EECS 489 - Winter 2024

Discussion 5

Reminders

- Assignment 2 is due February 23rd @ 11:59 pm EST
  - Next week!
  - 15% of your overall grade
- Autograder: https://eecs489.eecs.umich.edu/
  - 3 submits per day!
- Repositories for submitting under https://github.com/eecs489

Assignment 2 is due soon!

- If you have not started yet…

<u>START NOW</u>

- This is considered the hardest project in the class
- 2 large components that can be done in parallel
- ~1000 LOC

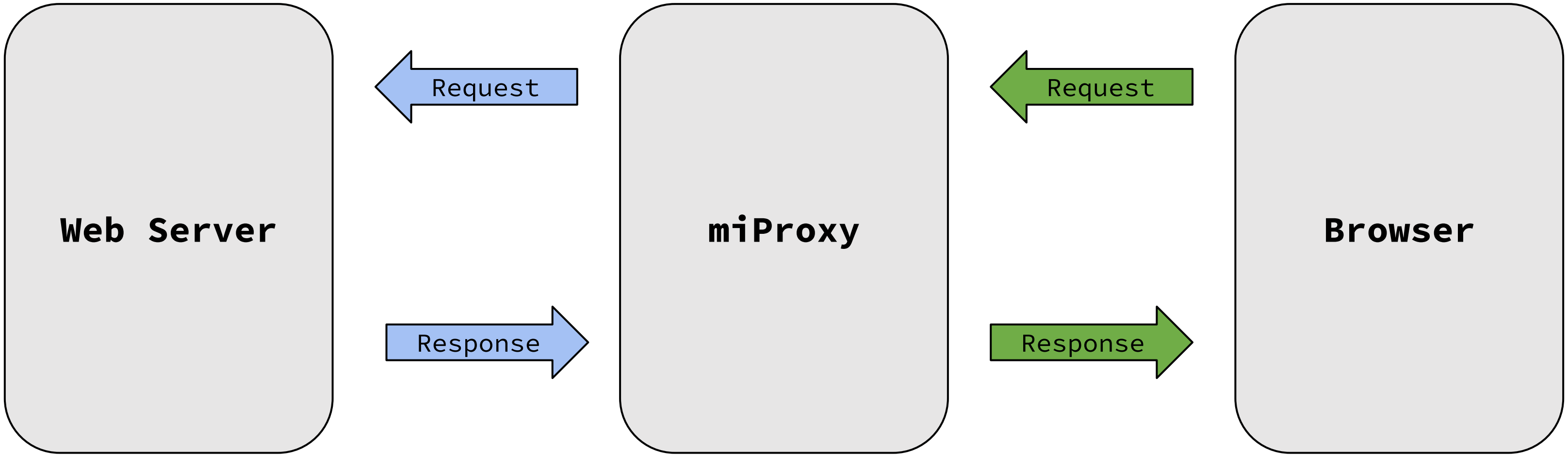
Midterm

- The Midterm will be on **Wednesday, March 6th** from **10:30 am - 12:00 pm EST**
    - This is class time on Wednesday that week
- Full logistics will be sent out before Spring Break

Today

- Assignment 2 Overview
- HTTP Streaming
- TCP Basics
- Flow + Congestion Control

Assignment 2 Overview



- The proxy only forwards messages between the browser and the web server
- It doesn’t care what is forwarded
  - **Do not** make any assumptions (unless given) on what is forwarded

Assignment 2 Overview

- Short Demo!

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.
    - Assume that $x < r$.
- Describe the behavior of the video output:

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.
  - Assume that $x < r$.
- Describe the behavior of the video output:
  - Alternates between layout and freezing

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.
  - Assume that $x < r$.
- Suppose the buffer starts out empty. How long will it be before the video can begin playout?

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.
  - Assume that $x < r$.
- Suppose the buffer starts out empty. How long will it be before the video can begin playout?
  - Q / x

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.
  - Assume that $x < r$.
- Assume the current buffer size is z (> Q). How long will the playout last?

Q1: HTTP Streaming

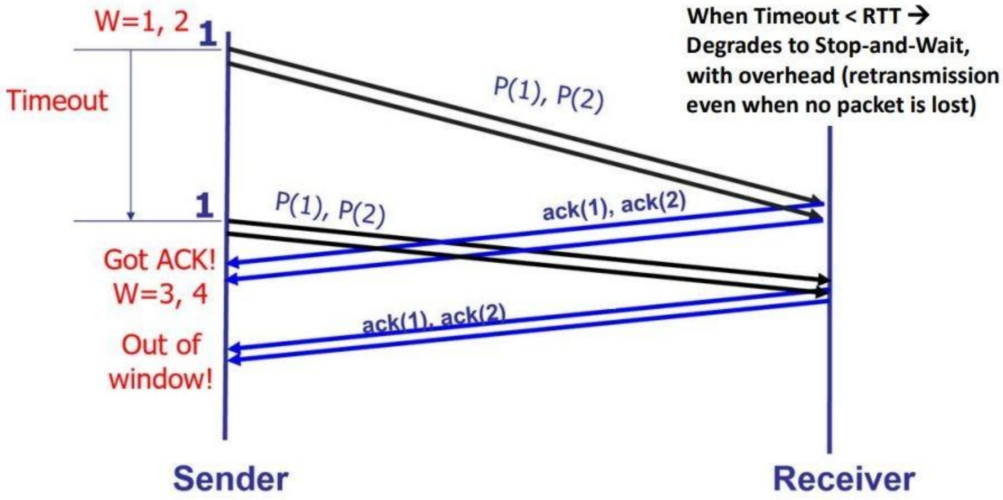
- Consider a simple HTTP streaming model.
  - Q is the number of bits that must be buffered before the client application begins playout.
  - r is the video consumption rate.
  - x is the bits sending rate whenever the client buffer is not full.
  - Assume that x < r.
- Assume the current buffer size 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?

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



When Timeout < RTT →
Degrades to Stop-and-Wait,
with overhead (retransmission
even when no packet is lost)

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?

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, for the same reason as for Selective Repeat

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

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 NACK 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 NACK only protocol

Q5: (N)ACK (again)

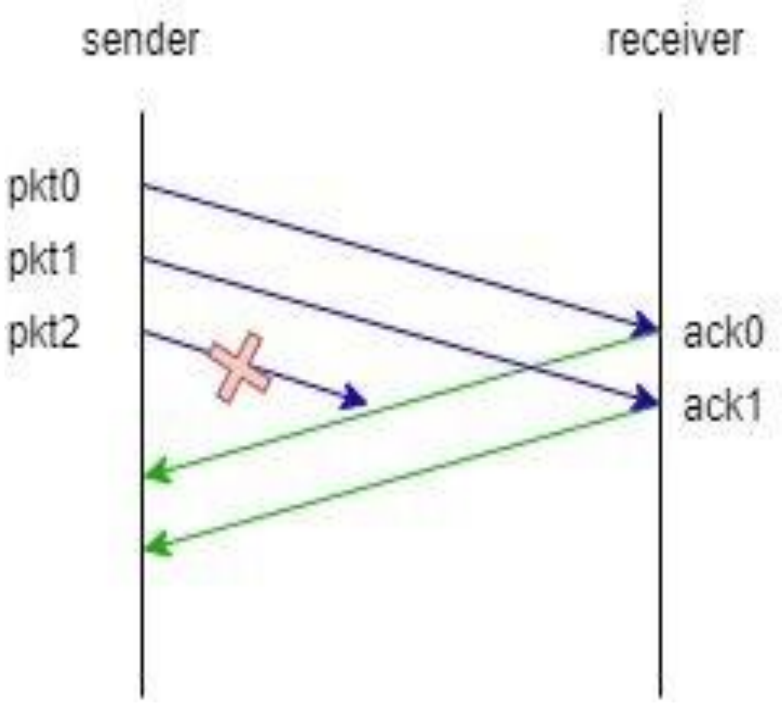
- Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NACK-only protocol be preferable to a protocol that uses ACKs? Why? (Assuming ACK/NACK is never lost)
    - ACK: send ACK upon packet arrival
    - NACK: send NACK upon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

Q5: (N)ACK (again)

- 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 NACK only scheme could happen quickly. Moreover, if errors are infrequent, then NACKs are only occasionally sent (when needed).

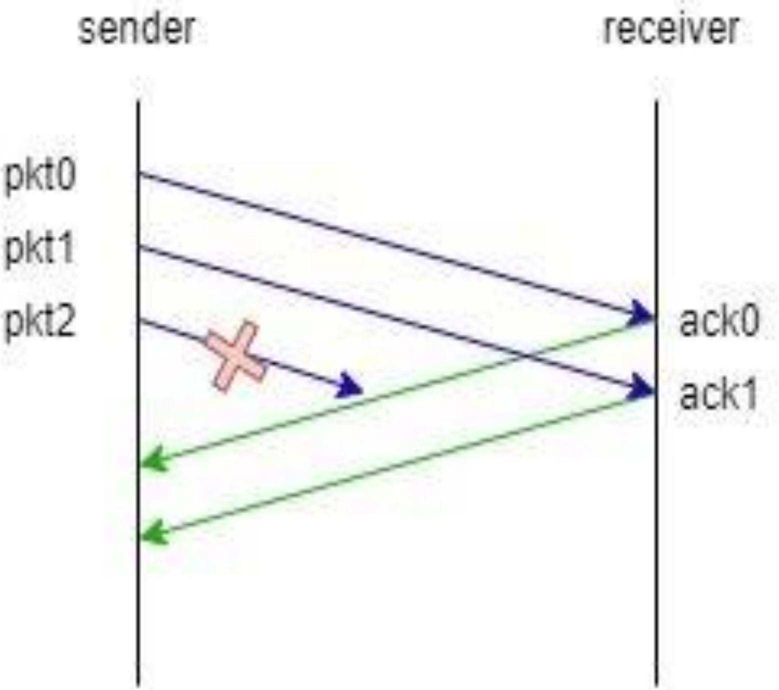
Q6: Sliding Window Protocols (1)

- 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?



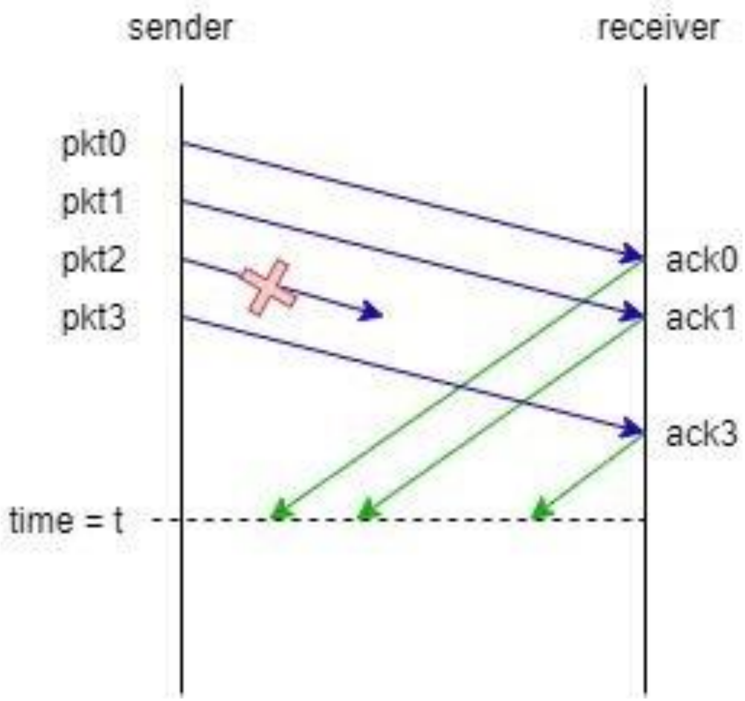
Q6: Sliding Window Protocols (1)

- 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.



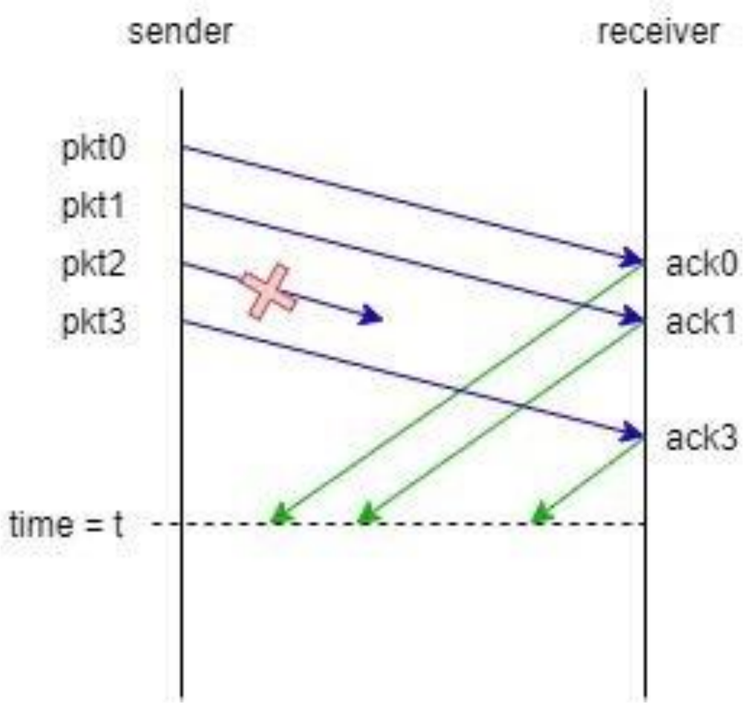
Q7: Sliding Window Protocols (2)

- 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?



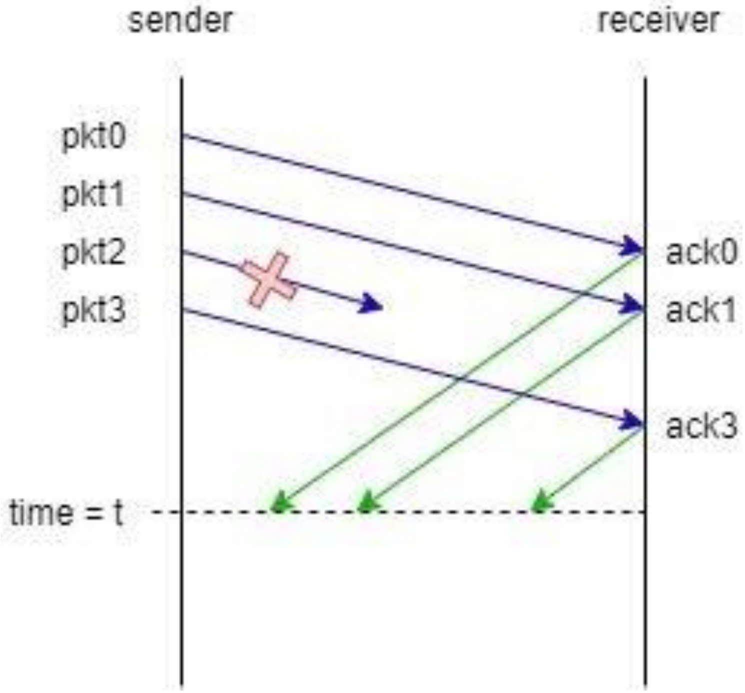
Q7: Sliding Window Protocols (2)

- 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.



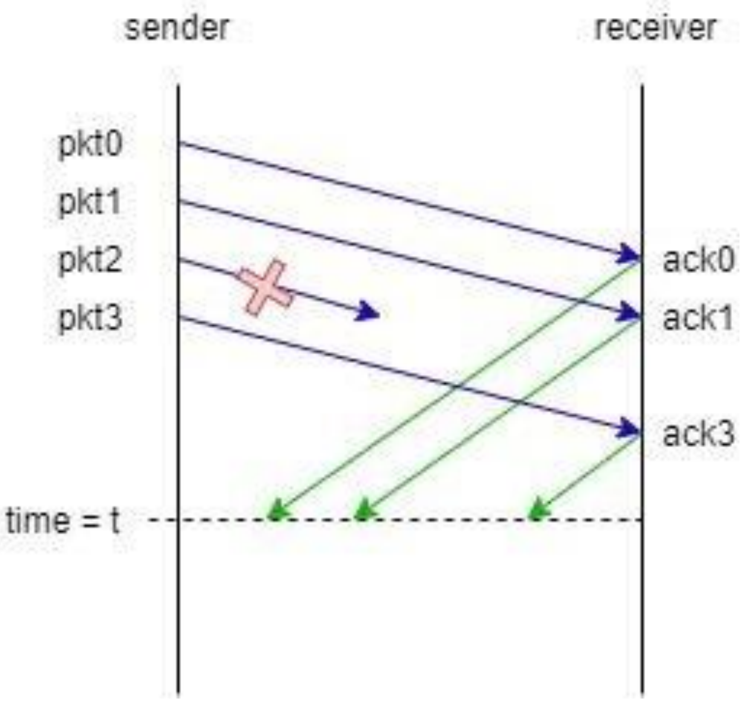
Q8: Sliding Window Protocols (3)

- 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).



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 …
- Reason: pkt2 failed to deliver, while pkt3 was successfully received. This means the sender window is not shifted yet, while the receiver has since it has gotten pkt0 and pkt1



Wrap-Up

- Thanks for coming!
- Make sure to continue working on Assignment 2!
    - Due next week!
- Start to think about the midterm!