

Data Link Layer-4

11.11.2019

BLM 305 I Veri İletişimi (*Data Communication*)

Tech. Assist. Kübra ADALI
Assoc. Prof. Dr. Veli Hakkoymaz

References:

- *Computer Networks*, Andrew Tanenbaum, Pearson, 5th Edition, 2010.
- *Computer Networking, A Top-Down Approach Featuring the Internet*, James F.Kurose, Keith W.Ross, Pearson-Addison Wesley, 6th Edition, 2012.
- **BLG 337 Slides** from İTÜ prepared by Assoc. Prof.Dr. Berk CANBERK

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collision Detection)

✓ **General Information**

- The main principles are same with the prev. CDMA.
- Collusions are detected in a short period of time.
- Colliding transmissions aborted, reducing channel wastage
- **Collision detection:**
 - easy in wired LANs: measure signal strengths, compare transmitted, received signals
 - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collision Detection)

✓ **Ethernet CSMA/CD Algorithm**

- NIC takes datagram from network layer, creates frame.
- If NIC understands the channel idle, starts frame transmission.
- If NIC senses channel busy, waits until channel idle, then transmits.
- If NIC transmits entire frame without detecting another transmission, NIC has been successful with the frame.
- If NIC detects another transmission while transmitting, aborts and sends jam signal.

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collision Detection)

✓ **Ethernet CSMA/CD Algorithm**

- **After aborting, NIC enters binary (exponential) backoff:**
 - after m th collision, NIC chooses K at random from $\{0, 1, 2, \dots, 2^m - 1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
 - longer backoff interval with more collisions

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collusion Detection)

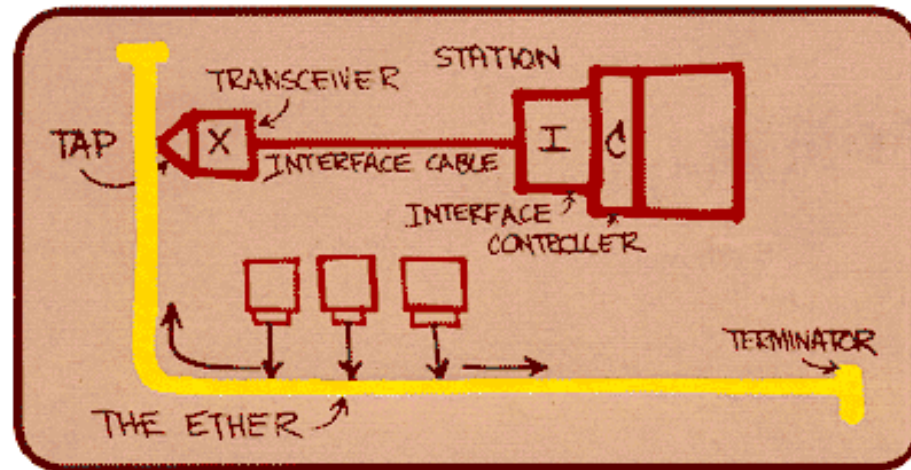
✓ **Ethernet General Information**

- ✓ Developed by Bob Metcalfe and others at Xerox PARC in mid-1970s
 - Roots in Aloha packet-radio network
 - Standardized by Xerox, DEC, and Intel in 1978
 - LAN standards define MAC and physical layer connectivity
 - IEEE 802.3 (CSMA/CD - Ethernet) standard – originally 2Mbps
 - IEEE 802.3u standard for 100Mbps Ethernet
 - IEEE 802.3z standard for 1,000Mbps Ethernet

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collusion Detection)

✓ Ethernet General Information

- CSMA/CD: Ethernet's Media Access Control (MAC) policy (1-persistent CSMA/CD with binary exponential backoff)
- Bandwidths: 10Mbps, 100Mbps, 1Gbps
- Max bus length: 2500m: 500m segments with 4 repeaters
- Bus and Star topologies are used to connect hosts
- Manchester Encoding



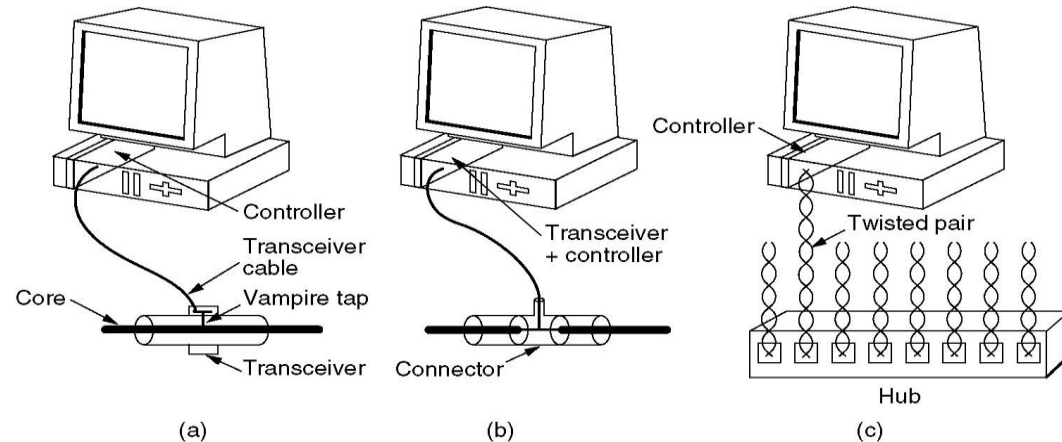
RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collusion Detection)

✓ Ethernet Cabling

- Unshielded twisted pair (UTP) medium
- Star-shaped topology
 - Stations connected to central point (hub), (multiport repeater)
 - Two twisted pairs (transmit and receive)
 - Repeater accepts input on any one line and repeats it on all other lines
- Link limited to 100 m on UTP
- Multiple levels of hubs can be cascaded

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

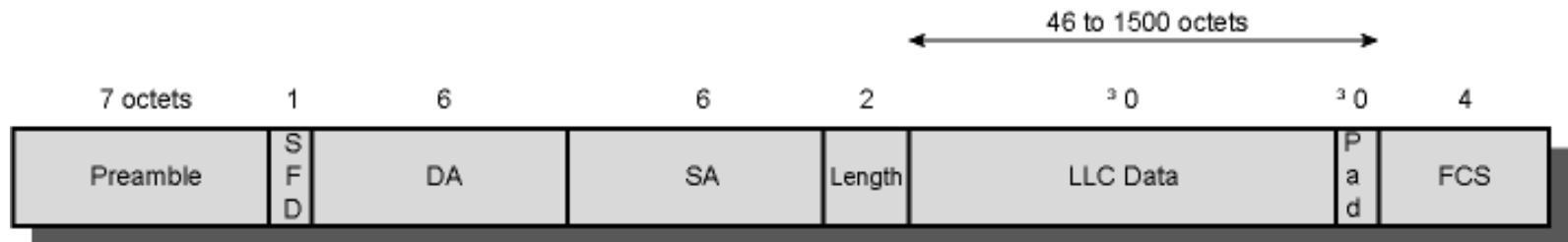
(a) 10Base5, (b) 10Base2, (c) 10Base-T.



RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collusion Detection)

✓ 802.3 Ethernet Frame Format

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**
 - **Preamble (+SFD)**: 7 bytes with pattern 10101010 followed by 1 byte with pattern 10101011 used to synchronize receiver
 - **Start of Frame Delimiter (SFD)**: indicates start of frame (1 byte with pattern 10101011)



SFD = Start of frame delimiter
DA = Destination address
SA = Source address
FCS = Frame check sequence

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collusion Detection)

✓ 802.3 Ethernet Frame Format

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**
 - **Addresses:** 6 bytes, frame is received by all adapters on a LAN and dropped if address does not match, globally unique address assigned by manufacturer, e.g. 8:0:e4:b1:2
 - **Length:** frame size
 - **Pad:** Zeroes used to ensure **minimum frame length of 64 Bytes (WHY??)**
 - **FCS (CRC) Cyclic Redundancy Check:** check sequence to detect bit errors, if error is detected, the frame is simply dropped
 - Body can contain up to 1500 bytes of data

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collision Detection)

- ✓ CSMA an improvement over ALOHA because no station transmits when it senses the channel busy
- ✓ Another improvement: **stations abort their transmissions as soon as they detect a collision.**
 - if two stations sense the channel idle and begin transmission simultaneously they will both detect the collision immediately
 - Rather than finishing transmitting their frames, which will be corrupted, stop transmitting frames as soon as collision detecte
 - **Quickly terminating damaged frames saves time and bandwidth!**
- ✓ This protocol is called CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collusion Detection)

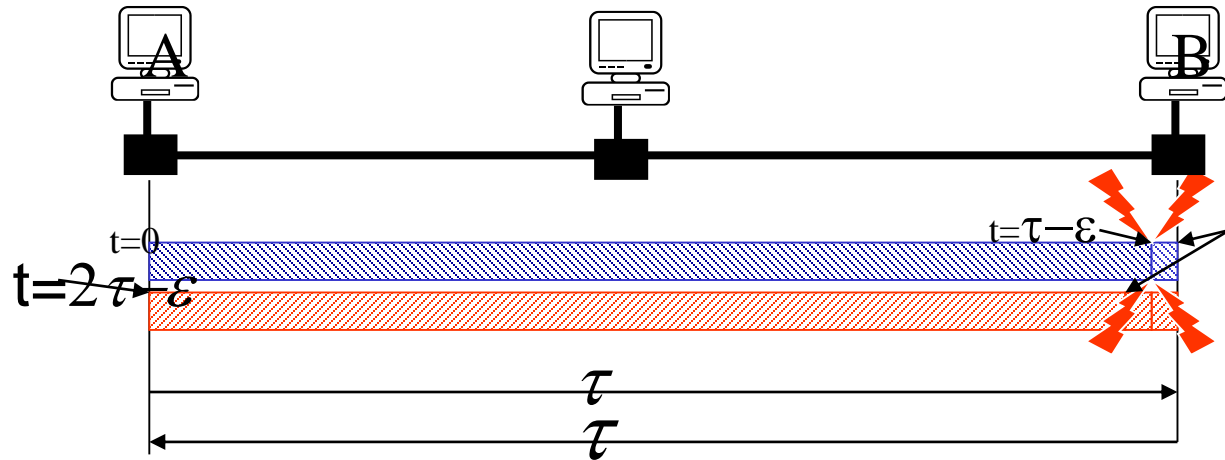
✓ **General Information**

- The main principles are same with the prev. CDMA.
- Collusions are detected in a short period of time.
- Colliding transmissions aborted, reducing channel wastage.
- Persistent and non-persistent retransmission.
- **Collision detection:**
 - easy in wired LANs: measure signal strengths, compare transmitted, received signals
 - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collision Detection)

✓ Time to Detect Collision

- Let the time for a signal to propagate between two farthest stations be t
- It takes $2t$ seconds for two stations to realize that there has been a collision after starting the transmission



Events:

- $t=0$: Host A starts transmitting a packet.
- $t=\tau-\epsilon$: Just before the first bit reaches Host B, Host B senses the line to be idle and starts to transmit a packet. A collision takes place near Host B.

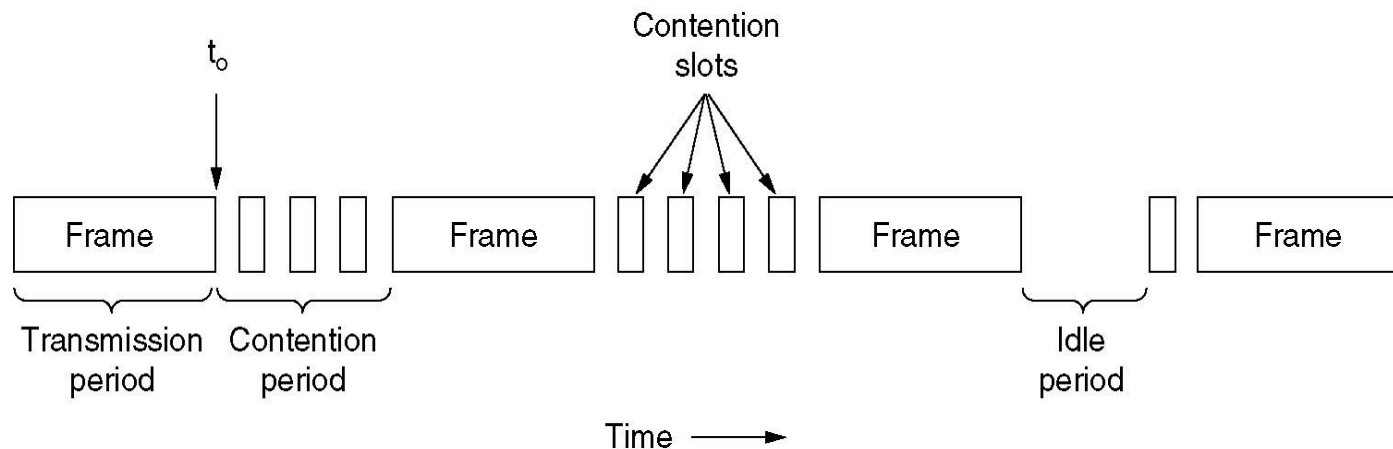
$t=2\tau-\epsilon$: Host A receives the noise burst caused by the collision

RAMP: Carrier Sense Multiple Access (CSMA/CD) Protocols-2 (Collision Detection)

✓ Time to Detect Collision

- CSMA/CD can be in one of three states: contention, transmission, or idle.
- The minimum time it takes to detect a collision is just the time it takes for the signal to propagate from any computer to any other computer and back again → **$2t$ Slot Time**
 - E.g., for a 1km long cable: $t=(1000 \text{ m})/ (2 \times 10^8 \text{ m/sec})=5 \text{ msec}$

Contention Period: modeled as a Slotted ALOHA System with Slot Time of $2t$



“Taking Turns” MAC Protocols

We have three channel partitioning MAC protocols:

- ✓ **channel partitioning**

- At high load, shares channel efficiently and fairly.
- In low load, delay in the channel access, unused bandwidth allocated when there is a waiting node.

- ✓ **random access**

- Efficient in low load: a node can fully use the channel.
- In high load: much more collision occurs.

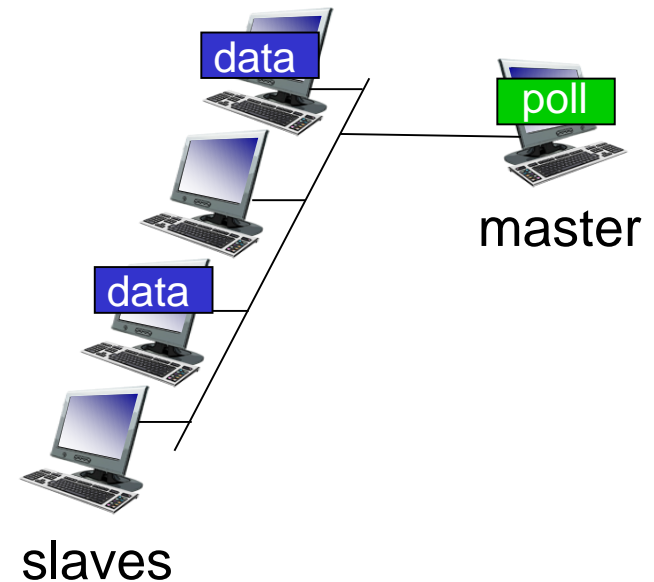
- ✓ **taking turns**

- Tries to do best combination of two past perspectives.

“Taking Turns” MAC Protocols

✓ Polling

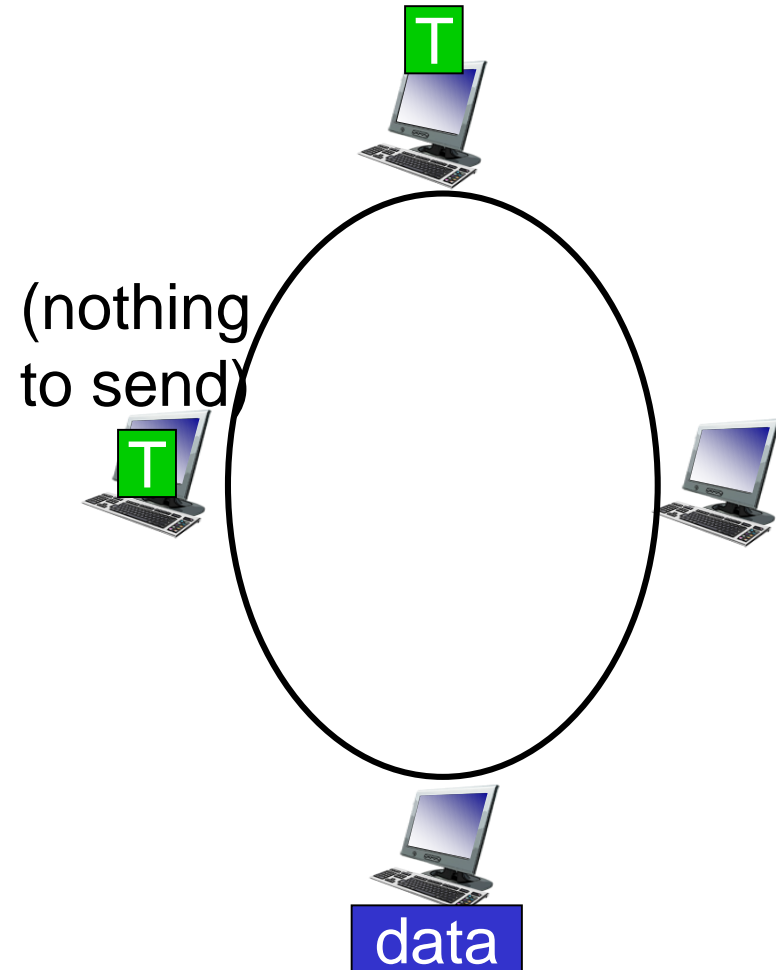
- master node “invites” slave nodes to transmit in turn
- typically used with “dummy” slave devices
- Disadvantages:
 - The limited number of polling
 - latency
 - single point of failure (master)



“Taking Turns” MAC Protocols

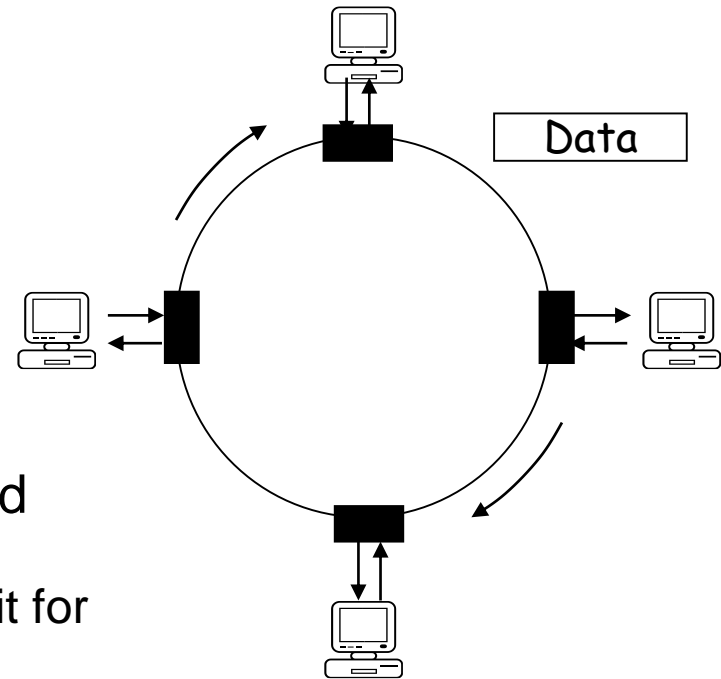
✓ Token Passing

- Check token given from one node to another sequentially.
- Token message
- Disadvantages:
 - The limited number of token
 - latency
 - single point of failure (token)



Summary of Token Ring

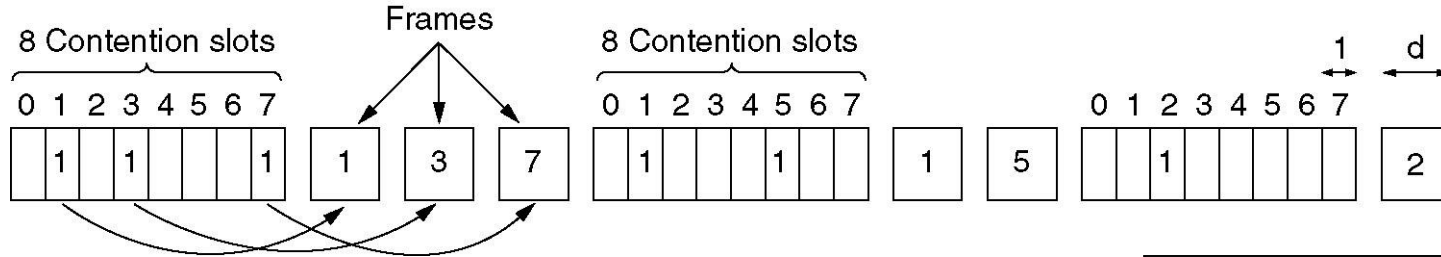
- ✓ Planned to take the advantages of predictability, fairness and reliability
 - Bandwidth scope: 4Mbps and 16Mbps
- ✓ Developed by IBM in early 80's as a new LAN architecture
 - Has nodes connected into a ring
 - Special message called a token is passed around the ring
 - When nodes gets the token it can transmit for a limited time
 - Every node gets an equal opportunity to send
 - IEEE 802.5 standard for Token Ring
- ✓ Still used and sold but became old because of Ethernet



Collusion Free Protocols

- ✓ Collusion usually effects the system performance negatively.
- ✓ Collisions can still occur with CSMA/CD
- ✓ The negative effect gets bigger as the used cable gets longer
- ✓ Resolve contention without any collision!
 - Collusion-free protocols:
 - Bit-Map Protocol,
 - Binary Countdown Protocol etc.

Bit-Map Protocol



If a station is ready just after its bit slot has passed by, it must wait until the bitmap has come around again!!

Reservation protocol !

✓ Assumption:

- N Stations each with unique address from 0 to N-1
- Each contention period consists of exactly N slots.

✓ Procedures:

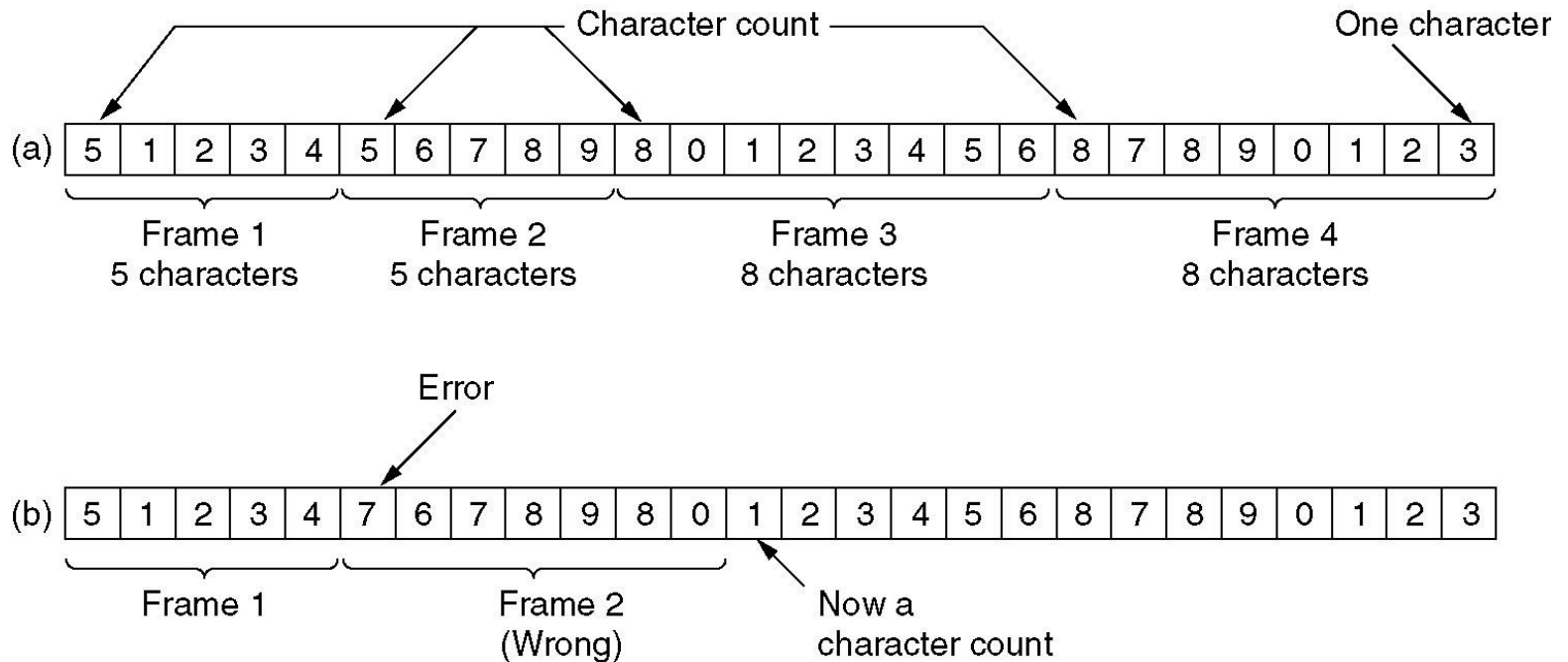
- If station 0 has a frame to send, it transmits a 1 bit during 0th slot
- Other stations are not allowed to transmit during this slot
- Station j may announce that it has a frame to send (only if so) by sending 1 bit in jth slot
- After all N slots have passed by, each station knows which station will transmit
- They begin sending in the numerical order as agreed before → **NO COLLISION!**

Framing

- ✓ DLL breaks bit stream into discrete frames
- ✓ Calculates checksum of each frame
- ✓ Start + end of frame determination:
 - Character count
 - Start/end characters with character stuffing
 - Start/end flags with bit stuffing
 - Physical layer coding violations

Framing

Problem occurs when control field is corrupted!



Character stream. (a) Without errors. (b) With one error.

Framing

✓ Bit Stuffing

- A frame starts and ends with a special bit pattern called a flag byte [01111110].
- Whenever sender data link layer detects five sequential 1 in the data stream, it automatically stuffs a 0 bit into the outgoing stream.
- When the receiver sees five sequential incoming 1 followed by a 0 bit, it automatically destuffs the 0 bit before sending the data to the network layer.


Framing

✓ Bit Stuffing : Example

- (a) The original data.
- (b) The data as they appear on the line.
- (c) The data as they are stored in receiver's memory after destuffing.

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0



Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Data Link Layer Functions

✓ Medium Access Control

- Less oftenly used on low bit error link (fiber, some twisted pair)
- wireless links: high error rates

✓ Framing

- encapsulate datagram into frame, adding header, trailer
- “MAC” addresses used in frame headers to identify source, destination
- channel access if shared medium

✓ Error Control(Detection & Correction):

○ Error Detection

- errors caused by signal attenuation, noise.
- Receiver does this mission (warns sender in order to retransmission or drops frame.)

○ Error Correction

- receiver finds *and corrects* bit error(s) without another retransmission

✓ Flow Control

- Adaptation of data rates between adjacent sending and receiving nodes

Error Control(Main Points)

- ✓ Applications require certain reliability level
 - Data applications require error-free transfer
 - Voice & video applications tolerate some errors
- ✓ Transmission errors exist
 - Single bit errors, Burst errors (which one is better???)
 - Lost frames vs. Damaged frames (when??)
- ✓ **Error detection**
 - Error-detecting codes: CRC, checksum etc.
 - Would suffice (along with a retransmission-based strategy) in relatively reliable channels
- ✓ **Error correction**
 - Error-correcting codes: Hamming codes...
 - Required for error-prone channels
- ✓ Two basic approaches:
 - Error **detection** & retransmission (ARQ: Automatic Repeat reQuest)
 - Forward error **correction** (FEC)

Error Control(Main Points)

✓ Error Rate

- **Bit error rate (BER)** : probability of a transmitted bit being received wrong
 - e.g., 10^{-7} for satellite, 10^{-9} for MW, 10^{-11} for fiber
- Packet/frame error rate.

✓ Formulation for a given BER and frame length n

$$\mathbf{P[\text{frame correct}] = (1-BER)^n}$$

$$\mathbf{P[\text{frame has error}] = 1 - (1-BER)^n \cong n \times BER}$$

Error Control : Error Detection

✓ To detect or correct errors

- Additional (redundant) bits added by transmitter to the original data to form codewords

- **Codeword length $n = m + r$**

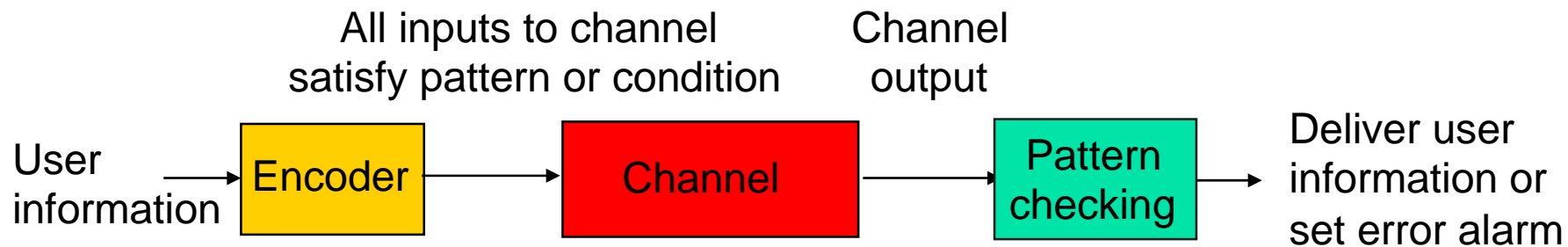


- For example: Parity.
 - Value of parity bit is such that character has even (even parity) or odd (odd parity) number of ones

Error Control : Error Detection

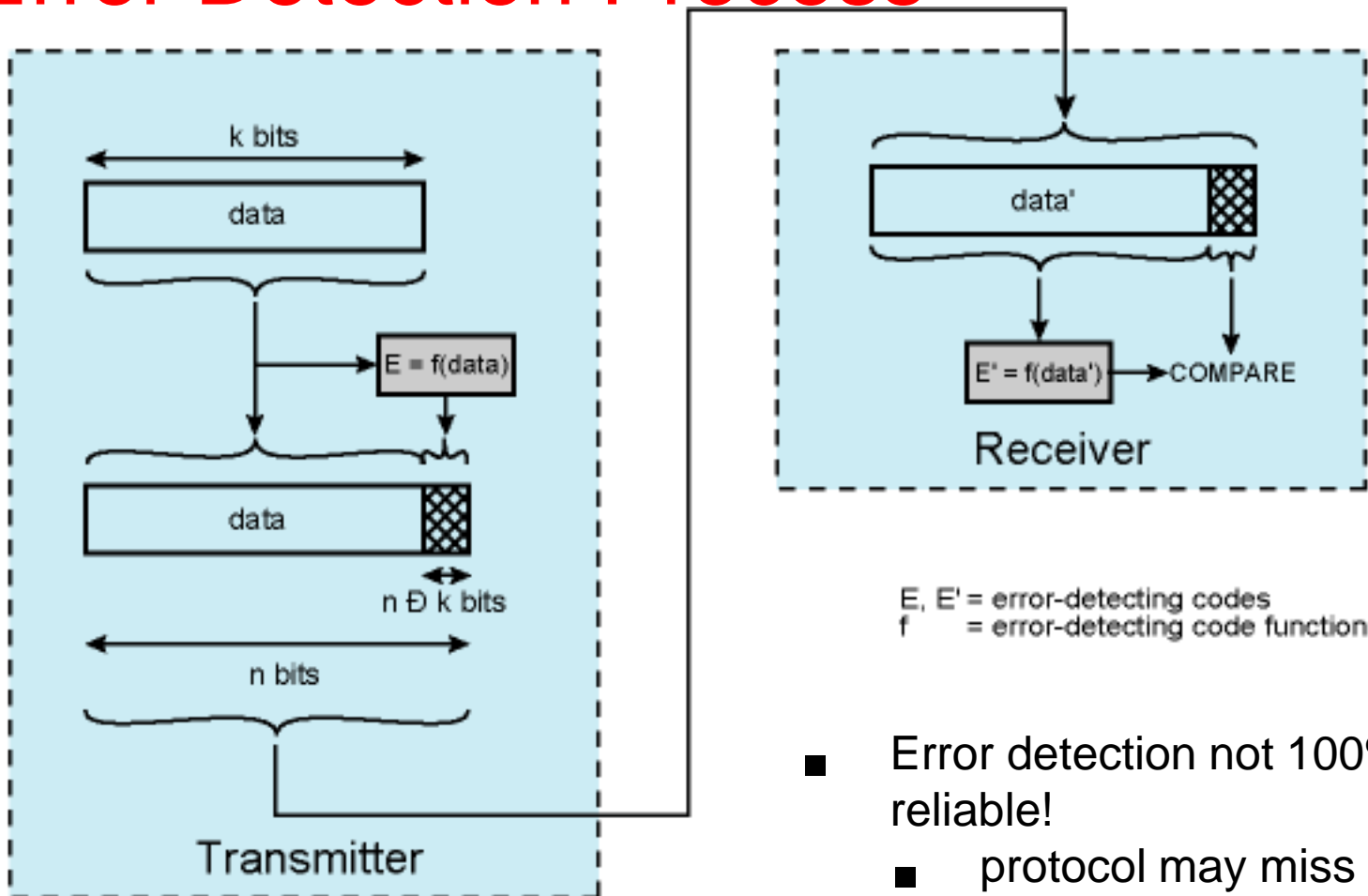
✓ Main Principle

- All transmitted data blocks (“codewords”) satisfy a pattern
- If received block does not satisfy pattern, it is in error
- Redundancy: Only a subset of all possible blocks can be codewords
- Blindspot: when channel transforms a codeword into another codeword



Error Control : Error Detection

✓ Error Detection Process



- Error detection not 100% reliable!
 - protocol may miss some errors, but rarely
 - larger ED field yields better detection

Error Control : Error Detection

✓ Single Parity Check

- Append an overall parity check to k information bits

Info Bits: $b_1, b_2, b_3, \dots, b_k$

Check Bit: $b_{k+1} = b_1 + b_2 + b_3 + \dots + b_k \text{ modulo } 2$

Codeword: $(b_1, b_2, b_3, \dots, b_k, b_{k+1})$

- All codewords have even # of 1s
- Receiver checks to see if # of 1s is even
- All error patterns that change an odd # of bits are detectable
- Parity bit used in ASCII code while gaining bits.

All even-numbered error patterns are undetectable...

Error Control : Error Detection

✓ Single Parity Check : Example

- Information (7 bits): (0, 1, 0, 1, 1, 0, 0)
- Parity Bit: $b_8 = 0 + 1 + 0 + 1 + 1 + 0 = 1$
- Codeword (8 bits): (0, 1, 0, 1, 1, 0, 0, 1)

- If single error in bit 3 : (0, 1, 1, 1, 1, 0, 0, 1)
- # of 1's =5, odd
- Error detected

- If errors in bits 3 and 5: (0, 1, 1, 1, 0, 0, 0, 1)
- # of 1's =4, even
- Error not detected

Error Control : Error Detection

✓ Other Error Detection Codes

- Many applications need very low error rate
- Require codes that detect the vast majority of errors
- Single parity check codes are not enough to detect errors
- The error detecting codes used in today systems:
 - CRC Polynomial Codes
 - Internet Check Sums (at Transport Layer)

Error Control : Error Detection

✓ Polynomial Codes

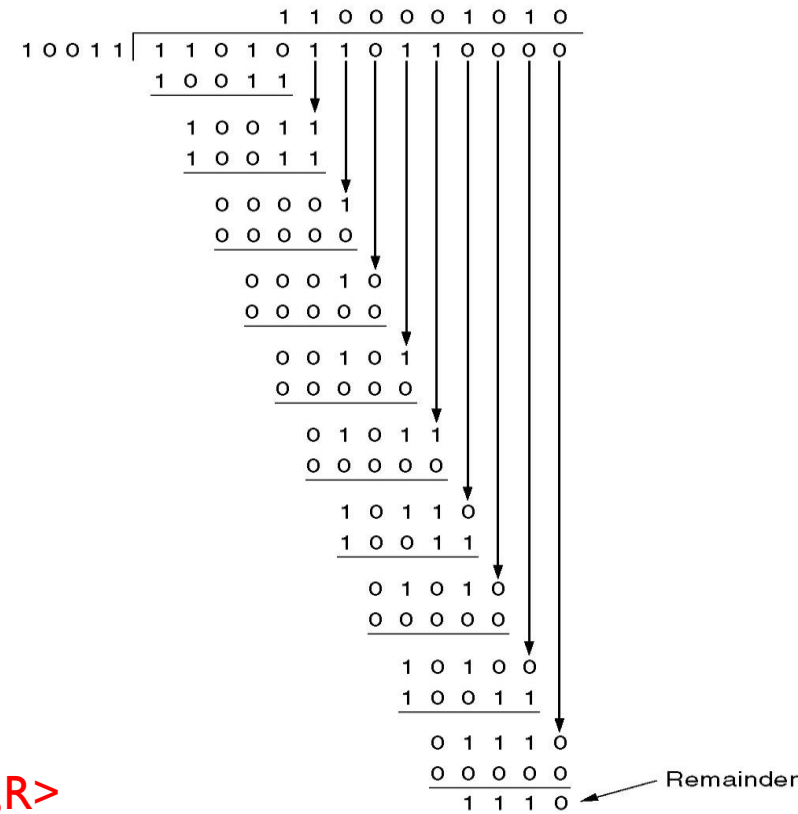
- Polynomials for codewords and polynomial arithmetic
- Implemented using shift-register circuits
- Named as redundancy check (CRC) codes
- Most data communications standards use polynomial codes for error detection
 - For a block of k bits transmitter produces n bit sequence
 - Transmit $k+n$ bits which is exactly divisible by some number
 - Receiver divides frame by that number
 - If no remainder, assume no error

Error Control : Error Detection

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Polynomial Codes (CRC Codes)

- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
 - <D,R> exactly divisible by G (modulo 2)
 - receiver knows G, divides <D,R> by G.
 - If non-zero remainder: error detected!

Divide D by G, get remainder R, and transmit <D,R>

can detect all burst errors less than r+1 bit

$R = \text{remainder} \left[\frac{D \cdot 2^r}{G} \right]$

widely used in practice (ATM, HDLC)

Error Control : Error Detection

✓ Standard Generator Polynomial Codes (CRC Codes)

- CRC-8 $= x^8 + x^2 + x + 1$ ATM
- CRC-16 $= x^{16} + x^{15} + x^2 + 1$
 $= (x + 1)(x^{15} + x + 1)$ Bisync
- CCITT-16 $= x^{16} + x^{12} + x^5 + 1$ HDLC, XMODEM, V.41
IEEE 802, DoD, V.42
- CCITT-32 $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Data Link Layer Functions

✓ Medium Access Control

- Less oftenly used on low bit error link (fiber, some twisted pair)
- wireless links: high error rates

✓ Framing

- encapsulate datagram into frame, adding header, trailer
- “MAC” addresses used in frame headers to identify source, destination
- channel access if shared medium

✓ Error Control(Detection & Correction):

○ Error Detection

- errors caused by signal attenuation, noise.
- Receiver does this mission (warns sender in order to retransmission or drops frame.)

○ Error Correction

- receiver finds *and corrects* bit error(s) without another retransmission

✓ Flow Control

- **Adaptation of data rates between adjacent sending and receiving nodes**

Flow Control & Error Control

✓ Flow Control

- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment from the receiver
 - Stop-and-Wait flow control
 - Sliding-Window flow control

✓ Error Control

- Refers to procedures to detect and correct errors
- Includes the following actions which called **Automatic Repeat Request (ARQ)**:
 - Error detection
 - **Positive Acknowledgement (ACK)**: if the frame arrived with no errors
 - **Negative Acknowledgement (NAK)**: if the frame arrived with error
 - Retransmissions after timeout: Frame is retransmitted after certain amount of time if no acknowledgement was received

Flow Control & Error Control

- ✓ Usually Error and flow control protocols are combined together to provide reliable data transfer service called data link control
 - Stop-and-Wait ARQ
 - Go-Back-N ARQ
 - Selective repeat ARQ
- ✓ ARQ provide **reliable data transfer** service over **unreliable networks**
- ✓ Despite errors, ARQ ensure that transmitted data is delivered accurately and satisfies the following:
 - **Error free**
 - **Without duplicates**
 - **Same order** in which they are transmitted
 - **No loss**

Flow Control

✓ Main Purpose

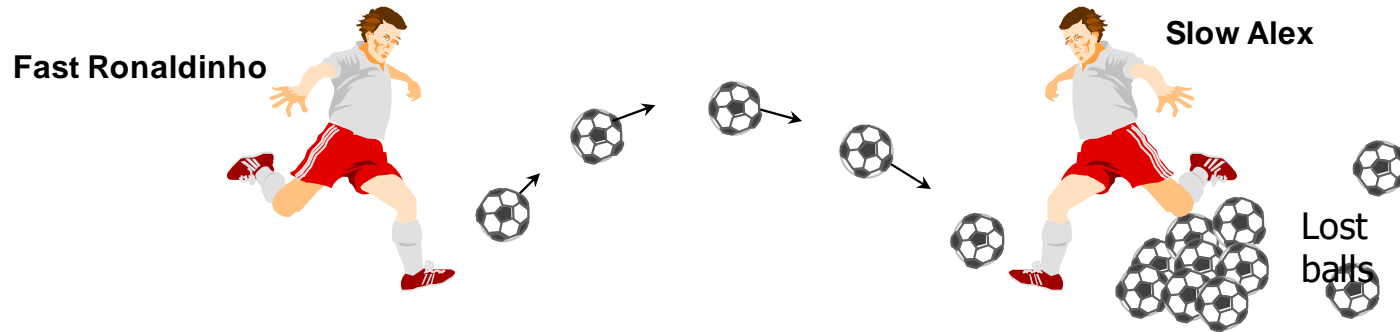
- It is to prevent the receiver from getting overloaded by the data sent by the faster-transmitting sender.
 - If a sender is on a powerful machine and it is transmitting the data at the faster rate,
 - it may happen that the receiver on slower-end is unable to receive data at that spit may loose some data.
 - eed
- Two methods of flow control,
 - feedback-based flow control
 - rate-based flow control

Flow Control

✓ Flow Control vs Congestion Control

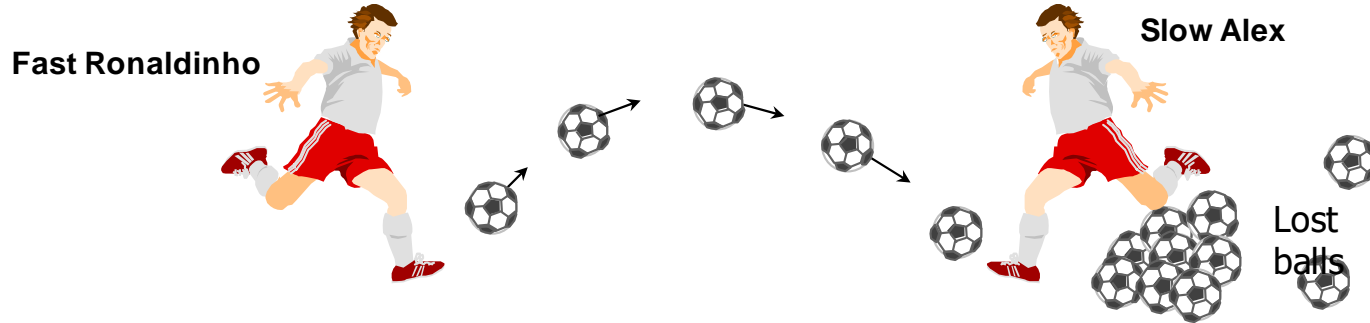
BASIS FOR COMPARISON	FLOW CONTROL	CONGESTION CONTROL
Basic	It controls the traffic from a particular sender to a receiver.	It controls the traffic entering the network.
Purpose	It prevents the receiver from being overwhelmed by the data.	It prevents the network from getting congested.
Responsibility	Flow control is the responsibility handled by data link layer and the transport layer.	Congestion Control is the responsibility handled by network layer and transport layer.
Responsible	The sender is responsible for transmitting extra traffic at receivers side.	The transport layer is responsible transmitting extra traffic into the network.
Preventive measures	The sender transmits the data slowly to the receiver.	Transport layer transmits the data into the network slowly.
Methods	Feedback-based flow control and Rate-based flow control	Provisioning, traffic-aware routing and admission control

Flow Control



- ✓ What to do with a sender that wants to transmit frames faster than the receiver can accept them ???
- ✓ Even if transmission is error free, the receiver may be unable to handle the frames
 - Might be possible for the sender to simply insert a delay to slow down sufficiently to keep from swamping the receiver

Flow Control



✓ Two approaches for flow control

- **Feedback-based flow control:** the receiver sends back information to the sender giving it permission to send more data or at least telling the sender how the receiver is doing.
- **Rate-based flow control:** the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver.

Flow Control

✓ Feedback-based flow control:

- it informs the sender,
- permits it to send more information
- it also inform about the status of the receiver
- There are two protocols of feedback-based flow control:
 - sliding window protocol
 - and stop-and-wait protocol

✓ Rate-Based flow control:

- when a sender transmits the data at a faster rate to the receiver
 - It has a the built-in mechanism in the protocol
 - **limit the rate of transmission of the sender without any feedback from the receiver.**

Flow Control

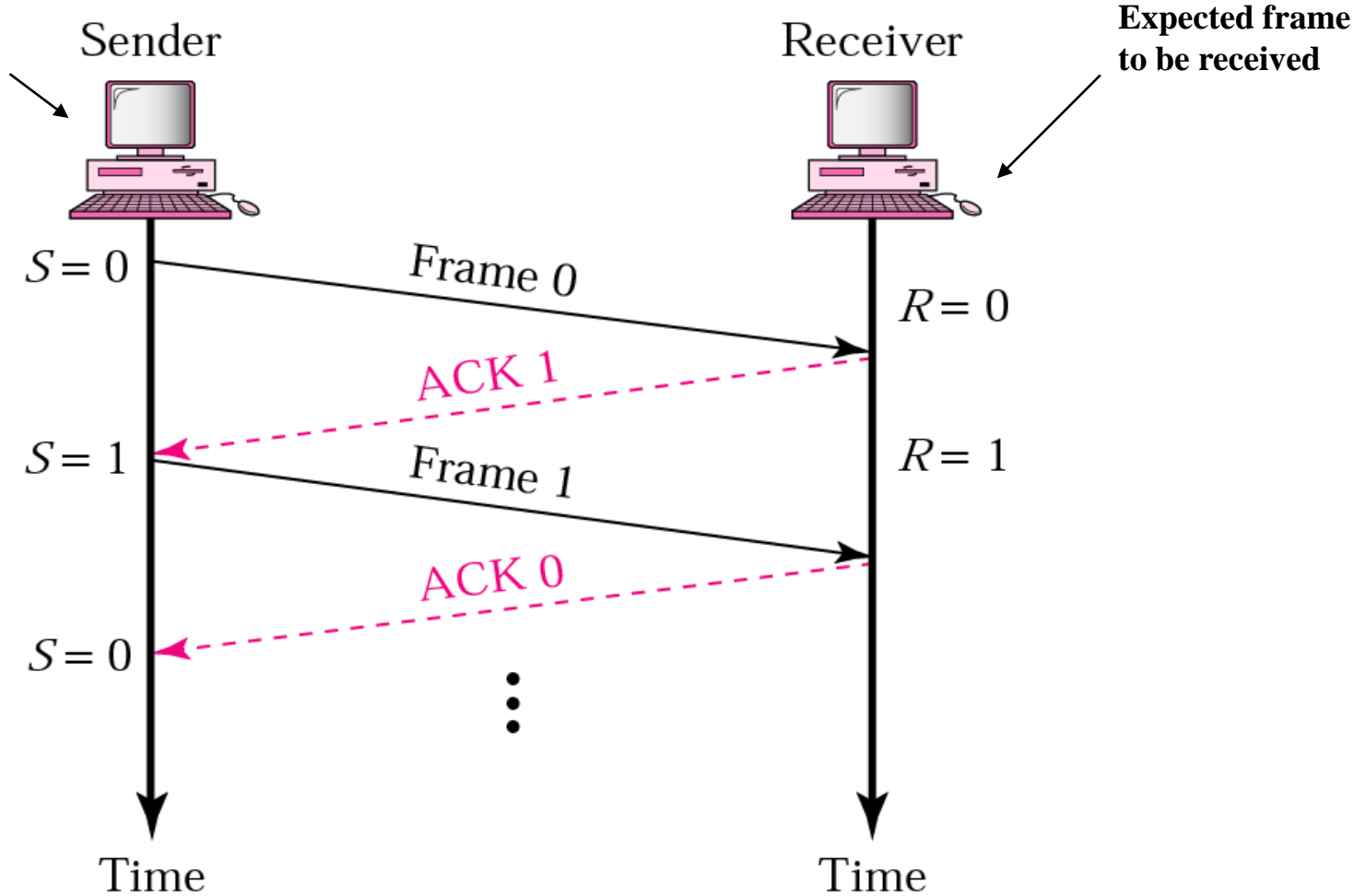
✓ Important Terms:

- Ensuring the sending entity does not overwhelm the receiving entity
 - Preventing buffer overflow
- Transmission time (t_{frame})
 - Time taken to emit all bits into medium
- Propagation time (t_{prop})
 - Time for a bit to traverse the link

Flow Control

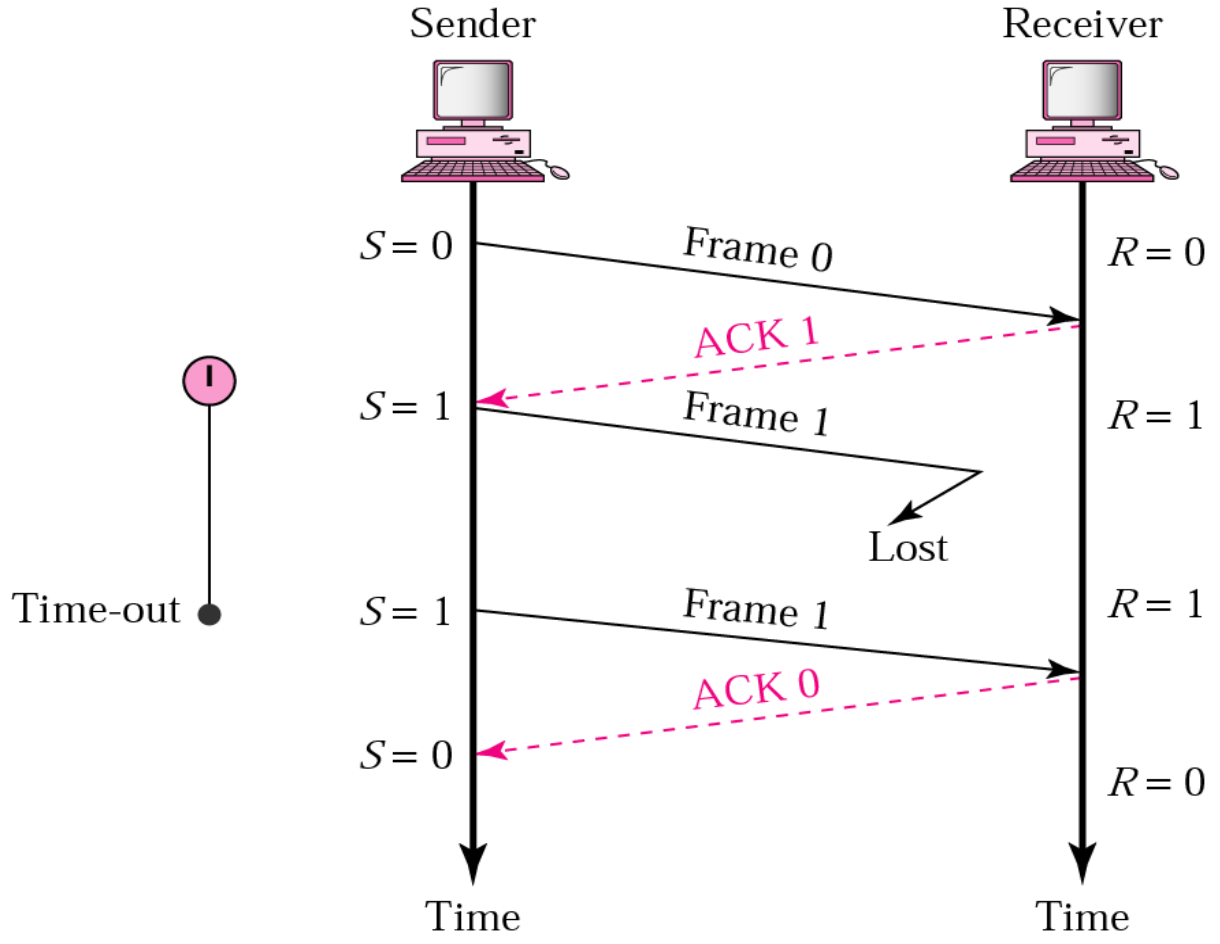
✓ Normal Operation

Last transmitted frame



Flow Control

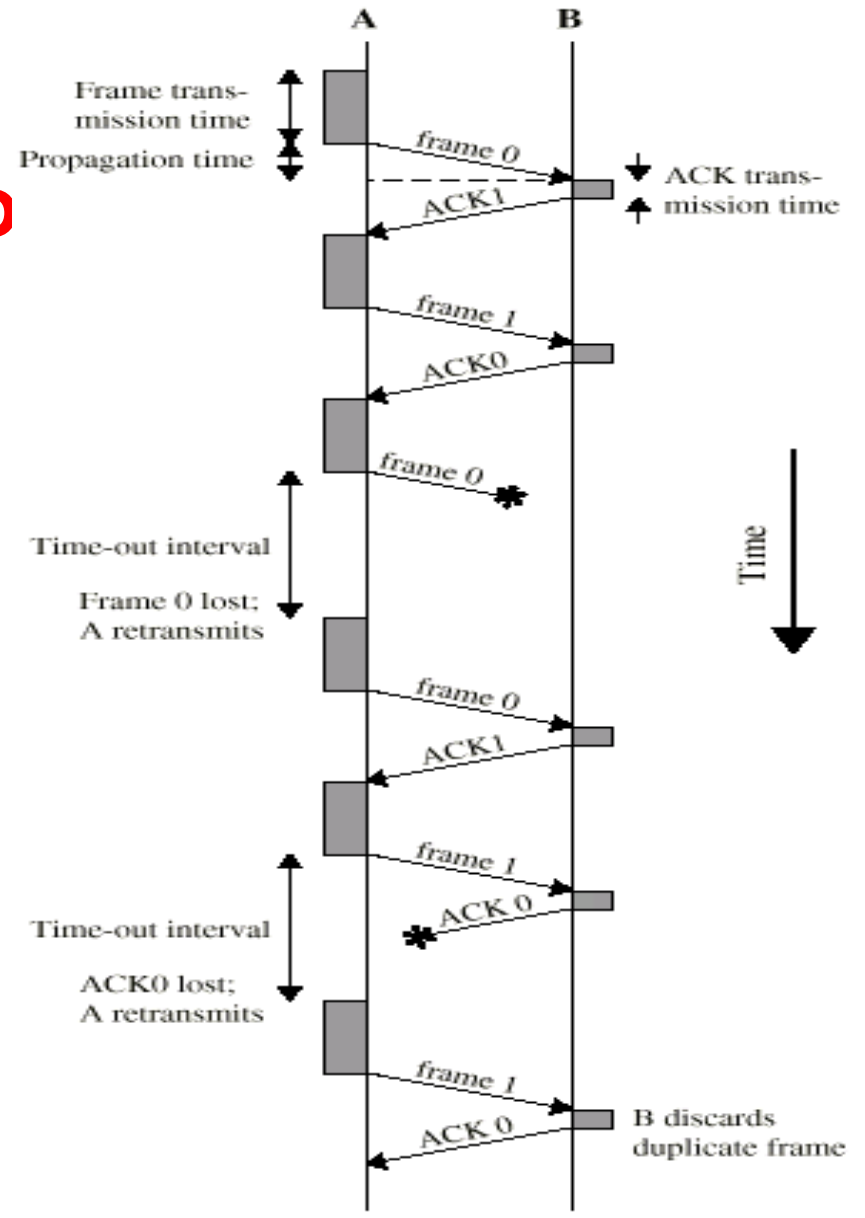
✓ Stop & Wait Operation : Loss Frame



Flow Control

✓ Stop & Wait Operatio

- Simple
- Efficient



Flow Control

✓ Flow Control Effect on Performance

○ Bandwidth Delay Product (BDP):

[data link's capacity (in bits per second)]

*

[round-trip delay time (in seconds)].

- The result, an amount of data measured in bits (or bytes), is equivalent **to the maximum amount of data on the network circuit at any given time, i.e., data that has been transmitted but not yet acknowledged**

Flow Control

✓ Flow Control Performance Example

Assume that, in a Stop-and-Wait system, the bandwidth of the line is 1 Mbps, and propagation time is 10 ms. What is the bandwidth-delay product? If the system data frames are 1000

Solution

The bandwidth-delay product is

$$(1 \times 10^6) \times (10 \times 10^{-3}) = 10,000 \text{ bits}$$

What is the time needed for an ACK to arrive? (ignore the ACK frame transmission time)?

$$= \text{Frame Transmission time} + 2 * \text{Propagation time} = 1000 / 10^6 + 2 \times 10 \times 10^{-3} = 0.021 \text{ sec}$$

How many frames can be transmitted during that time?

$$= (\text{time} * \text{bandwidth}) / (\text{frame size in bits})$$

$$0.021 \times (1 \times 10^6) = 21000 \text{ bits} / 1000 = 21 \text{ frames}$$

What is the Link Utilization if stop-and-wait is used?

$$\text{Link Utilization} = (\text{number of actually transmitted frame} / \# \text{ frames that can be transmitted}) * 100$$

$$(1/21) \times 100 = 5 \%$$

Flow Control

✓ Feedback-based flow control:

- There are two protocols of feedback-based flow control:
 - **Sliding window protocol**
 - Sliding Window with Go Back N
 - Sliding Window with Selective Repeat
 - Stop-and-wait protocol

✓ Rate-Based flow control:

- when a sender transmits the data at a faster rate to the receiver
 - It has a the built-in mechanism in the protocol
 - **limit the rate of transmission of the sender without any feedback from the receiver.**

Flow Control

✓ Sliding Window flow control:

- Let multiple frames to be in transit
- The buffer of the receiver is W long (**receiver window**)
- Transmitter can send up to W frames without ACK (**sender window**)
- Each frame is numbered (**sequence number**)
- ACK has number of next frame expected
 - Ack for frame n = I am expecting frame $n+1$ (not “I received frame n ”)
- Sequence number bounded by size of field
 - e.g. k bits: Frames are numbered modulo 2^k

Flow Control

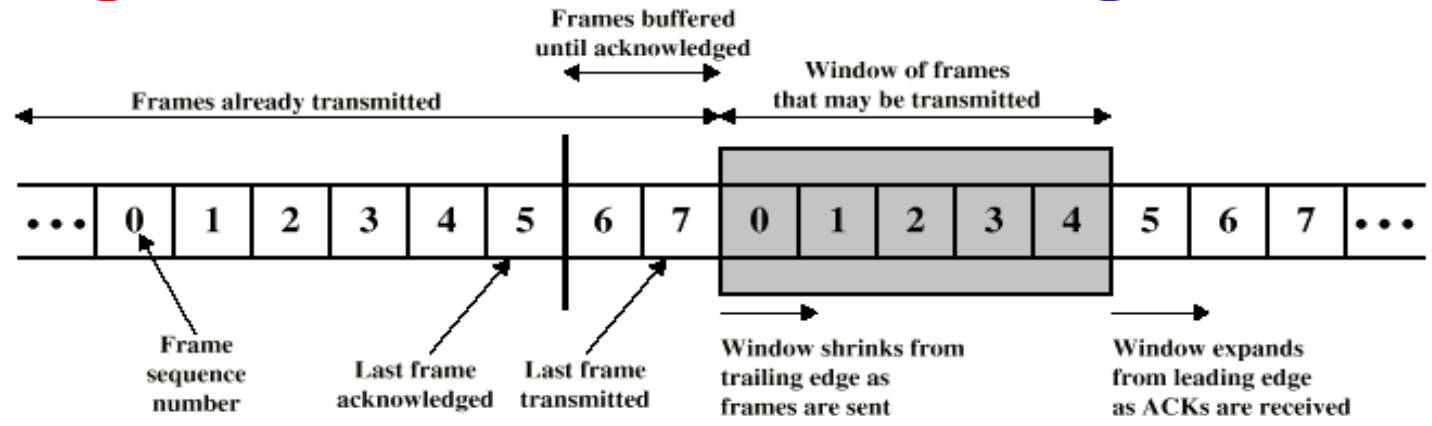
✓ Sliding Window flow control with Window Size

W:

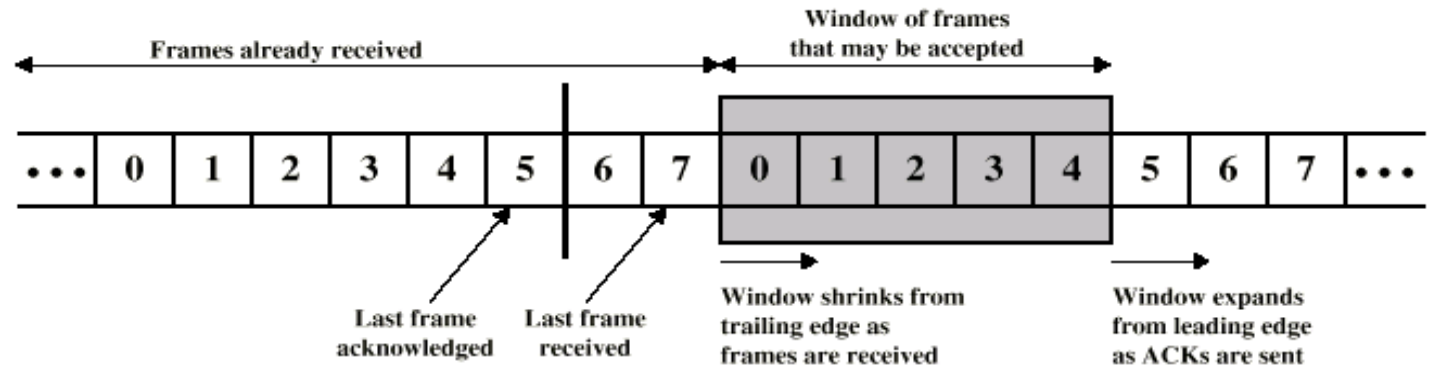
- With a window size of 1
 - the sender waits for an ACK before sending another frame
 - **This protocol behaves identically to stop and wait for a noisy channel!**
- With a window size of W , the sender can transmit up to W frames before “being blocked”
- We call using larger window sizes by using **Pipelining**

Flow Control

✓ Sliding Window flow control Diagram:



(a) Sender's perspective

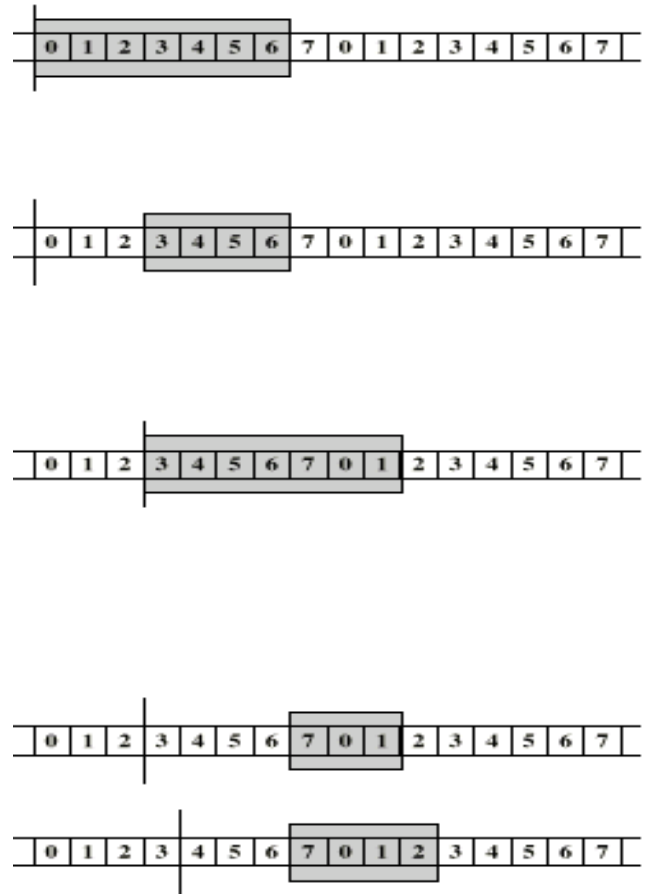


(b) Receiver's perspective

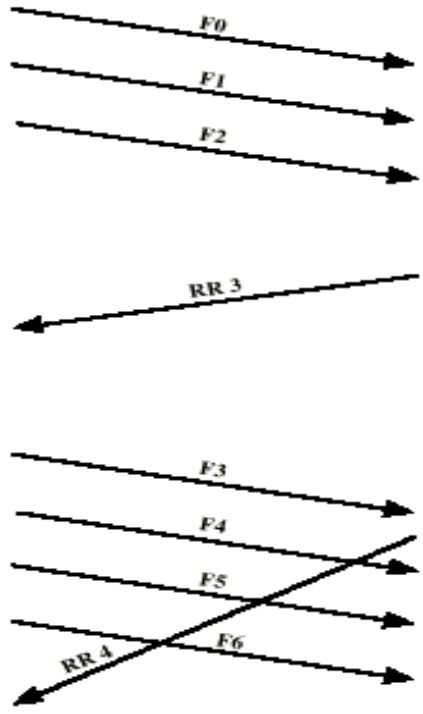
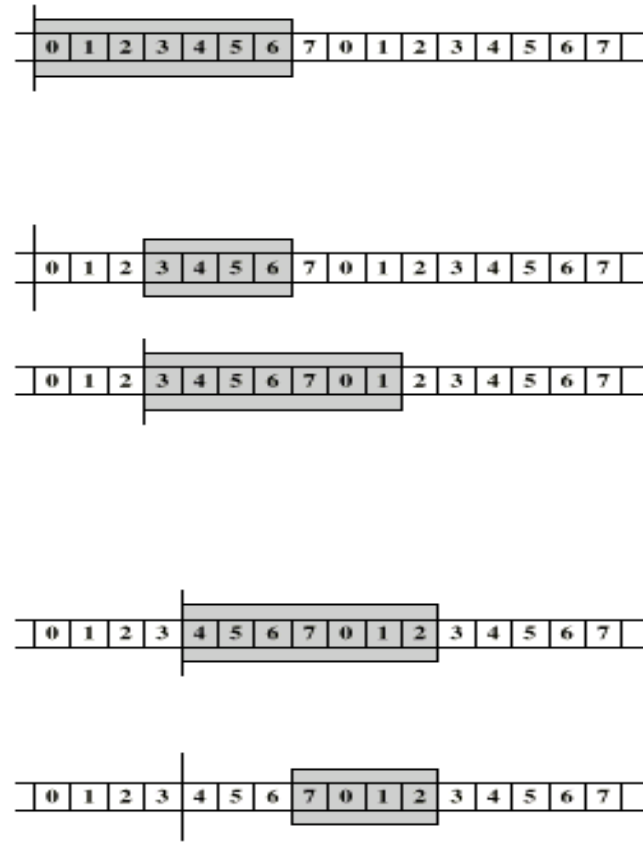
Flow Control

✓ Sliding Window flow control Example:

Source System A



Destination System B



Flow Control

✓ Feedback-based flow control:

- There are two protocols of feedback-based flow control:
 - Sliding window protocol
 - **Sliding Window with Go Back N**
 - **Sliding Window with Selective Repeat**
 - Stop-and-wait protocol

✓ Rate-Based flow control:

- when a sender transmits the data at a faster rate to the receiver
 - It has a the built-in mechanism in the protocol
 - **limit the rate of transmission of the sender without any feedback from the receiver.**

Flow Control

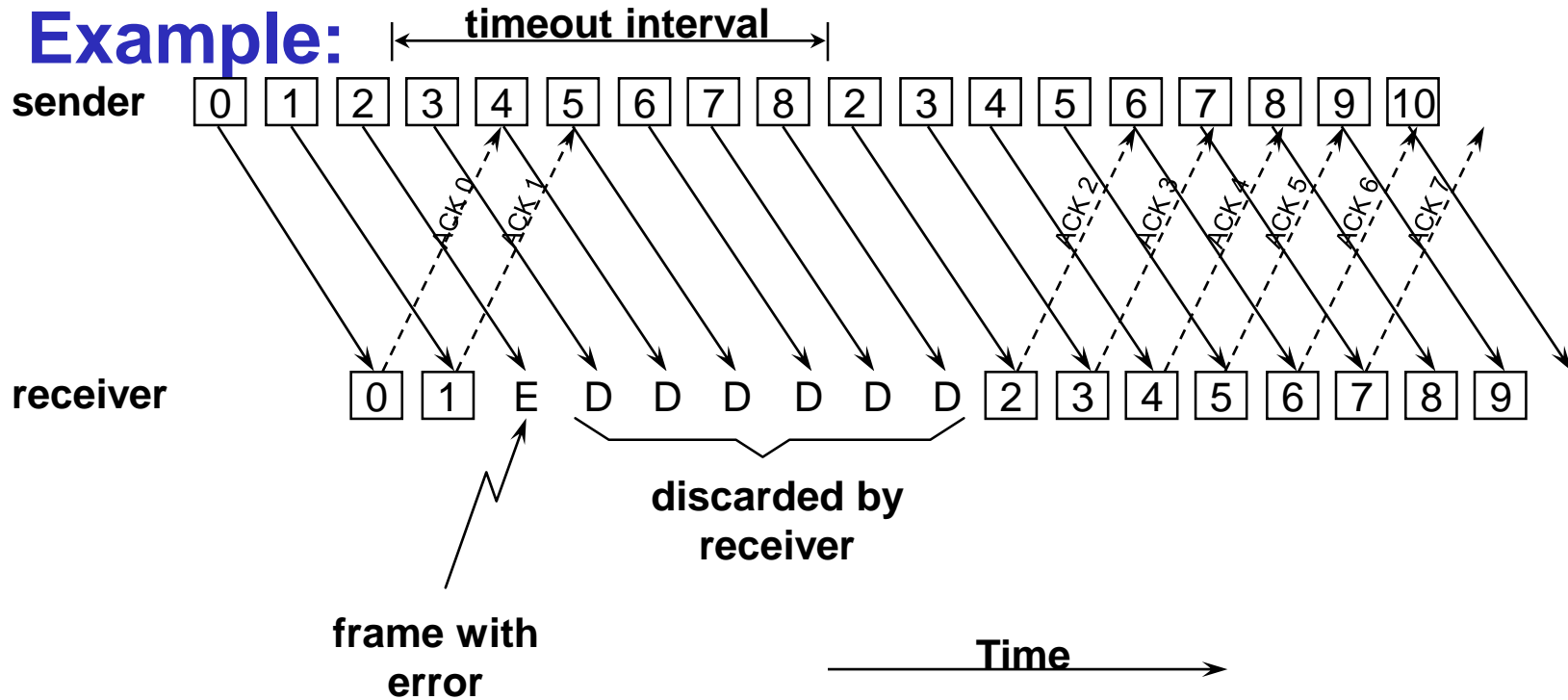
✓ **Sliding Window flow control with Go Back N:**

- ✓ Depending on sliding window
- ✓ If no error occurs, ACK as normal with next frame is wanted
- ✓ Use window to control amount of outstanding frames
- ✓ If error occurs, reply with rejection
 - Discard that frame and all future frames until error/lost frame received correctly
 - Transmitter must go back and retransmit that frame and all subsequent frames

Flow Control

✓ Sliding Window flow control with Go Back N

Example:



- ✓ Assume ACK per frame by receiver (receiver window=1)
- ✓ Go Back N can recover from erroneous or missing packets
- ✓ **Inefficient** → if there are a lot of errors, the sender will spend most of its time retransmitting

Flow Control

✓ **Sliding Window flow control with Selective**

Repeat

○ The sender retransmits only the frame with errors.

- **The receiver stores all the correct frames that arrive following the bad one (receiver window > 1)**
- Needs a significant amount of buffer space at the receiver
- When the sender understands that something is wrong, it just retransmits the one bad frame, not all its successors
- Might be combined with Negative Acknowledgment (NACK)

Flow Control

✓ **Sliding Window flow control with Selective Repeat**

Example:

