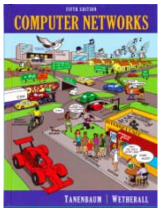
Lab Exercise – Ethernet



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

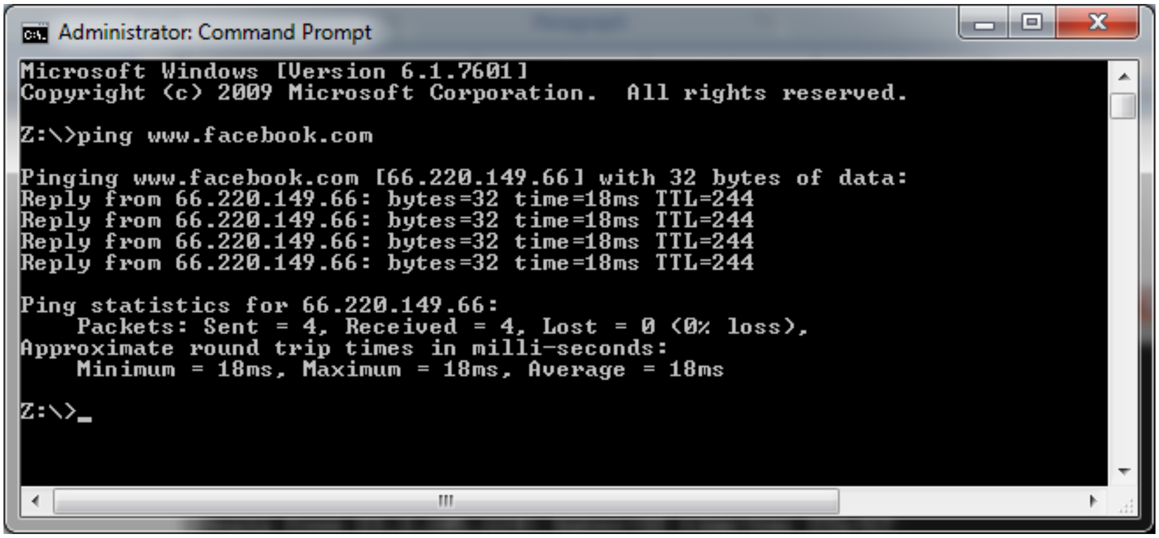
Wireshark: This lab uses the Wireshark software tool to capture and examine a packet trace. A packet trace is a record of traffic at a location on the network, as if a snapshot was taken of all the bits that passed across a particular wire. The packet trace records a timestamp for each packet, along 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 color-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 download it from 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`" 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 supplied 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.

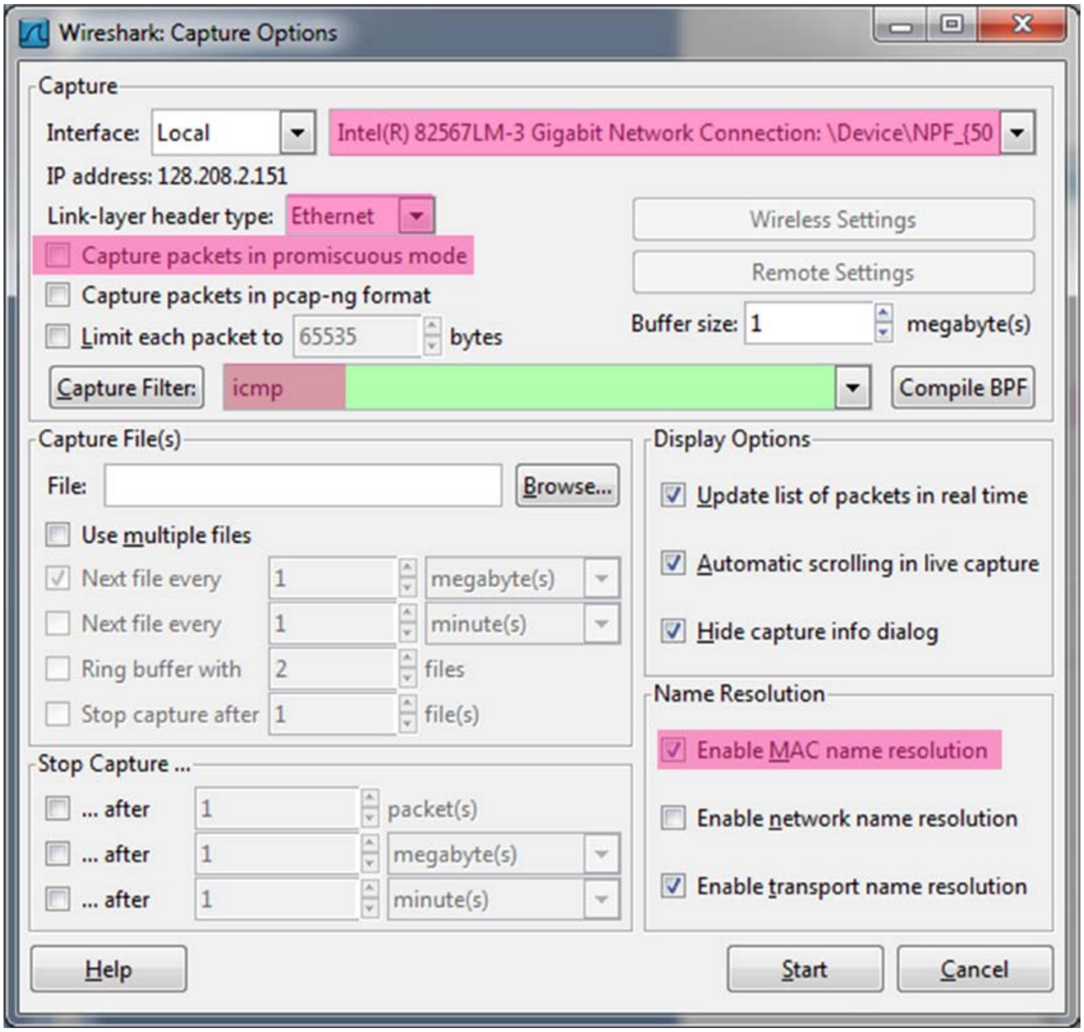
1. *Pick a remote web server or other publicly reachable Internet host and use* `ping` *to send some ping messages and check that it sends replies*. For example, "`ping www.bing.com`". You should see several replies indicating that the pings reached the remote host and were returned. The figure below shows a successful example. Note that some versions of `ping` will continue to bounce messages off of a remote server until you tell the program to stop by signaling it with ^C. If your ping test does not succeed then try another server.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

Z:\>ping www.facebook.com

Pinging www.facebook.com [66.220.149.66] with 32 bytes of data:
Reply from 66.220.149.66: bytes=32 time=18ms TTL=244
Reply from 66.220.149.66: bytes=32 time=18ms TTL=244
Reply from 66.220.149.66: bytes=32 time=18ms TTL=244
Reply from 66.220.149.66: bytes=32 time=18ms TTL=244

Ping statistics for 66.220.149.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 18ms, Average = 18ms

Z:\>_
```

Figure 1: Using `ping` to bounce messages off a remote host

2. *Launch Wireshark and start a capture of Ethernet frames with a filter of* "`icmp`"*, making sure that "enable MAC name resolution" is checked*. The latter will translate Ethernet (MAC) addresses to provide vendor information. Also check that the Link-layer header type pulldown says "Ethernet". Your capture window should be similar to the one pictured below, other than our highlighting. Select the interface from which to capture as the main wired or wireless interface used by your computer to connect to the Internet. If unsure, guess and revisit this step later if your capture is not successful. Uncheck "capture packets in promiscuous mode". This mode is useful to overhear packets sent to/from other computers on broadcast networks. We only want to record packets sent to/from your computer. Leave other options at their default values. The capture filter, if present, is used to prevent the capture of other traffic your computer may send or receive. On Wireshark 1.8, the capture filter box is present directly on the options screen, but on Wireshark 1.9, you set a capture filter by double-clicking on the interface.



Capture

Interface: Local | Intel(R) 82567LM-3 Gigabit Network Connection: \Device\NPF_{50

IP address: 128.208.2.151

Link-layer header type: Ethernet

□ Capture packets in promiscuous mode

□ Capture packets in pcap-ng format

□ Limit each packet to 65535 bytes

Buffer size: 1 megabyte(s)

Capture Filter: icmp

Capture File(s)

File:

□ Use multiple files

☑ Next file every 1 megabyte(s)

□ Next file every 1 minute(s)

□ Ring buffer with 2 files

□ Stop capture after 1 file(s)

Stop Capture ...

□ ... after 1 packet(s)

□ ... after 1 megabyte(s)

□ ... after 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☑ Hide capture info dialog

Name Resolution

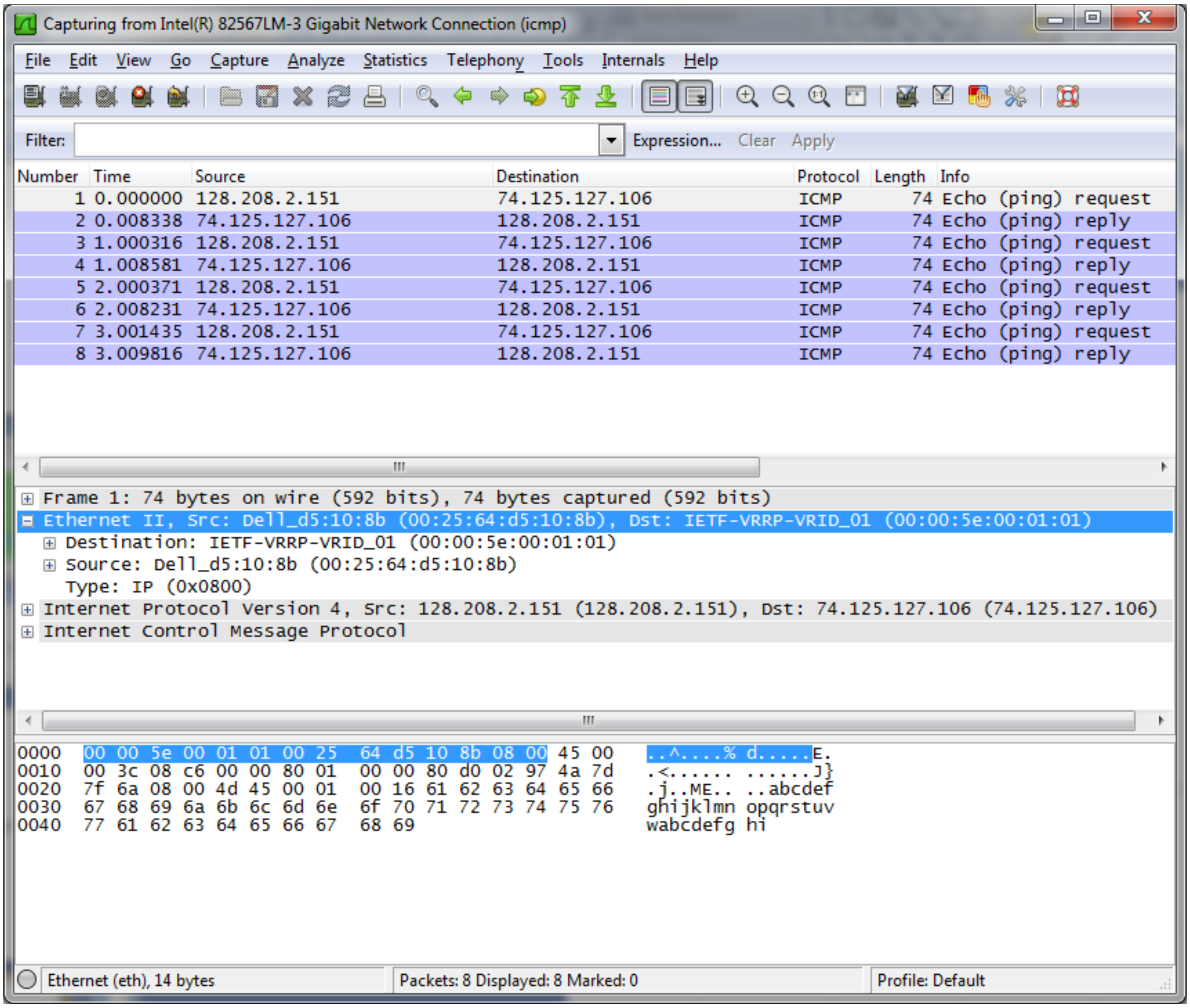
☑ Enable MAC name resolution

□ Enable network name resolution

☑ Enable transport name resolution

Figure 2: Setting the capture options for `ping` traffic

3. *When the capture is started, repeat the* `ping` *command above.* This time, the packets will also be recorded by Wireshark.
4. *After the* `ping` *command is complete, return to Wireshark and use the menus or buttons to stop the trace.* You should now have a short trace similar to that shown in the figure below. If you do not succeed in capturing a trace then use the supplied one. Note that the trace we supply begins with ping messages, and then has other kinds of Ethernet frames.


Capturing from Intel(R) 82567LM-3 Gigabit Network Connection (icmp)
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
Number Time      Source              Destination         Protocol Length Info
     1 0.000000  128.208.2.151       74.125.127.106      ICMP         74 Echo (ping) request
     2 0.008338  74.125.127.106      128.208.2.151       ICMP         74 Echo (ping) reply
     3 1.000316  128.208.2.151       74.125.127.106      ICMP         74 Echo (ping) request
     4 1.008581  74.125.127.106      128.208.2.151       ICMP         74 Echo (ping) reply
     5 2.000371  128.208.2.151       74.125.127.106      ICMP         74 Echo (ping) request
     6 2.008231  74.125.127.106      128.208.2.151       ICMP         74 Echo (ping) reply
     7 3.001435  128.208.2.151       74.125.127.106      ICMP         74 Echo (ping) request
     8 3.009816  74.125.127.106      128.208.2.151       ICMP         74 Echo (ping) reply
```

```
⊞ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
⊟ Ethernet II, Src: Dell_d5:10:8b (00:25:64:d5:10:8b), Dst: IETF-VRRP-VRID_01 (00:00:5e:00:01:01)
  ⊞ Destination: IETF-VRRP-VRID_01 (00:00:5e:00:01:01)
  ⊞ Source: Dell_d5:10:8b (00:25:64:d5:10:8b)
    Type: IP (0x0800)
⊞ Internet Protocol Version 4, Src: 128.208.2.151 (128.208.2.151), Dst: 74.125.127.106 (74.125.127.106)
⊞ Internet Control Message Protocol
```

```
0000  00 00 5e 00 01 01 00 25  64 d5 10 8b 08 00 45 00   ..^....% d.....E.
0010  00 3c 08 c6 00 00 80 01  00 00 80 d0 02 97 4a 7d   .<...... ......J}
0020  7f 6a 08 00 4d 45 00 01  00 16 61 62 63 64 65 66   .j..ME.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Ethernet (eth), 14 bytes | Packets: 8 Displayed: 8 Marked: 0 | Profile: Default

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 Ethernet 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 details. 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 real 802.11 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 mechanism 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 payload 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 hardware 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 computers using the network.

Step 3: Ethernet Frame Structure

To show your understanding of the Ethernet frame format, draw a figure of the ping message that 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 lower 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 address 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.

Start a capture for broadcast and multicast Ethernet frames with a filter of "`ether multicast`", *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.

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.

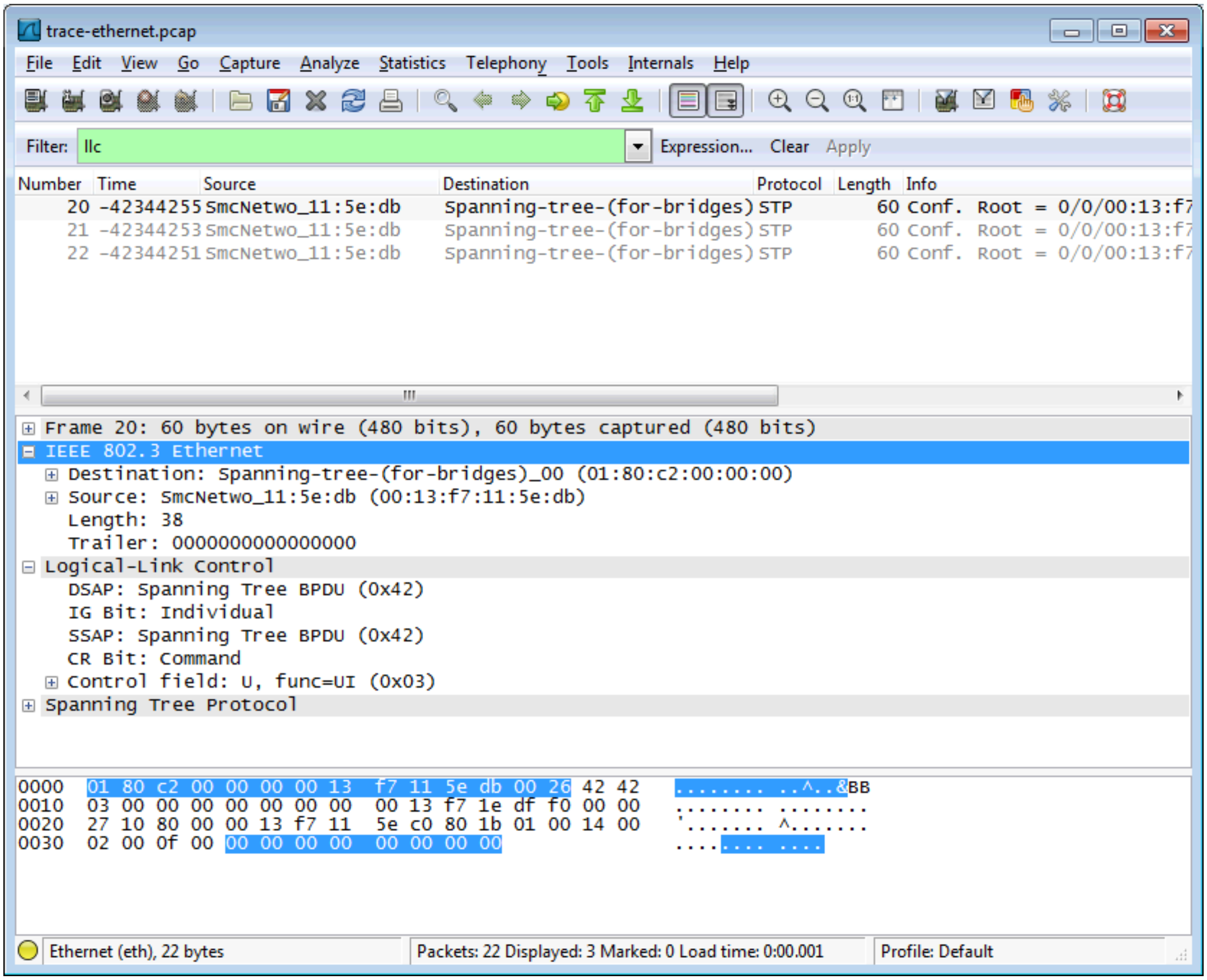
Answer the following questions:

1. *What is the broadcast Ethernet address, written in standard form as Wireshark displays it?*
2. *Which bit of the Ethernet address is used to determine whether it is unicast or multicast/broadcast?*

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 possibility, 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 802.3 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 protocol on top of it. LLC is also present on top of IEEE 802.11 wireless, but it is not present on DIX Ethernet.



Filter: llc

```
Number Time      Source           Destination                  Protocol Length Info
    20 -42344255 SmcNetwo_11:5e:db Spanning-tree-(for-bridges) STP          60 Conf. Root = 0/0/00:13:f7
    21 -42344253 SmcNetwo_11:5e:db Spanning-tree-(for-bridges) STP          60 Conf. Root = 0/0/00:13:f7
    22 -42344251 SmcNetwo_11:5e:db Spanning-tree-(for-bridges) STP          60 Conf. Root = 0/0/00:13:f7
```

```
⊞ Frame 20: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊟ IEEE 802.3 Ethernet
  ⊞ Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  ⊞ Source: SmcNetwo_11:5e:db (00:13:f7:11:5e:db)
    Length: 38
    Trailer: 0000000000000000
⊟ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    IG Bit: Individual
    SSAP: Spanning Tree BPDU (0x42)
    CR Bit: Command
  ⊞ Control field: U, func=UI (0x03)
⊞ Spanning Tree Protocol
```

```
0000  01 80 c2 00 00 00 00 13  f7 11 5e db 00 26 42 42   ........ ..^..&BB
0010  03 00 00 00 00 00 00 00  00 13 f7 1e df f0 00 00   ........ ........
0020  27 10 80 00 00 13 f7 11  5e c0 80 1b 01 00 14 00   '....... ^.......
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ........ ....
```

Ethernet (eth), 22 bytes | Packets: 22 Displayed: 3 Marked: 0 Load time: 0:00.001 | Profile: Default

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:

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.
2. 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.
3. If IEEE 802.3 has no Type field, then how is the next higher layer determined? Use Wireshark to look for the demultiplexing key.

Turn-in: Your answers to the above questions.

[END]