

# **EECS 489 – FA 21**

## **Discussion 5**

# Assignment-2

Assignment 2 is out. Due date: **10/22 2021, 11:59 PM**

Much harder than A-1. (~1,000 loc)

# START EARLY!

Hosted in GitHub under <https://github.com/eecs489>

Please make sure you are in the correct GitHub team and have access to your repo.

# Q1 HTTP Streaming

Consider a simple HTTP streaming model.

$Q$  is the number of bits that must be buffered before the client application begins playout.

$r$  is the video consumption rate.

$x$  is the bits sending rate whenever the client buffer is not full.

Also assume that  $x < r$ .

Describe the behavior of the video output:

Alternate between playout and freeze.

Suppose the buffer starts out empty.

How long will it be before the video can begin playout?

$$Q / x$$

Assume the current buffer size is  $z$  ( $> Q$ ).

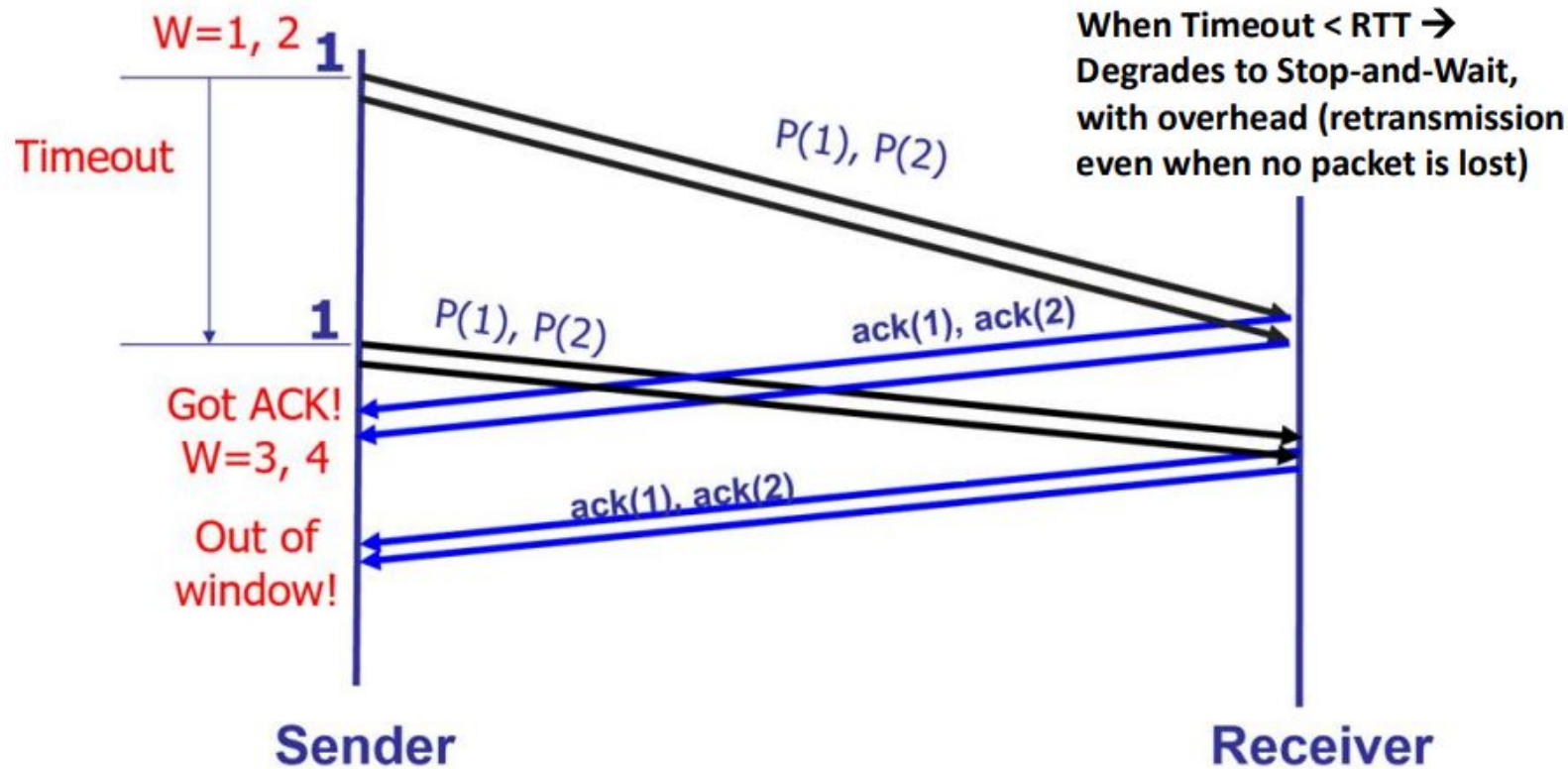
How long will the playout last?

$$z / (r - x)$$

# Q2 Selective Repeat

With the Selective Repeat (SR) protocol, Is it possible for the sender to receive an ACK for a packet that falls outside of its current window? Why?

True



# Q3 Go-Back-N

With the Go-Back-N (GBN) protocol, Is it possible for the sender to receive an ACK for a packet that falls outside of its current window? Why?

**True**

**Same reason**

# Q4 (N)ACK

Consider a reliable data transfer protocol that uses only negative acknowledgments (NACK). Suppose the sender sends data only **infrequently**. Would a NACK-only protocol be preferable to a protocol that uses ACKs? Why?

- ACK: send ACK upon packet arrival
- NACK: send NACK upon packet loss

**No. In a NAK only protocol, the loss of packet  $x$  is only detected by the receiver when packet  $x+1$  is received. If there is a long delay between the transmission of  $x$  and the transmission of  $x+1$ , then it will be a long time until  $x$  can be recovered, under a NAK only protocol**

# Q5 (N)ACK

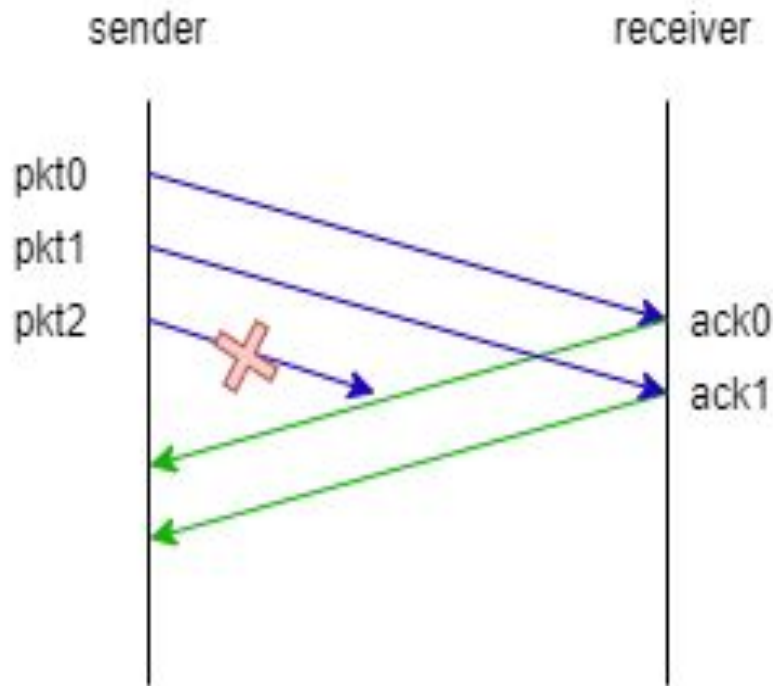
Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NACK-only protocol be preferable to a protocol that uses ACKs? Why? (Assuming ACK/NACK is never lost)

- ACK: send ACK upon packet arrival
- NACK: send NACK upon packet loss

**Yes. If data is being sent often, then recovery under a NAKonly scheme could happen quickly. Moreover, if errors are infrequent, then NAKs are only occasionally sent (when needed).**

# Q6 Sliding Window Protocols (I)

Consider the sliding window protocol in the following figure. Does this figure indicate that GBN is being used, SR is being used, or there is not enough information to tell?

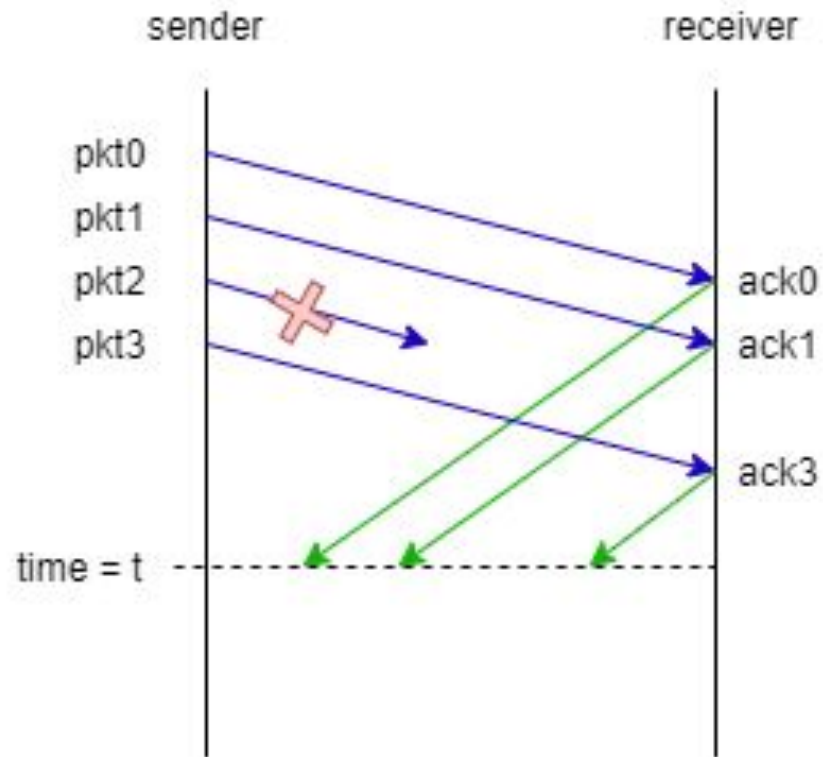


**There is not enough information to tell, since both GBN and SR will individually ACK each of the first two messages as they are received correctly.**



# Q7 Sliding Window Protocols (2)

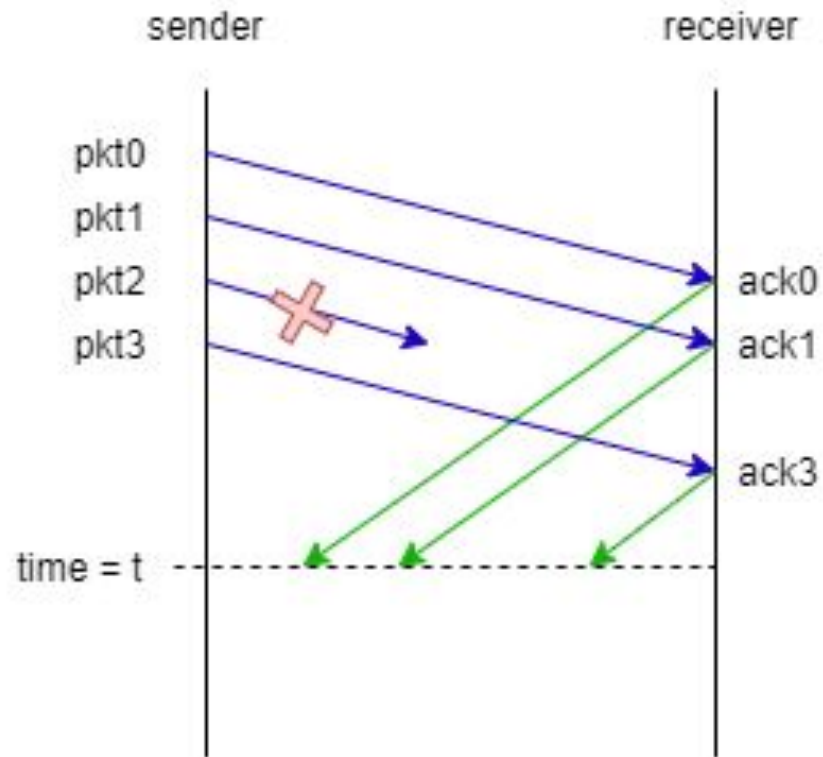
Consider the sliding window protocol in the following figure. Does this figure indicate that GBN is being used, SR is being used, or there is not enough information to tell?



**This must be the SR protocol since **pkt3** is acked even though **pkt2** was lost. GBN will discard **pkt3** if **pkt2** was dropped.**

# Q8 Sliding Window Protocols (3)

Consider the sliding window protocol in the following figure. Suppose the window size = 5 for both sides. Show the positions of windows for the sender and receiver at time =  $t$  (no ack has been received).



Sender:

0 1 2 3 4 5 6 7 8 ...

Receiver:

0 1 2 3 4 5 6 7 8 ...

# Thanks

Have a good one!