

УДК 004.715

Т. П. ДОВЖЕНКО, аспірант,  
Государственный университет телекоммуникаций, Киев

## ИССЛЕДОВАНИЕ ОЧЕРЕДИ МАРШРУТИЗАТОРА В СЕТИ TCP/IP С ПРИМЕНЕНИЕМ ОСНОВНЫХ TCP-АЛГОРИТМОВ АКТИВНОГО УПРАВЛЕНИЯ ОЧЕРЕДЬЮ И ПРЕДОТВРАЩЕНИЯ ПЕРЕГРУЗОК

**Рассмотрены основные разновидности TCP-протоколов предотвращения перегрузок сети (TCP-Reno, TCP-Vegas) и проведено исследование сети TCP/IP с применением указанных протоколов.**

**Ключевые слова:** TCP-Reno; TCP-Vegas; RED; REM; алгоритм активного предотвращения перегрузок сети; TCP-протокол.

### Введение

Пакетная коммутация позволяет повышать эффективность использования каналов, приводя, однако, к некоторому снижению надежности доставки. При перегрузке канала с пакетной коммутацией данные на входе канала не смогут поместиться во входной буфер и будут сброшены. Для обеспечения гарантированной доставки пакетов по каналам без гарантированной доставки были разработаны специальные протоколы, один из которых — протокол TCP. Со временем требования к передаче пакетных данных повысились: система передачи данных должна обеспечивать малое значение задержки и малую вариацию задержки. Эта задача решается с помощью регулирования скорости передачи и приема сообщений. Управление скоростью передачи осуществляется согласно алгоритму управления перегрузкой (*TCP congestion control algorithm*). В данной работе предлагается исследование сети TCP/IP с применением различных современных TCP-алгоритмов предотвращения перегрузок, таких как Reno, New Reno, Vegas.

### Постановка задачи

**Главная цель статьи** — исследование работы системы предотвращения перегрузок с использованием основных TCP-протоколов (Reno, New Reno, Vegas) в условиях изменения нагрузки трафика.

### Обзор основных алгоритмов предотвращения перегрузок

Перегрузка в сетях TCP/IP возникает в случае, когда количество передаваемых данных начинает приближаться к значению допустимой пропускной способности сети [1]. При этом ухудшаются основные показатели качества обслуживания, а также происходит рост числа потерянных пакетов и времени задержки. Основной задачей TCP-алгоритмов предотвращения перегрузок является поддержание значения потока передаваемых по сети данных ниже уровня, при котором пропускная способность сети начинает резко падать, ограничивая потоки входящего и исходящего трафика.

Для эффективного использования пропускной способности требуется выбрать именно тот противоперегрузочный алгоритм, который сможет обеспечить максимальное использование ресурсов сети TCP/IP при минимальных потерях.

### Протокол TCP-Reno

При использовании алгоритма TCP-Reno размер окна изменяется циклически. Он увеличивается при каждом цикле до потери пакета [2]. Когда происходит потеря пакета, TCP-Reno уменьшает размер окна до половины текущего размера. Это называется *аддитивным увеличением и мультипликативным уменьшением*.

TCP-Reno имеет два этапа изменения размера окна:

- 1) фаза медленного старта;
- 2) фаза избежания перегрузки.

Когда отправитель получает подтверждение доставки в момент времени  $t + t_A$ , с, текущее значение размера окна перегрузки  $cwnd(t)$  преобразуется в  $cwnd(t + t_A)$ , согласно формуле:

$$cwnd(t + t_A) = \begin{cases} cwnd(t) + 1 & \text{при } cwnd(t) < ssth(t), \\ cwnd(t) + \frac{1}{cwnd(t)} & \text{при } cwnd(t) \geq ssth(t), \end{cases} \quad (1)$$

где  $ssth(t)$  (пакет) — значение порога, при котором TCP переходит от фазы медленного старта в фазу избежания перегрузки.

Если потери пакетов происходят в результате тайм-аута, то значения  $cwnd(t)$  и  $ssth(t)$  обновляются следующим образом:

$$\begin{cases} cwnd(t) = 1, \\ ssth(t) = \frac{cwnd(t)}{2}. \end{cases} \quad (2)$$

Если же потери пакетов обнаружены согласно алгоритму быстрой передачи, то  $cwnd(t)$  и  $ssth(t)$  обновляются иначе:

$$\begin{cases} cwnd(t) = ssth(t), \\ ssth(t) = \frac{cwnd(t)}{2}. \end{cases} \quad (3)$$

### Протокол TCP-Vegas

Протокол TCP-Vegas використовує різницю між очікуваною і фактичною швидкістю потоків для оцінки пропускної спроможності мережі [3]. Основна ідея заключається в тому, що коли мережа не переконажена, то фактична швидкість потоку близька до очікуваної. В протилежному випадку фактична швидкість потоку менше очікуваної.

При допомозі різниці швидкостей протокол TCP-Vegas оцінює рівень затворів в мережі і відповідним чином оновлює розмір вікна. Істочник, використовуючи очікувану і фактичну швидкість потоку, вираховує оцінку відставання в череді за формулою [4]:

$$Diff = (Expected - Actual) BaseRTT, \quad (4)$$

де *Expected* і *Actual* — відповідно очікувана і фактична швидкість; *BaseRTT* — мінімальне значення RTT при першому підключенні.

Опишемо деталі алгоритму.

1. Істочник вираховує очікувану швидкість потоку за формулою

$$Expected = \frac{cwnd(t)}{BaseRTT}, \quad (5)$$

де *cwnd(t)* — поточний розмір вікна.

2. Істочник оцінює поточну швидкість потоку при допомозі фактичного RTT — часу проходження сигналу туди і назад:

$$Actual = \frac{cwnd(t)}{RTT}. \quad (6)$$

Істочник оновлює розмір вікон наступним чином:

$$cwnd(t) = \begin{cases} cwnd(t) + 1, & \text{якщо } Diff < \alpha, \\ cwnd(t) - 1, & \text{якщо } Diff > \beta, \\ cwnd(t) & \text{— в іншому випадку,} \end{cases} \quad (7)$$

де  $\alpha$  і  $\beta$  — деякі константи [5]; *t* — поточний момент часу.

Ця модифікація TCP потребує високого розрешення таймера відправителя.

### Розв'язання задачі

Дослідження мережі проводилося з використанням програмного комплексу NS-2. Її схема (рис. 1) складається з трьох (S1, S2, S3) FTP джерел повідомлення, які за допомогою двох маршрутизаторів передають інформацію на TCP-приймач (S4).

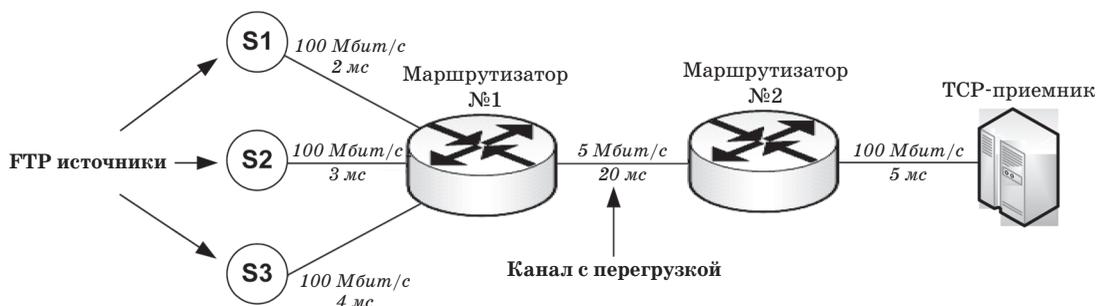


Рис. 1. Схема мережі для імітаційного моделювання

рутизаторів передають інформацію на TCP-приймач (S4).

Частина програмного коду на мові Otcl (NS-2), що відповідає за створення вузлів мережі, представляється наступним чином:

```
set ns [new Simulator]
set node_(s1) [$ns node]#FTP-джерело
set node_(s2) [$ns node]#FTP-джерело
set node_(s3) [$ns node]#FTP-джерело
set node_(r1) [$ns node]#Маршрутизатор №1
set node_(r2) [$ns node]#Маршрутизатор №2
set node_(s4) [$ns node]#TCP-приймач
```

Швидкість каналу між кінцевими вузлами повідомлення і маршрутизаторами становить 100 Мбіт/с.

Значення затримки для кожного кінцевого вузла різні:

$$S1 = 2 \text{ мс}, S2 = 3 \text{ мс}, S3 = 4 \text{ мс}.$$

Швидкість каналу між двома транзитними маршрутизаторами становить 5 Мбіт/с (канал з навантаженням), а затримка — 20 мс.

При дослідженні моделюваної мережі навантаження на маршрутизатор буде поступово збільшуватися. Першим почне працювати джерело S1. Потім через 30 с — S2. На 60-й секунді увімкниться S3. Програмний код, що відповідає за встановлення параметрів і з'єднання між вузлами, представляється в наступному вигляді

```
$ns duplex-link $node_(s1) $node_(r1) 100Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 100Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 5Mb 20ms ***#Місце вставки AQM-алгоритму
$ns duplex-link $node_(s3) $node_(r1) 100Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 100Mb 5ms DropTail
```

```
$ns duplex-link-op $node_(s1) $node_(r1) orient right-down
$ns duplex-link-op $node_(s2) $node_(r1) orient right-up
$ns duplex-link-op $node_(r1) $node_(r2) orient right
```

```
$ns duplex-link-op $node_(r1) $node_(r2) queue
Pos 0
$ns duplex-link-op $node_(r2) $node_(r1) queue
Pos 0
$ns duplex-link-op $node_(s3) $node_(r1) orient
right-down
$ns duplex-link-op $node_(s4) $node_(r2) orient
left-up
```

```
set tcp1 [$ns create-connection TCP/** $node_(s1)
TCPSink $node_(s4) 0]
#Вместо знака «**» прописываем нужный алго-
ритм
set tcp2 [$ns create-connection TCP/** $node_(s2)
TCPSink $node_(s4) 1]
#Вместо знака «**» прописываем нужный алго-
ритм
set tcp3 [$ns create-connection TCP/** $node_(s3)
TCPSink $node_(s4) 2]
# Вместо знака «**» прописываем нужный алго-
ритм
```

```
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
set ftp3 [$tcp3 attach-source FTP]
```

Продолжительность процесса моделирования составляет 100 с. Приведем код, отвечающий за включение источников сообщения в сеть TCP/IP.

```
$ns at 0.0 "$ftp1 start"
$ns at 30.0 "$ftp2 start"
$ns at 60.0 "$ftp3 start"
$ns at 100 "finish"
```

Детальное описание основных алгоритмов (RED, REM) активного управления очередью приведено в [6].

После выполнения программы получим изображенные на рис. 2–5 графики зависимости длины очереди, пакетов, от времени работы сети для каждого алгоритма.

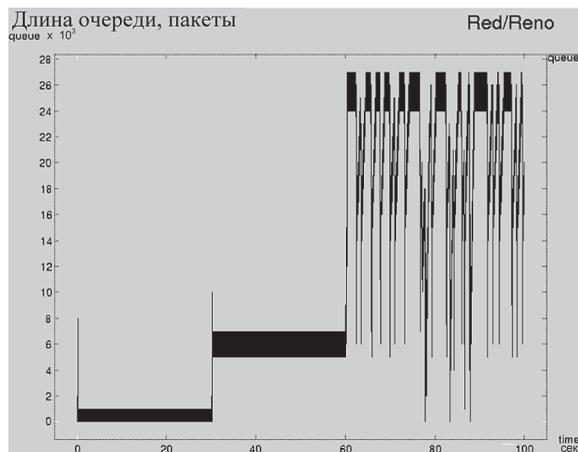


Рис. 2. Длина очереди при совместном использовании RED- и Reno-алгоритмов

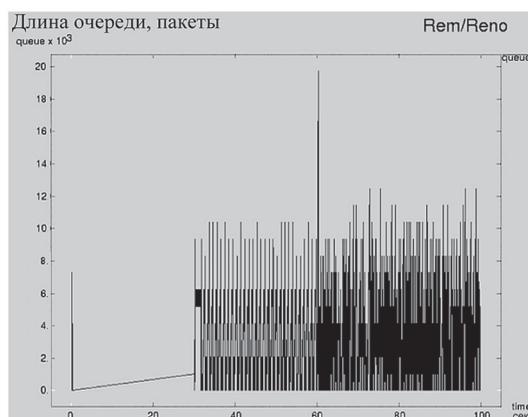


Рис. 3. Длина очереди при совместном использовании REM- и Reno-алгоритмов

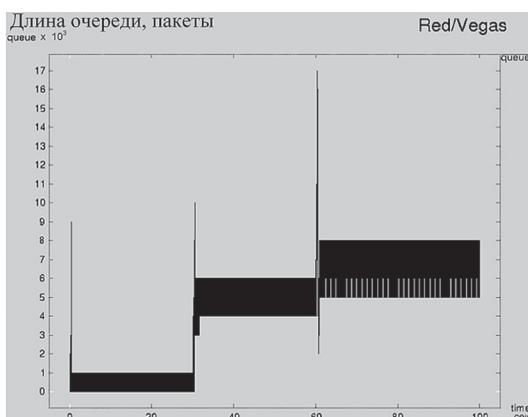


Рис. 4. Длина очереди при совместном использовании RED- и Vegas-алгоритмов

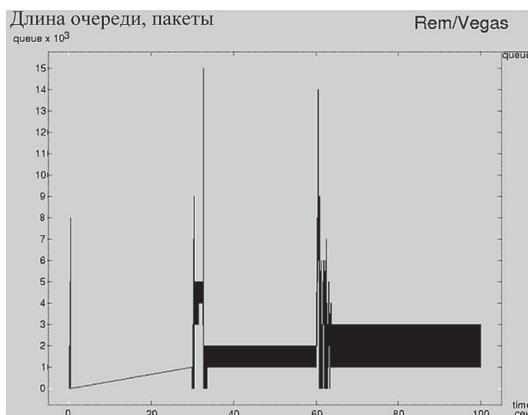


Рис. 5. Длина очереди при совместном использовании REM- и Vegas-алгоритмов

### Вывод

Проанализировав приведенные и полученные данные, приходим к выводу о том, что метод предотвращения перегрузки Vegas наиболее подходит для использования в паре с REM-алгоритмом, поскольку благодаря такому сочетанию перегруженный канал связи имеет наиболее стабильное и наиболее низкое значение очереди при постепенном увеличении нагрузки сети по сравнению со всеми каналами связи, использующими другие алгоритмы и методы.

## Литература

1. Коваленко, А. А. Проблемы производительности протокола TCP в гетерогенных сетях и методы ее улучшения / А. А. Коваленко, Ю. Ю. Завизиступ, С. А. Партыка // Вестник ХНТУ.— 2005.— № 1(21).— С. 305–311.

2. *Analysis and comparison of TCP Reno and Vegas [Электронный ресурс]* / [Jeonghoon Mo, Richard J. La, Venkat Anantharam, Jean Walrand] // INFOCOM–1999.— Режим доступа:

<http://www.eecs.berkeley.edu/~ananth/1999-2001/Richard/MoLaInfocom1999.pdf>

3. Семенов, Ю. А. Протокол TCP [Электронный ресурс].— Режим доступа:

[http://book.itep.ru/4/44/tcp\\_443.htm](http://book.itep.ru/4/44/tcp_443.htm)

4. Коваленко, А. А. Динамічне керування параметрами протоколу TCP Vegas / А. А. Коваленко // Системи озброєння і військова техніка.— 2007.— №2(10).— С. 81–86.

5. Kurata, K. *Fairness Comparisons Between TCP Reno and TCP Vegas for Future Deployment of TCP Vegas [Электронный ресурс]* / Kenji Kurata, Go Hasegawa, Masayuki Murata.— Режим доступа: [http://www.isoc.org/inet2000cdproceedings/2d/2d\\_2.htm](http://www.isoc.org/inet2000cdproceedings/2d/2d_2.htm)

6. Гостев, В. И. Исследование сети TCP/IP с применением основных алгоритмов активного управления очередью / В. И. Гостев, Т. П. Довженко, А. С. Артющик // Системи управління, навігації та зв'язку.— 2014.— № 2(30).— С. 87–91.

Т. П. Довженко

## ДОСЛІДЖЕННЯ ЧЕРГИ МАРШРУТИЗАТОРА В МЕРЕЖІ TCP/IP

## З ВИКОРИСТАННЯМ ОСНОВНИХ АЛГОРИТМІВ АКТИВНОГО УПРАВЛІННЯ ЧЕРГОЮ ТА ЗАПОБІГАННЯ ПЕРЕВАНТАЖЕННЯМ

Розглянуто основні різновиди TCP-протоколів запобігання перевантаженням мережі (TCP-Reno, TCP-Vegas) і проведено дослідження мережі TCP/IP із застосуванням зазначених протоколів.

**Ключові слова:** TCP-Reno; REM; RED; TCP-Vegas; алгоритм запобігання перевантаженням мережі; TCP-протокол.

T. Dovzhenko

## RESEARCH STAGE ROUTERS IN THE NETWORK TCP/IP USING BASIC TCP-ALGORITHM OF ACTIVE QUEUE MANAGEMENT AND CONGESTION AVOIDANCE

Consideration of the basic TCP-protocols to prevent network congestion (TCP-Reno, TCP-Vegas) and a study TCP/IP network using these protocols.

**Keywords:** TCP-Reno; REM; RED; TCP-Vegas; algorithm actively preventing network congestion; TCP-protocol.

УДК 656.8.001

Л. О. ЯЩУК, доктор техн. наук, професор,  
Одеська національна академія зв'язку ім. О. С. Попова

## Проблеми імплементації Директиви Європейського Парламенту та Ради 97/67/ЄС про спільні правила розвитку внутрішнього ринку поштових послуг Співтовариства та покращення якості обслуговування

Розглянуто питання впровадження в Україні свростандартів з якості надання універсальних поштових послуг і системи постійного моніторингу та контролю за їх дотриманням.

**Ключові слова:** Директива 97/67/ЄС; імплементація Директиви 97/67/ЄС; універсальна поштова послуга; собівартість універсальної поштової послуги; стандарти якості універсальної поштової послуги.

## Вступ

Кабінет Міністрів України своїм Розпорядженням від 18.03.2015 №222-р ухвалив розроблений Міністерством інфраструктури План імплементації Директиви 97/67/ЄС Європейського Парламенту та Ради Європи від 15 грудня 1997 р. про спільні правила розвитку внутрішнього ринку поштових послуг Співтовариства та покращення якості обслуговування (далі — Директива 97/67/ЄС) [1].

Метою імплементації Директиви 97/67/ЄС визначено приведення чинного законодавства України у сфері надання поштових послуг до норм права

ЄС та виконання зобов'язань відповідно до Угоди про асоціацію, зокрема щодо:

- фінансування універсальних поштових послуг (УПП);
- дотримання тарифних принципів та прозорості розрахунків для надання УПП;
- встановлення стандартів якості для надання УПП і системи забезпечення постійного моніторингу та контролю за дотриманням стандартів якості надання УПП.

Утім установлений зазначеним планом нереальний строк імплементації Директиви 97/67/ЄС —