NTNU
The Norwegian University of
Science and Technology
Department of Telematics

TTM4100
Communication – Services and Networks

Wireshark Lab: IP

The three Wireshark labs in this course are not mandatory, but highly recommended exercises which will help improve your understanding of relevant subjects. Any questions about the exercises can be sent to the same e-mail or be posted on the forum.

Deadline of submission: **N/A**

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 book to update yourself on the operation of the traceroute program. You'll also want to read Section 4.4 in the text book, 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.

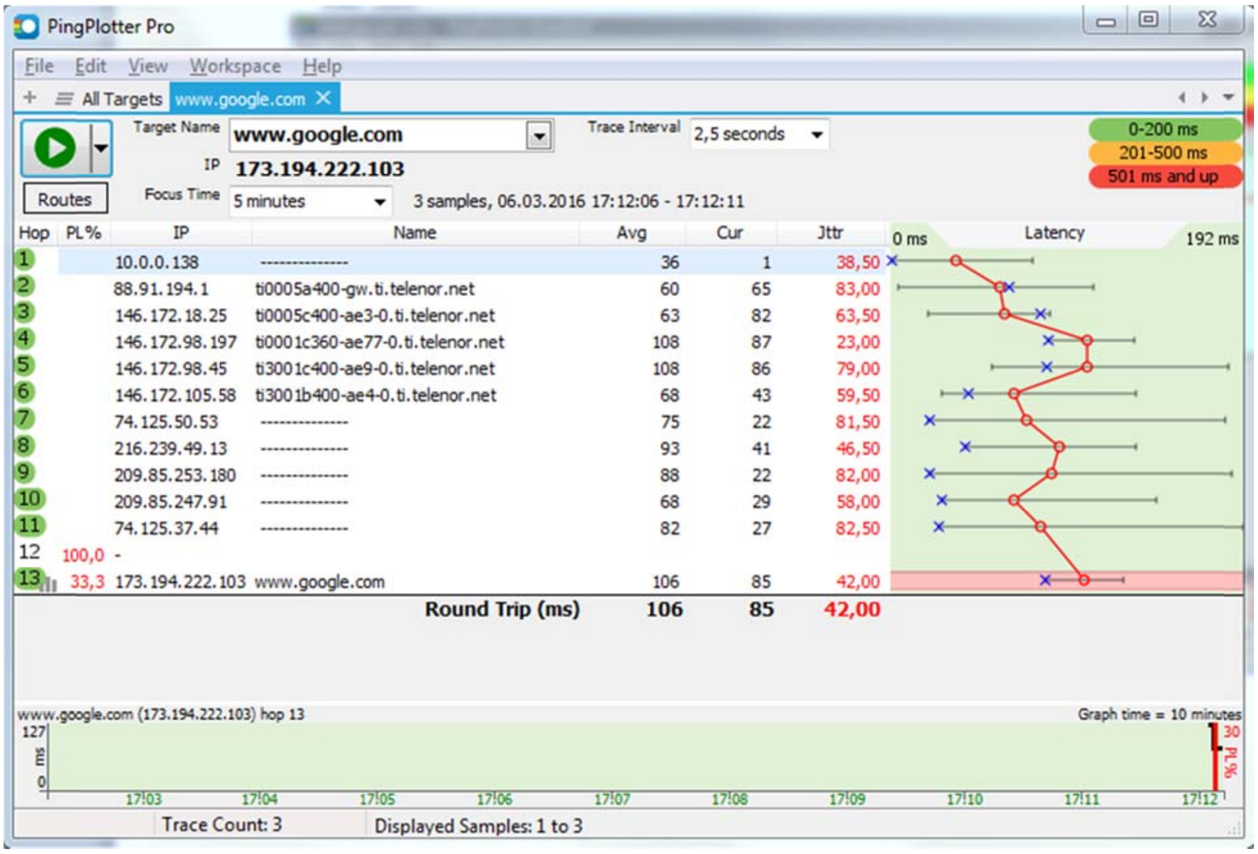
- **Windows.** The tracert program provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nicer Windows traceroute program is *pingplotter*, available both in free version and shareware versions at http://www.pingplotter.com. Download and install *pingplotter*, and test it out by performing a few traceroutes to your favorite sites. The size of the ICMP echo request message can be explicitly set in *pingplotter* by selecting the menu item *Edit-> Options->Engine* and then filling in the *Packet Size* field. 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.** With the Unix 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 item.ntnu.no, the command would be:

```
    %traceroute item.ntnu.no 2000
```

Do the following:

- Close other Internet sessions (browsing and so on) in order to get a Wireshark packet capture with less irrelevant packets.
- 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 "Target Name", e.g. www.google.com. Also, limit the number of traces to 3 by clicking on the arrow next to the "Play" symbol, and selecting "limit trace count to…". Select the menu item *Edit->Options->Engine* and enter a value of 56 in the *Packet Size* field and then press OK. Then press the Trace button. You should see a *pingplotter* window that looks something like this:



Target Name: www.google.com
IP: 173.194.222.103
Trace Interval: 2,5 seconds
Focus Time: 5 minutes
3 samples, 06.03.2016 17:12:06 - 17:12:11

Hop	PL%	IP	Name	Avg	Cur	Jttr
1		10.0.0.138	--------------	36	1	38,50
2		88.91.194.1	ti0005a400-gw.ti.telenor.net	60	65	83,00
3		146.172.18.25	ti0005c400-ae3-0.ti.telenor.net	63	82	63,50
4		146.172.98.197	ti0001c360-ae77-0.ti.telenor.net	108	87	23,00
5		146.172.98.45	ti3001c400-ae9-0.ti.telenor.net	108	86	79,00
6		146.172.105.58	ti3001b400-ae4-0.ti.telenor.net	68	43	59,50
7		74.125.50.53	--------------	75	22	81,50
8		216.239.49.13	--------------	93	41	46,50
9		209.85.253.180	--------------	88	22	82,00
10		209.85.247.91	--------------	68	29	58,00
11		74.125.37.44	--------------	82	27	82,50
12	100,0	-				
13	33,3	173.194.222.103	www.google.com	106	85	42,00
			Round Trip (ms)	**106**	**85**	**42,00**

www.google.com (173.194.222.103) hop 13
Graph time = 10 minutes
Trace Count: 3
Displayed Samples: 1 to 3

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.

Stop Wireshark tracing.

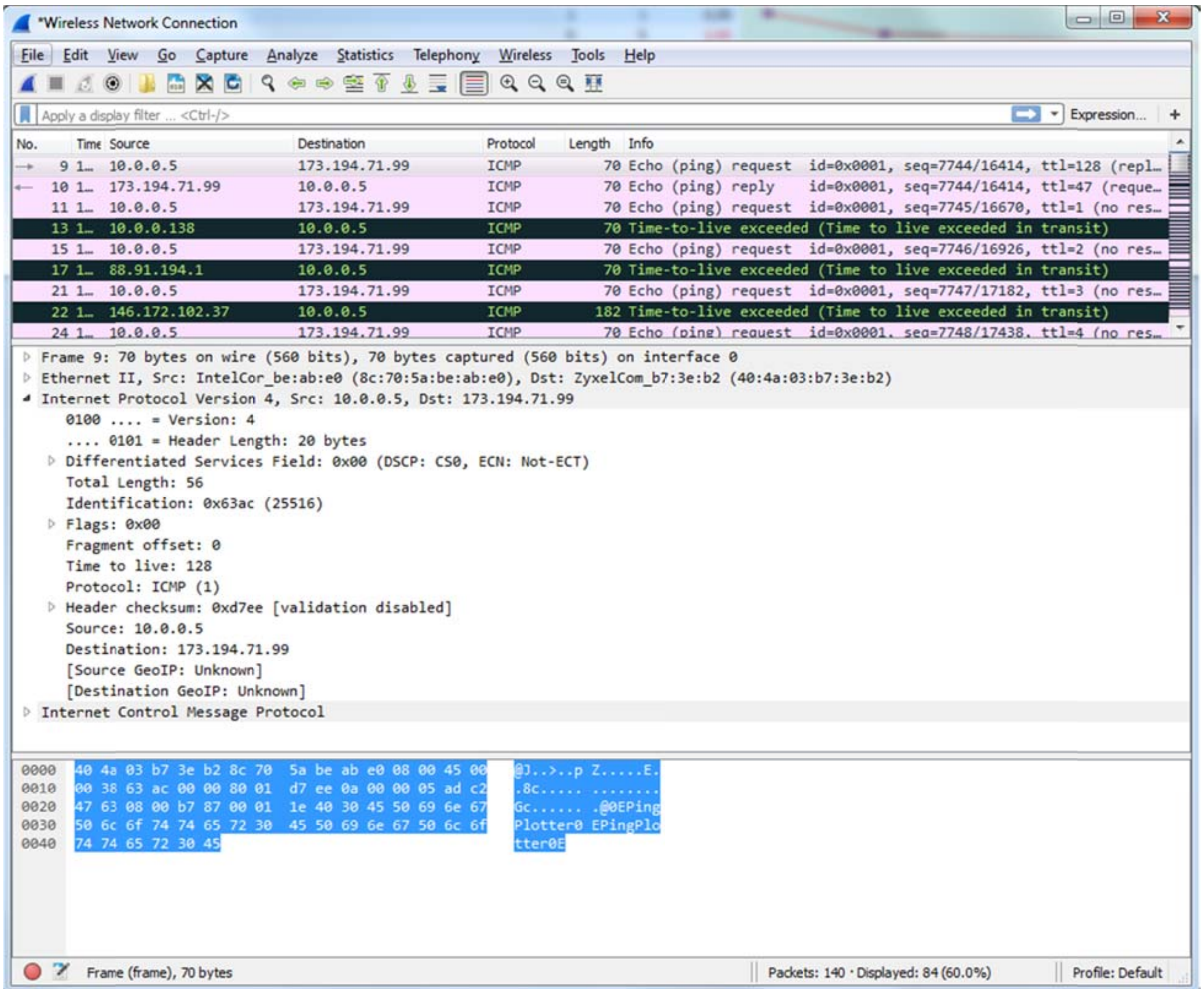
- If you are using a Unix platform, enter two traceroute commands (for example to google.com), one with a length of 56 bytes, and one with a length of 2000 bytes.

  Stop Wireshark tracing.

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 a Windows machine (filter the messages in Wireshark by applying the "icmp" filter); the corresponding questions for the case of a Unix machine should be clear.

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. What is the IP address of your computer?


*Wireless Network Connection
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Frame (frame), 70 bytes
Packets: 140 · Displayed: 84 (60.0%)
Profile: Default

```
No.   Time Source          Destination      Protocol  Length  Info
→  9  1…   10.0.0.5        173.194.71.99    ICMP          70  Echo (ping) request  id=0x0001, seq=7744/16414, ttl=128 (repl…
← 10  1…   173.194.71.99   10.0.0.5         ICMP          70  Echo (ping) reply    id=0x0001, seq=7744/16414, ttl=47 (reque…
  11  1…   10.0.0.5        173.194.71.99    ICMP          70  Echo (ping) request  id=0x0001, seq=7745/16670, ttl=1 (no res…
  13  1…   10.0.0.138      10.0.0.5         ICMP          70  Time-to-live exceeded (Time to live exceeded in transit)
  15  1…   10.0.0.5        173.194.71.99    ICMP          70  Echo (ping) request  id=0x0001, seq=7746/16926, ttl=2 (no res…
  17  1…   88.91.194.1     10.0.0.5         ICMP          70  Time-to-live exceeded (Time to live exceeded in transit)
  21  1…   10.0.0.5        173.194.71.99    ICMP          70  Echo (ping) request  id=0x0001, seq=7747/17182, ttl=3 (no res…
  22  1…   146.172.102.37  10.0.0.5         ICMP         182  Time-to-live exceeded (Time to live exceeded in transit)
  24  1…   10.0.0.5        173.194.71.99    ICMP          70  Echo (ping) request  id=0x0001, seq=7748/17438, ttl=4 (no res…
```

```
▷ Frame 9: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
▷ Ethernet II, Src: IntelCor_be:ab:e0 (8c:70:5a:be:ab:e0), Dst: ZyxelCom_b7:3e:b2 (40:4a:03:b7:3e:b2)
◢ Internet Protocol Version 4, Src: 10.0.0.5, Dst: 173.194.71.99
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x63ac (25516)
  ▷ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (1)
  ▷ Header checksum: 0xd7ee [validation disabled]
    Source: 10.0.0.5
    Destination: 173.194.71.99
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▷ Internet Control Message Protocol
```

```
0000  40 4a 03 b7 3e b2 8c 70  5a be ab e0 08 00 45 00   @J..>..p Z.....E.
0010  00 38 63 ac 00 00 80 01  d7 ee 0a 00 00 05 ad c2   .8c..... ........
0020  47 63 08 00 b7 87 00 01  1e 40 30 45 50 69 6e 67   Gc...... .@0EPing
0030  50 6c 6f 74 74 65 72 30  45 50 69 6e 67 50 6c 6f   Plotter0 EPingPlo
0040  74 74 65 72 30 45                                  tter0E
```

2. How many bytes are in the IP header? How many bytes are in the payload *of the IP datagram*? Explain how you determined the number of payload bytes.

3. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

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.

4. Which fields in the IP datagram *always* change from one datagram to the next within this series of ICMP messages sent by your computer? Which fields stay constant?
5. Which of the fields *must* stay constant? Which fields must change?

Next (with the packets still sorted by source address) find the series of ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router.

6. What is the value in the Identification field and the TTL field?

Fragmentation

Sort the packet listing according to time again by clicking on the *Time* column. To see fragmentation, you will have to remove the “icmp” filter.

7. Find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* in *pingplotter* to be 2000. Has that message been fragmented across more than one IP datagram?
8. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram has been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?