Wireshark IP lab quiz

Due 19 May at 17:00	**Points** 9	**Questions** 9
Available until 26 May at 17:00	**Time limit** None	**Allowed attempts** 5

This quiz was locked 26 May at 17:00.

Attempt history

	Attempt	Time	Score
LATEST	**Attempt 1**	1,263 minutes	9 out of 9

Score for this attempt: **9** out of 9

Submitted 11 May at 20:08

This attempt took 1,263 minutes.

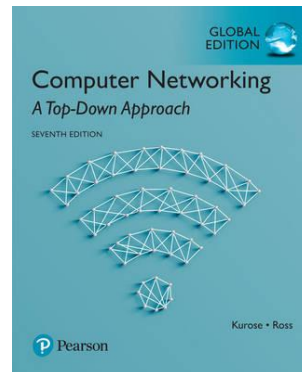### Question 1   1 / 1 pts

Wireshark Lab: IP v7.0

Adapted from: Supplement to *Computer Networking: A Top-Down Approach, 7th ed.,* J.F. Kurose and K.W. Ross

"Tell me and I forget. Show me and I remember. Involve me and I understand." Chinese proverb

© 2005-2016, J.F Kurose and K.W. Ross, All Rights Reserved

In this lab, we'll investigate the IP protocol, focusing on the IP datagram. We'll do so by analyzing a trace of IP datagrams sent and received by an execution of the traceroute program. We'll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

Before beginning this lab, you'll probably want to review sections 1.4.3 in the text[1] and section 3.4 of RFC 2151 [ftp://ftp.rfc-editor.org/in-

notes/rfc2151.txt] to update yourself on the operation of the traceroute program. You'll also want to read Section 4.3 in the text, and probably also have RFC 791 [ftp://ftp.rfc-editor.org/in-notes/rfc791.txt] on hand as well, for a discussion of the IP protocol.

1. ## Capturing packets from an execution of traceroute

In order to generate a trace of IP datagrams for this lab, we'll use the traceroute program to send datagrams of different sizes towards some destination, *X*. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series of datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1 (actually, RFC 791 says that the router must decrement the TTL by *at least* one). If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. As a result of this behavior, a datagram with a TTL of 1 (sent by the host executing traceroute) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram sent with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this manner, the host executing traceroute can learn the identities of the routers between itself and destination *X* by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

We'll want to run traceroute and have it send datagrams of various lengths.

- **Windows.** The tracert program (used for our ICMP Wireshark lab) provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nicer Windows traceroute program is *pingplotter*, available both in free version and shareware versions at **http://www.pingplotter.com** ⬚ **(http://www.pingplotter.com/)** . Download and install *pingplotter*, and test it out by performing a few traceroutes to your favorite sites. You'll need to run the pro version (the 14 day free trial should be enough to complete this quiz!). The size of the ICMP echo request message can be explicitly set in *pingplotter*. The default packet size is 56 bytes. Once *pingplotter*

has sent a series of packets with the increasing TTL values, it restarts the sending process again with a TTL of 1, after waiting *Trace Interval* amount of time. The value of *Trace Interval* and the number of intervals can be explicitly set in *pingplotter*.

- **Linux/Unix/MacOS.** With the Unix/MacOS traceroute command, the size of the UDP datagram sent towards the destination can be explicitly set by indicating the number of bytes in the datagram; this value is entered in the traceroute command line immediately after the name or address of the destination. For example, to send traceroute datagrams of 2000 bytes towards gaia.cs.umass.edu, the command would be:

%traceroute gaia.cs.umass.edu 2000

Do the following:

- Start up Wireshark and begin packet capture *(Capture->Start)* and then press *OK* on the Wireshark Packet Capture Options screen (we'll not need to select any options here).
- If you are using a Windows platform, start up *pingplotter* and enter the name of a target destination in the "Address to Trace Window." Select the menu item *Edit->Options->Packet Options* and enter a value of 56 in the *Packet Size* field and then press OK. Then press the Trace button. Press the pause button once you have 3 Hops (this will likely happen quite quickly). You should see a *pingplotter* window that looks something like this:
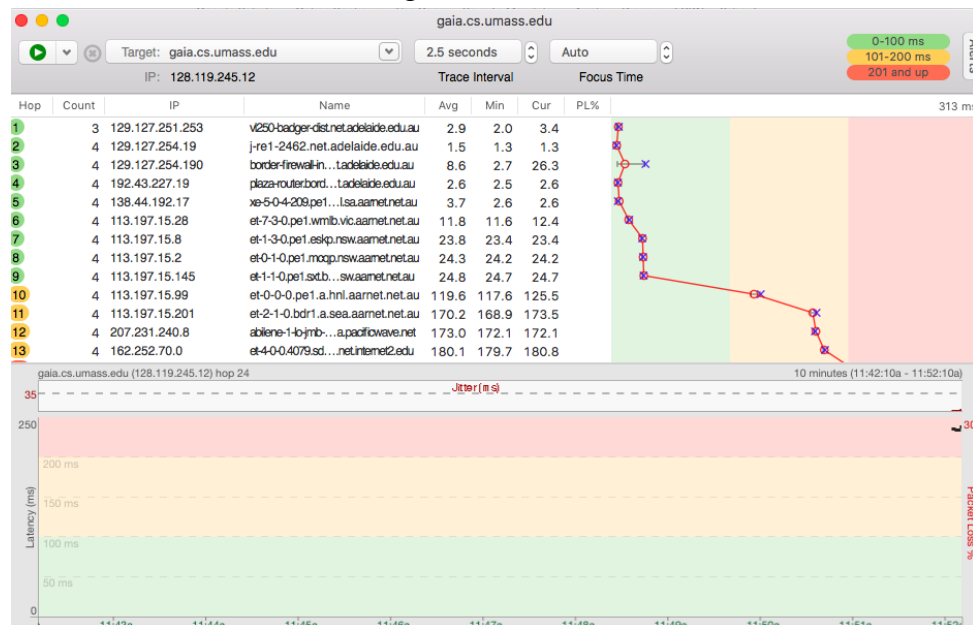
- Next, send a set of datagrams with a longer length, by selecting *Edit->Options->Engine* and enter a value of 2000 in the *Packet Size* field and then press OK. Then press the Resume button. Again, stop the trace after you have 3 traces.

- Finally, send a set of datagrams with a longer length, by selecting *Edit->Options->Engine* and enter a value of 3500 in the *Packet Size* field and then press OK. Then press the Resume button.
  - Stop Wireshark tracing and pingplotter tracing.
- If you are using a Unix or Mac platform, enter three traceroute commands, one with a length of 56 bytes, one with a length of 2000 bytes, and one with a length of 3500 bytes. Use the option -m 3 to only trace the first 3 hops:
  % traceroute -m 3 *hostname*
  - Stop Wireshark tracing.

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author's Windows computers[2]. You may well find it valuable to download this trace even if you've captured your own trace and use it, as well as your own trace, when you explore the questions below.

2. A look at the captured trace

In your trace, you should be able to see the series of ICMP Echo Request (in the case of Windows machine) or the UDP segment (in the case of Unix) sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. In the questions below, we'll assume you are using the trace file, which was captured on a windows machine; so be sure to either download the trace file to answer the questions, or adjust your answer to account for the differences between Windows and Unix).

1. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window.

"In the **IP packet header** of the traceroute trace file, what is the protocol in the upper layer protocol field? (note we want the protocol acronym not the protocol number in this question. For example: HTTP SMTP etc)"

---

[1] References to figures and sections are for the 7th edition of our text, *Computer Networks, A Top-down Approach, 7th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2016.*

[2] Download the zip file **http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip** ⇨ **(http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip)** and extract the file *ip-ethereal-trace-1*. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the ip-ethereal-trace-1 trace file.

Correct!

> ICMP

orrect Answers   *

                            ICMP

Question 2   1 / 1 pts

How many bytes are in the IP header?

Correct!

○ 20 bytes

⌐ correct ⌐

○ 40 bytes

○ 40 Kbytes

○ 10 bytes

Question 3   1 / 1 pts

How many bytes are in the IP payload?

Correct!

○ 64 bytes

> correct

○ 128 bytes

○ 14 bytes

○ 44 bytes

Question 4   1 / 1 pts

The destination host knows that the fragments are part of **same** message by the

Correct!

⦿ IP packet Identification field is the same

> correct

○ More Fragments flag is set

○ Don't fragment flag is not set

○ Fragment offset is greater than 0

Question 5   1 / 1 pts

Next, sort the traced packets according to IP source address by clicking on the *Source* column header; a small downward pointing arrow should appear next to the word *Source*. If the arrow points up, click on the *Source* column header again. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the "details of selected packet header" window. In the "listing of captured packets" window, you should see all of the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your computer) below this first ICMP. Use

the down arrow to move through the ICMP messages sent by your computer.

Which header fields *always* change from one datagram to the next in the series of *unfragmented* ICMP echo request messages in the trace file?

(choose ALL that apply)

☐ version

☐ flags

☐ header length

☐ fragment offset

☐ destination address

Correct!   ☑ header checksum

Correct!   ☑ time to live

☐ protocol

☐ total length

☐ source address

Correct!   ☑ identification

Question 6   1 / 1 pts

2. Fragmentation

Sort the packet listing according to time again by clicking on the *Time* column.

1. Find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* to be 2000. Has that message been fragmented across more than one IP datagram?

[Note: if you find your packet has not been fragmented, you should download the zip file **http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip** ⤷ **(http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip)** and extract the *ip-ethereal-trace-1* packet trace. If your computer has an Ethernet interface, a packet size of 2000 *should* cause fragmentation.**[1]**]

2. Click on the link to the first fragment (each of the fragments is a link in wireshark), note that you will need to remove the icmp filter if you have it set (click the x to remove the filter). Wireshark will not display the individual fragments if you are filtering by icmp.

   What field in the IP header indicates that this is a datagram is the first fragment?

[1] The packets in the *ip-ethereal-trace-1* trace file in **http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip** ⤷ **(http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip)** are all less that 1500 bytes. This is because the computer on which the trace was gathered has an Ethernet card that limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP/IP header data and 1460 bytes of upper-layer protocol payload). This 1500 byte value is the standard maximum length allowed by Ethernet. If your trace indicates a datagram longer 1500 bytes, and your computer is using an Ethernet connection, then Wireshark is reporting the wrong IP datagram length; it will likely also show only one large IP datagram rather than multiple smaller datagrams.. This inconsistency in reported lengths is due to the interaction between the Ethernet driver and the Wireshark software. We recommend that if you have this inconsistency, that you perform this lab using the *ip-ethereal-trace-1* trace file.

○ IP packet Identification field is 0

○ More Fragments flag is set

Correct!

◉ Fragment offset is 0

   ⌐ correct ⌐

○ Don't fragment flag is not set

Question 7　　　　　　　　　　　　　　　1 / 1 pts

What size, in bytes, were the largest IP packets sent (ie at what size did the ICMP echo requests fragment?)

Correct!

1500

Correct Answers

*

1500 bytes

1500

Question 8　　　　　　　　　　　　　　　1 / 1 pts

What high level (above IP) protocol is used to return a response to an ICMP echo request?

Correct!

ICMP

Correct Answers　　ICMP

Question 9　　　　　　　　　　　　　　　1 / 1 pts

In the CS labs, traceroute's UDP/ICMP traffic is blocked by the firewall. The firewall responds with an ICMP message of Destination Unreachable (Communication administratively filtered). What is the ICMP response code number for this response?

Correct!

13

Correct - the response code 13 Communication administratively filtered is returned by ICMP from the firewall.

orrect Answers   13

                        10

Quiz score: **9** out of 9