1. Рассмотреть классическую многоуровневую модель OSI. Определить какие протоколы возможны на каждом уровне (привести по 2 примера на каждом уровне).

**Физический**

Протоколы: Ethernet, USB

Описание: Физическая передача данных

**Канальный**

Протоколы: IEEE 802.3, ARP

Описание: Физическая передача данных

**Сетевой**

Протоколы: Internet Protocol (IP), Internet Control Message Protocol (ICMP)

Описание: Физическая адресация

**Транспортный**

Протоколы: TCP, UDP

Описание: Определение маршрута и логическая адресация

**Сеансовый**

Протоколы: L2TP, gRPC

Описание: Прямая связь между конечными пунктами и надёжность

**Представления**

Протоколы: SSL, gzip

Описание: Управление сеансом связи

**Прикладной**

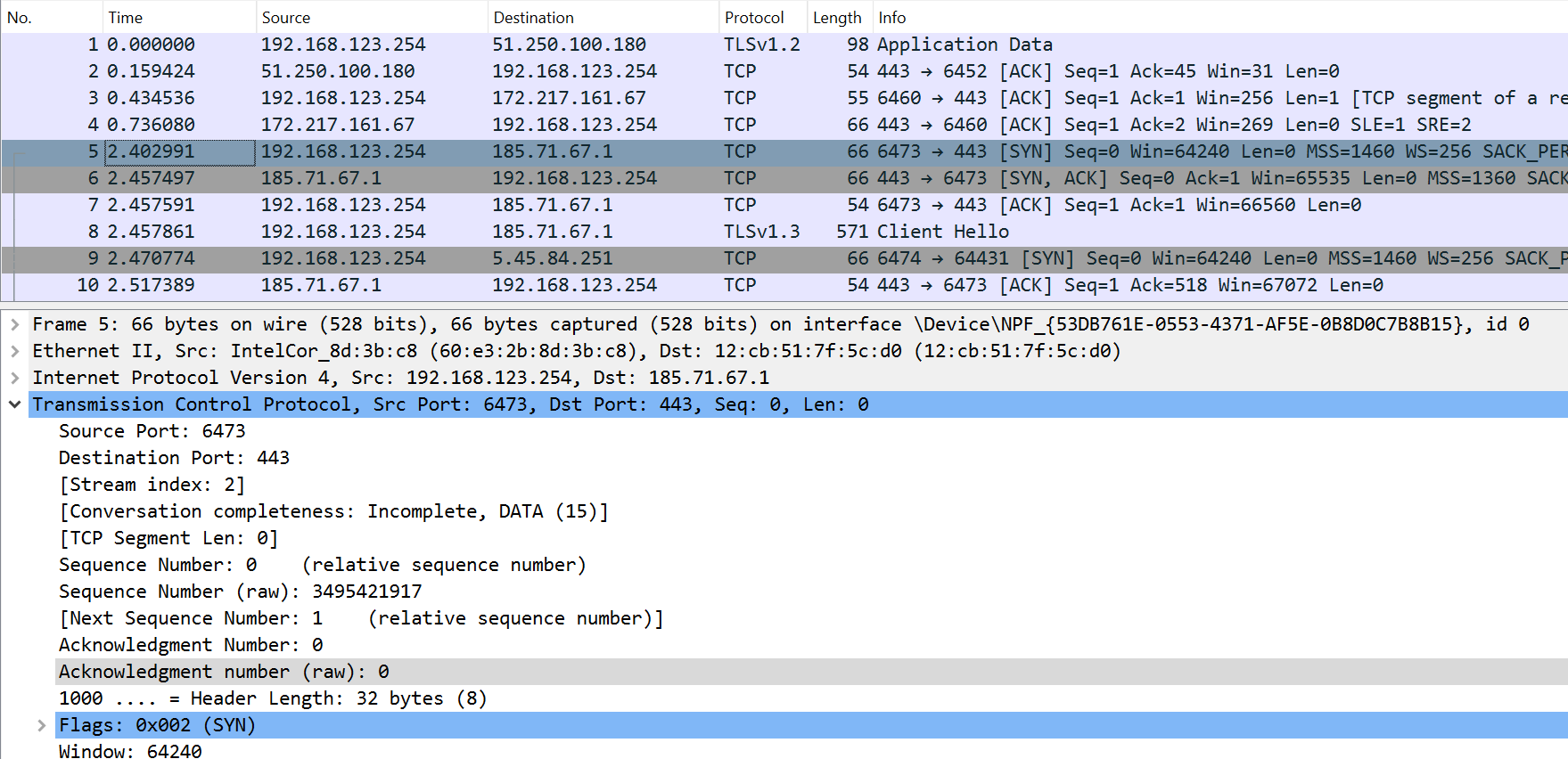
Протоколы: HTTP, FTP

Описание: Доступ к сетевым службам

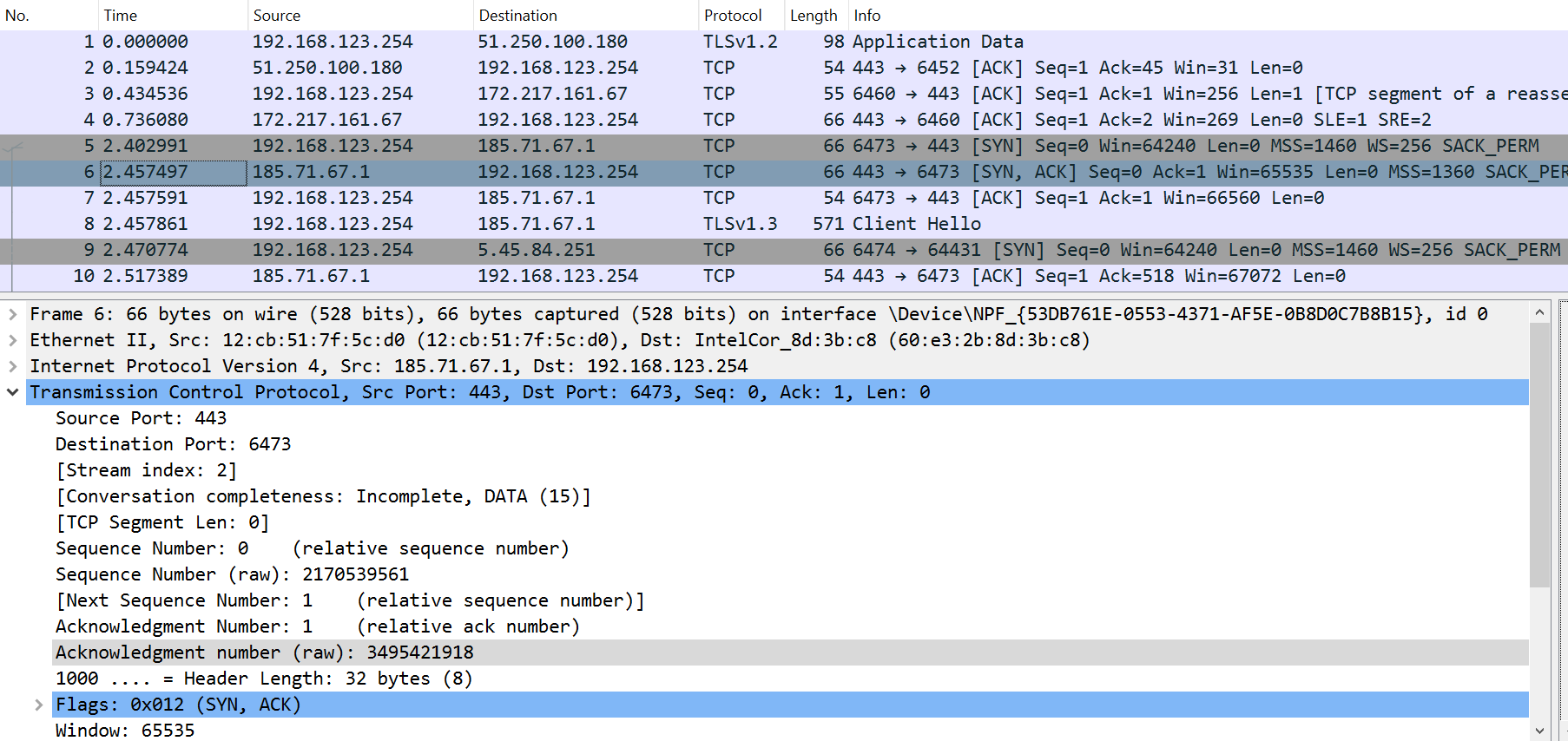
1. Запустить программу Wireshark. Настроить программу на активное интернет соединение. Рассмотреть различные TCP и UDP соединения. Привести в трехстороннее рукопожатие и трехстороннее завершение сеанса TCP (привести не только сам процесс, но и раскрыть флаги для каждого пакета).

**TCP Рукопожатие**

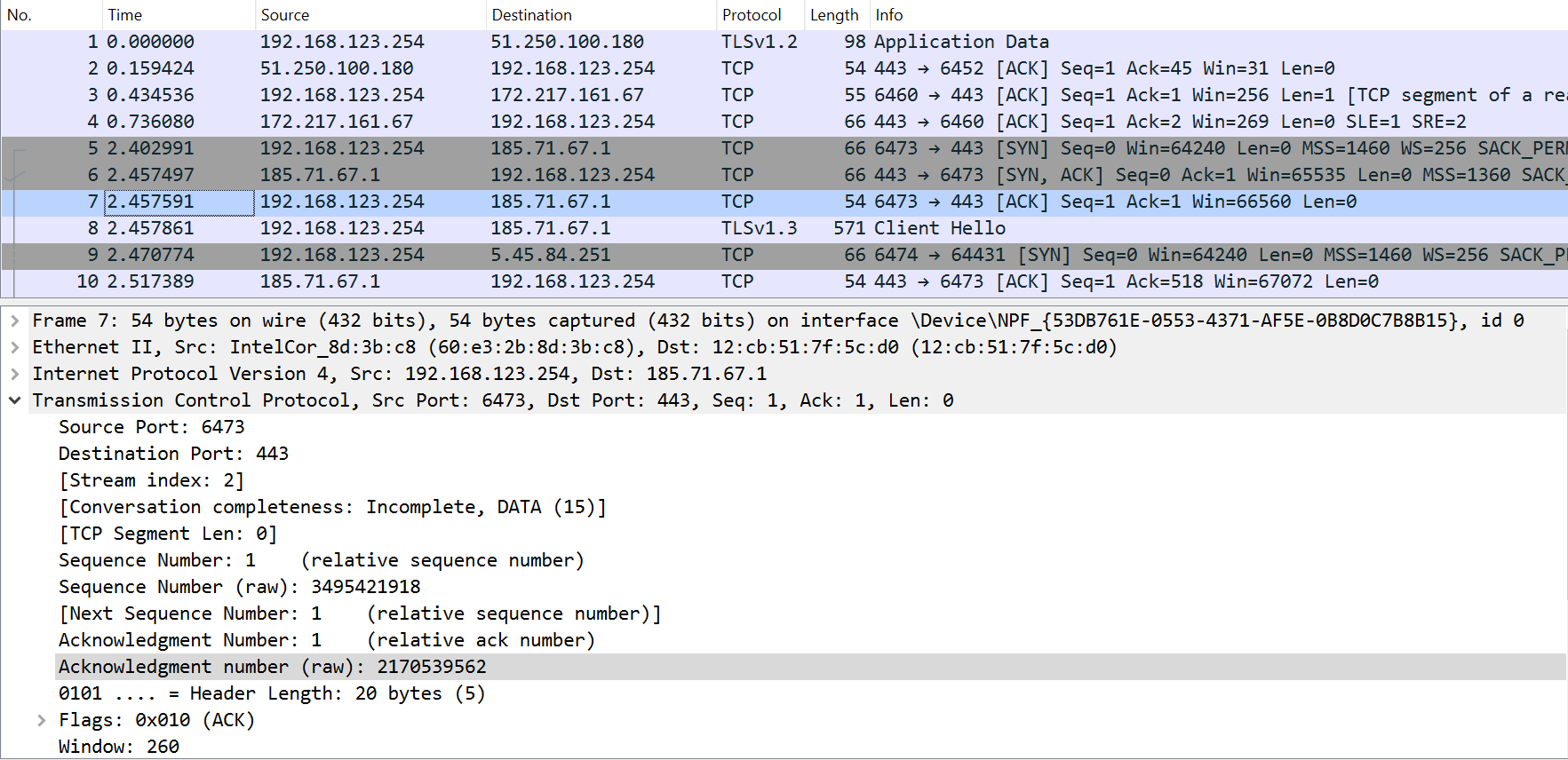
1) Клиент отправляет сегмент с установленным флагом SYN. При этом сегменту присваивается произвольный порядковый номер (sequence number) в интервале от 1 до 232 (т.н. initial sequence number), относительно которого будет вестись дальнейший отсчет последовательности сегментов в соединении.



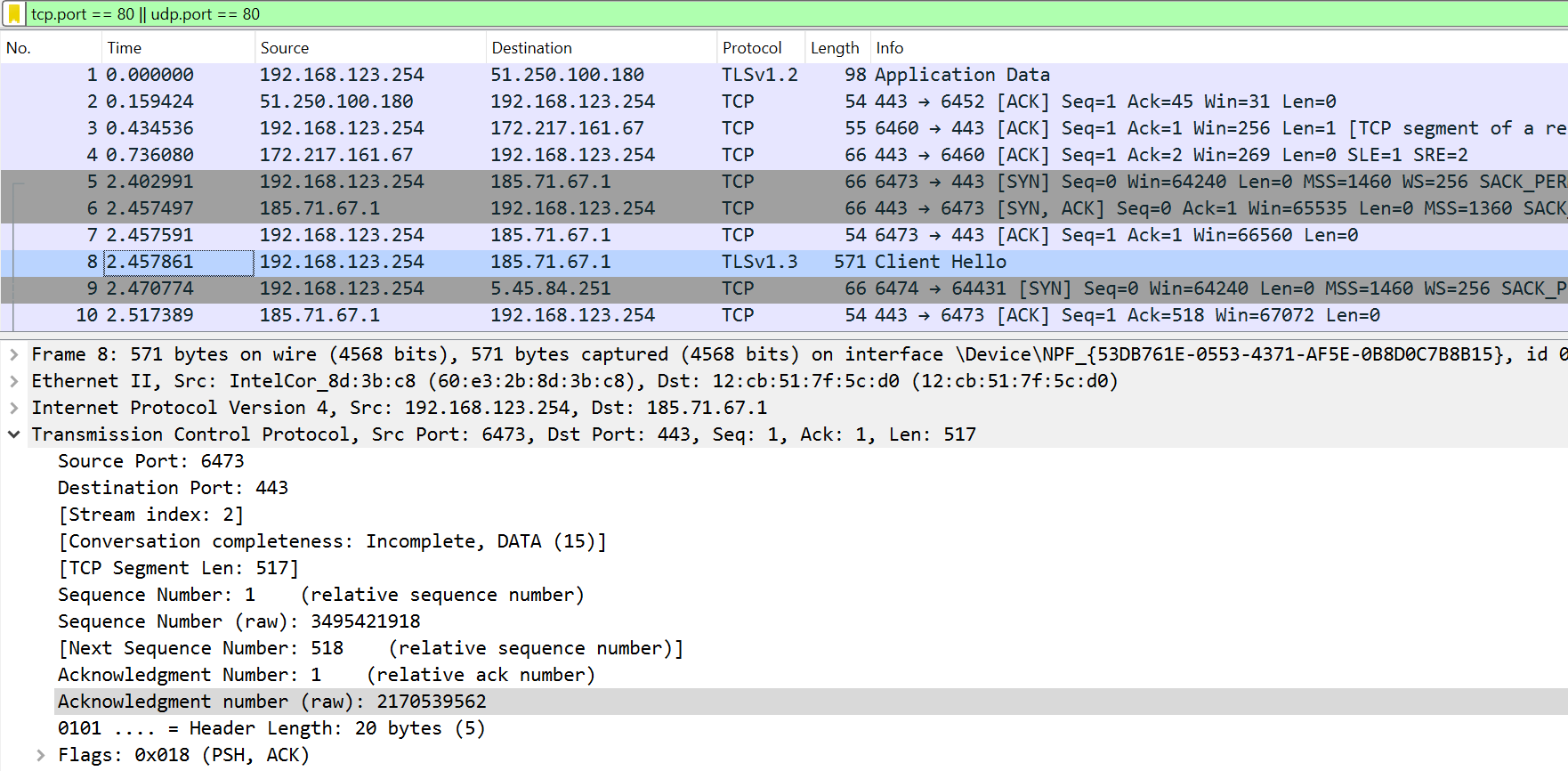
2) Сервер получает запрос и отправляет ответный сегмент с одновременно установленными флагами SYN+ACK, при этом записывает в поле «номер подтверждения» (acknowledgement number), полученный порядковый номер, увеличенный на 1 (что подтверждает получение первого сегмента), а также устанавливает свой порядковый номер, который, как и в SYN-сегменте, выбирается произвольно.



3) После получения клиентом сегмента с флагами SYN+ACK соединение считается установленным, клиент, в свою очередь, отправляет в ответ сегмент с флагом ACK, обновленными номерами последовательности, и не содержащий полезной нагрузки.

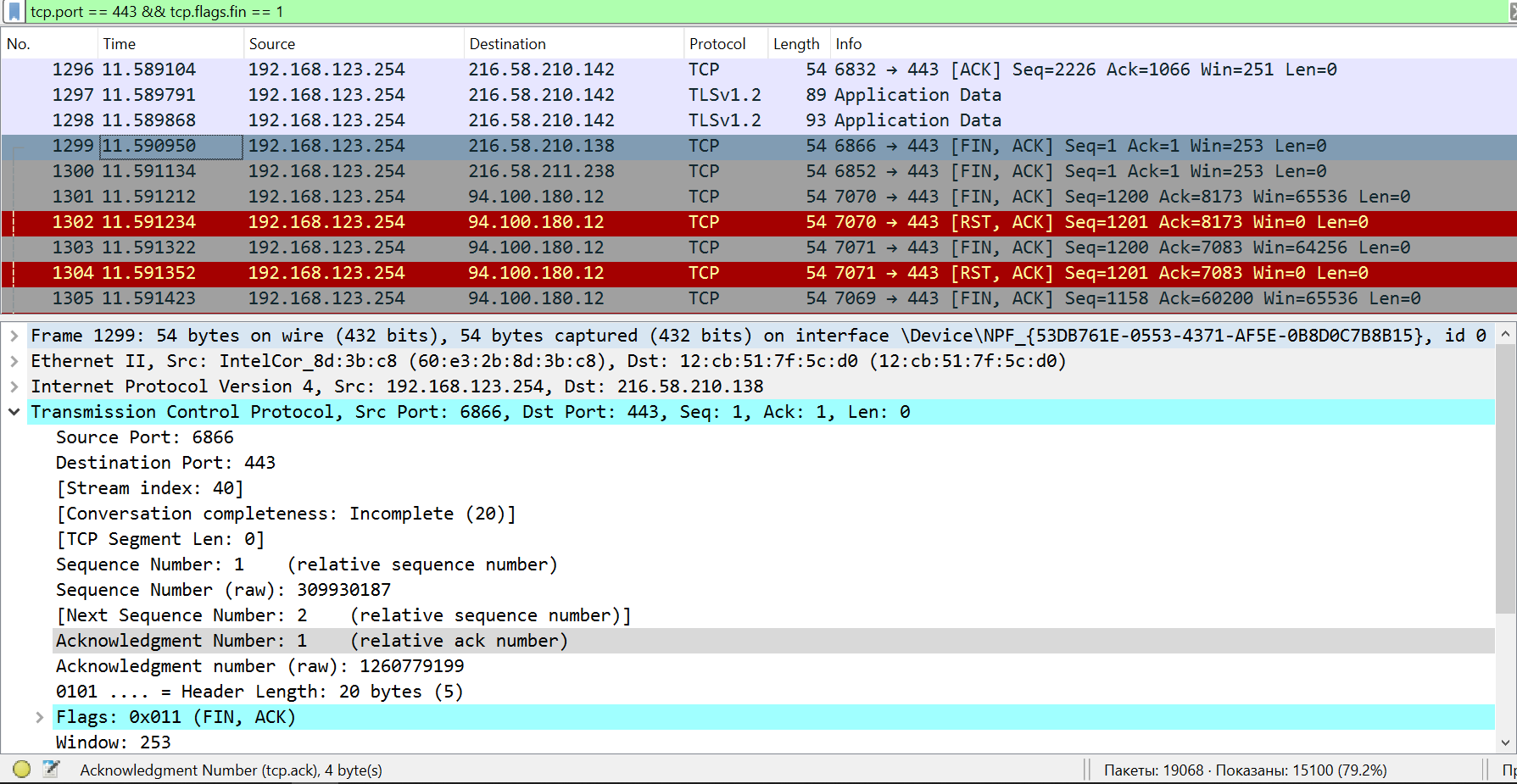


4) Начинается передача данных.

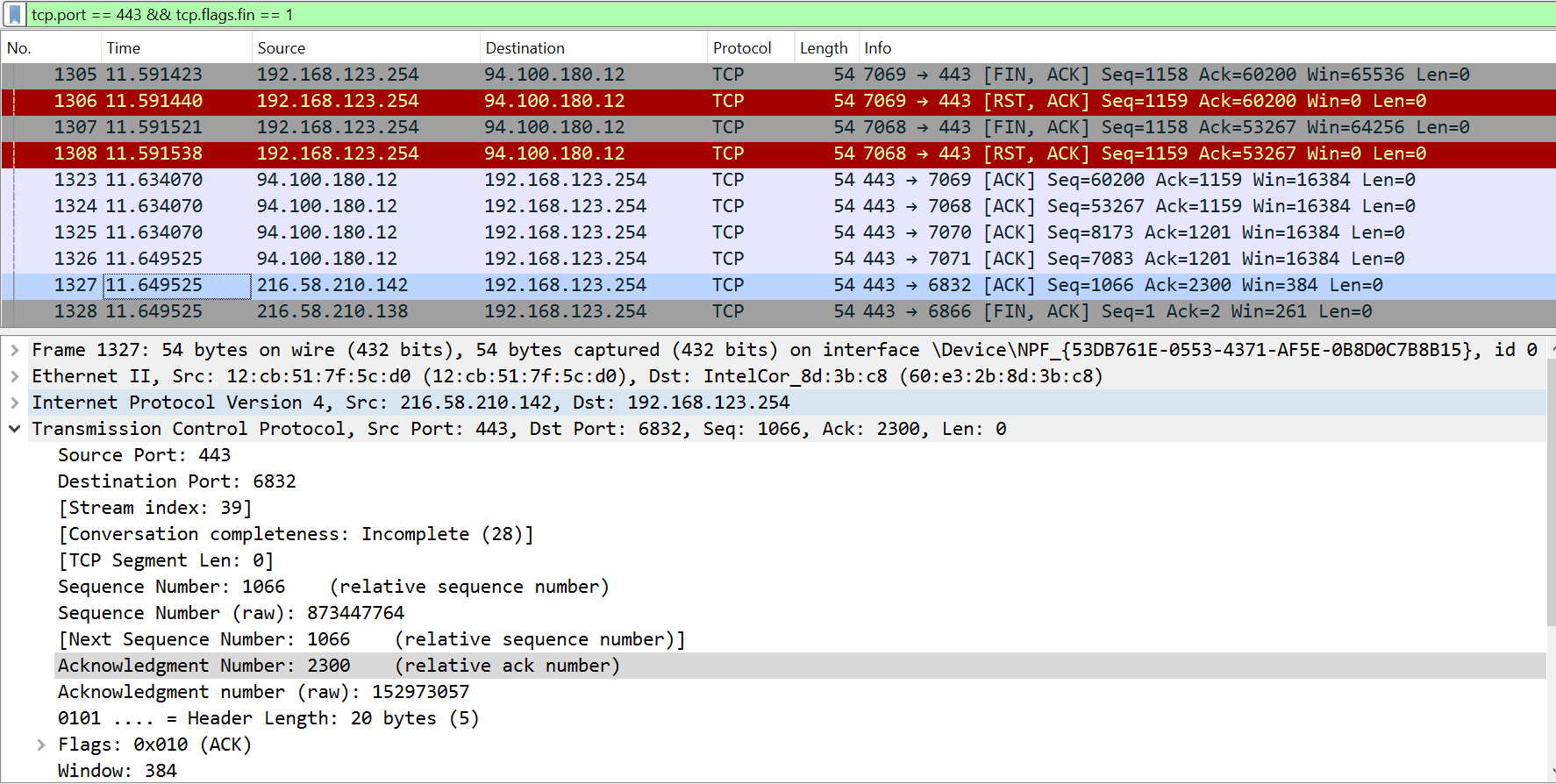


**TCP завершение сеанса**

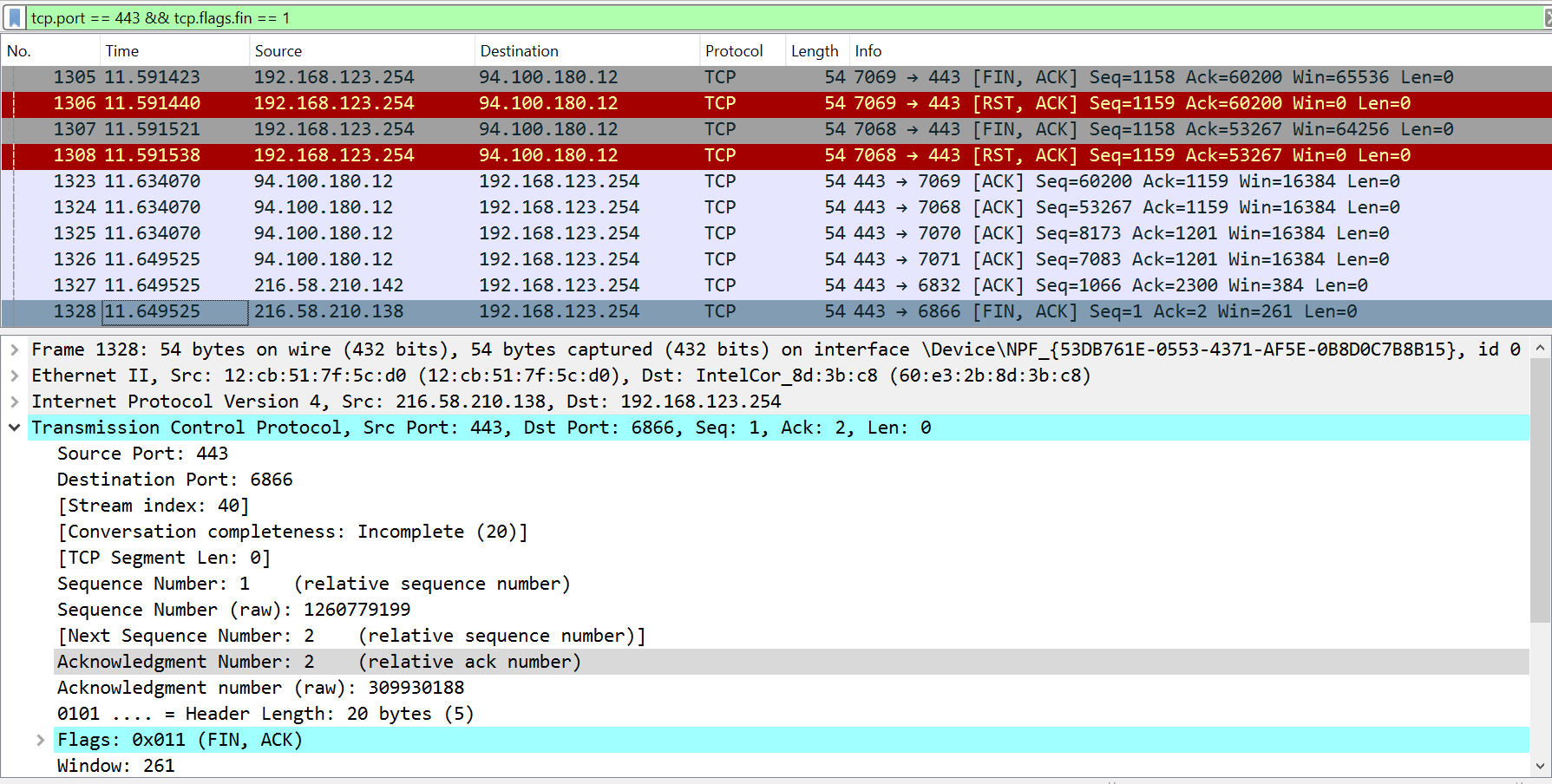
1. Один из задействованных процессов приложения сигнализирует своему уровню TCP, что сеанс связи больше не нужен. Со стороны этого устройства отправляется сообщение с установленным флагом FIN (отметим, что этот пакет не обязательно должен быть пустым, он также может содержать полезную нагрузку), чтобы сообщить другому устройству о своем желании завершить открытое соединение.



1. Затем получение этого сообщения подтверждается (сообщение от отвечающего устройства с установленным флагом ACK, говорящем о получении сообщения FIN).



1. Когда отвечающее устройство готово, оно также отправляет сообщение с установленным флагом FIN, и, после получения в ответ подтверждающего получение сообщения с установленным флагом ACK или ожидания определенного периода времени, предусмотренного для получения ACK, сеанс полностью закрывается.



1. Рассмотреть процесс установления TLS соединения. В отчете привести заголовки стека протоколов основных сообщений при создании TSL соединения.
2. С помощью браузера отправить запрос для получения HTML страницы. Найти соответствующий запрос и ответ в программе Wireshark. В отчете полностью отобразить все содержимое стека протоколов на каждом уровне для получения HTML (не приводить полностью код HTML, а только до конца страницы и блок HTTP chunked response)
3. Проанализировать содержимое данных, отправленных в запросе и полученных в ответе. Расписать в отчете основные моменты соединения для каждого из протоколов.
4. Рассмотреть загрузку других ресурсов сайта (CSS/ PNG/ JPG и т.д.) как запрос на данный ресурс, так и ответ.