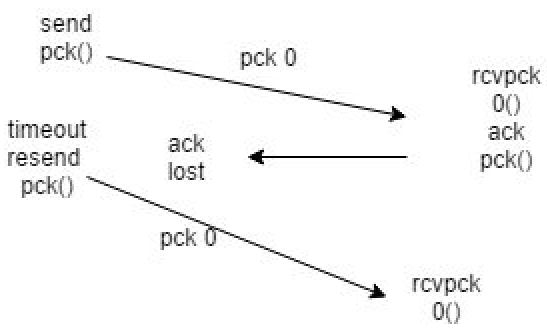
John Donnelly
Jed126
Networks

1.

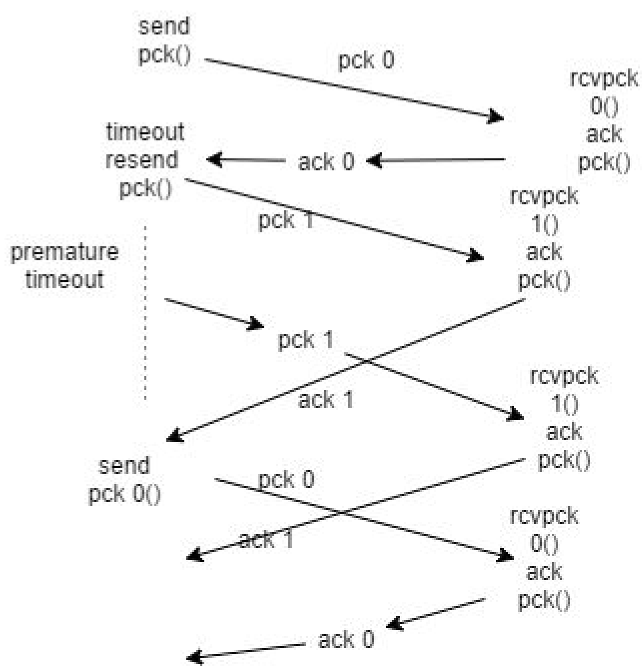
The messages need to include an acknowledgement that they received the previously sent message. Then the two sides will know what messages have been received and which messages failed to be received.

2.

a. If the acknowledgement fails to arrive then packet 0 will be resent instead of packet 1.




b. If a premature timeout occurs a delayed acknowledge 1 could arrive after already receiving a different one and sending a 0, meaning a ack 0 was expected but an ack 1 arrived




3.

John Donnelly
Jed126
Networks

If you sent k -2 to k and only k - 1 and k - 2 make it. Say ack(k - 1) is then sent back but delayed. Then if k - 2 to k is resent and they all make it ack(k) could make it back and beat the delayed ack(k -1).

4.

They would both work. SR is good at handling out of order data (and lost data), there would be no issues. The windows won't change until all bits are received so there will be no issue with increasing numbers.

GBN would work as well but only because of the unlimited number sequence. It will not be very efficient as if bits 1-2-3-4 are sent, it could receive them in the order 4-3-2-1 which would only let 1 get acked. This would be very inefficient as 2-4 would need to be resent and it could keep happening.

5.

a.

So if 1 is sent and 2-3-4 are lost. ack(1) is received and 2-3-4-1 are sent. Then if 2 and 3 are lost, and 4 and 1 make it, ack(1) will be sent back. The sender though will think that 2-3-4-1 were all received since it has no way to tell which 1 was acked. It would assume all the data was received and keep going on with the sequence.

b.

The windows must be one less than the sequence length. If the sequence is 1-4 the window must be 3. Then when say 3-4-1 are sent and 1 is the only one that made it, ack(2) would be sent back still, so the correct 3-4-1 would be reset. The problem only occurs when the last received packet shares the same name with one of the ones also being sent. I.e. 2-3-4-1 is sent and ack(1) was the lasted ack so the new sent 1 will mess things up like part a.

6.

a. A group a 120 (1-120) bits could be sent and then the ack delayed several seconds. Then a resend could occur and be acknowledged. It's possible that after around 4 seconds the full sequence of packets (1-240) was acked and is now back to 1-120 again. Then if the original ack for 1-120 arrives, the next group of data would be sent even though the intended 1-120 for the second cycle through the sequence may never have been received.
b. It needs to be larger than the possible timeouts. So if 10 seconds is the max delay length, and a timeout happens after 1 second, and the transmission speed is 300 packets per second there must be enough so that after 10 seconds the sequence can't be completed (It's possible to send 1-120 and have 1-120 arrive from a difference cycle almost immediately, meaning no 1 second timeout). 300 * 10 = 3000 is the sequence length needed.

7. A.

The formula is

EstimatedRTT(1) = SampleRTT(1)

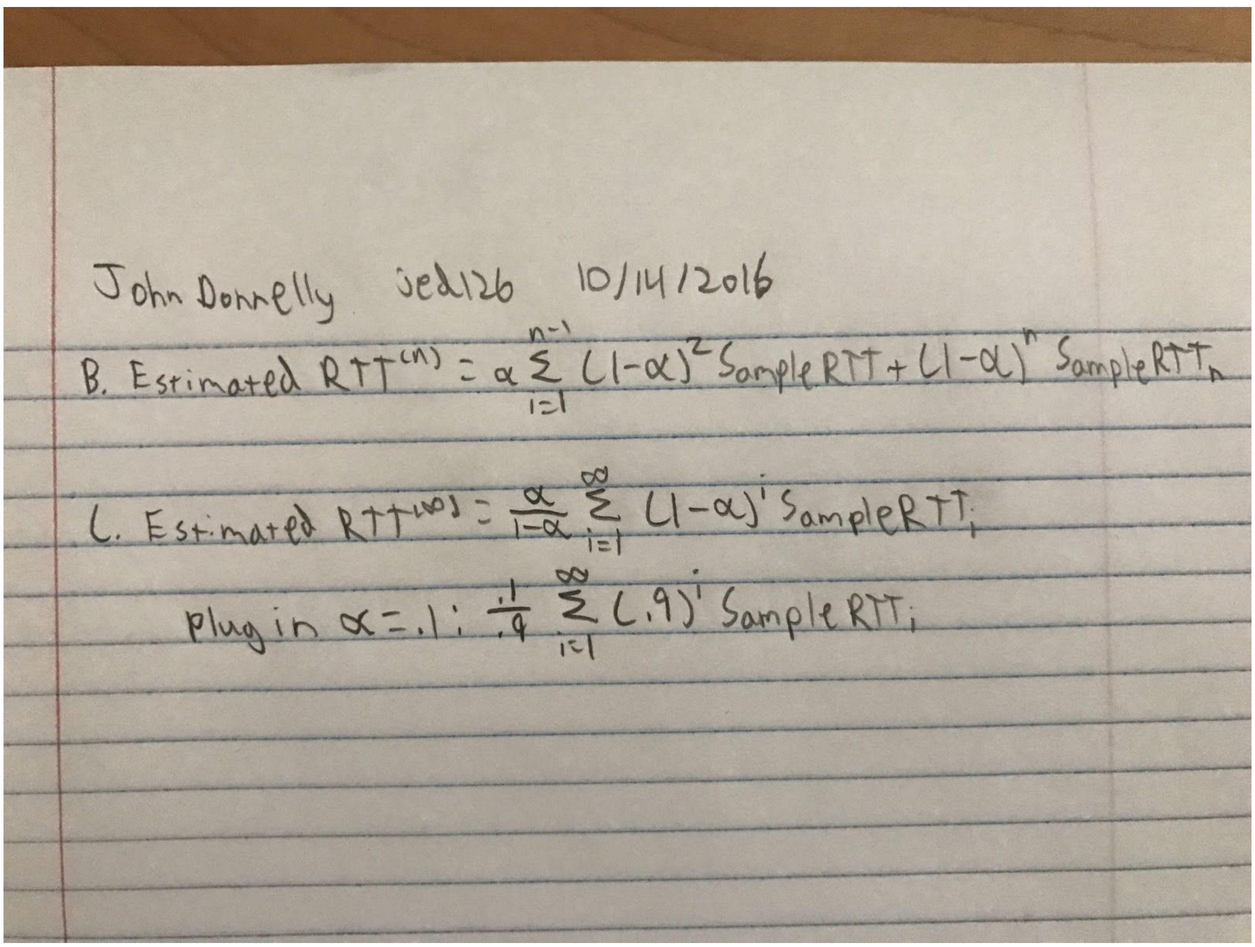
John Donnelly
Jed126
Networks

EstimatedRTT(2) = (1-α)EstimatedRTT(1) + αSampleRTT(1) = (1-α)SampleRTT(2) + αSampleRTT(1)

EstimatedRTT(3) = (1-α)EstimatedRTT(2) + αSampleRTT(1) = (1-α)^2SampleRTT(3) + α(1-α)SampleRTT(2) + xSampleRTT(1)

EstimatedRTT(4) = (1-α)EstimatedRTT(3) + αSampleRTT(1) = (1-α)^3SampleRTT(4) + α(1-α)^2SampleRTT(3) + α(1-α)SampleRTT(2) + αSampleRTT(1)

B. B and C were difficult to type out, so I wrote them down and uploaded it:



John Donnelly Jed126 10/14/2016

B. $\text{Estimated RTT}^{(n)} = \alpha \sum_{i=1}^{n-1} (1-\alpha)^2 \text{SampleRTT} + (1-\alpha)^n \text{SampleRTT}_n$

C. $\text{Estimated RTT}^{(\infty)} = \frac{\alpha}{1-\alpha} \sum_{i=1}^{\infty} (1-\alpha)^i \text{SampleRTT}_i$

plug in $\alpha = .1$: $\frac{.1}{.9} \sum_{i=1}^{\infty} (.9)^i \text{SampleRTT}_i$

As for the second part of c, it is known as an exponential moving average because the weight of the past samples is decreasing at an exponential rate when alpha = .1

John Donnelly
Jed126
Networks

8. If it tried to measure the time for retransmitted segments it could receive the original ack which is still out in the wild somewhere.  This could throw off the estimate because it could be impossibly quick because it could by chance arrive almost instantly after the retransmit.