Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчет по лабораторной работе №3**

**по курсу «Технологии компьютерный сетей»**

«Wireshark: TCP & UDP»

**Работу выполнил студент группы №5130901/10203:**

Алексеев Лев Сергеевич

Подпись:

**Работу принял преподаватель:**

Богач Наталья Владимировна

Подпись:

Санкт-Петербург

2024

**Wireshark: TCP**

В этой лабораторной подробно исследуем поведение знаменитого протокола TCP.

Для выполнения трассировки пакетов, передаваемых по TCP-соединению от компьютера к удаленному серверу, необходимо использовать Wireshark, чтобы начать исследование протокола TCP. Чтобы сделать это, обратимся к веб-странице, на которой сможем указать имя файла, хранящегося на компьютере (с ASCII-текстом книги «Алиса в стране чудес»), и далее передать файл веб-серверу, используя метод POST протокола HTTP.

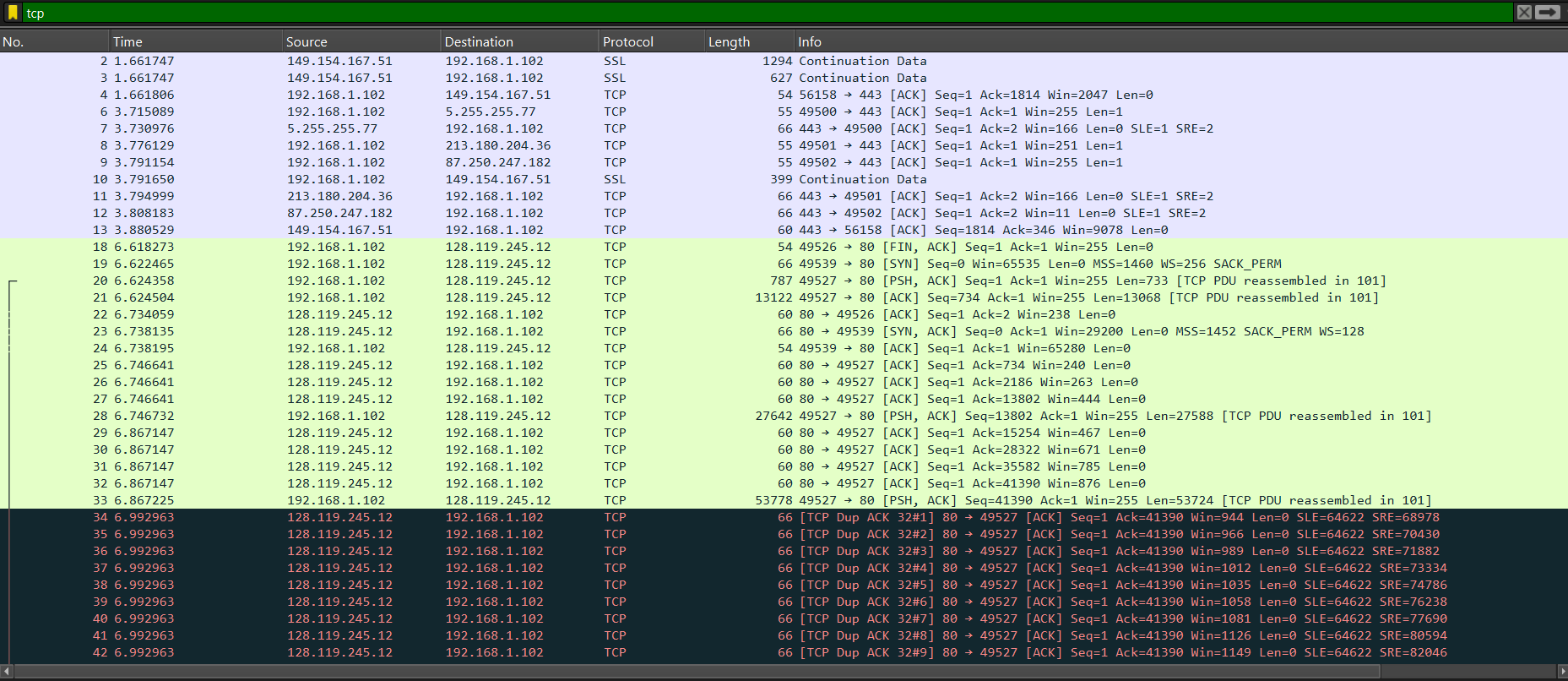
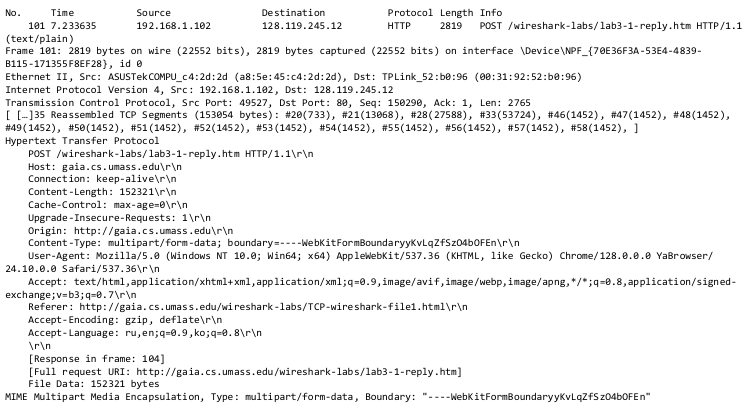
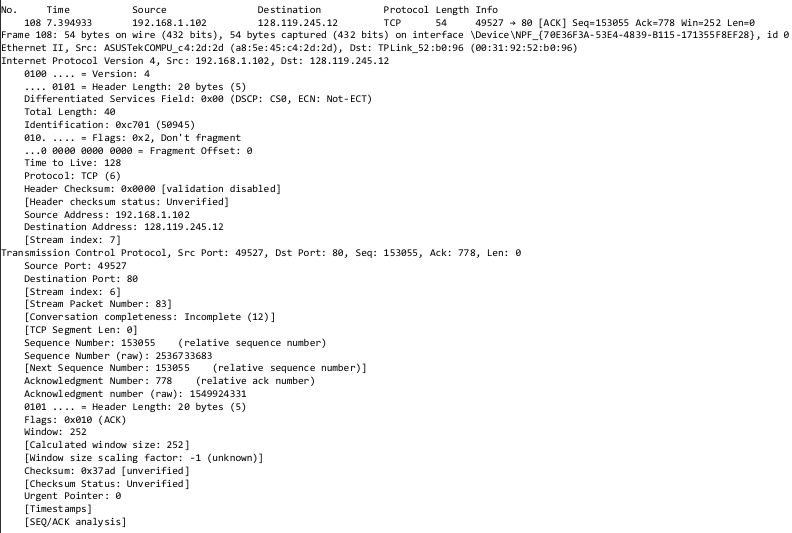


Рис. 1. Окно программы Wireshark после захвата

**Сообщение HTTP с методом POST:**

****

**Сообщение TСP:**



1. Какой IP-адрес и номер порта TCP использует компьютер клиент (отправитель), передающий файл серверу gaia.cs.umass.edu?

IP-адрес: 192.168.1.102

Номер порта TCP: 49527

1. Какой IP-адрес у сервера gaia.cs.umass.edu? Какие номера портов для отправки и приема TCP-сегментов этого соединения?

Компьютер получатель: gaia.cs.umass.edu

IP-адрес: 128.119.245.12

Номер порта TCP: 80

1. Какой порядковый номер у TCP SYN-сегмента, который используется для установления TCP-соединения между компьютером клиента и gaia.cs.umass.edu? Что указывает на то, что это именно SYN-сегмент?

Порядковый номер TCP SYN-сегмента, используемого для установления TCP-соединения между компьютером клиентом и сервером gaia.cs.umass.edu. В данной трассировке – значение 1. Флаг SYN равен 1 и это означает, что данный сегмент является SYN-сегментом.

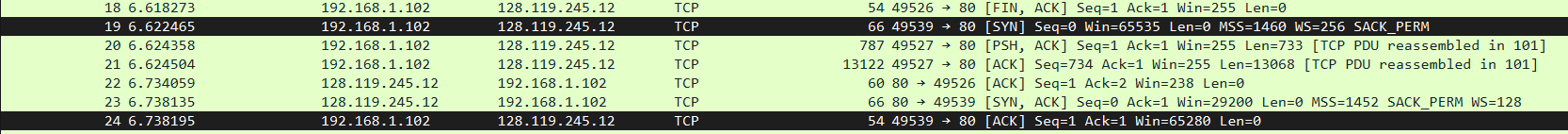
****

Рис. 2. SYN и SYNACK

1. Какой порядковый номер у SYNACK-сегмента, отправленного сервером gaia.cs.umass.edu на компьютер клиента в ответ на SYN-сегмент? Какое значение хранится в поле подтверждения в SYNACK-сегменте? Как сервер gaia.cs.umass.edu определил это значение? Что указывает на то, что это именно SYNACK-сегмент?

Порядковый номер SYNACK-сегмента, отправленного от сервера gaia.cs.umass.edu клиентскому компьютеру в ответ на SYN-сегмент, имеет значение 2 в этой трассировке.

Значение в поле ACK SYNACK-сегмента равно 1. Значение поля ACK SYNACK-сегмента определено сервером gaia.cs.umass.edu добавлением 1 к начальному порядковому номеру SYN-сегмента от клиентского компьютера

Флаги SYN и флаг ACK указывают на то, что сегмент является SYNACK-сегментом

1. Какой порядковый номер у TCP-сегмента, содержащего команду POST протокола HTTP? Заметим, что для нахождения команды POST вам потребуется проникнуть внутрь поля содержимого пакета в нижней части окна Wireshark, чтобы найти сегмент, в поле DATA которого хранится значение POST.

Сегмент номер 11 – это TCP-сегмент, содержащий команду POST протокола HTTP. Порядковый номер этого сегмента равен 1.

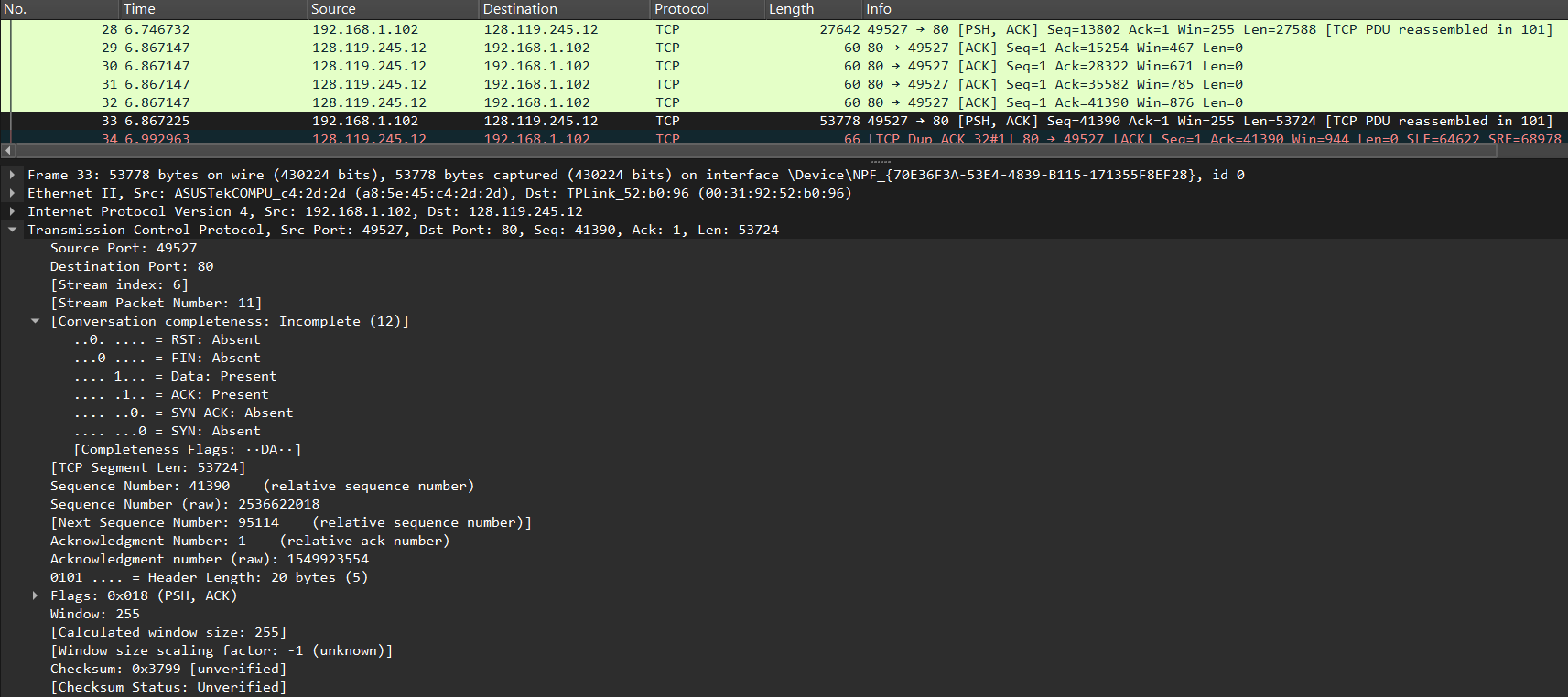


Рис. 3. Порядковый номер и номер подтверждения SYNACK-сегмента

1. Рассмотрите TCP-сегмент, содержащий команду POST протокола HTTP, как первый TCP-сегмент соединения. Какие порядковые номера у первых шести сегментов TCP-соединения (включая сегмент содержащий команду POST протокола HTTP)? Когда был отправлен каждый сегмент? Когда был получен ACK-пакет для каждого сегмента? Покажите разницу между тем, когда каждый TCP-сегмент был отправлен и когда было получено каждое подтверждение, чему равно значение RTT для каждого из 6 сегментов? Чему равно значение EstimatedRTT (см. раздел 3.5.3 в книге) после получения каждого ACK-пакета? Если предположить, что значение EstimatedRTT равно значению RTT для первого сегмента, и затем вычислить его, используя уравнение для значения EstimatedRTT из раздела 3.5.3 в книге для всех последующих сегментов.

Сегмент, содержащий команду POST протокола HTTP, рассматривается как первый сегмент. Сегменты 1 – 6 имеют номера 20, 28, 33, 73, 94, и 101 в этой трассировке, соответственно. ACK-пакеты сегментов 1 – 6 имеют номера 24, 29, 68, 89, 95, и 102 в этой трассировке.

Порядковый номер сегмента 1: 1

Порядковый номер сегмента 2: 6

Порядковый номер сегмента 3: 11

Порядковый номер сегмента 4: 51

Порядковый номер сегмента 5: 72

Порядковый номер сегмента 6: 79

Время отправки и время получения ACK приведено в следующей таблице.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Время отправки | Время получения ACK | RTT (в секундах) |
| Сегмент 1 | 6.624358 | 6.738195 | 0.113837 |
| Сегмент 2 | 6.746732 | 6.867147 | 0.120415 |
| Сегмент 3 | 6.867225 | 6.993325 | 0.1261 |
| Сегмент 4 | 6.993355 | 7.111240 | 0.117885 |
| Сегмент 5 | 7.111316 | 7.233547 | 0.122231 |
| Сегмент 6 | 7.233635 | 7.348526 | 0.114891 |

**EstimatedRTT после получения ACK-пакета для сегмента 1:**

EstimatedRTT = 0.875×1+0.125×0.130836 = 0.889229625 с

**EstimatedRTT после получения ACK-пакета для сегмента 2:**

EstimatedRTT = 0.875×0.889229625+0.125×0.120415 = 0.793127797 с

**EstimatedRTT после получения ACK-пакета для сегмента 3:**

EstimatedRTT = 0.875×0.793127797 +0.125×0.1261 = 0.709749322 с

**EstimatedRTT после получения ACK-пакета для сегмента 4:**

EstimatedRTT = 0.875×0.709749322 +0.125×0.117885 = 0.635766282 с

**EstimatedRTT после получения ACK-пакета для сегмента 5:**

EstimatedRTT = 0.875×0.635766282 +0.125×0.122231 = 0.571574372 c

**EstimatedRTT после получения ACK-пакета для сегмента 6:**

EstimatedRTT = 0.875×0.571574372 +0.125×0.114891 = 0.51448895 с

1. Какая длина у каждого из шести первых TCP-сегментов?

Длина первого TCP-сегмента, содержащего команду POST протокола HTTP, равна 733 байт. Длина каждого из оставшихся пяти TCP-сегментов равна 2765 байт.

1. Какой минимальный объем доступного буферного пространств установлен при получении всей трассировки? Может ли отсутствие места в буфере получателя замедлить отправителя?

Минимальный объем пространства буфера (окно получения), установленный сервером gaia.cs.umass.edu во всей трассировке равен 32768 байт, которые показаны в первом подтверждении от сервера. Это окно получения неуклонно увеличивается, пока максимальный размер буфера получателя не будет равен 62780 байтам. Отправитель никогда не замедлится из-за блокировки пространства буфера получателя, судя по этой трассировке.

1. Присутствуют ли повторно переданные сегменты в файле трассировки? Что вы проверяли (в трассировке) для ответа на этот вопрос?

В файле трассировки нет повторно переданных сегментов. Мы можем проверить это, сверив порядковые номера TCP-сегментов в файле трассировки. В графике Time-Sequence-Graph (Stevens)) этой трассировки, все порядковые номера от отправителя (192.168.1.102) до получателя (128.119.245.12) возрастают со временем. Если существует повторно переданный сегмент, порядковый номер такого сегмента должен быть меньше, чем номер его соседнего сегмента.

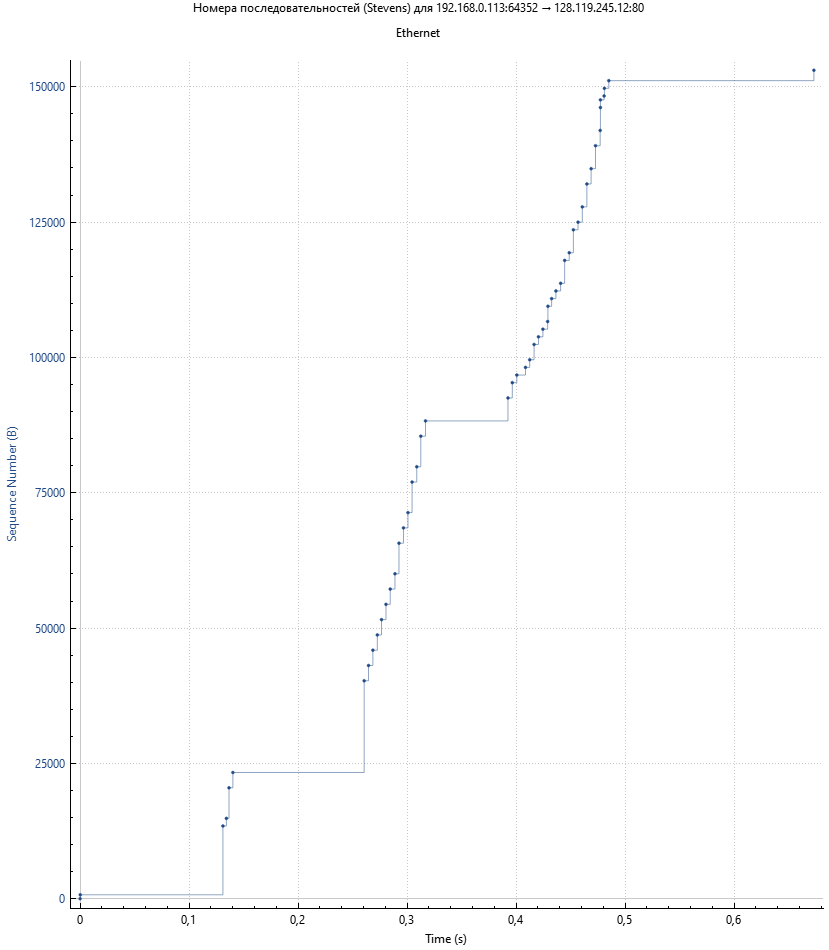


Рис. 3. Порядковые номера сегментов от отправителя к получателю

1. Сколько данных обычно подтверждает получатель ACK-пакетом?

|  |  |  |
| --- | --- | --- |
|  | Порядковый номер подтверждения | Данные подтверждения |
| ACK 6 | 52 | 2824 |
| ACK 7 | 53 | 2824 |
| ACK 8 | 54 | 2824 |
| ACK 9 | 55 | 2824 |
| ACK 10 | 56 | 2824 |
| ACK 11 | 57 | 2824 |
| ACK 12 | 58 | 2824 |
| ACK 13 | 59 | 2824 |
| ACK 14 | 60 | 5648 |

Отличие между порядковыми номерами подтверждений двух последовательных ACK-пакетов указывает на данные, которые были получены сервером между двумя этими ACK-пакетами.

1. Чему равна пропускная способность (количество байтов, передаваемых в единицу времени) для этого TCP-соединения? Объясните, как вы получили это значение.

Вычисление пропускной способности TCP-соединения во многом зависит от выбора усредненного временного периода. Средняя пропускная способность этого TCP-соединения равна отношению общего числа данных к общему количеству передач.

Общее количество переданных данных может быть подсчитано как разность между порядковыми номерами первого TCP-сегмента и порядковым номером подтверждения последнего ACK-пакета. Тогда мы получим, что общее количество данных равно 149717 – 729 = 148988 байта. Общее время передач равно разности между временем отправки первого TCP. Тогда общее время передачи равно 5.079104 – 0.034069 = 5.045035 секунд. Следовательно, пропускная способность TCP-соединения вычисляется, как 148988 / 5.045035 = 28.839 Кб/с

1. Используйте инструмент построения графиков Time-Sequence-Graph(Stevens), чтобы представить изменение порядковых номеров на временной шкале для сегментов, отправленных от клиента серверу gaia.cs.umass.edu. Можете ли вы определить, где начался и закончился этап медленного старта TCP-соединения, и где закончился этап предотвращения перегрузки? Прокомментируйте моменты, в которых измеренные данные отличаются от эталонного поведения TCP, которое мы изучим далее.

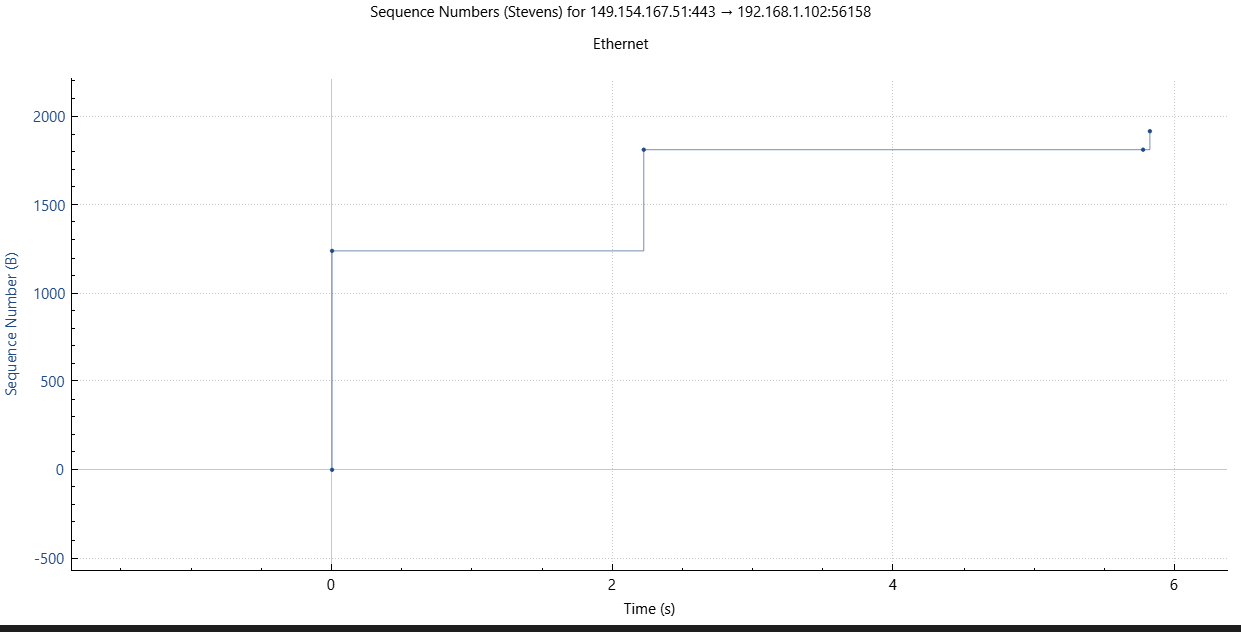
Этап медленного старта TCP-соединения начинается в начале соединения, т.е., при отправке сегмента, содержащего команду POST протокола HTTP. Идентификация этапа медленного старта TCP-соединения и этапа предотвращения перегрузки зависит от значения размера окна перегрузки TCP-отправителя. Однако значение размера окна перегрузки не может быть получено из графика Time-Sequence-Graph (Stevens). Тем не менее, мы можем вычислить нижнюю границу размера окна TCP, посчитав количество отправленных данных, поскольку количество отправленных данных – это количество неподтвержденных данных. Нам также известно, что окно TCP ограничивается размером окна получателя, и размер буфера получателя может быть верхней границей размера окна TCP. В этой трассировке буфер получателя не мал, поэтому такая верхняя граница не очень полезна в определении размера окна TCP. Следовательно, мы сфокусируемся на нижней границе размера окна TCP.

Из следующей таблицы мы не можем понять, насколько быстро возрастает количество переданных данных в этом потоке TCP; но оно никогда не превышает 8192 байт. Поэтому мы можем убедиться, что размер окна TCP больше, чем 8192 байт. Тем не менее, мы не можем определить конец этапа медленного старта и начало этапа предотвращения перегрузки для этой трассировки. Главная причина в том, что этот TCP-отправитель не отправляет данные достаточно агрессивно, чтобы вызвать состояние перегрузки. Проверяя количество переданных данных, мы можем установить, что приложение в большинстве случаем отправляет блоки данных по 8192 байта. До получения подтверждения на весь блок данных из этих 8192 байтов приложение не будет больше отправлять данные. Это указывает на конец этапа медленного старта, приложение уже временно останавливает передачу.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип | Номер | Порядковый номер | Подтвержденные порядковые номера | Необработанные данные |
| Данные | 34 | 729 |  | 729 |
| АСК | 46 |  | 729 | 1460 |
| Данные | 47 | 13437 |  | 2920 |
| АСК | 48 |  | 3553 | 2025 |
| Данные | 49 | 14849 |  | 4380 |
| АСК | 50 |  | 4965 | 3812 |
| Данные | 51 | 20497 |  | 5527 |
| АСК | 52 |  | 12025 | 3007 |
| АСК | 53 |  | 13437 | 1460 |
| Данные | 54 | 23321 |  | 7300 |
| Данные | 58 | 40265 |  | 8192 |
| АСК | 59 |  | 16261 | 5840 |
| Данные | 60 | 43089 |  | 6724 |
| АСК | 61 |  | 17673 | 5272 |
| АСК | 63 |  | 19085 | 2352 |
| Данные | 64 | 48737 |  | 6732 |
| Данные | 66 | 51561 |  | 8192 |
| АСК | 67 |  | 21909 | 6732 |
| АСК | 69 |  | 23321 | 5272 |
| АСК | 71 |  | 26145 | 3812 |
| АСК | 73 |  | 27557 | 2352 |
| АСК | 75 |  | 28969 | 892 |
| АСК | 77 |  | 31793 | 0 |
| Данные | 78 | 71329 |  | 1460 |
| Данные | 80 | 76977 |  | 2920 |
| Данные | 82 | 79801 |  | 4380 |
| Данные | 84 | 85801 |  | 5840 |
| Данные | 87 | 88273 |  | 7300 |
| Данные | 89 | 92509 |  | 8192 |
| АСК | 90 |  | 43089 | 5272 |
| АСК | 92 |  | 44501 | 2352 |
| АСК | 94 |  | 45913 | 0 |
| Данные | 95 | 98157 |  | 1460 |
| Данные | 97 | 99569 |  | 2920 |
| Данные | 99 | 102393 |  | 4380 |
| Данные | 101 | 103805 |  | 5840 |
| Данные | 103 | 105217 |  | 7300 |
| Данные | 105 | 106629 |  | 8192 |
| АСК | 106 |  | 54385 | 5272 |
| АСК | 108 |  | 55797 | 2352 |
| АСК | 110 |  | 57209 | 0 |
| Данные | 111 | 112277 |  | 1460 |

1. Прокомментируйте, в каких измеряемых данных отличие от идеального поведения TCP, которое мы изучали в книге.

Идеализированное поведение TCP в тексте книги предполагало, что TCP-отправители агрессивно отправляют данные. Слишком много трафика может привести к перегрузке сети; поэтому TCP-отправители должны использовать алгоритм AIMD, чтобы при обнаружении сетевой перегрузки (то есть потери пакетов), размер окна отправления должен уменьшаться. На практике поведение TCP также сильно зависит от приложения. В веб-приложениях некоторые веб-объекты имеют очень маленький размер. Перед завершением этапа медленного старта передача прекращается, следовательно, передача этих маленьких веб-объектов страдает от ненужной длительной задержки этапа медленного старта протокола TCP.



**Wireshark: UDP**

Начнем захват пакетов в приложении Wireshark, затем сделаем так, чтобы хост отправил и получил несколько UDP-пакетов. После захвата пакетов, в фильтре необходимо указать: UDP. Для того чтобы отображались только нужные нам пакеты.

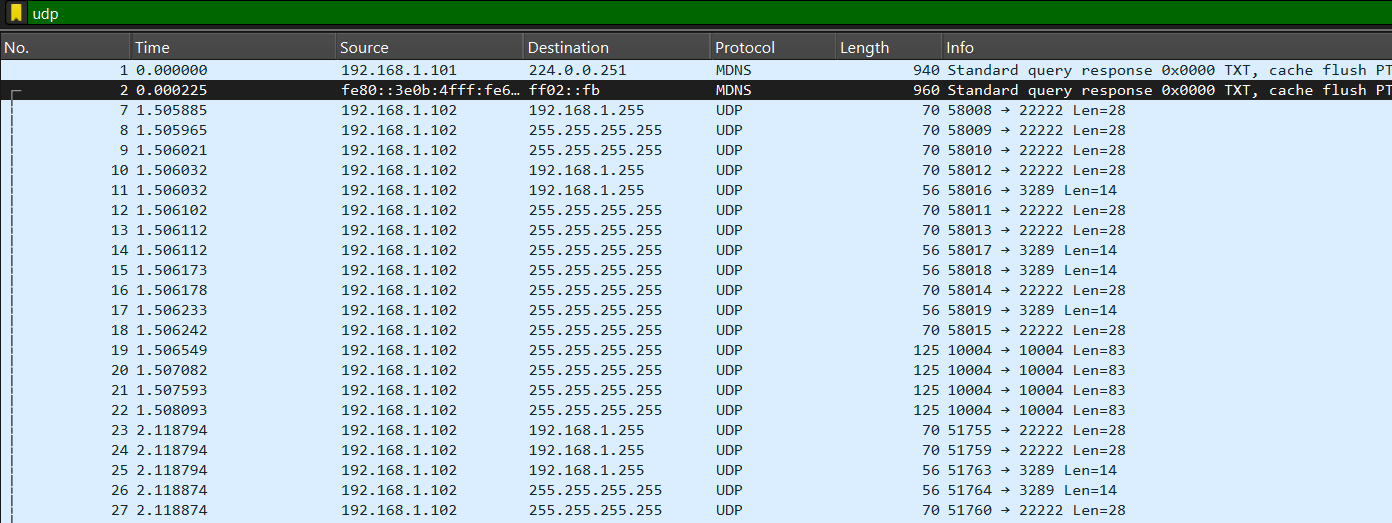


Рис. 1. Результат захваченных паетов

Ниже представлено содержимое трассировки первого пакета:

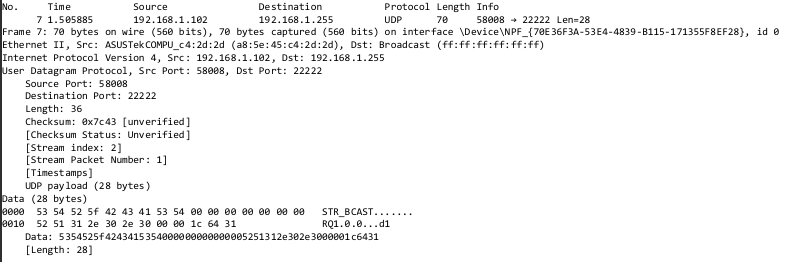


Рис. 2. Содержимое первого пакета

1. Выберите один UDP-пакет из трассировки. На основе этого пакета определите, сколько полей содержит UDP-заголовок. (Не следует подсматривать в книгу! Ответьте на вопрос на основании того, что вы видите в трассировке пакета.) Назовите эти поля.

В заголовке присутствует четыре поля: порт отправителя, порт получателя, длина и контрольная сумма.

1. Определите длину (в байтах) для каждого поля UDP заголовка, обращаясь к отображаемой информации о содержимом полей в данном пакете в программе Wireshark.

Нажав на поле порта отправителя (Source Port: 58008), видим значение, соответствующее значению номера порта в окне содержимого пакета внизу экрана Wireshark. Номер порта представлен в шестнадцатеричном виде(10 ee 00 a1 00 3a 65 f8) и в формате ASCII, и он составляет 2 байта.

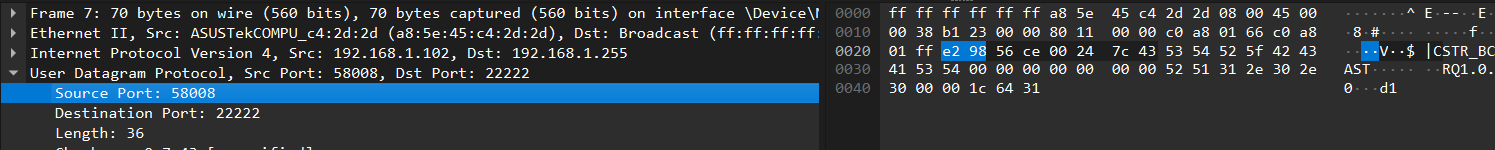


Рис. 2. Заголовок UDP

1. Значение в поле Length (Длина) – это длина чего? (Вы можете обратиться к тексту книги для ответа на этот вопрос.) Проверьте ваше утверждение на вашем захваченном UDP пакете.

Длина в UDP – это длина заголовка и поля данных UDP-сегмента в байтах. Показанный пакет имеет длину 28 байт. Мы знаем, что 8 байт приходится на заголовок. Если мы рассматриваем поле содержимого пакета, мы также обнаружим 20 байт данных в шестнадцатеричном или ASCII-формате, которые соответствуют полезной нагрузке этого UDP-сегмента.

1. Какое максимальное количество байт может быть включено в полезную нагрузку UDP?

Поскольку есть только 16 бит, максимальная длина UDP-сегмента (включая заголовок) составляет 216 – 1 или 65535 байт.

1. Чему равно максимально возможное значение номера порта источника?

Поскольку есть только 16 бит, максимальное значение номера порта равно 216 – 1 или 65535 байт.

1. Какой номер протокола для UDP? Дайте ответ и для шестнадцатеричной и десятеричной системы. Чтобы ответить на этот вопрос, вам необходимо заглянуть в поле Протокол в IP дейтаграмме, содержащей сегмент UDP.

Номер протокола UDP равен 2

1. Проверьте UDP-пакет и ответный UDP-пакет, отправляемый вашим хостом. Определите отношение между номерами портов в двух пакетах.

Рассмотрим пакеты 1 и 2 в трассировке. Эти пакеты содержат SNMP-сообщения прикладного уровня внутри UDP-пакета. IP-адрес отправителя в пакете 1 – это IP-адрес получателя в пакете 2, и IP-адрес получателя в пакете 1 – это IP-адрес отправителя в пакете 2, имя в поле Info get-запроса и get-ответа сообщает, что второй пакет (ответ) отправлен в ответ на первый пакет (запрос).

Значение порта ***источника*** в пакете 1 – это значение порта ***получателя*** в пакете 2;

Значение порта ***получателя*** в пакете 1 – это значение порта ***источника*** в пакете 2.