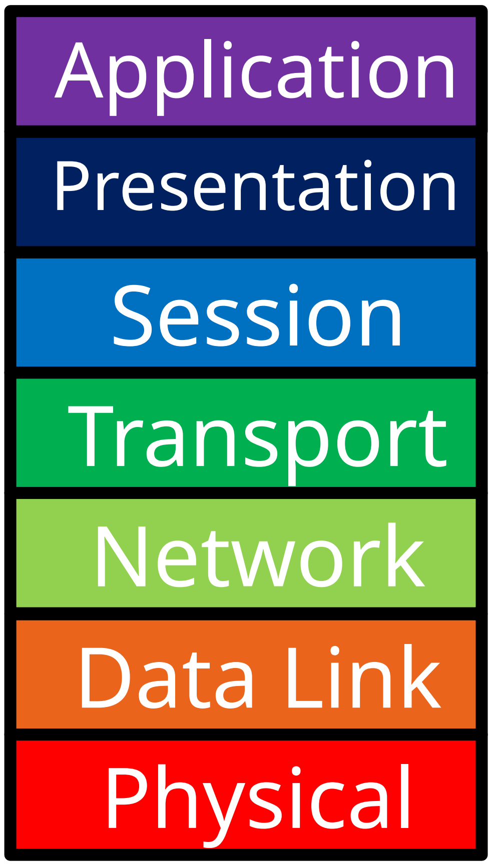
Computer Networks

Lecture 3: Data Link - part II

Based on slides from D. Choffnes Northeastern U. and P. Gill from StonyBrook University
Revised Autumn 2015 by S. Laki

Data Link Layer



- Function:
  - Send blocks of data (frames) between physical devices
  - Regulate access to the physical media
- Key challenge:
  - How to delineate frames?
  - How to detect errors?
  - How to perform media access control (MAC)?
  - How to recover from and avoid collisions?

Outline

- Framing
- Error Checking and Reliability
- Media Access Control
  - 802.3 Ethernet
  - 802.11 Wifi

Error control

- Error Control Strategies
  - Error Correcting codes (Forward Error Correction (FEC))
  - Error detection and retransmission Automatic Repeat Request (ARQ)

Error control

- Objectives
  - Error detection
    - with correction
      - Forward error correction
    - without correction -> e.g. drop a frame
      - Backward error correction
      - The erroneous frame needs to be retransmitted
  - Error correction
    - without error detection
      - e.g. in voice transmission

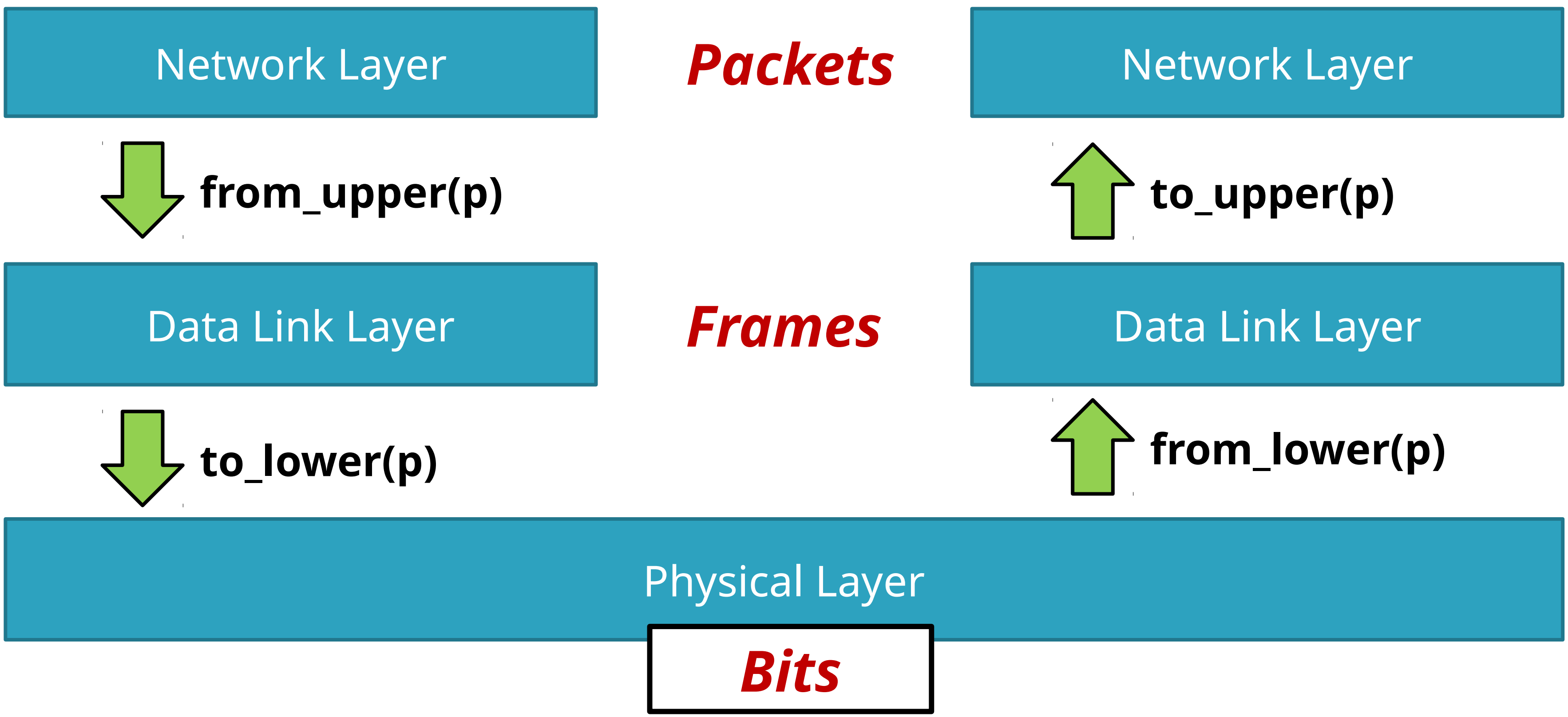
Should We Error Check in the Data Link?

- Recall the End-to-End Argument
- Cons:
  - Error free transmission cannot be guaranteed
  - Not all applications want this functionality
  - Error checking adds CPU and packet size overhead
  - Error recovery requires buffering
- Pros:
  - Potentially better performance than app-level error checking
- Data link error checking in practice
  - Most useful over lossy links
  - Wifi, cellular, satellite

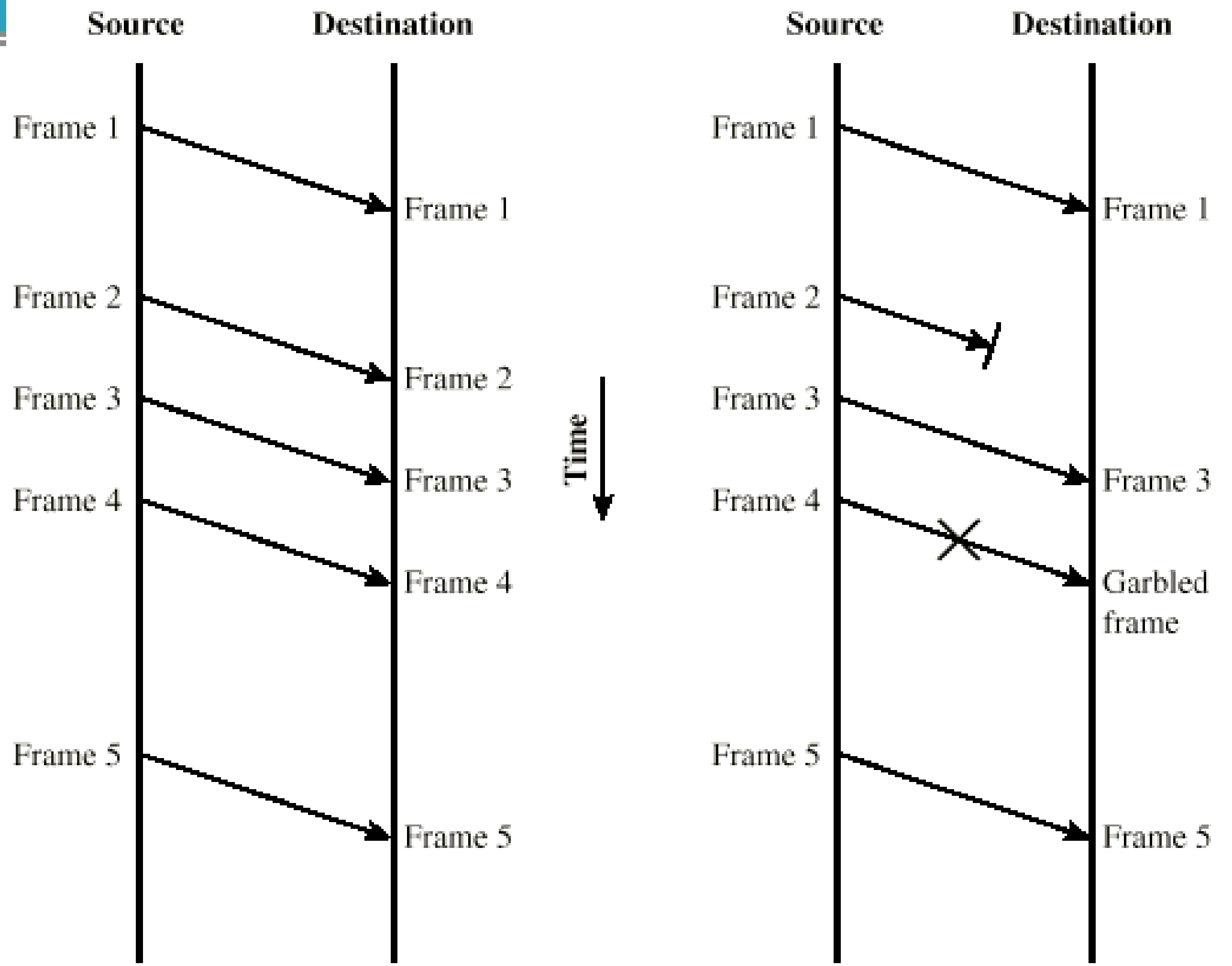
Backward Error Correction

Backward error correction

- Error detection at the receiver side
- The sender retransmits a frame until it received by the other side correctly.



Model of Frame Transmission



(a) Error-free transmission

(b) Transmission with losses and errors

Elementary Data Link Protocols

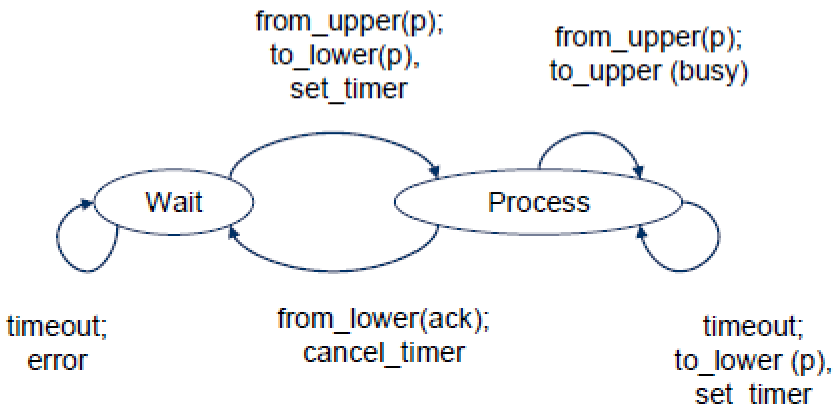
- Simplex Stop-and-Wait Protocol
- Alternate Bit Protocol
- Sliding Window Protocol

Simple Stop-and-Wait Protocol

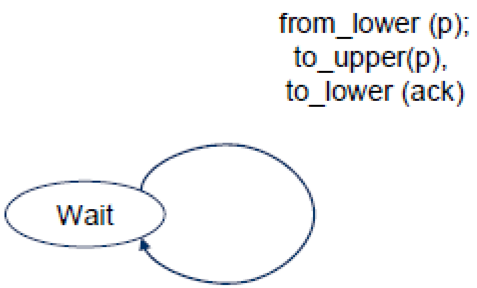
- A sends a message to B
- A stops and waits for an answer from B
  - Acknowledgement message (ACK)
- After receiving the message B sends an ACK back to the sender.
- A retransmits the message until it receives an ACK from B
- If the ACK arrived, the next message may be sent.

Simplex Stop-and-Wait Protocol

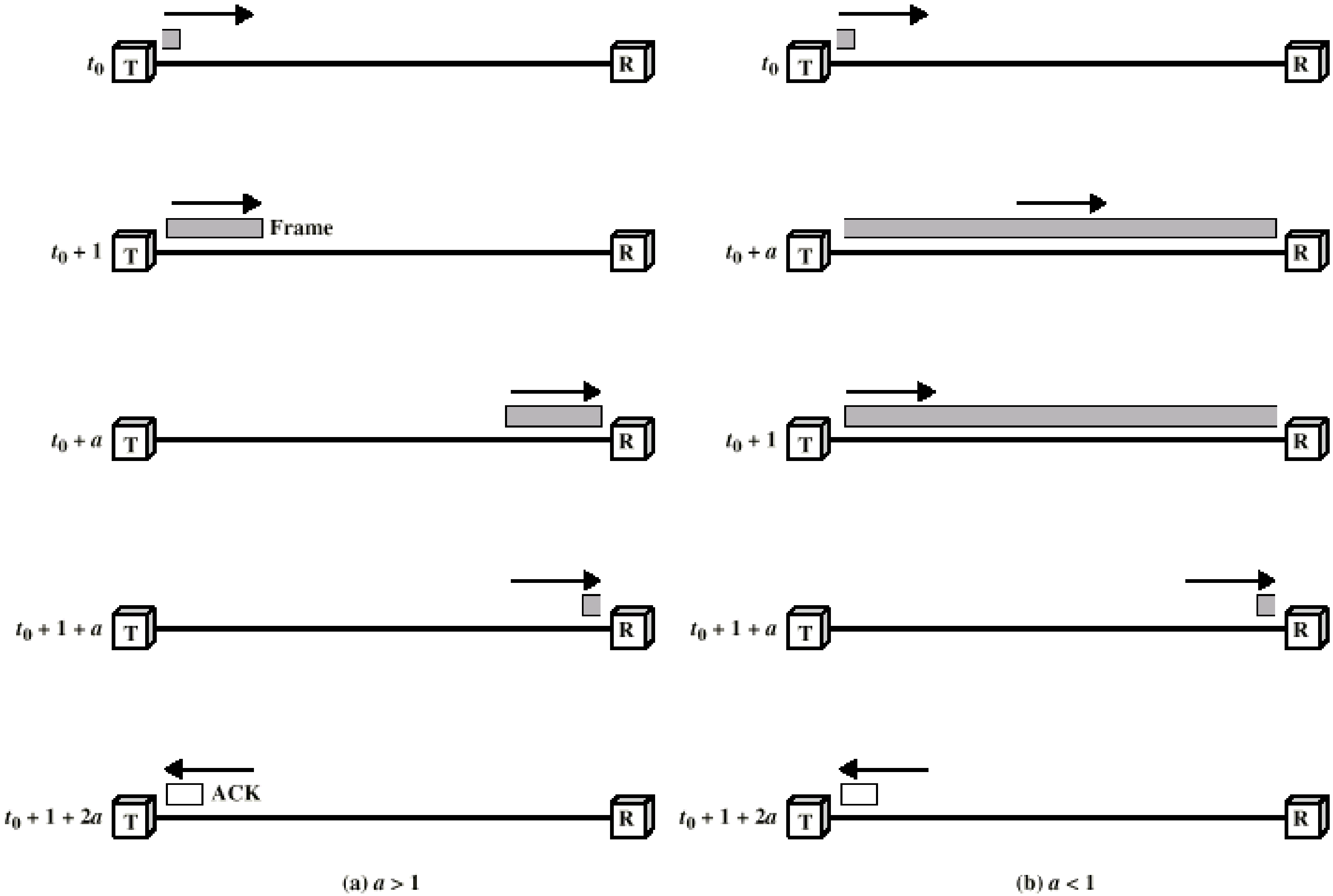
Sender



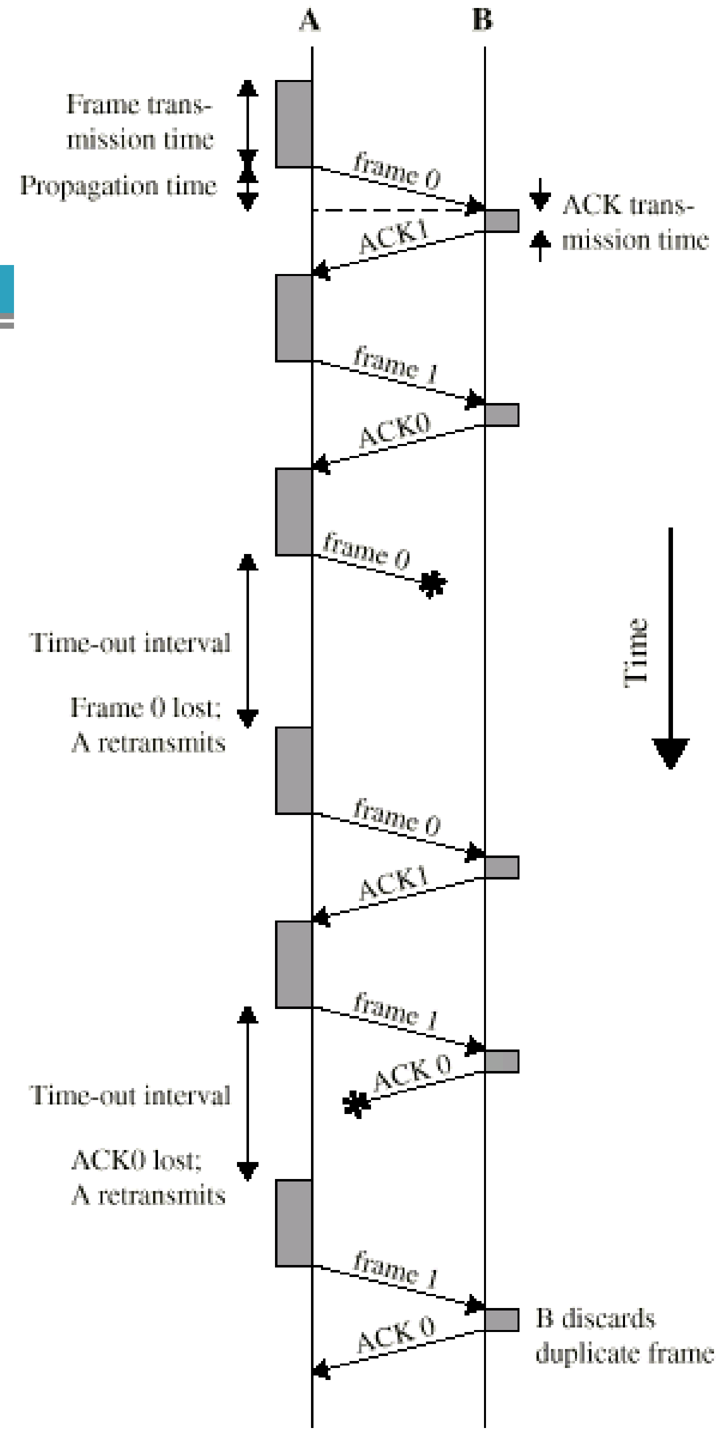
Receiver



Stop-and-Wait Link Utilization



Stop-and-Wait Diagram

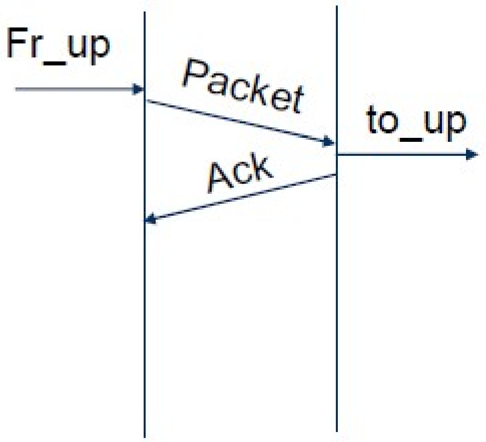


Simple, but inefficient for long distance and high speed applications.

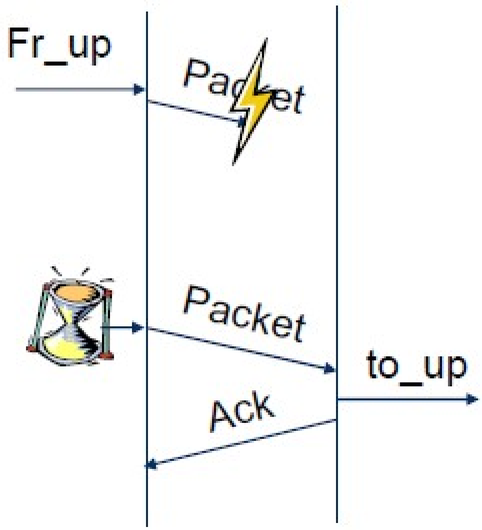
We can use sliding-window technique to improve the efficiency.

What’s the problem?

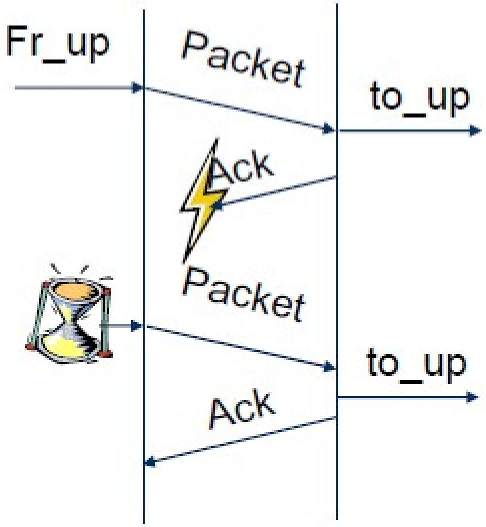
Usually



Packet loss



ACK loss

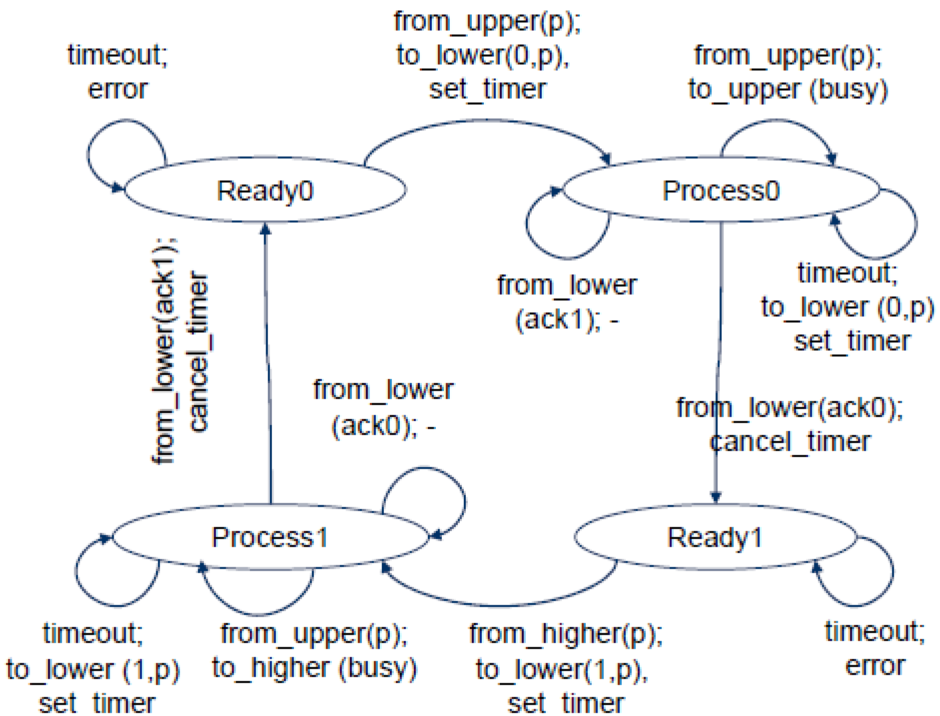


Alternating Bit Protocol (ABP)

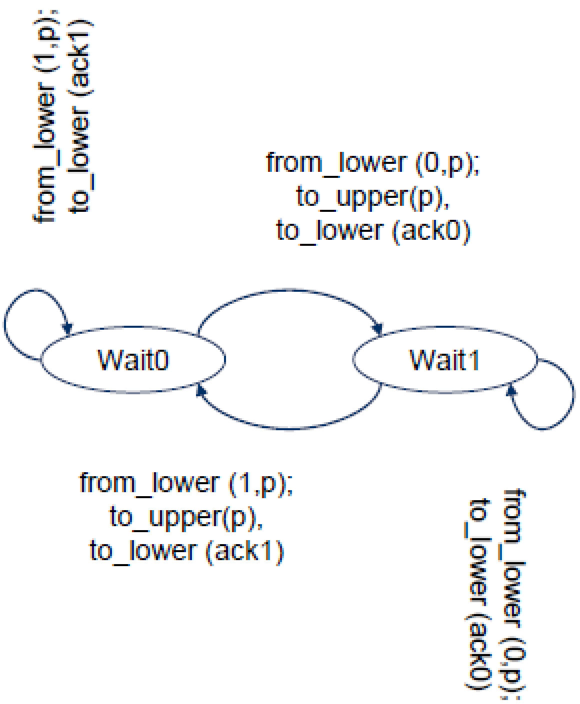
- **Let**
  - A be the sender
  - B be the receiver
- **A and B maintain internal one-bit counter**
  - A value that is 0 or 1
- **Each message from A to B contains**
  - a data part and
  - a one-bit sequence number
    - E.g. a value that is 0 or 1
- After receiving A's message, **B sends an ACK back to A**
  - which also contains a one-bit sequence number
- **Retransmission until A receives an ACK from B with the same sequence number**
  - Then A complements its sequence number
    - 0->1 or
    - 1->0

Alternating Bit Protocol (ABP)

Sender



Receiver

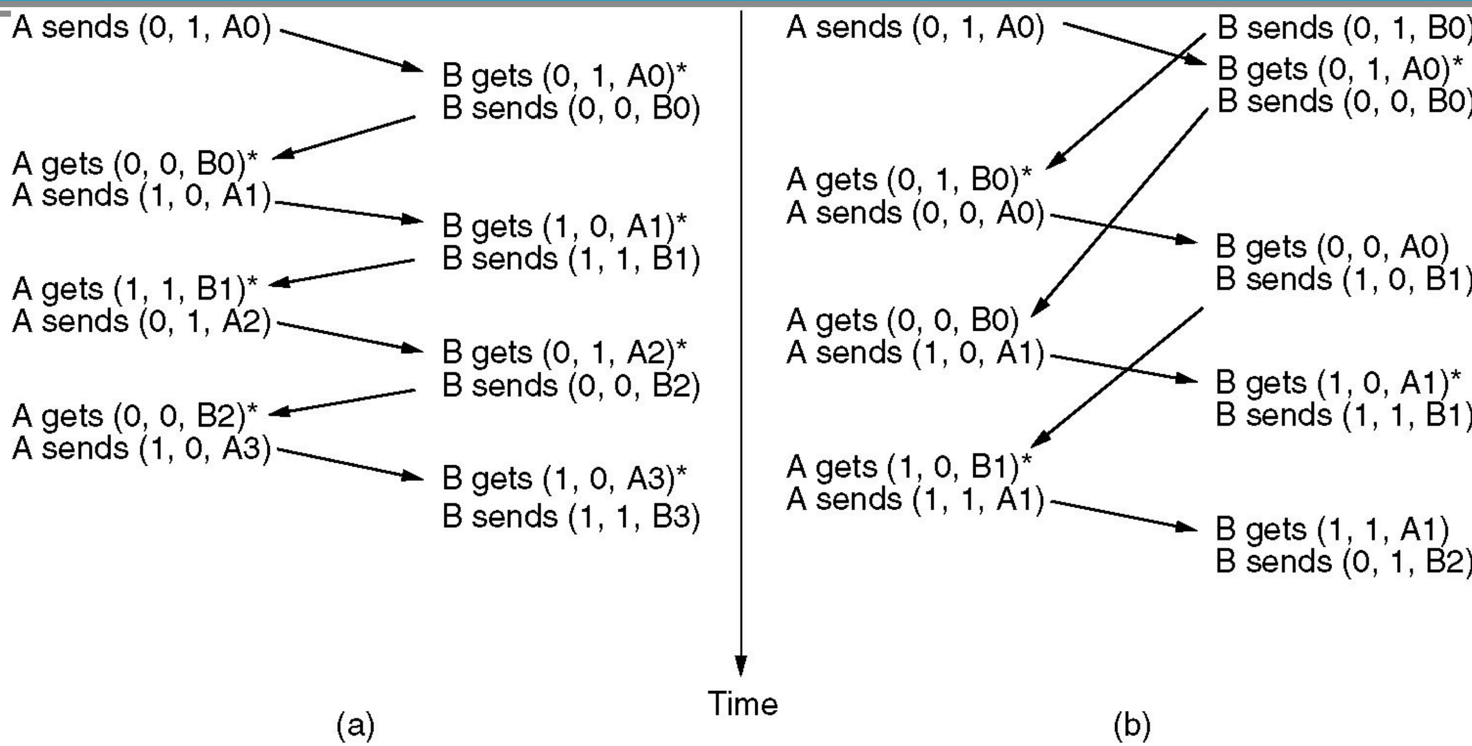


Alternating Bit Protocol (ABP)

- A reliable data transport over a noisy channel
- Basic flow control
  - The sender has to wait for the ACK from the receiver before sending the next message
- Automatic Repeat reQest (ARQ) protocol

- An acknowledgement
  - marks that the new message has been delivered.
  - allows the sender to tranmit the next frame.

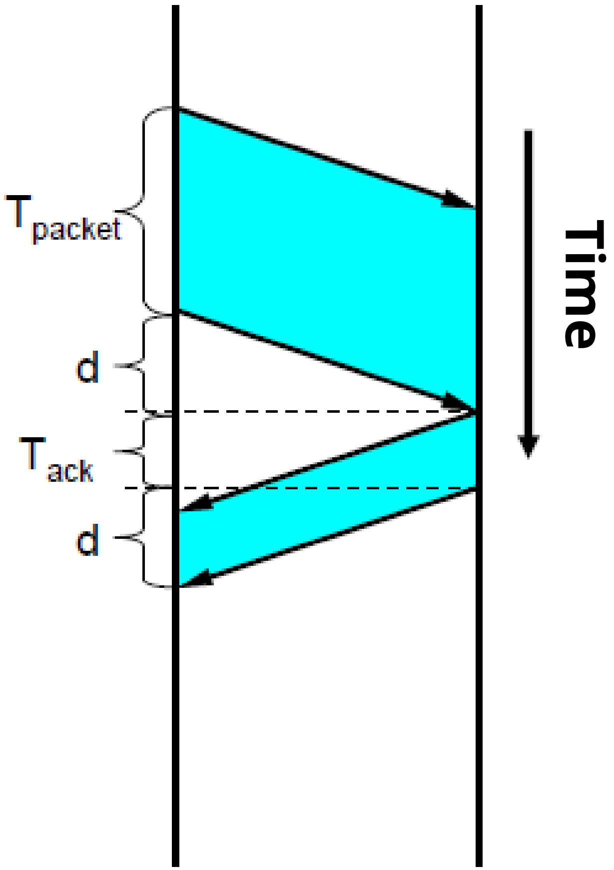
Alternating Bit Protocol



Two scenarios for ABP. (a) Normal case. (b) Abnormal case. The notation is (seq, ack, packet number). An asterisk indicates where a network layer accepts a packet.

ABP – Channel utilization

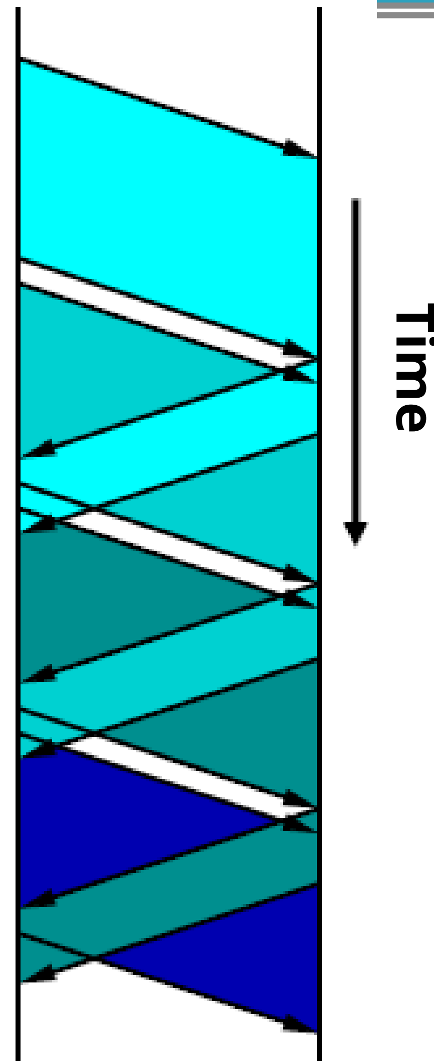
- Utilization (η) is the ratio of
  - The time needed for the transmission of a frame (T_{packet})
  - The time ellapsed until the next frame can be transmitted
    - In the fig.: (T_{packet} + d + T_{ack} + d)
- Now:
  - $\eta = T_{packet} / (T_{packet} + d + T_{ack} + d)$
- If the propagation delay is large, the ABP is not efficient.



How to improve the efficency?

- The sender transmit frames continously one after another
  - More frames are sent out, but not acknowledged.
  - Pipeline technique

- Introduce sequence numbers



Sliding Window Protocols

- Similar to ABP
- but allow multiple frames to transmit
  - Receiver has a buffer of W frames
  - Sender can send up to W frames without receiving ACK
- Bidirectional
- Each outgoing frame contains a seq. number from 0 to 2^n-1.
  - So it fits in an n-bit field
  - ABP uses n=1
- Each ACK carries the sequence number of the next expected frame by the receiver

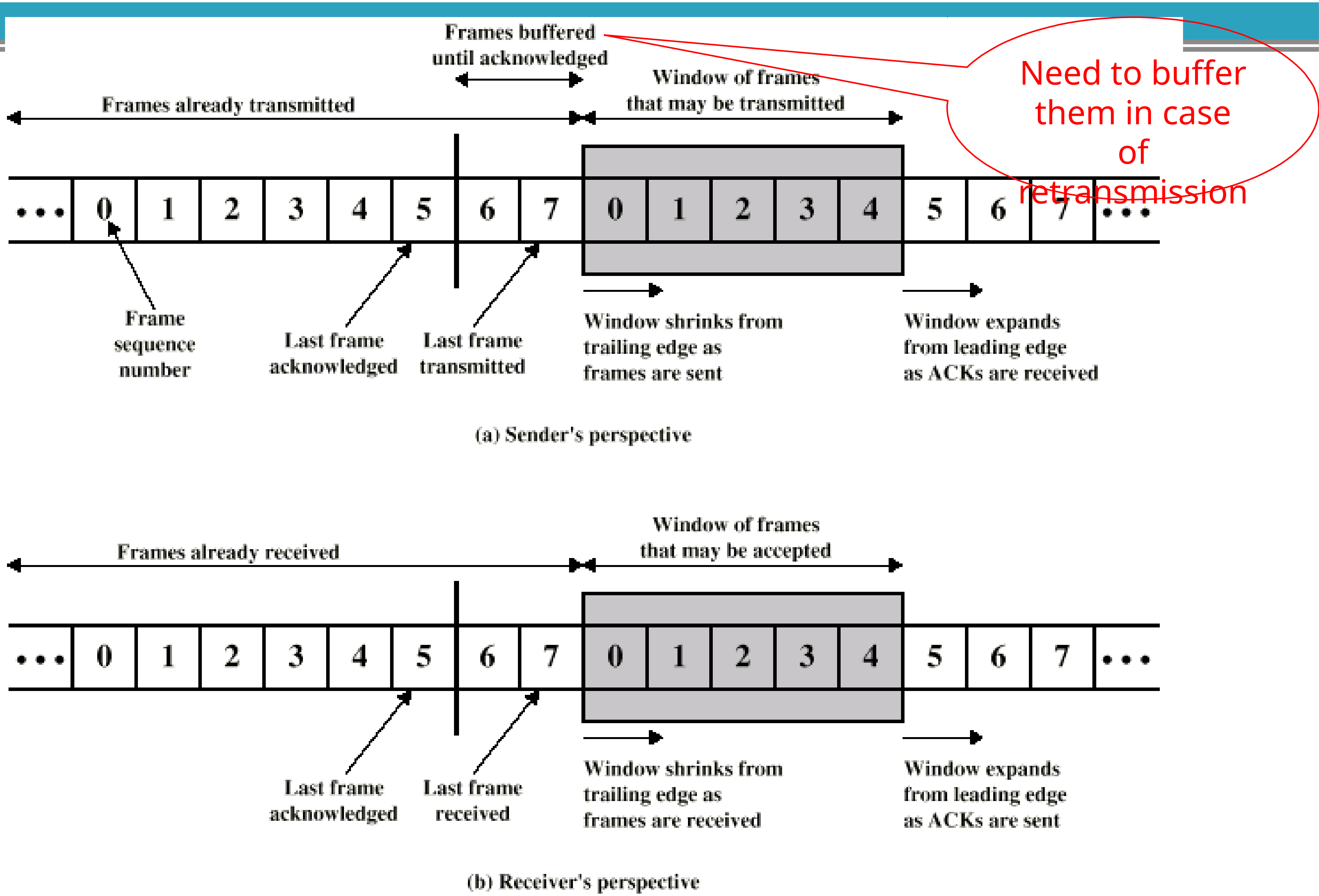
Sliding Window Protocols

- **At the sender**
  - Sending window
    - a set of sequence numbers corresponding to frames being under transmission (finite range of numbers)
- **At the receiver**
  - Receiving window
    - Sequence numbers for frames it is permitted to accept (finite range of numbers)

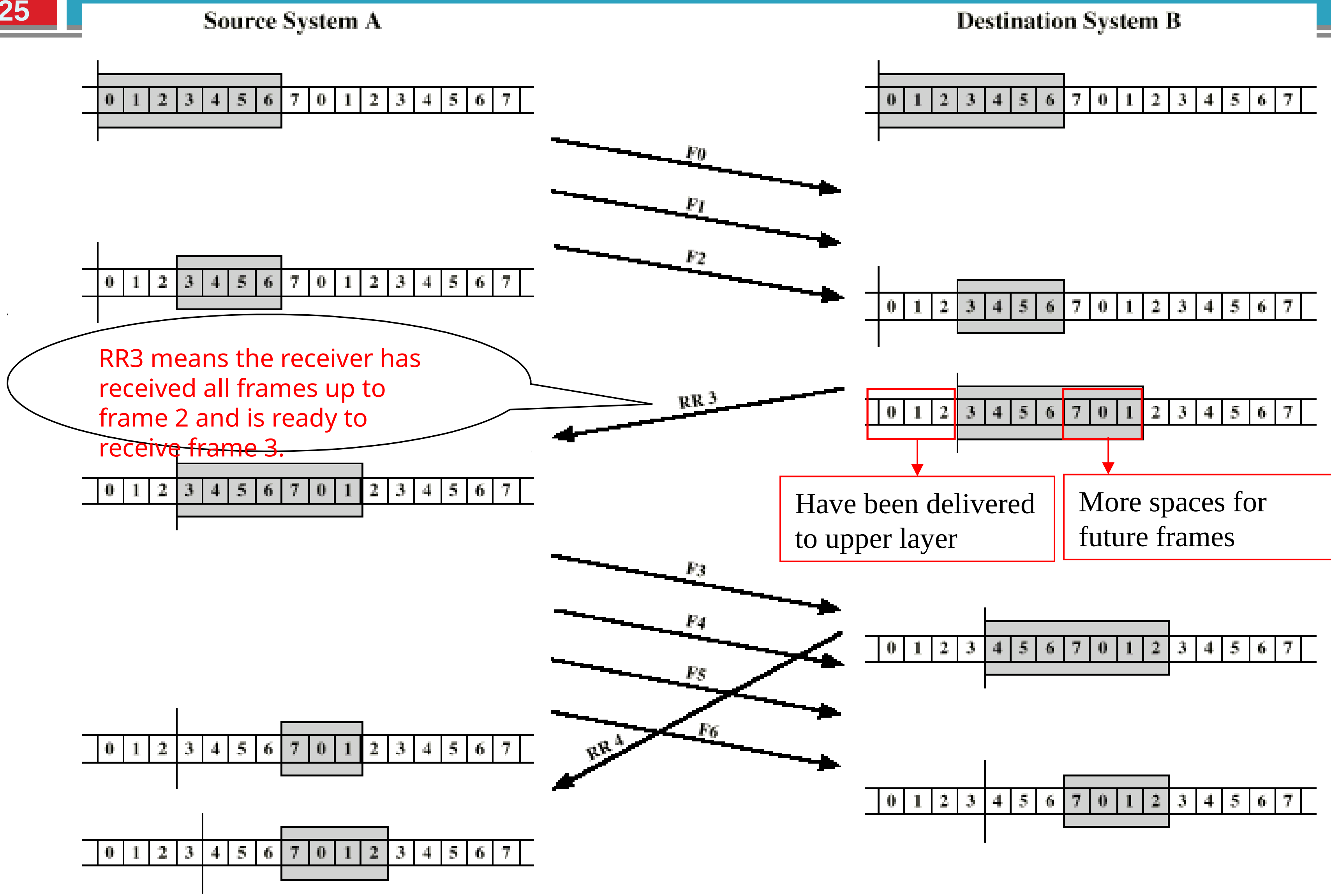
- The sender's and receiver's windows need
  - not have tha same lower and upper bounds and
  - even have the same size.

- The window size can be
  - fixed or
  - grow or shrink over the course of time as frames are sent and received

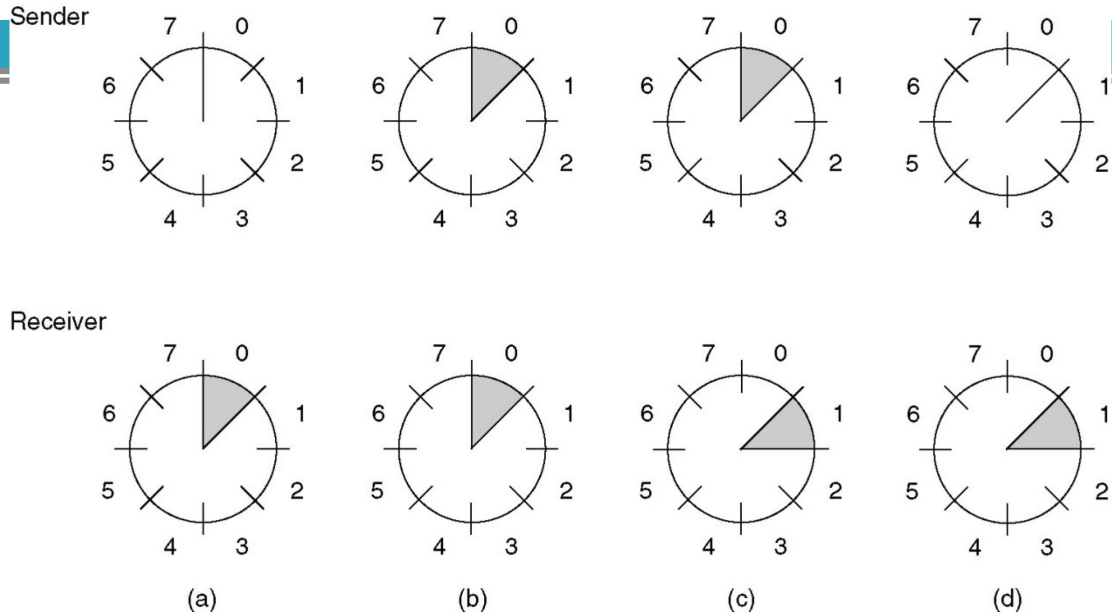
Sliding-Window Diagram



Example Sliding-Window



Sliding Window Protocols



A sliding window of size 1, with a 3-bit sequence number.

(a) Initially.

(b) After the first frame has been sent.

(c) After the first frame has been received.

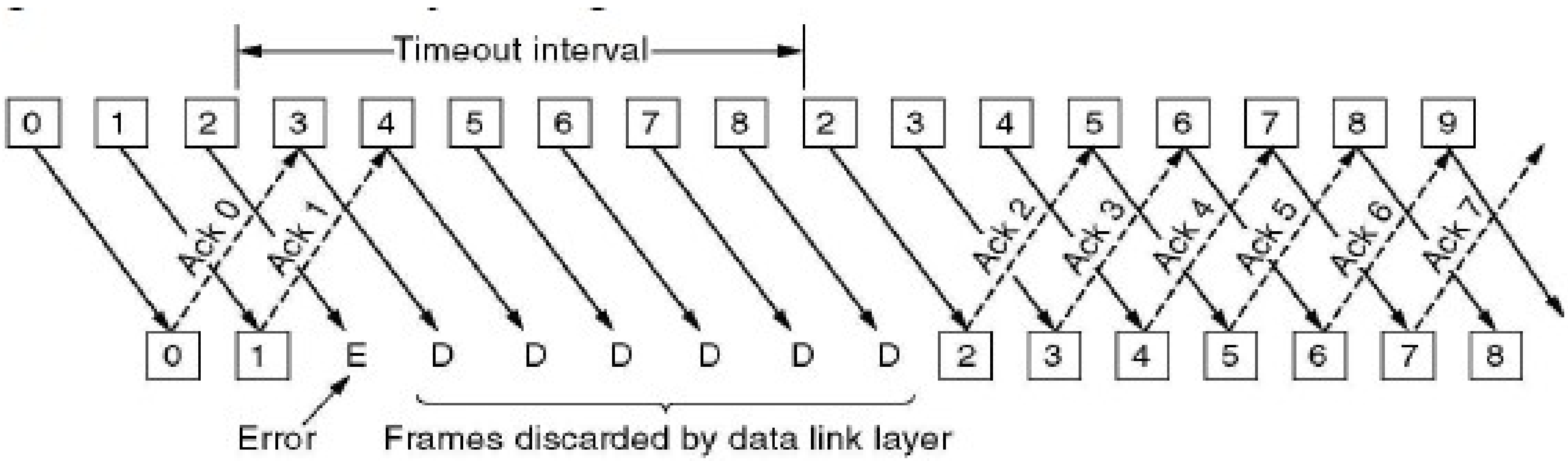
(d) After the first acknowledgement has been received.

Go-Back-N

- A sliding window protocol where
  - the receiver's window size is fixed to 1,
  - while the sender has window size > 0.

- After receiving a damaged frame
  - Receiver discards all subsequent frames
  - Sender retransmits the damaged frame and all its successors after the times out

Go-Back-N

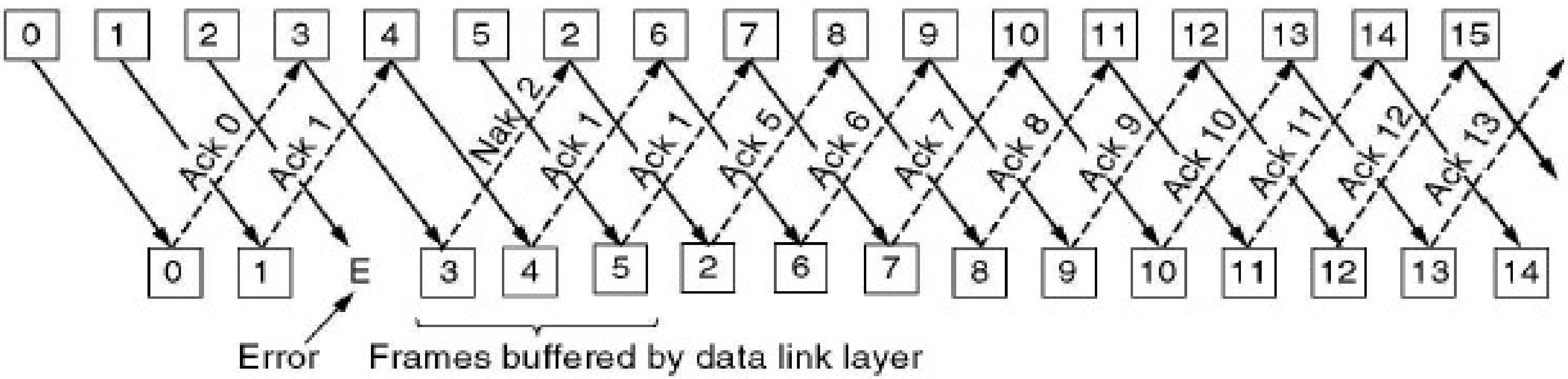


Timeout interval
0
1
2
3
4
5
6
7
8
2
3
4
5
6
7
8
9
Ack 0
Ack 1
Ack 2
Ack 3
Ack 4
Ack 5
Ack 6
Ack 7
0
1
E
D
D
D
D
D
D
2
3
4
5
6
7
8
Error
Frames discarded by data link layer

Selective Repeat

- Receiver’s window size is n (n >1)
  - At most n frames can be buffered
- Receiver stores all the correct frames following the bad one
- The sender retransmits only the bad frame not all its successors

Selective Repeat



Communication channels and piggybacking

- Simplex
  - Communication in one direction only
- Half-duplex
  - Communication in both directions, but only one direction at a time, not simultaneously.
- Full-duplex
  - Communication in both directions simultaneously

- The previous protocols assumed
  - a simplex channel to the upper (network) layer and
  - a (half-)duplex channel to the physical layer

- If we use duplex channel to the upper layers
  - Transmitting data packet and acknowledgements in both directions separately
  - Or using piggybacking
    - The header of a data packet sent in the opposite direction carries the acknowledgement back to the other side
    - widely applied in practice

Ethernet frame

802.3 Ethernet frame structure

<table>
<tr><th>Preamble</th><th>Start of frame delimiter</th><th>MAC destination</th><th>MAC source</th><th>802.1Q tag (optional)</th><th>Ethertype (Ethernet II) or length (IEEE 802.3)</th><th>Payload</th><th>Frame check sequence (32-bit CRC)</th><th>Interframe gap</th></tr>
<tr><td>7 octets</td><td>1 octet</td><td>6 octets</td><td>6 octets</td><td>(4 octets)</td><td>2 octets</td><td>42[note 2]–1500 octets</td><td>4 octets</td><td>12 octets</td></tr>
<tr><td colspan="2"></td><td colspan="6">64–1522 octets</td><td></td></tr>
<tr><td colspan="8">72–1530 octets</td><td></td></tr>
<tr><td colspan="9">84–1542 octets</td></tr>
</table>

Outline

- Framing
- Error Checking and Reliability
- Media Access Control
  - 802.3 Ethernet
  - 802.11 Wifi

What is Media Access?

- Ethernet and Wifi are both multi-access technologies
  - Broadcast medium, shared by many hosts
  - Simultaneous transmissions cause collisions
    - This destroys the data
- Media Access Control (MAC) protocols are required
  - Rules on how to share the medium
  - Strategies for detecting, avoiding, and recovering from collisions

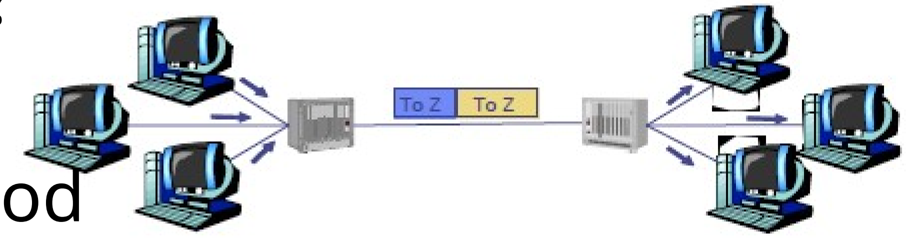
Strategies for Media Access

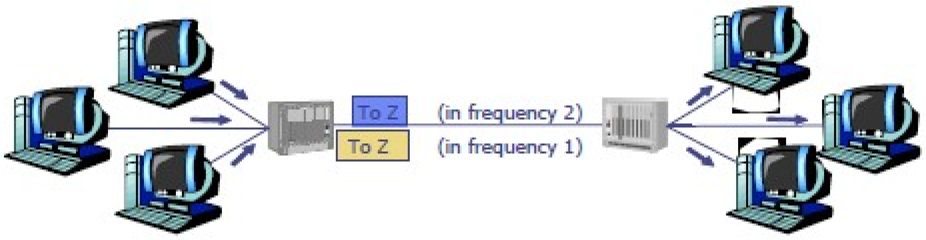
- Channel partitioning
  - Divide the resource into small pieces
  - Allocate each piece to one host
  - Example: Time Division Multi-Access (TDMA) cellular
  - Example: Frequency Division Multi-Access (FDMA) cellular
- Taking turns
  - Tightly coordinate shared access to avoid collisions
  - Example: Token ring networks

Channel Partitioning

- Frequency Division Multiplexing
  - E.g. a telephone trunk
- Time Division Multiplexing

- Are they good solutions?
  - if data rates are fixed
  - if the link utilization is good





What’s the problem?

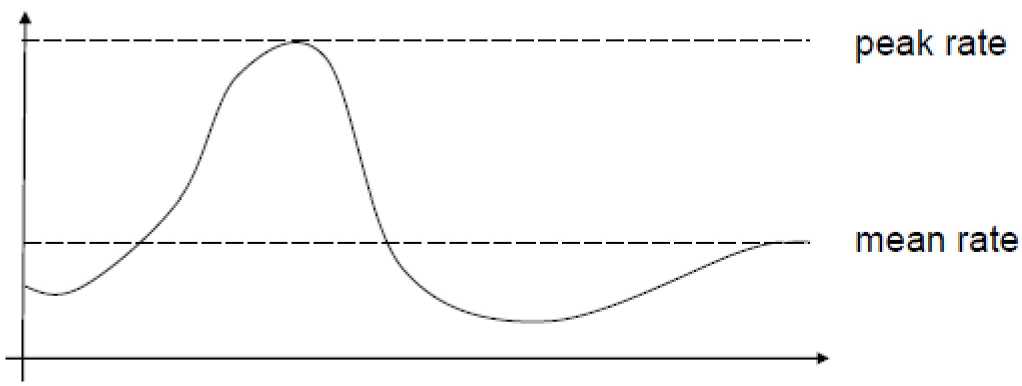
The number of senders is large

continuously varying

bursty traffic

Bursty traffic

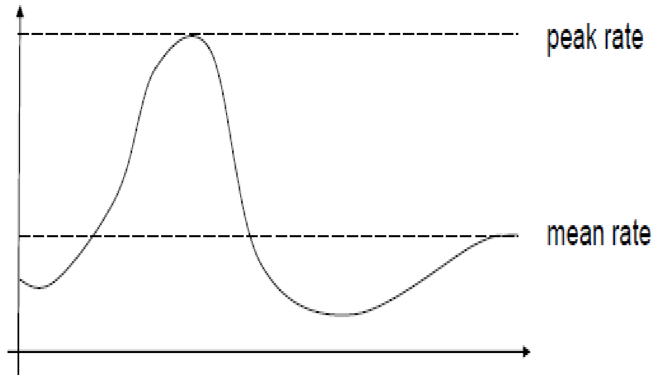
- If there's **huge difference** between the **peak** rate and the **mean (**or **average)** rate
- In computer networks it is not rare
  - peak rate/mean rate = 1000/1

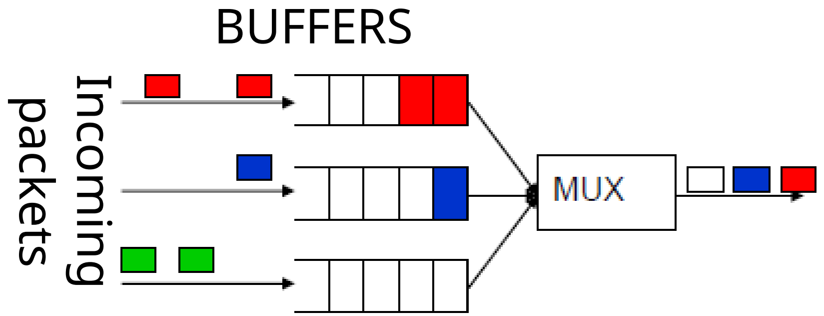


Bursty traffic
with Static Channel Allocation

The capacity of the channels must be

- Either quite large to handle peak rates
  - Waste of resources
  - The mean rate is much less than the peak one
- Or based on the mean rate
  - We need buffers/queues to store packets coming faster than the mean rate
  - Queuing delays

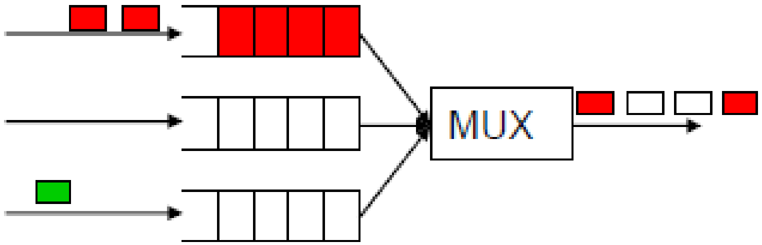




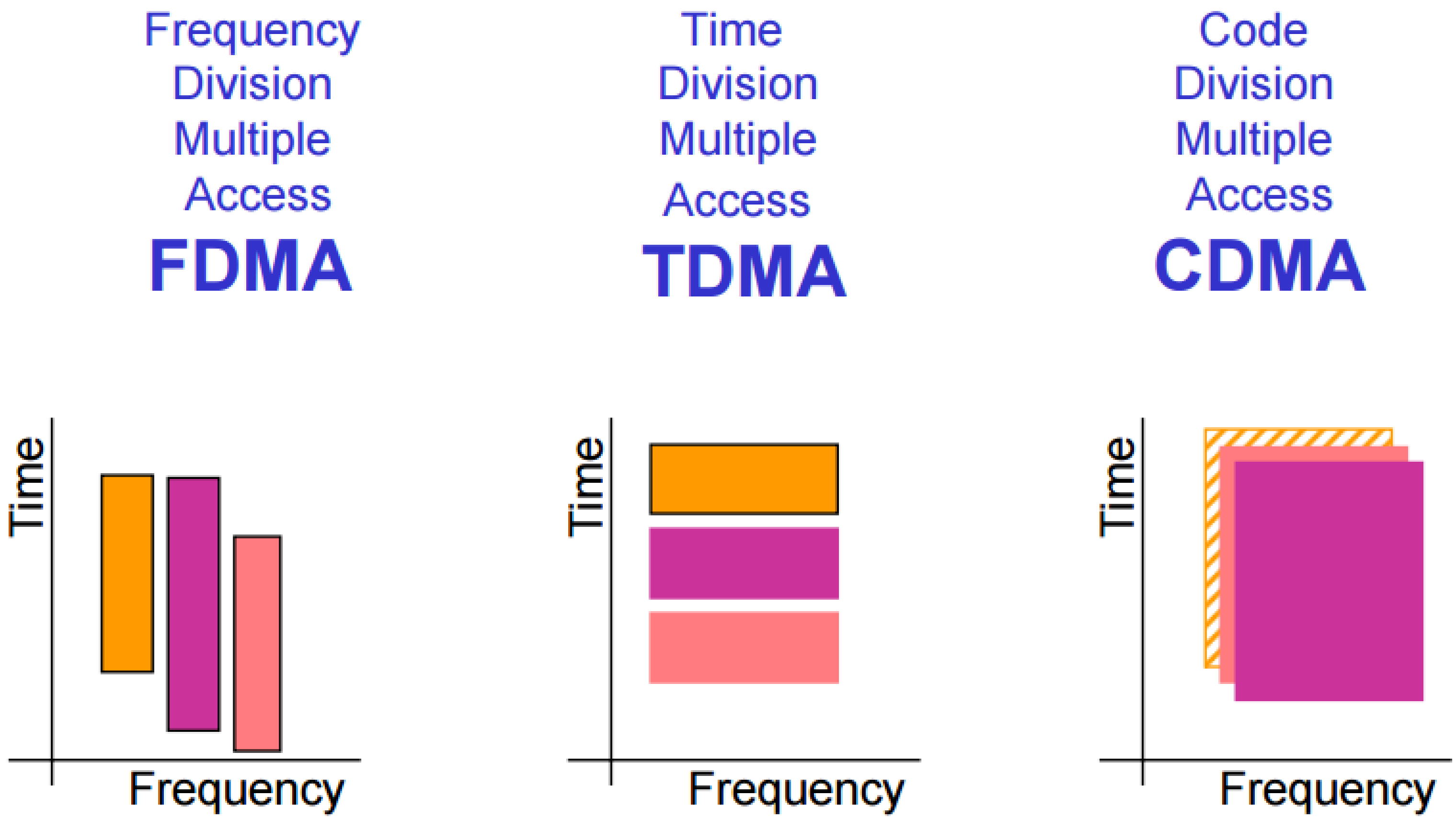
What's about delays?

- **If there's no multiplexing**
  - A source has data rate of p bps
  - The capacity of the link is C bps
  - The delay is T
- **Bursty traffic with static channel allocation**
  - Dividing the channel into N static subchannels
    - With sending rate p/N bps and capacity C/N
  - The delay is N T

Static channel allocation increases the packet delays
- Because of the idle subchannels



CDMA – Code Division Multiple Access



CDMA Analogy

- 10 people in a room.
  - 5 speak English, 2 speak Spanish, 2 speak Chinese, and 1 speaks Russian.
- Everyone is talking at relatively the same time over the same medium – the air.
- Who can listen to whom and why?
- Who can't you understand?
- Who can't speak to anyone else?

CDMA – Code Division Multiple Access

- Used by 3G and 4G cellular networks
- Each station can broadcast at any time in the full frequency spectrum
- The signals may interfere
  - Resulting in a linear combination of individual signals

- Algorithm
  - We assign a vector of length m to each station: v
    - Pairwise orthogonal vectors!!!
  - Each bit is encoded
    by the chip vector of the sender or it’s complement: v or -v
  - If it sends bit 1, it transmits v
  - If it sends bit 0, it transmits -v

- Result is a sequence of vectors of length m

CDMA – Code Division Multiple Access

- Interference
  - A sends a,-a,a,a
  - B sends b,b,-b,-b
  - After interference we receive: a+b,-a+b,a-b,a-b ???

- How to decode?

CDMA – Code Division Multiple Access

- Interference
  - A sends a,-a,a,a
  - B sends b,b,-b,-b
  - After interference we receive: a+b,-a+b,a-b,a-b ???

- Decoding the message of A
  - Take the dot product by the sender’s chip code
    - (a+b)a > 0 => 1
    - (-a+b)a < 0 => 0
    - (a-b)a >0 => 1
    - (a-b)a > 0 => 1

If the dot product is

<0: bit 0 was sent by A

>0: bit 1 was sent by A

=0: nothing was sent by A

the channel is not used by A