

ДИНАМИЧЕСКАЯ МОДЕЛЬ TCP-СЕАНСА ПРИ УПРАВЛЕНИИ ДЛИНОЙ ОЧЕРЕДИ С ИСПОЛЬЗОВАНИЕМ МЕХАНИЗМА RED

Введение

Увеличение количества пользователей услугами связи, расширение перечня услуг, повышение требований к предоставляемым услугам определяет постоянную необходимость развивать и усовершенствовать технологии, положенные в основу функционирования современных телекоммуникационных систем (ТКС). Это приводит к поиску адекватного описания процессов, происходящих в ТКС, на математическом уровне, что позволяет проанализировать функционирование системы в тех или иных условиях (структуры сети, пропускные способности каналов передачи, функциональные особенности и т.д.) с помощью имитационного моделирования.

Надежная работа ТКС организуется за счет протоколов и механизмов, которые формируют реакцию приемопередающих устройств на искажения и/или потерю данных во время передачи. Например, логика работы TCP, наиболее распространенного протокола транспортного уровня, ориентирована на передачу потока сегментов данных с максимальной интенсивностью при условии сохранения приемлемого уровня потерь, позволяющего обеспечить их компенсацию повторными передачами. Так как при использовании современных высокоскоростных каналов связи практически не возникают потери и искажения, то причиной потерь может быть отбрасывание пакетов на промежуточных узлах с ограниченными размерами буферного пространства и объема памяти. Предотвратить экстремальные значения интенсивности потоков в TCP-сеансах можно за счет введения дополнительных механизмов управления потоком на узлах связи, наибольшее распространение среди которых получил механизм RED (*Random Early Detection*). Основная идея такого управления заключается в превентивном обнаружении признаков перегрузки и последующем увеличении количества отброшенных пакетов. В результате фактически предотвращается переполнение буфера и инициализируется механизм снижения скорости передачи в TCP.

Таким образом, математическое описание особенностей совместной работы протокола TCP и механизма RED позволит повысить их согласованность и оптимизировать работу ТКС. Аналитический подход к описанию процесса управления потоком в рамках TCP и RED дает возможность проанализировать производительность сети и выявить возможные явления неустойчивой работы ТКС.

Анализ особенностей реализации протокола TCP различных версий

Поскольку на данном этапе существует множество реализаций протокола TCP, стоит упомянуть основные задачи, которые решают различные версии данного протокола [1–4]. Регулирование трафика в TCP подразумевает существование двух независимых процессов: *контроль доставки*, управляемый получателем с помощью параметра *window*, и *контроль перегрузки*, управляемый отправителем с помощью окна перегрузки (переполнения) *swnd* (*congestion window*) и порога *ssthresh* (*slow start threshold*). Первый процесс отслеживает заполнение входного буфера получателя, второй – регистрирует перегрузку канала, а также связанные с этим потери и понижает уровень трафика.

Общие принципы работы протокола TCP состоят в следующем [1, 2]:

1. При инициализации соединения переменная *swnd* устанавливается равной размеру одного сегмента.

2. При возникновении перегрузки (что обнаруживается по таймауту или по трем входящим дубликатам одной и той же квитанции) переменной *ssthresh* присваивается значение, равное половине ширины ограничивающего окна (меньшего из двух: окна *swnd* или окна приема, предоставленного получателем), но не менее удвоенного размера сегмента. В качестве дополнительной меры, когда перегрузка проявляется в таймауте, переменной *swnd* присваивается значение, равное размеру одного сегмента (при этом передающая сторона автоматически переходит в режим начального разгона).

3. При поступлении новых квитанций от получателя на принятые им данные значение $cwnd$ каждый раз увеличивается, но величина приращения вычисляется по-разному – в зависимости от того, действует в этот момент режим *slow start* (медленного старта, начального разгона) или режим *congestion avoidance* (исключения перегрузки, подавления перегрузки). Если размер $cwnd$ не превышает порогового значения $ssthresh$, то осуществляется режим *slow start*; в противном случае вступает в силу режим *congestion avoidance*.

В режиме *slow start* первичное значение $cwnd$ равно размеру одного сегмента MSS (*Maximum Segment Size*) и увеличивается на один MSS при получении подтверждения (ACK). В режиме *congestion avoidance* при получении каждого ACK текущее значение $cwnd$ увеличивается на приращение, равное $\frac{MSS}{8} + \frac{MSS \cdot MSS}{cwnd(t)}$.

На рис. 1 показаны графики изменения окна перегрузки для версий Reno и Tahoe протокола TCP. Таким образом, в случае реализации версии TCP Tahoe при получении отправителем подтверждения доставки ACK текущее значение окна перегрузки $cwnd$ рассчитывается следующим образом [1, 2]:

$$cwnd(t + t_1) = \begin{cases} \text{режим } slow\ start : \\ cwnd(t) + MSS, \text{ if } cwnd(t) < ssthresh(t); \\ \text{режим } congestion\ avoidance : \\ cwnd(t) + \frac{MSS}{8} + \frac{MSS \cdot MSS}{cwnd(t)}, \text{ if } cwnd(t) \geq ssthresh(t). \end{cases} \quad (1)$$

Индикатором потери сегмента для отправителя являются два события: истечение таймаута (*Retransmission Timeout, RTO*) и приход трех дублирующих ACK-подтверждений. В рамках классического алгоритма TCP Tahoe оба случая приводят к запуску режима *slow start*, который предполагает установку окна перегрузки равным MSS . Более поздняя версия, TCP Reno, предусматривает при получении трех дублирующих подтверждений следующий алгоритм обновления $cwnd$ и $ssthresh$:

$$\begin{aligned} ssthresh(t + t_1) &= \frac{cwnd(t)}{2}; \\ cwnd(t + t_1) &= ssthresh(t + t_1). \end{aligned} \quad (2)$$

Далее осуществляется процедура повторной передачи потерянного сегмента вне ожидания таймаута (*fast retransmit*) и процедура быстрого восстановления без режима начального разгона (*fast recovery*) в соответствии со вторым режимом в формуле (1).

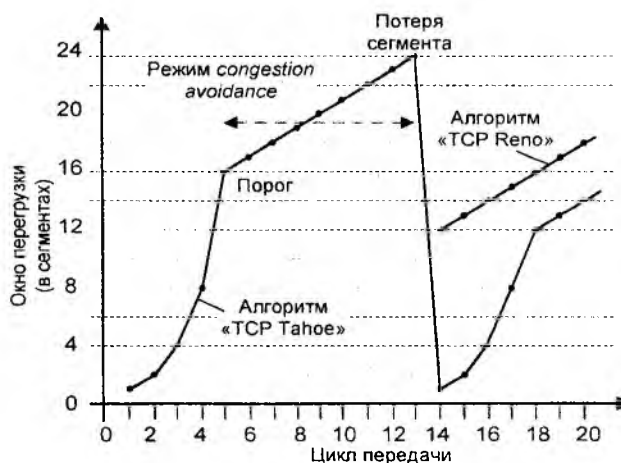


Рис. 1

Существует множество модификаций приведенного алгоритма (RFC 2018, RFC 2582). Так, например, в алгоритме Вегаса предпринимается попытка избежать перегрузок и одновременно обеспечить высокую скорость передачи [3, 4]. Основными идеями алгоритма Вегаса являются обнаружение перегрузок в маршрутизаторах на пути соединения до появления потерь пакетов и линейное снижение скорости передачи при перегрузках. Прогнозирование перегрузок осуществляется путем анализа времени оборота: чем больше это время, тем более значительной является перегрузка в маршрутизаторах.

Модель TCP-сеанса при управлении длиной очереди с использованием механизма RED

Наиболее точно оценить производительность функционирования ТКС можно лишь располагая адекватным математическим описанием – моделью того или иного сетевого процесса информационного обмена или управления, учитывающей стохастическую природу и динамику изменения состояния сети. Рассмотрим простейший фрагмент IP-сети, состоящий из двух сетевых узлов, которые при установке и поддержке TCP-сеанса используют механизм RED [5]. Узел-отправитель узел формирует поток передаваемых данных в виде набора последовательностей сегментов данных. Размер отдельной последовательности называется окном перегрузки $cwnd$ и, в общем случае, зависит от времени $cwnd(t)$.

Скорость передачи данных $\lambda(t)$ (скорость TCP-потока) определяется соотношением

$$\lambda(t) = \frac{cwnd(t)}{RTT}, \quad (3)$$

где RTT – время обмена данными (*RoundTrip Time*).

Тогда за интервал времени Δt число прибывших сегментов данных

$$\lambda(t) \cdot \Delta t = \frac{cwnd(t) \cdot \Delta t}{RTT}. \quad (4)$$

При распространении одного сегмента из последовательности длиной $cwnd$ возможны два варианта:

- потеря сегмента с вероятностью P ;
- успешная доставка сегмента получателю с вероятностью $(1 - P)$.

При этом, если обмен данными происходит в соответствии с алгоритмом TCP Tahoe и управлении интенсивностью передачи с помощью механизма RED путем внесения потерь, то после прохождения одного сегмента действующее изменение размера окна $\Delta cwnd$ определяется выражениями:

$$\Delta cwnd(t) = \begin{cases} \text{режим } slow\ start : \\ (1 - P(t)) \cdot cwnd(t) - P(t) \cdot (cwnd(t) - MSS); \\ \text{режим } congestion\ avoidance : \\ (1 - P(t)) \cdot \left(\frac{MSS}{8} + \frac{MSS \cdot MSS}{cwnd(t)} \right) - P(t) \cdot (cwnd(t) - MSS). \end{cases} \quad (5)$$

Если же обмен данными происходит в соответствии с алгоритмом Reno, который используется в большинстве современных реализаций TCP, также с использованием механизма RED, то действующее изменение размера окна $\Delta cwnd$ при обнаружении потери вследствие истечения таймаута вычисляется в соответствии с выражениями (5). Если же потеря была обнаружена вследствие получения трех дублирующих ACK-подтверждений выражения (5) принимает вид

$$\Delta cwnd(t) = \begin{cases} \text{режим slow start :} \\ (1 - P(t)) \cdot cwnd(t) - P(t) \cdot \frac{cwnd(t)}{2}; \\ \text{режим congestion avoidance :} \\ (1 - P(t)) \cdot \left(\frac{MSS}{8} + \frac{MSS \cdot MSS}{cwnd(t)} \right) - P(t) \cdot \frac{cwnd(t)}{2}. \end{cases} \quad (6)$$

Учитывая, что $\Delta cwnd(t) = cwnd(t + \Delta t) - cwnd(t)$, определим среднее изменение размера окна перегрузки в течение времени Δt , умножив $\Delta cwnd(t)$ на число сегментов, пришедших за этот интервал $\frac{cwnd(t) \cdot \Delta t}{RTT}$, для варианта реализации версии TCP Tahoe (т.е. только для выражений (5)):

$$\Delta cwnd(t) = \begin{cases} \text{режим slow start :} \\ \left[(1 - P(t)) \cdot cwnd(t) - P(t) \cdot (cwnd(t) - MSS) \right] \cdot \frac{cwnd(t) \cdot \Delta t}{RTT}; \\ \text{режим congestion avoidance :} \\ \left[(1 - P(t)) \cdot \left(\frac{MSS}{8} + \frac{MSS \cdot MSS}{cwnd(t)} \right) - P(t) \cdot (cwnd(t) - MSS) \right] \cdot \frac{cwnd(t) \cdot \Delta t}{RTT}. \end{cases} \quad (7)$$

Подставив (7) в (3), получим:

$$\Delta \lambda(t) = \begin{cases} \text{режим slow start :} \\ \left[(1 - P(t)) \cdot \frac{MSS}{RTT} \cdot \lambda(t) - P(t) \cdot (\lambda(t))^2 + P(t) \cdot MSS \cdot \lambda(t) \right] \cdot \Delta t; \\ \text{режим congestion avoidance :} \\ \left[(1 - P(t)) \cdot \left(\frac{MSS}{8 \cdot RTT} \cdot \lambda(t) + \frac{MSS \cdot MSS}{RTT^2} \right) - P(t) \cdot (\lambda(t))^2 + P(t) \cdot MSS \cdot \lambda(t) \right] \cdot \Delta t. \end{cases} \quad (8)$$

При $\Delta t \rightarrow 0$ получим дифференциальное уравнение для изменения интенсивности TCP-потока для различных режимов:

$$\frac{d\lambda(t)}{dt} = \begin{cases} \text{режим slow start :} \\ (1 - P(t)) \cdot \frac{MSS}{RTT} \cdot \lambda(t) - P(t) \cdot (\lambda(t))^2 + P(t) \cdot MSS \cdot \lambda(t); \\ \text{режим congestion avoidance :} \\ (1 - P(t)) \cdot \left(\frac{MSS}{8 \cdot RTT} \cdot \lambda(t) + \frac{MSS \cdot MSS}{RTT^2} \right) - P(t) \cdot (\lambda(t))^2 + P(t) \cdot MSS \cdot \lambda(t). \end{cases} \quad (9)$$

Анализ математической модели TCP-сеанса передачи данных

Для проверки адекватности математической модели TCP-сеанса передачи данных, которая описывается дифференциальными уравнениями (9), произведем вычисление интенсивности TCP-потока $\lambda(t)$ для технологии FastEthernet (пропускная способность $B = 12.5$ Мбайт/с). При этом предполагается, что окно приема на узле-получателе равно 64 кбайт и размер сегмента данных $MSS = 1460$ байт. Вероятность блокировки (отбрасывания) пакетов $P(t)$ согласно механизму RED определяется равенством [5]:

$$P(t) = \frac{N(t) - N_{\min}}{N_{\max} - N_{\min}}, \quad (10)$$

где $N(t)$ – текущий средний размер очереди; N_{\min} , N_{\max} – соответственно минимальный и максимальный средний размер очереди.

Средний размер очереди на сетевом узле можно рассчитать исходя из формулы, приведенной в теории массового обслуживания [6, 7]:

$$N(t) = \frac{\lambda(t)}{B - \lambda(t)}, \quad (11)$$

B – пропускная способность канала связи (байт/с).

Графически решения уравнения (9) с учетом выражения (10) и (11) при различных значениях RTT изображено на рис. 2.

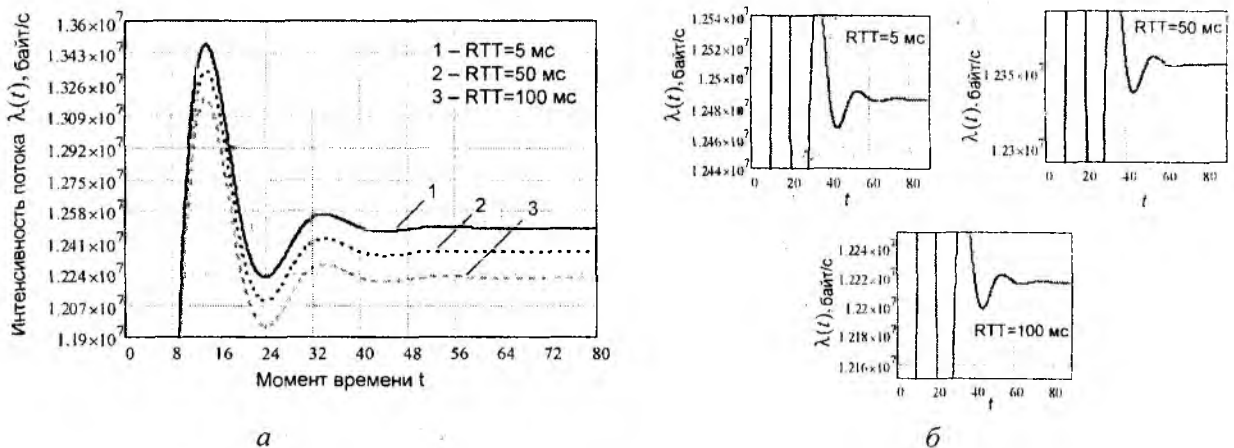


Рис. 2

Из рис. 2 видно, что, во-первых, изменение скорости передачи $\lambda(t)$ носит колебательный характер и в некоторый момент времени устанавливается на определенном значении, что соответствует реальному процессу информационного обмена при работе протокола TCP. Во-вторых, с уменьшением времени оборота сегмента RTT (что говорит, например, об улучшении канала либо свободных канальных и буферных ресурсах) интенсивность, которую достигает TCP-поток в установившемся режиме, возрастает и стремится к значению пропускной способности канала передачи.

Выводы

В работе предложен подход к математическому описанию процесса информационного обмена, организованного согласно протоколу транспортного уровня TCP. Математическая модель представлена нелинейным дифференциальным уравнением изменения интенсивности TCP-потока, учитывающая динамику изменения режимов передачи данных в соответствии с различными версиями протокола TCP. В рамках предложенной модели формализована зависимость изменения окна перегрузки от результата работы механизма превентивного управления длиной очереди RED.

Исследования предложенной модели показали, что она достаточно адекватно описывает работу TCP-сеанса и может быть использована для анализа производительности ТКС в пределах отдельных соединений при изменении таких важных сетевых параметров как время оборота сегмента RTT , пропускной способности канала, модели отказов алгоритма RED и максимального объема буфера. Данную модель целесообразно использовать при оптимизации параметров TCP-сеанса, а также для исследования системы на устойчивость в условиях колебаний сетевых параметров.

Список литературы: 1. *W. Stevens* TCP Slow Start, Congestion Avoidance, Fast Retransmit, RFC2001, Jan. 1997. 2. *Стивенс У.Р.* Протоколы TCP/IP. Практическое руководство. Пер. с англ. и коммент. А.Ю. Глебовского: СПб.: «Невский диалект»-«БХВ-Петербург», 2003. 672 с. 3. *Джеймс Ф. Куроуз, Кит В. Росс* Компьютерные сети: многоуровневая архитектура интернета. 2-е изд. СПб.: Питер, 2004. 765 с. 4. *Сидни Фейт* TCP/IP. Архитектура, протоколы, реализация (включая IP версии 6 и IP Security). 2-е изд. М.: «Лори», 2000. С. 424. 5. *Вегешна Ш.* Качество обслуживания в сетях IP: Пер. с англ. М.: Изд. дом «Вильямс», 2003. 368 с. 6. *Вишневецкий В.М.* Теоретические основы проектирования компьютерных сетей. М.: Техносфера, 2003. 512 с. 7. *Крылов В.В., Самохвалова С.С.* Теория телетрафика и ее приложения. СПб.: БХВ-Петербург, 2005. 288 с.

*Харьковский национальный
университет радиоэлектроники*

Поступила в редколлегию 05.09.2009