COMPSCI 314 S1T Assignment 1
2009

Department of Computer Science
The University of Auckland

*This assignment contributes to **5%** of your overall course mark. Submit your assignment in **PDF** format to **Assignment Drop Box**. Include all **workings** and **explanations**. Marks will be deducted for ambiguous solutions. Zero marks are awarded if the answers contain no explanation. Also, refer to the Departmental Policy on Cheating on Assignments.*

***Assignment Drop Box** (https://adb.ec.auckland.ac.nz/adb/)*
***Departmental Policy on Cheating on Assignments** (http://www.cs.auckland.ac.nz/CheatingPolicy.php)*

Important: For the questions **Q2, Q3 and Q4**, you must also attach one or two pages of your actual capture files for each question, as proof that you did the work, e.g., output screen-shots. You can use File/Export/File in Wireshark to save a text file.

[Total: 50 marks]

Notes: This assignment is intended to help you understand the way that packets flow across a network, and how various protocols fit on top of one another. *Carefully review the tutorial document before starting the assignment*. It is recommended that you perform the work in one of the Computer Science labs. Some people may prefer to install Wireshark on their own computer and work at home, but the results could be less interesting.

Sample Solutions for markers

Q1. Packet capturing [10 marks]
[Mark allocation: 2, 2, 3, 3]

a) Go to Capture Options in Wireshark and find '*Name Resolution*' (bottom right). Briefly describe and explain the following checklists:

- *Enable MAC name resolution*
- *Enable network name resolution*
- *Enable transport name resolution*
- MAC: resolves a layer 2 address name; the first 3 bytes reveals the manufacturer's name (OID) e.g. 00:21:70:29:6e:84 displays Dell_29:6e:84. The other 3 bytes reveals the manufacturer's specific (model/device) number.
- Network: resolves a layer 3 (IP) address name; Wireshark will attempt to lookup IP address to return a name, e.g. 130.216.33.106 displays www.cs.auckland.ac.nz
  Unlike other name resolution, this list is unchecked by default as this uses a lot of resources to perform a live name lookup.
- Transport: resolves a port number according to IANA port number assignment, e.g. port 80 = HTTP, UDP = 53

b) Explain why a filter is useful.
- displays packets of interest, allows to specify various fields in the packet
- flexible expression (including basic comparison such as equal, less-than, greater-than)

c) What filter string would you use to filter out packets of 'UDP, and TCP port 80, and at least 1200 bytes of TCP segment size, and IP address 2001:4860:b005::68' ? (hint: look at 'Expression' button)
Filter in:
udp and tcp.port == 80 and tcp.len >=1200 and ipv6.addr == 2001:4860:b005::68
Filter out:
!(udp) and !(tcp.port == 80) and !(tcp.len >=1200) and !(ipv6.addr == 2001:4860:b005::68)

Note: udp.port == 80 , tcp.segment >= 1200 is not correct, but okay to accept both.

d) Would your answer (c) work? Will it display some packets?
If yes, what packet type was it? If no, give a better filter example.
Filter in:
No, both UDP and TCP are mutually exclusive, e.g. packet cannot be UDP and TCP at the same time. Appropriate answer would be:
udp or (tcp.port == 80 and tcp.len >=1200 and ipv6.addr == 2001:4860:b005::68)
Filter out:
Yes, Wireshark displays some packets, e.g. ARP packets

Q2. IP packet observation [10 marks]
[Mark allocation: 2, 3, 3, 2]
Use a web browser to visit at least two HTTP web servers, e.g. www.cs.auckland.ac.nz and www.google.com. Capture the full packets that are travelling between you and the web page. You only need to visit once to capture all the packets. Make sure to avoid HTTP status code *304*; this is most likely to happen if you are refreshing the web page. Also, it is best to clear the browser cache before starting.

a) You would observe some DNS packets, which field in IP/UDP packet identifies the corresponding request/response?
Transaction ID number (Note, source/destination IP, source/destination port number is not correct)

b) Did the system use IPv4, IPv6, or a mixture? Can you explain why? (If you want to test for IPv6 access, try http://ipv6.google.com)
Also, list at least two frequently observed packet sizes, e.g. a small size and a large size. Explain why these two different sizes exist, e.g. what is the difference in contents of the two sizes?
Small packet sizes (~66 bytes) show that they contain no application data; indicating that these are most likely the acknowledgement packets.

Bigger packet sizes (~1518 bytes) contain most likely the application data, i.e. the actual user-data traffic. (There would be a subtle difference between v4 and v6 IP packet size)

c) Explore the details of the first HTTP request packet, what 'TCP option' have you observed? What TCP segment length is used by your computer and the web server? Explain what is meant by the Maximum Segment Size (MSS)?
TCP options
Maximum Segment Size: 1460 bytes, this indicates the maximum datagram that can be travelled without the fragmentation.
(Note: it is okay if TCP options are not found. But students should still mention Segment length and MSS so as to explain that both sender/receiver agrees on the same size. Also, if IPv6 packets are used, then MSS would be less, e.g. 1440 bytes)

d) Which TCP flag field is mostly set to 'true' (value 1)?
ACK flag (PSH flag can be acceptable)

Q3. ICMP packet observation [10 marks]
[Mark allocation: 2, 2, 3, 3]
Start Wireshark capturing and send ICMP messages through the command prompt.
Type: ping www.cs.auckland.ac.nz -4 -l 1000

a) Which field in the ICMP packet identifies the corresponding request/reply? How does ping know to match the corresponding packets?
Type field (1 byte) - Request code: 8, Reply code: 0
Sequence number (2 bytes) – identifies corresponding match

a) What is the size of ICMP packet (ICMP header and payload)? What value is the first byte of the payload?
1008 bytes: 8 bytes (ICMP header) + 1000 bytes (payload)
First byte: Vista, 0x61 ('a'), 0x41 ('A'), etc. Linux: increasing value, e.g. 0x62, 0x63, etc.

b) What is the length limit (-l option) before the packet is fragmented? Try high/low value until you find it. How did Wireshark find whether the packet was fragmented? Explain in conjunction with Maximum Transmission Unit (MTU) and header fields.
-l 1472
This length makes total 1518 bytes for each frame, which is the maximum frame length for the Ethernet. We know this because 'More Fragment Bit' is set in the IP header field for each ICMP request packet. Note, exact length can be different depending on your configuration.

c) Try sending fragmented ICMP packets to different servers (such as www.auckland.ac.nz, www.google.com). Describe and explain your observation.
It appears that many internal networks support large packet length. Sending the fragmented ICMP packets to the external networks mostly fail to receive responses. This shows that nodes somewhere outside drops the packets with MF field on.

Q4. Packet trace file [20 marks]
[Mark allocation: 2, 3, 4, 5, 6]
Use a web browser to download a file '**BigFile3**' from the 314 webpage *Assignment* section. Make sure to capture only packets belonging to the file download (i.e. set a filter). You should right-click on one of the packet and use the 'Follow TCP stream' to ensure you are only observing the downloaded file. Save the captured packets to the trace file (e.g. name it '**BigFile3-down**').

a) Calculate the MB/s (use the Wireshark Summary menu)
Total bytes (frames): 10,842,329 byes
Total duration: 2.109 s
10,842,329 / 2.109 / 10^6 = 5.141 MB/s

Or using Avg. MBit/s
41.124(Mb/s) / 8 = 5.141 (MB/s)

b) Consider '**BigFile3**' to be the *user-data*. Observing from the frame level with Wireshark, each datagram contains four parts: three kinds of overhead (extra bytes) per datagram and a section of user-data. List and explain the overheads for each protocol layer (Ethernet, IP and TCP). How does their position in the datagram correspond to the numbered protocol layers mentioned in the course Introduction?
Ethernet frame header (18 bytes):
IP packet header (v4 - 20 bytes, v6 - 40 bytes):
TCP packet header (20 bytes):
About 58 bytes are added for each user-data segment
- Overheads for the physical layers (such as signaling) are not measured.
- OSI model's session, presentation, and application layers are merged as Application layer in TCP/IP model.

c) Calculate the efficiency *ratio* of your 'user-data' to data transmitted by the switch. In other words, approximately what percentage of the total transmission should be user-data? Use the following formula: efficiency-ratio = file-size / captured-frames
'**BigFile3**' size: 10,098,110 bytes
Captured-frames: 10,842,329 bytes
= 93.136%

d) Can we improve the efficiency-ratio if the user-data segment is changed to smaller, or larger? Explain with some examples.

We'd improve the efficiency-ratio if we could increase the user-data segment larger. With fewer segments to transmit, fewer overheads are associated. However, this large segment means that it could potentially reduce the overall efficiency of the available bandwidth if not correctly transmitted/received.

Use Wireshark to open the file '**BigFile3**'. Compare with the captured file '**BigFile3-down**'.

e) Both transactions used TCP, however they are quite different. Can you identify some of the differences? Use Statistics menu (e.g. Summary, Packet length, Port type, etc) to support your answer. What kind of traffic do you think was carried in '**BigFile3**'?

'**BigFile3-down**': File transfer over TCP

- Fewer packets (~10k)
- Short duration (~2 seconds)
- Large packet sizes (~1000 bytes)
- High-rate (~40Mb/s)

'**BigFile3**': VoIP traffic over TCP (Note: students are not required to get this right)

- Many packets (~61k)
- Long duration (15 minutes+)
- Small packet sizes (~150 bytes)
- Low-rate (~0.08 Mb/s)
