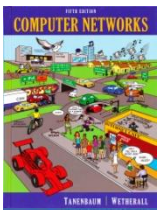
Lab Exercise – UDP



Objective

To look at the details of UDP (User Datagram Protocol). UDP is a transport protocol used throughout the Internet as an alternative to TCP when reliability is not required. It is covered in §6.4 of your text. Review that section before doing this lab.

The trace file is here: http://scisweb.ulster.ac.uk/~kevin/com320/labs/wireshark/trace-udp.pcap

Step 1: Capture a Trace

There are many ways to cause your computer to send and receive UDP messages since UDP is widely used as a transport protocol. The easiest options are to:

- Do nothing but wait for a while. UDP is used for many "system protocols" that typically run in the background and produce small amounts of traffic, e.g., DHCP for IP address assignment and NTP for time synchronization.

- Use your browser to visit sites. UDP is used by DNS for resolving domain names to IP addresses, so visiting fresh sites will cause DNS traffic to be sent. Be careful not to visit unsafe sites; pick recommended sites or sites you know about but have not visited recently. Simply browsing the web is likely to cause a steady stream of DNS traffic.

- Start up a voice-over-IP call with your favorite client. UDP is used by RTP, which is the protocol commonly used to carry media samples in a voice or video call over the Internet.

Proceed as follows to capture a trace of UDP traffic; alternatively, you may use a supplied trace:

1. *Launch Wireshark and start a capture with a filter of "*`udp`*".*

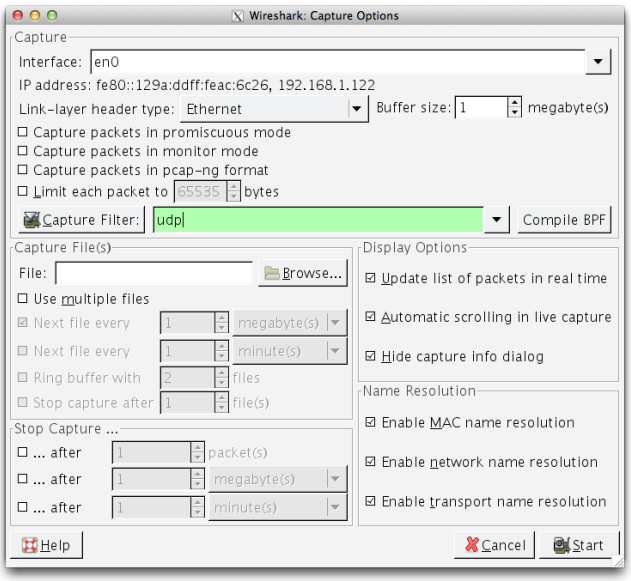


Figure 1: Setting up the capture options

2. *When the capture is started, perform some activities that will generate UDP traffic.* We described several options above, e.g., browse the web or start a short VoIP call.

3. *Wait a little while (say 60 seconds) after you have stopped your activity to also observe any background UDP traffic*. It is likely that you will observe a trickle of UDP traffic because system activity often uses UDP to communicate. We want to see some of this activity.

4. *Use the Wireshark menus or buttons to stop the capture*. You should now have a trace with possibly many UDP packets. Our example is shown below. We have selected a packet and expanded the detail of the UDP header.

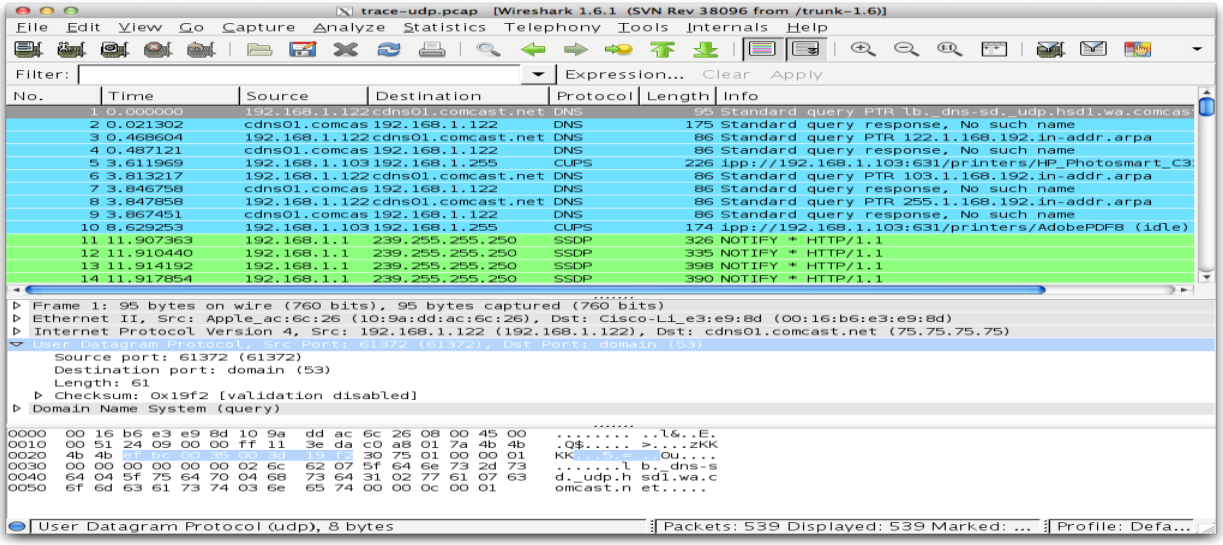


Figure 2: Trace of UDP traffic showing the details of the UDP header

Step 2: Inspect the Trace

Different computers are likely to capture different kinds of UDP traffic depending on the network setup and local activity. Observe that the protocol column is likely to show multiple protocols, none of which is UDP. This is because the listed protocol is an application protocol layered on top of UDP. Wireshark gives the name of the application protocol, not the (UDP) transport protocol unless Wireshark cannot determine the application protocol. However, even if the packets are listed as an application protocol, they will have a UDP protocol header for us to study, following the IP and lower-layer protocol headers.

Select different packets in the trace (in the top panel) and browse the expanded UDP header (in the middle panel). You will see that it contains the following fields:

- Source Port, the port from which the UDP message is sent. It is given as a number and possibly a text name; names are given to port values that are registered for use with a specific application.
- Destination Port. This is the port number and possibly name to which the UDP message is destined. Ports are the only form of addressing in UDP. There computer is identified using the IP address in the lower IP layer.
- Length. The length of the UDP message.
- Checksum. A checksum over the message that is used to validate its contents. Is your checksum carrying 0 and flagged as incorrect for UDP messages sent from your computer? On some computers, the operating system software leaves the checksum blank (zero) for the NIC to compute and fill in as the packet is sent. This is called protocol offloading. It happens after Wireshark sees the packet, which causes Wireshark to believe that the checksum is wrong and flag it with a different color to signal a problem. You can remove these false errors if they are occurring by telling Wireshark not to validate the checksums. Select “Preferences” from the Wireshark menus and expand the “Protocols” area. Look under the list until you come to UDP. Uncheck “Validate checksum if possible”.

Step 3: UDP Message Structure

To check your understanding of UDP, you should sketch a figure of the UDP message structure as you observed. It should show the position of the IP header, UDP header, and UDP payload. Within the UDP header, show the position and size of each UDP field you can observe using Wireshark. Your figure can simply show the message as a long, thin rectangle.

Try not to look at the figure of a UDP segment in the answer on next page. To work out sizes, observe that when you click on a protocol block in the middle panel (the block itself, not the "+" expander) then Wireshark will highlight the bytes it corresponds to in the packet in the lower panel and display the length at the bottom of the window.

By looking at the details of the UDP messages in your trace, answer these questions:

1. *What does the Length field include? The UDP payload, UDP payload and UDP header, or UDP payload, UDP header, and lower layer headers?*

2. *How long in bits is the UDP checksum?*

3. *How long in bytes is the entire UDP header?*

Step 4: UDP Usage

To complete our understanding of UDP, we will look at how UDP is used in practice as a transport by applications. Beginning with IP, the next lower protocol layer, there are several issues we can consider. A first issue is how IP knows that the next higher protocol layer is UDP. The answer is that there is a Protocol field in the IP header that contains this information.

1. *Give the value of the IP Protocol field that identifies the upper layer protocol as UDP.*

A second issue is how UDP messages are typically addressed at the IP layer. You might be surprised to find UDP messages in your trace that neither come from your computer or are sent only to your computer. You can see this by sorting on the Source and Destination columns. The source and destinations will be domain names, if Network layer name resolution is turned, and otherwise IP addresses. (You can toggle this setting using the View menu and selecting Name resolution.) You can find out the IP address of your computer using the "**`ipconfig`**" command (Windows).

The reason you may find UDP messages without your computer's IP address as either the source or destination IP address is that UDP is widely used as part of system protocols. These protocols often send messages to all local computers who are interested in them using broadcast and multicast addresses. In our traces, we find DNS (the domain name system), MDNS (DNS traffic that uses IP multicast), NTP (for time synchronization), NBNS (NetBIOS traffic), DHCP (for IP address assignment), SSDP (a service discovery protocol), STUN (a NAT traversal protocol), RTP (for carrying audio and video samples), and more. Your trace may have other protocols you have not heard about; it is OK, as there are a lot of protocols out there. You can look them up on the web for fun.

2. *Examine the UDP messages and give the destination IP addresses that are used when your computer is neither the source IP address nor the destination IP address.* (If you have only your computer as the source or destination IP address then you may use the supplied trace.)

Finally, let us look at the lengths of typical UDP messages. We know that UDP messages can be as large as roughly 64Kbytes. But as you browse you should see that most UDP messages are much shorter than this maximum, so that UDP messages fit in a single packet.

3. *What is the typical size of UDP messages in your trace?*