Faculty of Arts and Science
University of Toronto
CSC 457 - Principles of Computer Networks

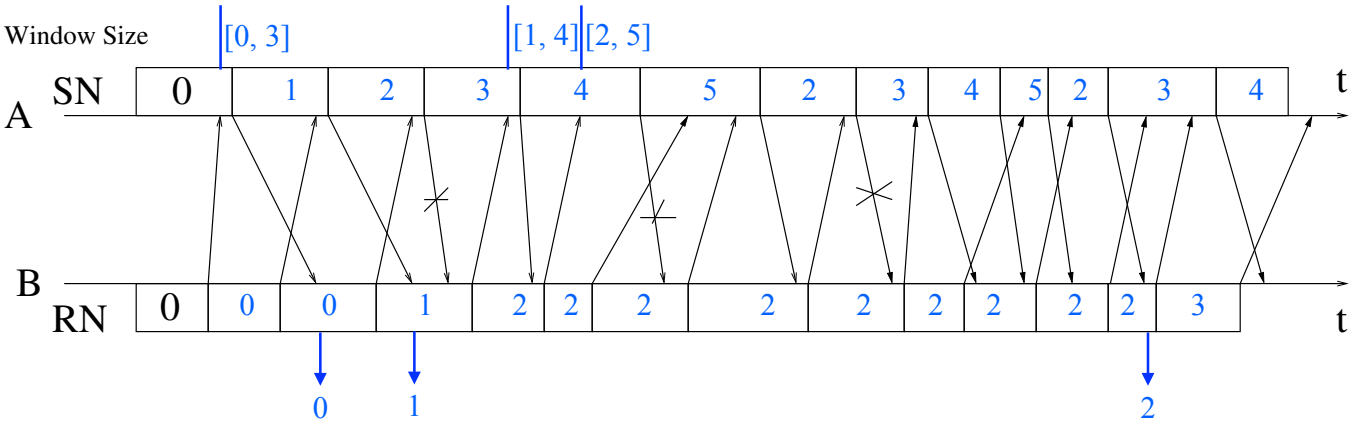
Assignment 2

Due Friday Oct. 23, 11:00pm.

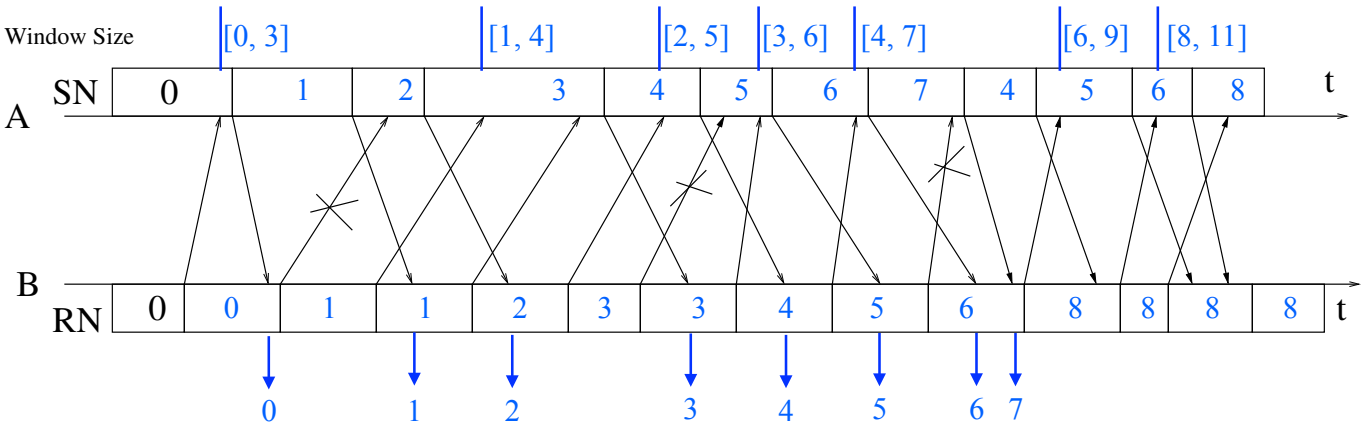
Question 1: Go-Back n ARQ

In this question, we review how Go-Back n ARQ can be used to ensure a reliable data transfer. Use $n = 4$ for Question (a)-(c). Use the convention that B when A has to retransmit packets it starts with the SN at the beginning of the window and retransmits packets in order of their sequence number.

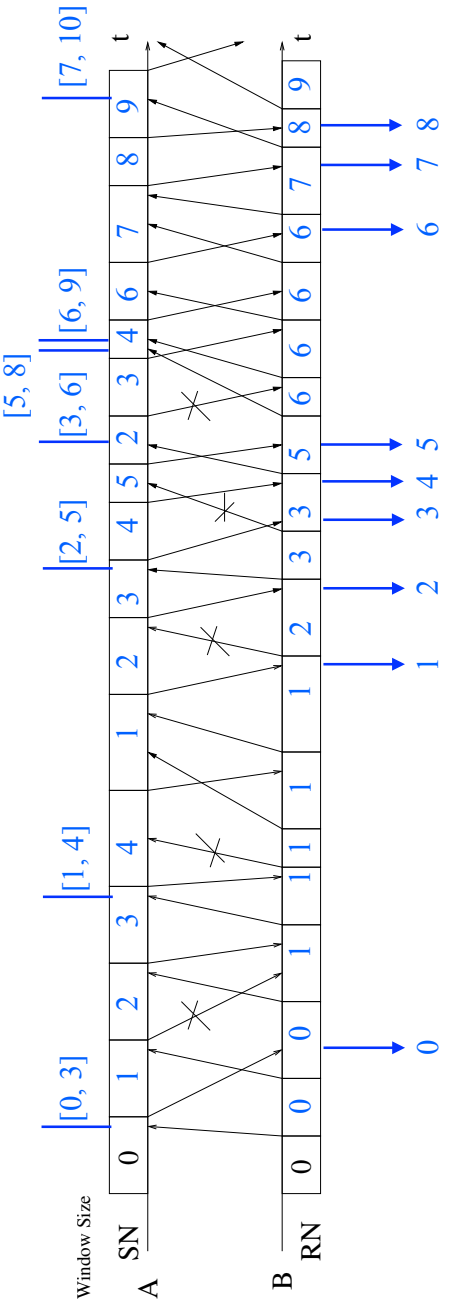
(a) (Error in Transmission from A to B) Fill in the values for SN and RN, indicate the window size, as well as the packets delivered to the next higher layer.



(b) (Error in Transmission from B to A): Repeat (a).



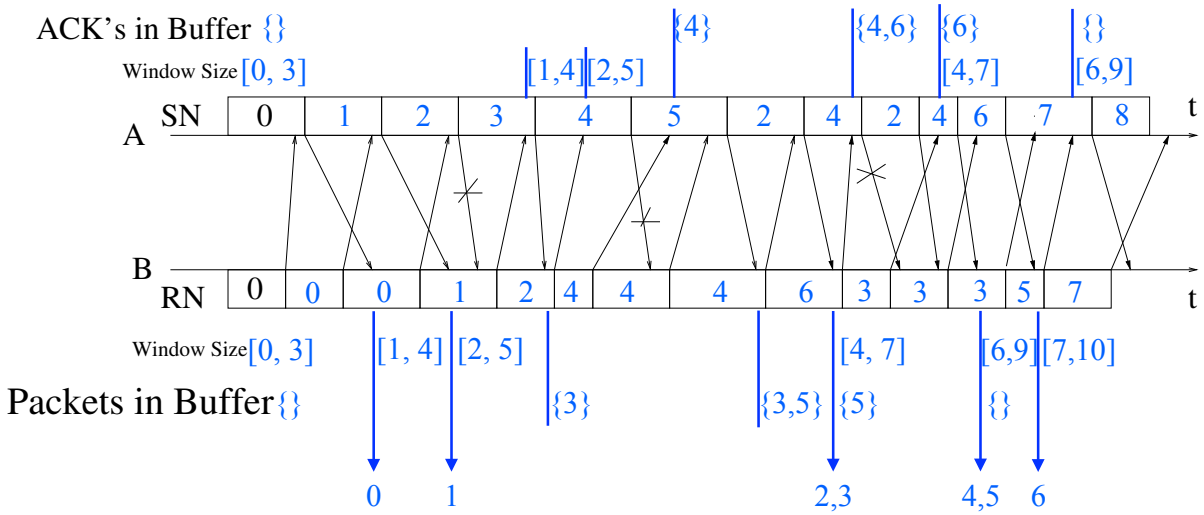
(c) Repeat (a).


Window Size
A
SN
B
RN
t
t
[0, 3]
[1, 4]
[2, 5]
[3, 6]
[5, 8]
[6, 9]
[7, 10]
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8

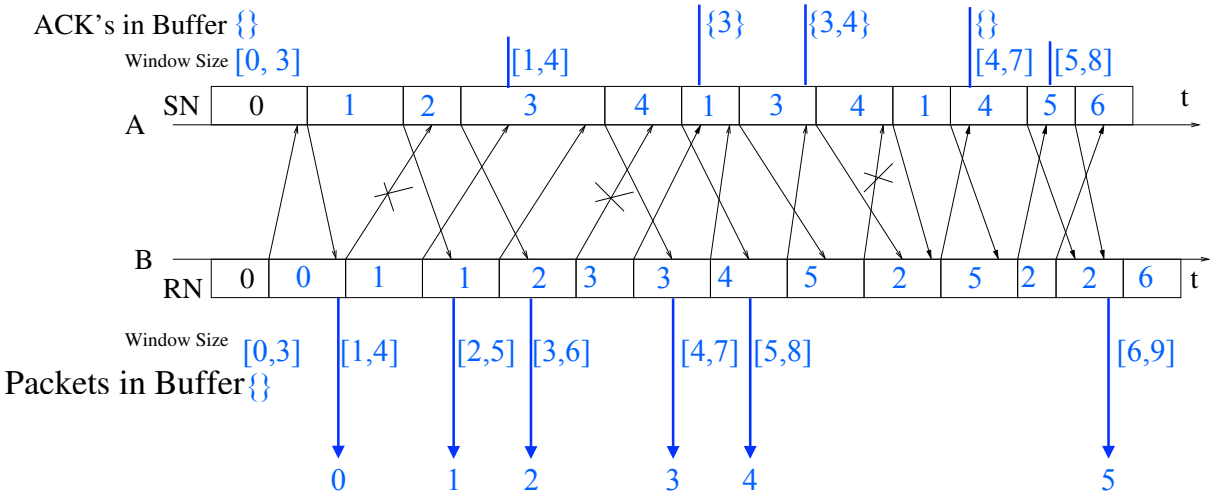
Question 2: Selective Repeat ARQ

In this question, we review how Selective Repeat ARQ can be used to ensure a reliable data transfer. Use $n = 4$ for Question (a)-(c). Use the convention that B always acknowledges the last error-free packet from A, and when A has to retransmit packets it starts with the SN at the beginning of the window and retransmits packets that have not yet been acknowledge by B in order of their sequence number.

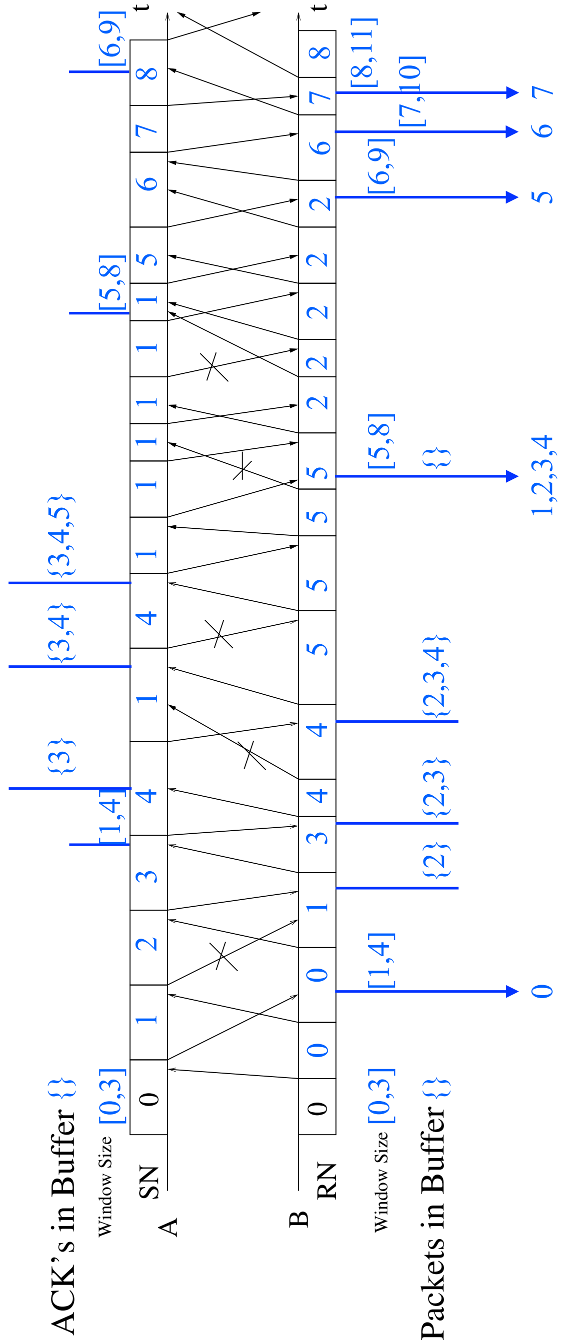
(a) (Error in Transmission from A to B) Fill in the values for SN and RN, indicate the window size at A and B, as well as the packets delivered to the next higher layer.



(b) (Error in Transmission from B to A): Repeat (a).



(c) Repeat (a).


ACK's in Buffer
Window Size
SN
A
B
RN
Window Size
Packets in Buffer
t
t

Question 3: Stop-and-Wait ARQ

Consider the situation where two peer processes, A and B, implement Stop-and-Wait ARQ to send data from A to B. The communication channel (link) between A and B (and B and A) is such that all packets that arrive are in the same order as transmitted. Below, we make different assumptions whether packets can arrive with errors and whether packets can be lost. For the different assumptions, indicate **whether it is necessary** to use sequences numbers SN for the packets sent from A to B, as well **whether it is necessary** to use request numbers RN for ACK's sent from B to A. To answer each sub-question, first indicate your answer by checking either "YES" or "NO". If you checked "YES" for SN and/or RN, then give a counter-example that shows that the protocol fails if we do not use SN and/or RN.

(a) Assume that packets (from A to B) and ACK's (from B to A) can have an arbitrary and variable delay, but the delay is always finite. Furthermore assume that packets and ACK's are **never lost** and always arrive **error free**.

	YES	NO
SN	✓	
RN	✓	

Suppose A sends a packet 0 that is received by B and an ACK is sent but before it is received A a time-out occurs so A resends packet 0. Eventually packet 0 is received twice violating the requirement of a reliable service that each packet is delivered once and only once.

Now, consider a fresh scenario where A attaches a SN to packet 0 that is received by B and an ACK is replied back but before it is received a time-out occurs and packet 0 with the same SN is sent, received, and replied with another ACK. If the first ACK is received A will send packet 1 with the next SN but if before B's ACK reply to packet 1 the second ACK for the resent packet 0 is received by A then the SN will increment potentially forever losing packet 1 if never received by B, or the transmission loses order if received.

By introducing a RN for the next awaited packet to be received by B that's sent instead of an ACK we arrive at a working protocol. Consider the same scenario pausing when packet 1 is lost but B sends its second RN for packet 1. Instead of A proceeding to send packet 2 it now resends packet 1. Now transmission order is preserved and packets are delivered once and only once.

(b) Assume that packets (from A to B) and ACK's (from B to A) can have an arbitrary and variable, but the delay is always finite.. Furthermore assume that packets and ACK's are **never lost** and that **ACK's always arrive error free**. However, **packets might arrive with errors**.

	YES	NO
SN		✓
RN		✓

Sender SN's and receiver RN's are not required when ACK's and NAK's are used.

(c) Assume that packets (from A to B) and ACK's (from B to A) can have an arbitrary and variable delay, but the delay is always finite. Furthermore assume that **ACK's are never lost** and **ACK's always arrive error free**. However, **packets might are lost or arrive with errors**.

	YES	NO
SN	✓	
RN		✓

In the scenario where A experiences a time-out and resends, say, packet 0 application B is unable to distinguish resent packets from first-sent packets. Thus, a SN is required but a RN is not because B can send no ACK and refuse to transmit to the next higher layer in the case that a packet arrives with a SN that has been seen before.

Question 4: Modulus m

For Go-Back n ARQ and Selective Repeat ARQ we have to be careful how we apply a modulus m to the sequence number. In this question, we illustrate this issue.

(a) Give an example where Go-Back n with modulus m fails if $m = n$. Use $n = 5$ for your example.

(b) Give an example where Selective Repeat with modulus m fails if $m = 2n - 1$. Use $n = 5$ for your example.

(a) Suppose we have a sending application A and a receiving application B that will transmit received packets to the next higher layer. Further, assume that the delay for B to send back a RN is arbitrary and variable. Now, let A send packets 0, 1, 2, 3, and 4 to B but in this duration of time beginning from receiving packet 0 until packet 4 is received B transmits a RN of 0. With arbitrary delay, say that RN 0 is received by A only after A has sent packet 0 again which is interpreted to be packet 5 by the next higher layer.

(b) Suppose that receiving application B sends its RN's with an arbitrary delay. Let sending application A transmit packets with SN 0, 1, 2, 3, and 4 that are received by B sequentially pushing B's RN to 5 and its window to [5, 10]. However, on receiving packets with SN 0, 1, 2, 4, and 4 suppose the five ACK's that B sends to A are lost. Now, due to time-out, A resends the packet with SN 0 but B's window is [5, 10] so B stores the resent packet in its buffer because it is considered out-of-order. Since all five ACK's sent by B are lost, time-outs keep occurring and A will continually send sequences of packets with SN's 0, 1, 2, 3, and 4 but each will be placed into B's buffer. Buffers are finite in size and eventually a buffer overflow will occur. However, if m is greater or equal to 2n then A can always send packets with SN's within B's window so transmission continues.

Faculty of Arts and Science
University of Toronto
CSC 457 - Principles of Computer Networks

Cover sheet for Assignment 2

Complete this page and attach it to the front of your assignment.

Name: Grant Wong
(Underline your last name)

Student number: 1000549445

I declare that this assignment is solely my own work, and is in accordance with the University of Toronto Code of Behavior on Academic Matters.

Signature: