

УДК 621.372.061.3.001.63:681.3

А.В. Карпухин

Харьковский национальный университет радиоэлектроники, Харьков

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПРОТОКОЛА TCP В СОВРЕМЕННЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

Рассматриваются вопросы реализации современных версий основного транспортного протокола Internet - TCP. Проведено сравнение алгоритмов работы различных версий протокола. Приведены результаты имитационного моделирования работы основных версий протокола, а также сравнение их характеристик.

Ключевые слова: TCP, сетевые протоколы, потери пакетов, окно перегрузки, качество обслуживания сети, управление перегрузкой.

Введение

Протокол TCP является основным протоколом работы в Internet. Он осуществляет доставку данных в виде потоков байтов с установлением соединения и применяется в случаях, когда нужна гарантированная доставка сообщений. Протокол TCP использует контрольные суммы пакетов для проверки их целостности и освобождает прикладные процессы от необходимости таймаутов и повторных передач для обеспечения надежности.

Многочисленные исследования процессов в сети Internet показали, что статистические характеристики трафика имеют свойство временной масштабной инвариантности (самоподобия). Причина такого эффекта - в особенностях распределения файлов по серверам, их размерам, а также в типичном поведении пользователей. Оказалось, что потоки данных сначала не проявляют свойства самоподобия, но пройдя обработку на узловых серверах и активных сетевых элементах, начинают подавать ярко выраженные признаки самоподобия. Из-за этого возможна быстрая перегрузка буферов устройств даже при небольших коэффициентах использования. Если не принять мероприятий по ограничению поступающего трафика очереди на наиболее нагруженных линиях будут неограниченно расти и, наконец, превысят размеры буферов в соответствующих узлах. Это приводит к тому, что пакеты, которые вновь поступают в узлы, в которых отсутствует свободное место в буфере, будут отброшены и должны будут передаваться повторно, что приводит к нерациональной затрате ресурсов сети [1 – 3].

В протокол TCP постоянно вносятся изменения и дополнения, которые стараются решить проблемы, возникающие в процессе применения протокола, или улучшить его характеристики для систем узкой специализации.

Целью данной работы является сравнительный анализ разновидностей протоколов семейства TCP и алгоритмов их работы; выявление недостатков и преимуществ каждого алгоритма при разных сценариях работы в сети.

Анализ работы протоколов TCP

Рассмотрим основные разновидности протоколов семейства TCP: Tahoe, Reno, NewReno, SACK, Vegas [4, 5].

Для проведения имитационного моделирования в сетевом симуляторе Opnet была построена модельная сеть, которая состоит из нескольких отправителей, получателя и установленного между ними маршрутизатора (рис. 1).

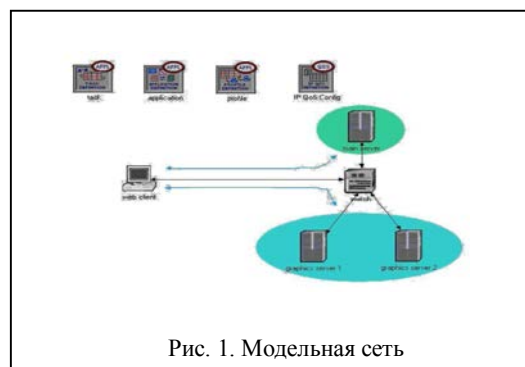


Рис. 1. Модельная сеть

Узким местом в сети является маршрутизатор и канал, который выходит. В ходе проведения численного эксперимента изменялись алгоритмы работы протокола TCP, размеры буфера маршрутизатора и получателя, полоса пропускания на выходе маршрутизатора. Трафик в данной сети является самоподобным случайным процессом с параметрами, которые задаются пользователем. Одним из параметров является показатель Херста, который характеризует долгосрочную зависимость процесса и лежит в диапазоне $[0,5; 1]$. Характеристиками, по которым проводился сравнительный анализ работы сети, были: количество потерянных данных, загрузка буфера маршрутизатора, утилизация канала, производительность работы сети.

Имитационное моделирование показало, что для всех вариантов протокола TCP при работе алгоритмов *Медленного старта* и *Избежания перегрузки* размер окна $CWND$ (congestion window) со временем растёт линейно (линия 1 на рис. 2).

При одном потерянном сегменте для протокола TCP Tahoe (линия 2) и протокола TCP Reno (линия 3) уменьшение $CWND$ происходит значительно раньше, соответственно загрузка канала и буфера является более равномерным и ресурсы системы используются более оптимально

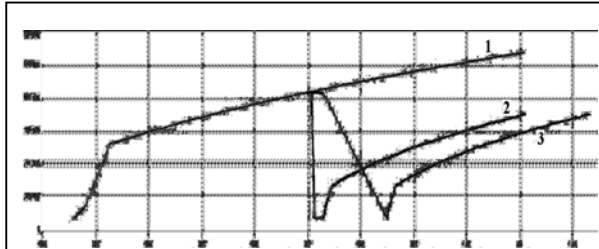


Рис. 2. Изменение $CWND$:
линия 1 – Медленный старт и Избегание перегрузки;
линия 2 – Tahoe при одной потере сегмента;
линия 3 – Reno при одной потере сегмента

В работе был проведен сравнительный анализ образования очередей в буфере для разных алгоритмов протокола TCP при увеличении количества потоков входных данных (табл. 1). Это имеет особое значение, например, для сетей с несколькими каналами VPN и многопортовыми маршрутизаторами.

Таблица 1
Значение средней длины очереди (Мб)
при различном количестве потоков

	1	2	3	4	5	6	7
Tahoe	9900	11700	11900	15000	15300	15600	16000
Reno	9800	11500	11500	14700	15800	16000	16100
NewReno	9100	9500	9700	10600	12000	13800	15200
Sack	5300	5600	6900	7200	8600	8800	9100
Vegas	5200	5600	7000	7000	7700	8500	8900

При увеличении длины очереди рост RTT (Round Trip Time) неминуем. При достижении нижнего порога очереди начинаются потери пакетов. Когда же будет превышен верхний порог, начинают отбрасываться пакеты, которые приходят с низким приоритетом. Если бы отбрасывание пакетов не начиналось до переполнения буфера, можно было бы ожидать линейного роста RTT. Достигнув предельного значения очереди, количество утраченных пакетов возрастало бы практически скачкообразно.

Одним из важнейших показателей качества обслуживания сети (QoS) есть коэффициент использования канала. В табл. 2 представлены значения этого показателя при эволюции размера окна со временем.

В последние годы были предложены несколько новых модификаций протокола TCP: Binary Increase Control TCP (BIC-TCP), CUBIC TCP, Westwood TCP (TCPW), Parallel TCP Reno (P-TCP), Scalable TCP (S-TCP), Fast TCP, HighSpeed TCP (HS-TCP), High-Speed TCP Low Priority (HSTCP-LP), Hamilton TCP (H-TCP), Yet Another Highspeed TCP (YeAH-TCP), Africa TCP, Compound TCP и т.д.

Таблица 2
Коэффициент использования канала
при эволюции размера окна (мс)

	100	200	300	400	500	600	700
Tahoe	1	0,8	0,72	0,72	0,65	0,6	0,56
Reno	1	0,82	0,75	0,78	0,62	0,54	0,4
NewReno	1	0,87	0,79	0,85	0,74	0,68	0,6
Sack	1	0,93	0,88	0,99	0,82	0,87	0,99
Vegas	1	0,94	0,88	1	0,85	0,9	0,98

Почти все они базируются на известных старых версиях TCP и отличаются различными способами предотвращения заторов (congestion) (точнее, различными способами определения факта наличия потери пакетов, что означает возникновение затора). В различных модификациях используются различные формулы для расчета $cwnd$ (congestion window).

В частности, ученые из Университета Северной Каролины разработали сетевой протокол, который, по их замыслу, призван сменить устаревший TCP. Создававшийся в 80-х годах прошлого столетия стандарт TCP был рассчитан на совершенно другие линии связи и намного меньшие объемы информации, нежели те, которыми приходится оперировать рядовым пользователям Internet и научным организациям сегодня. В TCP возможны длительные временные задержки, кроме того, данный протокол не обеспечивает необходимую масштабируемость и надежность. В новом стандарте, названном учеными BIC-TCP (сокращенно от Binary Increase Congestion Transmission Control Protocol), устранены все недостатки, присущие традиционному TCP. В частности, протокол BIC-TCP обеспечивает примерно в 6000 раз более высокую скорость передачи информации по широкополосным каналам и в 150 раз большую пропускную способность, нежели коммутируемые линии связи, использующие TCP. Ключевой особенностью BIC-TCP является применение двоичного поиска для быстрого выявления оптимального пути передачи данных с минимальными потерями. Плюс к этому, в BIC-TCP существенно сокращены объемы передаваемой служебной информации за счет применения пакетов большего объема. В результате, по словам исследователей, те задачи, на выполнение которых в TCP уходит несколько часов, в BIC-TCP могут быть обработаны буквально за минуты, а то и секунды.

Сегодня существует насущная потребность перемещения больших объемов данных (терабайты) между сайтами в Европе, Японии и США. Это, в первую очередь, актуально в таких областях, как физика высоких энергий и обмен данными между многими отдаленными компьютерами и дата-центрами во всем мире.

В настоящее время в большинстве случаев перемещения больших объемов данных по сети используется множественные параллельные стандартные потоки TCP (на основе TCP Reno). Оптимизация раз-

меров окна и числа параллельных потоков является трудоёмкой, сложной задачей и осуществляется в некоторых случаях в течение нескольких часов в зависимости от конфигурации сети и ее загрузки. В работе [6] приведены результаты измерений, сделанных в реальных сетях большой производительности с различными реализациями протокола TCP на маршрутах пакетов с различными RTT при использовании оптимальных и субоптимальных размеров окна. В ней было проведено сравнение протокола New Reno со следующими протоколами: HS-TCP, Fast TCP, S-TCP, HSTCP-LP, H-TCP и BIC-TCP. Сравнивались достижимая пропускная способность, влияние RTT, внутри- и межпротокольная «равнодоступность», устойчивость и влияние обратного трафика. Измерения проводились для трех конфигураций сети: малой, средней и большой протяженности.

Протоколы P-TCP, S-TCP, HS-TCP, HSTCP-LP, H-TCP, BIC-TCP, CUBIC TCP являются улучшением TCP Reno. FAST TCP является развитием TCP Vegas [6].

Необходимость передачи больших объемов данных с большим быстродействием требует развертывания соответствующих сетевых инфраструктур. Однако основной транспортный сетевой протокол TCP не отвечает этому требованию.

Ниже приведено описание и анализ основных характеристик вышеупомянутых модификаций протоколов.

P-TCP (Parallel TCP). Сегодня физики чаще всего используют TCP Reno со многими параллельными потоками для того, чтобы достичь большой производительности [6].

Однако:

- он может быть сверхагрессивным и «несправедливым» к другим версиям протокола TCP.
- оптимальное число параллельных потоков может значительно варьироваться в зависимости от изменений (например, маршрутов пакетов) или использования возможностей сетей.

Чтобы быть эффективными в условиях высокой производительности, лучшие новые усовершенствованные протоколы при использовании одного TCP-потока, должны обеспечивать характеристики, подобные характеристикам P-TCP (параллельный Reno TCP) и кроме того, они должны иметь лучшую «справедливость» (по отношению к другим вариантам TCP) чем P-TCP.

В работе [6] использовалась реализация 2.4.22 ядра GNU/Linux, которая включает SACK (RFC 2018) и New Reno (RFC 2582). В ней все еще используется механизм AIMD, реализующий формулы:

Congestion Avoidance

$$\text{ACK: } \text{new}_{\text{cwnd}} = \text{old}_{\text{cwnd}} + \frac{a}{\text{old}_{\text{cwnd}}}$$

$$\text{DROP: } \text{new}_{\text{cwnd}} = \text{old}_{\text{cwnd}} - b * \text{old}_{\text{cwnd}}$$

Where $a = 1, b = 0.5, c = 1$.

S-TCP (Scalable TCP). Scalable TCP изменяет традиционный алгоритм управления перегрузкой TCP: вместо того, чтобы использовать аддитивное увеличение, используется показательное увеличение, и мультипликативный коэффициент уменьшения b устанавливается равным 0,125, чтобы уменьшить потерю производительности после возникновения перегрузки [7].

Fast TCP. Протокол Fast TCP – единственный протокол, который основан на Vegas TCP вместо TCP Reno. В нем используются задержка в очереди и потеря пакета для определения факта возникновения перегрузки [8, 9]. Это уменьшает массовые потери пакетов благодаря использованию пошагового алгоритма в передатчике, что приводит к быстрой сходимости к равновесному значению.

HS-TCP. HighSpeed TCP был описан в [10, 11] как модификация механизма управления перегрузкой в TCP, чтобы улучшить рабочие характеристики TCP в быстродействующих сетях с большой задержкой. Эта модификация ведет себя как TCP Reno для малых значений cwnd , но выше выбранного значения cwnd используется более «агрессивная» функция реакции. Когда величина cwnd больше (больше чем 38 пакетов, что эквивалентно коэффициенту потерь 1 в 1000), данная версия TCP использует таблицу, чтобы указать, насколько должно быть увеличено окно перегрузки, когда ACK (acknowledgment – подтверждение) получено. При этом используется меньше пропускной способности сети, чем $1/2 \text{ cwnd}$ при потере пакета. Существует две версии HighSpeed TCP [12, 13].

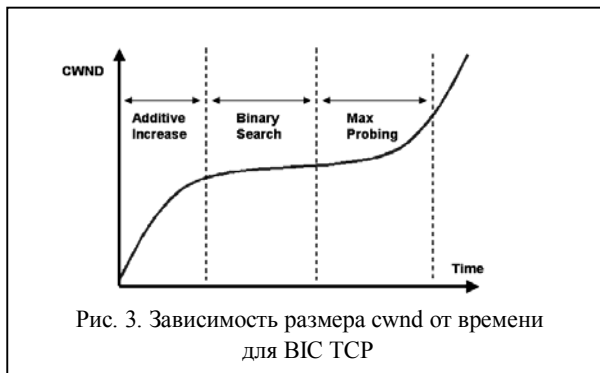
HSTCP-LP. Целью этой модификации, которая основана на TCP-LP [14], является использование только избыточной пропускной способности сети, при этом оставшаяся неиспользованная ее часть может быть использована другими TCP-потоками.

Давая более высокий приоритет всем TCP-потокам, не использующим протокол HSTCP-LP, эта модификация использует простой двухклассовый механизм приоритетов без какой-либо поддержки со стороны сети. Протокол HSTCP-LP был реализован, как объединение HS-TCP и TCP-LP.

BIC TCP [15]. Медленная реакция TCP в быстродействующих сетях большой длины (fast long-distance networks) оставляет большую неиспользованную пропускную способность в таких сетях. BIC TCP и CUBIC TCP – это протоколы управления перегрузкой, разработанные, чтобы устранить эту проблему.

Когда происходит потеря пакета, BIC TCP сокращает свое окно на мультипликативный коэффициент (β). Размер окна как раз перед сокращением устанавливается в максимум и размер окна сразу после сокращения устанавливается в минимум. Затем BIC TCP выполняет двоичный поиск, используя эти два параметра и переходя к "середине" между максимумом (W_{max}) и минимумом (W_{min}). С того момента, когда произошла потеря пакета при размере окна

W_{max} , размер окна, который сеть может в настоящее время обрабатывать без потери, должен быть где-нибудь между этими двумя значениями (рис. 3).

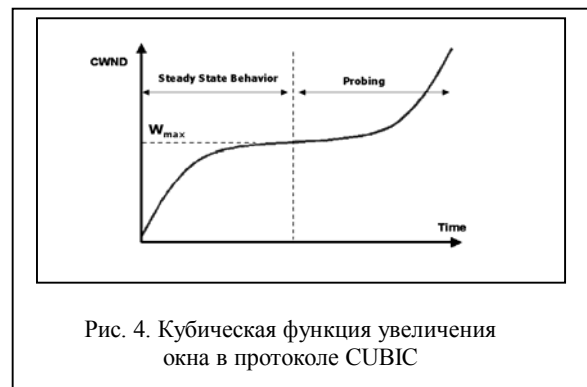


Однако, переход к середине может быть слишком большим увеличением в пределах одного RTT, так, если расстояние между серединой и текущим минимумом является большим, чем фиксированная постоянная (названная S_{max}), BIC TCP увеличивает текущий размер окна на S_{max} (линейное увеличение). Если потери пакета при обновленном размере окна не происходит, этот размер окна становится новым минимумом. Если потеря пакета происходит, этот размер окна становится новым максимумом. Этот процесс продолжается, пока приращение окна не меньше, чем некоторая малая константа S_{min} , и в этой точке размер окна устанавливается равным текущему максимуму. Так, функция роста после уменьшения окна, наиболее вероятно, будет линейной, а затем логарифмической (эти области обозначены на рис. 3 как **additive increase** ("аддитивное увеличение") и **binary search** ("двоичный поиск").

Если окно становится больше предыдущего максимума, равновесный размер окна должен быть большим, чем текущий максимум и тогда должен быть найден новый максимум. BIC TCP вступает в новую фазу, называемую "исследование на максимум". Функция роста в этот период является обратной функцией роста в двоичном поиске и аддитивном увеличении. Она растет по экспоненте (вначале очень медленно) и затем линейно. Исследование на максимум использует функцию роста окна, симметричную функции роста, используемой в двоичном увеличении (аддитивное увеличение и двоичный поиск) - только в другом порядке: этот этап использует инверсию двоичного поиска (который является логарифмическим; его обратная величина будет экспоненциальной), и затем аддитивное увеличение. Рис. 3 показывает функцию роста в течение исследования на максимум. В течение исследования на максимум, окно растет первоначально медленно, чтобы обнаружить новый максимум поблизости, и после некоторого времени медленного роста, если новый максимум не обнаруживается (т.е. есть потери пакетов), тогда предполагается, что новый максимум — еще дальше, таким образом, алгоритм переключается к более бы-

строму увеличению путем перехода к аддитивному увеличению, где размер окна увеличивается с большим постоянным приращением. Хорошие рабочие характеристики BIC TCP определяются медленным увеличением (см. горизонтальный участок на рис. 3) вокруг W_{max} и линейным увеличением в течение аддитивного увеличения и исследования на максимум.

CUBIC TCP [16]. Хотя BIC TCP достигает довольно хорошей масштабируемости, «справедливости» (по отношению к другим протоколам) и стабильности в современных высокоскоростных сетях, функция роста BIC TCP может быть слишком агрессивной для TCP, особенно при малых RTT или в низкоскоростных сетях.



Кроме того, несколько различных фаз управления окном значительно увеличивают сложность анализа протокола. В новой реализации BIC TCP используется новая функция роста окна — кубическая. На рис. 4 показана кубическая функция, форма которой подобна кривой окна BIC TCP (см. рис.3). Функция возрастает намного медленнее, чем двоичное увеличение (которое является логарифмическим) на горизонтальном участке.

Выводы

Таким образом, в ходе проведенных экспериментов и анализа результатов, выявлено, что при работе TCP Vegas качество обслуживания сети значительно лучше и количество утраченных пакетов значительно меньше, чем при работе других протоколов TCP. Протокол TCP Vegas: является более стойким при потерях пакетов и может намного быстрее выявить и повторно передать утраченный пакет, чем Tahoe; не должен ждать 3-х двойных ACK и потому он может быстрее повторно передать утраченный пакет, чем Reno и NewReno; благодаря измененным алгоритмам *Предотвращения перегрузки* и *Медленного старта* осуществляет меньше повторных передач, что приводит к более эффективному использованию ресурсов сети, чем в случае NewReno и Tahoe; при оценивании перегрузки измеряет пропускную способность и изменяет ее вместо потери пакета, что дает лучшее использование полосы пропускания и меньше перегрузок, чем в Tahoe и SACK; выравнива-

ет свою норму посылки пакетов к получателю в оптимальной полосе пропускания, повышая таким образом стабильность, в отличие от SACK.

Современные коммуникационные сети являются в большей части гетерогенными – в них работают различные протоколы (в том числе и транспортные). Практически невозможно обеспечить переход сразу всей сетевой инфраструктуры на использование более новой версии какого-либо транспортного протокола (аналогичная ситуация существует с внедрением версии IP v6). В связи с этим большой интерес представляет исследование производительности транспортных протоколов в средах, где работают различные версии протоколов. Это необходимо для того, чтобы быть уверенным, что внедрение нового транспортного протокола или его версии не создаст проблем в сети.

Целью разработки усовершенствованных вариантов протокола TCP является стремление улучшить его характеристики, такие как «дружественность» по отношению к другим версиям TCP, стабильность, коэффициент утилизации пропускной способности сети. В своем большинстве эти протоколы используют по существу эмпирические алгоритмы, обоснование которых осуществляется в большей мере на «интуитивном» уровне.

Более перспективным, на наш взгляд, является синергетический подход к моделированию сетевых информационных систем с протоколом TCP. В таких системах возможно возникновение синхронизации конкурирующих TCP - потоков за счет явлений самоорганизации либо с использованием т.н. пейсмейкера. Хотя очевидным является то, что транспортные протоколы будут развиваться и модифицироваться для достижения лучшей производительности.

Данная работа выполнена при поддержке ДФФД (грант Ф25.1/027 «Аналіз поведінки інформаційних систем з великою нелінійністю: синергетичний підхід»).

Список литературы

1. On the self-similar Nature of Ethernet Traffic / W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson // *IEEE/ACM Transactions of Networking*. – 1994. – Vol. 2(1). – P. 1-15.

2. Столлингс В. *Современные компьютерные сети* / В. Столлингс. – СПб.: Питер, 2003. – 783 с.

3. Kirichenko L. Improvement quality of network service under selfsimilar loading / L. Kirichenko, T. Radivilova, O. Karpukhin // *Стратегия качества в промышленности и образовании: 4-я междунар. конф.: материалы конф.* – Варна, 2008. – С. 612-615.

4. Floyd S. Random Early Detection Gateways for Congestion Avoidance / S. Floyd and V. Jacobson. // *IEEE/ACM Transactions on Networking*. – 1993. – Vol. 1 (4). – P. 397-413.

5. Fall K. Simulation-based comparison of Tahoe, Reno, and SACK TCP / K. Fall and S. Floyd // *Computer Communication Review*. – July 1996. – Vol. 26. – P. 5-21.

6. Bullo H. Evaluation of Advanced Stacks on Fast Long-Distance / H. Bullo, R. Les Cottrell, R. Hughes-Jones // *Production Networks J.Grid Comput.* 1. – 2003. – P.345-359.

7. Kelly T. Scalable TCP: Improving performance in highspeed wide area networks. *First International Workshop on Protocols for Fast Long-Distance Networks* / T. Kelly. – Geneva, February 2003.

8. FAST TCP: From theory to experiments. In *First International Workshop on Protocols for Fast Long-Distance Networks (PFLDNet 2003)* / C. Jin, D. Wei, S.H. Low, G. Bushmaster, J.Bunn, D.H. Choe, R.L.A Cottrell, J.C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot and S. Singh. – Geneva, February 2003.

9. Jin C. FAST TCP: Motivation, architecture, algorithms, performance / C. Jin, D.X. Wei, S.H. Low. // *In IEEE INFOCOM 2004*. – Hong Kong, March 2004.

10. Floyd S. Limited slow-start for TCP with large congestion windows / S. Floyd // *RFC3742, ICSI*. – March 2004.

11. Floyd S. HighSpeed TCP for large congestion windows / S. Floyd // *RFC3649, ICSI*. – December 2003.

12. Y.Li. [Электронный ресурс] – Режим доступа к док.: <http://www.hep.ucl.ac.uk/~ytl/tcpip/hstcp/>.

13. T. Dunigan. [Электронный ресурс] – Режим доступа к док.: <http://www.csm.ornl.gov/~dunigan/net100/>.

14. Kuzmanovic A. TCP-LP: A distributed algorithm for lowpriority data transfer / A. Kuzmanovic, E.W. Knightly. // *In IEEE INFOCOM*. – San Francisco, April 2003.

15. Xu L. Binary increase congestion Control (BIC) for Fast Long-Distance Networks / L. Xu, K. Harfoush, I. Rhee // *Proceedings of IEEE INFOCOM 2004*. – Hong Kong, March 2004.

16. Ha S. CUBIC: A New TCP-Friendly High-Speed TCP Variant / S. Ha, I. Rhee, L. Xu, // *ACM SIGOPS Operating System Review*. – July 2008. – Vol. 42, Issue 5. – P. 64-74.

Поступила в редколлегию 25.09.2009

Рецензент: д-р техн. наук, проф. О.Г. Руденко, Харьковский национальный университет радиоэлектроники, Харьков.

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПРОТОКОЛУ TCP В СУЧАСНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

О.В. Карпукін

Розглянуті питання реалізації сучасних версій основного транспортного протоколу Internet – TCP. Проведено порівняння алгоритмів роботи різних версій протоколу. Наведені результати імітаційного моделювання роботи основних версій протоколів, а також порівняння їх характеристик.

Ключові слова: TCP, мережеві протоколи, втрати пакетів, вікно переваантаження, якість обслуговування мережі, управління переваантаженням.

TCP IMPLEMENTATION HABITS IN MODERN COMPUTER NETWORKS

O.V. Karpukhin

Questions of implementation of modern versions of the basic transport protocol of the Internet (TCP) are observed. Comparison of various protocol algorithm versions was made. Results of simulation modelling of the basic versions of protocols and also the comparison of their characteristics are included.

Keywords: TCP, network protocols, losses of packets, congestion window, network quality of service, overloading control.