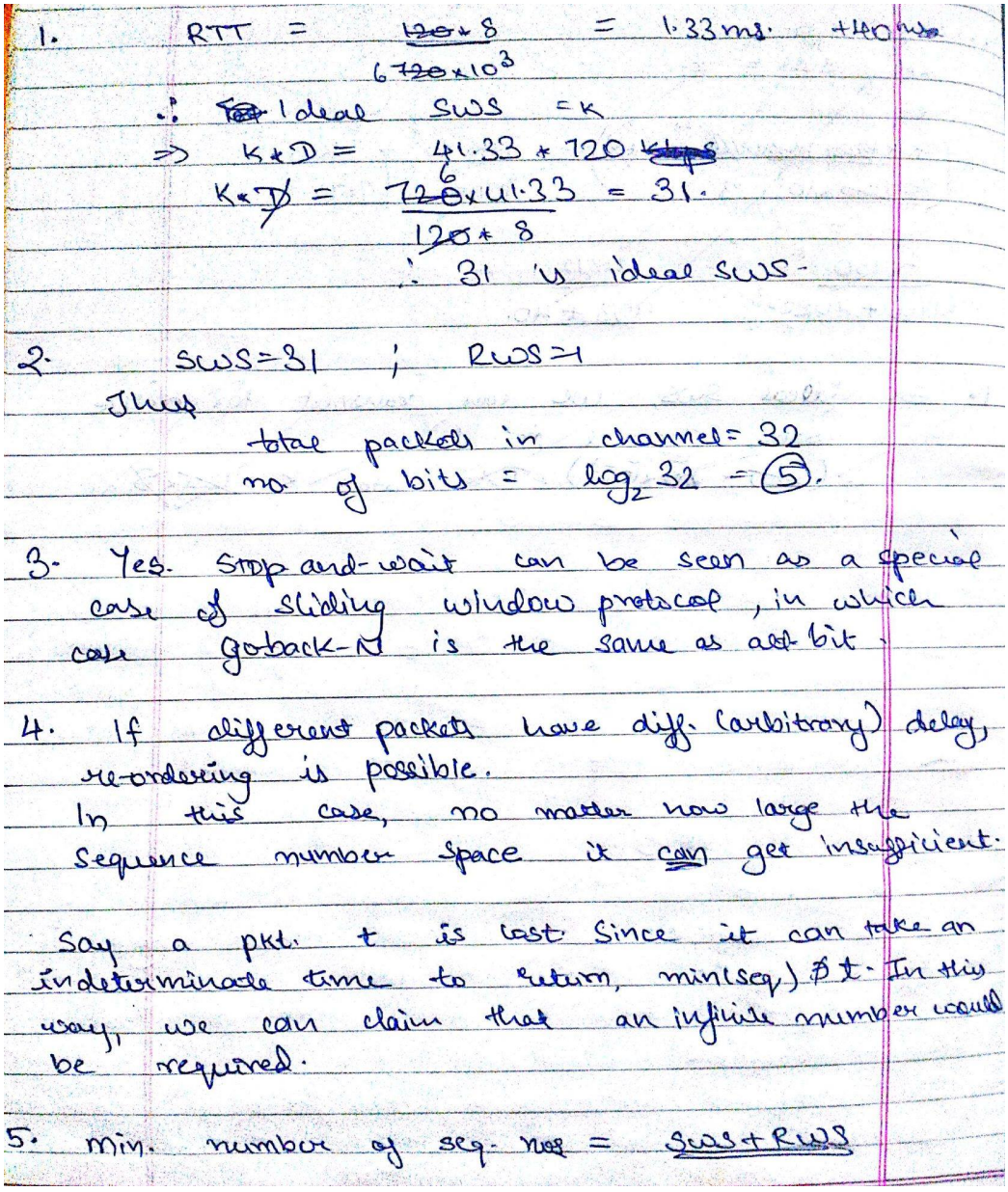
CS 224(M): Tutorial 5

Sliding Window Protocol

Name: Dhruv Ilesh Shah **Roll No.**: 150070016



1. RTT = $\frac{120 * 8}{6720 \times 10^3}$ = 1.33 ms. + 40 ms

∴ Ideal SWS = K

⇒ K * D = $\frac{41.33}{6} * 720$

K * D = $\frac{720 \times 41.33}{120 * 8}$ = 31.

∴ 31 is ideal SWS.

2. SWS = 31 ; RWS = 1

Thus

total packets in channel = 32

no of bits = $\log_2 32$ = (5).

3. Yes. Stop and wait can be seen as a special case of sliding window protocol, in which case goback-N is the same as alt bit.

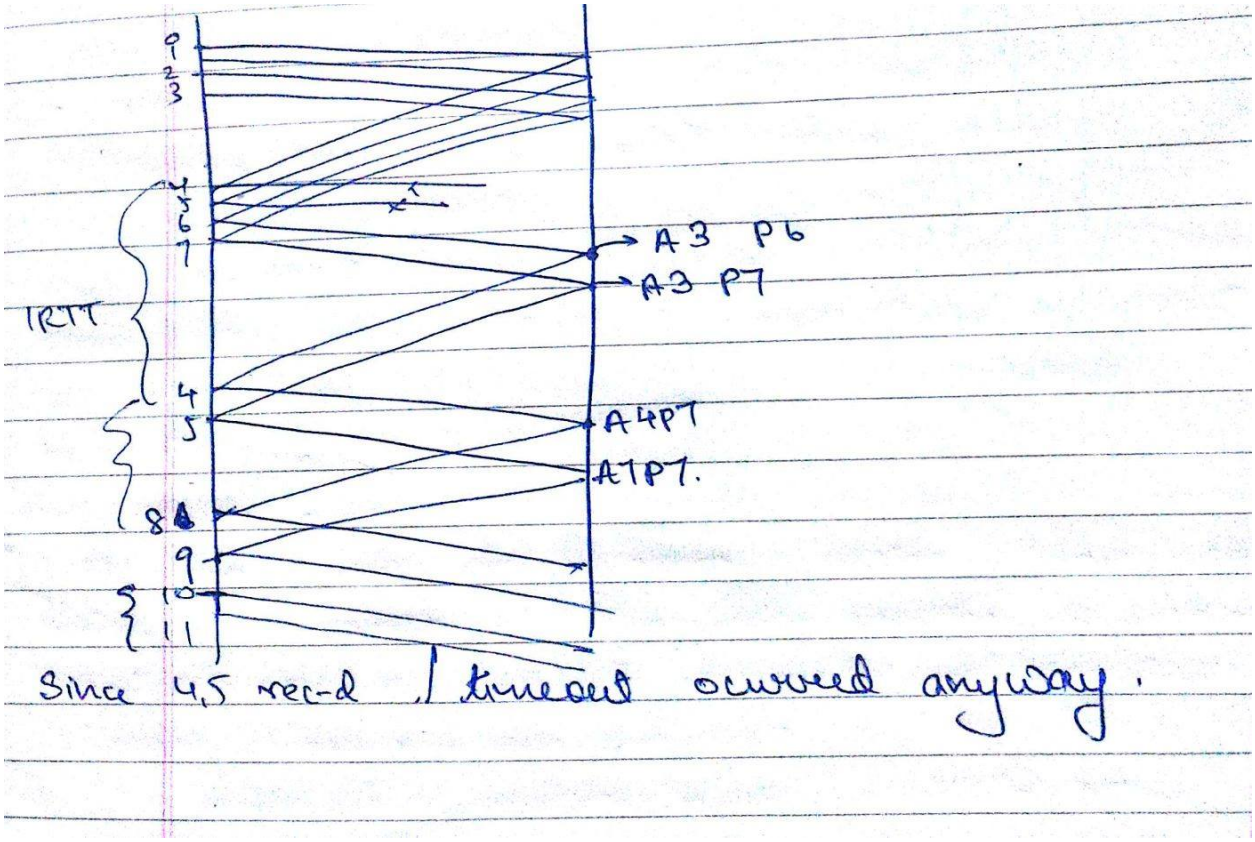
4. If different packets have diff. (arbitrary) delay, re-ordering is possible.

In this case, no matter how large the sequence number space it can get insufficient.

Say a pkt. t is lost. Since it can take an indeterminate time to return, min(seq.) ∄ t. In this way, we can claim that an infinite number would be required.

5. Min. number of seq. nos = SWS + RWS

6. Draw a timeline of the sliding window protocol that employs selective acknowledgment with SWS=RWS=4. Assume frames 4,5 are lost. Use a time-out of 2*RTT. Show the timeline till frame 11 is sent.



---

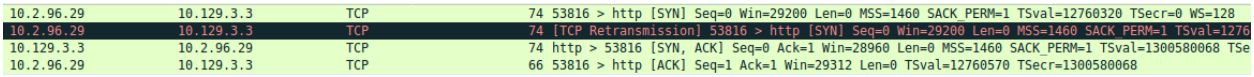
7. Observe the TCP connection establishment prior to a HTTP request-response, using wireshark. Observe the SYN and ACK flags. Then observe the use of random initial sequence numbers. Observe the cumulative nature of the ACKs.

```
4.453053000  10.2.96.29      10.200.17.31    TCP    66 35104 > http [ACK] Seq=611 Ack=985 Win=32128 Len=0 TSval=12555071 TSecr=2972266765
4.453078000  10.200.17.31    10.2.96.29      TCP    66 http > 35104 [FIN, ACK] Seq=985 Ack=611 Win=30208 Len=0 TSval=2972266765 TSecr=12555056
4.455028000  10.2.96.29      10.200.17.31    TCP    66 35104 > http [FIN, ACK] Seq=611 Ack=986 Win=32128 Len=0 TSval=12555072 TSecr=2972266765
4.455326000  10.200.17.31    10.2.96.29      TCP    66 http > 35104 [ACK] Seq=986 Ack=612 Win=30208 Len=0 TSval=2972266767 TSecr=12555072
```

Webpage used: home.iitb.ac.in/~dhruv.shah/cs224m.html

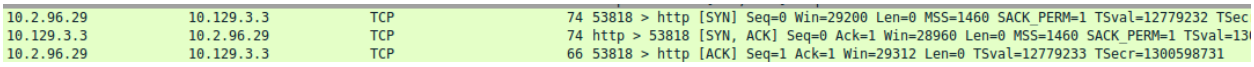
8. Using telnet to a web server (say www.cse.iitb.ac.in), and wireshark, show that the source ports are different for two different TCP connections, between the same two machines, and to the same destination port.

Connection 1: *(using telnet over bash)*



10.2.96.29	10.129.3.3	TCP	74 53816 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=12760320 TSecr=0 WS=128
10.2.96.29	10.129.3.3	TCP	74 [TCP Retransmission] 53816 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1276
10.129.3.3	10.2.96.29	TCP	74 http > 53816 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1300580068 TSe
10.2.96.29	10.129.3.3	TCP	66 53816 > http [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=12760570 TSecr=1300580068

Connection 2: *(using telnet over bash)*



10.2.96.29	10.129.3.3	TCP	74 53818 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=12779232 TSec
10.129.3.3	10.2.96.29	TCP	74 http > 53818 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=13
10.2.96.29	10.129.3.3	TCP	66 53818 > http [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=12779233 TSecr=1300598731