Lab 6 UDP

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

Objective

- To see how UDP works.
- Know the format of UDP datagram.
- Know the calculation of UDP checksum.
- Understand the advantages and disadvantages of UDP protocol.

Requirements

You need to install following tools on your computer beforehand:

- **Wireshark:** This lab uses the Wireshark software tool to capture and examine a packet trace. Refer to previous labs for details.
- **Browser:** This lab uses a web browser to find or fetch pages as a workload. Any web browser will do.

Some tasks need to write program to generate UDP messages. I have provided you the source code and **executable binaries** on the course website. You can directly use **the executable binary on Windows**. Or, you can compile your source code by yourself. If so, you need to install MinGW on Windows. (The source code of the program can be easily ported to Linux OS.)

- **MinGW:** MinGW, a contraction of "Minimalist GNU for Windows", is a minimalist development environment for native Microsoft Windows applications. Refer to http://www.mingw.org/ to install the basic runtime environment of MinGW. (You MUST install packages of both “MinGW Base System” and MSYS).

Exercise

Task 1: Capture and Explore Trace of UDP

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 websites. UDP is used by DNS for resolving domain names to IP addresses, so visiting fresh sites will cause DNS traffic to be sent. Pick you 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.

1. Launch Wireshark and start a capture with a filter of "*udp*". (Remember to choose correct interface)
2. Perform some activities that will generate UDP traffic as described above, e.g., browse website or start a short VoIP call.
3. Stop Wireshark and inspect the trace you captured.

   Tips:

   - Different computers are likely to capture different kinds of UDP traffic depending on the network set-up and local activity.
   - The protocol column in Wireshark is likely to show multiple protocols in addition to 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 transport protocol (e.g., UDP) unless Wireshark cannot determine the application protocol.
   - You may capture UDP messages, *sent from your computer*, which have checksum of all 0 and are flagged as *incorrect* by Wireshark. This is because that some Operating Systems leave the checksum blank (zero) for the NIC to compute and fill in when the packet is sent out. This process 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".
4. Check whether there are UDP messages without your computer's IP address as either the source or destination IP address. Examine these 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.

   Tips:

   - The source and destination addresses may be domain names, if "Network layer name resolution" of Wireshark is turned, and otherwise IP addresses. You can toggle this setting using the View menu and selecting Name resolution.
   - You might find that most UDP messages in your trace either come from your computer or are sent only to your computer. However, there may be some UDP messages without your computer's IP address as either the source or destination IP address.
   - The reason 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 ad-dresses. For example, DNS (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. If you are interested, you can search for details for these protocols on the Internet.

- If you have only your computer as the source or destination IP address, you can may use the supplied trace on course website.

Answer the following questions:

1. How does IP know that the next higher protocol layer is UDP?
2. What does the *Length* field of UDP include? And how long in bytes is the entire UDP header?
3. What is the typical size of UDP messages in your trace?
4. How long in bits is the UDP checksum? What should be included when calculating UDP the checksum?
5. Explain why the UDP "Checksum" should include the *pseudo header*?

Task 2: UDP Server and Client

In this task, you will be required to write socket program to setup a UDP server and client. Suppose the UDP server works on port number 60001.

Tips:

- You NEED to choose an *unused* port number for server to avoid confliction. You can use "*netstat -a -n -p UDP*" to check all existing UDP services on local computer.

1. Download and compile the "*udpserver.c*" and "*udpclient.c*" in MinGW. Or you can use the "*udpserver.exe*" and "*udpclient.exe*" that I have compiled for you.
2. Prepare two computers, one for UDP server, another for UDP client.
   Launch Wireshark and start a capture with a filter of "*udp.port==60001"* (port number is the UDP server port) on both computers.
   Tips:
   - Remember to choose the correct interface in Wireshark.
   - If you don't have two computers, you can run both *udpclient.exe* and *udpserver.exe* on a single computer. In this case, UDP packets among them won't be sent out the computer, and you need choose the interface of "**Adapter for loopback traffic capture**", see following figure.

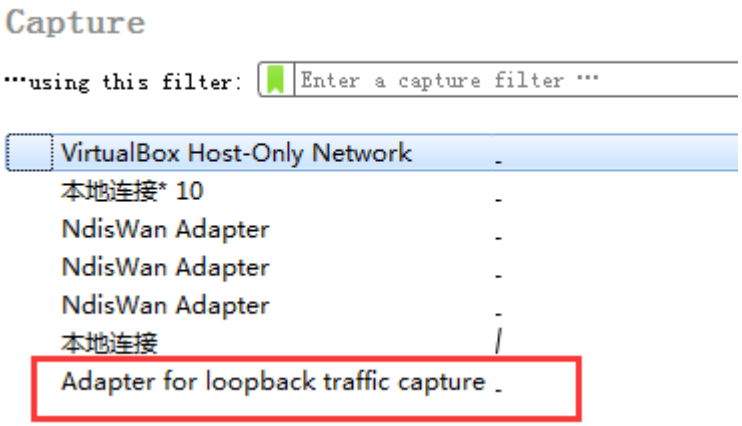

Figure 1. Choose adapter for loopback when running UDP server and client on the same computer

3. Run "*udpserver.exe*" and "*udpclient.exe*" using the following command:
   On the Server side: *udpserver.exe [Port of server, e.g., 60001]*
   On the Client side: *udpclient.exe [IP of server] [Port of server]*
   Tips:
   - You NEED to run these commands as Administrator.
4. Send any random message from UDP client.
5. Stop Wireshark and inspect the trace you captured. If you can not capture any traffic, use the Wireshark traces and program output that we provided for you.

Answer the following questions:

1. Is UDP a connection-oriented protocol? Explain the advantages and disadvantages of this feature.
2. Does the conversation of UDP protocol include acknowledgement datagram? Explain the advantages and disadvantages of this feature.
3. What happened if the client sends a UDP message to a port that is not available on the server?

Tips: The unavailable port means UDP port that are not listed in "*netstat -a -n -p UDP*". You can use Wireshark to capture the traffic for analysis.

Task 3: Explore on your own (Not required in the lab report)

We encourage you to keep exploring on your own.

- You can modify the program to conduct following experiments:
  1) **All-0 Checksum:** Set the checksum of an UDP packet to be all "0", repeat Task 2, and check whether it can be received by the receiver?
  2) **UDP Broadcast:** Set the destination IP address to be "255.255.255.255", and repeat Task 2. Check whether the receiver can receive the UDP packet, and if so, what is the Ethernet MAC address of the packet?
- You might examine the traffic of UDP-based applications to look at packet sizes and loss rates. Voice-over-IP and its companion protocols like RTP

(Real-Time Protocol) are good candidates. Similarly, you might explore streaming and real-time applications to see which use UDP and which use TCP as a transport.

Questions:

1. Can UDP use all "0" as the checksum?
2. In a LAN, a computer sends out a UDP message. If the destination MAC address is the broadcast MAC address and the destination IP address is the IP address of a host within the LAN, what would happen?
3. Briefly explain the ability of UDP on handling errors, e.g., bit flips, packet loss, duplication, and out-of-order?
4. Use your own words to compare the reliability of UDP and IP?
   Tips: both of them are unreliable.