Lab 2 IPv4

IP is the network layer protocol used throughout the Internet. We will examine IP version 4 in this lab, since it is ubiquitously deployed, while the IP version 6 is partly deployed. IP is covered in §5.7.1 to §5.7.3 of your textbook. Review these sections before doing this lab.

Objective

- To learn about the details of IP (Internet Protocol);
- Know the format of IP packet;
- Know how to calculate IP checksum.

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.
- ***wget/curl*:** This lab uses *wget* (Linux and Windows) and *curl* (Mac) to fetch web resources. Refer to previous labs for details (Lab 0).
- ***Traceroute / tracert*:** This lab uses "*traceroute*" to find the router level path from your computer to a remote Internet host. traceroute is a standard command-line utility for discovering the Internet paths that your computer uses. It is widely used for network troubleshooting. It comes pre-installed on Window and Mac, and can be installed using your package manager on Linux. On Windows, it is called "*tracert*". It has various options, but simply issuing the command "*traceroute www.jnu.edu.cn*" will cause your computer to find and print the path to the remote server *www.jnu.edu.cn*.

Exercise

Task 1: Capture a Trace

Proceed as follows to capture a trace for IP. The trace we want to gather is a simple webpage fetched from a remote server. This will cause your computer to send and receive IP packets. Then, you will be required to perform a traceroute to the remote server to find the path it uses over the Internet.

Tips:

- In this lab, we assume you have **IPv4 connectivity**. If such network is unavailable, you may use the supplied trace on course website.
- Web traffic uses HTTP and the protocol layer model is as follows:

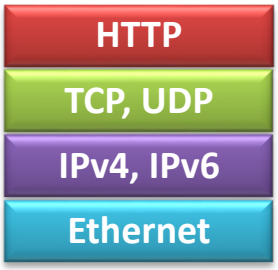

Figure 1: Protocol layers for web traffic

1. Pick a URL at a remote server, e.g., http://www.jnu.edu.cn/ and check that you can fetch the contents with *wget* or *curl*, e.g., "*wget* http://www.jnu.edu.cn/" or "*curl* http://www.jnu.edu.cn/".
   Tips:
   - This will fetch the resource and either write it to a file (*wget*) or to the screen (*curl*). With *wget*, you expect a single response with status code "200 OK".
   - If the fetch does not work, try a different URL.
2. Perform a *traceroute* to the same remote server to check that you can discover information about the network path. On Windows, type, e.g., "*tracert* www.jnu.edu.cn". On Linux / Mac, type, e.g., "*traceroute* www.jnu.edu.cn". A successful example is shown below; save the output as you will need it for later steps.
   Tips:
   - If you are on Linux / Mac and behind a NAT (as most home users or virtual machine users), you can use the "-I" option (that was a capital i) to *traceroute*, e.g., "*traceroute -I* www.jnu.edu.cn". This will cause traceroute to send ICMP probes like *tracert* instead of its usual UDP probes.
   - *traceroute* may take up to a minute to run. Each line in the output shows information about the next IP hop from the computer running *traceroute* towards the target destination.
   - The lines with "*" indicate that there was no response from the network to identity that segment of the Internet path. Some unidentified segments are to be expected. However, if *traceroute* is not working correctly then nearly all the path will be "*". In this case, try a different remote server, or use the supplied traces on the course website.

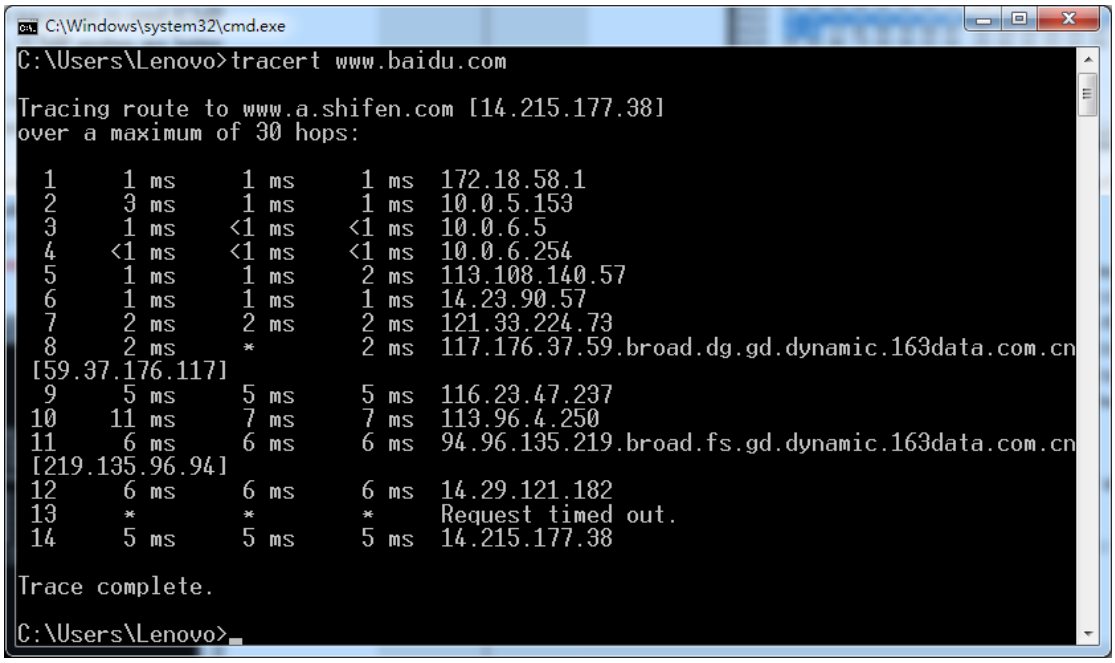

Figure 2: Running traceroute (as *tracert* on Windows)

3. Launch Wireshark and start a capture with a filter of "*tcp port 80*". Make sure to check "*Resolve network names*".

   Tips:

   - We use the filter to record only standard web traffic.
   - Name resolution will translate the IP addresses of the computers sending and receiving packets into names. It will help you to recognize whether the packets are going to or from your computer.

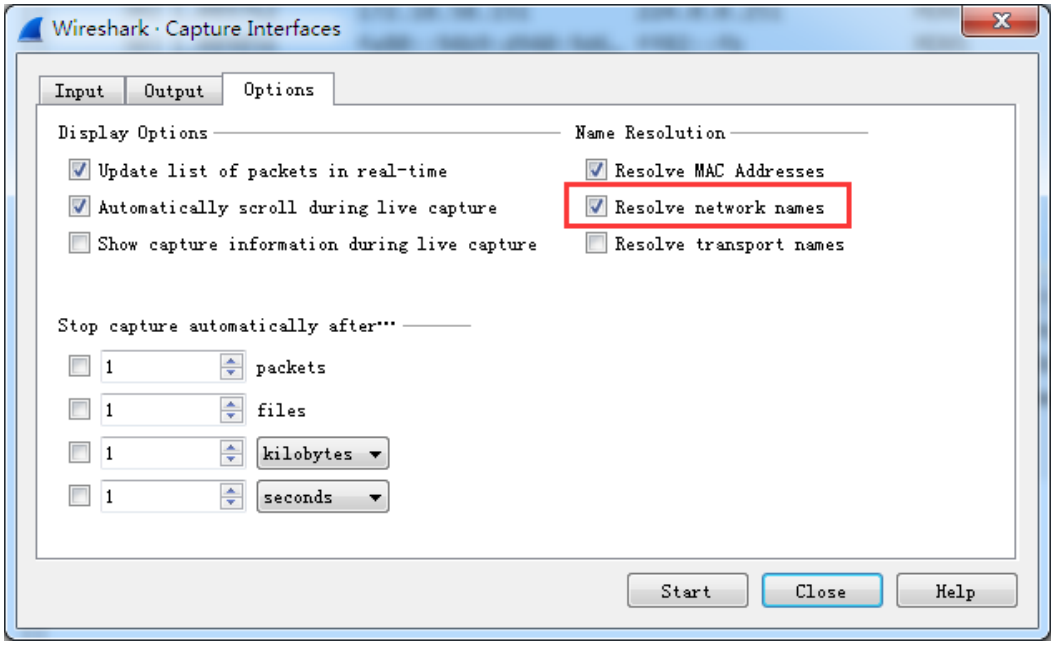

Figure 3: Setting up the capture options

4. After the capture is started, repeat the *wget/curl* command above. This time, the packets will also be recorded by Wireshark.
5. After the command is complete, return to Wireshark and stop the trace. You

should now have a short trace similar to that shown in the figure below, along with the output of a *traceroute* you ran earlier to the corresponding server.

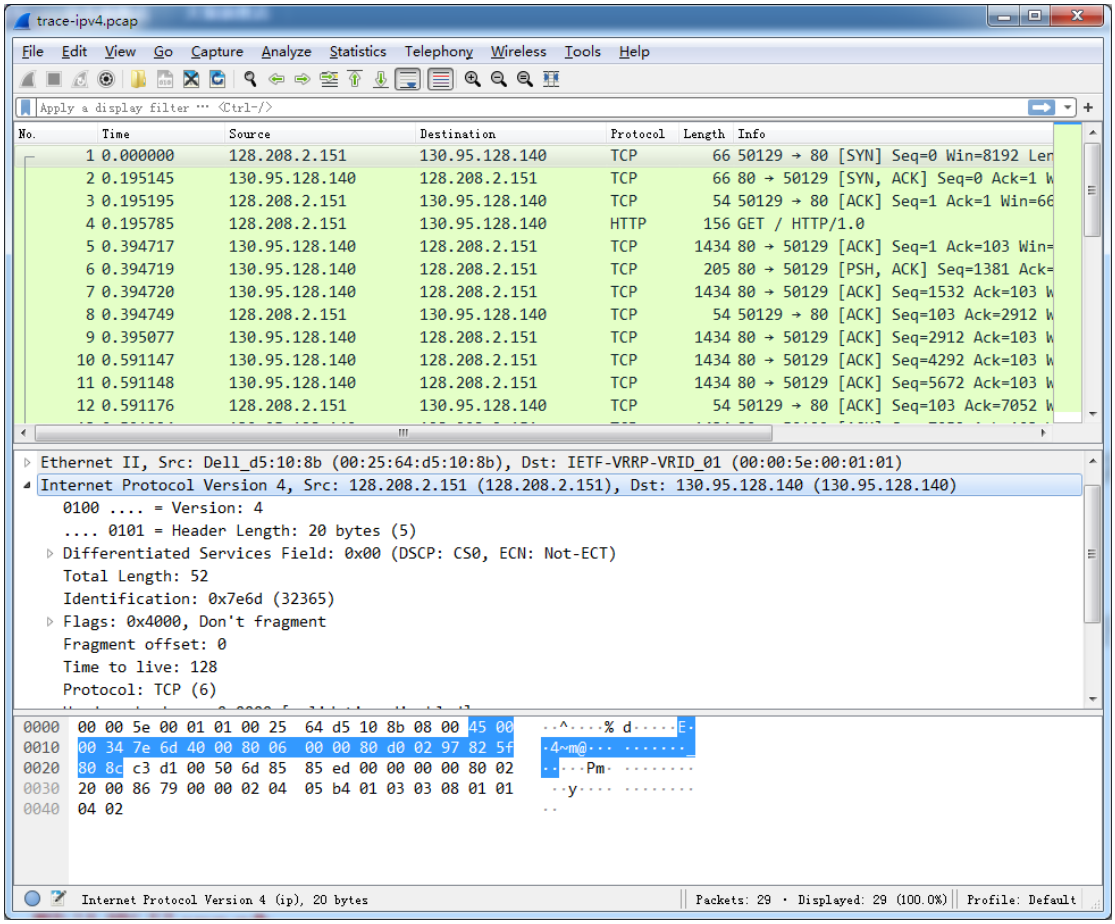

Figure 4: Trace of *wget/curl* traffic showing the details of the IP header

Task 2: Inspect the Trace and IP Packet Structure

1. 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).
2. In the middle panel, expand the IP header fields, and check each of these fields.

   Tips:

   - Our interest is the IP header, and you may ignore the other higher and lower layer protocols (which are TCP and HTTP in this case).
   - You can click on the IP header to see the bytes that correspond to it in the packet highlighted in the bottom panel.

Answer the following questions:

1. What are the IP addresses of your computer and the remote server?
2. Does the *Total Length* field include the IP header plus IP payload, or just the IP payload?
3. How does the value of the *Identification* field change or stay the same for different packets? For instance, does it hold the same value for all packets in a

TCP connection or does it differ for each packet? Is it the same in both directions? Can you see any pattern if the value does change?

4. What is the initial value of the TTL field for packets sent from your computer? Is it the maximum possible value, or some lower value?
5. What is the length of the IP Header and how is this encoded in the header length field?

Task 3: Internet Paths

Using the *traceroute* output, draw a figure of the network path from *your computer* to the *remote server*, as well as the *routers* along the path between them.

- Label the IP addresses for all nodes on the path.
- Number each *router* by their distance on hops from the start of the path.
- If possible, try to label the routers along the path with the name of the real-world organization to which they belong.
  - To do this, you will need to interpret the domain names of the routers given by traceroute.
  - If you are unsure, label the routers with the domain name of what you take to be the organization. Ignore routers for which there is no domain name (or no IP address).
- Ensure your traceroute output is included in your report, e.g., a figure.

Tips:

- You can find the IP address of your computer and the remote server on the packets in the trace that you captured.
- The output of traceroute will tell you the *hop number* for each router.
- If you are using the supplied trace, note that the corresponding *traceroute* output are provided as a separate file.

Task 4: IP Header Checksum

This task will look at the IP header *checksum* calculation by validating a packet.

1. From the Wireshark trace, pick a packet sent from the *remote server* to *your computer*, and ensure that you have a *non-zero value* in the *checksum* field.

   Tips:

   - The checksum value sent over the network will be non-zero, so if you have a zero value it is because of the capture setup.
   - To make this exercise easier, try a packet that has an IP header of 20 bytes, which is the minimum header size when there are no options.
2. Follow these steps to check that the checksum value is correct:
   (1) Divide the header into 10 *two byte (16 bit) words*. Each word will be 4 hexadecimal digits shown in the packet data panel in the bottom of the Wireshark window, e.g., 05 8c.
   (2) Add these 10 words using regular addition. You may the following two ways,

in whatever way you feel convenient:

- You may add them with a hexadecimal calculator (Google to find one).
- Or you can convert them to decimal, add them, and convert them back to hexadecimal.

(3) To compute the 1s complement sum from your addition so far, take any leading digits (beyond the 4 digits of the word size) and add them back to the remainder. For example: 5a432 will become a432 + 5 = a437.

(4) The end result should be 0xffff. This is actually zero in 1s complement form, or more precisely 0xffff is -0 (negative zero) while 0x0000 is +0 (positive zero).

(5) In your calculation, next to each word, please add notes of the IPv4 fields to which it corresponds.

Tips:

- If you cannot get your sum to come out and are sure that the checksum must be wrong, you can get Wireshark to check it. See whether it says "[correct]" already.
- If it does not then use the menus to go to "Preferences", expand "Protocols", choose "IPv4" from the list, and check "validate header checksum". Then Wireshark will check the checksum and tell you if it is correct.

Answer the following questions:

1. Which fields should be included when calculating the checksum of IP packet?

Task 5: Explore on your own (Optional, not required in the lab report)

We also encourage you to explore other aspects of IP on your own once you have completed previous tasks. Some ideas:

- You can use the XB-Ether-Tester (I have uploaded to the course website) to create and send your customized IP packets, and capture them using Wireshark.
- Modern operating systems already include support for IPv6, so you may be able to capture IPv6 traffic on your network. You can also "join the IPv6" backbone by tunneling to an IPv6 provider.
- Learn about tunnels, which wrap an IP packet within another IP header.
- Read about IP geolocation. It is the process of assigning a geographical location to an IP address using measurements or clues from its name administrative databases. Try a geolocation service.
- Learn about IPsec or IP security. It provides confidentiality and authentication for IP packets, and is often used as part of VPNs.

Questions:

1. Explain the difference between MAC and IP addresses in a network? Why should

we use them?

2. What fields will be changed when an IP packet is forwarded by a router?
3. The checksum in IP header doesn't verify the data of IP payload. What are the advantages and disadvantages of such scheme?

Questions on Special IP addresses:

Since most students do not have access to run Wireshark on both sides of a router, we cannot conduct experiments on special IP addresses. Therefore, we prepare several questions for you. You can find answers on the textbook, or research on Google.
(If you are interest, you can also use network simulators, e.g., Packet Tracer, eNSP or GNS3, to conduct experiment for these questions.)

4. What is the scope of limited broadcast address? Does a router forward limited broadcast packets?
5. What is difference between limited broadcast address and direct broadcast address?
6. Suppose there are two computers connected to the same LAN: A and B. Can computer A receive an IP packet which is sent by computer B with the destination of 127.0.0.1? Explain why?