
Name: Itish Agarwal
Roll: 18CS30021
Computer Networks Class Test 3

Q1.

In general, TCP's congestion control can be described as a combination/hybrid of GBN and Selective Repeat. This is because :

-> TCP is similar to GBN because both protocols have a limit on the number of unACK'd packets that the sender can send into the network. However, TCP is different from GBN because GBN requires the re-transmission of every unACK'd packet when packets are lost, but TCP only retransmits the oldest unACK'd one.

-> TCP is similar to selective repeat because, when packets are lost due to congestion, the protocols do not require the sender to retransmit EVERY unACK'd packet sent by the sender. The sender just retransmits the oldest unACK'd packet.

Q2.

In a TCP's three-way handshake the server allocates and initializes connection variables and buffers in response to a received SYN. The server then sends a SYNACK in response, and awaits an ACK segment from the client. If the client does not send an ACK to complete the third step of this 3-way handshake, eventually (often after a minute or more) the server will terminate the half open connection and reclaim the allocated resources.

This TCP connection management protocol sets the stage for a classic Denial of Service (DoS) attack known as the **SYN flood attack**. In this attack, the attacker(s) sends a large number of TCP SYN segments, without completing the third handshake step. With this deluge of SYN segments, the server's connection resources become exhausted as they are allocated (but never used) for half-open connections; legitimate clients are then denied service.

A solution to this is to use cryptographic functions to generate sequence numbers ("SYN cookies") for SYN segments. The receiver after sending acknowledgement to sender temporarily closes the connection with the sender. When the sender sends a data segment, the sequence number is verified by using the same cryptographic function and connection is resumed.

Q3.

No, it is not safe to use this 16-bit sequence number field for a sliding window based flow control algorithm. It is not safe because :

We have,

BDP = (Bandwidth) * (Delay ProducT) = 100 Mb

Now, since each segment is of 1 byte, then
Total number of segments = BDP / 1 byte = 10000000 (approximately.)

But our sequence number range is only 2^16 = 65536. Clearly, (or logically due to the pigeonhole principle) it is unsafe to use 16 bit sequence numbers since we will run out of unique sequence numbers very soon.

Q4.
Silly Window Syndrome is a problem that arises due to poor implementation of TCP. It degrades the TCP performance and makes the data transmission extremely inefficient. Here data is passed to the sending TCP in large blocks, but an interactive application on the receiver side reads the data only 1 byte at a time. The two major causes of this syndrome are :

A. Sender window transmitting one byte of data repeatedly
B. Receiver window accepting one byte of data repeatedly

Clarke's solution suggests that the receiver should not send a window update for 1 byte. Receiver should wait until it has a decent amount of space available. Receiver should then advertise that window to the sender.

Q5. My roll number is 18CS30021.
So,
a = ((1 + 8 + 3 + 19 + 3 + 0 + 0 + 2 + 1) mod 5 / 10) + 0.5 = 0.2 + 0.5 = 0.7
b = a / 2 = 0.35
SRTT0 = 0ms
RTTVAR0 = 0ms
RTT1 = 1100ms
RTT2 = 1500ms
SRTT1 = aSRTT0 + (1-a)RTT1 = 330
RTTVAR1 = bRTTVAR0 + (1-b) |SRTT1-RTT1| = 500.5
RTO1 = max(SRTT + 48RTTVAR, 1 sec) = 1sec

Q6. pto

Q6.

(a) False

We can include the SACK option in the Option parameter of the TCP header. This allows the sender to send only specific segments to the receiver.

(b) False

Because the Maximum Transfer Unit (MTU) is generally set to 1500 bytes and not Maximum Segment Size (MSS) so it is false.

(c)False

During the slow start phase, the CWnd parameter in TCP Tahoe increases at an exponential rate.

(d) True

The receiver waits for data on which to piggyback an acknowledgement, and the sender waits on the acknowledgement to send more data, thus causing a temporary deadlock.

(e) False

The URG flag has higher priority than PSH flag because it is used to send the data on a priority basis to the application layer. When URG is set to 1, the “urgent” data is sent directly to the application layer while bypassing the buffer.