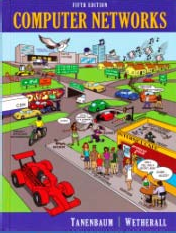
Lab Exercise – Ethernet

# Objective

To explore the details of Ethernet frames. Ethernet is a popular link layer protocol that is covered in §4.3 of your text; modern computers connect to Ethernet switches (§4.3.4) rather than use classic Ethernet (§4.3.2). Review section §4.3 before doing this lab.

# Requirements

**Wireshark**: This lab uses the Wireshark software tool to capture and examine a packet trace. A packet trace is a record of traffic at a location on the network, as if a snapshot was taken of all the bits that passed across a particular wire. The packet trace records a timestamp for each packet, along with the bits that make up the packet, from the lower-layer headers to the higher-layer contents. Wireshark runs on most operating systems, including Windows, Mac and Linux. It provides a graphical UI that shows the sequence of packets and the meaning of the bits when interpreted as protocol headers and data. It col- or-codes packets by their type, and has various ways to filter and analyze packets to let you investigate the behavior of network protocols. Wireshark is widely used to troubleshoot networks. You can down- load it from [www.wireshark.org](http://www.wireshark.org/) if it is not already installed on your computer. We highly recommend that you watch the short, 5 minute video “Introduction to Wireshark” that is on the site.

**ping**: This lab uses “ping” to send and receive messages. ping is a standard command-line utility for checking that another computer is responsive. It is widely used for network troubleshooting and comes pre-installed on Window, Linux, and Mac. While ping has various options, simply issuing the command “ping [www.bing.com](http://www.bing.com/)” will cause your computer to send a small number of ICMP ping requests to the remote computer (here www.bing.com), each of which should elicit an ICMP ping response.

# Step 1: Capture a Trace

*Proceed as follows to capture a trace of ping packets; alternatively you may use a supplie**d trace.* We will use ping simply as an easy way to collect a small trace. Perhaps surprisingly, you can capture a trace for this lab from a computer connected to the Internet using either wired Ethernet or wireless 802.11.

1. Open the trace file from blackboard:
   * trace-ethernet.pcap
2. Observe the following:

|  |
| --- |
|  |

1. Apply the filter “icmp” and apply:

|  |
| --- |
|  |

Figure 3: Trace of ping traffic, showing Ethernet details of the first packet

# Step 2: Inspect the Trace

*Select any packet in the trace (in the top panel) to see details of its structure (in the middle panel**) and the bytes that make up the packet (in the bottom panel).* Now we can inspect the details of the packets. In the figure, we have selected the first packet in the trace. Note that we are using the term “packet” in a loose way. Each record captured by Wireshark more correctly corresponds to a single frame in Ether- net format that carries a packet as its payload; Wireshark interprets as much structure as it can.

*In the middle panel, expand the Ethernet header fields (using the “+” expander or icon) to see their de- tails*. Our interest is the Ethernet header, and you may ignore the higher layer protocols (which are IP and ICMP in this case). You can click on the Ethernet header to see the bytes that correspond to it in the packet highlighted in the bottom panel. We have performed both steps in the figure.

If you are capturing traffic over an 802.11 interface, you may wonder why you have an Ethernet header at all, instead of an 802.11 header. This happens because we asked Wireshark to capture packets in Ethernet format on the capture options (in Figure 2). In this case, the OS software converted the real802.11 header into a pseudo-Ethernet header. We are seeing the pseudo-Ethernet header.

*Compare the fields you see with the picture of an Ethernet frame in Fig. 4-14 of your text.* You will see both similarities and differences:

* There are two kinds of Ethernet shown in your book, IEEE 802.3 and DIX Ethernet. IEEE 802.3 is rare and you are not likely to see it. The frames in the figure and likely your capture are DIX Ethernet, called “Ethernet II” in Wireshark.
* There is no preamble in the fields shown in Wireshark. The preamble is a physical layer mecha- nism to help the NIC identify the start of a frame. It carries no useful data and is not received like other fields.
* There is a destination address and a source address. Wireshark is decoding some of these bits in the OUI (Organizationally Unique Identifier) portion of the address to tell us the vendor of the NIC, e.g., Dell for the source address.
* There is a Type field. For the ping messages, the Ethernet type is IP, meaning the Ethernet pay- load carries an IP packet. (There is no Length field as in the IEEE 802.3 format. Instead, the length of a DIX Ethernet frame is determined by the hardware of a receiving computer, which looks for valid frames that start with a preamble and end with a correct checksum, and passed up to higher layers along with the packet.)
* There is no Data field per se – the data starts with the IP header right after the Ethernet header.
* There is no pad. A pad will be present at the end if the frame would otherwise be less than 64 bytes, the minimum Ethernet frame size.
* There is no checksum in most traces, even though it really does exist. Typically, Ethernet hard- ware that is sending or receiving frames computes or checks this field and adds or strips it. Thus it is simply not visible to the OS or Wireshark in most capture setups.
* There are also no VLAN fields such as the Tag shown in Fig. 4-49. If VLANs are in use, the VLAN tags are normally added and removed by switch ports so they will not be visible at host comput- ers using the network.

# Step 3: Ethernet Frame Structure

*To show your understanding of the Ethernet frame format, draw a figure of the ping message th**at shows the position and size in bytes of the Ethernet header fields.* Your figure can simply show the frame as a long, thin rectangle. The leftmost fields come first in the packet and are sent on the wire first. On this drawing, show the range of the Ethernet header and the Ethernet payload. Add a dashed box at the end to represent the 4-byte checksum; we know it is there even if Wireshark does not show us this field.

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 low- er panel and display the length at the bottom of the window. You may also use the overall packet size shown in the Length column or Frame detail block.

**Turn-in**: Hand in your drawing of an Ethernet frame.



# Step 4: Scope of Ethernet Addresses

Each Ethernet frame carries a source and destination address. One of these addresses is that of your computer. It is the source for frames that are sent, and the destination for frames that are received. But what is the other address? Assuming you pinged a remote Internet server, it cannot be the Ethernet ad- dress of the remote server because an Ethernet frame is only addressed to go within one LAN. Instead, it will be the Ethernet address of the router or default gateway, such as your AP in the case of 802.11. This is the device that connects your LAN to the rest of the Internet. In contrast, the IP addresses in the IP block of each packet do indicate the overall source and destination endpoints. They are your computer and the remote server.

*Draw a figure that shows the relative positions of your computer, the router, and the remote server. Label your computer and the router with their Ethernet addresses. Label your computer and the remote server with their IP addresses. Show where the Ethernet and the rest of the Internet fit on the drawing.*

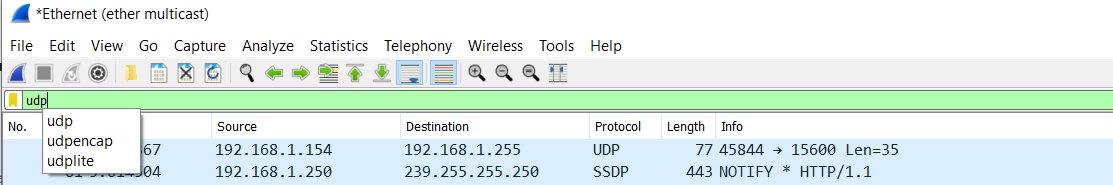
**Turn-in**: Hand in your drawing.



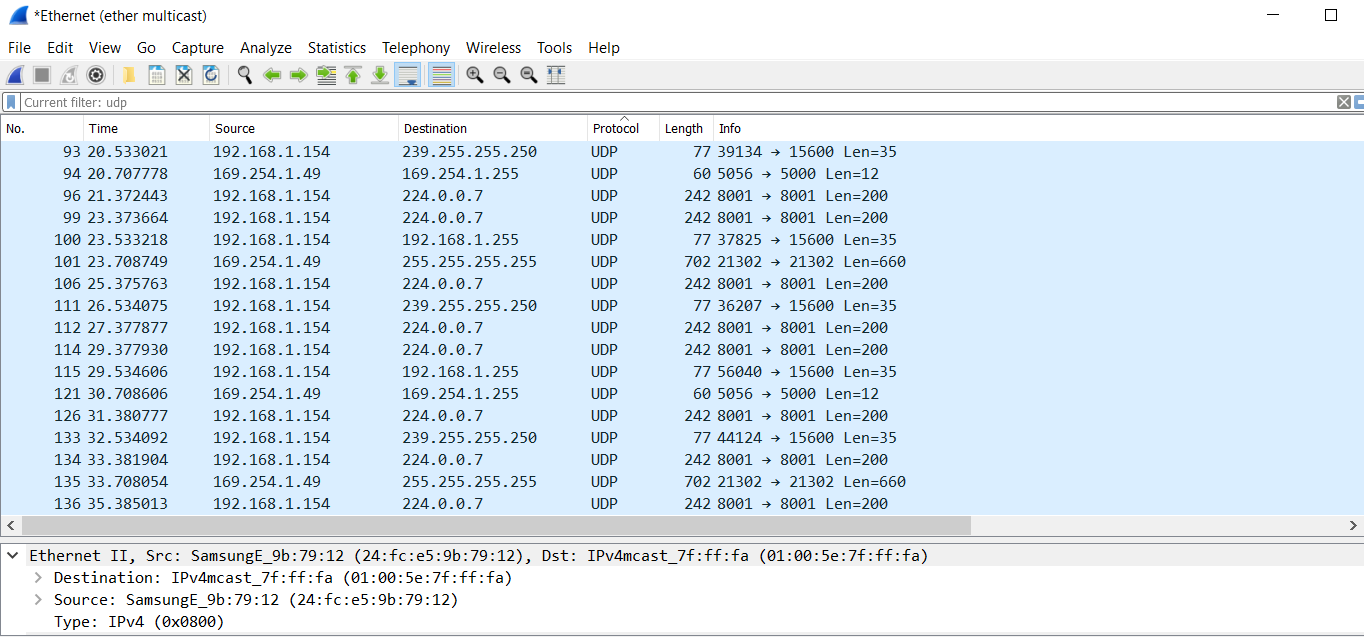
# Step 5: Broadcast Frames

The trace that you gathered above captured unicast Ethernet traffic sent between a specific source and destination, e.g., your computer to the router. It is also possible to send multicast or broadcast Ethernet traffic, destined for a group of computers or all computers on the Ethernet, respectively. We can tell from the address whether it is unicast, multicast, or broadcast. Broadcast traffic is sent to a reserved Ethernet address that has all bits set to “1”. Multicast traffic is sent to addresses that have a “1” in the first bit sent on the wire; broadcast is a special case of multicast. Broadcast and multicast traffic is widely used for discovery protocols, e.g., a packet sent to everyone in an effort to find the local printer.

1. *Start a capture for broadcast and multicast Ethernet frames with a filter of* “ether multicast”,



1. *wait up to 30 seconds to record background traffic, and then stop the capture. If you do not capture any packets with this filter then use the trace that we supplied.* On most Ethernets, there is a steady chatter of background traffic as computers exchange messages to maintain network state, which is why we try to capture traffic without running any other programs. The capture filter of “ether multicast” will capture both multicast and broadcast Ethernet frames, but not regular unicast frames. You may have to wait a little while for these packets to be captured, but on most LANs with multiple computers you will see at least a packet every few seconds.
2. *Examine the multicast and broadcast packets that you captured, looking at the details of the source and destination addresses.* Most likely one has the broadcast Ethernet address, as broadcast frames tend to be more common than multicast frames. Look at a broadcast frame to see what address is used for broadcast by Ethernet. Expand the Ethernet address fields of either broadcast or multicast frames to see which bit is set to distinguish broadcast/multicast or group traffic from unicast traffic.



*Answer the following questions:*

1. *What is the broadcast Ethernet address, written in standard form as Wireshark displays it?*

The broadcast ethernet address is ff:ff:ff:ff:ff:ff; This represents 48 bits of 1s in standard form.

1. *Which bit of the Ethernet address is used to determine whether it is unicast or multicast/broadcast?*

The bit of the Ethernet address used to determine multicast/broadcast is shown as ‘…. ...1 …. …. …. ….’. This is also a ‘1’ in the “low-order” bit f the first byte (address). This is also known as “01:00:00:00:00:00”. This is also the bit that is transmitted on the wire first. Ethernet says the transmission order is the “least significant bit of each byte first”

1. **Turn-in**: Hand in your answers to the above questions.

# Explore on your own (IEEE 802.3)

We encourage you to explore Ethernet on your own once you have completed this lab. As one possibil- ity, recall that there are two types of Ethernet frame, IEEE 802.3 and DIX Ethernet. DIX is common and what we considered above, while IEEE 802.3 is rare. If you are rather lucky, you may see some IEEE

* 1. frames in the trace you have captured. If not, then there are some of these packets in the trace that we supplied. To search for IEEE 802.3 packets, enter a display filter (above the top panel of the Wireshark window) of “llc” (that was lowercase “LLC”) because the IEEE 802.3 format has the LLC proto- col on top of it. LLC is also present on top of IEEE 802.11 wireless, but it is not present on DIX Ethernet.

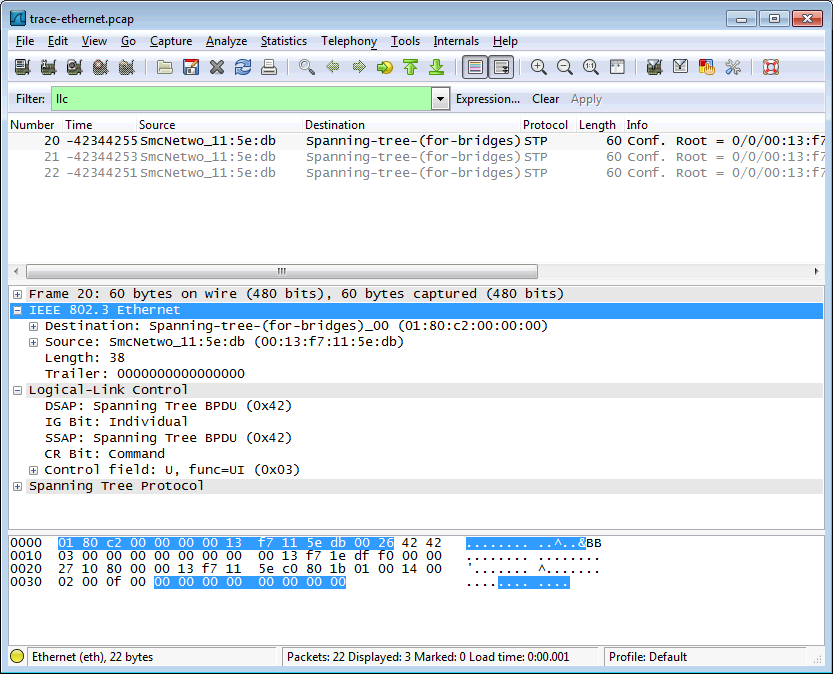


Figure 4: IEEE 802.3 frames with Ethernet and LLC header detail

Have a look at the details of an IEEE 802.3 frame, including the LLC header. The figure shows the details for our trace. Observe that the Type field is now a Length field. In our example, the frame is short enough that there is also padding of zeros identified as a Trailer or Padding. The changes lead to a few questions for you to ponder:

How long are the combined IEEE 802.3 and LLC headers compared to the DIX Ethernet headers? You can use Wireshark to work this out. Note that the Trailer/Padding and Checksum may be shown as part of the header, but they come at the end of the frame.

1. How long are the combined IEEE 802.3 and LLC headers compared to the DIX Ethernet headers? You can use Wireshark to work this out. Note that the Trailer/Padding and Checksum may be shown as part of the header, but they come at the end of the frame.

The IEEE 802.3 header is 14 bytes long, this is the same as DIX ethernet. The LLC adds 3 bytes and totals out at 17 bytes of header.

1. How does the receiving computer know whether the frame is DIX Ethernet or IEEE 802.3? Hint: you may need to both use Wireshark to look at packet examples and read your text near where the Ethernet formats are described.

The DIX Ethernet Type Field and the 802.3 length field are in the same position. Depending on if the value is less than 0x600 aka (1536) then it is seen as a frame length. If it is longer than this value, it is a type value.

1. If IEEE 802.3 has no Type field, then how is the next higher layer determined? Use Wireshark to look for the demultiplexing key.

802.3 adds the LLC header after the IEEE 802.3 header to display the next upper layer protocol. LLC uses a initial byte called DSAP aka (destination service access point)

rather than two bytes in the Type field.

**Turn-in**: Your answers to the above questions. [END]