National University of Singapore
School of Computing

CS2105 **Tutorial 3** Answer paper

1. Launch your browser and open its network diagnostic tool (e.g. press F12 if you use Chrome on Windows, or Cmd + Opt + I for Mac). Then click the "Network" tab to observe network communication.
   Copy-and-paste the following URL in the address bar of your browser:
   http://tiny.cc/atupaz

   Enter your choice and press the "Submit" button.

   a) Look at the entry named "formResponse". What is the HTTP request method issued?

   POST

   b) Briefly explain when HTTP POST and GET methods are used.

   (From Wikipedia) The POST request method requests that a web server accepts and stores the data enclosed in the body of the request message. It is often used when uploading a file or submitting a completed web form. In contrast, the HTTP GET request method is designed to retrieve information from the server.

To tutors:

Please demo it in class.

2. **[KR, Chapter 2, P21]** Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e. not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

   You may use 'dig' program to query the local DNS server. For example,

   ```
   dig -t a www.abc.com
   ```

   If IP address of this Web page has been queried by another computer seconds ago, your local DNS server should keep this knowledge in local DNS cache and is able to answer your query quickly. Otherwise, the query time will be long.

3. **[Modified from KR, Chapter 2, P31]** You are given 4 programs: **TCPEchoServer.py**, **TCPEchoClient.py**, **UDPEchoServer.py** and **UDPEchoClient.py**.

   a) Suppose you run **TCPEchoClient** before you run **TCPEchoServer**. What happens? Why?

   When creating a local socket, client attempts to make a TCP connection to a non-existent server process. Exception will be thrown.

   b) Suppose you run **UDPEchoClient** before you run **UDPEchoServer**. What happens? Why?

   UDP client doesn't establish connection to server when creating local socket. Thus it works fine if you start client program first and then server program (but data sent to server are all lost).

To tutors:

Please show the demo by running the programs on the instructor's PC.

4. **[KR, Chapter 3, R7]** Suppose a process in Host C has a UDP socket with port number 6,789. Suppose both Host A and Host B each sends a UDP segment to Host C with destination port number 6,789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

   Yes, both segments will be directed to the same socket. Sock APIs are available for us to learn the IP address of the origin of a packet (e.g. in Java that is `getAddress()` method).

5. **[Modified from KR, Chapter 3, P4]**

   a) Suppose you have the following 2 bytes: `01011100` and `01100101`. What is the 1's complement of the sum of these 2 bytes?

   Sum: `11000001`, checksum: `00111110`

   b) Suppose you have the following 2 bytes: `11011010` and `01100101`. What is the 1's complement of the sum of these 2 bytes?

   Sum: `01000000`, checksum: `10111111`

   (Note: UDP and TCP use 16-bit words in computing their checksums. For simplicity you are asked to consider 8-bit checksums in this problem).

6. **[Modified from KR, Chapter 3, P5]** Suppose that UDP receiver computes the checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? You may use Q5 as an example to explain.

   If sender transmits the following two bytes: 01011100 and 01100101, and the two bits highlighted in red flip, then checksum remains unchanged and receiver will fail to detect this error.

7. **[KR, Chapter 3, R9]** In our `rdt` protocols, why did we need to introduce sequence numbers?

   Sequence numbers are required for a receiver to find out whether an arriving packet contains new data or is a retransmission.

National University of Singapore
School of Computing

CS2105 **Tutorial 4** Answer paper

To students:

Please be reminded that our **Mid-term assessment** is on **Mon, 7 Oct 2019 2-4pm** at **MPSH2 Section C**. The scope of this test is everything taught before recess week.

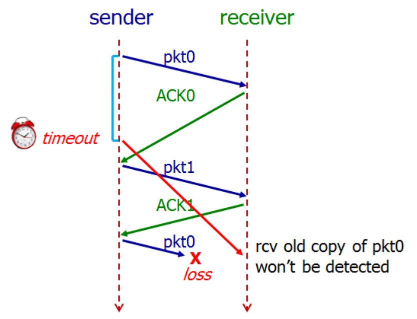
1. **[KR, Chapter 3, R6]** Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

   Note: this is exactly what you are supposed to do in Assignment 2. ☺

   One would have to implement reliability checking and recovery mechanisms (ACK, seq #, checksum, timeout, re-transmission, etc.) at application layer. For example, sender needs to include relevant header/trailer fields in every packet (as illustrated below).

IP Header	UDP Header	App Header fields + data + Trailer

2. Show an example that if the communication channel between the sender and receiver can reorder messages (i.e. two messages are received in different order they are sent), then protocol **`rdt3.0`** will not work correctly.



3. **[KR, Chapter 3, P29]** It is generally a reasonable assumption, when sender and receiver are connected by a single wire, that packets cannot be reordered within the channel between the sender and receiver. However, when the "channel' connecting the two is a network, packet reordering may occur. One manifestation of packet reordering is that old copies of a packet with a sequence or acknowledgement number of *x* can appear, even though neither sender's nor receiver's window contains *x*. With packet reordering, the channel can be thought of as essentially buffering packets and spontaneously emitting these packets at any point in the future. What is the approach taken in practice to guard against such duplicate packets?

   The approach taken in practice is to ensure that a sequence number is not reused until the sender is "sure" that any previously sent packets with the same sequence number are no longer in the network.

   Firstly, TCP use large sequence number field (32-bit) to lower the chance a sequence number is to be reused. Secondly, a packet cannot "live" in the network forever. For example, IP protocol specifies TTL in packet header to ensure that datagrams do not circulate infinitely in the network. This field is decreased by one each time the datagram arrives at a router along the end-to-end path. If TTL field reaches 0, router will discard this datagram. In practice, a maximum packet lifetime of approximately three minutes is assumed in the TCP extensions for high-speed networks.

4. **[Modified from KR, Chapter 3, P37]** Host A is sending data segments to Host B using a reliable transport protocol (either GBN or SR). Assume timeout values are sufficiently large such that all data segments and their corresponding ACKs can be received (if not lost in the channel) by Host B and the Host A respectively. Suppose Host A sends 5 data segments to Host B and the 2nd data segment is lost. Further suppose retransmission is always successful. In the end, all 5 data segments have been correctly received by Host B.

   How many segments has Host A sent in total and how many ACKs has Host B sent in total if either GBN or SR protocol is used? What are their sequence numbers? Answer this question for both protocols.

   GBN: Host A sends 9 segment. They are initially sent segments 1, 2, 3, 4, 5 and later resent segments 2, 3, 4 and 5. Host B sends 8 ACKs. They are 4 ACKs with seq # 1 and 4 ACKs with seq # 2, 3, 4 and 5.

   SR: Host A sends 6 segment. They are initially sent segments 1, 2, 3, 4, 5 and later resent segment 2. Host B sends 5 ACKs. They are 4 ACKs with seq # 1, 3, 4, 5 and 1 ACK with seq # 2 (for resent segment).

5. **[KR, Chapter 3, R15]** Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 65; the second has sequence number 92.

   a) How much data is in the first segment?

      92 – 65 = 27 bytes

   b) Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

      65. Note that TCP acknowledgment is cumulative and states the expected in-order sequence number.

6. **[KR, Chapter 3, P26]** Consider transferring an enormous file of *L* bytes from Host A to Host B. Assume an MSS of 512 bytes.

   1. What is the maximum value of *L* such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field is 32 bits.

      TCP sequence number doesn't increase by one with each TCP segment. Rather, it increases by the number of bytes of data sent. Therefore the size of the MSS is irrelevant here -- the maximum size file that can be sent from A to B without exhausting TCP sequence number is simply 2^{32} bytes.

   2. For the *L* you obtain in (a), find how long it takes to transmit this file. Assume that a total of 64 bytes of transport, network, and data-link header are added to each packet before the resulting packet is sent out over a 155 Mbps link. Ignore flow control, congestion control and assume Host A can pump out all segments back to back and continuously.

      Number of packets = L / MSS = 2^{32} / 512 = 8,388,608
      64 bytes of headers will be added to each packet. Therefore,
      Total bytes sent = 2^{32} + 8388608 * 64 = 4,831,838,208 bytes
      Transmission delay = 4831838208 * 8 / (155 * 10^6) ≈ 249 seconds

7. Wireshark: TCP

   Do the following:

   1. Start up your web browser. Go the http://gaia.cs.umass.edu/wiresharklabs/alice.txt and retrieve an ASCII copy of Alice in Wonderland. Store this file somewhere on your computer.

2. Next, go to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.
3. Use the Browse button to enter the full path name on your computer containing Alice in Wonderland. Don't yet press the "Upload alice.txt file" button.
4. Startup Wireshark and begin packet capture.
5. Returning to your browser, press the "Upload alice.txt file" button. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.
6. Stop Wireshark packet capture.

Answer the following questions:

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?
   Ans: select an HTTP message sent from your computer to gaia.cs.umass.edu and explore the details (src-ip & src-port) of the TCP packet used to carry this HTTP message.
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?
   Ans: ip: select an HTTP message explore the details of the TCP packet used to carry this HTTP message. port: 80.

8. Do you have any question on Assignment 1 to clear?

National University of Singapore
School of Computing

CS2105 **Tutorial 5** Answer paper

1. **[KR, Chapter 4, R13]** What is the 32-bit binary equivalent of the IP address 202.3.14.25?

   11001010 00000011 00001110 00011001

2. Write down the first and last IP addresses of the subnet associated with the following IP address:

   a) 244.233.234.12/4

   244 = 11110100
   First IP address in this subnet = 11110000 00000000 00000000 00000000 = 240.0.0.0

   Last IP address in this subnet = 11111111 11111111 11111111 11111111 = 255.255.255.255

   b) 10.45.123.34/19

   123 = 01111011
   First IP address in this subnet = 00001010 00101101 01100000 00000000 = 10.45.96.0

   Last IP address in this subnet = 00001010 00101101 01111111 11111111 = 10.45.127.255

3. Combine the following three blocks of IP addresses into a single block:

   a) 16.27.24.0/26
   b) 16.27.24.64/26
   c) 16.27.24.128/25

   16.27.24.0/24

4. **[Modified from KR, Chapter 4, P16]**

   a) Consider a subnet with network prefix 192.168.56.128/26. Give an example IP address (of form xxx.xxx.xxx.xxx) that belongs to this network.

   Any IP address in the range 192.168.56.128 to 192.168.56.191

b) Suppose an ISP owns the block of addresses of the form 192.168.56.128/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the network prefixes (of form a.b.c.d/x) for the four subnets?

Network Prefix	Binary Expression
192.168.56.128/28	**11000000 10101000 00111000 10000000**
192.168.56.144/28	**11000000 10101000 00111000 10010000**
192.168.56.160/28	**11000000 10101000 00111000 10100000**
192.168.56.176/28	**11000000 10101000 00111000 10110000**

5. **[KR, Chapter 4, P7]** Consider a datagram network using 8-bit addresses. Suppose a router has the following forwarding table:

Prefix Match	Interface
11	0
101	1
100	2
otherwise	3

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

Prefix Match	Interface	IP Range	No. of IP
11	0	**1100 0000 – 1111 1111**	**64**
101	1	**1010 0000 – 1011 1111**	**32**
100	2	**1000 0000 – 1001 1111**	**32**
otherwise	3	**0000 0000 – 0111 1111**	**128**

6. What is private IP address? Does LumiNUS use private or public IP? When your laptop is connected to NUS network, does it receive a private or public IP?

(Part of the following answer is modified from RCF1918: https://tools.ietf.org/html/rfc1918)

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of IP address space for private networks:

10.0.0.0 - 10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

An enterprise that decides to use private IP addresses can do so without any coordination with IANA or an Internet registry. The address space can thus be used by many enterprises.

However, private IP addresses cannot have IP connectivity to any host outside of the enterprise. NAT or application layer gateways are needed to map private to public address and vice versa when traffic goes in and out private network.

Private IP is initially designed for experimentation purpose, but now used as a way to alleviate IPv4 address exhaustion. Its use is very common today.

LumiNUS use public IP address (137.132.10.10). Students may use `ping` command to check it. Laptops of students are assigned private IP addresses (e.g. 172.26.184.76).