

# Chapter 16b

## Protocol Techniques

### Network Services

Network protocols need to provide more complex services than just best-effort packet delivery. Examples include:

- Connections/Connectionless
- Reliable Delivery
- Ordered Delivery
- Request/Reply Communication
- QoS
- ... etc ...

The following briefly outlines common techniques used to implement some of these services.

## Reliable Delivery

- **Question:** What does **Reliable Delivery** mean?
  - **Answer:** it depends.
- 
- Reliable delivery **minimally** means:
    1. no data values change
    2. no data is missing
    3. no data is added/duplicated
  - Minimal requirements called **Error Control**
  - Three other related services:
    1. **Ordered Delivery**
    2. **Flow Control**
    3. **Congestion Control**
  - we will define each of these services and describe common techniques to implement them

## Data Corruption

- Goal: make sure the data does not change during transmission
- Techniques:

## Packet Loss

- Packet is dropped/lost before reaching the receiver
- Probably the most widespread/common networking problem
- Why are packets lost?

- Packet Loss is a really tough problem. Why?

## Recovering From Packet Loss

- **Question:** How do you recover from packet loss (assuming you can detect it)?
  - **Answer:** Send the packet again, and again, and again, ... until it gets there.
- 

- Basic Error Control Algorithm uses:
  - Positive acknowledgements
  - Timeouts
  - Retransmissions
- Protocol works as follows:
  - normal (errorless) mode:
  - packet loss mode:



## Duplicate Packets

- Because we don't know if the packet was lost or just delayed, the destination may receive **Duplicate Packets**
- Solution (basic idea): discard duplicates before handing them to the application.
- This is not good enough. Why?
  
- How do you know it is a duplicate?



## Stop-N-Go

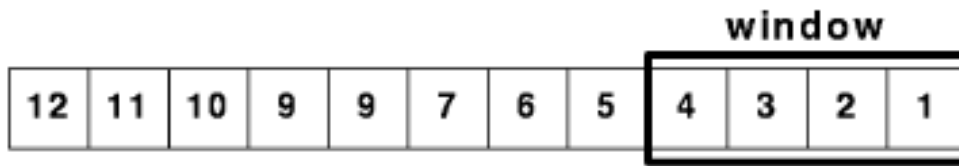
- Sender only sends one packet at a time.
- Receiver sends small control packet when it is ready for next packet
- Sender waits for control packet before sending next packet
- Can be very inefficient. Why?

## Sliding Window

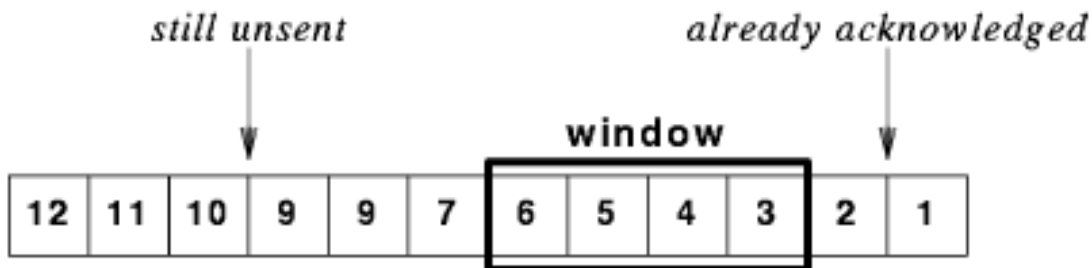
- Allows sender to transmit multiple packets before receiving an acknowledgment
- Number of packets that can be sent is defined by the protocol and called the window
- As acknowledgments arrive from the receiver, the window is moved along the data packets; hence the term “*sliding window*”

## Sliding Window Example

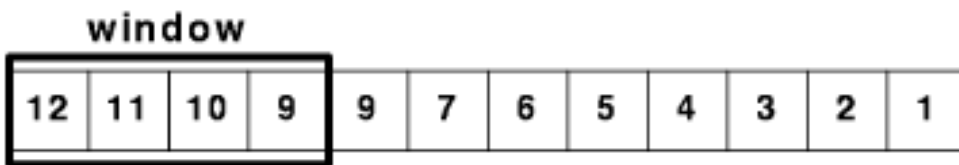
- data packets buffered from right to left
- window limits how many unacknowledged packets you can send



(a)



(b)

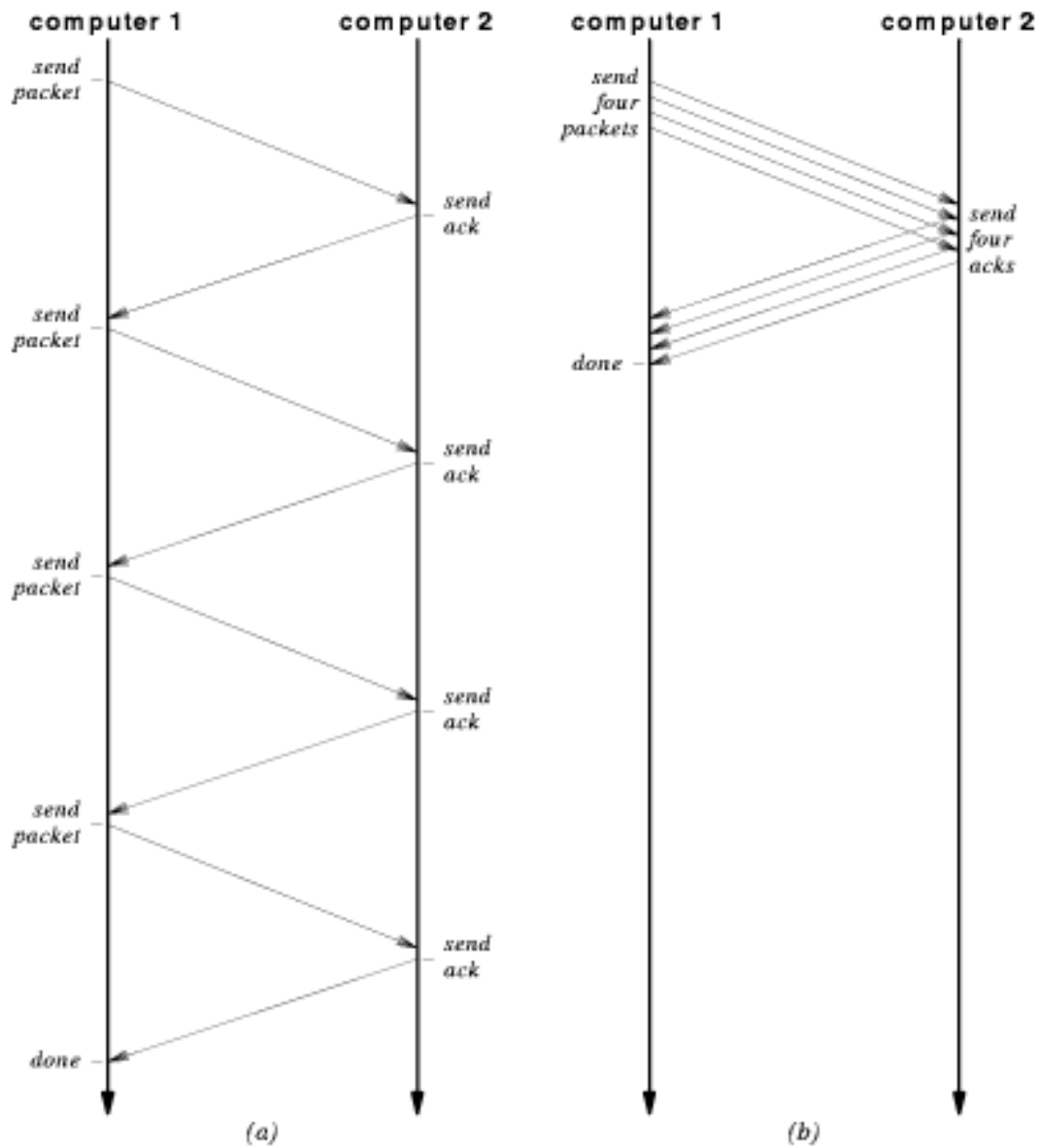


(v)

Figure 1: Window moves from right to left

## Stop-N-Go vs. SWP

- stop-n-go wastes bandwidth
- SWP needs to buffer and manage the window



## Sliding Window Details

- Goal: send multiple unack'ed packets to make better use of the pipe than Stop-N-Go protocols.
  - SubGoal: Don't send so fast that you overrun the receiver
- 
- **Sender:**
    - sender sends up to a window full of data, then blocks and waits for acknowledgements
    - sender receives **cumulative acks** , which advances the window
    - advancing win means more pkts can be sent
    - each packet has its own timeout and is retransmitted if not cumulatively ACK'd in time.

## Sliding Window Details: (continued)

- **Receiver:**
  - only accept packets in the window
  - if the packet that arrives is not the next frame expected (NFE), do not ACK it
  - when NFE arrives, calculate the highest consecutive packet received, send a cumulative ACK for it, reset NFE to the following packet, and advance window

## Window Sizes

- Who defines the window size?
- What should the window size be?
- Do the window sizes need to be the same?

# Congestion Control

- Network congestion arises in network systems that include multiple links
- If input to some link exceeds maximum bandwidth, packets will queue up at connection to that link

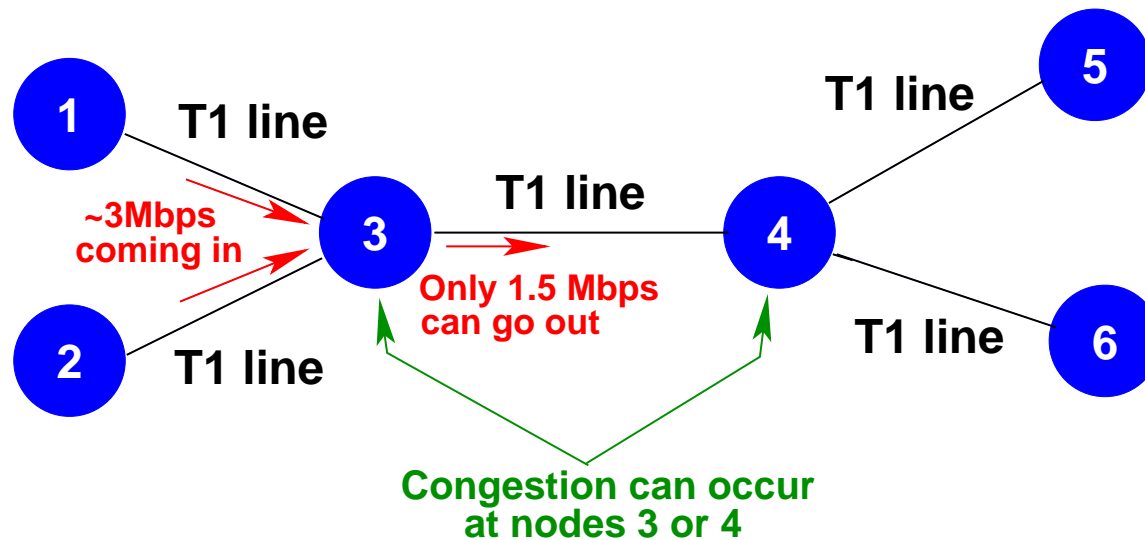


Figure 3: Network Congestion possible at 3 and 4.

## Congestion Control: (continued)

- Eventually, packets will be discarded and packets will be retransmitted
- Ultimately, network will experience congestion collapse
- Problem related to, but not identical to, data overrun