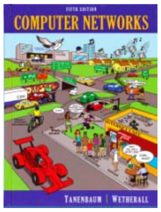
# Lab Exercise – ARP



## Objective

To see how ARP (Address Resolution Protocol) works. ARP is an essential glue protocol that is used to join Ethernet and IP. It is covered in §5.6.4 of your text. Review the text section 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.

**arp**: This lab uses the "`arp`" command-line utility to inspect and clear the cache used by the ARP protocol on your computer. `arp` is installed as part of the operating system on Windows, Linux, and Mac computers, but uses different arguments. It requires administrator privileges to clear the cache.

**ifconfig / ipconfig**: This lab uses the "`ipconfig`" (Windows) or "`ifconfig`" (Mac/Linux) command-line utility to inspect the state of your computer's network interface. `ifconfig/ipconfig` is installed as part of the operating system on Windows, Linux, and Mac computers.

**route / netstat**: This lab uses the "`route`" or "`netstat`" command-line utility to inspect the routes used by your computer. A key route is the default route (or route to prefix 0.0.0.0) that uses the default gateway to reach remote parts of the Internet. Both "`route`" and "`netstat`" are installed as part of the operating system across Windows and Mac/Linux, but there are many variations on the command-line parameters that must be used.

**Browser**: This lab uses a web browser to find or fetch pages as a workload. Any web browser will do.

## Network Setup

We want to observe the ARP protocol in action. Recall that ARP is used to find the Ethernet address that corresponds to a local IP address to which your computer wants to send a packet. A typical example of a local IP address is that of the local router or default gateway that connects your computer to the rest of the Internet. Your computer caches these translations in an ARP cache so that the ARP protocol need only be used occasionally to do the translation. The setup from the viewpoint of your computer is as shown in the example below.

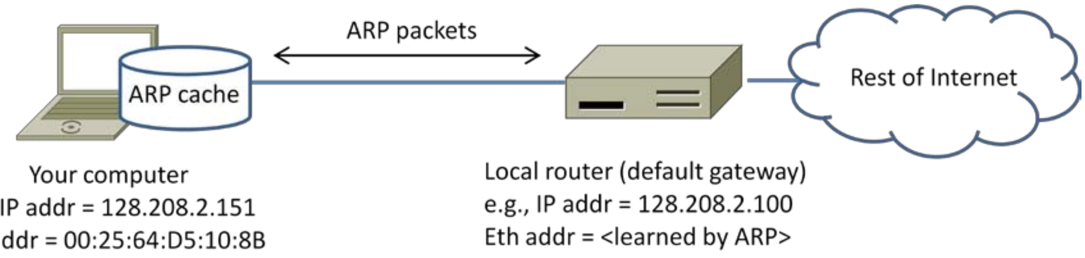
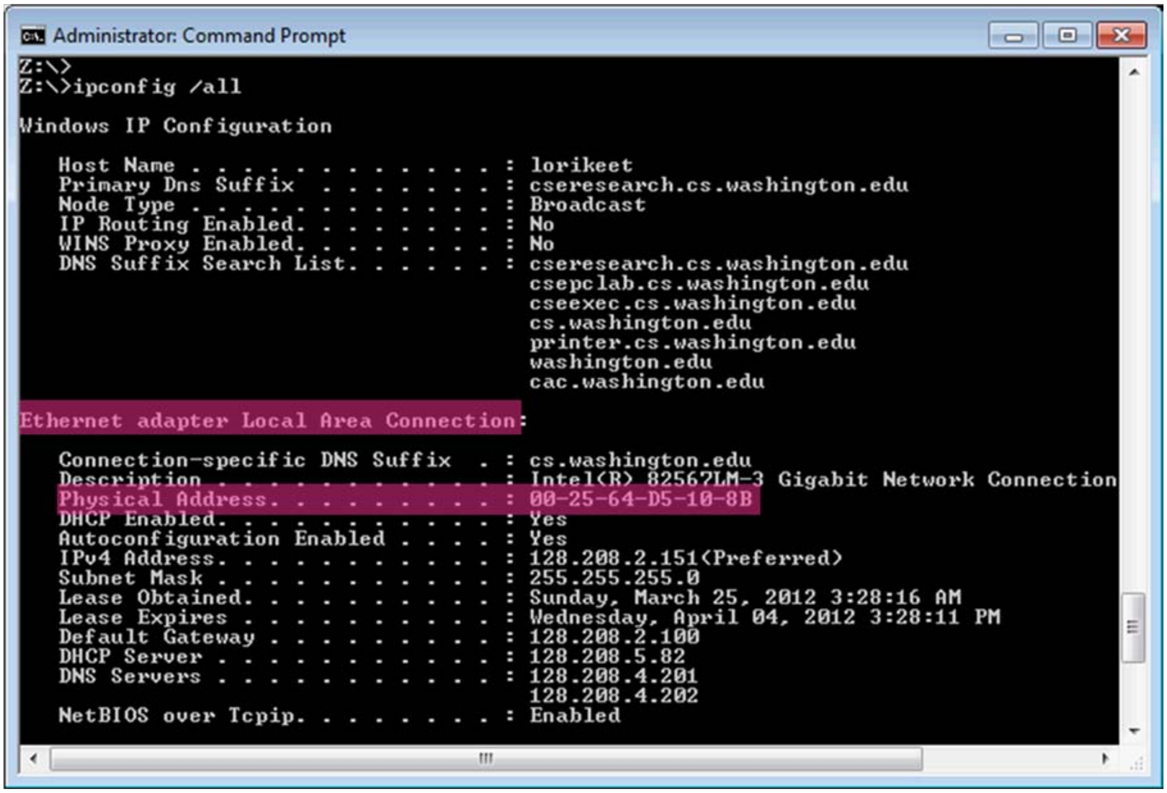

ARP packets
ARP cache
Rest of Internet
Your computer
e.g., IP addr = 128.208.2.151
Eth addr = 00:25:64:D5:10:8B
Local router (default gateway)
e.g., IP addr = 128.208.2.100
Eth addr = <learned by ARP>

Figure 1: Network setup under which we will study ARP

## Step 1: Capture a Trace

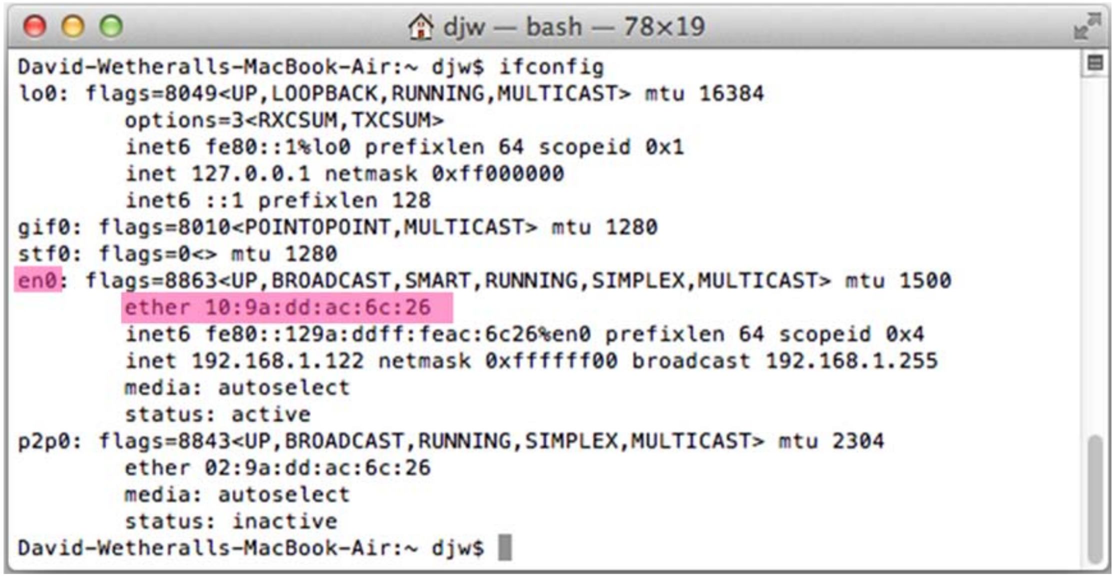
*Proceed as follows to capture a trace of ARP traffic; alternatively, you may use a supplied trace.* To gather ARP packets, we will cause your computer to send traffic to the local router when it does not know the router's Ethernet address – your computer will then use ARP to discover the Ethernet address.

1. *Find the Ethernet address of the main network interface of your computer with the* `ifconfig` / `ipconfig` *command*. You will want to know this address for later analysis. On Windows, bring up a command-line shell and type "`ipconfig /all`". On Mac/Linux, bring up a command-line shell and type "`ifconfig`". Among the output will be a section for the main interface of the computer (likely an Ethernet interface) and its Ethernet address. Common names for the interface are "eth0", "en0", or "Ethernet adapter". Two examples are shown below, with our added highlighting.



```
Z:\>
Z:\>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : lorikeet
   Primary Dns Suffix  . . . . . . . : cseresearch.cs.washington.edu
   Node Type . . . . . . . . . . . . : Broadcast
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : cseresearch.cs.washington.edu
                                       csepclab.cs.washington.edu
                                       cseexec.cs.washington.edu
                                       cs.washington.edu
                                       printer.cs.washington.edu
                                       washington.edu
                                       cac.washington.edu

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : cs.washington.edu
   Description . . . . . . . . . . . : Intel(R) 82567LM-3 Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-25-64-D5-10-8B
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 128.208.2.151(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Sunday, March 25, 2012 3:28:16 AM
   Lease Expires . . . . . . . . . . : Wednesday, April 04, 2012 3:28:11 PM
   Default Gateway . . . . . . . . . : 128.208.2.100
   DHCP Server . . . . . . . . . . . : 128.208.5.82
   DNS Servers . . . . . . . . . . . : 128.208.4.201
                                       128.208.4.202
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

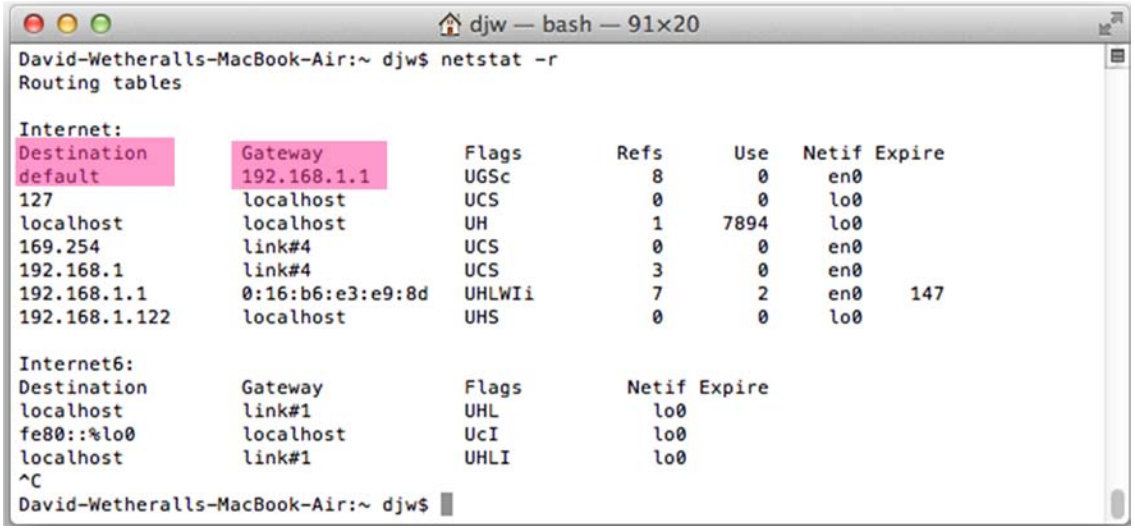
Figure 2: Finding the computer's Ethernet address with `ipconfig` (Windows)



```
David-Wetheralls-MacBook-Air:~ djw$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 10:9a:dd:ac:6c:26
        inet6 fe80::129a:ddff:feac:6c26%en0 prefixlen 64 scopeid 0x4
        inet 192.168.1.122 netmask 0xffffff00 broadcast 192.168.1.255
        media: autoselect
        status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        ether 02:9a:dd:ac:6c:26
        media: autoselect
        status: inactive
David-Wetheralls-MacBook-Air:~ djw$
```

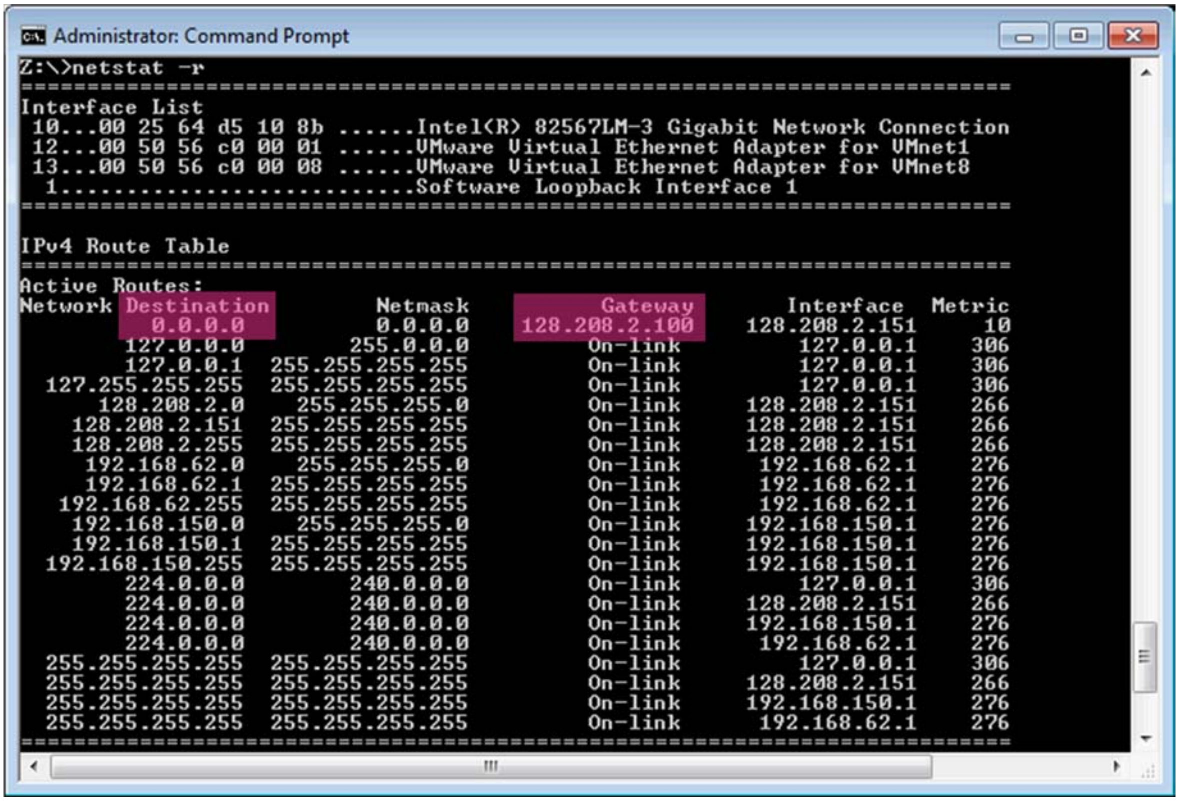
Figure 3: Finding the computer's Ethernet address with `ifconfig` (Mac)

2. *Find the IP address of the local router or default gateway that your computer uses to reach the rest of the Internet using the* `netstat` */* `route` *command.* You should be able to use the `netstat` command ("`netstat -r`" on Windows, Mac and Linux, may require ctrl-C to stop). Alternatively, you can use the route command ("`route print`" on Windows, "`route`" on Linux, "`route -n get default`" on Mac). In either case you are looking for the gateway IP address that corresponds to the destination of default or 0.0.0.0. Two examples are shown below for `netstat`, with our added highlighting.



```
David-Wetheralls-MacBook-Air:~ djw$ netstat -r
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.1.1        UGSc            8        0     en0
127                localhost          UCS             0        0     lo0
localhost          localhost          UH              1     7894     lo0
169.254            link#4             UCS             0        0     en0
192.168.1          link#4             UCS             3        0     en0
192.168.1.1        0:16:b6:e3:e9:8d   UHLWIi          7        2     en0    147
192.168.1.122      localhost          UHS             0        0     lo0

Internet6:
Destination        Gateway            Flags          Netif Expire
localhost          link#1             UHL              lo0
fe80::%lo0         localhost          UcI              lo0
localhost          link#1             UHLI             lo0
^C
David-Wetheralls-MacBook-Air:~ djw$
```

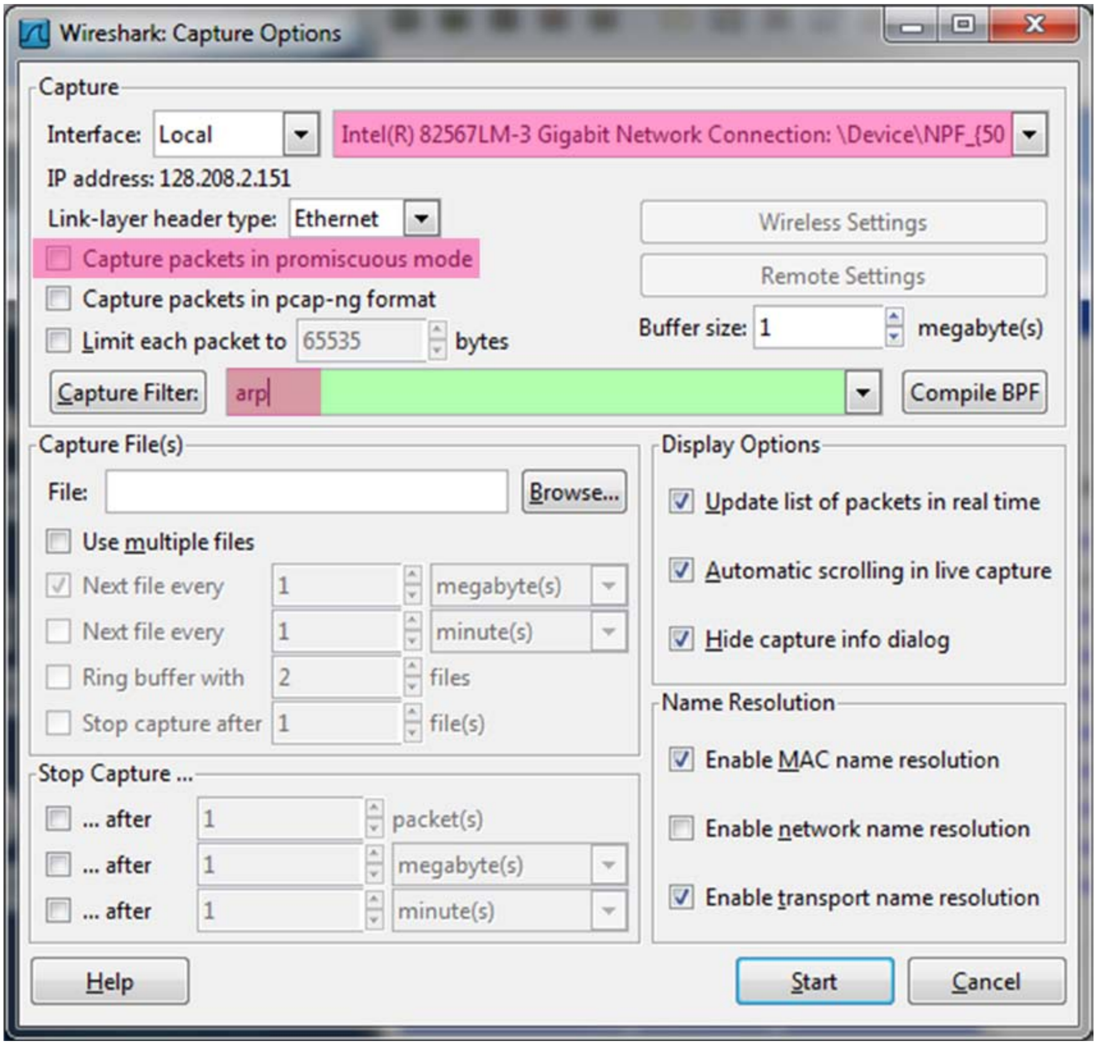
Figure 4: Finding the default gateway IP address with `netstat` (Mac)



```
Z:\>netstat -r
===========================================================================
Interface List
 10...00 25 64 d5 10 8b ......Intel(R) 82567LM-3 Gigabit Network Connection
 12...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1
 13...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    128.208.2.100    128.208.2.151     10
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      128.208.2.0    255.255.255.0         On-link     128.208.2.151    266
    128.208.2.151  255.255.255.255         On-link     128.208.2.151    266
    128.208.2.255  255.255.255.255         On-link     128.208.2.151    266
     192.168.62.0    255.255.255.0         On-link      192.168.62.1    276
     192.168.62.1  255.255.255.255         On-link      192.168.62.1    276
   192.168.62.255  255.255.255.255         On-link      192.168.62.1    276
    192.168.150.0    255.255.255.0         On-link     192.168.150.1    276
    192.168.150.1  255.255.255.255         On-link     192.168.150.1    276
  192.168.150.255  255.255.255.255         On-link     192.168.150.1    276
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link     128.208.2.151    266
        224.0.0.0        240.0.0.0         On-link     192.168.150.1    276
        224.0.0.0        240.0.0.0         On-link      192.168.62.1    276
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link     128.208.2.151    266
  255.255.255.255  255.255.255.255         On-link     192.168.150.1    276
  255.255.255.255  255.255.255.255         On-link      192.168.62.1    276
===========================================================================
```

Figure 5: Finding the default gateway IP address with `netstat` (Windows)

3. *Launch Wireshark and start a capture with a filter of* "`arp`". 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: arp

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 4: Setting up the capture options

4. *When the capture is started, use the "`arp`" command to clear the default gateway from the ARP cache.* Using the command "`arp –a`" will show you the contents of the ARP cache as a check that you can run "`arp`". You should see an entry for the IP address of the default gateway. To clear this entry, use the arp command with different arguments ("`arp –d`" on Windows, "`arp -d –a`" on Mac, "`arp –d xx.xx.xx.xx`" where xx.xx.xx.xx is the IP address of the default gateway on Linux). This usage of `arp` will need administrator privileges to run, so you may run as a privileged user on Windows or issue "`sudo arp –d xx.xx.xx.xx`" on Linux/Mac. Note that the command should run without error but the ARP entry may not appear to be cleared if you check with "`arp –a`". This is because your computer will send ARP packets to repopulate this entry as soon as you need to send a packet to a remote IP address, and that can happen very quickly due to background activity on the computer.
5. *Now that you have cleared your ARP cache, fetch a remote page with your Web browser*. This will cause ARP to find the Ethernet address of the default gateway so that the packets can be sent. These ARP packets will be captured by Wireshark. You might clear the ARP cache and fetch a document a couple of times. Hopefully there will also be other ARP packets sent by other

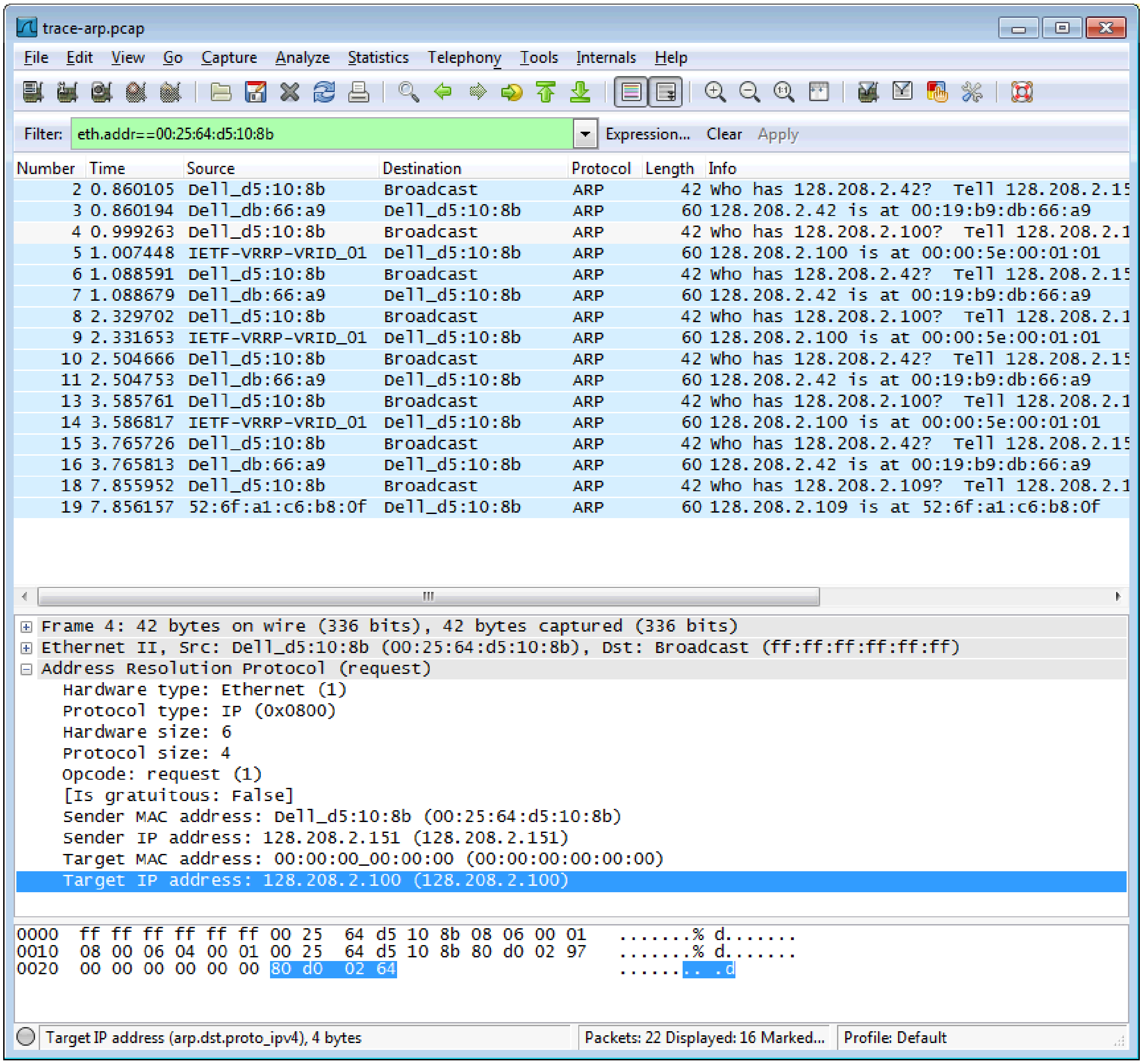
computers on the local network that will be captured. These packets are likely to be present if there are other computers on your local network. In fact, if you have a busy computer and extensive local network then you may capture many ARP packets. The ARP traffic of other computers will be captured when the ARP packets are sent to the broadcast address, since in this case they are destined for all computers including the one on which you are running Wireshark. Because ARP activity happens slowly, you may need to wait up to 30 seconds to observe some of this background ARP traffic.

6. *Once you have captured some ARP traffic, stop the capture.* You will need the trace, plus the Ethernet address of your computer and the IP address of the default gateway for the next steps.

## Step 2: Inspect the Trace

Now we can look at an ARP exchange! Since there may be many ARP packets in your trace, we'll first narrow our view to only the ARP packets that are sent directly from or to your computer.

*Set a display filter for packets with the Ethernet address of your computer.* You can do this by entering an expression in the blank "Filter:" box near the top of the Wireshark window and clicking "Apply". The filter to enter depends on your Ethernet address. For example, if your Ethernet address is 01:02:03:04:05:06 then enter a filter expression of "`eth.addr==01:02:03:04:05:06`". Note the double equal sign. If you are using the supplied trace, it comes with an additional text file giving the Ethernet address and default gateway IP address. After applying this filter your capture should look something like the figure below, in which we have expanded the ARP protocol details.


trace-arp.pcap
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

Filter: eth.addr==00:25:64:d5:10:8b

```
Number Time     Source             Destination    Protocol Length Info
     2 0.860105 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.42?  Tell 128.208.2.15
     3 0.860194 Dell_db:66:a9      Dell_d5:10:8b  ARP          60 128.208.2.42 is at 00:19:b9:db:66:a9
     4 0.999263 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.100?  Tell 128.208.2.1
     5 1.007448 IETF-VRRP-VRID_01  Dell_d5:10:8b  ARP          60 128.208.2.100 is at 00:00:5e:00:01:01
     6 1.088591 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.42?  Tell 128.208.2.15
     7 1.088679 Dell_db:66:a9      Dell_d5:10:8b  ARP          60 128.208.2.42 is at 00:19:b9:db:66:a9
     8 2.329702 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.100?  Tell 128.208.2.1
     9 2.331653 IETF-VRRP-VRID_01  Dell_d5:10:8b  ARP          60 128.208.2.100 is at 00:00:5e:00:01:01
    10 2.504666 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.42?  Tell 128.208.2.15
    11 2.504753 Dell_db:66:a9      Dell_d5:10:8b  ARP          60 128.208.2.42 is at 00:19:b9:db:66:a9
    13 3.585761 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.100?  Tell 128.208.2.1
    14 3.586817 IETF-VRRP-VRID_01  Dell_d5:10:8b  ARP          60 128.208.2.100 is at 00:00:5e:00:01:01
    15 3.765726 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.42?  Tell 128.208.2.15
    16 3.765813 Dell_db:66:a9      Dell_d5:10:8b  ARP          60 128.208.2.42 is at 00:19:b9:db:66:a9
    18 7.855952 Dell_d5:10:8b      Broadcast      ARP          42 Who has 128.208.2.109?  Tell 128.208.2.1
    19 7.856157 52:6f:a1:c6:b8:0f  Dell_d5:10:8b  ARP          60 128.208.2.109 is at 52:6f:a1:c6:b8:0f
```

```
⊞ Frame 4: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
⊞ Ethernet II, Src: Dell_d5:10:8b (00:25:64:d5:10:8b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: False]
    Sender MAC address: Dell_d5:10:8b (00:25:64:d5:10:8b)
    Sender IP address: 128.208.2.151 (128.208.2.151)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 128.208.2.100 (128.208.2.100)
```

```
0000  ff ff ff ff ff ff 00 25  64 d5 10 8b 08 06 00 01   .......% d.......
0010  08 00 06 04 00 01 00 25  64 d5 10 8b 80 d0 02 97   .......% d.......
0020  00 00 00 00 00 00 80 d0  02 64                     ........ .d
```

Target IP address (arp.dst.proto_ipv4), 4 bytes | Packets: 22 Displayed: 16 Marked... | Profile: Default

Figure 5: Capture of ARP packets, showing details of a request

*Find and select an ARP request for the default gateway and examine its fields.* There are two kinds of ARP packets, a request and a reply, and we will look at each one in turn. The Info line for the request will start with “Who has …”. You want to look for one of these packets that asks for the MAC address of the default gateway, e.g., “Who has xx.xx.xx.xx …” where xx.xx.xx.xx is your default gateway. You can click on the + expander or icon for the Address Resolution Protocol block to view the fields:

- Hardware and Protocol type are set to constants that tell us the hardware is Ethernet and the protocol is IP. This matches the ARP translation from IP to Ethernet address.
- Hardware and Protocol size are set to 6 and 4, respectively. These are the sizes of Ethernet and IP addresses in bytes.
- The opcode field tells us that this is a request.

- Next come the four key fields, the sender MAC (Ethernet) and IP and the target MAC (Ethernet) and IP. These fields are filled in as much as possible. For a request, the sender knows their MAC and IP address and fills them in. The sender also knows the target IP address – it is the IP address for which an Ethernet address is wanted. But the sender does not know the target MAC address, so it does not fill it in.

*Next, select an ARP reply and examine its fields*. The reply will answer a request and have an Info line of the form "xx.xx.xx.xx is at yy:yy:yy:yy:yy:yy":

- The Hardware and Protocol type and sizes are as set as before.
- The opcode field has a different value that tells us that this is a reply.
- Next come the four key fields, the sender MAC (Ethernet) and IP and the target MAC (Ethernet) and IP just as before. These fields are reversed from the corresponding request, since the old target is the new sender (and vice versa). The fields should now be all filled in since both computers have supplied their addresses.

## Step 3: ARP request and reply

*To show your understanding of an ARP exchange, draw a figure that shows the ARP request and reply packets sent between your computer and the default gateway. Make it for the case we examined of your computer doing an ARP for the default gateway. Label one packet the request and the other the reply. Give the sender and target MAC and IP addresses for each packet; you can use Wireshark to inspect the packets to get these values. Finally, circle the sought after Ethernet address on your drawing to show where it comes from in the exchange.*

**Turn-in**: Hand in your drawing of the ARP exchange.

## Step 4: Details of ARP over Ethernet

*To look at further details of ARP, examine an ARP request and ARP reply to answer these questions:*

1. *What opcode is used to indicate a request? What about a reply?*
2. *How large is the ARP header for a request? What about for a reply?*
3. *What value is carried on a request for the unknown target MAC address?*

ARP packets are carried in Ethernet frames, and the values of the Ethernet header fields are chosen to support ARP. For instance, you may wonder how an ARP request packet is delivered to the target computer so that it can reply and tell the requestor its MAC address. The answer is that the ARP request is (normally) broadcast at the Ethernet layer so that it is received by all computers on the local network including the target. Look specifically at the destination Ethernet address of a request: it is set to `ff:ff:ff:ff:ff:ff`, the broadcast address. So the target receives the request and recognizes that it is the intended recipient of the message; other computers that receive the request know that it is not meant for them. Only the target responds with a reply. However, anyone who receives an ARP packet can learn a mapping from it: the sender MAC and sender IP pair.

*Examine an ARP request and reply to answer these questions:*

4. *What Ethernet Type value which indicates that ARP is the higher layer protocol?*
5. *Is the ARP reply broadcast (like the ARP request) or not?*

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

## Explore on your own

We encourage you to explore ARP on your own once you have completed this lab. One suggestion is to look at other ARP packets that may have been recorded in your trace; we only examined an ARP request by your computer and the ARP reply from the default gateway.

*To see if there is other ARP activity, make sure to clear any Ethernet address filter that is set.* Other ARP packets may exhibit any of the following kinds of behavior for you to explore:

- ARP requests broadcast by other computers. The other computers on the local network are also using ARP. Since requests are broadcast, your computer will receive their requests.
- ARP replies sent by your computer. If another computer happens to ARP for the IP address of your computer, then your computer will send an ARP reply to tell it the answer.
- Gratuitous ARPs in which your computer sends a request or reply about itself. This is helpful when a computer or link comes up to make sure that no-one else is using the same IP address. Gratuitous ARPs have the same sender and target IP address, and they have an Info field in Wireshark that identified them as gratuitous.
- Other ARP requests sent by your computer and the corresponding ARP reply. Your computer may need to ARP for other hosts besides the default gateway after you flush its ARP cache.

[END]