**T08: Transport Layer II**

**Q1:** Are the objectives of flow control and congestion control the same? Why or why not?

Not the same. Flow control is co-ordinating of flow between sender and receiver. Objective is to prevent receiver buffer overflow by slowing down the sender. Congestion is a global issue, may not be able to identify what is causing the congestion/where. Objective is to protect network from being overloaded which is done by slowing down the sender whenever a datagram is lost or delayed excessively in the network. Though in both cases the sender is slowed down, the objective is different.

**Q2:** What is the advantage in using hop-by-hop choke packet over typical choke packet method as a solution to network congestion?

In typical choke packet method you need to go all the way back to the sender. The hop by hop method starts to take action when detected by the intermediate nodes. Hop by hop takes affect much quicker. Choke packets only affect the source where hop by hop choke packet will affect each hop it passes through

**Q3:** Describe two (2) major differences between the bit warning method and the RED method.

Bit warning: When packets arrive at a router that is congested and the buffer is full. The router will attach a tag (1 bit in the header) to inform that it is becoming congested (piggy back a warning bit in the header) and is carried all the way to the destination so when the destination sends an ACK it will copy the warning bit (flag) and piggy backed in the packet so when the sender receives the ACK packet they will see that there is congestion somewhere. Sender will monitor how many ACK its accepts with warning bits and adjusts its flow accordingly.

RED method (Random Early Discard): Does not wait for the buffer is full before it acts. When the packet queue time reaches the lower threshold (packet flow is ok) it wont discard any packets until the packet queue reaches a higher threshold (serious congestion before the buffer is completely full), an alarm will be raised and the router will randomly discard some of the packets to warn the sender (as the sender will not receive ACK packets)

Bit warning will only send a warning when the buffer is full, RED method will send a warning before the buffer is full. Warning bits and choke packets are directly sent to the sender to tell the sender there is congestion. RED there is no explicit communication to the sender, just start dropping packets. Wine vs Milk policy: Application layer chooses which packets to keep (old or new) FTP discard new packets (if old packets are dropped everything becomes corrupted). Real time audio communication discard old packets (new packets more important for real time communication).

**Q4:** Why was it difficult to detect congestion in old days?

More errors present in the old days due to unreliable network. Packets were frequently lost due to poor transmission quality so its hard to tell if packet loss is due to unreliable network or genuine congestion.

**Q5:** Consider the effect of using slow start on a line with a 10-msec round-trip time and no congestion. The receive window is 24KB and the maximum segment size is 2KB. How long does it take before the first full window can be sent?

Min(credit, cwnd). [Credit = 24KB]

After the 4th transmission (1: 2KB 2: 4KB 4: 8KB 8: 16KB 5: 24KB) 5th is cancelled because the cap is 24KB so the answer is after 4 round trips or 40ms

**Q6:** Why is TCP called a byte-stream protocol? How does UDP differ from TCP in this regard? Which layer is responsible to segment the data if UDP is used in Transport layer? (Is it Transport layer itself or Application layer)? What protocol would you use to multicast or broadcast a message?

TCP is not message oriented as its byte oriented (segmentation done in bytes). UDP is message oriented not byte oriented (application layer does segmentation if UDP is the transport layer protocol). UDP is a very light-weight protocol with minimal overhead. TCP is connection oriented and reliable (everything is acknowledged). UDP unreliable by design (no acknowledgments are sent, if packets get lost with error then there is no warning) Example: UDP used for video audio communication, if there are errors you hear or voice or video cutting out not a big deal as you don’t need the network to resend it, just need the newest packets. TCP only support unicasting. For broadcasting UDP must be used.

**Q7:** DNS uses UDP instead of TCP. What is the main difference between UDP and TCP? If a DNS packet is lost, will there be automatic recovery? Will that cause a problem? Why or why not?

CNCO2000 – Tutorial 08



TCP: Stream-Oriented

* Reliable: transmission is constantly checked for errors

UDP: Message-Oriented

* Connectionless: no 3 or 4 way handshakes required

DNS uses UDP. If there is an error there will be no auto-recovery. Not important because there is no harm, just send another query to the DNS server until the response is correct. When a process makes a DNS request a timer is started if the timer expires it makes the request again.

1 | P a g e