ORES

Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

РАЗМЕР ОКНА И ПРОПУСКНАЯ СПОСОБНОСТЬ В ПРОТОКОЛЕ TCP

Розанова Ольга Сергеевна,
Студент. ФГБОУ высшего образования «Санкт-Петербургский государственный экономический университет» (СПбГЭУ). Юридический факультет. Специальность юриспруденция. e-mail: tatyana.rozanova@ya.ru

Пиличев Павел Николаевич,
Студент. Северный (арктический) федеральный университет.
Высшая школа экономики управления и права. Специальность юриспруденция. e-mail: pavel2201@mail.ru

Кислянская Эллина Сергеевна,
Магистрант. Северный (арктический) федеральный университет.
Кафедра физической культуры и спорта. e-mail: Ellina2000@bk.ru

Аннотация

Статья посвящена исследованию вопроса влияния размера окна TCP на пропускную способность сети. В статье рассмотрен процессы установления соединения TCP, передачи данных и изменение размера окна в процессе передачи данных. В статье выполнен расчет пропускной сети в зависимости от размера окна передачи данных.

Ключевые слова: TCP, размер окна TCP, управление окнами, медленный старт, пропускная способность сети, пропускная способность TCP.

WINDOW SIZE AND BANDWIDTH IN TCP PROTOCOL

Olga S. Rozanova,
Student. St. Petersburg State University of Economics, Saint-Petersburg.
Law faculty. Specialty law. e-mail: tatyana.rozanova@ya.ru

Pavel N. Pilichev,
Student. Northern (Arctic) Federal University named after M.V. Lomonosov. Higher School of Economics, Management and Law. Specialty law. e-mail: pavel2201@mail.ru

Ellina S. Kislyanskaya,
Graduate student. Northern (Arctic) Federal University named after M.V. Lomonosov.
Department of Physical Culture and Sports. e-mail: Ellina2000@bk.ru

ABSTRACT

The article is devoted to the study of the effect of the TCP window size on network bandwidth. The article discusses the processes of establishing a TCP connection, data transmission and window resizing during data transmission. The article calculates the network bandwidth depending on the size of the data transmission window.

Keywords: TCP, TCP window size, window management, slow start, network bandwidth, TCP bandwidth.

TCP (Transmission Control Protocol) является одним из основных протоколов транспортного уровня, обеспечивающим надежную передачу данных между устройствами в компьютерной сети. Размер окна TCP - один из важнейших параметров этого протокола, оказывающий значительное влияние на скорость и эффективность передачи данных. В этой статье мы рассмотрим, почему размер окна TCP имеет такое значение, какие проблемы могут возникнуть при неправильной настройке этого параметра, и какие стратегии оптимизации могут помочь повысить производительность сети.

TCP - это протокол с установкой соединения, который отслеживает, сколько данных было передано. Отправитель передает некоторый объем данных, а получатель должен подтвердить факт получения этих данных. Если подтверждение не придет вовремя, отправитель повторно передаст данные.

TCP использует “управление окнами”, что означает, что отправитель отправляет один или несколько сегментов данных, а получатель подтверждает сколько сегментов получено - один или все сегменты. Для приема данных компьютеры используют буфер приема, в котором временно хранятся данные, прежде чем приложение сможет их обработать.

При отправке подтверждения получатель сообщает отправителю, сколько данных он может передать. Это и называется размером окна. По сути, размер окна указывает на размер принимающего буфера. Для указания размера окна в заголовке TCP предусмотрено поле «Размер окна».

Заголовок TCP содержит поля [4, с. 37], показанные на Рисунке 1 (описание полей выходит за рамки данной статьи).

<table>
<tr><td colspan="5">Порт отправителя</td><td colspan="5">Порт получателя</td></tr>
<tr><td colspan="10">Порядковый номер в сегменте</td></tr>
<tr><td colspan="10">Номер подтверждения</td></tr>
<tr><td>Длина заголовка</td><td>Резерв</td><td>NS</td><td>CWR</td><td>ECE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Размер окна</td></tr>
<tr><td colspan="5">Контрольная сумма</td><td colspan="5">Указатель срочности</td></tr>
<tr><td colspan="10">Дополнительные опции и заполнение</td></tr>
</table>

Рис.1 Заголовок TCP

Обычно TCP-соединение начинается с небольшого размера окна передачи данных, после получения каждого успешного подтверждения, размер окна будет увеличиваться [2, с. 288]. Рассмотрим как меняется размер окна. Компьютер PC1 отправляет один сегмент,

а компьютер PC2 отправляет подтверждение в ответ. На рисунке 2 показан размер окна 1 сегмент.

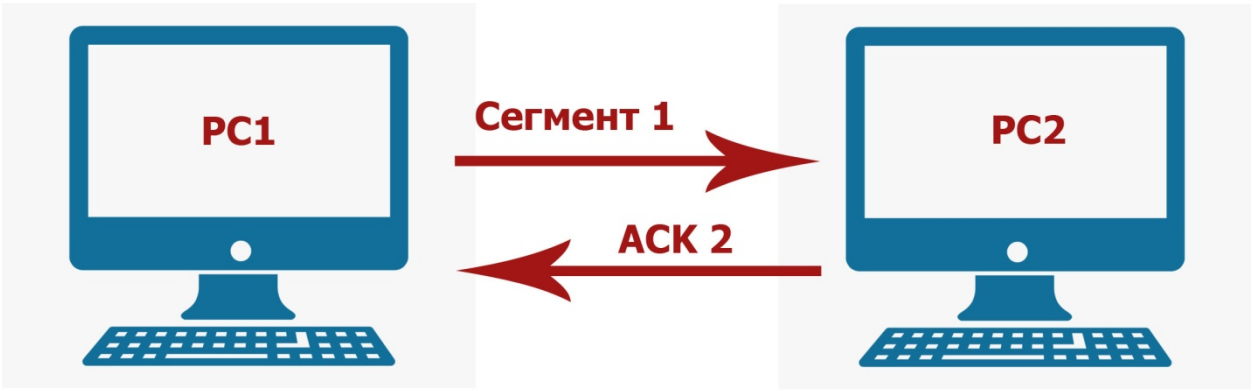

Рис.2 Размер окна в 1 сегмент

Так как подтверждение прошло успешно, размер окна увеличится. Компьютер PC1 теперь отправляет два сегмента, а компьютер PC2 возвращает одно подтверждение. На рисунке 3 показан размер окна 2 сегмента.

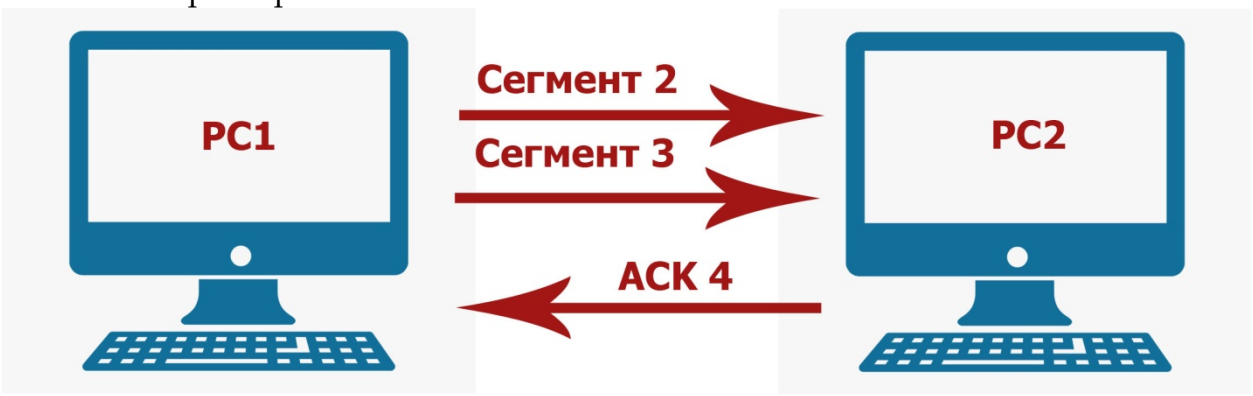

Рис.3 Размер окна в 2 сегмента

Так как подтверждение получено, размер окна снова увеличится. Теперь компьютер PC1 отправляет четыре сегмента, а компьютер PC2 отвечает одним подтверждением. На рисунке 4 показан размер окна 4 сегмента.

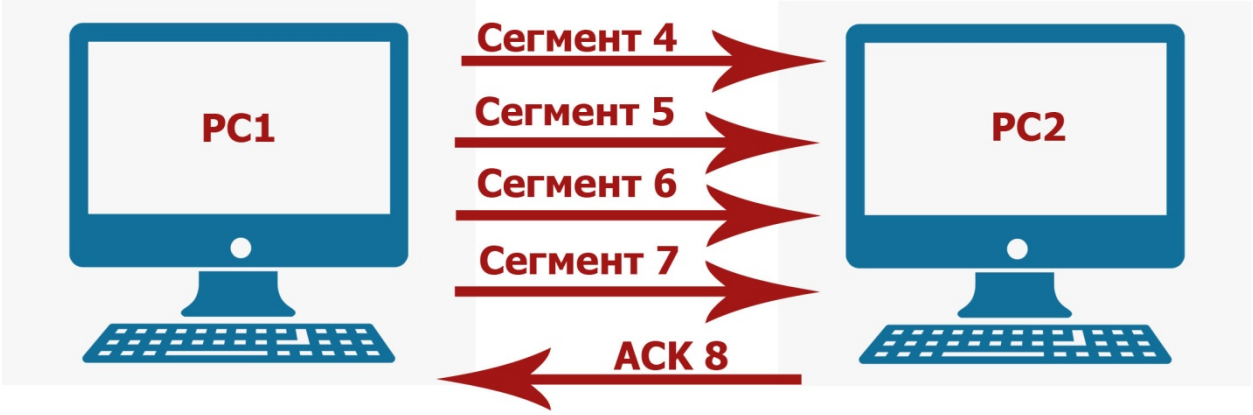

Рис.4 Размер окна в 4 сегмента

В приведенном выше примере размер окна продолжает увеличиваться до тех пор, пока получатель не отправит подтверждения для всех отправленных сегментов или когда размер окна достигнет определенного максимального предела. Если получатель не отправляет подтверждение в течение определенного периода времени (называемого временем обратной передачи), размер окна будет уменьшен.

Когда интерфейс получателя перегружен, IP-пакеты могут отбрасываться. В TCP есть ряд механизмов, которые занимаются контролем перегрузки. Один из них называется медленный старт.

Перегрузка возникает, когда интерфейсу приходится принимать больше данных, чем он может обработать. Буфер приема достигнет предела, и пакеты будут отброшены.

При медленном старте TCP размер окна сначала будет расти экспоненциально (размер окна удваивается), но как только пакет отбрасывается, размер окна уменьшается до одного сегмента [2, с. 290]. Затем оно снова будет расти экспоненциально, пока размер окна не станет вдвое меньше, чем был на момент возникновения перегрузки. После этого размер окна будет расти линейно, а не экспоненциально.

Когда интерфейс перегружен, возможна ситуация, что одновременно во всех TCP-соединениях произойдет медленный старт TCP. Пакеты одновременно будут отброшены, и тогда все TCP-соединения установят разом небольшой размер окна. Это называется глобальной синхронизацией TCP. На рисунке 5 показано, выглядит глобальная синхронизация TCP.

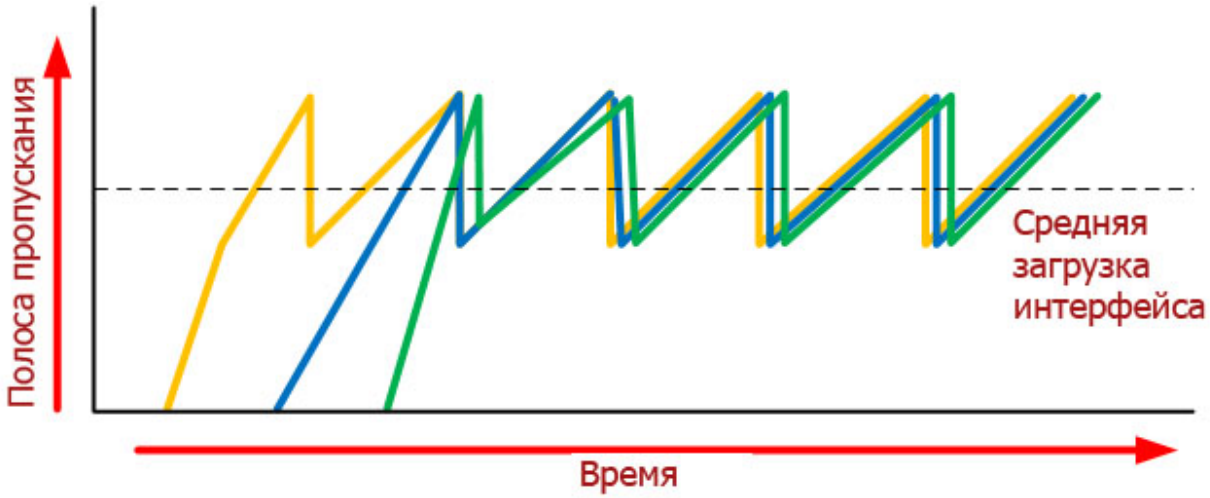

Рис.5 Глобальная синхронизация TCP

Оранжевая, синяя и зеленая линии представляют собой три разных TCP-соединения. Эти TCP-соединения запускаются в разное время, через некоторое время интерфейс перегружается, и пакеты всех TCP-соединений одновременно отбрасываются. В этот момент размер окна всех этих TCP-соединений уменьшится до одного сегмента, и как только перегрузка интерфейса исчезнет, размеры всех окон снова будут увеличиваться [2, с. 289].

Затем интерфейс снова становится перегруженным, размер окна возвращается к одному сегменту, и история повторяется. Результатом этого является то, что интерфейс загружается неравномерно, скачками и не используется вся доступная пропускная способность интерфейса. Пунктирная линия показывает, то, что средняя загрузка интерфейса не очень высока.

Чтобы предотвратить такую ситуацию (глобальную синхронизацию), используется RED (раннее случайное обнаружение). Это функция, которая удаляет “случайные” пакеты из потоков TCP на основе количества пакетов в очереди и маркировки

пакетов TOS (качества сервиса). При таком способе пакеты отбрасываются случайным образом до заполнения очереди, и глобальной синхронизации не происходит.

На рисунке 6 показано как будет выглядеть график при использовании механизма RED.

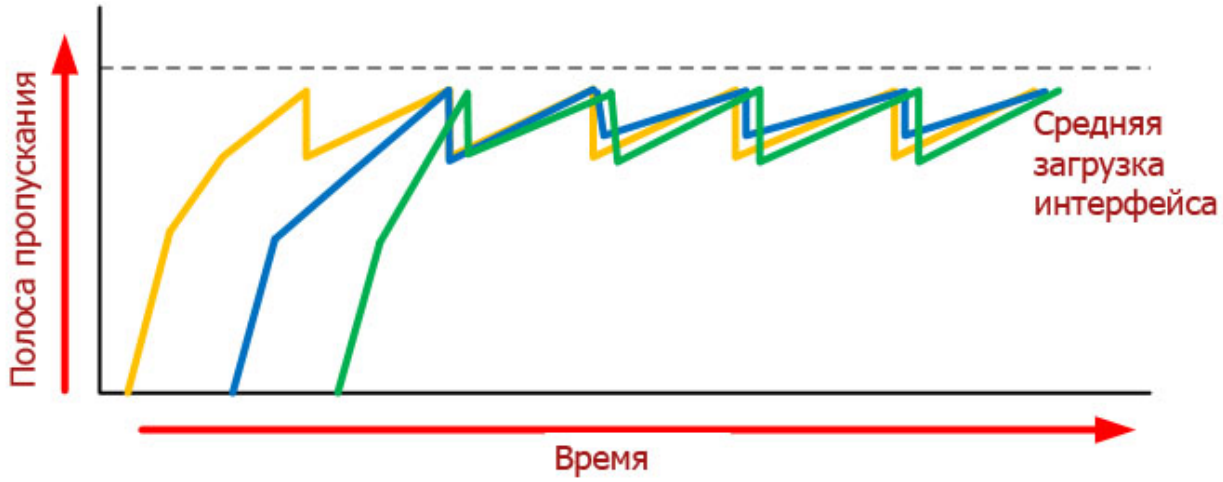

Рис.6 График нагрузки при использовании RED

Видим, что средняя загрузка интерфейса вырастет.

Разберем на реальном примере, как меняется размер окна при передаче данных. Для этого выполним захват пакетов в Wireshark

Для проверки размера окна TCP будем использовать два устройства, показанные на рисунке 7.

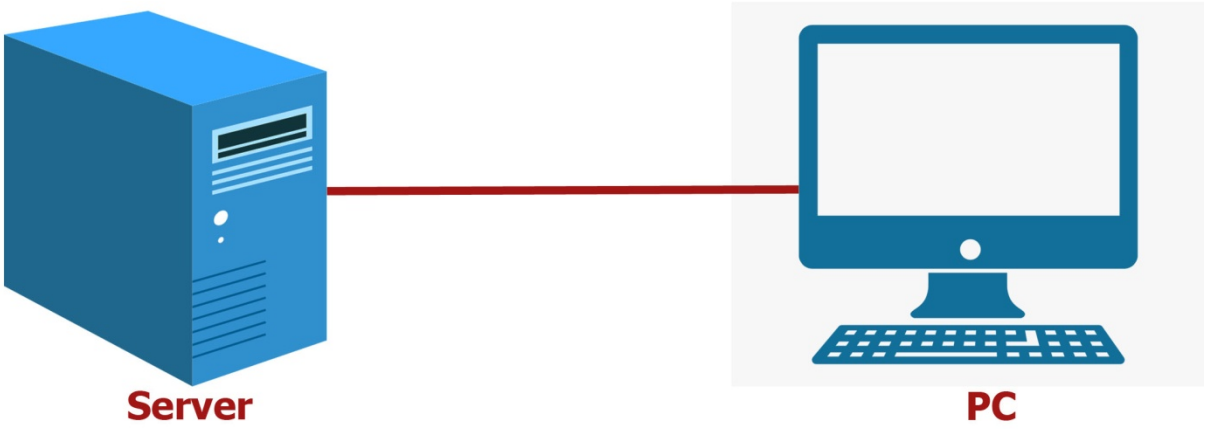

Рис.7 Сеть для проверки размера окна TCP

Устройство с левой стороны представляет собой современный сервер с гигабитным интерфейсом. С правой стороны установлен старый компьютер, с интерфейсом FastEthernet, с ограниченными ресурсами. Для проверки работы окна скопируем файл большого размера через протокол SSH с сервера на старый компьютер. Добьемся чтобы интерфейс компьютера был перегружен.

На рисунке 8 отобразим график того, как меняется размер окна в Wireshark.

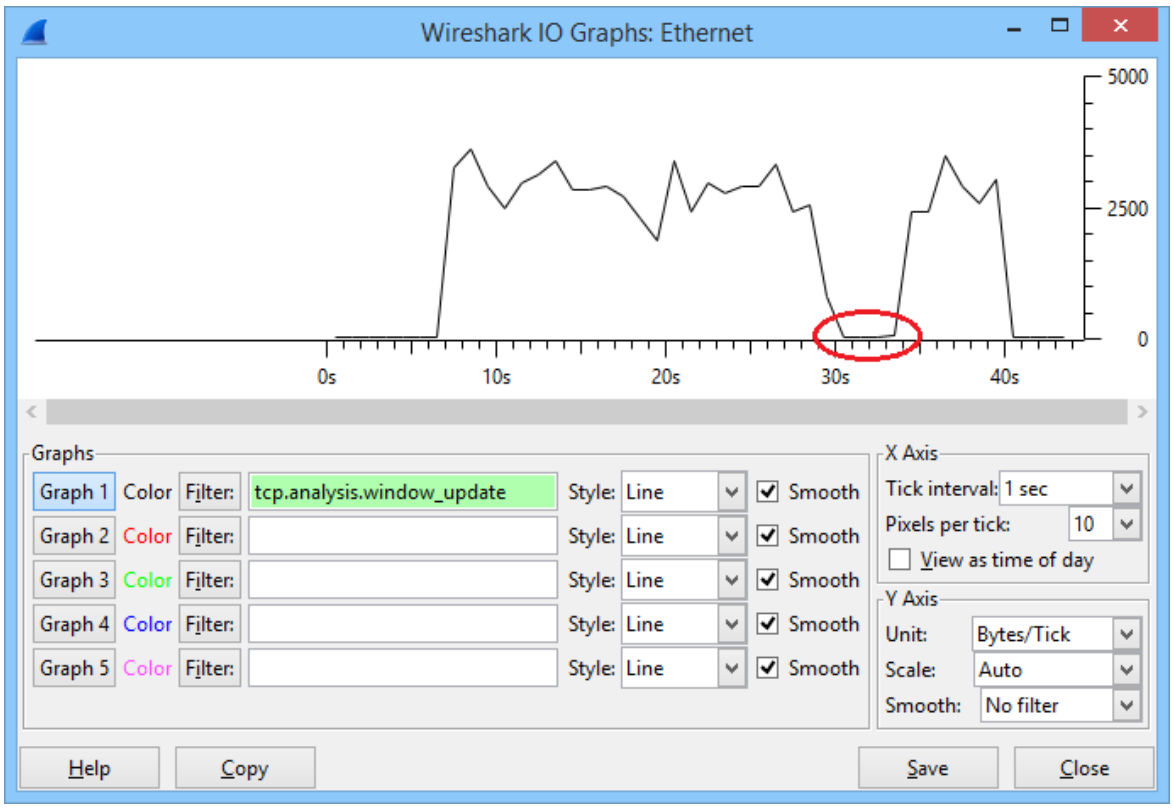

Рис.8 Размер окна в Wireshark

При анализе графика видно,что передача файла началась примерно через 6 секунд, и размер окна быстро вырос. Далее размер окна колебался, немного увеличивался и уменьшался, затем примерно через 30 секунд размер окна стал нулевым и передача файла полностью прекратилось. Далее через несколько секунд размер окна снова быстро вырос, и передача файла была возобновлена и выполнена.

Рассмотрим подробнее процесс передачи файла.

Вначале был выполнен процесс трехстороннего рукопожатия [3, с. 169], на рисунке 9 показан первый этап при котором сервер установливает связь с компьютером и отправляет запрос с установленным флагом SYN. После этого Server переходит в состояние SYN SENT. PC остается в состоянии LISTEN.

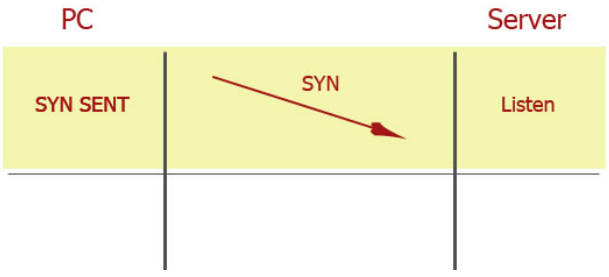

Рис.9 Первый этап трехстороннего рукопожатия

На рисунке 10 показано какие данные устанавливаются в заголовке TCP.

Порт отправителя											Порт получателя
Порядковый номер в сегменте ISN = 0											
Номер подтверждения ACK SN = 0											
Длина заголовка	Резерв	N	C	E	U	A	P	R	S	F	Размер окна

Контрольная сумма	Указатель срочности
Дополнительные опции и заполнение	

Рис.10 Заголовок TCP при первом этапе трехстороннего рукопожатия

На рисунке 11 показан второй этап, при котором компьютер принимает запрос SYN, формирует ответ и в ответе устанавливает 2 флага: SYN и ACK.

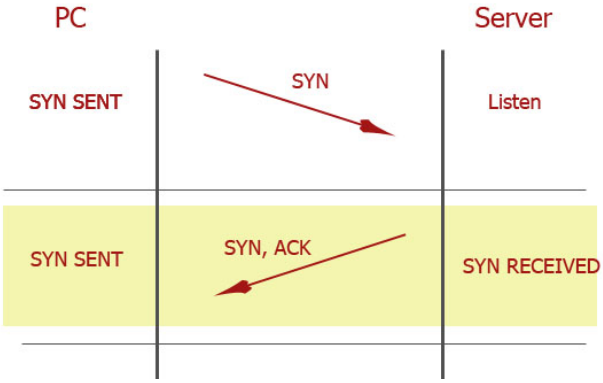

Рис.11 Второй этап трехстороннего рукопожатия

На рисунке 12 показано показано какие данные устанавливаются в заголовке TCP.

Порт отправителя											Порт получателя
Порядковый номер в сегменте ISN = 200											
Номер подтверждения ACK SN = 2											
Длина заголовка	Резерв										Размер окна
		N	C	E	U	A	P	R	S	F	
Контрольная сумма											Указатель срочности
Дополнительные опции и заполнение											

Рис.12 Заголовок TCP при втором этапе трехстороннего рукопожатия

На рисунке 13 показан третий этап, при котором сервер примет ответ в котором установлено 2 флага SYN и ACK и отправит в ответ третий пакет с флагом ACK.

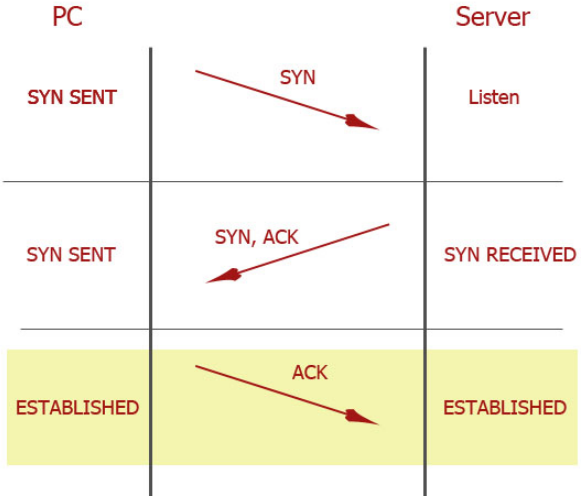

Рис.13 Третий этап трехстороннего рукопожатия

На рисунке 14 показано показано какие данные устанавливаются в заголовке TCP.

Порт отправителя											Порт получателя
Порядковый номер в сегменте ISN = 2											
Номер подтверждения ACK SN = 201											
Длина заголовка	Резерв	N	C	E	U	A	P	R	S	F	Размер окна
Контрольная сумма											Указатель срочности
Дополнительные опции и заполнение											

Рис.14 Заголовок TCP при третьем этапе трехстороннего рукопожатия

На рисунке 15 рассмотрим процесс трехстороннего рукопожатия в сетевом анализаторе Wireshark.

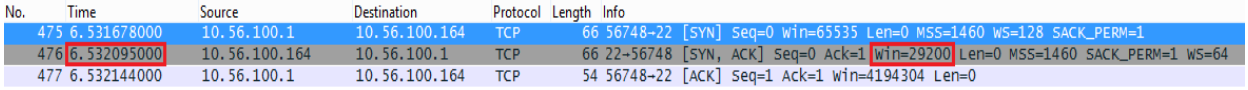

Рис.15 Процесс трехстороннего рукопожатия в Wireshark

IP адрес сервера - 10.56.100.1, IP адрес компьютера - 10.56.100.164. В ответе от компьютера видно, что он предлагает использовать размер окна 29200. Сервер предлагает использовать размер окна 8388480 (win = 65535 * ws = 128), но этот размер не критичен, т.к. данные отправляются с сервера на компьютер.

После отправки нескольких пакетов размер окна компьютера выглядит так, как показано на рисунке 16.

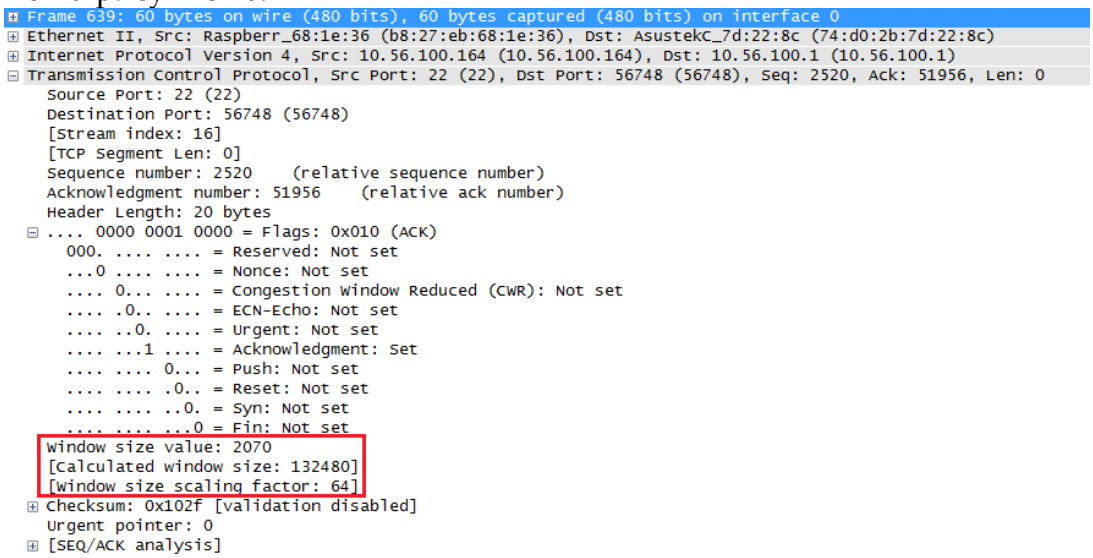

Рис.16 Размер окна в процессе передачи данных

Видно, что размер окна увеличился до 132480.

Примерно на 10-секунде размер окна уменьшился. На рисунке 17 видно, что буфер приема компьютера заполнился и он сообщает серверу, чтобы тот установил размер окна 26752.

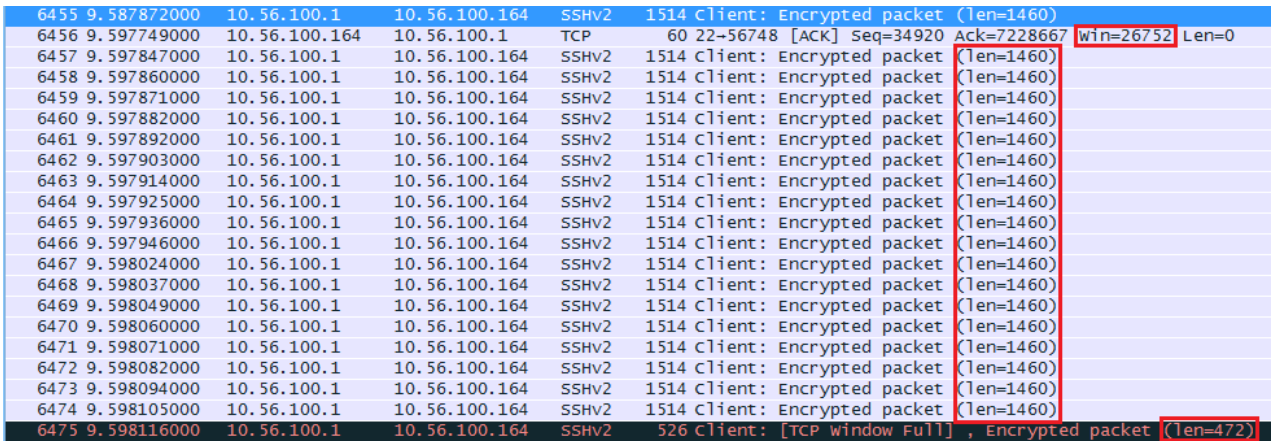

Рис.17 Заполнение буфера приема

Сервер отправляет 18 сегментов по 1460 байт и один сегмент по 472 байта (суммарно 26752 байта). Последний пакет подсвечен черным и в нем присутствует сообщение “TCP Window Full” (Окно TCP заполнено). Wireshark указывает, что буфер приема компьютера полностью заполнен.

Затем на 34-секунде компьютер отправляет на сервер подтверждение с размером окна 0, показанное на рисунке 18.

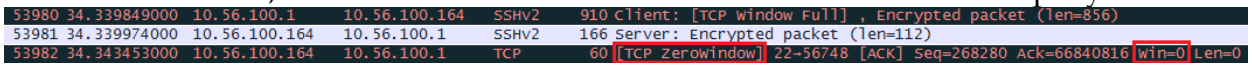

Рис.18 Нулевой размер окна

Пока размер окна будет оставаться равным 0 компьютер не сможет принимать какие-либо данные, и передача TCP будет приостановлена на некоторое время, до тех пор пока не освободится буфер приема. На рисунке 19 показан захваченный пакет с размером окна 0 в сетевом анализаторе Wireshark.

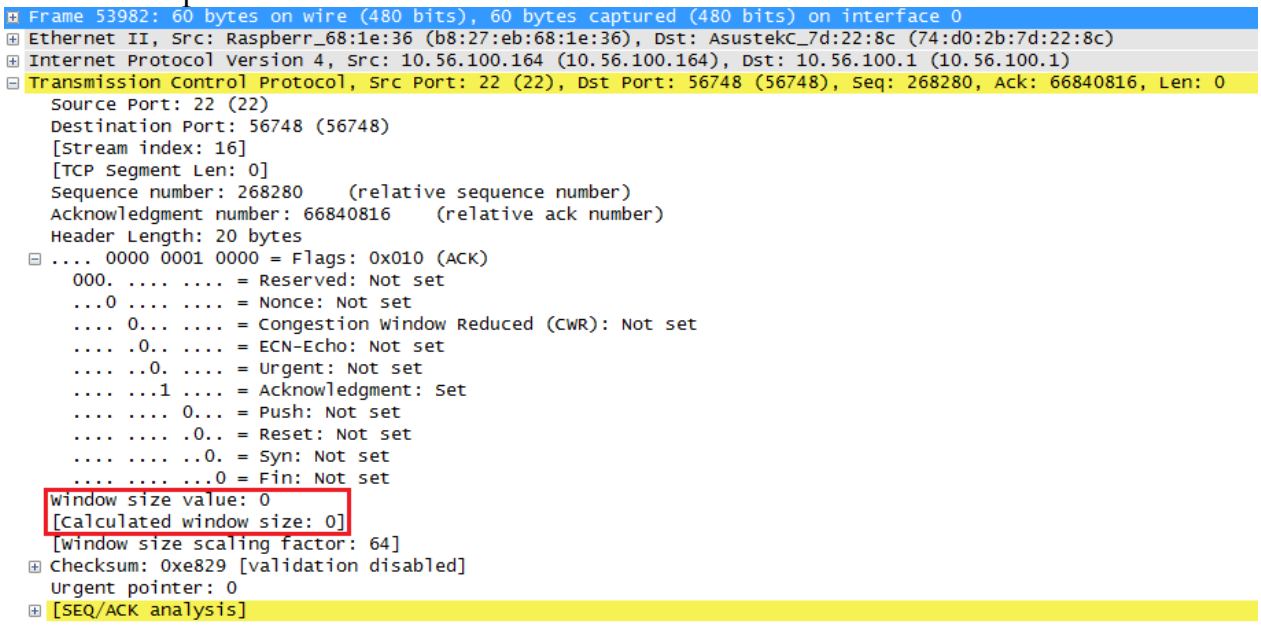

Рис.19 Нулевой размер окна в Wireshark

Как только освободится буфер приема, компьютер отправит подтверждение с новым размером окна, показанное на рисунке 20.

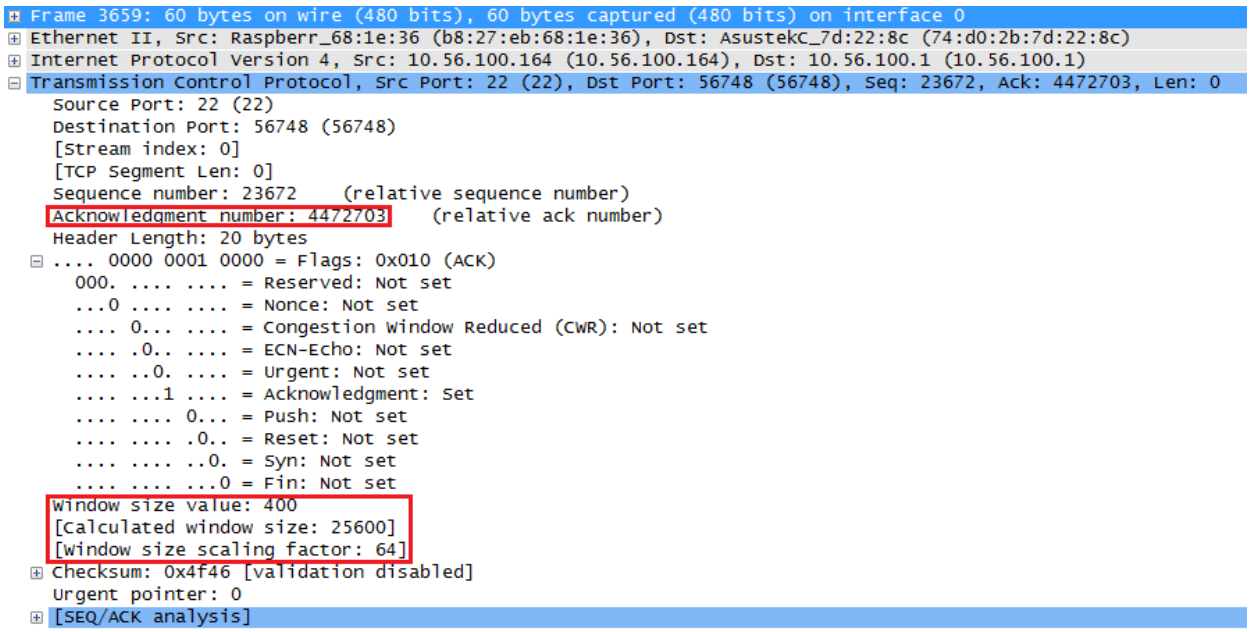

Рис.20 Размер окна после возобновления передачи в Wireshark

Размер окна будет установлен 25600 байт и затем будет увеличиваться. При дальнейшей передаче остальная часть файла была передана без сбоев и была завершена.

От размера окна зависит пропускная способность сети. Этот параметр демонстрирует, с какой скоростью данные способны передаваться по сети [1, с. 54]. Доступная пропускная способность сети представляет собой максимальную скорость передачи данных, которую способна поддерживать сеть. Обычно её измеряют в Мбит/с (мегабитах в секунду) или Гбит/с (гигабитах в секунду).

Пропускная способность сети служит теоретическим верхним пределом пропускной способности TCP и играет важнейшую роль в оптимизации передачи данных и обеспечении эффективности сетевых соединений.

Пропускная способность TCP относится к количеству данных, которые могут быть переданы по сети за определенный промежуток времени. Она измеряется в битах в секунду (бит/с) и зависит от различных факторов, включая пропускную способность сети, задержку, потерю пакетов и размер окна TCP.

Задержка представляет собой время, необходимое для прохождения пакета данных от источника к месту назначения. Обычно она измеряется в миллисекундах (мс). Более высокая задержка может оказывать влияние на пропускную способность TCP, особенно для приложений, требующих низкого времени отклика. Поэтому оптимизация задержки и управление ею играют важную роль в обеспечении эффективной пропускной способности TCP. Для снижения задержки можно применять различные методы, такие как оптимизация маршрутизации, использование более низколатентных сетевых технологий и оптимизация приложений.

Потеря пакетов - это ситуация, когда передаваемые пакеты данных не достигают своего назначения и отбрасываются во время передачи. Это может происходить из-за перегрузки сети или наличия ошибок в сетевой инфраструктуре. Потеря данных может серьезно сказаться на пропускной способности TCP, поскольку потерянные пакеты требуют повторной передачи, что отрицательно сказывается на общей производительности сети.

Расчет пропускной способности TCP выполняется по следующей формуле:

Пропускная способность (бит/с) = размер окна TCP (биты)/Время прохождения туда и обратно RTT (секунды)

Время прохождения туда и обратно (RTT) - это время, необходимое для прохождения небольшого пакета данных от отправителя к получателю и обратно. Оно включает в себя

как время, необходимое для передачи данных к получателю, так и время подтверждения возврата.

Пример:

Давайте рассмотрим TCP-соединение с размером окна TCP 64 000 бит и RTT 0,2 секунды.

Пропускная способность = 64 000 бит/0,2 секунды = 320 000 бит/с

Факторы, влияющие на пропускную способность TCP:

a. Произведение задержки полосы пропускания (BDP):

BDP - это максимальный объем данных, который может передаваться по сети в любой момент времени. Он рассчитывается путем умножения пропускной способности (в битах в секунду) на время передачи в оба конца (в секундах). Размер окна TCP должен быть установлен на значение, близкое к BDP, для достижения оптимальной пропускной способности.

b. Окно перегрузки (CWND):

CWND динамически настраивается TCP для управления объемом данных, которые могут быть отправлены до ожидания подтверждений. Это помогает избежать перегрузки сети. При небольшом CWND пропускная способность может быть ограничена, но она увеличивается по мере роста CWND.

c. Медленный старт TCP:

На начальном этапе TCP-соединения отправитель начинает с небольшого CWND и увеличивает его экспоненциально. Этот процесс, называемый медленным запуском, помогает избежать перегрузки сети слишком большим количеством данных одновременно.

Методы максимизации пропускной способности TCP

a. Увеличение размера окна TCP:

При увеличении размера окна TCP можно отправлять больше данных, не дожидаясь подтверждений. Однако слишком большой размер окна может привести к перегрузке и потере пакетов. Согласование размера окна с BDP имеет решающее значение для оптимизации пропускной способности.

b. Обнаружение MTU пути:

Определение максимальной единицы передачи по пути (MTU) помогает определить наибольший размер пакета, который может быть передан по сети без фрагментации. Использование оптимального размера MTU может снизить накладные расходы и повысить пропускную способность.

c. Механизмы разгрузки TCP (TOE):

TOE - это специализированные аппаратные компоненты, которые обрабатывают TCP, снимая нагрузку с основного центрального процессора. Это может повысить общую производительность системы и, в свою очередь, увеличить пропускную способность TCP.

d. Реализация качества обслуживания (QoS):

Механизмы QoS определяют приоритетность определенных типов трафика, гарантируя, что критически важные данные получают достаточные сетевые ресурсы. Это предотвращает перегрузку и потерю пакетов, что приводит к повышению пропускной способности для приложений с приоритетом.

Рассмотрим примеры вычислений пропускной способности TCP в различных сценариях:

Пример 1:

Высокая пропускная способность, подключение с низкой задержкой Рассмотрим сценарий, в котором есть высокоскоростное сетевое соединение с пропускной способностью 1 Гбит/с (1000 000000 бит/с) и низкой задержкой в 5 миллисекунд (0,005 секунды).

Рассчитаем пропускную способность TCP, используя формулу приведенную выше.

Предположим, что размер окна TCP установлен равным 1 048 576 битам (128 КБ).

Пропускная способность = 1 048 576 бит/0,005 секунды ≈ 209 715 200 бит/с или 209,7 Мбит/с

В этом примере высокая пропускная способность и низкая задержка приводят к значительной пропускной способности TCP, обеспечивая быструю передачу данных.

Пример 2:

Пропускная способность TCP с высокой потерей пакетов. Рассмотрим сценарий, при котором в сети наблюдается высокая потеря пакетов, что отрицательно влияет на пропускную способность TCP. Предположим, что пропускная способность 100 Мбит/с (100 000 000 бит/с), время перехода туда и обратно 50 миллисекунд (0,05 секунды) и размер окна TCP 32 768 бит (4 КБ). Из-за перегрузки сети 10% пакетов теряются во время передачи.

Эффективный размер окна TCP = размер окна TCP * (процент потери 1 пакета) Эффективный размер окна TCP = 32 768 бит * (1 – 0.10) = 29,491.2 биты (приблизительно 3,68 КБ).

Пропускная способность = 29 491,2 бит/0,05 секунды ≈ 589 824 бит/с или 0,59 Мбит /с.

Высокая потеря пакетов значительно снижает эффективный размер окна TCP и, следовательно, пропускную способность.

Пример 3:

Влияние размера окна TCP на пропускную способность Рассмотрим, как различные размеры окна TCP влияют на пропускную способность при времени перехода туда и обратно 10 мс и пропускной способности 100 Мбит/с.

a. Размер окна TCP = 64 000 бит (8 КБ)

Пропускная способность = 64 000 бит/0,01 секунды = 6 400 000 бит/с или 6,4 Мбит/с

b. Размер окна TCP = 128 000 бит (16 КБ)

Пропускная способность = 128 000 бит/0,01 секунды = 12 800 000 бит/с или 12,8 Мбит/с

c. Размер окна TCP = 256 000 бит (32 КБ)

Пропускная способность = 256 000 бит/0,01 секунды = 25 600 000 бит/с или 25,6 Мбит/с

d. Размер окна TCP = 512 000 бит (64 КБ)

Пропускная способность = 512 000 бит/0,01 секунды = 51 200 000 бит/с или 51,2 Мбит/с

В этом примере увеличение размера окна TCP приводит к линейному увеличению пропускной способности. Больший размер окна позволяет передавать больше данных до ожидания подтверждений, эффективно используя доступную полосу пропускания.

Заключение.

Мы рассмотрели, как TCP использует размер окна, чтобы сообщить отправителю, какой объем данных необходимо передать.

Размер окна TCP играет важнейшую роль в процессе передачи данных по сети и может значительно повлиять на производительность TCP-соединений. Правильная настройка этого параметра является необходимой для обеспечения эффективной передачи данных и оптимального использования доступной пропускной способности. С учетом значимости размера окна TCP для функционирования сетей, важно уделять особое внимание оптимизации этого параметра, чтобы улучшить производительность и стабильность сетевых соединений.

Расчет пропускной способности TCP имеет первостепенное значение для понимания и оптимизации сетевой производительности. Изучая различные примеры с различными условиями сети и настройками TCP, сетевые администраторы могут точно настроить свои системы для достижения максимально возможной пропускной способности. Повышение пропускной способности TCP приводит к более быстрой передаче данных, сокращению времени отклика и повышению общей эффективности сети.

Список литературы:

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Юбилейное издание / Олифер В.Г., Олифер Н.А. — СПб.: Питер, 2020. — 1008 с.
2. Мэтью Ногл, TCP/IP Иллюстрированный учебник. Москва, 2001. — 481 с.
3. Уэнделл Одом. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101, 2015. — 903 с.
4. Крейг Хант. TCP/IP. Сетевое администрирование, Санкт-Петербург – Москва, 2008. – 811 с.

References:

1. Olifer V.G., Olifer N.A. Computer networks. Principles, technologies, protocols: Anniversary edition / Olifer V.G., Olifer N.A. - St. Petersburg: Peter, 2020. - 1008 p.
2. Matthew Naugle, TCP/IP Illustrated textbook. Moscow, 2001. - 481 p.
3. Wendell Odom. Cisco Official Guide to Preparation for CCENT/CCNA ICND1 Certification Exams 100-101, 2015. - 903 pp.
4. Craig Hunt. TCP/IP. Network administration, St. Petersburg - Moscow, 2008. - 811 p.