,1	25,3	61	37,7
15 хвилин	36,6	25,9	55,2	36,5
30 хвилин	36,8	25,9	40	36,4

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ УПРАВЛЕНИЯ ПАМЯТЬЮ КОММУТАТОРОВ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

А.А. Гиблов, К.М. Руккас

В статье сравниваются четыре наиболее известных алгоритма управления памятью коммутатора на предмет величины потерь пакетов. В результате проведенной работы, во-первых, определен наиболее эффективный метод управления коммутатором. Во-вторых, достигнуто рациональное использование оперативной памяти, является наиболее важным потенциалом коммутатора.

Ключевые слова: программно-конфигурируемые сети, Clairvoyant, TCP-Based Heuristic, SmartTime, Cache-Based, потери пакетов, коммутатор.

SOFTWARE-DEFINED NETWORK SWITCH MEMORY MANAGEMENT ALGORITHMS EFFICIENCY ANALYSIS

A.O. Hiblov, K.M. Rukkas

The article compares the four most known memory management algorithms for routers in terms of packet loss amount. As a result of this work, firstly the most effective method of managing the router is identified. Second, efficient use of memory, which is the most important potential router, is reached.

Keywords: Software-Defined Networks, Clairvoyant, TCP-Based Heuristic, SmartTime, Cache-Based, packet loss, router.

Закінчення табл. 1

20 потоків, 15 в пам'яті				
5 хвилин	53,6	33,5	85,4	51,6
15 хвилин	53,8	33,8	73,8	55,8
30 хвилин	54,6	34	79,1	57,1

Висновки

При виборі різних алгоритмів різниця за кількістю втрат пакетів може відрізнятись в 1,5 - 2 рази. Тому від грамотного вибору алгоритму управління комутатором залежить загальна продуктивність мережі, що в разі великих навантажень виходить на перше місце за вимогами.

В ході проведення експериментальної частини було встановлено, що оптимізація інтервалів зберігання правил не завжди призводить до зниження втрат пакетів комутатором. Найбільш ефективним з усіх розглянутих виявився метод TCP-Based Heuristic, що видаляє правила по завершенні всіх потоків даних, що асоціюються з конкретним правилом.

Список літератури

1. Программно-конфигурируемые сети [Электронный ресурс]. – Режим доступа: www.osp.ru/os/2012/09/13032491.
2. Zarek A. OpenFlow Timeouts Demystified [Электронный ресурс] / A. Zarek. – Department of Computer Science University of Toronto. – Режим доступа: http://www.eecg.toronto.edu/~lie/papers/zarek_mscthesi.pdf.
3. Effective Switch Memory Management in OpenFlow Networks [Электронный ресурс] / A. Vishnoi, R. Poddar, V. Mann, S. Bhattacharya. – IBM Research. – Режим доступа: <http://researcher.watson.ibm.com/researcher/files/in-rishpodd/SmartTime.pdf>.
4. OpenFlow Switch Specification. Version 1.5.1 [Электронный ресурс]. – Режим доступа: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf>.

Надійшла до редколегії 11.06.2016

Рецензент: д-р техн. наук, проф. Г.А. Кучук, Харківський національний університет Повітряних Сил імені Івана кожедуба, Харків.