

МЕТОД ОБРОБКИ ТРАФІКУ В ЧЕРГАХ LINUX-МАРШРУТИЗАТОРІВ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

У статті розглянуто застосування нечіткого регулятора для обробки трафіку в черзі Linux-маршрутизатора.

Регулятор на основі нечіткої логіки FLC (Fuzzy Logic Controller) широко застосовується в системах управління для контролю за складними процесами нелінійною динамікою і добре підходить до задач обробки трафіку в чергах маршрутизаторів при перевантаженнях в мережі. З допомогою регулятора FLC можливо запобігати виникненню перевантажень і автоматично вибирати оптимальне значення для ймовірності скидання або маркування пакетів в маршрутизаторі, щоб регулювати інтенсивність вхідного навантаження при великій кількості TCP-сесій і обмеження швидкості в каналі передачі даних. Здатність регулятора FLC утримувати довжину черги біля заданого значення дозволить прогнозувати затримку пакетів у черги і зменшити джиттер.

Спільне застосування методу FLC з технологією явного повідомлення про перевантаження ECN здатне скоротити втрати пакетів TCP-з'єднань до мінімуму за рахунок використання маркування пакетів замість їх скидання. Використання відкритої платформи на базі операційної системи Linux, дає можливість реалізувати власний метод обробки трафіку на маршрутизаторі, а також скористатися вже реалізованими в Linux технологіями контролю та управління IP-трафіком. Нечіткий регулятор завантажується в маршрутизатор у вигляді розробленого програмного модуля для ядра Linux.

Для оцінки параметрів якості обслуговування трафіку, таких як затримка пакетів при передачі, джиттер, відсоток втрачених пакетів і коефіцієнт використання каналу, при використанні методу FLC, проведені тестові випробування маршрутизатора, як у віртуальному оточенні, так і на апаратній платформі. Одночасно через канал з обмеженою швидкістю передавалося до 100 TCP-сесій плюс трафік UDP с постійною швидкістю.

Не зважаючи на перевантаження розроблений метод FLC продемонстрував здатність утримувати довжину черги біля заданого значення і не допускати перепоповнення і спустошення черзі в момент перевантаження.

Результати випробувань показують перевагу розробленого методу перед традиційними FLC методами Tail Drop і RED в режимі перевантаження в мережі.

Ключові слова: активне управління чергами, запобігання перевантажень, нечітка логіка, FLC, Linux-маршрутизатор, TCP/IP.

Вступ. Стрімкий розвиток мережевих технологій і збільшення кількості користувачів приводить до нерівномірності пропускних здатностей каналів передачі даних і перепадів інтенсивності трафіку. Звідси, неминуче виникнення перевантажень на ділянках мережі. Для запобігання цьому в чергах маршрутизатора використовують активні методи керування але вони не завжди справляються з складною динамікою трафіку і нелінійною зміною навантажень. У зв'язку з цим погіршуються такі параметри якості обслуговування трафіку, як ефективна швидкість передачі даних, процент втрачених пакетів і затримки.

Регулятор на основі нечіткої логіки FLC (Fuzzy Logic Controller) широко застосовується в системах управління для контролю за складними процесами нелінійною динамікою і добре підходить до задач обробки трафіку в чергах маршрутизаторів при перевантаженнях в мережі. З допомогою регулятора FLC можливо запобігати виникненню перевантажень.

Метою даної статті є аналіз проведення експериментального дослідження нового механізму обробки трафіку в чергах маршрутизаторів на основі регулятора з нечіткою логікою FLC. Для запобігання перевантажень в мережі необхідна перевірка ефективності методу у вигляді програмного модуля для Linux-маршрутизатора. Такий модуль дозволяє регулювати

інтенсивність вхідного навантаження при великій кількості TCP-сесій і обмежувати швидкість в каналі передачі даних, а також здатний за допомогою регулятора FLC утримувати довжину черги біля заданого значення яка дозволить прогнозувати затримку пакетів у чергах і зменшить джиттер.

Виклад основного матеріалу Для експериментальної перевірки розробленого методу управління чергами на основі нечіткої логіки [1-3] було вибрано ядро операційної системи Linux, оскільки воно вільно доступне у вихідних кодах на мові програмування C на сайті проекту [4] і може бути скомпільована для великої кількості апаратних архітектур. У ядрі Linux вже є підтримка безлічі мережевих технологій, таких як стек протоколів TCP/IP, технології диференційованого обслуговування DiffServ[5], включаючи різні методи управління трафіком і чергами, наприклад метод управління чергами та запобігання перевантажень RED, а також можливе додавання нових методів. В якості транспортного протоколу за замовчуванням використовується реалізація TCP Cubic. Є також підтримка методу повідомлення про перевантаження ECN(RFC-3168), але вона відключена за замовчуванням. Таким чином, більшість мережевих технологій вже є і нам необхідно тільки впровадити свій метод управління у вигляді модуля ядра.

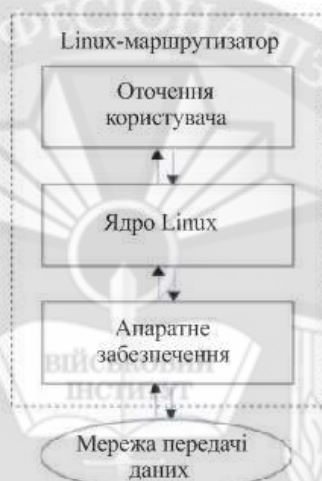


Рис. 1. Структура Linux-маршрутизатора

Безпосередньо з апаратним забезпеченням маршрутизатора взаємодіє ядро Linux і якщо отриманий на мережевому інтерфейсі IP-пакет, адресований локальному IP-адресу, то ядро передає пакет в користувацьке оточення обслуговуючому додатку. Схема проходження пакетів від мережевого інтерфейсу маршрутизатора через ядро Linux показана на рис. 1.

Для управління трафіком в ядрі використовується додаток tc (Traffic Control) [6], який працює в користувацькому оточенні. З допомогою користувацького додатка tc можливо створювати ієрархічні черги, призначати дисципліни обслуговування, проводити класифікацію трафіку і застосовувати політику обслуговування в залежності від класу трафіку. Для обробки пакетів, що надходять на вхід мережевого інтерфейсу, в ядрі організована вхідна дисципліна обслуговування, названа "Ingress Qdisc" на рис. 2. Далі в залежності від того, якщо IP-пакет, адресований локальному додатку, то пакет надходить в IP-стек для розпакування даних для інтерфейсу програми. Якщо одержувачем пакету є інший IP-адресу, то пакет надсилається на відповідний вихідний мережевий інтерфейс (пересилання пакетів повинна бути активована в ядрі). Вихідний трафік перед передачею в мережу може класифікуватися в блоці "Egress Classifier", і в залежності від присвоєного класу відправляється в окрему чергу зі своєю дисципліною обслуговування.

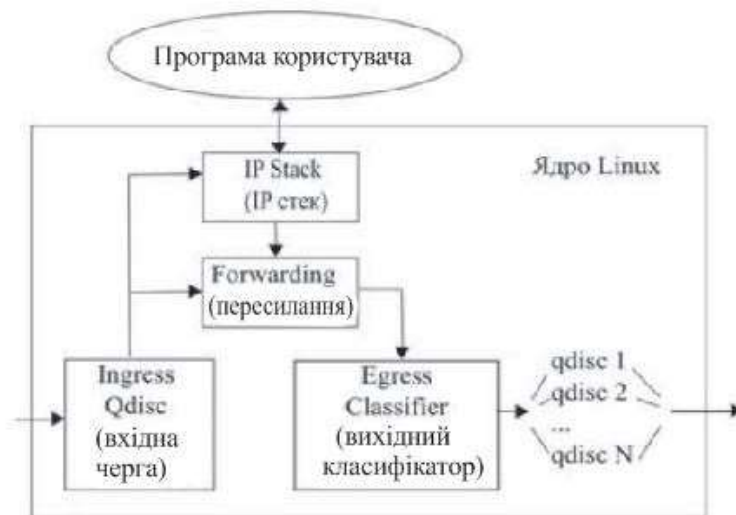


Рис. 2. Управління трафіком в Linux-маршрутизаторі

Для конфігурації розробленого модуля використовувалася модифікована утиліта tc. В якості основного методу контролю трафіку застосовувався ієрархічний метод "Корзини з жетонами" (Hierarhial Token Bucket - НТВ) [7]. Вхідний трафік поділявся на два класи: клас 1:10 для трафіку керування та клас 1:5 для тестового трафіка. Для тестового класу трафіку встановлено обмеження на максимальну швидкість передачі даних 50Мбіт/с і далі цей трафік обробляється дисципліною обслуговування FLC з максимальною довжиною черги 500 кбайт і заданою довжиною 250 кбайт. Схема класифікації трафіку наведена на рис. 3.

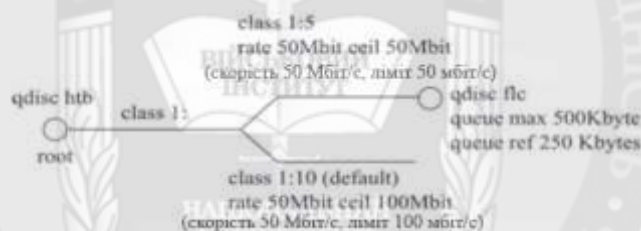


Рис. 3. Схема класифікації вихідного трафіку

Параметри утиліти tc для створення описаної вище конфігурації приведені в лістингу 1. Лістинг 1. Створення дисциплін обслуговування з допомогою утиліти tc

1. tc qdisc add dev eth0 root netem delay 10ms
2. tc qdisc replace dev eth0 root handle 1: htb default 10
3. tc class add dev eth0 parent 1: classid 1:1 htb rate 100Mbit
4. tc class add dev eth0 parent 1:1 classid 1:10 htb rate 50Mbit ceil 100Mbit prio 1
5. tc class add dev eth0 parent 1:1 classid 1:5 htb rate 50Mbit ceil 50Mbit prio 2
6. tc qdisc add dev eth0 parent 1:5 handle 5: flc limit 500K qref 250000 d_scale 12500 sampling 6 p_rate0 50000 q_lim 500000ecn
7. tc filter add dev eth0 parent 1:0 prio 1 protocol ip handle 5 fw flowid 1:5

У рядку 1 задається затримка 10 мс. Рядок 2 встановлює основний метод управління НТВ з класом за замовчуванням 10, а рядок 3 - максимальній загальну швидкості 100 Мбіт/с. Рядок 4 встановлює параметри для класу 1:10 з гарантованою швидкістю 50 Мбіт/с і максимальної швидкістю 100 Мбіт/с для трафіку управління. Рядок 5 встановлює параметри для класу 1:5 з гарантованою швидкістю 50 Мбіт/с і максимальної швидкістю теж 50 Мбіт/с для тестового трафіку. Далі в рядку 6 вказується, що трафік класу 1:5 треба обробляти методом flc з

відповідними параметрами. Значення функції управління для дисципліни обслуговування з нечітким регулятором FLC розраховувались попередньо на основі заданих параметрів, і передавалися в модуль ядра у вигляді таблиці цілих чисел, для уникнення обчислень з плаваючою крапкою всередині ядра Linux. В останньому рядку 7 задається правило фільтра, який буде призначати в клас 1:5 спеціально промарковані пакети тестового трафіку. Попередня маркування пакетів виконувалася за допомогою користувальницького додатка iptables, що є стандартним додатком Linux.



Рис. 4. Схема проходження пакетом ланцюжка фільтрів в Linux-маршрутизаторі

Вхідні пакети обробляються в блоці «Prerouting», далі в залежності від рішення про маршрутизацію пакета надходить в користувальницьке оточення через блок «Input» або проходить через блок «Forward» до вихідного порту. Якщо пакет відправляється з користувальницького оточення, то він проходить через блок «Output». Перед відправкою в мережу всі пакети можуть оброблятися в блоці фільтрів «Postrouting».

Приклад попередньої маркування тестового трафіку для його класифікації за допомогою утиліти iptables під час експерименту, приведений в лістингу 2.

Лістинг 2. Маркування трафіку для класифікації

```
iptables -A OUTPUT -t mangle -p tcp --dport 5001 -j MARK --set-mark 5
```

Таким чином, користувачу TCP-трафіком, що має TCP-порт призначення 5001 в ланцюжку «OUTPUT», буде присвоєна маркування з індексом 5, і далі пакети, марковані цим індексом будуть класифіковані у відповідний клас обслуговування 1:5, який обробляється в черзі з дисципліною обслуговування FLC.

Випробування методу FLC для обробки трафіку на віртуальній машині

Для експерименту по обробці трафіку в черзі маршрутизатора була запущена операційна система OpenWRT Linux [8] у віртуальному оточенні Oracle VM VirtualBox [9]. Дисципліна обслуговування виділеної черги для тестового трафіку flc була налаштована для проведення вимірювання довжини черги і інтенсивності трафіку кожні 6 мс і зміни, з допомогою нечіткого регулятора. ймовірності скидання/маркування пакетів на величину не більше $8 \cdot 10^{-5}$ за один інтервал вимірювання. Підтримка явного повідомлення про перевантаження ECN була включена для TCP передавачів. Схема експерименту наведена на рис. 5.

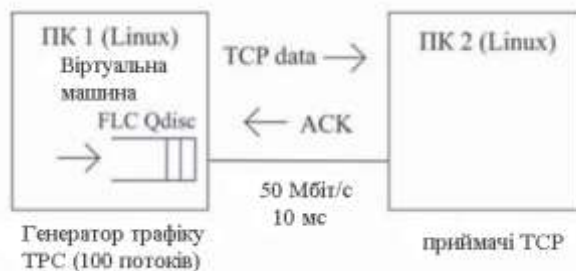


Рис. 5. Схема експерименту

У якості генератора TCP та UDP трафіку використовувалася утиліта iperf [10]. С допомогою цієї утиліти створювалося 100 потоків TCP з розміром TCP сегмента 964 байта. У

початковий момент часу всі 100 TCP передавачів встановлювали з'єднання і починали відправку пакетів. Через 100 секунд після початку 50 передавачів припиняли передачу і пакети передавали тільки 50 TCP передавачів. Після 100 секундного переривання поновлювалася передача всіх 100 передавачів. Таким чином імітувалася стрибкоподібна зміна інтенсивності трафіку. Загальний час експерименту склало 300 секунд. Додатково до TCP трафіку передавався також UDP трафіку зі швидкістю 128 кбіт/с і з розміром дейтаграми 980 байт. Зміна довжини черги маршрутизатора за час експерименту наведено на рис. 6.

Рис. 6 показує, що розроблений метод управління чергою FLC може утримувати довжину черги біля заданого значення 250 Кбайт, незважаючи на стрибкоподібну зміну інтенсивності трафіку і перевантаження.

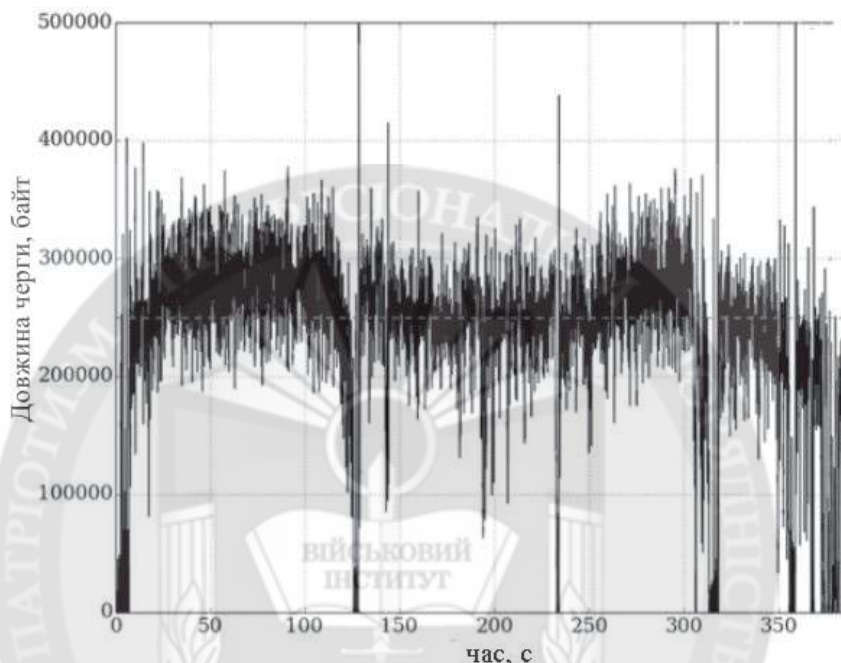


Рис. 6. Зміна довжини черги в Linux-маршрутизаторі при використанні методу FLC

На рисунку також видно, що всі 100 TCP з'єднання не встановлюються одночасно, і що потрібно близько 25 секунд для того щоб почати передачу. Через 300 секунд експерименту передача даних не закінчується різко, і потрібно ще близько 100 секунд, щоб завершити всі з'єднання. За час експерименту через 100 TCP з'єднань було передано 2 Гбайт даних (або 2 млн пакетів) доступна смуга пропускання каналу була повністю утилізована. Скидання пакетів TCP з черги через переповнення було незначним (близько сотні пакетів), завдяки застосуванню маркування пакетів і повідомлення про перевантаження за допомогою технології ECN. Скидання пакетів UDP склав 3% і джиттер 3 мс. Більший відсоток втрати пакетів UDP по порівнянні з TCP, пояснюється тим, що застосовуваний метод FLC використовував маркування пакетів TCP замість їх скидання, а пакети UDP, скидалися, бо вони не мають механізму маркування [11].

Подібний сценарій тестування був застосований також для методу RED і Tail Drop, для того щоб виконати порівняння параметрів якості при використанні різних методів управління. Експеримент з кожним методом був повторений 5 раз для кожного методу для оцінки інтервалів вимірних характеристик. Для методу RED використовувався мінімальний поріг довжини черги 125 Кбайт, максимальний поріг 375 Кбайт і максимальна довжина черги 500 Кбайт. Тобто, у відповідності з рекомендаціями максимальний поріг у 3 рази вище мінімального, а середня довжина черги відповідала 250 Кбайт і дорівнювала заданій довжині черги для методу FLC. Ймовірність скидання для методу RED була встановлена 0,02. Підтримка ECN також включена. Для методу Tail Drop розмір черги дорівнював 500 Кбайт,

технологія ECN методом Tail Drop не підтримується [12]. Середні значення втрат пакетів для 5 експериментів трьома методами: FLC, Tail Drop і RED наведено на рис. 7.

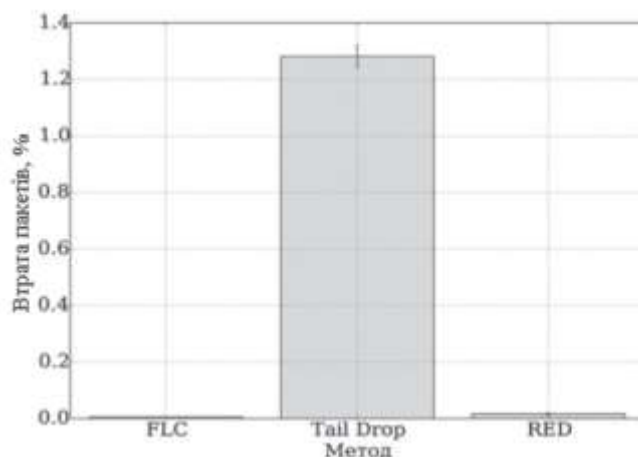


Рис. 7. Середні значення втрат пакетів

Завдяки застосуванню ECN втрати при використанні методів FLC і RED значно нижче, ніж для методу Tail Drop. Метод Tail Drop допускає значні втрати через переповнення черги. Середні величини значення джиттера за рівнем 95% трафіка UDP наведено на рис. 8.

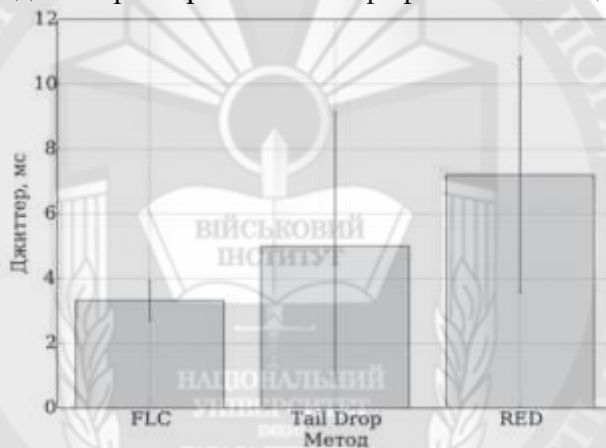


Рис. 8. Середні значення джиттера для UDP-потoku

Найменше значення джиттера зафіксовано для методу FLC. Таким чином метод FLC показав кращу здатність по стабілізації довжини черги серед випробуваних методів. Метод RED показав більшу величину джиттера тому, що контролює лише середню довжину черги в межах встановлених порогів і допускає коливання миттєвого значення довжини черги. Метод Tail Drop показав нестабільна поведінка довжини черги.

Випробування методу FLC для обробки трафіку в черзі маршрутизатора

Для впровадження методу обробки трафіку в чергах на основі нечіткого регулятора, код розробленого програмного модуля ядра був підготовлений для апаратного маршрутизатора WL-500gP. Технічні характеристики маршрутизатора наведено в табл. I.

Таблиця 1

Технічні характеристики маршрутизатора

Характеристика	Значення
Архітектура	MIPS
Система на кристалі	Broadcom BCM5354
Оперативна пам'ять	32 Мбайт
Частота процесора	240 МГц

Маршрутизатор був підключений між двома тестовими комп'ютерами, виконуючий роль генератора і приймача трафіку, причому один з комп'ютерів був підключений по бездротовому каналу WLAN по стандарту IEEE 802.11 g. Швидкість в каналі передачі даних була обмежена до 15 Мбіт/с. Схема підключення маршрутизатора наведена на рис. 9.



Рис. 9. Схема підключення маршрутизатора

Середнє значення двосторонньої затримки RTT склало 2,5 мс. Через маршрутизатор передавався трафік зі стрибкоподібною зміною інтенсивності. На всіх пристроях була включена підтримка протоколу ECN. Кількість TCP-з'єднань різко змінювалося з 50 потоків до 25 і назад до 25 з'єднань через інтервали за 100 секунд. Одночасно з TCP трафіком передавався тестовий потік UDP з постійною швидкістю 128 кбіт/с. Розмір пакетів для обох типів трафіку - 1000 байт. Максимальний розмір вихідної черги маршрутизатора був встановлений 500 кбайт, а задане значення Q_{ref} для методу FLC дорівнювало 250 кбайт. Середнє значення довжини черги за час експерименту склала 308 Кбайт при середньоквадратичному відхиленні 55кбайт. Загальні втрати пакетів - 0,027%. Значення джиттера UDP склало 20 мс.

Висновки. Розроблений метод для обробки трафіку в чергах і реалізований на базі Linux-маршрутизатора показав стійку роботу в режимі перевантаження при 100% використанні каналу передачі даних. Метод управління FLC спільно з використанням протоколу ECN продемонстрував здатність утримувати довжину черги біля заданого значення і не допускати переповнення буфера і втрат пакетів через переповнення.

ЛІТЕРАТУРА:

1. Деарт В.Ю. Применение нечёткого регулятора для стабилизации длины очереди маршрутизатора / В.Ю. Деарт, А.Г. Масленников. – М.: РУДН, 2012. – С. 83-85. – (Тезисы докладов Всероссийской конференции с международным участием. – "Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем» (ИТТММ-2012)).
2. Деарт В.Ю. Исследование влияния параметров канала передачи данных на процедуры управления очередью / В.Ю. Деарт, А.Г. Масленников –Т-Comm – Телекоммуникации и Транспорт. – 2012. – № 7. – С. 77-81.
3. Deart V. Fuzzy logic queue discipline processing over bottleneck link/ V. Deart, V.Maslennikov – Proc. of 11th Conference of Open Innovations Association Finnish-Russian University of Cooperation in Telecommunications (FRUCT-11), St.-Petersburg, State University of Aerospace Instrumentation (SUAI), pp. 40-46.
4. TheLinuxKernelArchives. [Електронний ресурс].Режимдоступу:<https://www.kernel.org>
5. Almasberger W. Differentiated Services on Linux. [Електронний ресурс].Режим доступу:<https://infoscience.epfl.ch/record/195>
6. Hubert B. Linux Advanced Routing & Traffic Control HOWTO[Електронний ресурс]. Режим доступу : <http://lartc.org/howto>
7. Гайдамака Ю.В.Об одной системе массового обслуживания с активным управлением очередью /Ю.В. Гайдамака ,А.Г. Масленников// Вестник РУДН. Серия «Математика. Информатика. Физика». – 2013. – № 4. – С. 56–64.
8. OpenWRTLinux. [Електронний ресурс]. Режим доступу : <https://openwrt.org>
9. OracleVMVirtualBox. [Електронний ресурс]. Режим доступу : <https://www.virtualbox.org>

10. NLANR/DAST, Iperf. [Электронный ресурс]. Режим доступа : <http://iperf.sourceforge.net>
11. Благодаров А.В. Моделирование и синтез оптимальной структуры Ethernet./ А.В. Благодаров, А.Н. Пыльнин, Д.М. Скуднев, А.П. Шибанов/ – :Издательство «Гор. Линия-Телеком» , 2011. – 112 с.
12. Коваленко Т.Н. Модель активного управления очередями в распределенных инфокоммуникационных системах, представленная сетью Петри / Т.Н. Коваленко // журнал «Проблемы телекоммуникаций». – 2012. – № 2(7). – С.58-67.

REFERENCES:

1. Deart V.Iu. Prymenenye nechYotkoho rehuliatora dlia stabylyzatsyy dlynyi ocheredy marshrutyzatora / V.Iu. Deart, A.H. Maslennykov – M.: RUDN, 2012. S. 83-85. – (Tezysyi dokladov Vserossyiskoi konferentsyy s mezhdunarodnym uchastyem. – "Ynformatsyonno-telekommunykatyonyie tekhnolohyy u matematycheskoe modelyrovanye vyisokotekhnolohychnyikh system» (YTTMM-2012)".
2. Deart V.Iu. Yssledovanye vliyaniya parametrov kanala peredachy dannyikh na protseduryi upravleniya ocherediu / V.Iu. Deart, A.H. Maslennykov – T-Comm - Telekommunykatyuy u Transport. - 2012. - No 7. - S. 77-81.
3. Deart V. Fuzzy logic queue discipline processing over bottleneck link/ V. Deart, V.Maslennikov – Proc. of 11th Conference of Open Innovations Association Finnish-Russian University of Cooperation in Telecommunications (FRUCT-11), St.-Petersburg, State University of Aerospace Instrumentation (SUAI), pp. 40-46.
4. The Linux Kernel Archives. [Elektronnyi resurs]. Rezhym dostupu : <https://www.kernel.org>
5. Almasberger W. Differentiated Services on Linux. [Elektronnyi resurs].Rezhim dostupu:<https://infoscience.epfl.ch/record/195>
6. Hubert 8. Linux Advanced Routing & Traffic Control HOWTO [Elektronnyi resurs]. Rezhym dostupu : <http://lartc.org/howto>
7. Gaydamaka Yu.V. Obodnoysistememassovogoobsluzhivaniyasaktivnyimupravleniemocheredyu / Yu.V. Gaydamaka ,A.G. Maslennikov // VestnikRUDN. Seriya «Matematika. Informatika. Fizika». – 2013. – # 4. – S. 56–64.
8. OpenWRT Linux. [Elektronnyi resurs]. Rezhym dostupu : <https://openwrt.org>
9. Oracle VM VirtualBox. [Elektronnyi resurs]. Rezhym dostupu : <https://www.virtualbox.org>
10. NLANR/DAST, Iperf. [Elektronnyi resurs]. Rezhym dostupu : <http://iperf.sourceforge.net>
11. Blagodarov A.V. Modelirovanie i sintez optimalnoy strukturyi Ethernet./ A.V. Blagodarov, A.N. Pyl'nin, D.M. Skudnev, A.P. Shibanov// – :Izdatelstvo «Gor. Liniya-Telekom» , 2011. – 112 s.
12. Kovalenko T.N. Model aktivnogo upravleniya ocheredyami v rasredelennyih infokommunikatsionnyih sistemah, predstavlenaya setyu Petri / T.N. Kovalenko // zhurnal «Problemyi telekommunikatsiy». – 2012. – # 2(7). – S.58-67.

Рецензент: д.т.н., проф. Шворов С.А., Національний університет біоресурсів і природокористування

к.т.н., доц. Огневой А.В., к.т.н., с.н.с. Красник А.В.,
Гунченко С.Ю., Самсонюк Б.С.

МЕТОД ОБРАБОТКИ ТРАФИКА В ОЧЕРЕДЯХ LINUX-МАРШРУТИЗАТОРОВ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

В статье рассмотрено применение нечеткого регулятора для обработки трафика в очереди Linux-маршрутизатора.

Регулятор на основе нечёткой логики FLC (Fuzzy Logic Controller) широко применяется в системах управления для контроля за сложными процессами с нелинейной динамикой и хорошо подходит для задач обработки трафика в очередях маршрутизаторов при перегрузках в сети. С помощью регулятора FLC возможно предупреждать возникновение перегрузок и автоматически выбирать оптимальное значение для вероятности сброса или маркировки пакетов в маршрутизаторе, чтобы регулировать интенсивность поступающей нагрузки при большом количестве TCP-сессий и ограниченной скорости в канале передачи данных. Способность регулятора FLC удерживать длину очереди около заданного значения позволит прогнозировать задержку пакетов в очереди и уменьшит джиттер.

Совместное применение метода FLC с технологией явного уведомления о перегрузки ECN способно сократить потери пакетов для TCP-соединений до минимума за счёт использования

маркировки пакетов вместо их сброса. Использование открытой платформы на базе операционной системы Linux, даёт возможность реализовать собственный метод обработки трафика в маршрутизаторе, а также воспользоваться уже реализованными в Linux технологиями контроля и управления IP-трафиком. Нечёткий регулятор загружается в маршрутизатор в виде разработанного программного модуля для ядра Linux.

Для оценки параметров качества обслуживания трафика, таких как задержка пакетов при передаче, джиттер, процент потерянных пакетов и коэффициент использования канала, при использовании метода FLC, проведены тестовые испытания маршрутизатора, как в виртуальном окружении, так и на аппаратной платформе. Одновременно через канал с ограниченной скоростью передавалось до 100 TCP-сессий плюс трафик UDP с постоянной скоростью.

Несмотря на перегрузку разработанный метод FLC продемонстрировал способность удерживать длину очереди около заданного значения и не допускать переполнений и опустошений очереди в момент перегрузки.

Результаты испытаний показывают преимущество разработанного метода FLC перед традиционными методами Tail Drop и RED в режиме перегрузки в сети.

Ключевые слова: активное управление очередями, предотвращение перегрузок, нечёткая логика, FLC, Linux-маршрутизатор, TCP/IP.

Ph.D. Assoc. Ognevyy A.V., Ph.D. Krasnik A.V.

Gunchenko S.Yu., Samsoniuk B.S.

METHOD OF PROCESSING TRAFFIC IN A QUEUE OF LINUX-BASED ROUTERS BASED ON FUZZY LOGIC

The article discusses the use of a fuzzy controller for traffic handle in the queue of the Linux router.

Regulator on the basis of fuzzy logic FLC (Fuzzy Logic Controller) is widely used in control systems to control complex processes with nonlinear dynamics and is well suited for traffic processing tasks in queues of routers when network is overloaded. With the FLC controller, it is possible to prevent overload and automatically select the optimal value for the probability of dropping or marking packets in the router to control the input load intensity for a large number of TCP sessions and speed limits in the data link. The ability of the FLC controller to hold the queue length near the given value will allow to predict packet delay in queues and to reduce jitter.

Shared usage of the FLC method with explicit message technology about the ECN overload can reduce the loss of packet TCP connections to a minimum through the use of marking packages instead of resetting them. Usage of an open platform on the basis of Linux operating system gives you the opportunity to implement your own method of handling traffic on the router, as well as to use the already implemented in Linux technologies for monitoring and managing IP traffic. The fuzzy controller is loaded into the router in the form of a developed software module for the Linux core.

In order to evaluate the quality parameters of traffic servicing, such as packet delay, jitter, percentage of lost packets and channel usage rate, usage of the FLC method, accompanied with test of the routers both in the virtual environment and on the hardware platform. Simultaneously, through the channel with limited speed were transmitted up to 100 TCP sessions plus UDP traffic with constant speed.

Despite the overload, the developed method of the FLC demonstrated the ability to hold the queue length near the specific value and prevent overflow and emptying of the queue at the moment of the overload. The test results showed the advantage of the developed method to traditional FLC Tail Drop or RED methods in network overload mode.

Keywords: active queue management, congestion avoidance, fuzzy logic, FLC, Linux-router, TCP/IP.