

УДК 681.518

В.В. Трикоз, С.С. Батир

ДВНЗ «Донецький національний технічний університет», м. Донецьк
кафедра автоматики та телекомунікацій
E-mail: trikoz.viktorya@yandex.ru

ДОСЛІДЖЕННЯ РОБОТИ ПРОТИПЕРЕВАНТАЖНИХ АЛГОРИТМІВ ПРОТОКОЛУ TCP ПРИ ПЕРЕВАНТАЖЕННЯХ В КАНАЛАХ ЗВ'ЯЗКУ

Виконані дослідження, моделювання та аналіз роботи алгоритмів для боротьби з перевантаженням протоколів транспортного рівня стека TCP/IP. Для дослідження обрані алгоритми TCP Reno та TCP Vegas. Розглядані принципи роботи обраних алгоритмів. Для моделювання використана гідродинамічна модель потоку, за допомогою якою були враховані особливості роботи обраних алгоритмів. Отримані результати моделювання дозволили оцінити поведінку алгоритмів при перевантаженнях та ефективність використання пропускної спроможності каналу зв'язку.

Ключові слова: перевантаження, алгоритм управління перевантаженням, TCP Reno, TCP Vegas, гідродинамічна модель.

Загальна постановка проблеми

Проблема перевантаження в мережах TCP/IP виникає у разі, коли кількість переданих даних починає наближатися до значення допустимої пропускної спроможності мережі. При цьому погіршуються основні показники якості обслуговування. Ці погіршення можуть виражатися в збільшенні числа втрачених пакетів і часу затримок. Управління перевантаженнями є актуальним завданням, оскільки кількість кінцевих користувачів глобальної мережі, а, отже, об'єми переданих даних, постійно збільшуються.

Зростає також доля мультимедійного трафіку реального часу, вплив перевантажень на який особливо критично. Завдання механізмів управління перевантаженнями полягає в тому, щоб підтримувати кількість даних, переданих по мережі, нижче за рівень, при якому пропускна спроможність мережі починає різко падати, обмежуючи потоки вхідного і вихідного трафіку.

Аналіз існуючих досліджень та публікацій

Перевантаження в каналах зв'язку мереж TCP/IP є досить давньою, але дуже актуальною навіть сьогодні, проблемою. Їй присвячена велика кількість робіт [1,2,7,8,9].

Дослідження роботи протиперевантажних алгоритмів транспортного рівня стеку TCP/IP описано в роботах [3-5,10].

Оскільки для спостереження поведінки алгоритмів для боротьби з перевантаженням необхідно використовувати таку математичну модель, яка буде враховувати всі особливості роботи алгоритму, то на даний час дуже розповсюдженою є гідродинамічна модель потоку (fluid flow model), яка наведена в роботі [6].

Постановка задач дослідження

Для дослідження роботи алгоритмів для при для боротьби з перевантаженнями необхідно вирішити наступні задачі:

1. Проаналізувати основні алгоритми, які використовуються в протоколі TCP для боротьби з перевантаженнями.

© Трикоз В.В., Батир С.С., 2013

2. Розробити реалізацію математичної гідродинамічної моделі для обраних протиперевантажних алгоритмів та провести моделювання роботи цих алгоритмів при різних вхідних параметрах.

3. На основі отриманих результатів провести аналіз та порівняти роботу різних алгоритмів у однакових умовах.

Вирішення задач та результати дослідження

На даний момент існує цілий ряд реалізацій методів управління перевантаженнями в протоколі TCP, що розрізняються в застосовності до вирішень різних завдань. Наприклад, TCP Vegas дозволяє враховувати стан каналу зв'язки, TCP Westwood враховує високий рівень втрат в каналах зв'язку, TCP Hamilton використовується для локальних мереж з високою пропускнуною спроможністю. Найбільш розповсюдженими реалізаціями протоколу є TCP Reno та TCP Vegas, які будемо використовувати для подальшого дослідження роботи протиперевантажних алгоритмів.

Проведемо аналіз роботи обраних алгоритмів.

В TCP Reno у нормальній ситуації розмір вікна змінюється циклічно. Розмір вікна збільшується під час кожного циклу до втрати пакету. Коли відбудеться втрата пакету, TCP Reno зменшує розмір вікна до половини поточного розміру. Це називається адитивне збільшення та мультиплікативне зменшення.

TCP Reno має два етапи зміни розміру вікна:

- 1) фаза повільного старту
- 2) фаза уникнення перевантаження.

Коли відправник отримує підтвердження доставки в момент часу $t + t_A$ [с] поточне значення розмір вікна перевантаження $cwnd(t)$ перетворюється на $cwnd(t + t_A)$ відповідно (1):

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

де $ssth(t)$ [пакет] – значення порогу, при якому TCP йде від фази повільного старту в фазу уникнення перевантаження.

Коли внаслідок таймауту виявлено втрати пакетів, значення $cwnd(t)$ і $ssth(t)$ оновлюються наступним чином:

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

З іншого боку, коли TCP виявляє втрати пакетів згідно алгоритму швидкої передачі, $cwnd(t)$ і $ssth(t)$ оновлюються іншим чином:

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

TCP Reno потім переходить у фазі швидкого відновлення. На цьому етапі розмір вікна збільшується на один пакет, коли отримується дублююче підтвердження. З іншого боку, $cwnd(t)$ дорівнює $ssth(t)$ при надходженні підтвердження на пакет, що надіслане знову.

У разі таймауту $cwnd(t)$ $ssth(t)$ приймають вигляд (2).

Як можна легко побачити, механізм уникнення перевантажень, прийнятий в TCP Reno, викликає періодичні коливання через постійне оновлення розміру вікна. Швидкість, з якою кожне з'єднання оновлює розмір вікна залежить від циклу затримки з'єднання. Отже з'єднання із коротшим затримки може оновити розмір вікна швидше, ніж з'єднання з більшим часом затримки і тим самим несправедливо вкрасти частку пропускнуною здатності.

TCP Vegas приймає більш складні схеми оцінювання пропускнуною здатності. Він використовує різницю між очікуваною та фактичною швидкістю потоків для оцінки

пропускної здатності мережі. Ідея в тому, що коли мережі не перевантажена, фактична швидкість потоку буде близький до очікуваної. В іншому випадку фактична швидкість потоку буде менше, ніж очікувана швидкість потоку. TCP Vegas за допомогою цієї різниці швидкостей потоку, оцінює рівень заторів у мережі та відповідним чином оновлює розмір вікна. Звернімо увагу, що ця різниця швидкостей потоку можуть бути легко перекладений на різницю між розмір вікна і кількістю визнаних пакетів у часі, використовуючи рівняння:

$$Diff = (Expected - Actual) BaseRTT, \tag{4}$$

де Expected - очікувана швидкість, Actual - фактичний швидкість, BaseRTT - мінімальний час проходження сигналу туди та назад при першому з'єднанні

Деталі алгоритму:

1. Джерело обчислює очікувану швидкість потоку за формулою

$$Expected = (CWND) / BaseRTT, \tag{5}$$

де CWND - поточний розмір вікна.

2. Джерело оцінює поточну швидкість потоку за допомогою фактичного часу проходження сигналу туди та назад:

$$Actual = (CWND) / (RTT), \tag{6}$$

де RTT є фактичним часом проходження сигналу туди та назад.

Джерело, використовуючи очікувану та фактичну швидкість потоку, обчислює оцінку відставання у черзі за формулою (4).

Джерело оновлює розмір вікон таким чином:

$$CWND = \begin{cases} CWND + 1, & \text{if } Diff < \alpha \\ CWND - 1, & \text{if } Diff > \beta \\ CWND, & \text{otherwise.} \end{cases} \tag{7}$$

Рис. 1 ілюструє поведінку. Розглянемо просту мережу з одним підключенням і одним каналом ємністю C . Нехай BaseRTT бути затримка мінімального проходження сигналу туди і назад. Пропускна здатність цього з'єднання є $window / (BaseRTT)$ коли $window < C \times BaseRTT$.

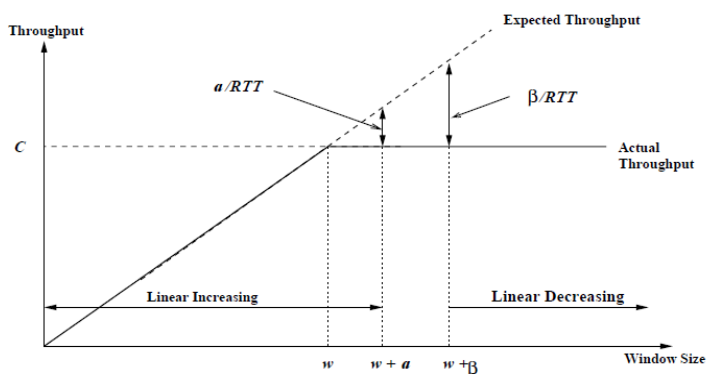


Рисунок 1 – Керування розміром вікна TCP Vegas

На рис. 1, w відповідає розміру вікна де $window = C \times BaseRTT$. Коли $window > w$, починає будуватися черга і $(Expected - Actual) > 0$. TCP Vegas збільшує розмір вікна під час наступного проходження сигналу якщо $window < w + \alpha$, зменшується розмір вікна, якщо $window > w + \beta$. В іншому випадку, він залишає розмір вікна без зміни.

На рис. 1, $Diff$ оцінює розмір черги. TCP Vegas намагається тримати прийнятні α пакети, але не більше, ніж β пакетів в черзі. Причиною цього є що TCP Vegas спробує виявити і використовувати додаткову пропускну здатність, коли вона стає доступна без

перевантажень в мережі. Таким чином, коли є тільки одне підключення, розмір вікна TCP Vegas сходиться до точки, що лежить між $w + \alpha$ і $w + \beta$.

Звернімо увагу, що цей механізм принципово відрізняється від того, що використовуються TCP Reno. TCP Reno завжди оновлює розмір вікна, щоб гарантувати повною утилізацію пропускної спроможності, що веде до постійної втрати пакетів, тоді як TCP Vegas викликає будь-яких коливання в розмірі вікна після того, як він сходиться до точка рівноваги.

Для дослідження роботи протиперевантажних алгоритмів скористаємося найпростішою моделлю мультисервісної мережі – моделлю з одним вузьким місцем. Для даної топології моделі мережі властиві всі характерні особливості телекомунікаційного трафіку інтегрованої мережі з комутацією пакетів, у тому числі його самоподібність і масштабна інваріантність.

На рис.2 наведена топологія такої мережі. За допомогою s_1, s_2, \dots, s_N - позначені відправники трафіку, r_1, r_2, \dots, r_N – одержувачі, R – маршрутизатори каналу. Вузьке місце приводить к тому, що пропускна спроможність всієї мережі обмежена одним або кількома компонентами, які називаються критичними елементами. В нашому випадку, критичними елементами є маршрутизатори каналу.

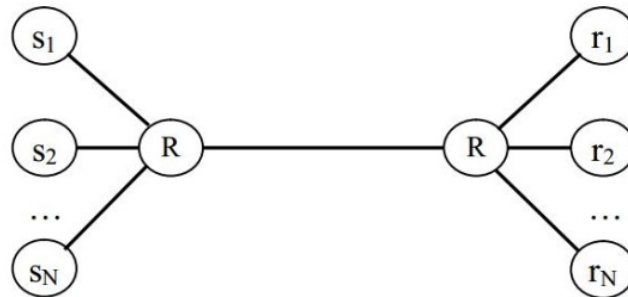


Рисунок 2 – Топологія мережі з вузьким місцем

Розрахунок зміни розміру вікна $\dot{W}(t)$ та довжини черги $\dot{Q}(t)$ виконується за допомогою гідродинамічної моделі потоку. Для кожного алгоритму розробляється своя гідродинамічної моделі, яка враховує конкретні особливості даного алгоритму. В даному випадку використовується алгоритми TCP Reno та TCP Vegas .

Гідродинамічна модель потоку включає такі поєднанні нелінійних диференціальних рівнянь з затримкою, яка змінюється з часом:

$$\dot{W}_{reno}(t) = \frac{1}{R(t)} - \frac{W(t) * W(t - R(t))}{2 * R(t - R(t))} p(t - R(t)) , \quad (8)$$

$$\dot{W}_{vegas}(t) = G(\alpha - Diff) - G(Diff - \beta) , \quad (9)$$

$$\dot{Q}(t) = \begin{cases} N(t) \frac{W(t)}{R(t)} - C, & \text{для } q > 0 \\ \max(N(t) \frac{W(t)}{R(t)} - C, 0), & \text{для } q = 0 \end{cases} \quad (10)$$

де $W(t)$ позначає середній розмір вікна TCP [пакети], $Q(t)$ – середня довжина черги [пакети], $R(t)$ – RTT [секунди], C – пропускна спроможність каналу [пакет/с], $N(t)$ – кількість TCP сесій, $p(t)$ – ймовірнісна функція маркірування пакету, $G(x)$ – нелінійна функція, яка описує зміну розміру вікна.

Функція $G(x)$ описується виразом:

$$G(x) = \begin{cases} 1, & \text{для } x > 0 \\ 0, & \text{для } x < 0 \end{cases} \quad (11)$$

Диференціальні рівняння (3) та (4) описують динамічне керування вікна TCP. Перший елемент $\frac{1}{R_i}$ описує у вікні адитивне збільшення фази, другий елемент $\frac{W(t)}{2}$ мультиплікативне зменшення фази (в тому числі ймовірнісна функція маркірування пакету). Елемент $G(\alpha - Diff)$ в рівнянні (9) описує збільшення розміру вікна на 1, а другий елемент $G(Diff - \beta)$ – зменшення розміру вікна на 1.

Рівняння (10) описує довжину черги вузьким місцем, як різниця між швидкістю прибуття пакету $\frac{NW}{R}$ і пропускною спроможністю C , припускаючи, що там є немає внутрішньої динаміки в вузькому місці (грубо кажучи, простих інтеграторів).

За допомогою гідродинамічної моделі потоку активне керування чергою можна інтерпретувати як зворотній зв'язок проблеми контролю, де дія контролю складається з маркування пакетів (з ймовірністю p) в залежності від довжини вимірної черги Q .

Для кращого розуміння принципу моделі гідродинамічного потоку представимо рівняння (8) та (10) схемою, яка приведена на рис. 3.

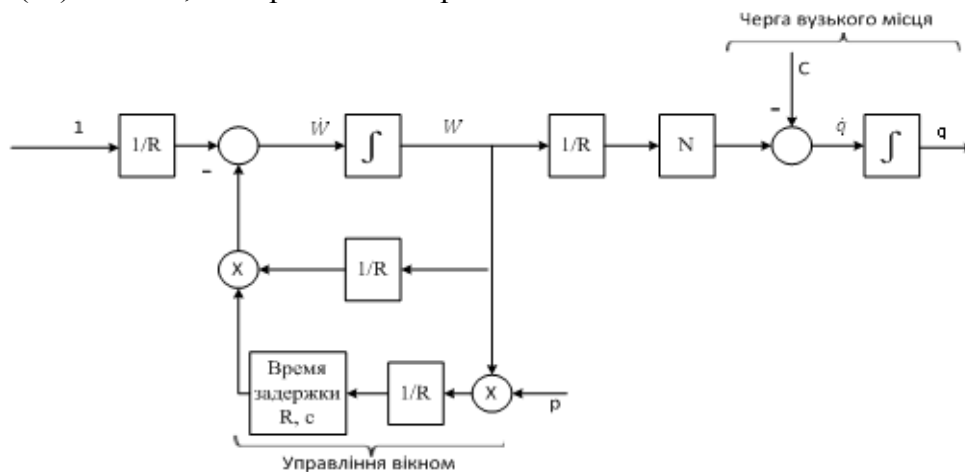


Рисунок 3 – Схема режиму управління потоком за для уникнення перевантаження в вузькому місці для TCP Reno

Принцип гідродинамічної моделі потоку для TCP Vegas наведено на рис. 4.

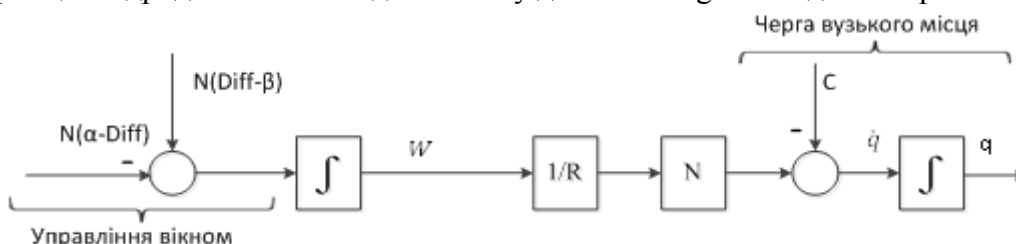


Рисунок 4 – Схема режиму управління потоком за для уникнення перевантаження в вузькому місці для TCP Vegas

Для моделювання використовуємо пакет прикладних програм MATLAB.

Вхідні параметри моделювання: RTT $R=0,1$ с, максимальна довжина черги $Q_{max}=200$ пакетів, кількість відправників S змінюється динамічно.

Результати моделювання приведені на рис.5-6 та в табл.1.

Як можна побачити з рис. 5 процес формування черги при пропускній спроможності $C=1$ Мбіт/с ідентичний – за короткий проміжок часу черга досягає свого максимального значення та не змінюється не дивлячись на зміну кількості користувачів. При збільшенні пропускної спроможності каналу зв'язку спостерігаються відмінності в процесі формування черги при використанні різних алгоритмів. Це ще раз доводить, що алгоритм TCP Vegas використовує пропускну спроможність каналу зв'язку більш ефективно.

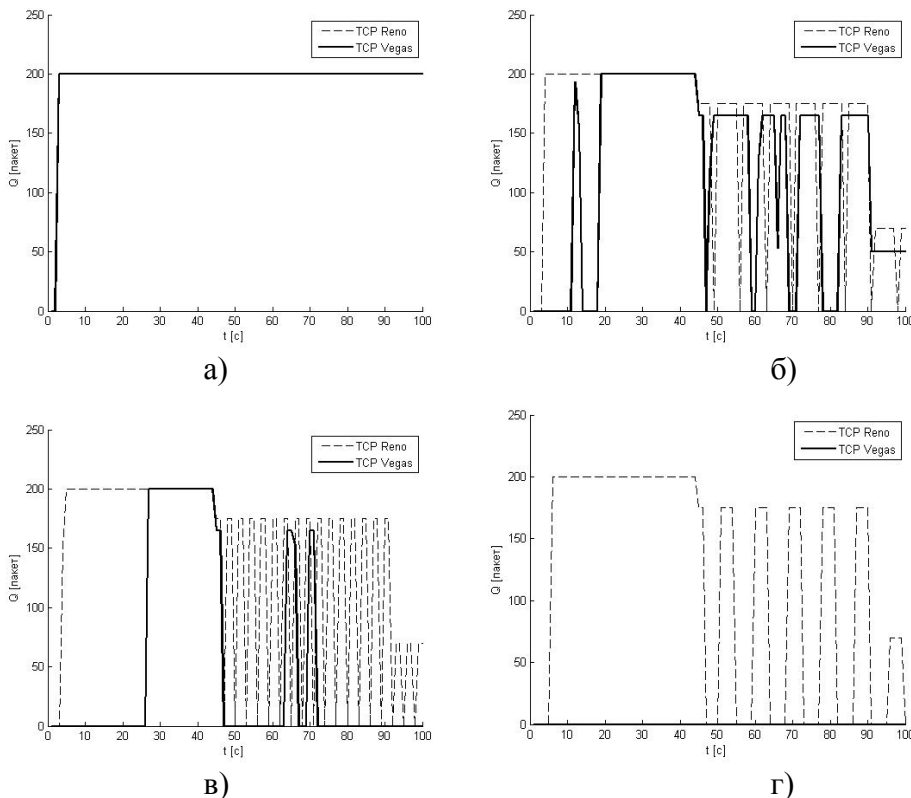


Рисунок 5 – Графік залежність довжини черги $Q(t)$ при а) $C = 1$ [Mbit/c], б) $C = 10$ [Mbit/c], в) $C = 50$ [Mbit/c], г) $C = 100$ [Mbit/c]

Проаналізувавши графіки, приведені на рис.6, можна побачити, що середній розмір вікна TCP Vegas менше ніж у TCP Reno, але за рахунок меншого коливання, TCP Vegas передає більшу кількість даних.

Як можна побачити з табл.1, канал зв'язку з використанням алгоритму TCP Vegas з пропускнуою спроможністю від 10 Мбіт/с можна вважати каналом високої якості та використовувати для передачі трафіку даних, для якого імовірність втрати пакету повинна бути не більше 10^{-4} . В той час коли канал зв'язку с використанням алгоритму TCP Reno, набуває такого рівня якості при пропускній спроможності від 50 Мбіт/с.

Таблиця 1

Результати моделювання

	C , [Mbit/c]	V , [пакет]	V_{lost} , [пакет]	p	\bar{Q} , [пакет]
Vegas	1	110707	58	$5 \cdot 10^{-4}$	198
	10	215395	21	$9 \cdot 10^{-5}$	143
	50	756532	15	$2 \cdot 10^{-5}$	37
	100	3632607	0	0	0
Reno	1	101648	58	$6 \cdot 10^{-4}$	198
	10	203934	43	$2 \cdot 10^{-4}$	159
	50	730570	36	$5 \cdot 10^{-5}$	143
	100	3366720	9	$3 \cdot 10^{-6}$	120

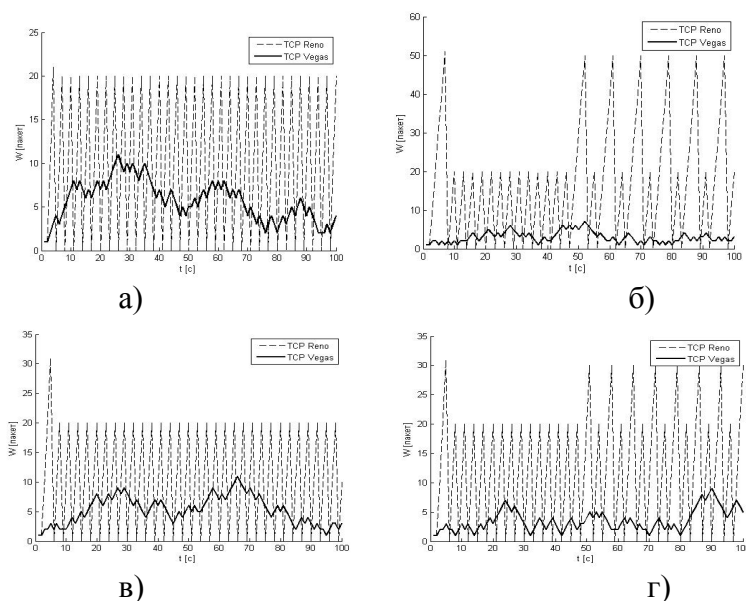


Рисунок 6 – Графік залежність розміру вікна $W(t)$ при а) $C = 1$ [Mbit/c], б) $C = 10$ [Mbit/c], в) $C = 50$ [Mbit/c], г) $C = 100$ [Mbit/c]

Висновки

1. Моделювання роботи алгоритмів протоколів TCP Reno та TCP Vegas показало що двофазна робота першого алгоритму призводить до значних коливань навантаження на мережу та більшому рівню втрат, ніж у алгоритму TCP Vegas. За рахунок реалізації більш складної схеми управління пропускну здатністю, досягнуто зменшення рівня коливань корисного навантаження на мережу та зменшення втрат.

2. Використання гідродинамічна модель потоку дозволяє враховувати особливості функціонування проти перевантажених алгоритмів. Модель може бути використана для моделювання поведінки різних реалізацій алгоритму TCP в умовах мереж складеної топології.

3. Проведене моделювання показує, що алгоритм TCP Vegas більш ефективно використовує пропускну здатність каналу зв'язку, ніж TCP Reno за рахунок меншого коливання розміру вікна.

Список використаної літератури

1. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет / Е.А. Кучерявый. - СПб.: Наука и техника, 2004. – 336 с.
2. Джамалипур А. Беспроводной мобильный Интернет. Архитектура, протоколы и сервисы / А. Джамалипур; [пер.с англ. Ш. Салиев, В. Орлов]. – М., Техносфера, 2009 – 496 с.
3. Fall K. Simulation based comparisons of Tahoe, Reno and SACK TCP / K. Fall, S. Floyd // Computer Communications Review. - July 1996. - 26(3). – PP. 5-21.
4. Analysis and comparison of TCP Reno and Vegas [Електронний ресурс] / [Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand]. - Режим доступу до статті: <http://www.eecs.berkeley.edu/~anath/1999-2001/Richard/MoLaInfocom1999.pdf>.
5. Thomas Bonald Comparison of TCP Reno and TCP Vegas via Fluid Approximation [Електронний ресурс] / Thomas Bonald // RR3563, Unite de recherche INRIA Sophia Antipolis Cedex (France). – 1998 – 34 pages. - Режим доступу до статті: netlab.caltech.edu/FAST/references/bonald_comparison.pdf. – Час доступу: січень 2013 р.
6. A control theoretic analysis of RED [Електронний ресурс] / [C.V. Hollot, Vishal Misra, Don Towsley and Wei-Bo Gong]. - Режим доступу до статті: [ftp://gaia.cs.umass.edu/pub/MisraInfocom01-RED-Control.pdf](http://gaia.cs.umass.edu/pub/MisraInfocom01-RED-Control.pdf). – Час доступу: січень 2013 р.

7. Srikant R. The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications) / Rayadurgam Srikant. – Birkhäuser Boston, 2004
8. Батыр С.С. Построение модели сети передачи данных для исследований технологии AQM / С.С. Батыр, А.В. Хорхордин // Сб. научн. трудов ДониЖТ. - 2011. – Вып. 28 – С. 108-116.

Надійшла до редакції:
18.04.2013

Рецензент:
д-р техн. наук, проф. Скобцов Ю.О.

V.V. Trikoz, S.S. Batyr

ГВУЗ «Донецкий национальный технический университет»

Исследование работы противонагруженных алгоритмов протокола TCP при перегрузках в каналах связи. Выполнены исследования, моделирование и анализ работы алгоритмов для борьбы с перегрузкой протоколов транспортного уровня стека TCP / IP. Для исследования были выбраны алгоритмы TCP Reno и TCP Vegas. Рассмотрены принципы работы выбранных алгоритмов. Для моделирования использована гидродинамическая модель потока, с помощью которой были учтены особенности работы выбранных алгоритмов. Полученные результаты моделирования позволили оценить поведение алгоритмов при перегрузках, и эффективность использования пропускной способности канала связи.

Ключевые слова: *перегрузки, алгоритм управления перегрузкой, TCP Reno, TCP Vegas, гидродинамическая модель.*

V.V. Trikoz, S.S. Batyr

Donetsk National Technical University

TCP Anti-congestion Algorithms Research in Telecommunication Channel's Congestions. Algorithms of transport layer of TCP/IP stack were researched, modeled and analyzed. Algorithms TCP Reno and TCP Vegas were selected for the research. The principles of operation of selected algorithms were reviewed. The window size in TCP Reno is changed cyclically in a normal situation. Window size increases during each cycle to packet loss. When there is the packet loss, TCP Reno reduces the window size to half the current size. This is called additive increase and multiplicative decrease. TCP Vegas adopts more complex evaluation capacity. It uses the difference between the expected and actual flow rate to estimate the network bandwidth. The idea is that when the network is not congested, the actual flow rate will be close to the expected. Otherwise, the actual flow rate will be less than the expected flow rate. TCP Vegas with this speed difference flow assesses the level of congestion in the network and accordingly updates the window size. For simulation the fluid flow model was used, which helped to take into account the features of the selected algorithms. Using fluid flow model the active queue management can be interpreted as a feedback control problem where the control action consists of marking packets (with probability p), depending on the length of the measured queue Q . For the study of anti-congestion algorithms we used a simple model of multiservice network - a model with a single bottleneck. For a given network topology model has all the characteristics of telecommunication traffic of integrated packet-switched networks, including its self-similarity and scale invariance. Based on these results the following conclusions were made: Simulation of algorithms protocols TCP Reno and TCP Vegas has shown that two-phase job of the first algorithm leads to significant fluctuations in network load and to greater losses than the algorithm TCP Vegas. At the expense of a more complex control circuit of bandwidth, we achieved the reduction of vibrations payload on the network and losses. Our simulations show that the algorithm of TCP Vegas more efficiently uses the bandwidth of the communication channel than TCP Reno due to the smaller window size fluctuations.

Keywords: *congestion, congestion control algorithms, TCP Reno, TCP Vegas, fluid flow model.*