19L-1196 Assignment #3 To: Sir Salman Mubarak,
Computer Networks

(P8):
Only timeout has been added in Sender Side Side protocal 3.0 which was not in protpcal 2.2. the addition of timeout adds the possibility of duplicates packet into Sender receiver side data stream. On the other hand, protocal 2.2 can also handle duplicate packets. Hence, 2.2 can also work as 3.0.

(P22/a)
window size N=3
let packets received : k-1
if has acknowledged all preceding packets
then sender window
[k, k+N-1]

let next acknowledgement be received at Sender, then Sender windows Contains

[k-N, k-1]
Hence, Sender window has range.

[k-N, k]

P22/b If receiver has received $k-1$ packet then then it is waiting for packet k.
Also, Acknowledgement for $k-1$ packet has been made.
If none of those N Acknowledgement has successfully been made by the sender, then acknowledgement with message $[k-N, k-1]$ may still be propagating.
Once, the receiver has sent an acknowledgement for $k-N-1$, it will never send acknowledgement for $k-N-1$.
Hence range $[k-N-1, k-1]$

P23 : We need to determine how large a sequence numbers can be covered at any given time by the receiver and sender windows, to avoid having the leading edge of the receiver's window wrap around the sequence number space and overlap with trailing edge.
Sequence number space must be large enough to fit the entire receiver window and entire sender window.

Hence, Sequence number space $\geq 2W$

sequence number, must be at least twice as large as the window size.

P25/a With UDP, application has more control of what the data is sent in a segment. with Tcp, the application writes data to the connection Send buffer and Tcp will hold a byte without puting a single message in the tcp segment. UDP, on the other hand, encapsulates in a segment whatever the application gives.

P25/b UDP does not suffer delays and congestion control, while Tcp suffers flow control and congestion control. Hence there may be a significant delay between application writes data to its Send buffer until the data is given to the network layer.

P26/a There are possible sequence numbers: 2^{32}

Since, sequence number depends on number of bytes of data sent, Hence size of max file is 2^{32} bytes. nearly 4 GB.

P26/b Sigment NO: $2^{32}/536$
$= 8012999.$

Now after adding 66 bytes of header to each segment
total: 528,857,934 bytes

total bytes transmitted:
$2^{32} + 528,857,934 = 4.8 \times 10^9$ bytes.
it would take
$\simeq 250$ sec to transmit over the bandwidth of 155 Mb.

P27/a

Source port NO: 302
Destination port NO: 80
In the second segment from Host A to B, the sequence number is 207.

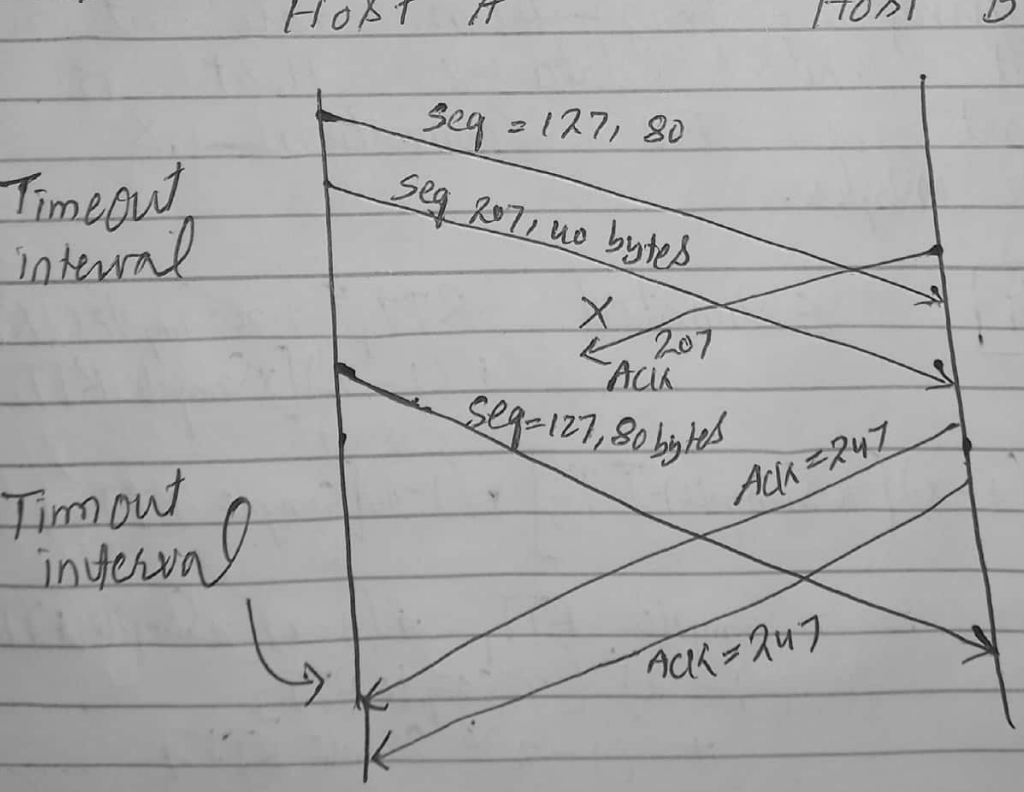
p27/b If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the Ack number is 207.

Source port NO: 80

Destination port No: 302

p27/c If the second segment arrives before the first segment, the Ack number is 127, indicating that it is still waiting for bytes 127 and onwards.

p27/d



P28 Link capacity 100 Mbps
Host A's sending rate can be almost 100 Mbps.
Still host A sends data into the receive buffer faster than Host B can read out data from buffer.
The receive buffer fills up at a rate of roughly 40 Mbps. When the buffer is full, Host B signals to host A to stop sending data by ~~ben~~ setting RecWindow to zero.
Host A Then stops sending until it receives a TCP segment with RecWindow > 0.
On average, long-term rate at which Host A sends data to Host B as part of this connection is 60 Mbps.

P32/a: Estimated $RTT_4 = x\,Sample\,RTT_1 + (1-x)[x\,Sample\,RTT_2 + (1-x)[x\,Sample\,RTT_3] + (1-x)[Sample\,RTT_4]$

$= x\,Sample\,RTT_1 + (1-x)(x\,Sample\,RTT_2 + (1-x)^2(x\,sample\,RTT_3 + (1-x)^3\,Sample\,RTT_4$

b P32/b Estimated RTT_n

$$= x \sum_{j=1}^{n-1} (1-x)^{j-1} \text{ sample } RTT_j + (1-x)^{n-1} \text{ Sample RTT}$$

P32/c estimated RTT_∞

$$= \frac{x}{1-x} \sum_{j=1}^{\infty} (1-x)^j \text{ sample } RTT_j$$

$$= \frac{1}{9} \sum_{j=1}^{\infty} 9^j \text{ Sample } RTT_j .$$

The weight given to past samples decays exponentially.

P46/a W = max window size

$W * MSS / RTT = 10 Mbps$

As packets will be dropped if the maximum sending rate exceeds link capacity.

$\frac{W * 1500 * 8}{0.15} = 10 \times 10^6$ Then W is about 125 segments.

P46/b windows varies $\frac{W}{2}$ TO W.

Avg window size: $\frac{\frac{W}{2} + W}{2} = 0.75W$

$\approx$ 94 Segments.

Then avg throughput is :

$$\frac{94 \times 1500 \times 8}{0.15} = 7.52 \text{ Mbps}$$

P46/c time : $\frac{94}{2} \times 0.15 = 7.05$ sec

as ~~RTT~~ RTT is $\frac{W}{2}$. window size increases by one in each RTT.

What is DTLS Datagram Transport Layer Security is a communication protocal for security for data-gram based application by letting them make communication to avoid eavesdropping and data integration. It is based on TLS (Transport layer security) protocal.
It does not undergo suffering of associated delays but since it uses UDP or SCTP instead of TCP it takes care of rearrangement of packets, lost datagrams usually caused by greater size of datagram network packet.