

## **Cover Page**

Lab 5 Preparation  
Name: Wei Lin  
Student#: 999595193

### **1. Explain the role of port numbers in TCP and UDP.**

A port number is a 16-bit integer in the TCP and UDP header appended to the data from the higher-layer applications. It specifies which higher-layer application is sending or receiving TCP/UDP packets and each higher-layer application has its own port number. The port number allows different applications on the same host to share network resources simultaneously.

### **2. Provide the syntax of the tcp command for both the sender and receiver, which executes the following scenario:**

**A TCP server has IP address 10.0.2.6 and a TCP client has IP address 10.0.2.7. The TCP server is waiting on port number 2222 for a connection request. The client connects to the server and transmits 2000 bytes to the server, which are sent as 4 write operations of 500 bytes each.**

```
#Server: tcp -r -p 2222 -l 500 -n 4
#ClientL tcp -t -p 2222 -l 500 -n 4 -D 10.0.2.6
```

### **3. Answer the following questions on Path MTU Discovery:**

#### **a. How does TCP decide the maximum size of a TCP segment?**

The maximum segment size (MSS) is a parameter in the options field of TCP header. It indicates the maximum number of bytes can be sent in a TCP segment. The TCP obtains MSS by asking for Maximum Datagram Data Size (MDDS) from the IP. Then, The IP will get the MTU information of direct connected network from its network interfaces. Finally, the TCP obtains the MSS information by subtracting IP header and TCP header size from the MTU size.

#### **b. How does UDP decide the maximum size of a UDP datagram?**

The maximum size of UDP datagram is imposed by the IPv4 protocol, which is 65,507 bytes (65,535 – 8 byte UDP header – 20 byte IP header).

#### **c. What is the ICMP error generated by a router when it needs to fragment a datagram with the DF bit set? Is the MTU of the interface that caused the fragmentation also returned?**

The ICMP error generated is type 3 code 4, "Fragmentation Needed and Don't Fragment was Set".

Yes, the MTU of the interface that caused the fragmentation will be included in the lower 16 bits of "unused" field of ICMP header.

#### **d. Explain why a TCP connection over an Ethernet segment never runs into problems with fragmentation.**

The maximum TCP segment size is 65535 bytes, which is far larger than the size of any packet you have ever seen in the real life, because the lower layers has limitation of packet size. For example, Ethernet in the data link layer allow MTU size of only 1500 bytes, which means TCP MSS can have only 1460 bytes.

**4. Assume a TCP sender receives an acknowledgement (ACK), that is, a TCP segment with the ACK flag set, where the acknowledgement number is set to 34567 and the window size is set to 2048. Which sequence numbers can the sender transmit?**

Because the acknowledgement number is 34567, which means packets up to sequence number  $34566 = 34567 - 1$  have been received. Therefore, the sequence number can be transmitted starting from 34567. Window size specifies the number of bytes that the receiver is willing to receive. Therefore, sequence numbers from 34567 to  $36615 = (34567 + 2048)$  are allowed to transmit.

**5. Describe the following heuristics used in TCP and explain why they are used:**

**a. Nagle's algorithm**

Applications such as Telnet could repeatedly transmit data in small segments (1 byte in size). This results in a 41 byte packet containing only 1 byte data and 40 bytes header information (TCP header + IP header), which is a huge overhead. Nagle's algorithm works by only allowing one 1-byte segment outstanding that has not been acknowledged in each TCP connection. Specifically, as long as a sender does not receive the acknowledgment of first 1-byte segment, the sender will buffer all subsequent bytes until the acknowledgment is received. Then, the sender will send all buffered bytes in a single segment. Nagle's algorithm improves the efficiency of TCP/IP networks by reducing the number of small segments that need to be sent over the network.

**b. Karn's Algorithm**

The round trip time (RTT) is estimated as the difference between the time that a segment was sent and the time that its acknowledgment was received. However, if packets are retransmitted, sender cannot tell whether the acknowledgment belongs to the original or the retransmission. Karn's algorithm addresses the problem by using exponential backoff algorithm:

The sender side will not update the SRTT (smoothed RTT) on any segments that have been retransmitted. Each time when TCP retransmits packets, it sets new  $RTO = \min(2 * \text{old } RTO, 64)$ , which means time between each retransmission is doubled each time and cannot exceed 64 seconds.

**6. Answer the following questions about TCP acknowledgements:**

**a. What is a delayed acknowledgement?**

The delayed acknowledgement means TCP delays transmission of acknowledgements for up to 200ms if the sender of acknowledgement packets doesn't want to send any extra data to the receiver. The goal is to avoid sending acknowledgement packets that do not carry data. The hope is that, within the delay, the sender will have data ready to be sent to the receiver, then the acknowledgement packet can be piggybacked with a data segment.

**b. What is a piggybacked acknowledgement?**

In two way communication, whenever a data frame is received, the receiver waits and does not send the acknowledgement back to the sender immediately. The receiver waits until its network layer passes in the next data packet. The delayed acknowledgement is then attached to this outgoing data frame. This technique

of temporarily delaying the acknowledgement so that it can be hooked with next outgoing data frame is known as piggybacking. The major advantage of piggybacking is better use of available channel bandwidth.

### **7. Describe how the retransmission timeout (RTO) value is determined in TCP.**

The host must set the initial value:

$$SRTT = RTT$$

$$RTTVAR = RTT/2$$

$$RTO = SRTT + 4 * RTTVAR$$

When a new round trip time  $RTT'$  is measured, the corresponding  $SRTT'$ ,  $RTTVAR'$  and  $RTO'$ :

$$SRTT' = \alpha * RTT' + (1 - \alpha) * SRTT$$

$$RTTVAR' = \beta * |SRTT - RTT| + (1 - \beta) * RTTVAR$$

$$RTO' = SRTT' + 4 * RTTVAR'$$

The above should be computed using  $\alpha=1/8$  and  $\beta=1/4$ . Whenever RTO is computed, if it is less than 1 second, then the RTO should be rounded up to 1 second.

### **8. Answer the following questions on TCP flow control and congestion control:**

#### **a. Describe the sliding window flow control mechanism used in TCP.**

Sliding window protocol allows an unlimited number of packets to be exchanged using fixed-size sequence numbers by setting a limit on the number of packets that can be transmitted or received at any given time. The receiver will send acknowledgment packets containing the current receiver buffer size, which informs sender the maximum number of bytes can be transmitted. The TCP header has a 16-bit window size field, which means the largest window size is  $2^{16} = 64$  kilobytes. For every ACK packet received, the window of sender slides by one sequence number and if the window threshold is reached, the sender sends one packet after one ACK packet received. On the receiver side, there is also a window moves one sequence number for every packet received. Sliding window protocols are used where reliable in-order delivery of packets is required.

#### **b. Describe the concepts of slow start and congestion avoidance in TCP.**

Slow-start is part of the congestion control strategy used by TCP. It begins slow start stage with a small congestion window size and a slow start threshold value. The window size increases by one with each ACK received until a loss event is detected or the slow start threshold is reached. If a loss event happens, the slow start threshold will be reduced to half of current window size and then, the window size will be reset to 1. If the slow start threshold is reached, the sender will enter congestion avoidance mode. In the congestion avoidance mode, the window size is increased by 1 for each ACK received. Congestion avoidance is to predict when congestion is about to happen and then to reduce the rate to prevent data loss.

#### **c. Explain the concept of fast retransmit and fast recovery in TCP.**

Fast retransmit is an enhancement to TCP which reduces the time a sender waits before retransmitting a lost segment. If a sender receives 3 or more number of duplicated acknowledgements with same acknowledge number, the sender believes that the segment with the next higher sequence number was dropped. Then, the sender will perform retransmission of lost packet without waiting for a timeout to happen and enter slow start stage.

Fast recovery avoids slow start stage after a fast retransmit. After detecting 3 duplicated acknowledgements, the sender will retransmit the loss packet but not enter the slow start stage. Instead, the sender will set the slow start threshold to half of current window size and then reduce the window size to the slow start threshold plus 3 ( i.e,  $ssthresh = cwnd/2$  and  $cwnd = ssthresh + 3$ ). The window size will be increased by 1 for each following duplicated acknowledgement. After receiving acknowledgement of new data, the sender will set the window size equal to slow start threshold and enter congestion avoidance mode.