1. Flow loop:

|  |  |
| --- | --- |
| Time | Action |
| [0, x] | Sender sends data packet with time x = (PS / Bandwidth) |
| [PD, PD+x] | Receiver receives data packet |
| [PD+x, PD+x+y] | Receiver sends ACK packet (its seq#) with time y |
| [2PD+x, 2PD+x+y] | Sender receives ACK packet |

PD = Propagation delay

PS = Data packet size (bits)

B = Bandwidth

We need only 6 bits for seq#. Including headers and CRC, an ACK packet size \ transmission time is still negligible compared to a data packet size \ transmission time. Therefore, y is negligible.

It took us 2PD+x+y seconds to send the first packet and receive its ACK using GBN.

We would like to send N packets in 2PD+x+y seconds

Setting y=0, we get:

x = 800\*8 bits / 10 Gbps = 0.64 msecs.

PD = 30 msecs.

bits(N packets) = N \* PS = 40000 bytes = 320000 bits

Without using the protocol, we will need Nx seconds to send N packets.

1. For a large window, the probability for some packet error (data or ACK) in the pipeline is increasing, resulting in retransmission of many packets that were already sent (and that will be received later) properly.
2. Assume N=1. So we have a window of size 1 and only 1 sequence numbers, meaning every numbered ACK is no different than any other ACK. Assume we have a long timer in comparison to the propagation delay. Scenario: We sent packet1 but it was lost. The window has moved, we sent packet2 and got an ACK, which we assumed belonged to packet 1 due to the long timer. After a very long time packet 2 timeout will expire. We then resend packet 2 and receive an ACK. So we believe we successfully sent packets 1 and 2, but it was actually two instances of packet 2 that were received.