IS8055556: Data and Computer Communications	**Recitation 12**
Semester 2 5786	**7 July 2026**
Lecturer: Michael J. May	**Tel Hai**

Wireshark with UDP and TCP

In this recitation we'll use Wireshark to analyze the behavior of UDP and TCP between computers in the network.

1 About Wireshark

Wireshark is a popular free protocol analyzer and packet sniffer that we've used in previous lectures and assignments. Wireshark is capable of recording packets that enter and exit the computer to create a *packet trace* of what data has entered and exited the computer. When running on Windows, Wireshark only sees packets that have already passed the basic filtering done by the network card and Windows, so we can't see low level packets such as Ethernet frames not meant for the computer, link layer management frames, or network layer packets not destined for or sent by the recording computer. We can, however, see all network layer and higher packets sent by the recording computer and received by the recording computer.

You can select a packet from a trace and examine its internal fields and values. We will use Wireshark to examine the packets sent between two computers which talk in UDP and TCP. The traces will let us see the fields of all packets sent as well as higher level actions such as TCP's three way handshake and the interaction between packet data and acknowledgement (ACK) messages.

2 UDP and Wireshark

Let's start with a simple UDP trace and analyze its contents. The trace is called UDP-Sample-Trace.pcapng and is found on Moodle.

First, open the file using Wireshark. You should see 6 packets inside. The trace is shown in Figure 1. The trace labels each packet using two different methods:

1. Packet number - a sequential numbering of the packets in the trace
2. Time - the time in seconds the packet was sent/received relative to the start of the trace.

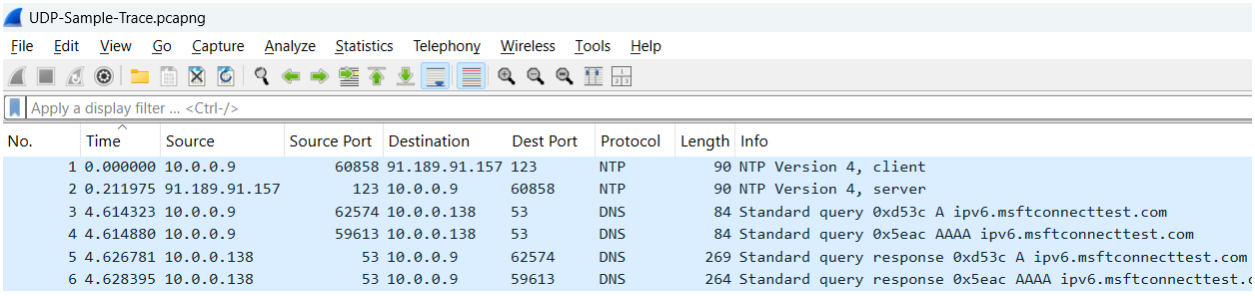

Figure 1: UDP Trace Contents

To make things a bit clearer, the screen shot above has 2 extra columns that you won't see in most traces by default - source port and destination port.

2.1 What to do: A Tour of the UDP Trace

Let's look through the packets step by step.

1. Select the packet from time 0.00000. What is the IP address of the computer that sent the packet? What's the IP address of the computer who is supposed to receive it?
2. What is the source port for the packet? What is the destination port for the packet?
3. What protocol is being sent here? How do you know? (Hint: Look at `https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers`)
4. Packet 0.00000 was a request. Find the packet that is the response for it. What time was is sent? How do you know it's the response?
5. Find the packet sent at time 4.614323. What are the IP address and port of the sender? What are the IP address and port of the destination?
6. What protocol is being sent here? How do you know? (Hint: Look at `https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers`)
7. The packet at 4.614323 is a request packet. Find the response packet. How do you know it's the response?

3 TCP and Wireshark

Looking at a trace of a download of a large file from `mirror.neolabs.kz` (195.93.153.37), we can use the filter to see packets which come from the remote server (see Figure 2). The first two packets show the SYN/ACK packet part of the three way handshake and the second one shows a regular ACK from the server.

If we click on the Analyze menu and select *Analyze* → *Conversation Filter* → *TCP* we can see just the packets in the conversation detailed. The result is shown in Figure 2. The conversation in the figure shows the full three way handshake in TCP (SYN → SYN/ACK → ACK) followed by the HTTP request which lead to the download (requesting `/linuxmint/iso/stable/18.1/linuxmint-18.1-cinnamon-64bit.iso`). Some of the following download can be seen in the segments below, including two ACKs from the client (10.0.0.6).

Selecting *Statistics* → *TCP Stream Graphs* → *Throughput* shows us a graph of the behavior of the TCP conversation (in this case it's Stream 6 - you can switch between streams using the bottom right hand side *Stream* selector). The result can be seen in Figure 3. Notice the slight ups and downs in the conversation over time due to occasional drops and out of order packets. Since the conversation took place more or less unimpeded, there isn't much to bother the download. If multiple conversations had been going on at the same time, we'd see more variance.

3.1 What to do: A TCP trace using Wireshark

Once you have Wireshark running, we will use it to track a TCP download.

1. Start a Wireshark packet capture on the wired or wireless interface that your computer is using.
2. Open your favorite browser and point it to a non-encrypted URL. Some examples you can try are:
   - `http://www.faqs.org/images/library.jpg`
   - `http://www.people.com.cn/img/2020peopleindex/img/copy_icon1.png`
3. Once the download is finished, stop the packet capture.

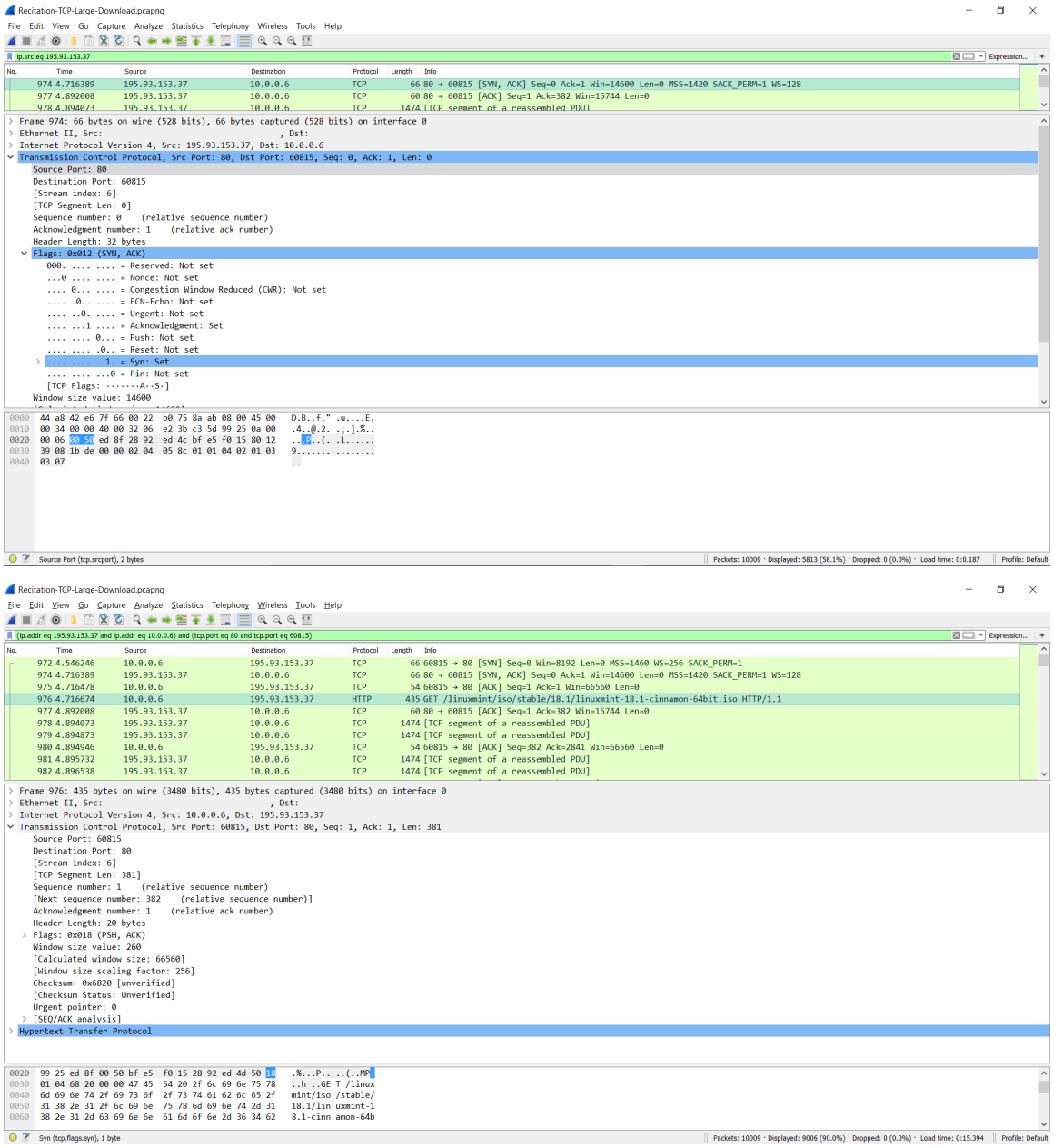

Figure 2: Wireshark showing only packets from a single source (top) and showing a TCP conversation for an HTTP request (bottom)

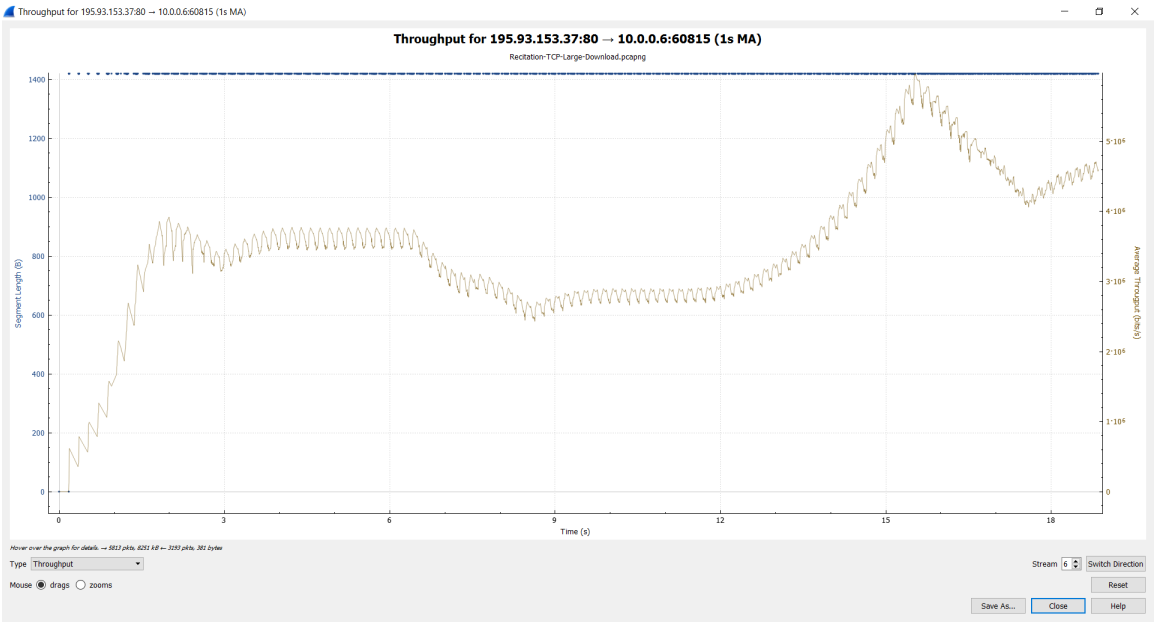

Figure 3: Wireshark showing a TCP throughput graph for the download shown above

4. Find the download session in the packet trace. You can use the packet filter to find TCP and HTTP conversations. You may want to use the IP address of the remote web site when searching since there are likely to be multiple TCP sessions going on at once from your computer. The IP addresses for the above two web sites are:
   - `www.faqs.org` → 199.231.164.68
   - `www.people.com.cn` → 138.113.247.70
   - For any other website (*e.g.* www.example.com), you can use the `nslookup` tool from the command line. To find out www.example.com's IP address, enter the following command: `nslookup www.example.com`. You'll get an answer that should include one or more lines that are called "Non-authoritative answer". The address shown there is the IP address.
5. Use Wireshark to show only the TCP conversation for the download. How many packets were sent? Were any of them fragmented? Were any of them dropped?
6. Use the TCP Stream Graphs feature shown above to display the throughput behavior for the conversation.

Repeat the steps above on a similar size file from other non-encrypted web sites such as: `http://httpforever.com/` or `http://httpbin.org/`.