

УДК 004.738.5.057.4

Максимов В. В., к.т.н., Чмихун С. О.

ДОСЛІДЖЕННЯ АЛГОРИТМУ БОРОТЬБИ З ПЕРЕВАНТАЖЕННЯМИ TCP VENO

Махунов В. В., Чмихун С. О. Research of the algorithm of controlling congestion TCP Veno. This work is devoted to the analysis of combating TCP Veno congestion algorithm in networks that use TCP as the primary data transmission protocol, and its further improvement. The phases of algorithm are considered. It is shown that the TCP Veno algorithm refines the additive increase and multiplicative decrease algorithms, which are implemented in TCP Reno. Multiplicative reduction phase provides the largest contribution to the prevention of congestion. An analysis shows that there is a possibility to improve TCP Veno algorithm by changing the standard value of the coefficient $\gamma = 4/5$ to a larger value in the range $[4/5; \sim 1)$. The formula is provided for calculating the throughput of the network in which TCP Veno protocol functions. It is shown that the γ coefficient is inversely proportional to network throughput and its increase may also result in network throughput increase. The service simulating test in NS-2 simulator was performed. The network model with two and six nodes demonstrates that the usage of TCP Veno algorithm $\gamma = 9/10$ coefficient allows to achieve improved network throughput (in average by (7.6 ... 26)%) in comparison to a standard $\gamma = 4/5$ coefficient usage. It is also shown that the TCP Veno remains 1,328 times longer in the large window range with $\gamma = 9/10$ coefficient, than with the standard value of $\gamma = 4/5$.

Keywords: TCP/IP network, TCP Veno algorithm, throughput, overload, simulator NS-2

Максимов В. В., Чмихун С. О. Дослідження алгоритму боротьби з перевантаженнями TCP VENO. Дана робота присвячена аналізу алгоритму боротьби з перевантаженнями TCP Veno в мережах, що використовують протокол TCP в якості основного протоколу передачі даних, і подальшого його вдосконалення. Розглянуто фази роботи алгоритму. Показано, що найбільший внесок в запобіганні перевантаження надає саме фаза мультиплікативного зменшення. Теоретично та практично доведено можливість збільшення пропускної здатності мережі, шляхом зміни параметрів алгоритму.

Ключові слова: TCP/IP мережа, TCP алгоритм Veno, пропускна здатність, перевантаження, симулятор NS-2

Максимов В. В., Чмихун С. А. Исследование алгоритма борьбы с перегрузками TCP VENO. Данная работа посвящена анализу алгоритма борьбы с перегрузками TCP Veno в сетях, использующих протокол TCP в качестве основного протокола передачи данных, и дальнейшего его совершенствования. Рассмотрены фазы работы алгоритма. Показано, что наибольший вклад в предотвращении перегрузки предоставляет именно фаза мультипликативного уменьшения. Теоретически и практически доказана возможность увеличения пропускной способности сети, путем изменения параметров алгоритма.

Ключевые слова: TCP/IP сеть, TCP алгоритм Veno, пропускная способность, перегрузка, симулятор NS-2

Вступ. Із поширенням використання пакетних мереж та, як наслідок, зростанням об'єму трафіку в мережі Інтернет, збільшується навантаження на мережеве обладнання та вірогідність виникнення перевантаження мережі. Для запобігання останньому застосовуються алгоритми боротьби з перевантаженнями, які допомагають протоколам передачі даних управляти мережею та запобігати виникненню колапсів.

На сьогоднішній день широко використовуються алгоритми, що базуються на протоколі TCP [1...8]. Серед них найбільш універсальним є алгоритм TCP Veno, який здатен ефективно працювати як в провідних мережах, так і в безпроводових. В багатьох роботах, наприклад, [9...11] розглядається використання даного алгоритму при стандартних значеннях його параметрів. У зв'язку зі зростанням об'єму трафіку, актуальним стає питання можливості збільшення пропускної здатності мережі шляхом оптимізації параметрів алгоритму TCP Veno.

Метою роботи є аналіз можливості збільшення пропускної здатності мережі передачі даних шляхом оптимізації параметрів алгоритму управління перевантаженнями TCP Veno.

Аналіз алгоритму TCP Veno. TCP Veno намагається виділити втрати, що не пов'язані з перевантаженням, для того щоб не застосовувати механізм боротьби з перевантаженням там, де це не потрібно. Даний алгоритм використовує механізм, аналогічний тому, що використовується в TCP Vegas, але з метою визначення характеру втрат пакетів. Зокрема, якщо втрата пакетів виявляється в той час як оцінка показує, що мережа не перевантажена, встановлюється, що втрата є випадковою [5]. Таким чином принцип дії даного алгоритму поділяється на три фази:

1) Фаза повільного старту. У пусковій фазі TCP Veno встановлює розмір вікна $swnd=1$ і посилає перший пакет. Після цього, кожного разу, коли прийняття пакету підтверджено, розмір вікна збільшується на одиницю. Таким чином, $swnd$ збільшується з 1 до 2 після першого RTT, від 2 до 4 після другого RTT (тому що прийшло два підтвердження) і так далі. Це призводить до експоненціального зростання швидкості відправки в часі, до тих пір, поки не буде втрачено пакет.

2) Фаза адитивного збільшення. Вводиться додатковий параметр – $ssthresh$ – поріг повільного старту. Коли $swnd < ssthresh$, алгоритм повільного старту використовується для регулювання $swnd$. Але, як тільки $swnd > ssthresh$, швидкість збільшення розміру вікна зменшується для запобігання перевантаження. Зокрема, $swnd$ буде збільшено на одиницю після кожного RTT, а не прийому ACK. По суті, $swnd$ встановлюється рівним $swnd+1/swnd$ після кожного ACK для досягнення ефекту лінійного зростання. Таким чином, алгоритм набуває вигляду:

if ($N < \beta$) – доступна смуга не повністю використана
 set $swnd = swnd + 1/swnd$ when each new ACK is received
else if ($N \geq \beta$) – доступна смуга повністю використана
 set $swnd = swnd + 1/swnd$ when every other new ACK is received

де N – кількість пакетів в буфері; β – верхня межа кількості пакетів в черзі [5].

3) Фаза мультиплікативного зменшення. Дана фаза починається при втратах пакетів.

В TCP Veno існує два способи визначення втрат пакетів. *Перший* спосіб реалізований аналогічно до TCP Reno. Тобто пакет може бути оголошено як втрачений, якщо за відведений час не прийшло підтвердження від приймальної сторони. В цьому випадку, ініціюється алгоритм повільного старту з пороговим значенням $ssthresh=swnd/2$ та $swnd=1$. Це має ефект раптового зменшення швидкості відправки на велику величину. Тобто, закінчення таймеру інтерпретується як наявність значного перевантаження.

Другий спосіб – швидка повторна передача. Кожного разу, коли приймач отримує пакет, що йде не по порядку, він ретранслює останній ACK відправнику. Коли відправник отримує підряд три дублюючих ACK, це означає, що пакет втрачено, навіть якщо таймер ще не закінчився. Алгоритм швидкої повторної передачі доповнюється алгоритмом швидкого відновлення, для того щоб зменшити перевантаження. В TCP Veno дана фаза має наступний вигляд:

if ($N < \beta$) – випадкова втрата, що сталася через бітові помилки
 ssthresh = $swnd * (4/5)$;
else *ssthresh* = $swnd/2$; – втрата, що сталася через перевантаження

Іншими словами, якщо з'єднання не знаходиться в перевантаженому стані ($N < \beta$), TCP Veno вважає, що втрата носить випадковий характер. Відповідно, зменшується $ssthresh$ та як наслідок $swnd$, на більш малу величину γ . В якості стандартної величини в [5] наведено $\gamma = 4/5$ (в практичних реалізаціях для швидкого бітового зсуву в обчисленнях використовувалось $\gamma = 1 - ((1/8)+(1/16))$). В [6] вказується, що $\gamma \sim 1$, бажано більше $3/4$, але, досліджень з $\gamma > 4/5$ в [5, 6] не наведено.

Видно, що алгоритм TCP Veno уточнює алгоритми адитивного збільшення та мультиплікативного зменшення, які реалізовані в TCP Reno. Найбільший внесок в запобіганні перевантаження надає саме фаза мультиплікативного зменшення. Її аналіз

показує, що існує можливість вдосконалення алгоритму TCP Veno, шляхом зміни стандартного значення коефіцієнту $\gamma=4/5$ на більше значення в інтервалі $[4/5; \sim 1)$, наприклад $\gamma=9/10$.

Пропускна здатність мережі, в якій функціонує алгоритм TCP Reno, описується формулою [7]:

$$B(p) \approx \frac{1}{RTT \sqrt{\frac{2bp}{3}} + T_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p(1+32p^2)}, \quad (1)$$

де RTT – поточне значення часу, витраченого на проходження зазначеного пакету від клієнта до сервера і назад; b – кількість пакетів, які підтверджені отриманим АСК; p – ймовірність того, що пакет загублений (як випадково, так і пов'язано з перевантаженням); T_0 – період часу, після якого повторно передаються непідтверджені пакети.

Для мережі, в якій функціонує алгоритм TCP Veno, пропускну здатність можна записати як:

$$B(p) \approx \frac{1}{RTT \sqrt{\frac{2b(1-\gamma)p}{1+\gamma}} + T_0 \min\left(1, 3\sqrt{\frac{b(1-\gamma^2)p}{2}}\right) p(1+32p^2)}. \quad (2)$$

З (2) видно, що коефіцієнт γ є обернено пропорційним до пропускну здатності мережі і його збільшення може також призвести до збільшення пропускну здатності мережі.

Для підтвердження висновків аналізу, було проведено імітаційне моделювання алгоритму TCP Veno в мережевому симуляторі NS-2 [10, 11]. Дослідження проводились на двох різних топологіях мережі, а саме: з двома і шістьма вузлами [8].

Мережа з двома вузлами. Для реалізації моделювання в NS-2 побудована мережа, що складається з двох вузлів з пропускну здатністю 100 Мбіт/с, затримкою 10 мс і розміром пакетного буфера 50 Кб, розмір пакету 1 Кб. Пакети передаються від вузла Src 1 до вузла Dst 1. Топологія мережі в симуляторі NS-2 показана на Рис. 1.

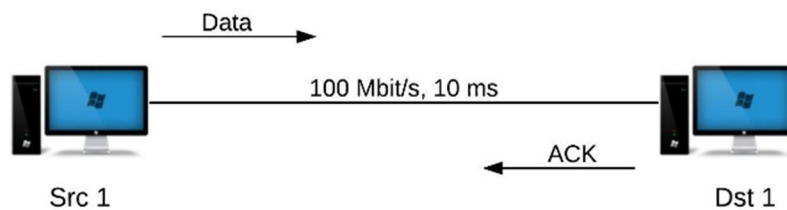


Рис. 1. Топологія мережі в симуляторі NS-2

У процесі моделювання вносились зміни до вихідного коду протоколу TCP Veno, а саме: змінювався так званий «коефіцієнт TCP Veno» (γ) – коефіцієнт зміни розміру вікна перевантаження при детектуванні випадкових втрат у фазі мультиплікативного зменшення вікна.

Досліджено наступні значення коефіцієнту γ : $4/5$ – стандартне значення, яке використовується протоколом TCP Veno; $9/10$ – запропоноване значення коефіцієнту з допустимого інтервалу $[1/2; 1]$. Отримані графічні залежності для алгоритму TCP Veno при наявності 5% випадкових втрат, пропускну здатності каналу 10 Мбіт/с та коефіцієнту $\gamma=(4/5; 9/10)$ показані на Рис. 2, 3.

Аналіз трейс-файлів, на основі яких побудовані графіки, дав наступні чисельні результати:

1) при $\gamma=4/5$ та часі моделювання $t = 8$ с :

– сумарна пропускна здатність склала 2,21 Мбіт/с;

- середнє значення розміру вікна перевантаження склало 6,537 пакетів/с;
- 2) при $\gamma=9/10$ та часі моделювання $t = 8$ с :
 - сумарна пропускна здатність склала 3,01 Мбіт/с;
 - середнє значення розміру вікна перевантаження склало 8,683 пакетів/с.

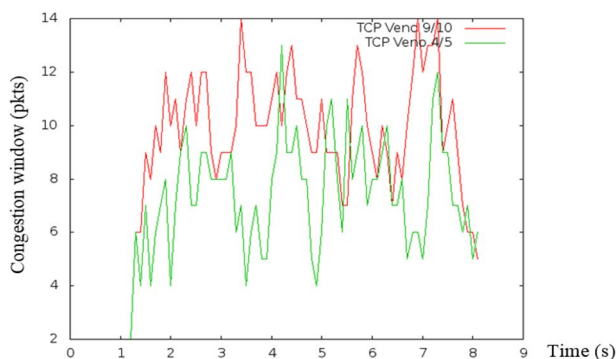


Рис. 2. Еволюція вікна перевантаження алгоритму TCP Veno за наявності 5% випадкових втрат при $\gamma=(4/5; 9/10)$

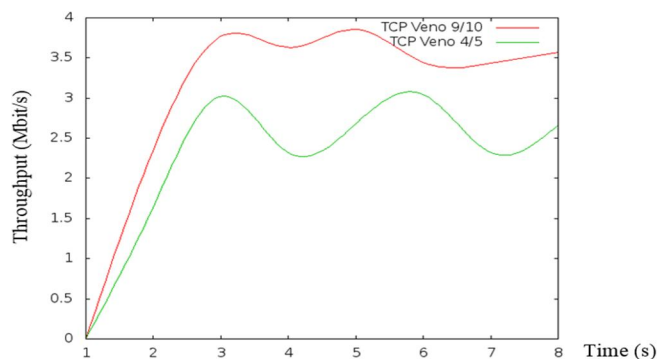


Рис. 3. Еволюція пропускної здатності алгоритму TCP Veno за наявності 5% випадкових втрат при $\gamma=(4/5; 9/10)$

Отримані результати показують, що TCP Veno при $\gamma=9/10$ залишається в області більшого вікна в 1,328 рази довше ніж при стандартному значенні $\gamma=4/5$. Як результат, збільшення вікна перевантаження призводить до збільшення пропускної здатності в 1,361 рази.

В Табл. 1 наведено порівняння результатів моделювання пропускної здатності мережі за формулою (2) в NS-2 з результатами її математичного моделювання в Mathcad.

Порівняння результатів моделювання

Табл. 1

| Час моделювання, с | RTT, с | В, Мбіт/с (отримане в Mathcad) | В, Мбіт/с (отримане в NS-2) | Відносне відхилення від розрахункового значення, % |
|--------------------|--------|--------------------------------|-----------------------------|--|
| 1 | | | - | - |
| 2 | | | 10,68 | 3,46 |
| 3 | | | 10,92 | 0,64 |
| 4 | | | 9,94 | 1,21 |
| 5 | | | 10,59 | 2,64 |
| 6 | | | 11,71 | 2,22 |
| 7 | | | 9,32 | -0,54 |
| 8 | | | 11,49 | 0,35 |

Видно, що розбіжність результатів мала (середнє відносне відхилення складає 1,43%), що свідчить про можливість використання даної формули для розрахунку пропускної здатності мережі. Також слід відзначити, що алгоритм TCP Veno з коефіцієнтом $\gamma=9/10$ робить збільшення вікна більш повільним (в 1,328 рази порівняно з $\gamma=4/5$) в критичній області, тим самим змушує TCP залишатися довше в області "оптимальної" швидкості передачі.

Моделювання алгоритму в мережі з шістьма вузлами. Топологія мережі в симуляторі NS-2 зображена на Рис. 4. З'єднання вузлів відправки пакетів Src 1, Src 2 і вузлів їх отримки Dst1, Dst 2 з Router 1 і Router 2 змодельовано з пропускною здатністю 10Мбіт/с, затримка 1 мс, розмір пакетного буфера 50 Кб. Для з'єднання між Router 1 і Router 2 розмір пакетного буфера встановлено рівним 20 Кб, розмір пакету 1 Кб. Його пропускна спроможність і затримки змінювались під час експериментів.

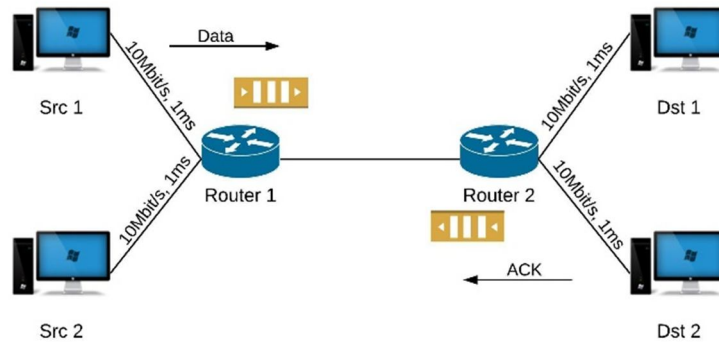


Рис. 4. Топологія експерименту в симуляторі NS-2

Проведено моделювання двох сценаріїв:

- 1) Пропускна здатність з'єднання Router 1 і Router 2 встановлена 2 Мбіт/с, величина затримки у вузькому місці приймає значення 40 мс та 160 мс відповідно.
- 2) Пропускна здатність з'єднання Router 1 і Router 2 залишилася рівною 2 Мбіт/с, величина затримки у вузькому місці становить 40 мс та додатково вводиться фоновий трафік інтенсивністю 750 Кбіт/с у вузькому місці. Для імітації фонового трафіку створено два UDP з'єднання з розподілом Парето через вузьке місце мережі. Розмір пакета UDP – 512 байт.

Отримані графічні залежності для алгоритму TCP Veno при наявності 5% випадкових втрат, величині затримки 40 мс і 160 мс у вузькому місці та значенні коефіцієнту γ – 4/5 та 9/10 для першого сценарію показані на Рис. 5...8.

Результати аналізу трейс-файлів, на основі яких побудовані графіки, наведені в Табл. 2.

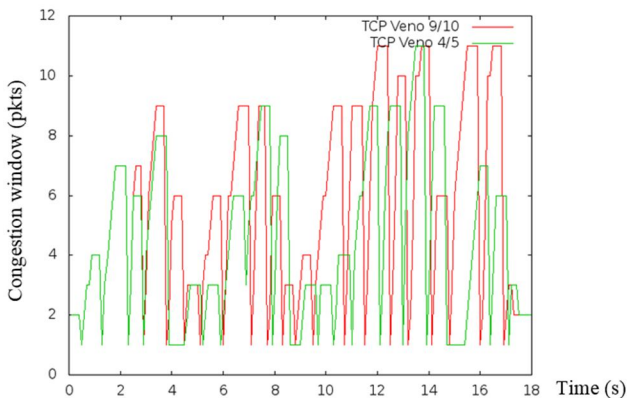


Рис. 5. Еволюція вікна перевантаження алгоритму TCP Veno за наявності 5% випадкових втрат, $\gamma=(4/5; 9/10)$ та Delay=40 мс

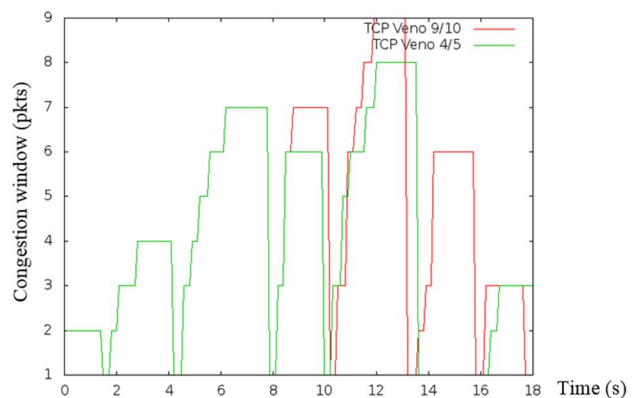


Рис. 6. Еволюція вікна перевантаження алгоритму TCP Veno за наявності 5% випадкових втрат, $\gamma=(4/5; 9/10)$ та Delay=160 мс

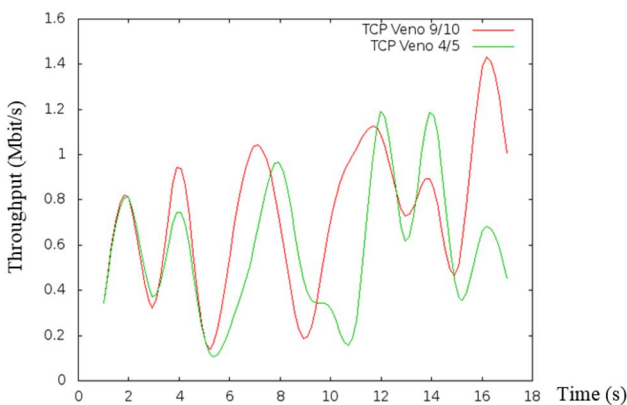


Рис. 7. Еволюція пропускної здатності алгоритму TCP Veno за наявності 5% випадкових втрат, $\gamma=(4/5; 9/10)$ та Delay=40 мс

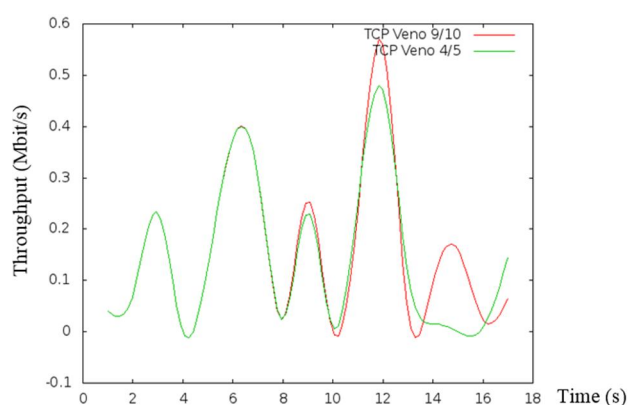


Рис. 8. Еволюція пропускної здатності алгоритму TCP Veno за наявності 5% випадкових втрат, $\gamma=(4/5; 9/10)$ та Delay=160 мс

Результати аналізу трейс-файлів

Табл. 2

| Вихідні данні | Середнє значення розміру вікна перевантаження, пакетів/с | Сумарна пропускна здатність, Мбіт/с |
|-------------------------------------|--|-------------------------------------|
| $\gamma=4/5, t=18$ с, Delay=40 мс | 4,550 | 0,574 |
| $\gamma=9/10, t=18$ с, Delay=40 мс | 5,611 | 0,728 |
| $\gamma=4/5, t=18$ с, Delay=160 мс | 3,967 | 0,143 |
| $\gamma=9/10, t=18$ с, Delay=160 мс | 4,517 | 0,154 |

Виходячи з наведених вище результатів видно, що TCP Veno при $\gamma=9/10$ залишається в області більшого вікна в 1,233 рази довше при величині затримки 40 мс та в 1,138 рази при 160 мс ніж при стандартному значенні $\gamma=4/5$. Збільшення вікна перевантаження призводить до збільшення пропускної здатності в 1,268 рази при 40 мс та в 1,077 рази при 160 мс.

Отримані графічні залежності для алгоритму TCP Veno для другого сценарію при наявності фонового трафіку 750 Кбіт/с, величині затримки 40 мс та значенні коефіцієнту γ - 4/5 та 9/10 показані на Рис. 9, 10. Результати аналізу трейс-файлів, на основі яких побудовані графіки, наведені в Табл. 3.

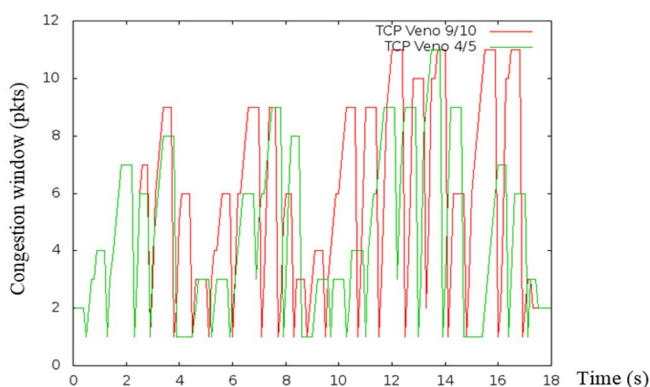


Рис. 9. Еволюція вікна перевантаження алгоритму TCP Veno при $\gamma=(4/5; 9/10)$. Рівень фонового трафіку 750 Кбіт/с

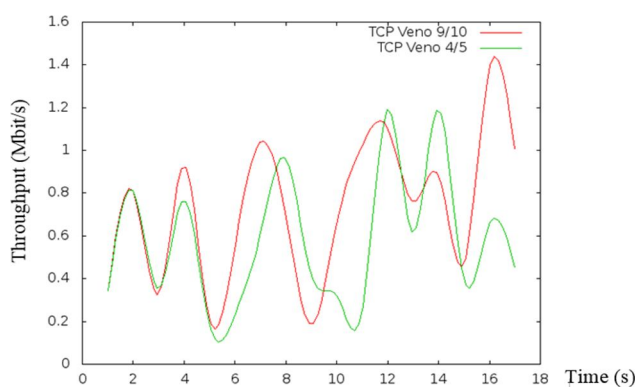


Рис. 10. Еволюція пропускної здатності алгоритму TCP Veno при $\gamma=(4/5; 9/10)$. Рівень фонового трафіку 750 Кбіт/с

Результати аналізу трейс-файлів

Табл. 3

| Вихідні данні | Середнє значення розміру вікна перевантаження, пакетів/с | Сумарна пропускна здатність, Мбіт/с |
|---------------------------------------|--|-------------------------------------|
| $\gamma=4/5, t=18$ с, UDP=750 Кбіт/с | 4,583 | 0,574 |
| $\gamma=9/10, t=18$ с, UDP=750 Кбіт/с | 5,656 | 0,729 |

З Табл. 3 видно, що наявність фонового трафіку не погіршила ефективності алгоритму. Використання TCP Veno з $\gamma=9/10$ дозволяє алгоритму залишатися в області більшого вікна в 1,234 рази довше при рівні фонового трафіку 750 Кбіт/с ніж при стандартному значенні $\gamma=4/5$. Крім того, збільшення вікна перевантаження призводить до збільшення пропускної здатності в 1,270 рази.

Висновки

- 1) Існує можливість вдосконалення алгоритму TCP Veno, шляхом зміни стандартного значення коефіцієнту $\gamma=4/5$ на більше значення в інтервалі $[4/5; \sim 1]$, наприклад $\gamma=9/10$.
- 2) Використання алгоритму TCP Veno з коефіцієнтом $\gamma=9/10$ дозволяє домогтися поліпшення пропускної здатності мережі у середньому на (7,6... 26)% і залишатися в області великого вікна в 1,328 рази довше ніж при стандартному значенні $\gamma = 4/5$.

Плануються подальші теоретичні дослідження поліпшення роботи алгоритму TCP Veno.

Література

1. Jacobson V. Congestion Avoidance and Control / Van Jacobson // SIGCOMM '88, ACM. – Aug, 1988.
2. Jacobson V. Berkeley TCP evolution from 4.3-tahoe to 4.3-reno / V. Jacobson // Proc. of the 18th Internet Engineering Task Force /– Vancouver. – August, 1990.
3. Brakmo L. TCP Vegas: New techniques for congestion detection and avoidance / L. Brakmo, S. O'Malley, L. Peterson // SIGCOMM 94, ACM. – 1994.
4. Floyd S. NewReno Modification to Fast Recovery Algorithm / S. Floyd, T. Henderson, A. Gurtov , RFC 3782.
5. Fu C. P. TCP Veno: End-to-End Congestion Control Over Heterogeneous Networks [Електронний ресурс] / C. P. Fu. // Ph.D. dissertation, The Chinese Univ. Hong Kong. – Hong Kong, 2001 // – Режим доступу: : <http://ccc.cs.lakeheadu.ca/cs5400/wurefl.pdf>.
6. Fu C. P. TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks [Електронний ресурс] / Fu C. P., Chen YoChuan (speaker) // IEEE journal on selected areas in communications. – February 200. –V. 21, No. 2 // – Режим доступу: <http://mnet.cs.nthu.edu.tw/paper/kemel/030710.pdf>.
7. Копійка О. В. Архітектура мережі в сучасних дата-центрах / О. В. Копійка // Наукові записки Українського науково-дослідного інституту зв'язку. – 2014. – № 2(30). – С.34-41.
8. Padhye J. (1998). Modeling TCP Throughput: A Simple Model and its Empirical Validation [Електронний ресурс] / J. Padhye, V. Firoiu, D. Towsley and J.Kurose // Proc. of ACM SIGCOMM'98. Vancouver, Canada. – 1998. – PP. 303-314 // – Режим доступу: <http://www.arl.wustl.edu/~gorinsky/spring2005/cse573s/TCP-throughput.pdf>.
9. Галкин А. М. Пакет моделирования NS-2 / А. М. Галкин, Е. А. Кучерявый, Д. А. Молчанов ; учебное пособие. – Санкт-Петербург, СПбГТ, 2007.
10. NS-2 – User Information. Wikipedia [Електронний ресурс] // – Режим доступу: http://nslam.isi.edu/nslam/index.php/User_Information
11. Сиропятов О. А. Проблема моделювання трафіку у мережах доступу до недовірених систем / О. А. Сиропятов, Н. Ф. Казакова // Інформаційна безпека. – 2013. – № 1(9). – С.185-189.

Автори статті

Максимов Володимир Васильович – кандидат технічних наук, доцент кафедри телекомунікаційних систем, Національний технічний університет України «Київський політехнічний інститут», м. Київ. Тел. +380 (68) 810 40 47. E-mail: maksimov46@ukr.net

Чмихун Сергій Олександрович – магістрант, кафедра телекомунікаційних систем, Національний технічний університет України «Київський політехнічний інститут», м. Київ. E-mail: serega4sa@yandex.ru

Дата надходження в редакцію: 04.08.2015 р.

Рецензент: д.т.н., проф. Л. О. Уривський