CPSC 317 COMPUTER NETWORKING

2023W2: Transport – Day 4 - Windowing Protocols

1

READING

- Reading: 3.4.2, 3.4.3, 3.4.4

LEARNING GOALS

Explain the throughput problem with the alternating bit protocol

Sliding Window Protocols

- Determine an appropriate window size
- Trace the execution of Go-Back-N (GBN)
- Trace the execution of Selective-Repeat (SR)
- Analyze GBN and SR under segment loss
- Trace the execution of GBN and SR when the range of sequence numbers is restricted
- Given a range of sequence numbers determine if a set of sender and receiver window sizes is legal

ALTERNATING BIT PROTOCOL IN PRACTICE

- Assume a connection from Vancouver to Montreal
  - Round Trip Time is 30 ms
  - Link speed (bandwidth) is 1Gbps
  - Segment size is 1000 bytes, including overhead
- How much of the link bandwidth are we actually using?
- Transmission delay, one segment: $\frac{8000}{10^9} = 0.008\, ms$
- Throughput: $\frac{8000}{30+0.008} = 0.267\ Mbps$
- Link bandwidth utilization: $\frac{0.267}{1000} = 0.00027$ (or 0.027%)

SENDING MULTIPLE SEGMENTS

- Sender can send multiple segments
- Don’t wait for each acknowledgement

CLICKER QUESTION

- Assume a connection from Vancouver to Montreal
  - Round Trip Time is 30 ms
  - Link speed (bandwidth) is 1Gbps
  - Segment size is 1000 bytes, including overhead
- How many segments do I need to have going at once to get the utilization up to 1%? I want an integer!

WHY NO FINITE STATE MACHINE?

- When we allow multiple segments in flight, modelling this with a finite state machine is less than ideal
- Why?

WINDOWING PROTOCOLS: THE BASIC IDEA

- Sender sends a bunch of segments before waiting for an ACK
  - How many?
- Expand sequence numbers to integers to account for multiple in-transit and unacknowledged packets
- Any or all of these segments might get lost
- Sender has to be ready to re-send any segments that get lost
  - Buffer in-transit segments until acknowledged
- Receiver has to be ready to handle segments arriving out of order
  - Buffer segments received out of order under "gaps" filled

SENDER'S WINDOW

- Sender's window: range of segments that are stored for potential re-send
- Usually considered to be fixed size
- Window only moves when the first segment in the window is acknowledged
  - What if other ACKs are received?
- New segments are sent only when they "fit" in the window

RECEIVER'S WINDOW

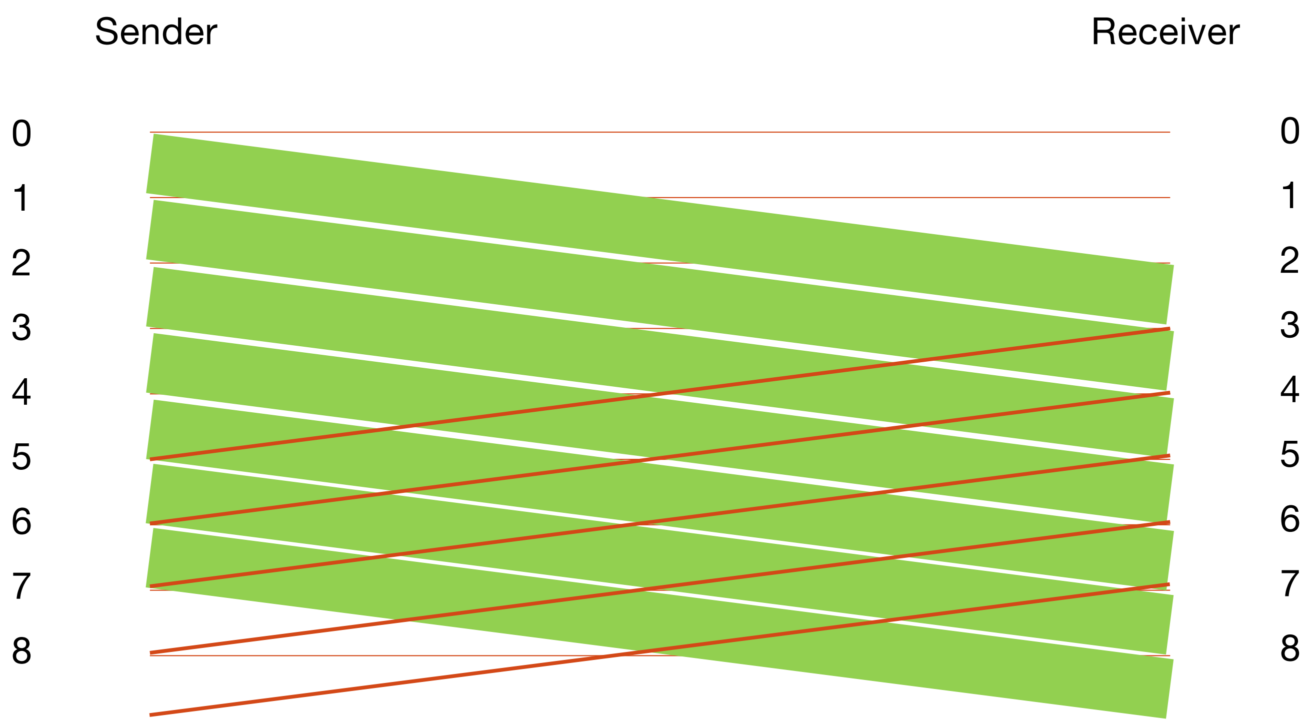
- Receiver's window: store segments received out of order
  - (Out of order because of drops or re-ordering in the network)
  - If window is 1 segment: out of order segments are dropped
- Once missing segments arrive, window is processed
- Segments received beyond the window limit will be discarded

IN A PERFECT WORLD …

- Assume:
  - It takes 1 time unit for the sender to transmit each segment
  - It takes 2 time units for a bit to get from the sender to the receiver or from the receiver to the sender
  - No segments are ever lost or delayed
- How large of a sender window do we need to enable the sender to keep sending all the time?

CLICKER QUESTION

- Assume:
  - It takes 1 time unit for the sender to transmit each segment
  - It takes 2 time units for a bit to get from the sender to the receiver or from the receiver to the sender
  - No segments are ever lost or delayed
- How large of a sender window do we need to enable the sender to keep sending all the time?


Sender
Receiver
0
1
2
3
4
5
6
7
8
0
1
2
3
4
5
6
7
8

PROBLEMS TO CONSIDER

- Sender
  - How does the sender know that data got lost?
  - Can lost data be distinguished from a lost ACK?
  - If we send more than one segment, how many segments can we remember?
- Receiver
  - How can you tell if data is out-of-order or missing?
  - What should be ACKed?

GO-BACK-N STRATEGY

Receiver:

- Window of size 1
- When a segment is received, send an **ACK for the last segment that was received in order**
- Deliver the arriving segment if received in order, otherwise discard the segment

Sender:

- ***Sender's window*** determines the number of outstanding (unacknowledged) segments held in memory
- Start timer on first segment sent
- Received ACKs may be cumulative (re-start timer on receipt)
- On timeout go to last unack'ed segment and re-send everything (re-start timer)

GO-BACK-N DEMO

https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/go-back-n-protocol/index.html

SEQUENCE NUMBER RANGE

- The range of possible sequence numbers is limited by n, the number of bits used to represent them
  - Example: 3 bits gets numbers 0-7, 8 bits gets numbers 0-255
  - Sequence number arithmetic is modulo 2^n (e.g., with $n = 3$ bits, the range is 0-7, so after 7 comes 0)
- What is the maximum sender window size for the range 0-255?
  - Maybe easier to compute: what about range 0-3?
  - Can the receiver distinguish a new 0 from a resent old 0?

SEQUENCE NUMBER RANGE

- Assume that segments can't be re-ordered in the network
- Rule: sender's window size + receiver's window size <= sequence number range
- For a range with n numbers (0 to $n-1$):
  - Go-Back-N:
    - receiver's window size is 1
    - sender's maximum window size is $n-1$
- Why?

IN-CLASS ACTIVITY

- ICA44