

Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's [policies and procedures](#).

Further information on data protection and best practice when using videoconferencing software is available at
https://www.tcd.ie/info_compliance/data-protection/.

© Trinity College Dublin 2020





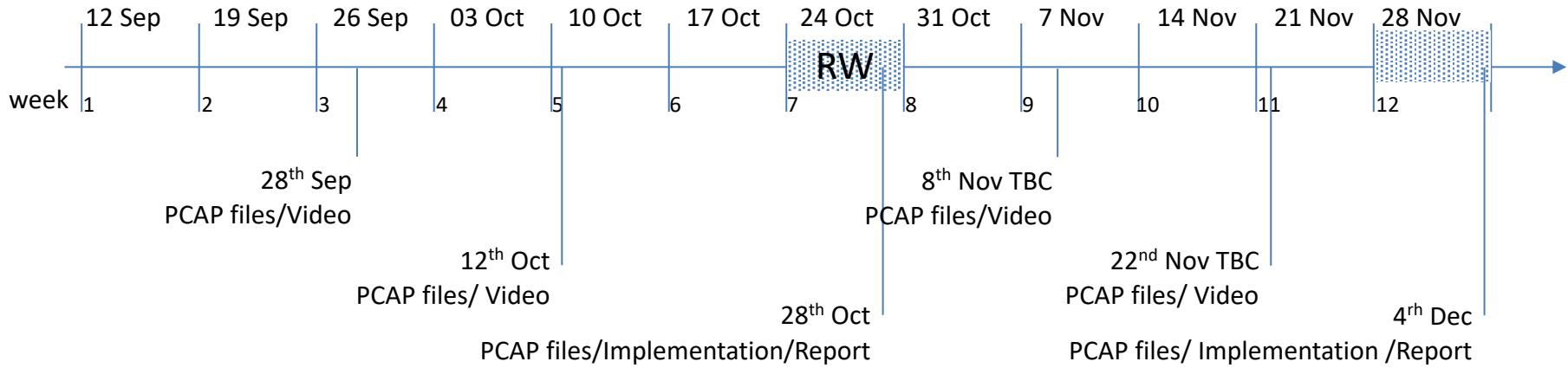
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU33031 Computer Networks

Flow Control

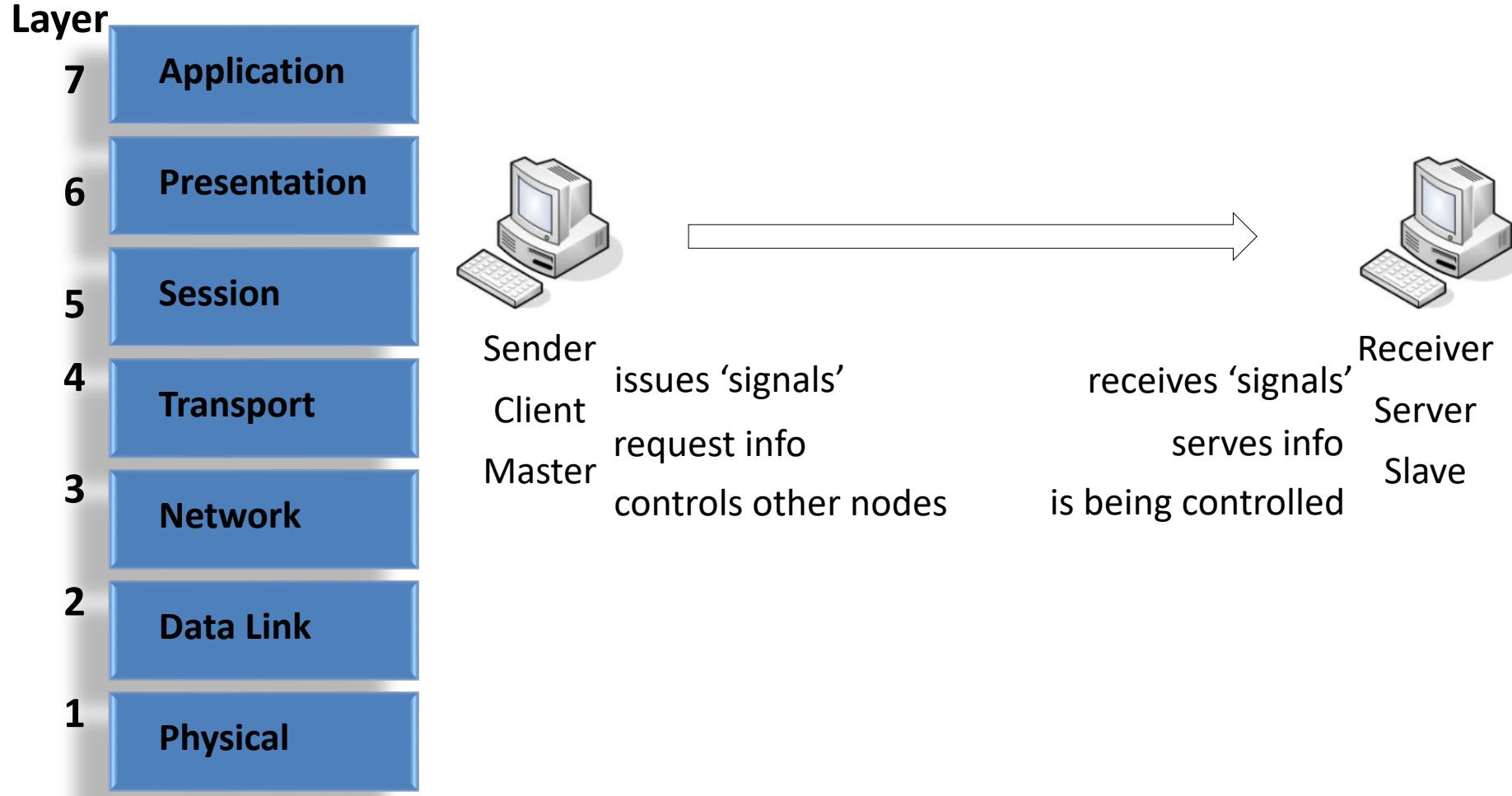
Stefan Weber
email: sweber@tcd.ie
Office: Lloyd 1.41

CSU33031 Timeline & Weights

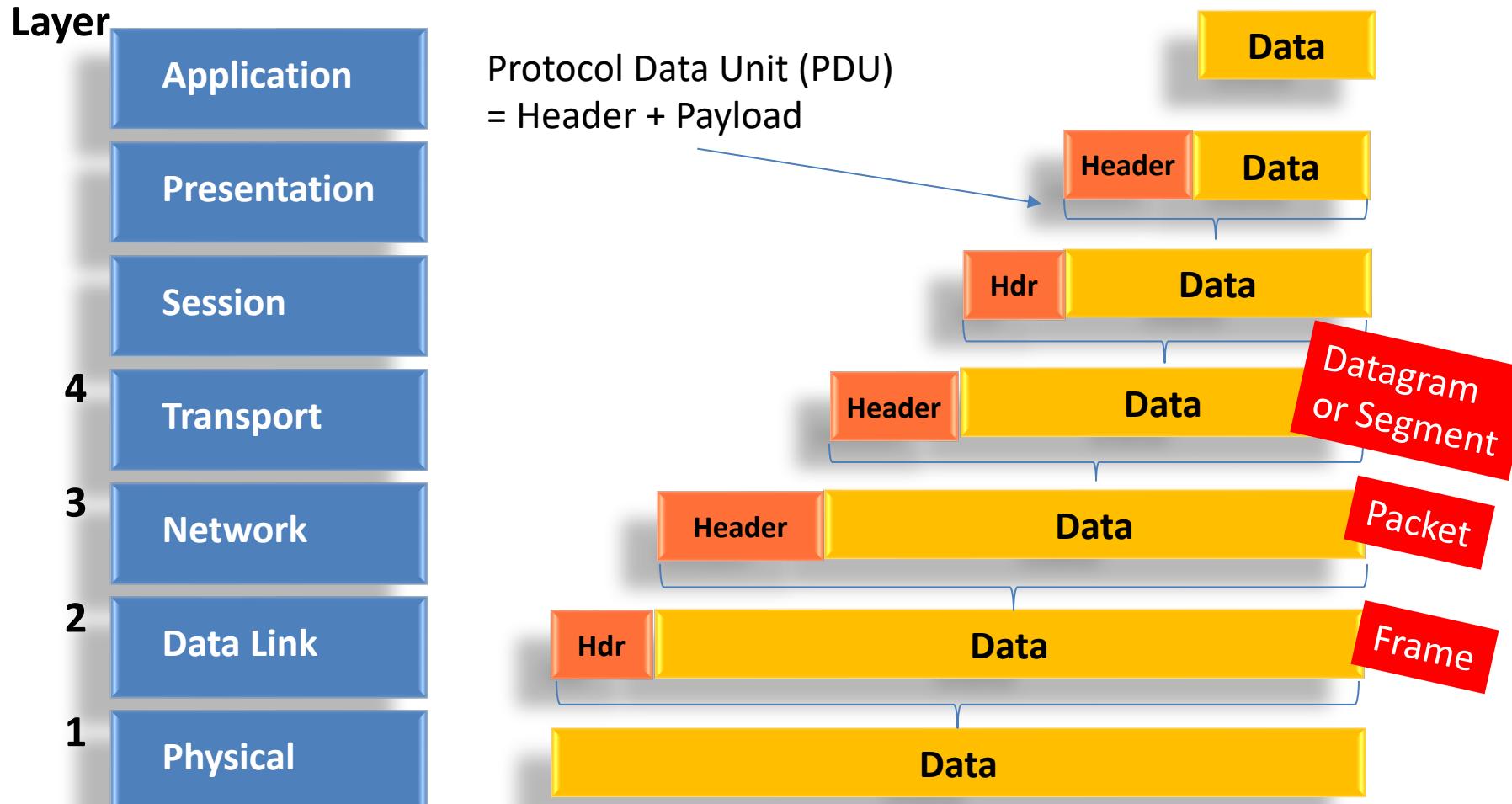


- 60% Coursework
 - 30% Assignment 1
 - 30% Assignment 2
- 40% Exam
- Supplemental exam - 100%

OSI Stack



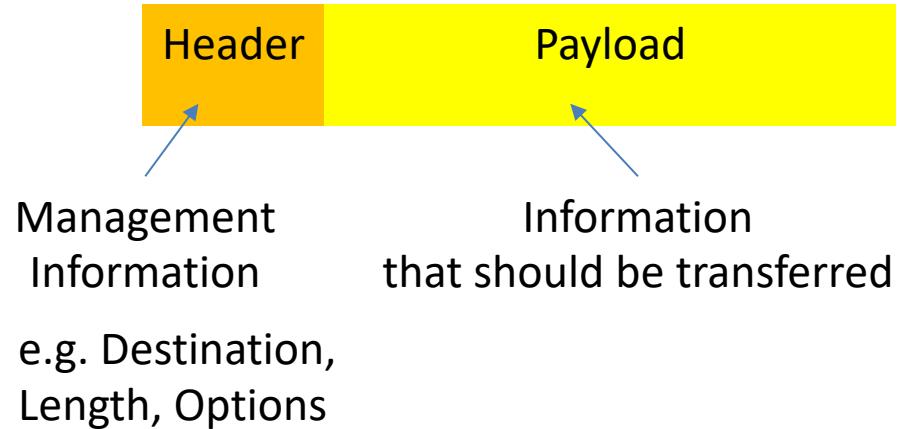
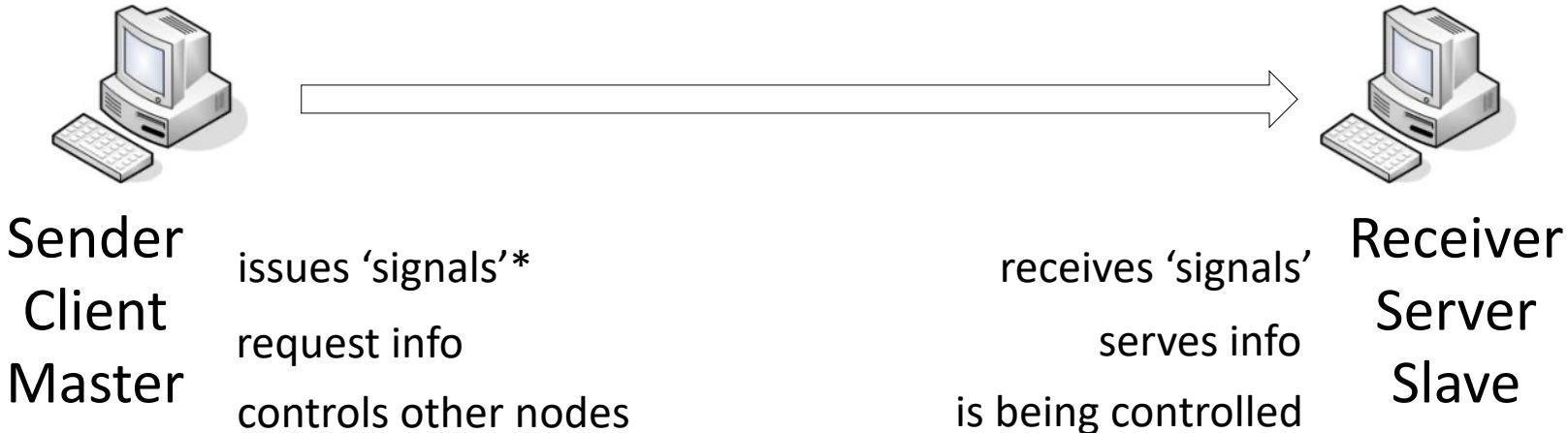
OSI Stack & Data Units



Payload - also called Service Data Unit (SDU) - if you say SDU, though, most people will blankly stare at you!

General Terms

- Endpoints, Header, Protocols



Protocols:

- Defined by information in the header
- Keep state between endpoints consistent.

*Sometimes called Transmitter/Receiver, or short Tx and Rx

HTTP – TCP – IPv4 - Ethernet

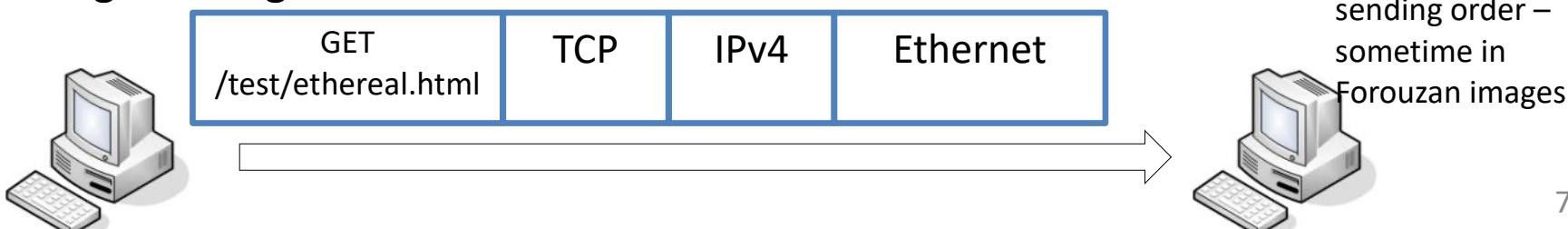
1	0.000000	192.16...	192.168.69.1	TCP	74	34059 → 80	[SYN]	Seq=0 Win=5840 Len=0 MSS=1
2	0.000059	192.16...	192.168.69.2	TCP	74	80 → 34059	[SYN, ACK]	Seq=0 Ack=1 Win=5792
3	0.000153	192.16...	192.168.69.1	TCP	66	34059 → 80	[ACK]	Seq=1 Ack=1 Win=5888 Len=0
4	0.000282	192.16...	192.168.69.1	HTTP	511	GET /test/ethereal.html	HTTP/1.1	
5	0.000320	192.16...	192.168.69.2	TCP	66	80 → 34059	[ACK]	Seq=1 Ack=1 Win=5888 Len=0

- > Frame 4: 511 bytes on wire (4088 bits), 511 bytes captured (4088 bits)
- > Ethernet II, Src: 00:0a:95:67:49:3c, Dst: 00:c0:f0:2d:4a:a3
- > Internet Protocol Version 4, Src: 192.168.69.2 (192.168.69.2), Dst: 192.168.69.1 (192.168.69.1)
- > Transmission Control Protocol, Src Port: 34059, Dst Port: 80, Seq: 1, Ack: 1, Len: 445
- > Hypertext Transfer Protocol



from left to right
- order in which
most publ.
display protocols

In some diagrams: right-to-left

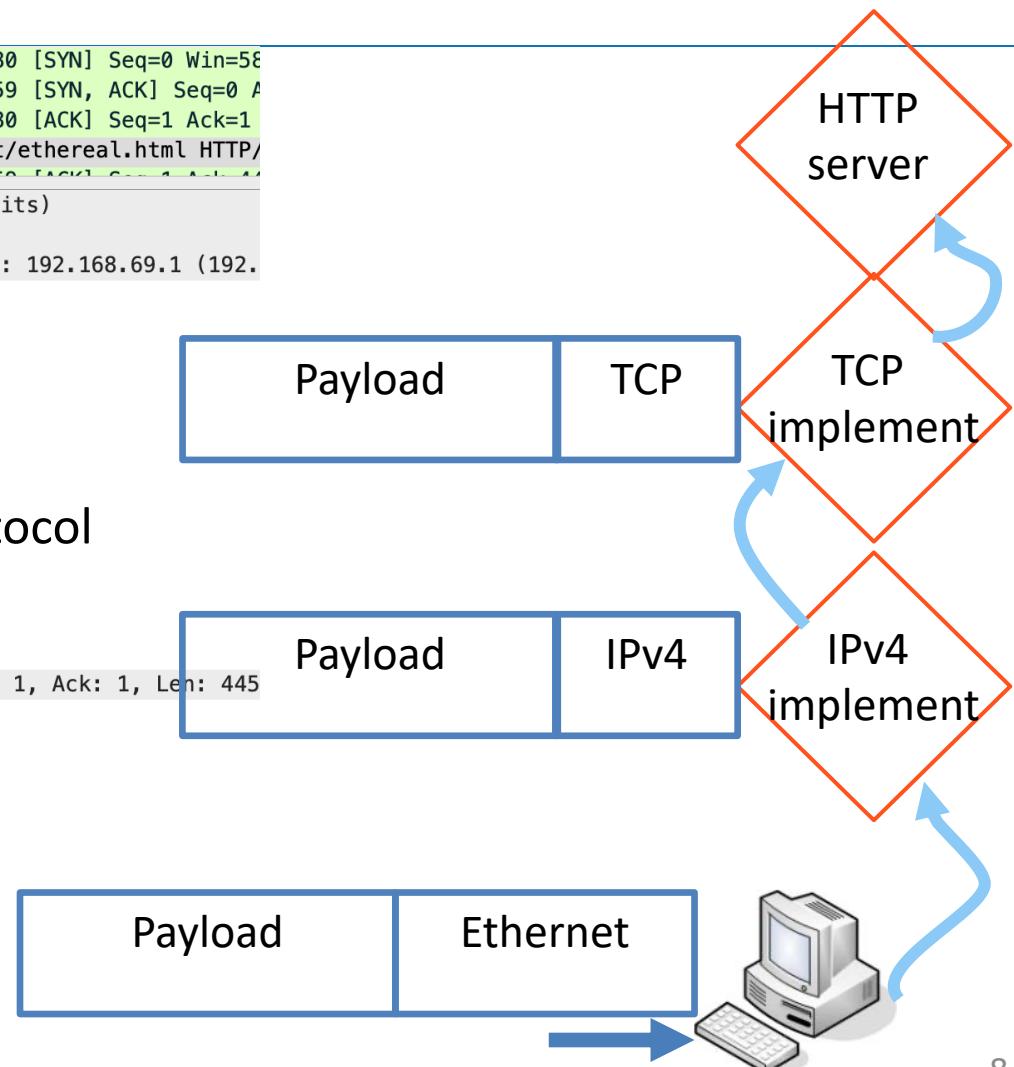


right to left –
sending order –
sometime in
Forouzan images

HTTP – TCP – IPv4 - Ethernet

```
1 0.000000 192.16... 192.168.69.1 TCP 74 34059 → 80 [SYN] Seq=0 Win=58
2 0.000059 192.16... 192.168.69.2 TCP 74 80 → 34059 [SYN, ACK] Seq=0 Ack=1
3 0.000153 192.16... 192.168.69.1 TCP 66 34059 → 80 [ACK] Seq=1 Ack=1
4 0.000282 192.16... 192.168.69.1 HTTP 511 GET /test/ethereal.html HTTP/1.1
5 0.000320 192.16... 192.168.69.2 TCP 66 80 → 34059 [ACK] Seq=1 Ack=1

> Frame 4: 511 bytes on wire (4088 bits), 511 bytes captured (4088 bits)
> Ethernet II, Src: 00:0a:95:67:49:3c, Dst: 00:c0:f0:2d:4a:a3
< Internet Protocol Version 4, Src: 192.168.69.2 (192.168.69.2), Dst: 192.168.69.1 (192.0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 497
Identification: 0xf5fdb (62939)
> Flags: 0x40, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6) ← next protocol
Header Checksum: 0x37d7 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.69.2 (192.168.69.2)
Destination Address: 192.168.69.1 (192.168.69.1)
< Transmission Control Protocol, Src Port: 34059, Dst Port: 80, Seq: 1, Ack: 1, Len: 445
Source Port: 34059
Destination Port: 80
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 445]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2415239731
[Next Sequence Number: 446 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
```

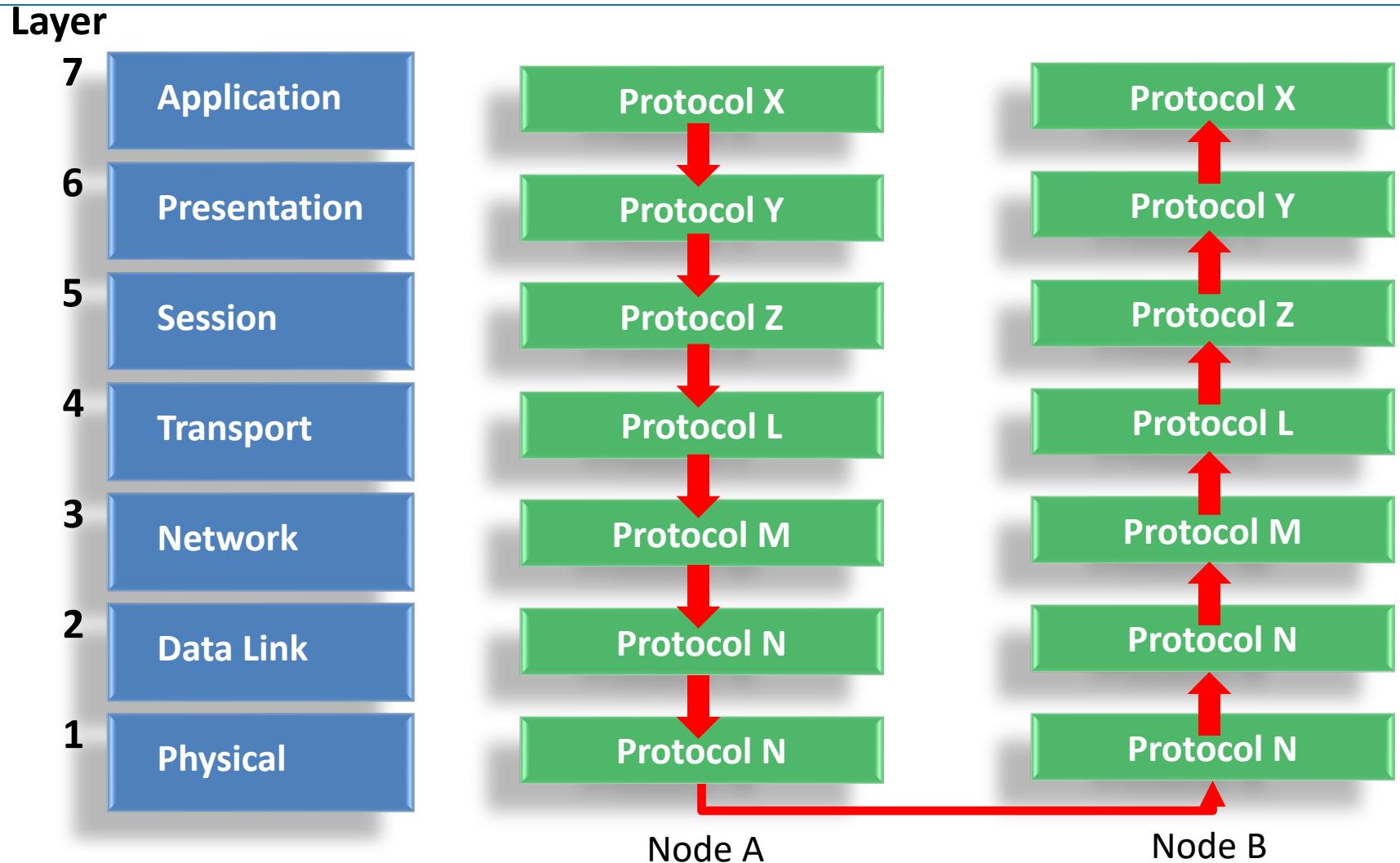


OSI Stack Populated with Protocols

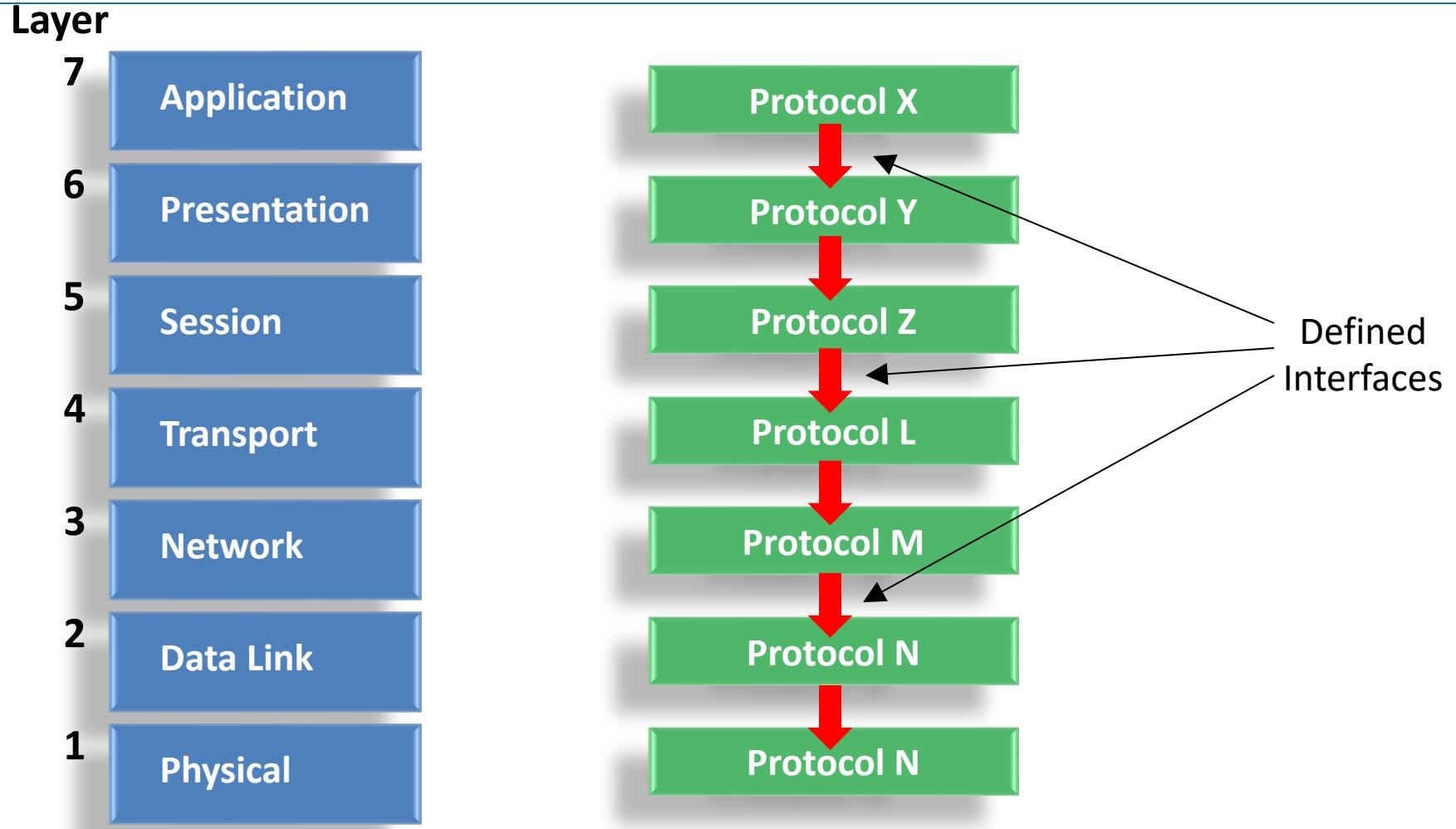
7	FTAM ISO 8571	CMISE ISO 9596 ISO 9595	Application
6	ACSE X.227, ISO 8650 X.217, ISO 8649	ACSE X.227, ISO 8650 X.217, ISO 8649	ROSE ISO 9072 X.219, X.229
5	X.226, ISO 8823 X.209, ISO 8825 BER X.216, ISO 8822		Presentation
4	X.225, ISO 8327 X.215, ISO 8326		Session
3	X.224, ISO 8073 / AD 2 (X.214, ISO 8072 / AD 2) class 4		Transport
2	ISO 9542 (ES-IS) ISO 10589 (IS-IS Level 1) / ISO 10747 (IS-IS Level 2)		Network
1	ISO 8473-3 (CLNS) ISO 8208 X.25 Packet Level	ISO 8473-2 (CLNS)	ISO 8473-4 (CLNS)
2	ISO 7776 LapB X.25 Data Link Layer	ISO 802.2 LLC ISO 802.3 MAC	Q.921 LapD
1	X.21 / X.21bis V.35 / G.703 2M TS n	ISO 802.2 Ethernet	SDH-DCC 2M TS n

<http://www.cellsoft.de/telecom/dcn.htm> 9

OSI Stack



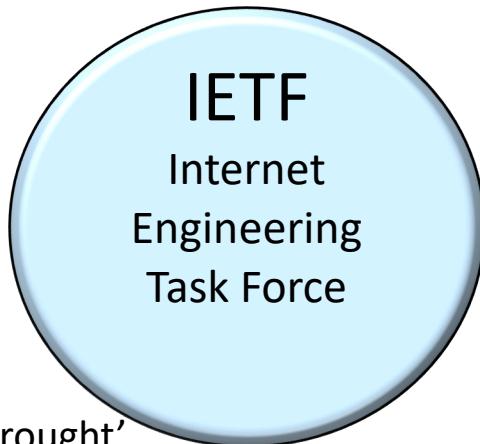
OSI* Stack



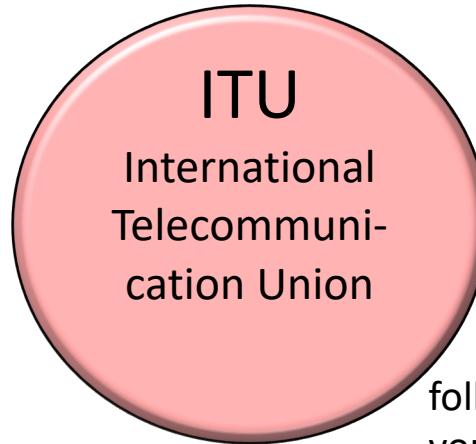
*OSI = Open Systems Interconnection 11

The Good, the Bad, and the Ugly

Also known as “your beloved standards bodies”



folks who ‘brought’
you the Internet
i.e. RFCs, IANA, etc



folks who ‘brought’
you ATM, 4G, 5G,
etc



folks who ‘brought’ you Wifi
and Ethernet i.e. 802.11*
and 802.3*, etc

ITU Diagram Example

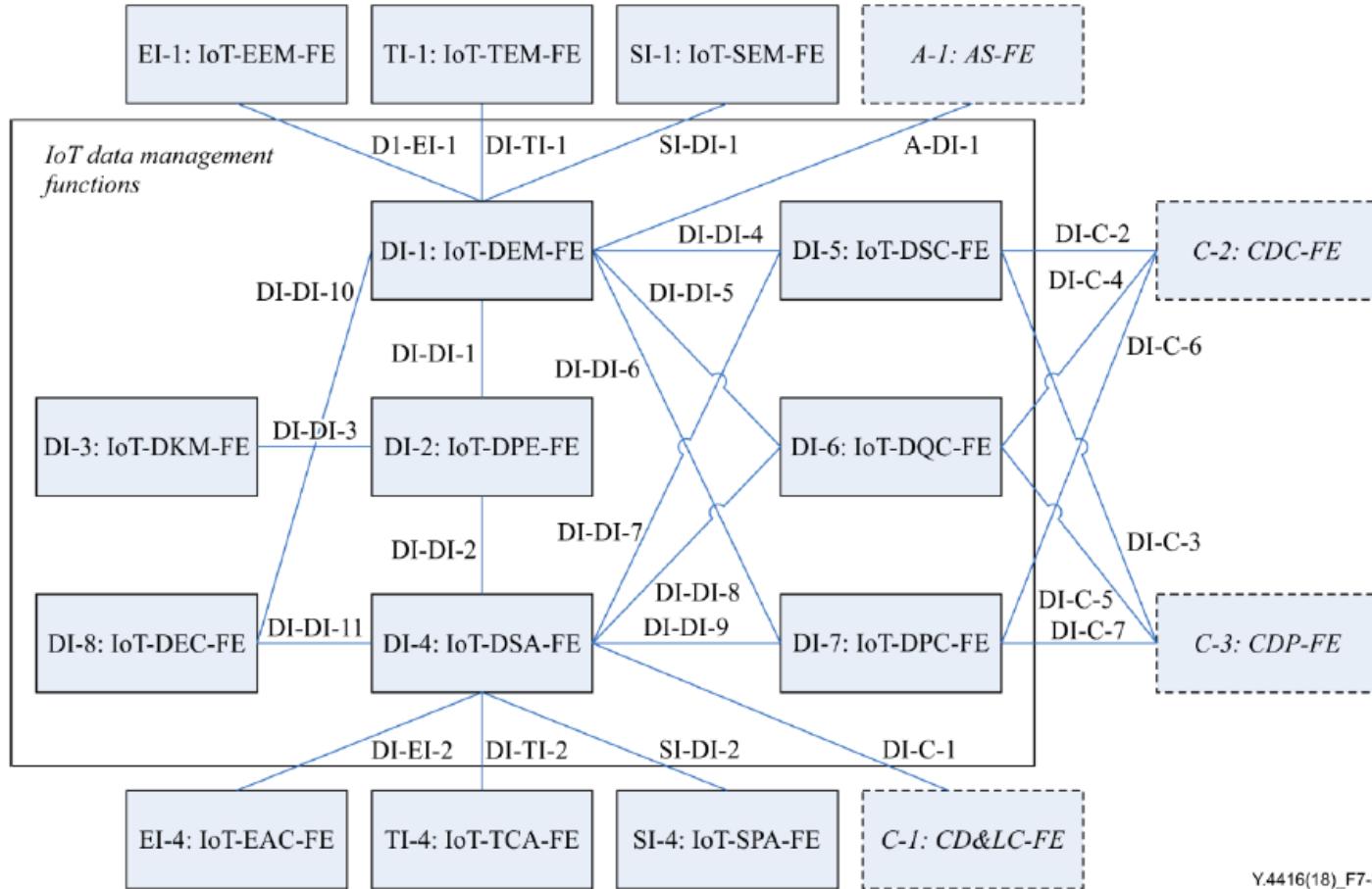
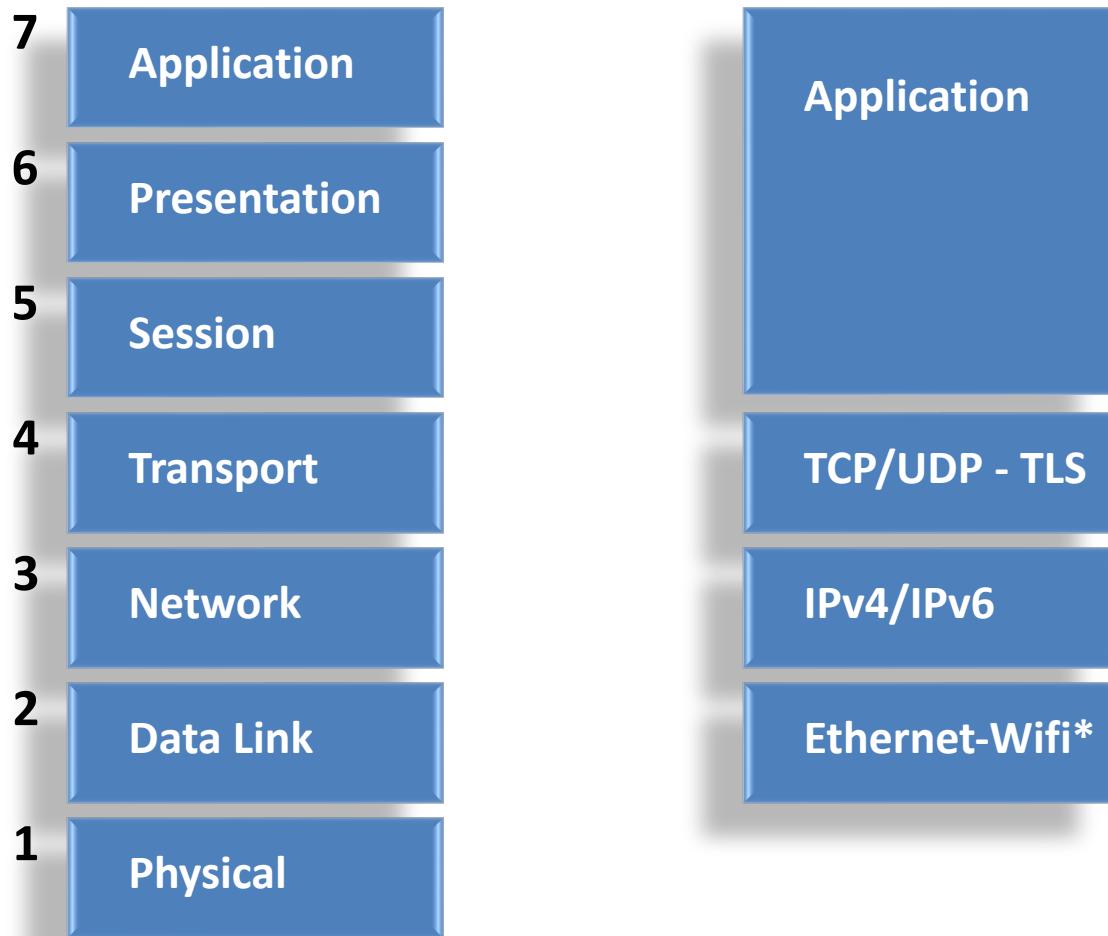


Figure 7-3 – IoT data management functional entities

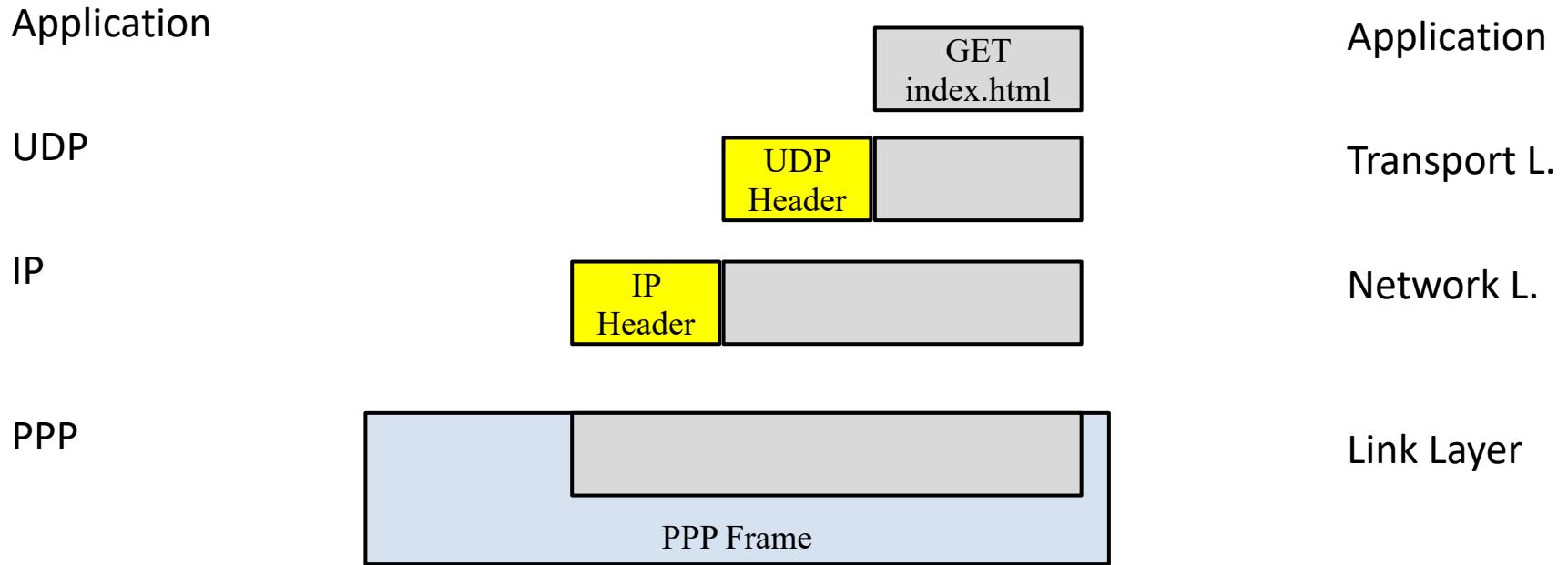
Rec. ITU-T Y.4416 (06/2018) 13

OSI Stack vs TCP/IP



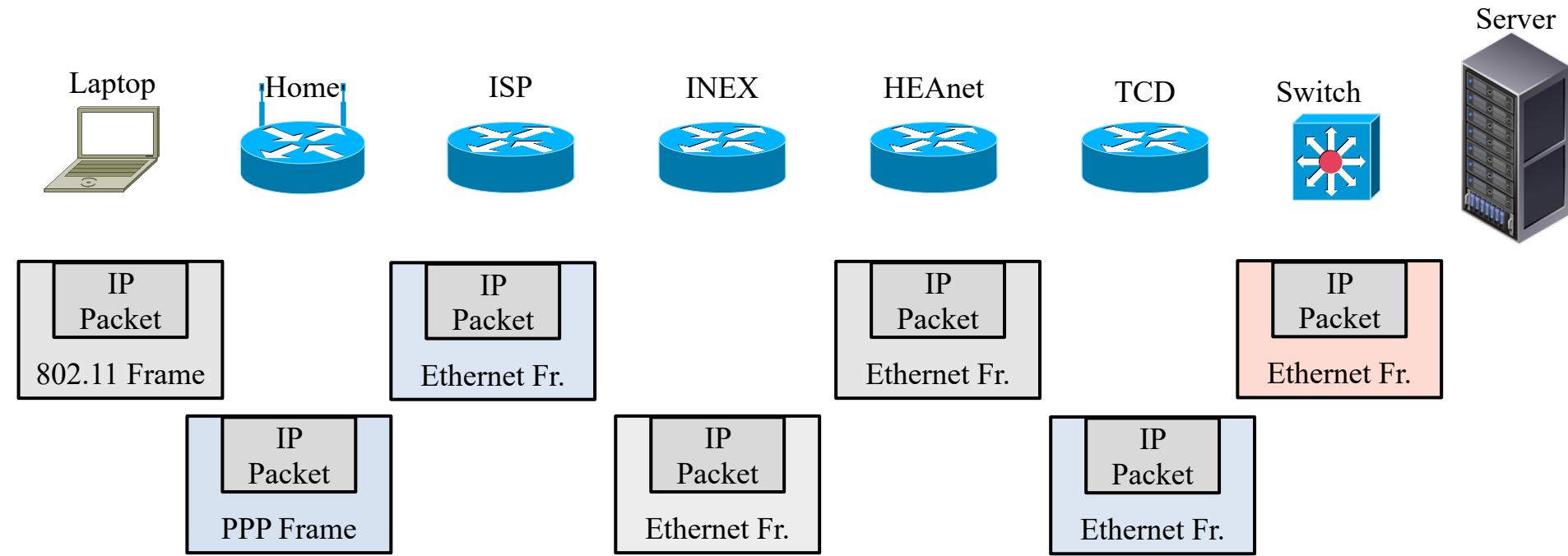
Wifi – spelled 802.11 b/g/n/ac/ax/?? 14

Encapsulation

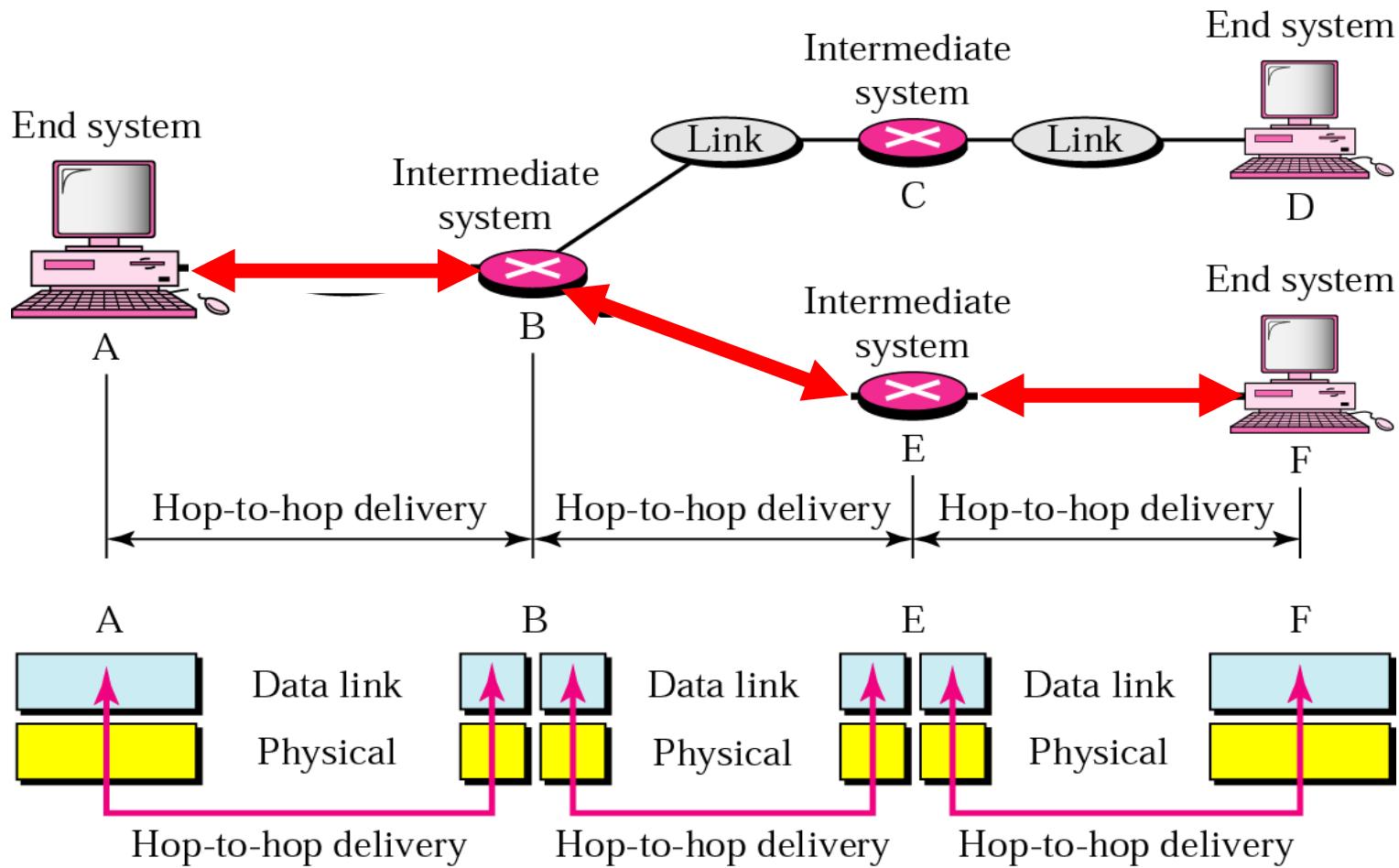


Overhead: Management information needed to deliver payload
eg. UDP header 8 bytes, IP header 20 bytes, PPP 8 bytes

Hop-by-Hop Communication

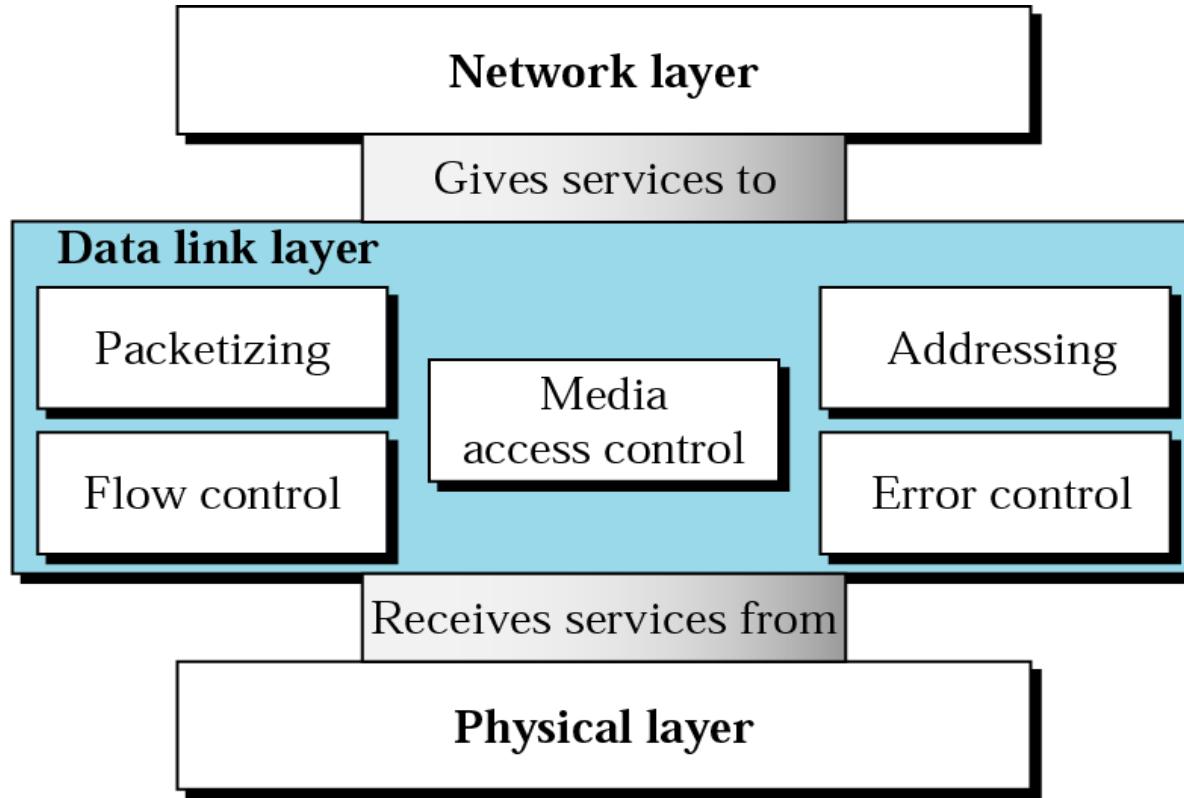


Link Layer



* Figure is courtesy of B. Forouzan

Duties of the Data Link Layer



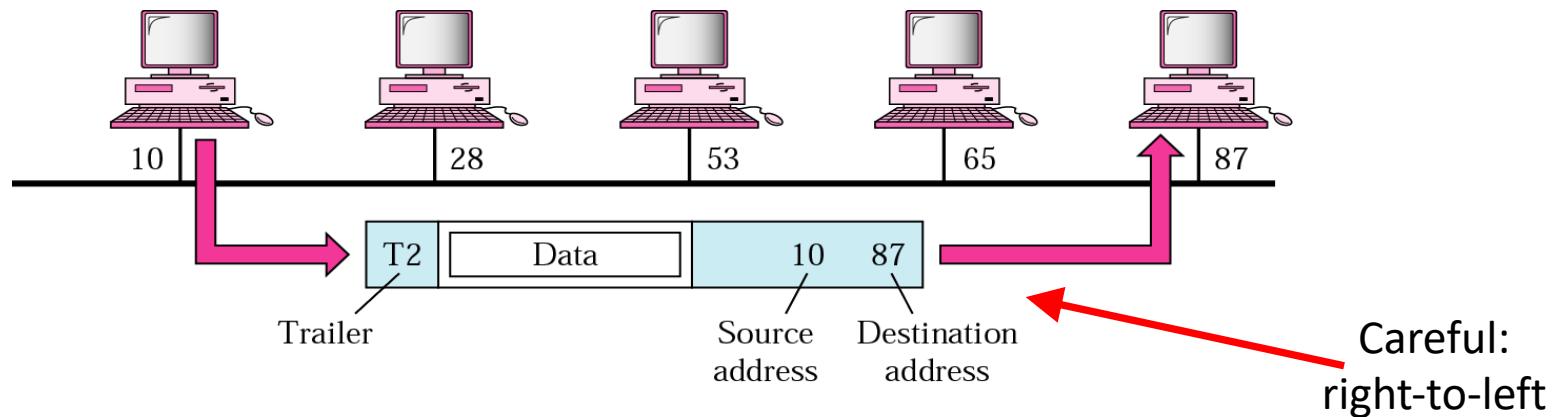
The data link layer is responsible for transmitting frames from one node to the next.

* Figure is courtesy of B. Forouzan

18

Packetizing & Addressing

- Packetizing: Encapsulating data in frame or cell i.e. adding header and trailer
- Addressing: Determining the address of the next hop (LANs) or the virtual circuit address (WANs)



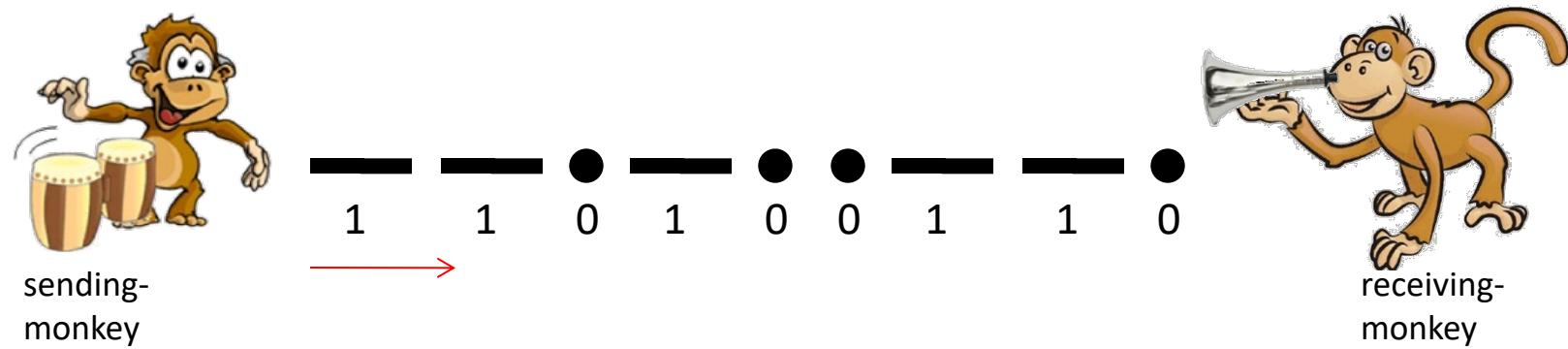
* Figure is courtesy of B. Forouzan 19

Communication

- What is telecommunication really about?

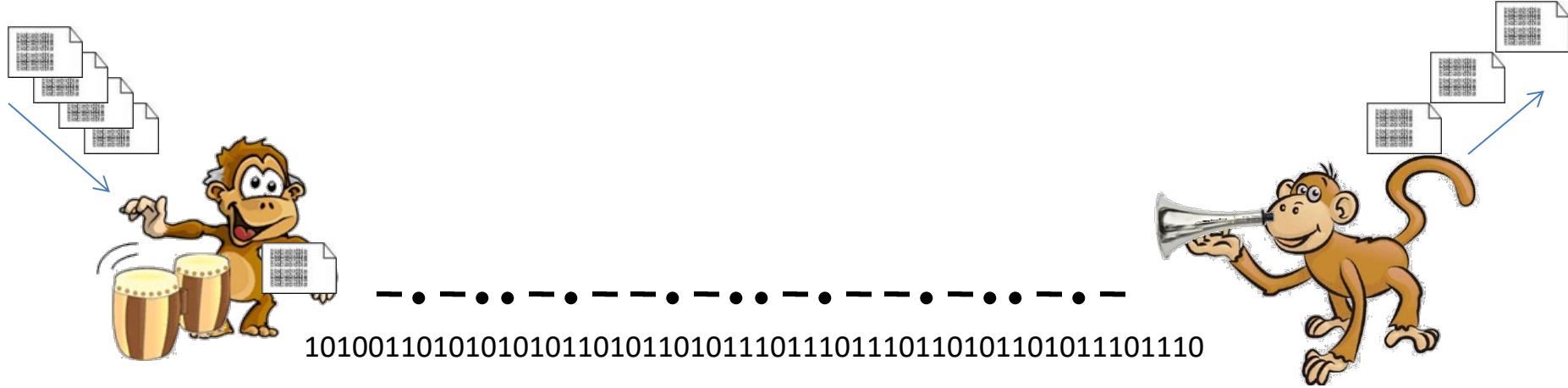
Communication

- What is telecommunication really about?



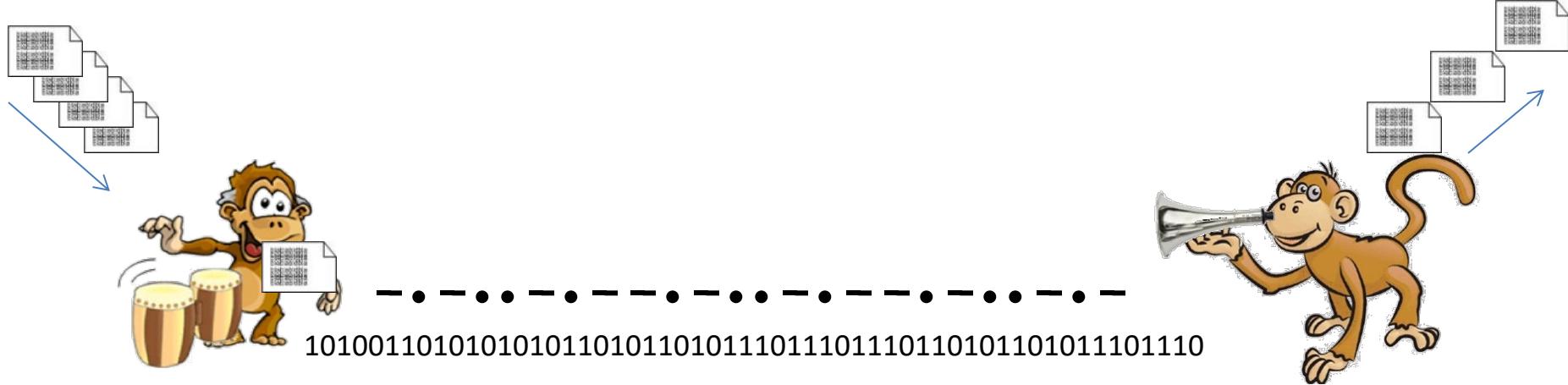
Bongo-Playing Monkeys Doing Morse-code!

Framing



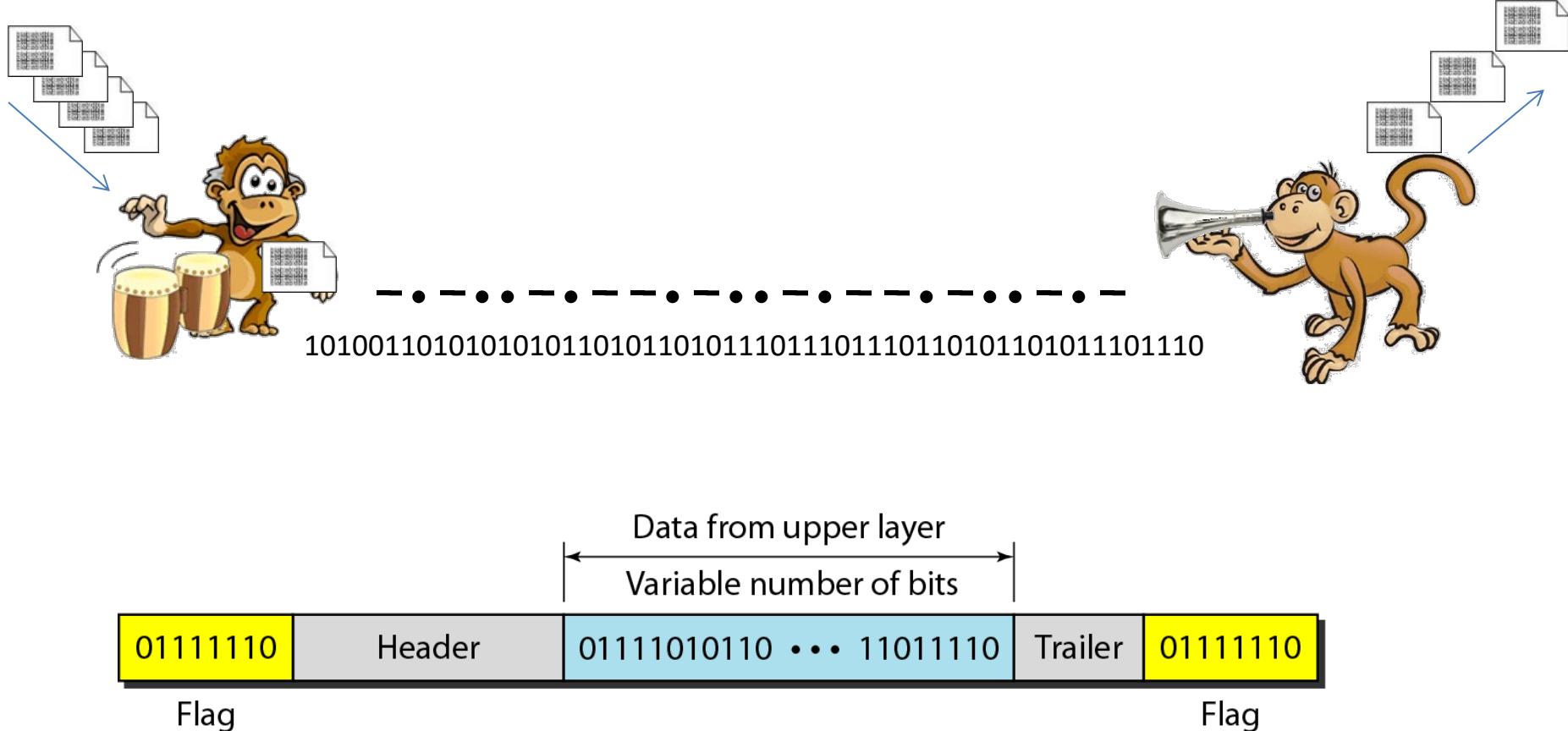
- Spews out signals at a furious rate

Framing



- Spews out signals at a furious rate
- How does receiving-monkey know when a unit of information begins?

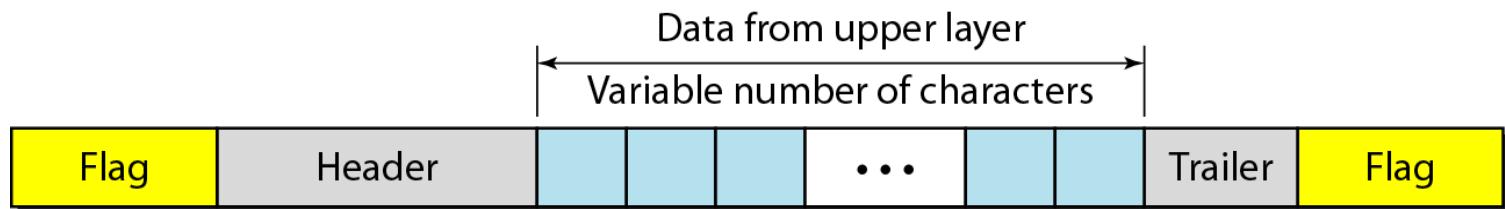
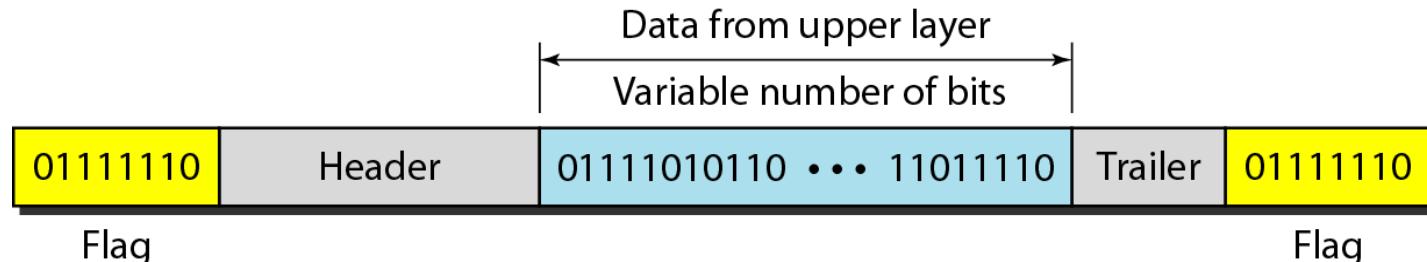
Framing



* Figure is courtesy of B. Forouzan 24

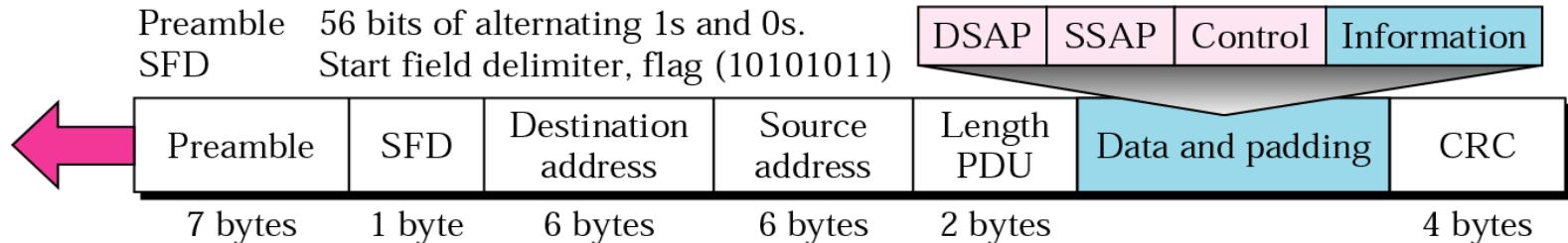
Bit- & Byte-Oriented Protocols

- Two Variations



* Figure is courtesy of B. Forouzan 25

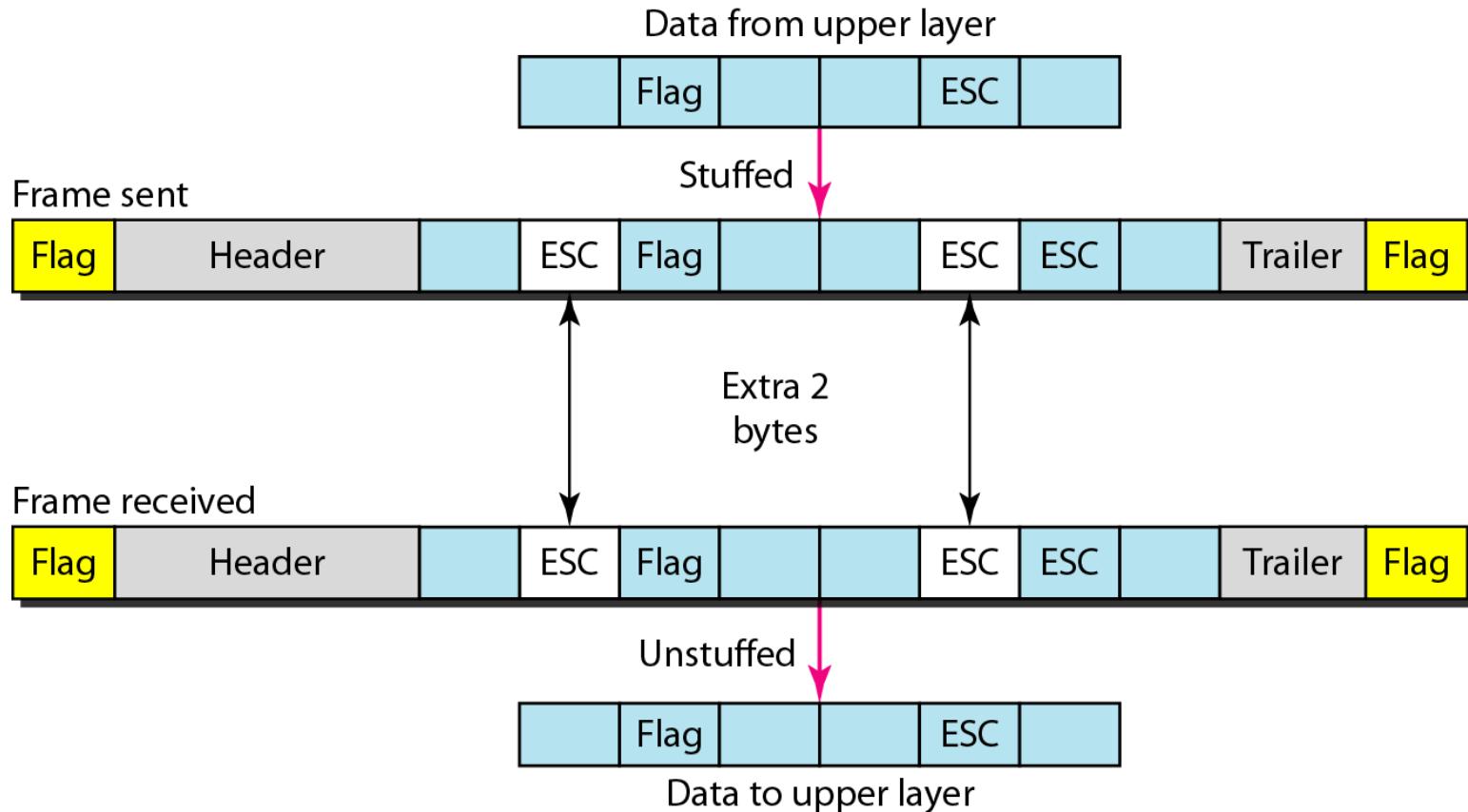
Example: 802.3 MAC Format



- 64-bit frame preamble (10101010) used to synchronize reception
 - 7 bit preamble (10101010) + 1 start flag (10101011)
- Maximum frame length: 1518 bytes
 - ⇒ max 1500 bytes payload
- Minimum frame length: 64 bytes
 - ⇒ min 46 bytes payload

* Figure is courtesy of B. Forouzan

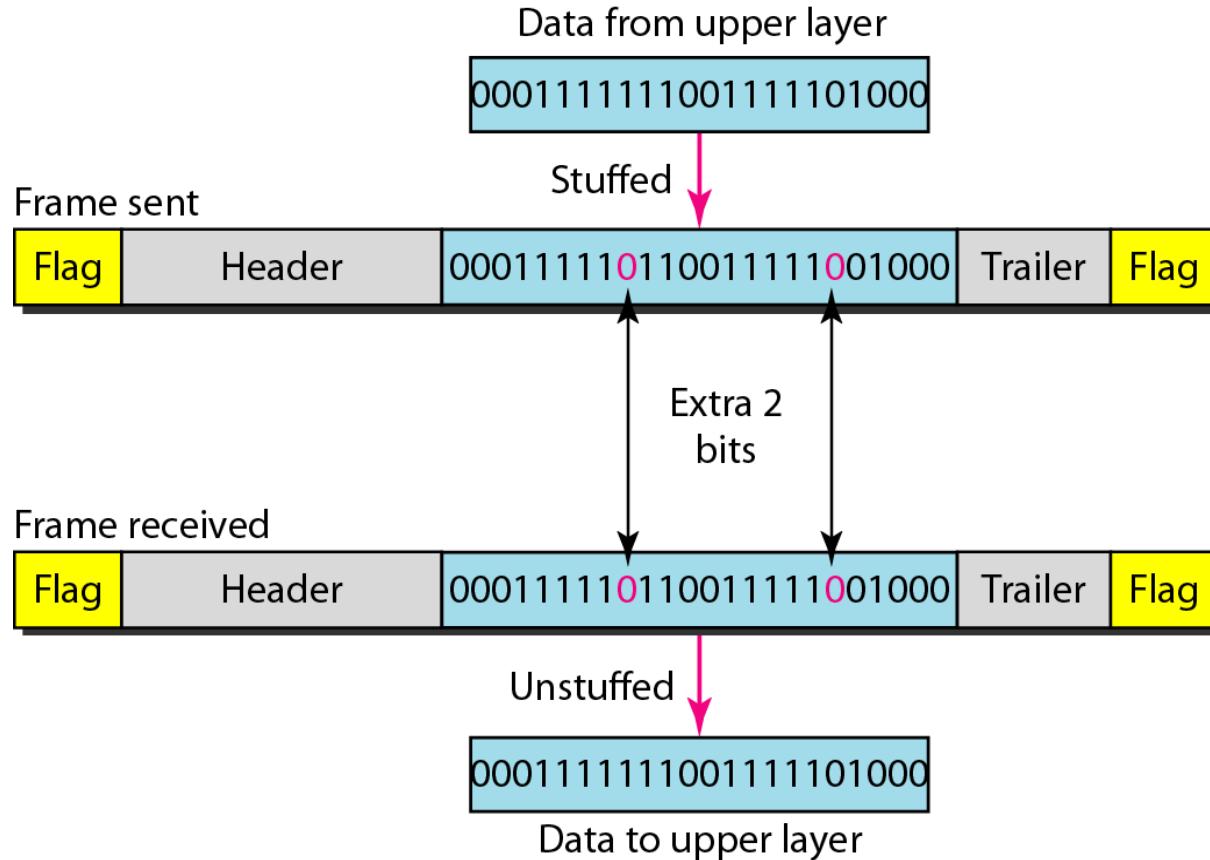
Byte Stuffing



Process of adding 1 extra byte whenever there is a flag or escape character in the text.

* Figure is courtesy of B. Forouzan 27

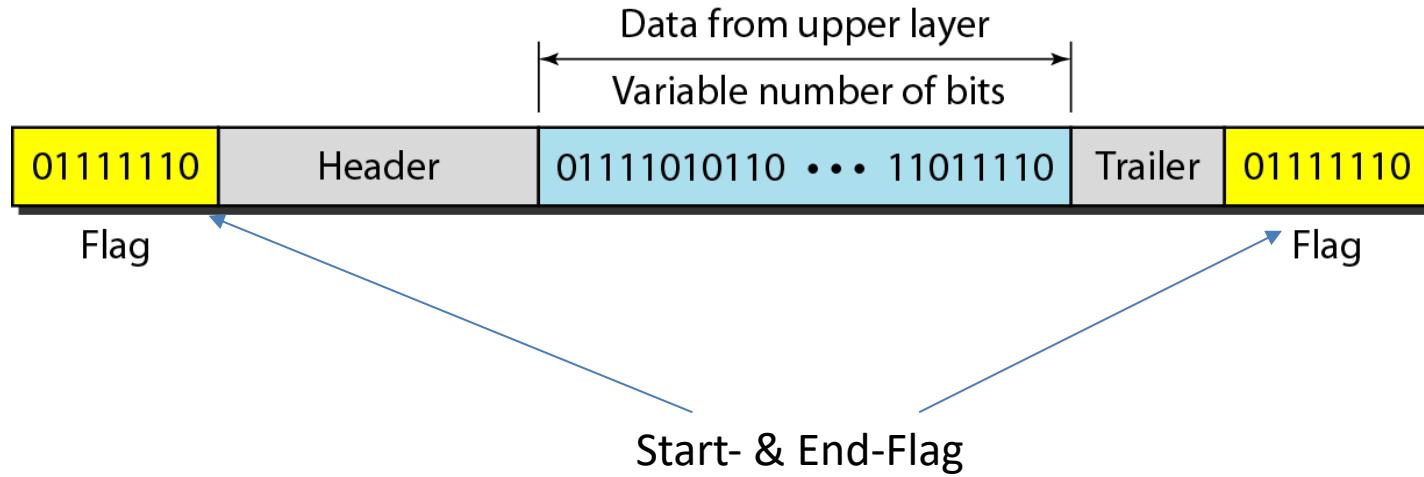
Bit Stuffing



Process of adding an extra 0 whenever five consecutive 1s follow a 0 in the data

* Figure is courtesy of B. Forouzan

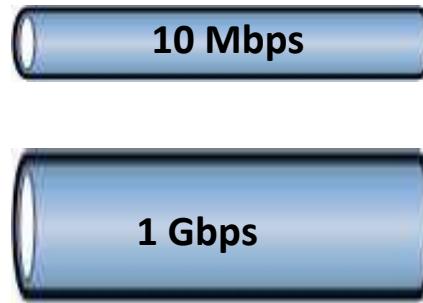
Framing



* Figure is courtesy of B. Forouzan 29

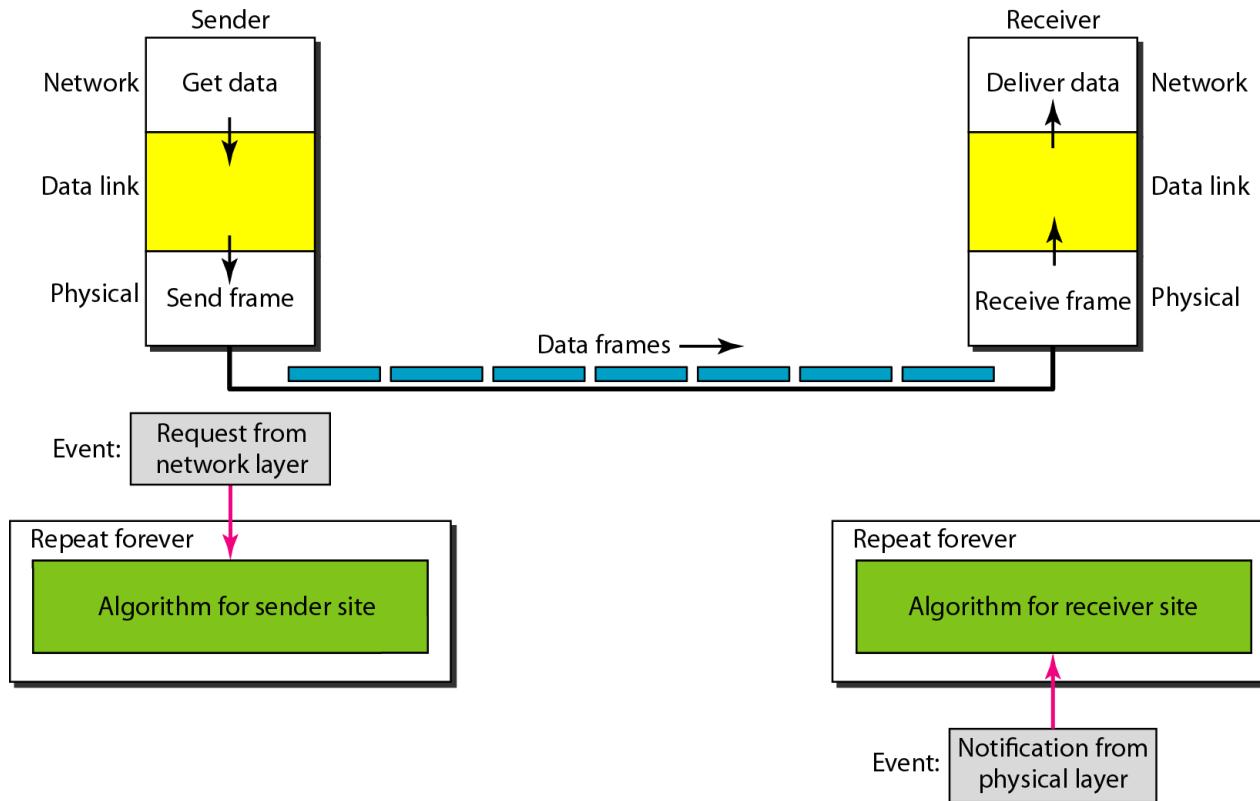
Networking Issues

- Time → Latency
- Amount → Throughput



- Management Information → Overhead
 - **Trade-Off:** May lead to better efficiency
- Overhead vs Payload

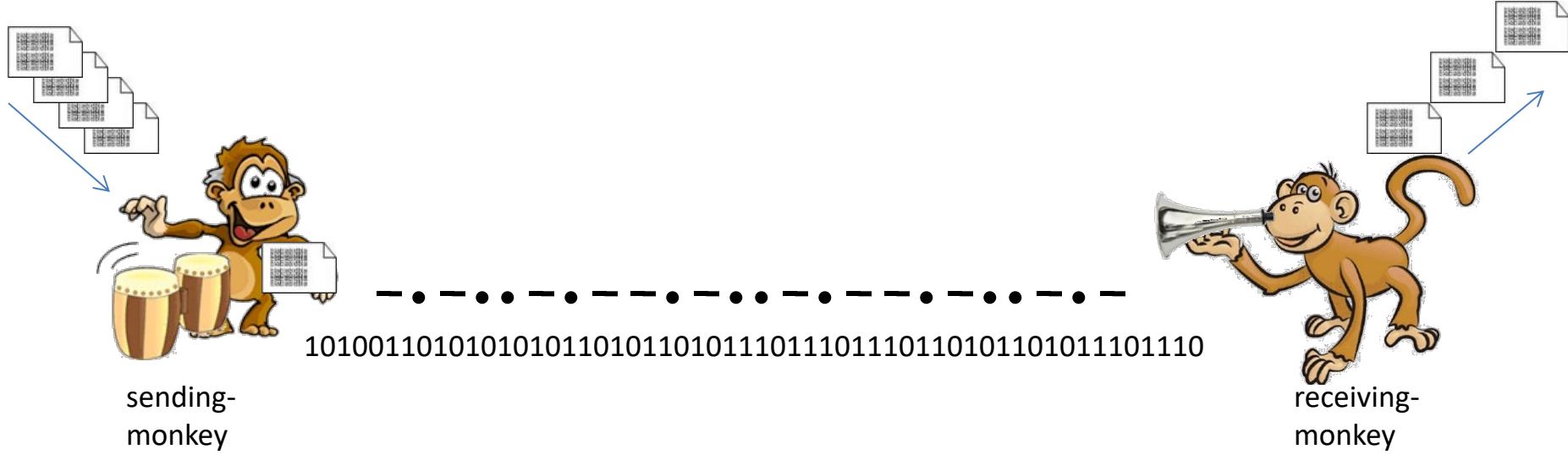
Simplest Protocol



- Hope that the receiver is fast enough!
- No overhead

* Figure is courtesy of B. Forouzan

Flow Control

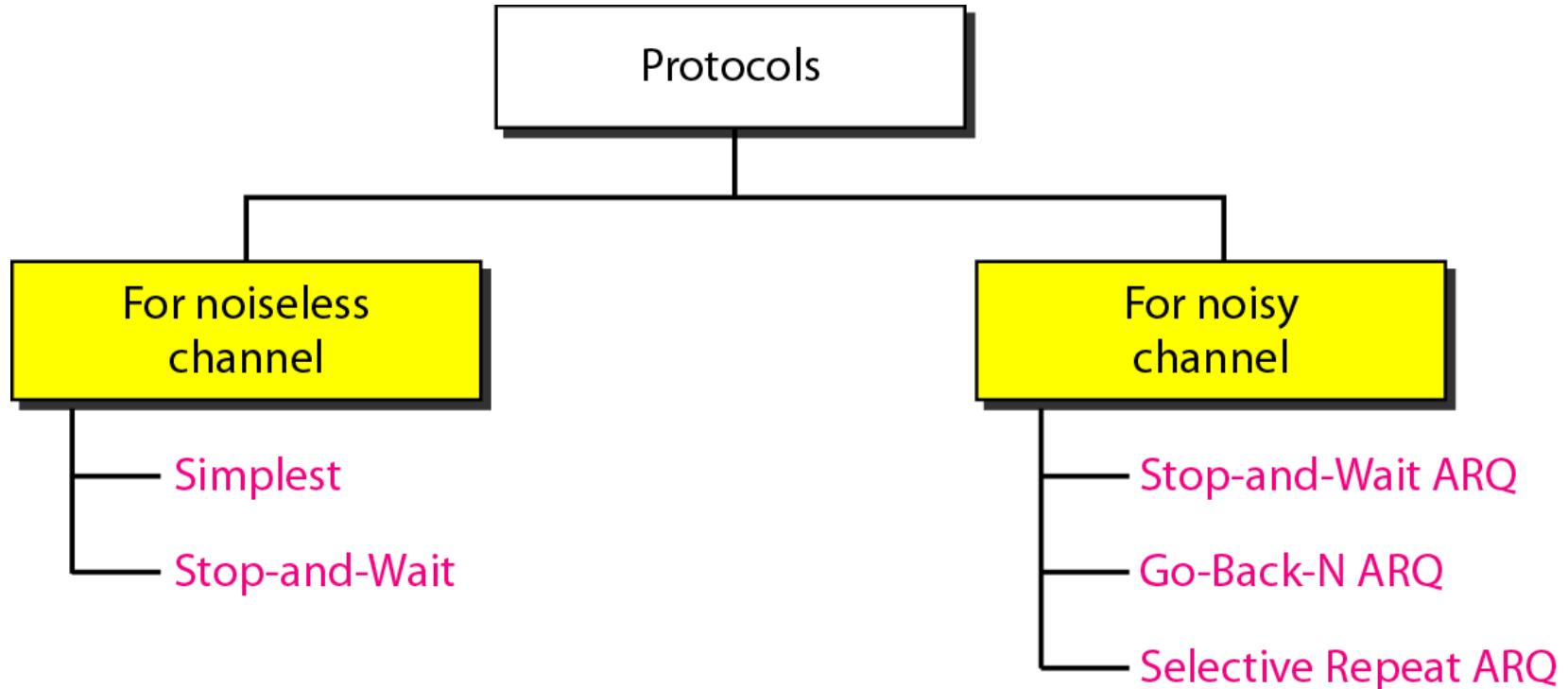


- What happens if sending-monkey can drum faster than receiving-monkey can write?

Flow Control

- Forouzan's Definition: Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.
- “My” Definition: Flow Control refers to the control of the amount of data that a sender can transmit without overflowing the receiver.

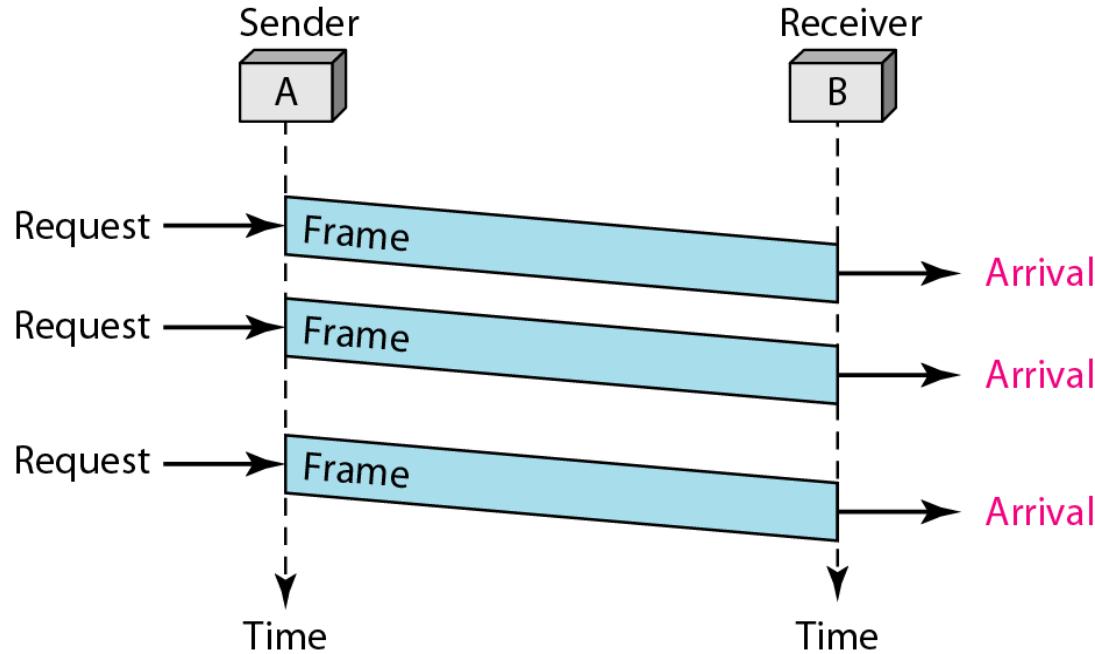
Flow Control Protocols



* Figure is courtesy of B. Forouzan 34

Simplest Protocol: Flow Diagram

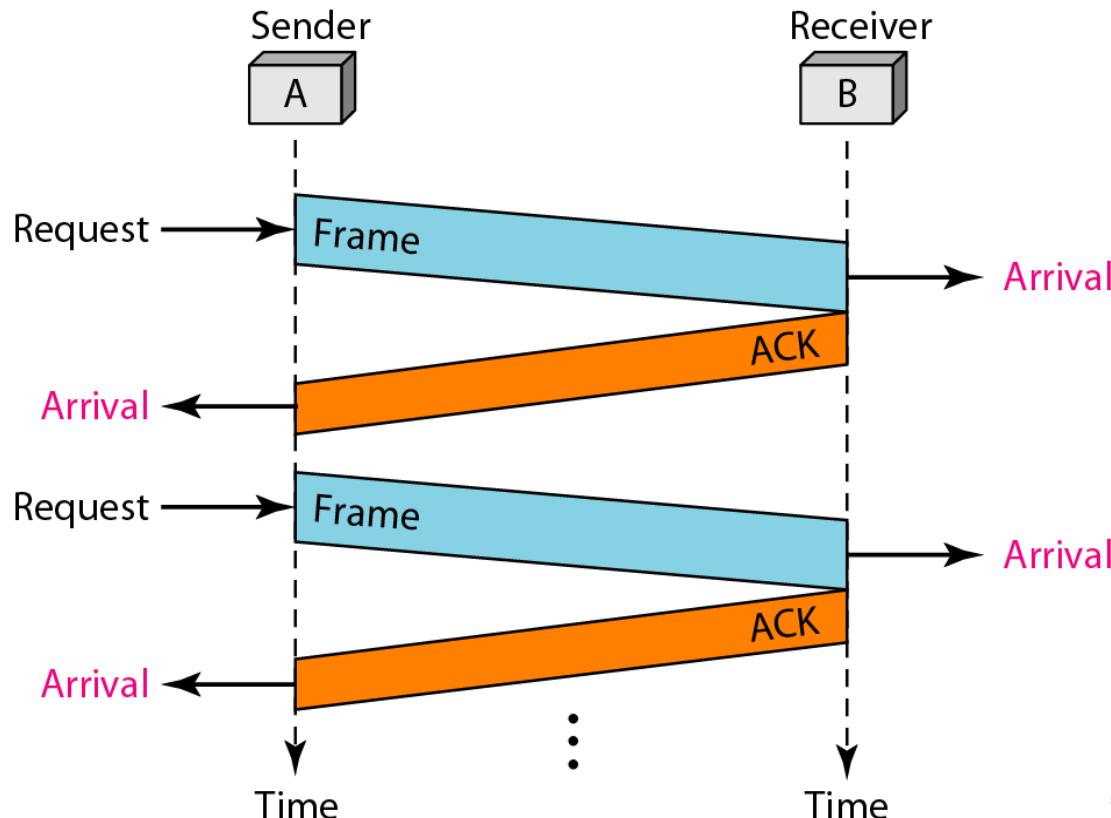
- Sender sends frames as fast as data arrives
- Receiver receives all data sent



* Figure is courtesy of B. Forouzan 35

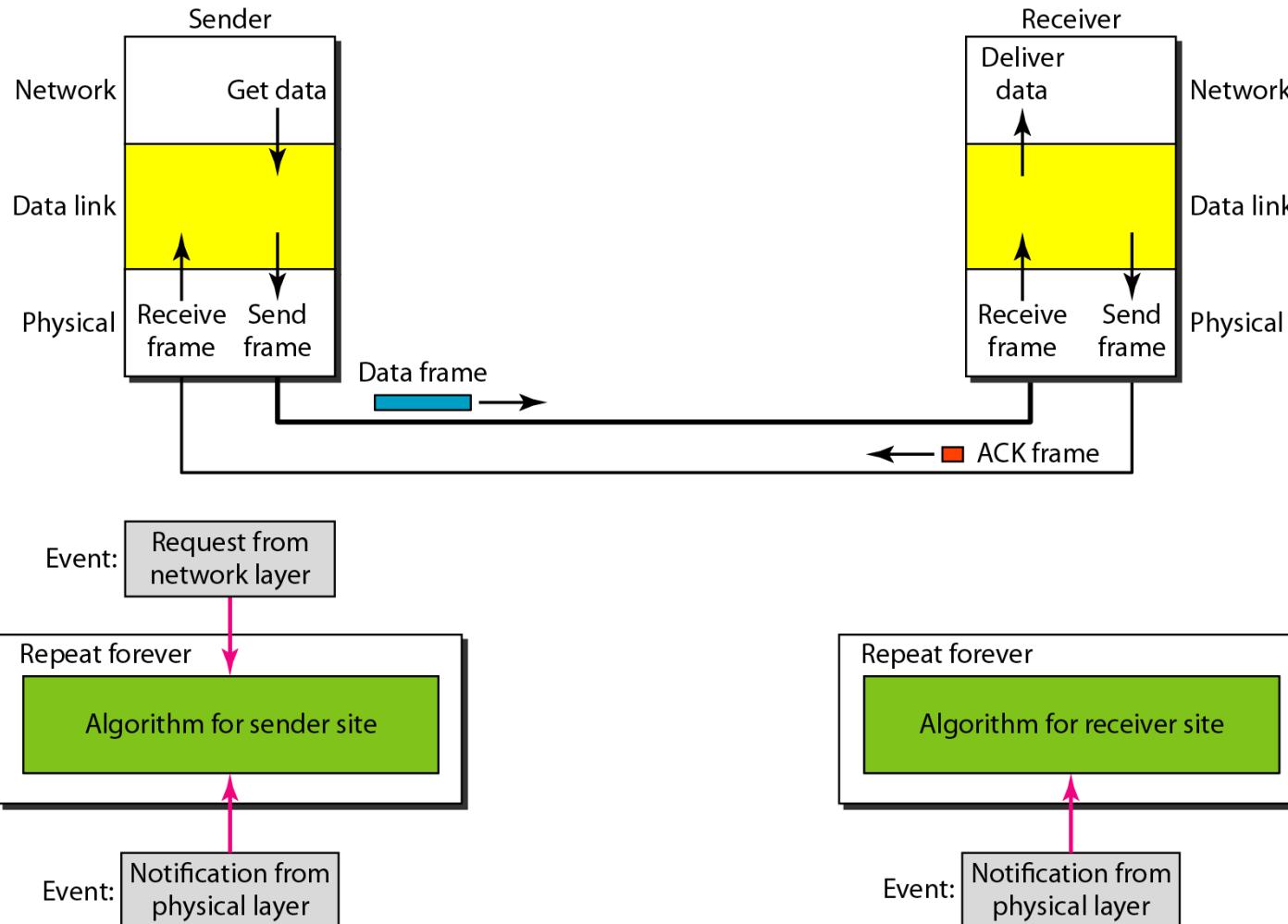
Stop-and-Wait: Flow Diagram

- Sender sends frame and waits for ACK
- Receiver replies to received frame with ACK



* Figure is courtesy of B. Forouzan

Stop-and-Wait Protocol



* Figure is courtesy of B. Forouzan

Error Control

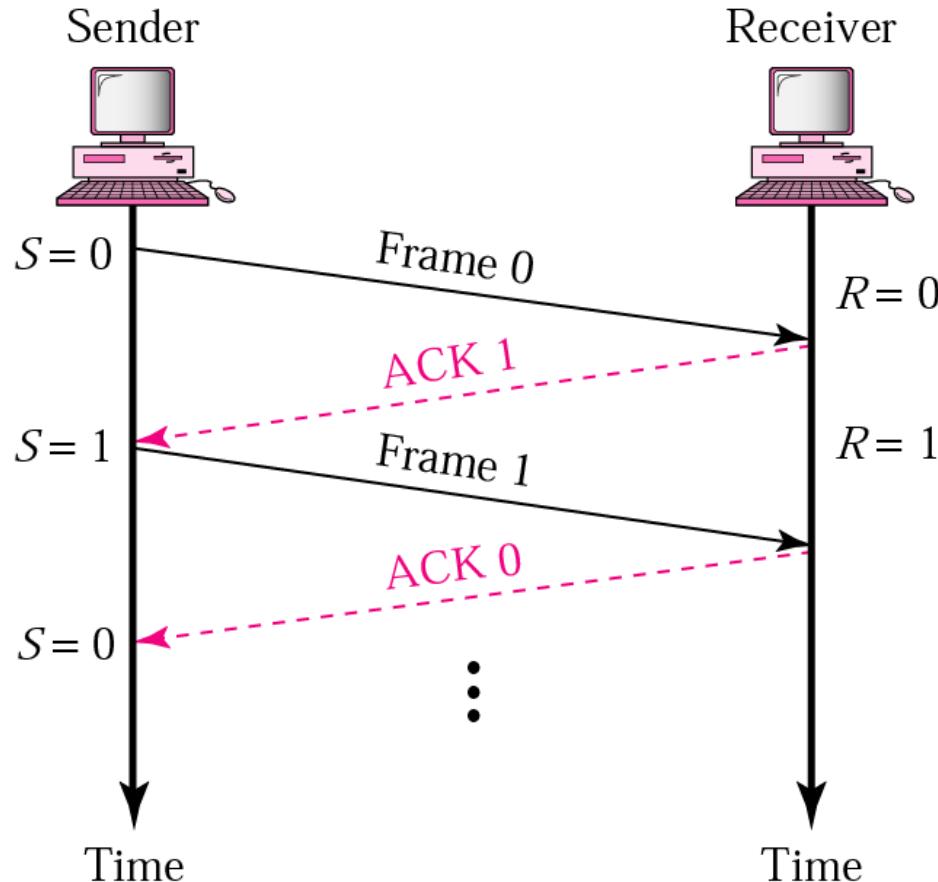
- Frames may get lost or corrupted
 - Incorrect checksum, CRCs, etc
- Error control need to ensure retransmission
- Error Control Protocols:
 - Stop-and-Wait ARQ*
 - Go-back-N ARQ
 - Selective Repeat ARQ

*ARQ = Automatic Repeat Request

Ingredients for Error Control

- Error detection
- Positive acknowledgement
 - Receiver returns positive ACK for received, error-free frames
- Retransmission after timeout
 - Sender retransmit packet after given time
- Negative acknowledgement and retransmission
 - Receiver returns negative ACK - or NACK - for packets with errors

Stop-and-Wait ARQ

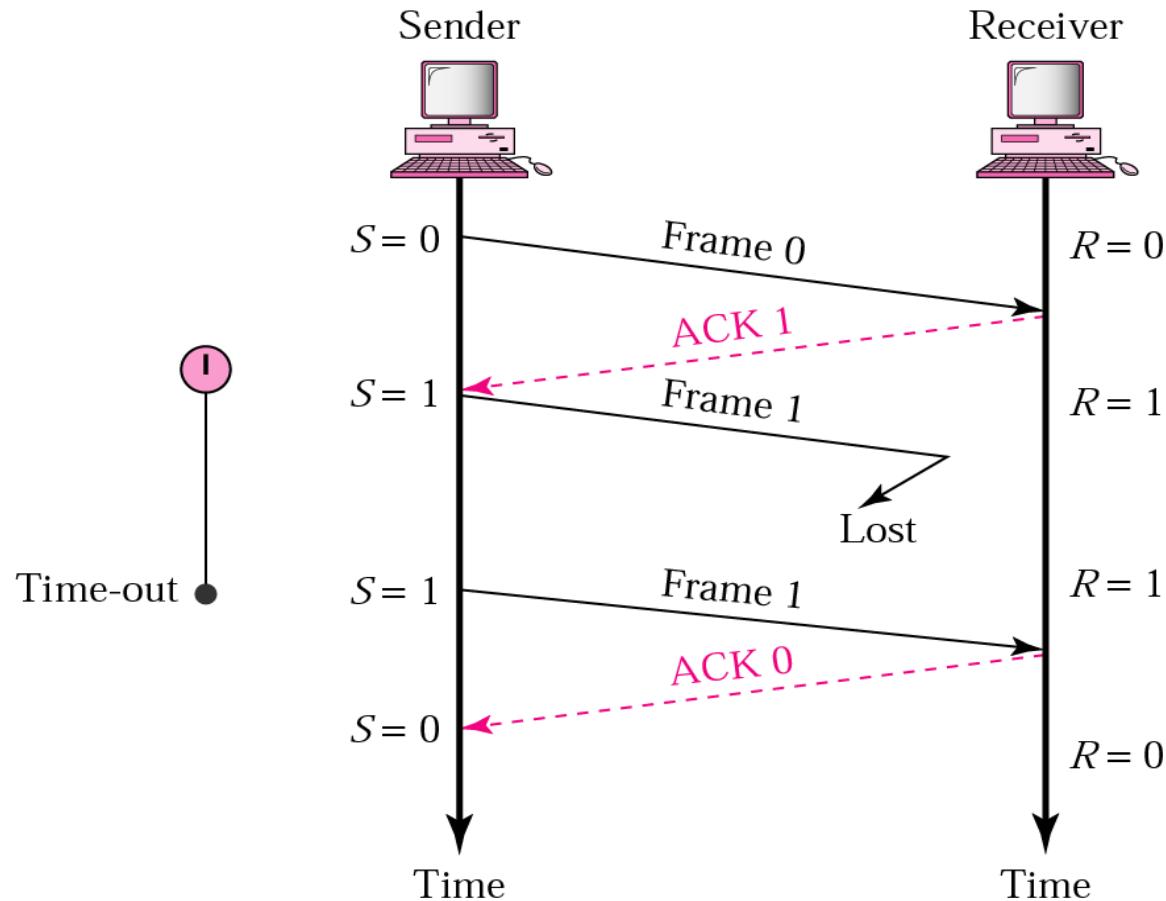


- ACK = received packet, ready to receive packet #
- ARQ = Automatic Repeat Request

* Figure is courtesy of B. Forouzan 40

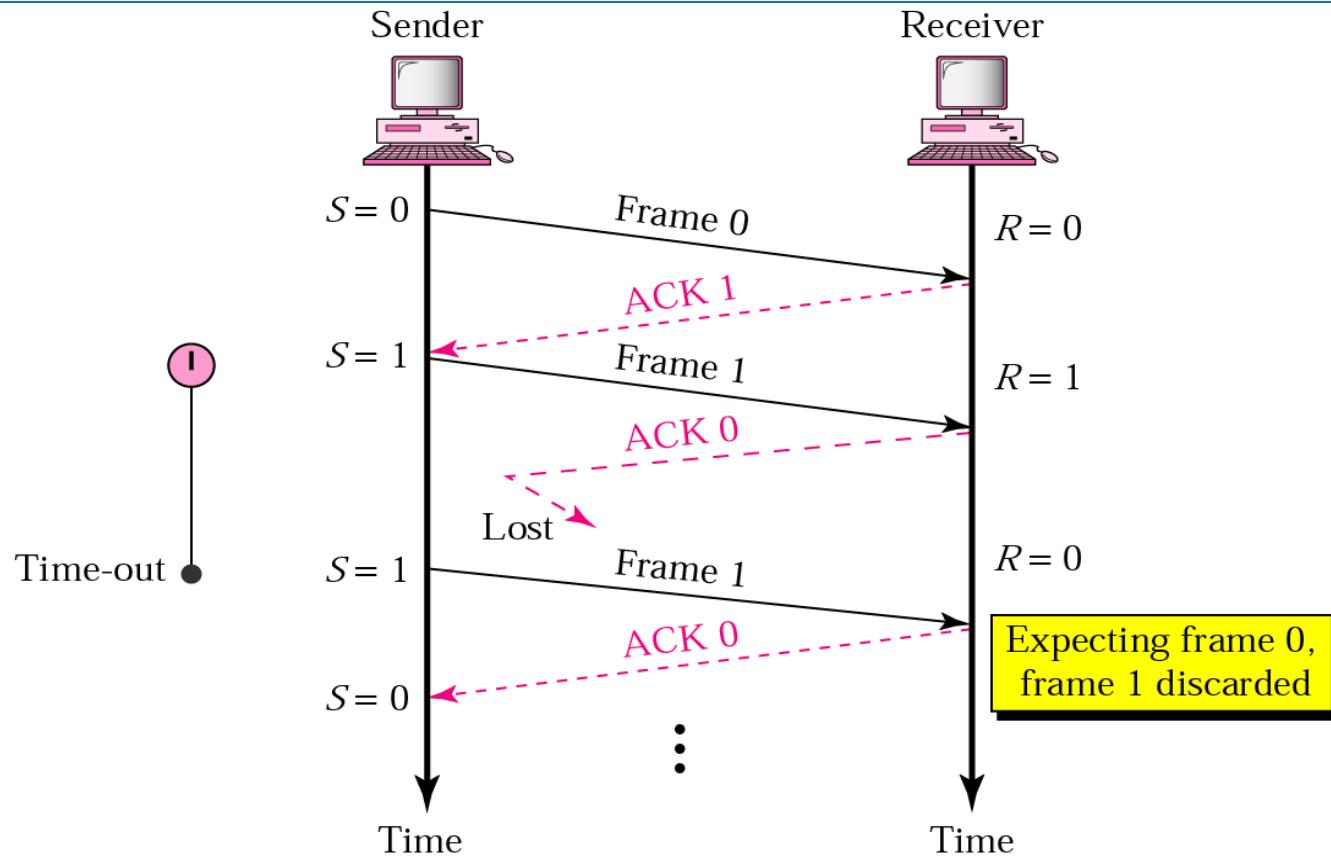
Stop-and-Wait ARQ: Time-Out

- Frame is lost during transmission



* Figure is courtesy of B. Forouzan 41

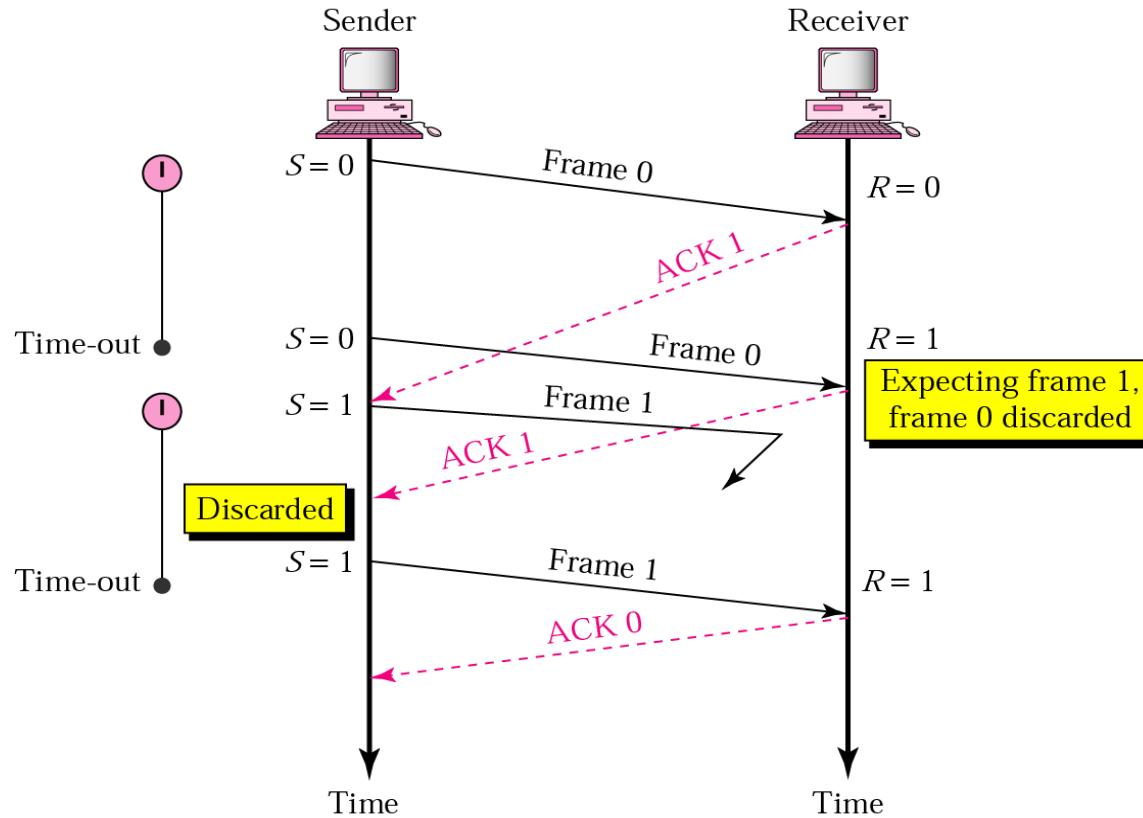
Stop-and-Wait ARQ: Lost-ACKs



- Numbering frames prevents retaining duplicate frames
- Every received frame is acknowledged

* Figure is courtesy of B. Forouzan 42

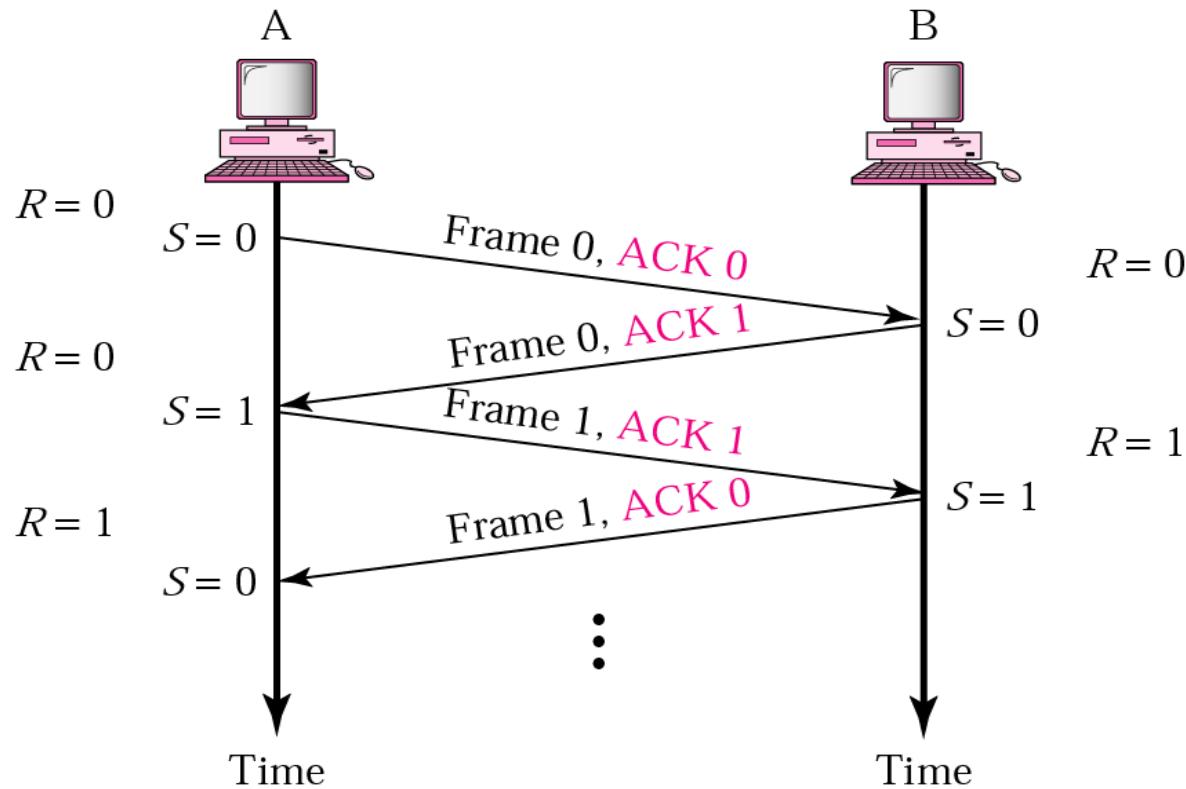
Stop-and-Wait ARQ: Delayed ACK



Numbered acknowledgments are needed if an acknowledgment is delayed and the next frame is lost.

* Figure is courtesy of B. Forouzan

Piggybacking ACKs

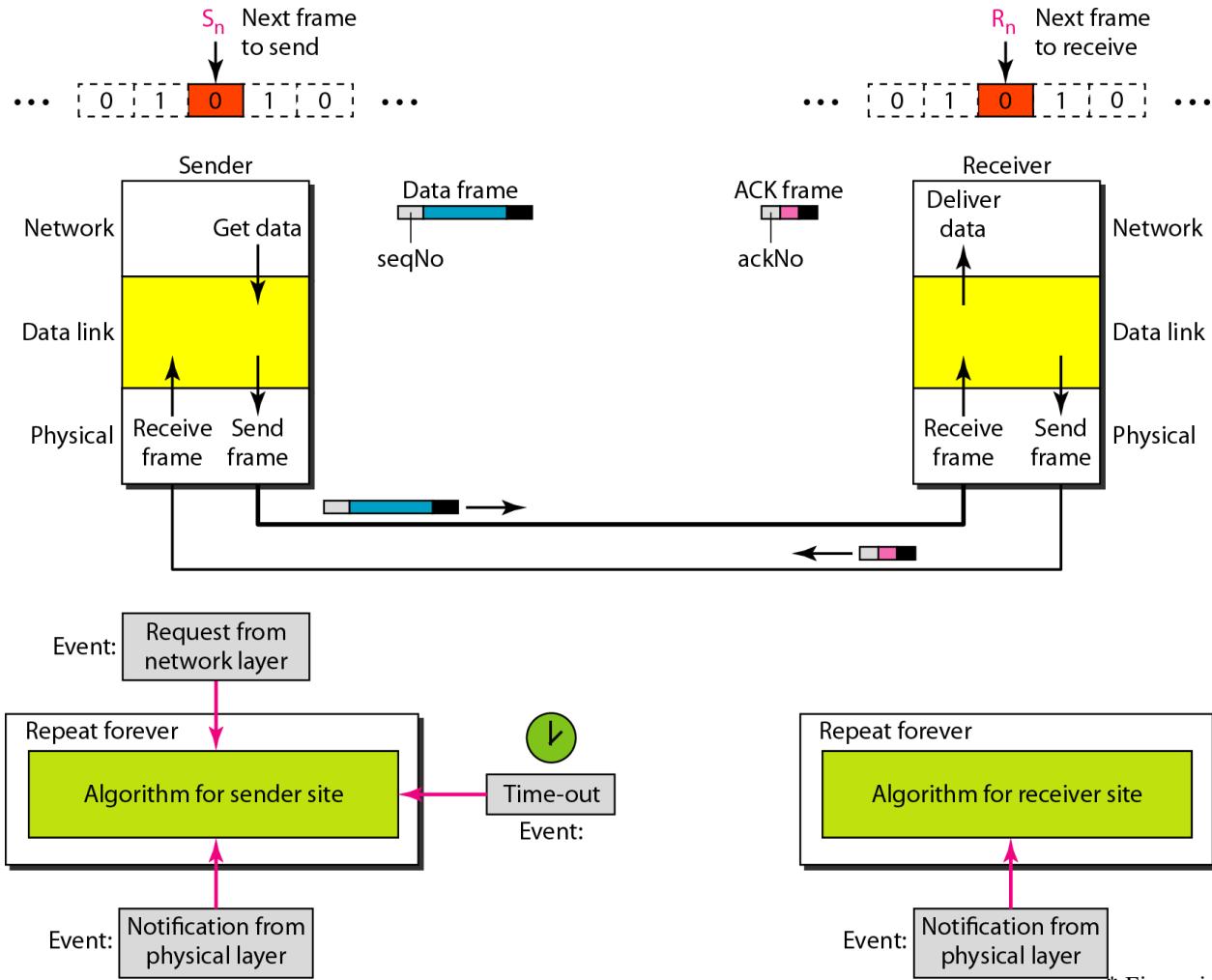


Next data frame send carries
the acknowledgement for the last frame received

* Figure is courtesy of B. Forouzan

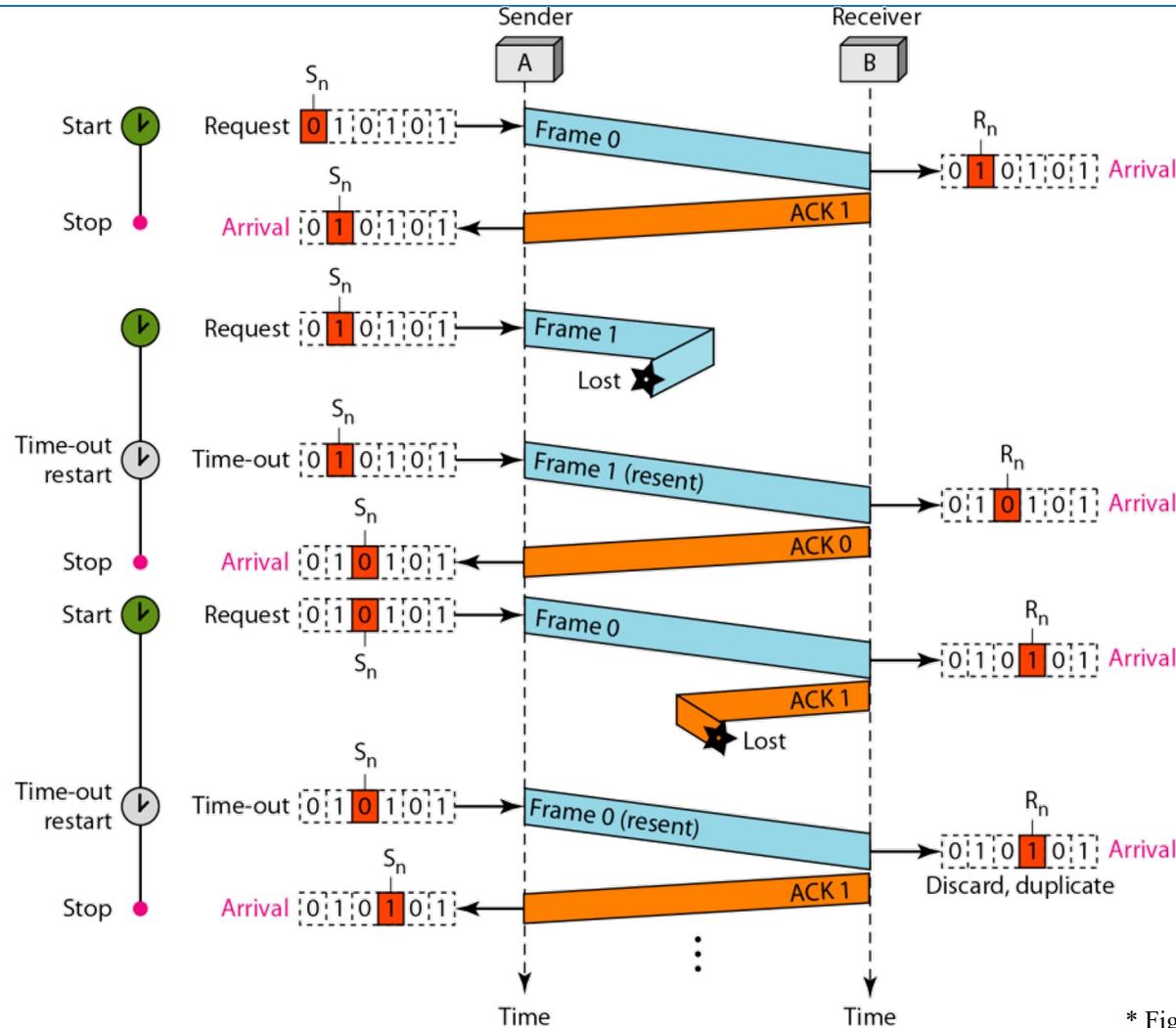
44

Stop-and-Wait ARQ



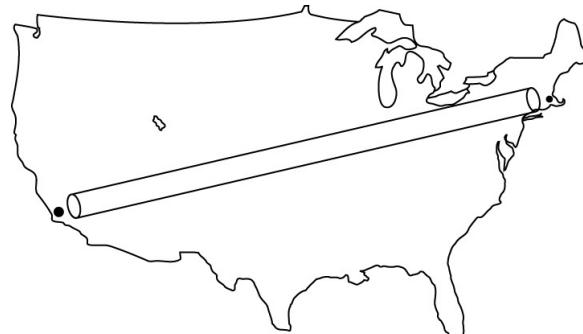
* Figure is courtesy of B. Forouzan

Stop-and-Wait ARQ: Flow Diagram

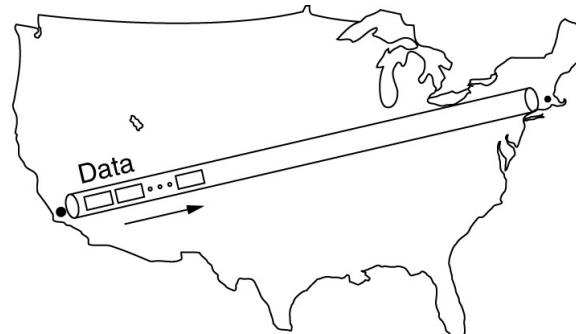


* Figure is courtesy of B. Forouzan

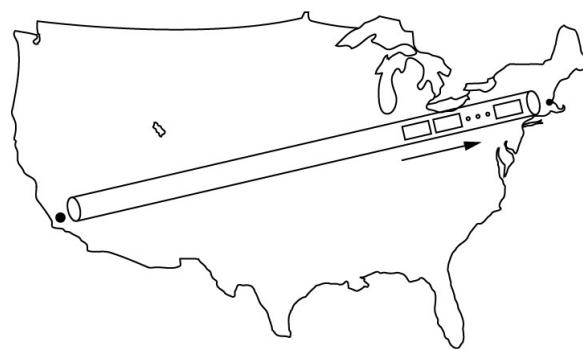
Round Trip Time



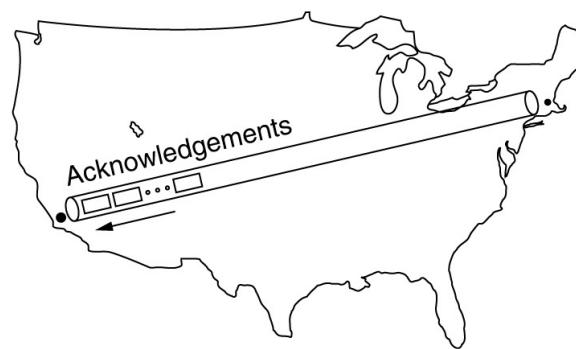
(a)



(b)



(c)



(d)

(a) At $t = 0$ (b) After 500 μ sec (c) After 20 msec (d) after 40 msec

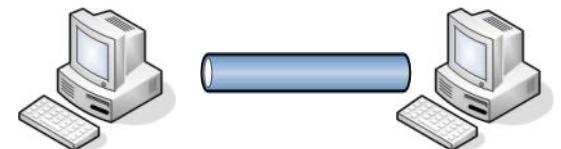
* Figure is courtesy of A. Tanenbaum 47

Flow Control

- Definitions
 - Transmission time
 - Time taken to emit all bits onto the medium
 - Proportional to length of frame
 - Propagation time
 - Time for a bit to traverse the link

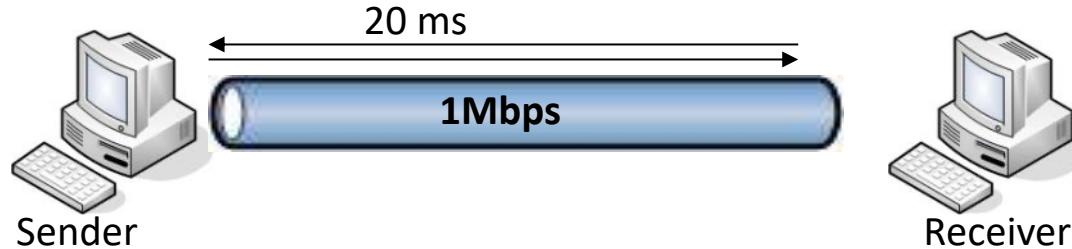
Bandwidth-Delay Product

- Bandwidth:
 - Size of the pipe
 - Determines how much data can be send
- Round-Trip Time (RTT)
 - Determines how long an ACK takes
- High Bandwidth (big pipe)
 - Lots of data can be send
- Depending on RTT
 - Sender may exhaust window quickly
- $\text{Bandwidth} \times \text{RTT}$
 - Gives indication of amount of data that can be send while waiting for ACK



Delay Before Receiving ACK

- Communication link with 1Mb/s
- Round-Trip time: $20 \text{ ms} = 20 * 10^{-3} \text{ s}$

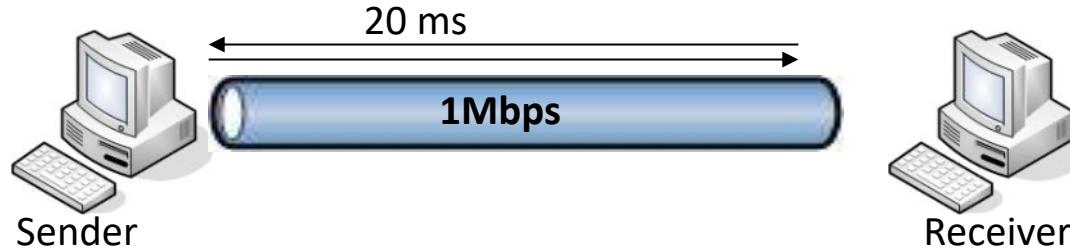


- How much data can you send during the time it takes for 1 bit e.g. an ACK to arrive at the sender:

$$20 * 10^{-3} \text{ s} * 1 * 10^6 \text{ b/s} = 20.000 \text{ bits}$$

Delay Before Receiving ACK

- Communication link with 1Mb/s
- Round-Trip time: $20 \text{ ms} = 20 * 10^{-3} \text{ s}$

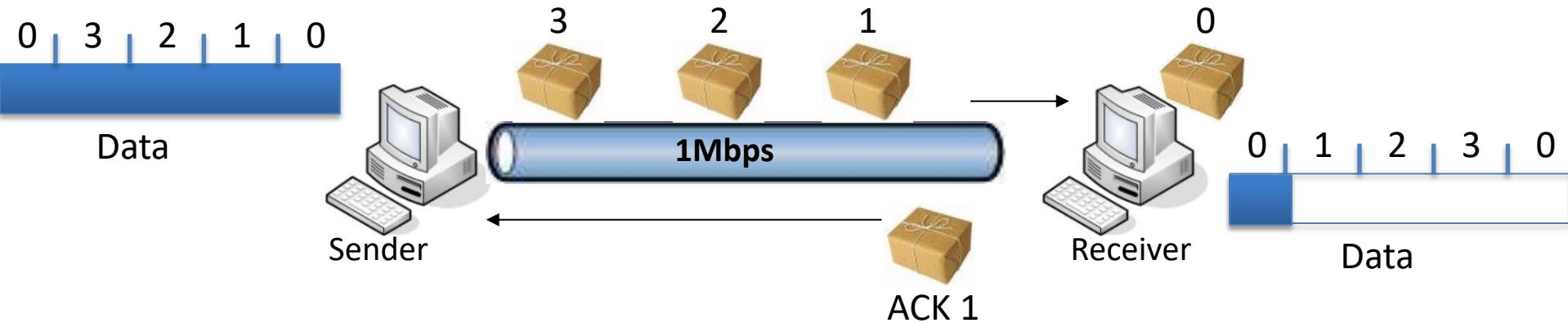


- How much data can you send during the time it takes for 1 bit e.g. an ACK to arrive at the sender:

$$20 * 10^{-3} \text{ s} * 1 * 10^6 \text{ b/s} = 20,000 \text{ bits}$$

- Frame of 2000 bit \Rightarrow 10% of bandwidth used

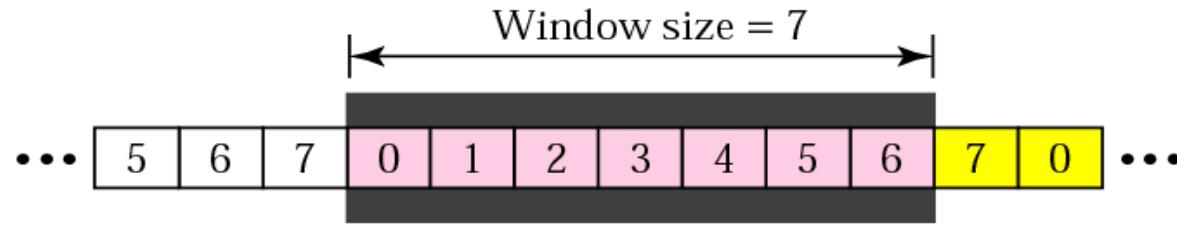
Ideal Solution to Filling the Pipe



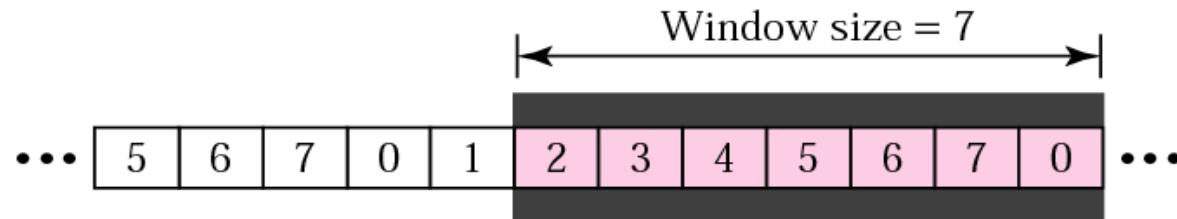
- Allow multiple frames to be in transit
- Receiver has a buffer
- Transmitter can send a number of frames
 - without receiving an ACK
- Each frame is numbered
- ACK includes number of next frame expected

Sliding Window

- m : Size of the sequence number field in bits
- $1 \dots 2^m$: Sequence numbers
- Send window: Box of size $2^m - 1$



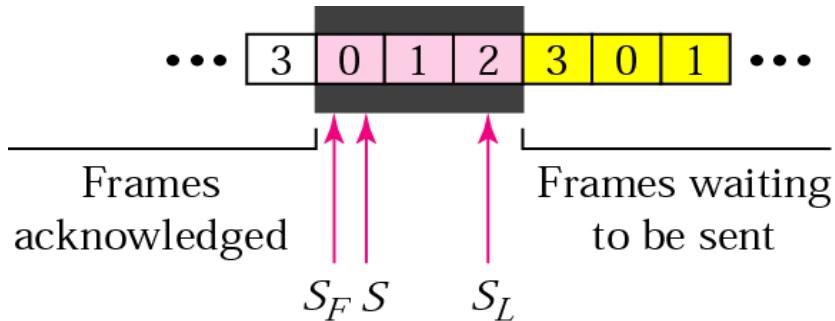
a. Before sliding



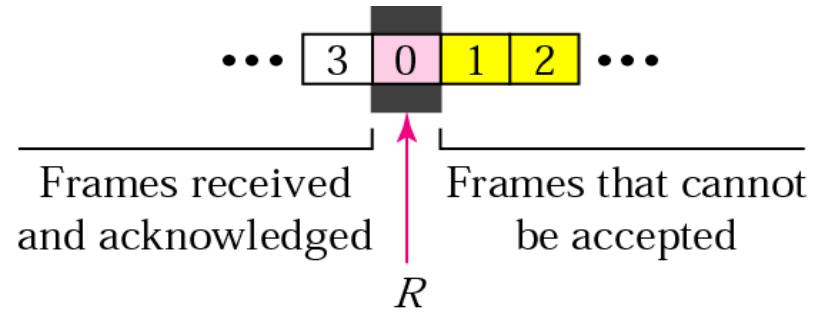
b. After sliding two frames

* Figure is courtesy of B. Forouzan

Go-Back-N ARQ: Control variables



a. Sender window

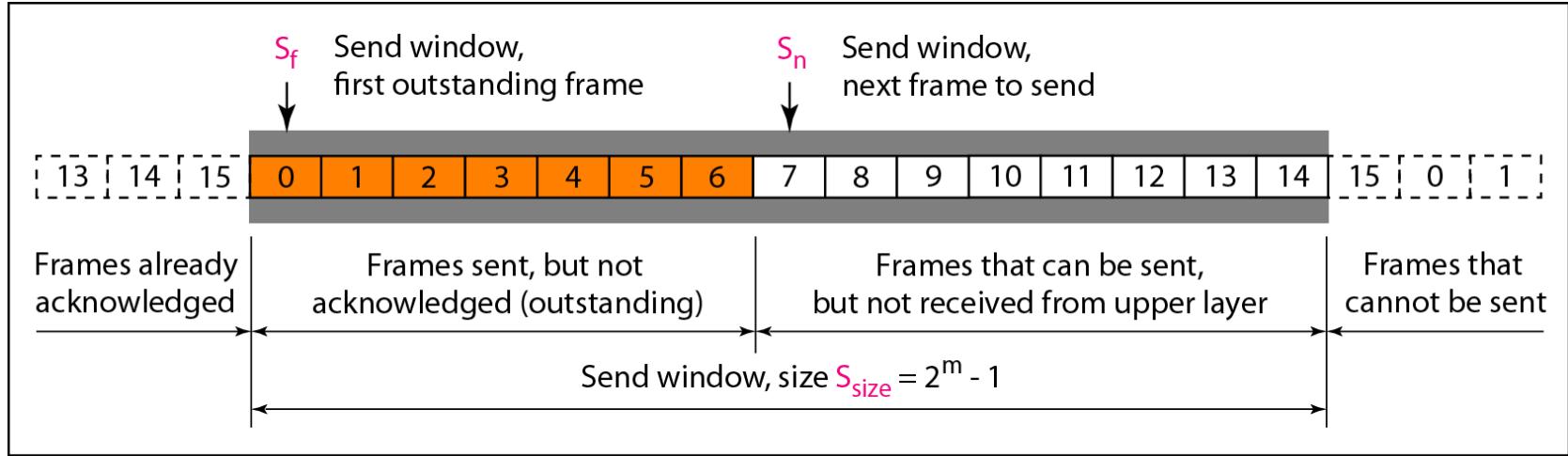


b. Receiver window

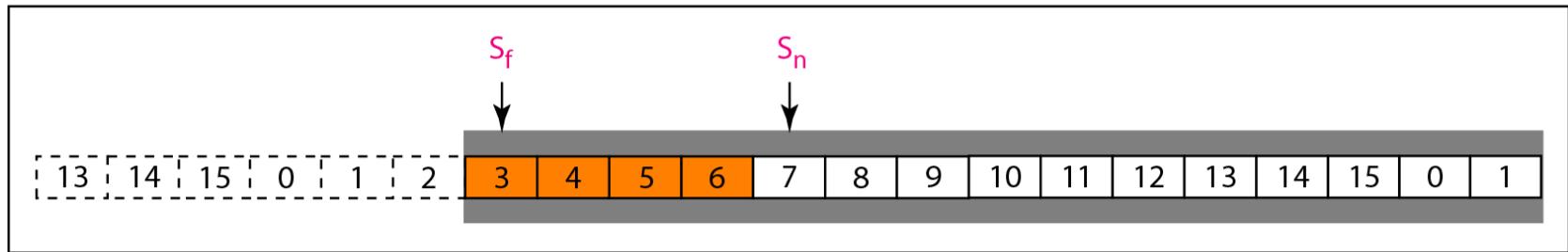
- $R = \# \text{ of recently received frame}$

* Figure is courtesy of B. Forouzan

Sliding Window



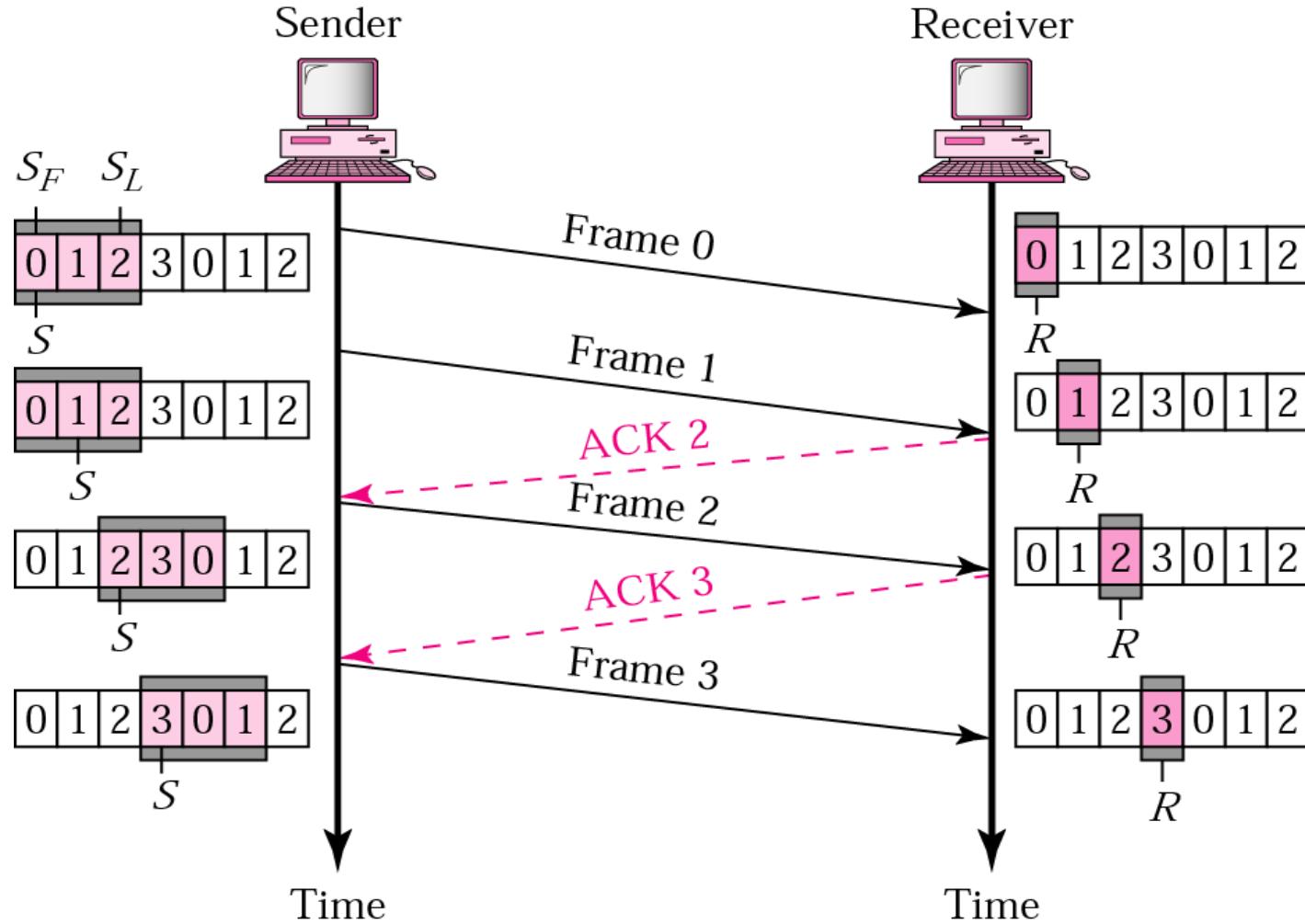
a. Send window before sliding



b. Send window after sliding

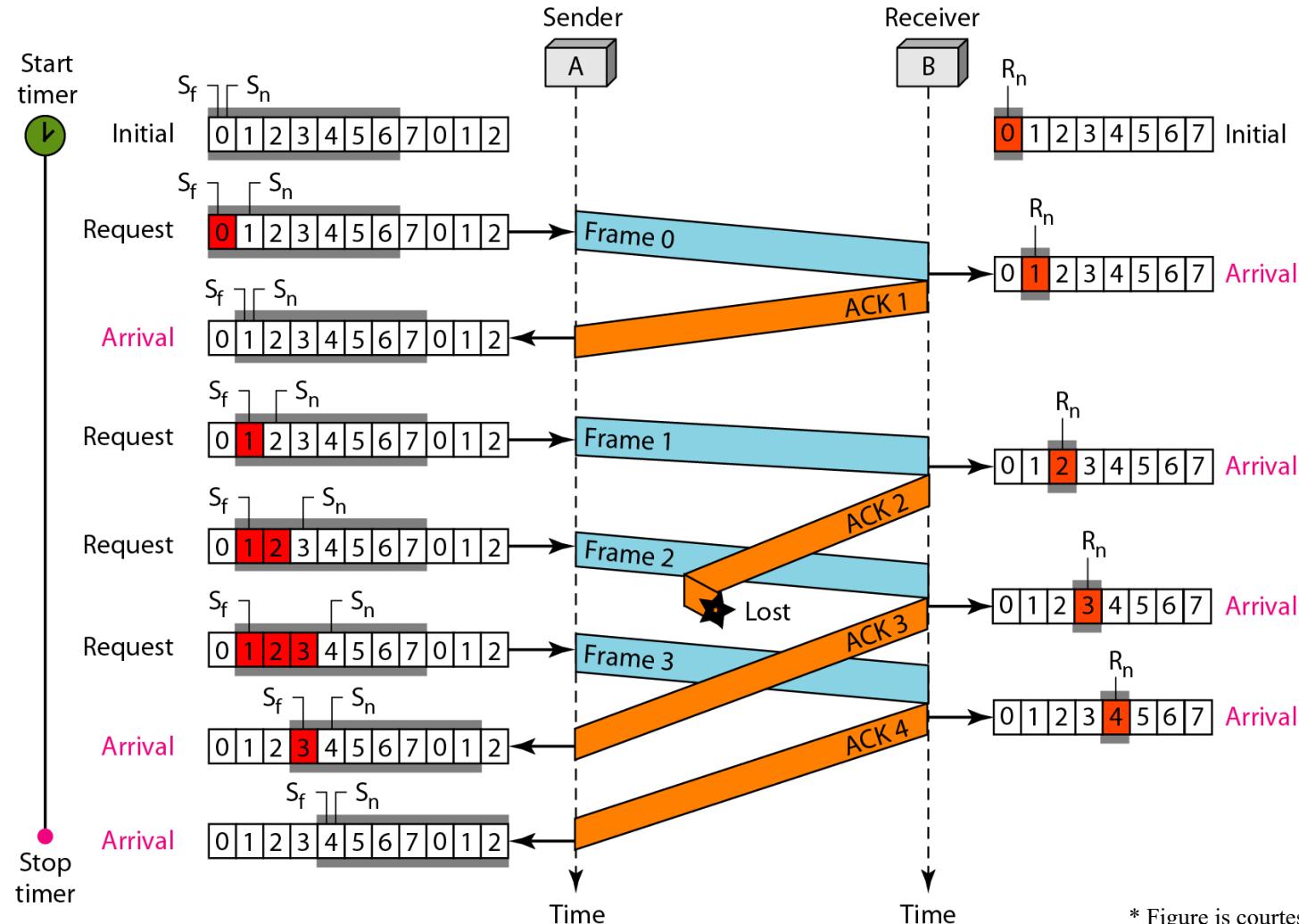
* Figure is courtesy of B. Forouzan

Go-Back-N ARQ



* Figure is courtesy of B. Forouzan

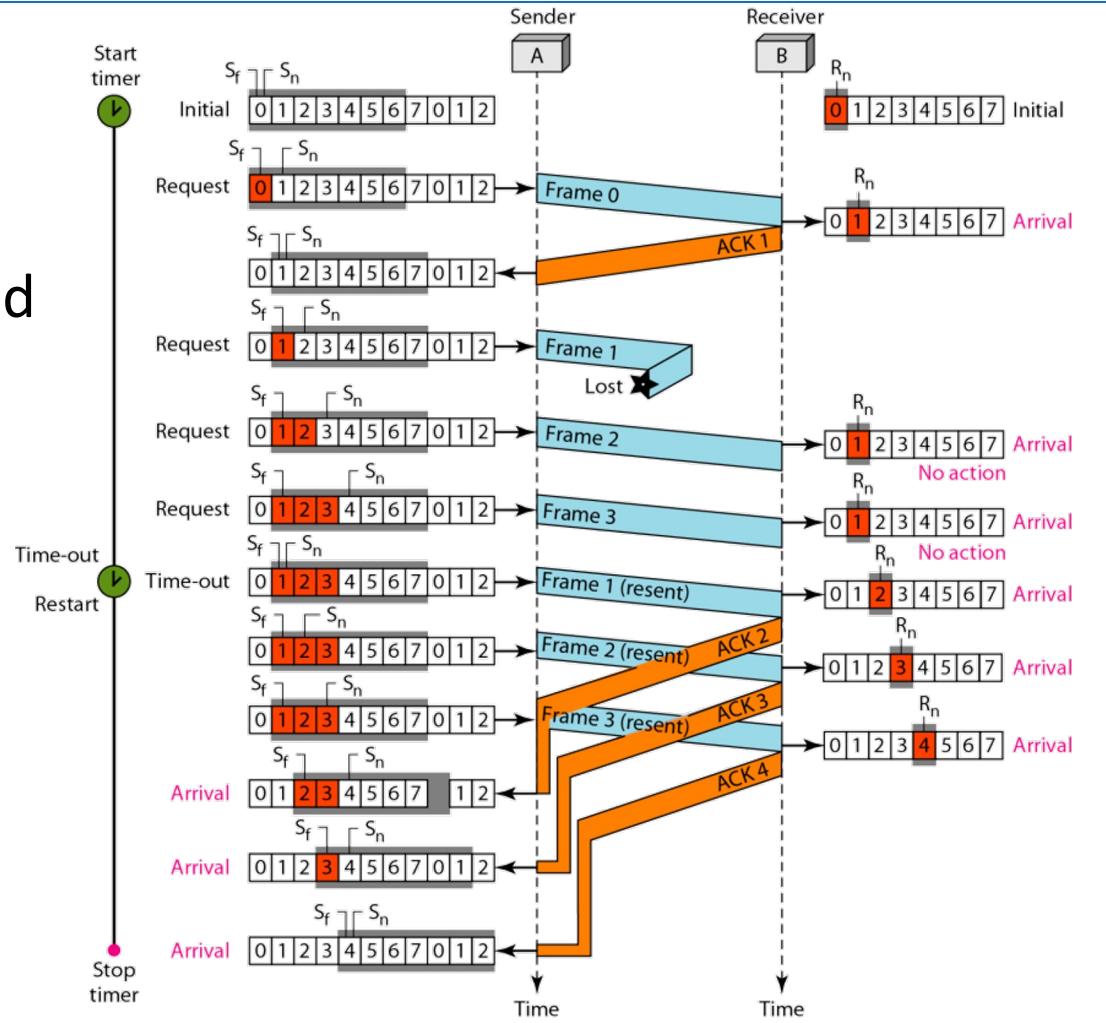
Go-Back-N: Lost ACK + Subs. Ack



* Figure is courtesy of B. Forouzan

Go-Back-N ARQ: Lost Frame!

- Frame 1 lost
- Subsequent frames send
- All frames need to be resend



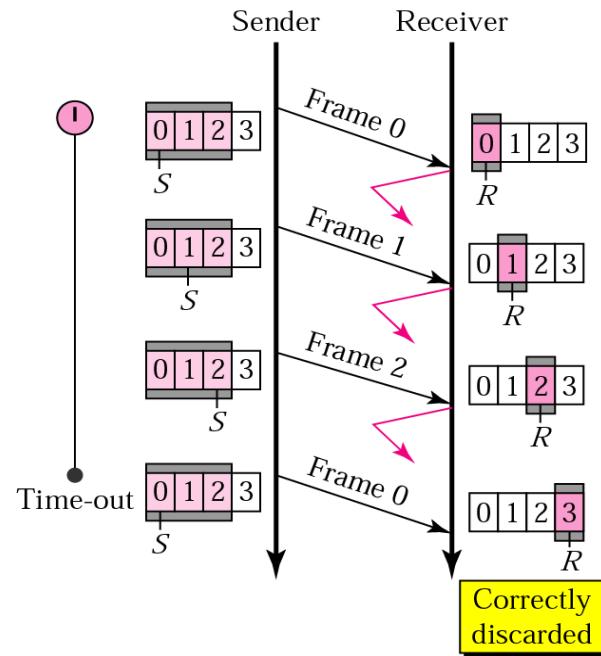
* Figure is courtesy of B. Forouzan

Window Size for Go-Back-N

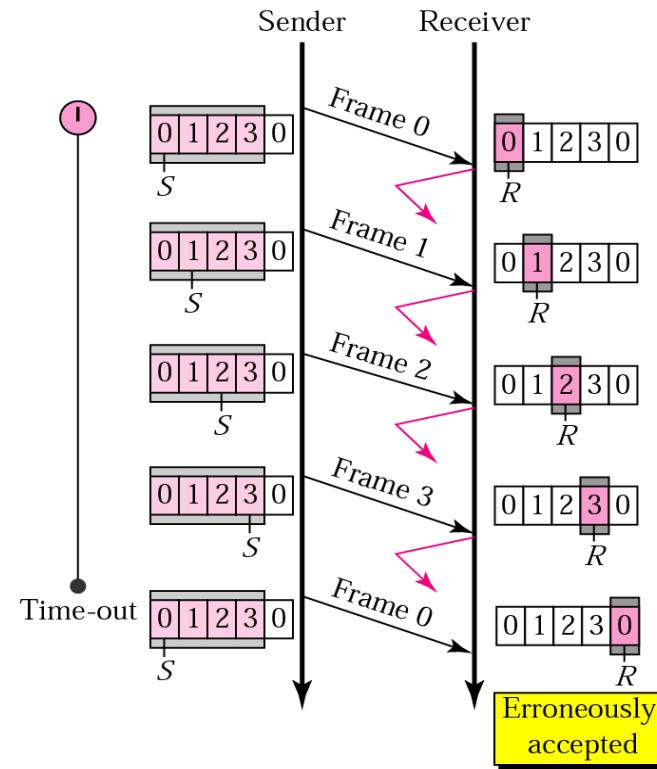
- Depends on size of max. frame number
 - Frame # needs to be included in every frame
 - e.g. 4 bits – $2^4 = 16$ frame numbers
- **Trade-off between window size and header size**

Go-Back-N: Limitation of window size

Size of the sender window must be less than 2^m



a. Window size $< 2^m$



b. Window size $= 2^m$

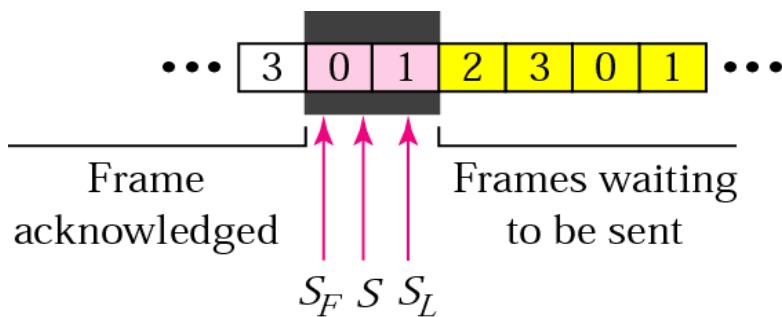
$m = \# \text{ of bits for index}$

* Figure is courtesy of B. Forouzan

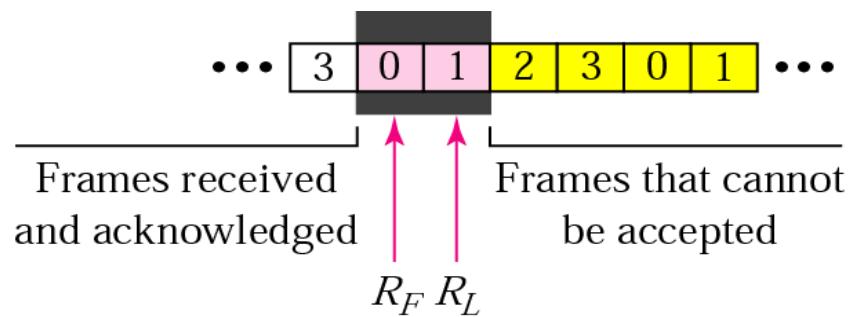
60

Selective Repeat

- Two Windows:
 - 1 Sender Window – 1 Receiver Window



a. Sender window

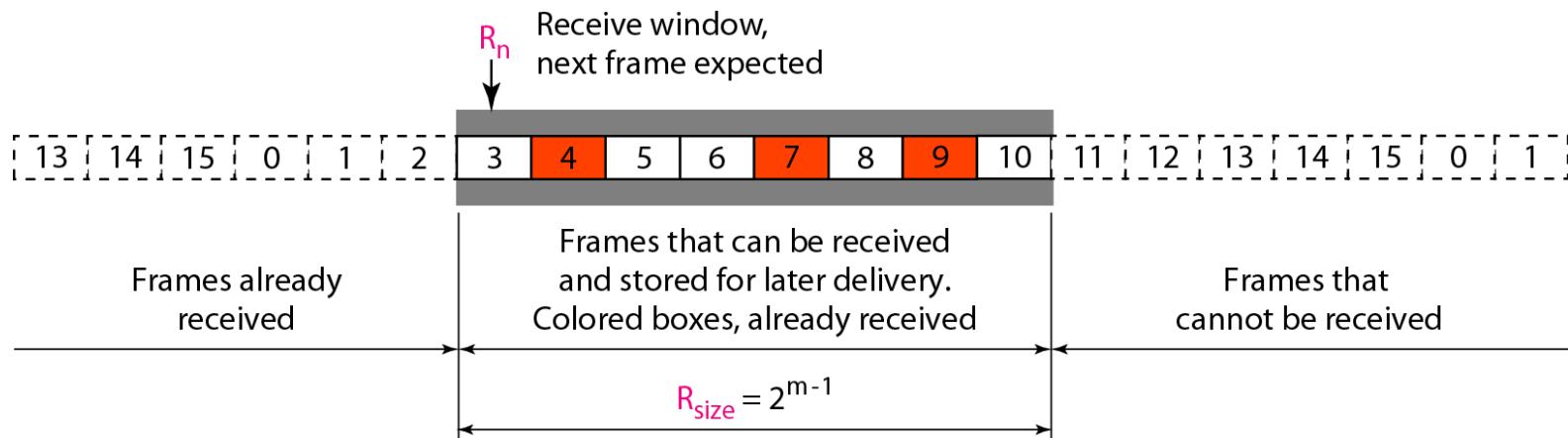


b. Receiver window

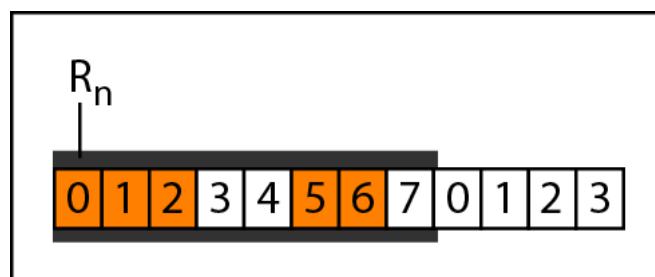
* Figure is courtesy of B. Forouzan 61

Selective Repeat ARQ

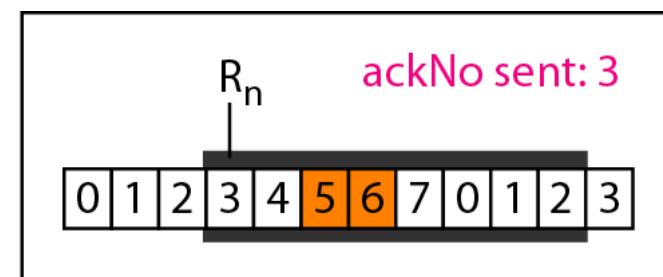
- Window records received frames:



Waiting for #0:



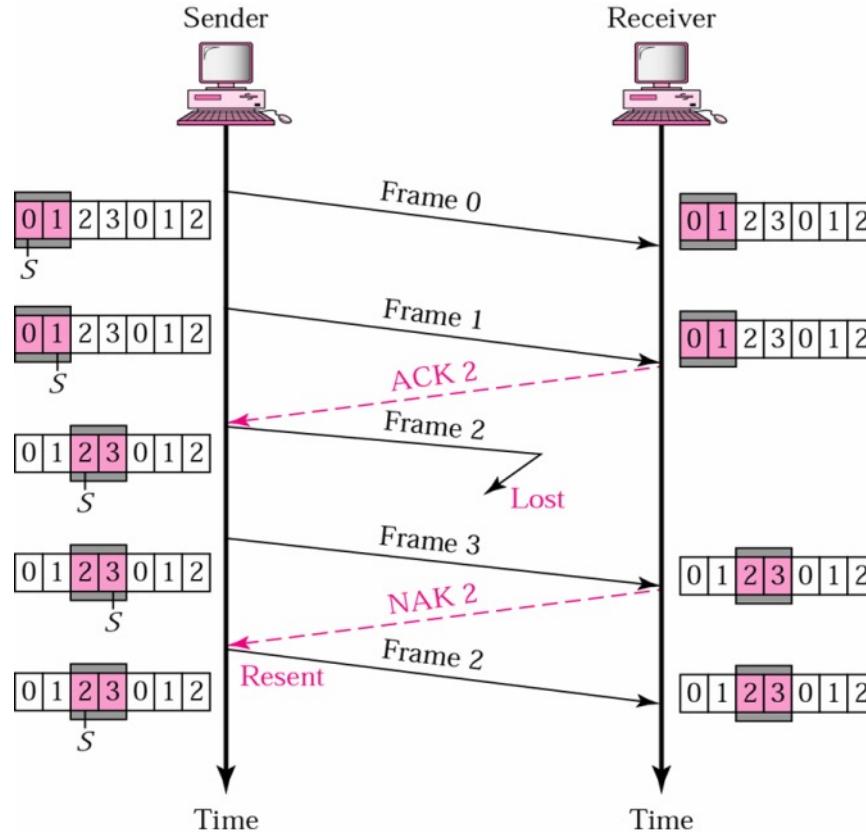
a. Before delivery



b. After delivery

* Figure is courtesy of B. Forouzan

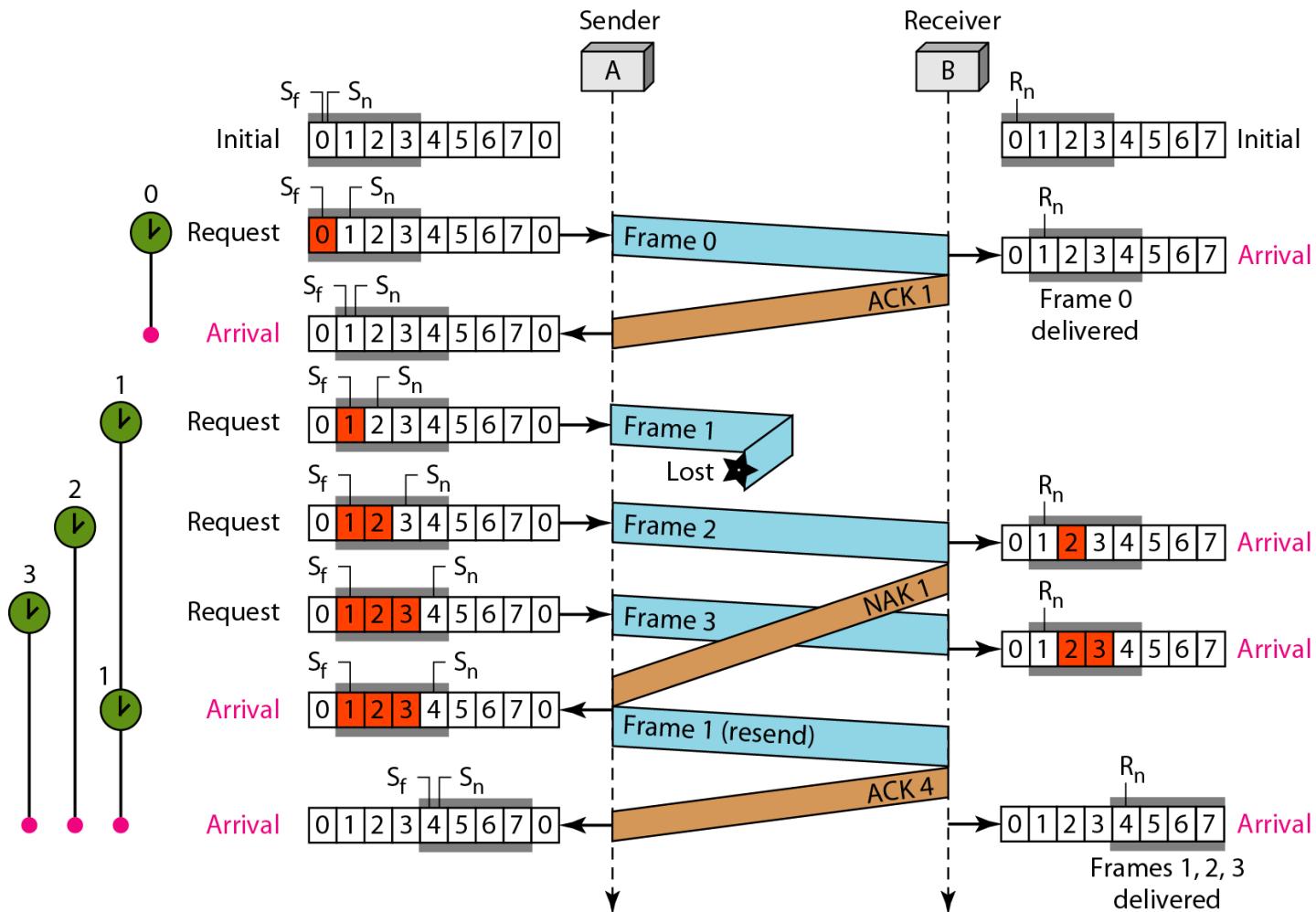
Selective Repeat ARQ: Lost Frame



- NAK = Negative Acknowledgement
- Sender still maintains timers for packets in case NAK gets lost

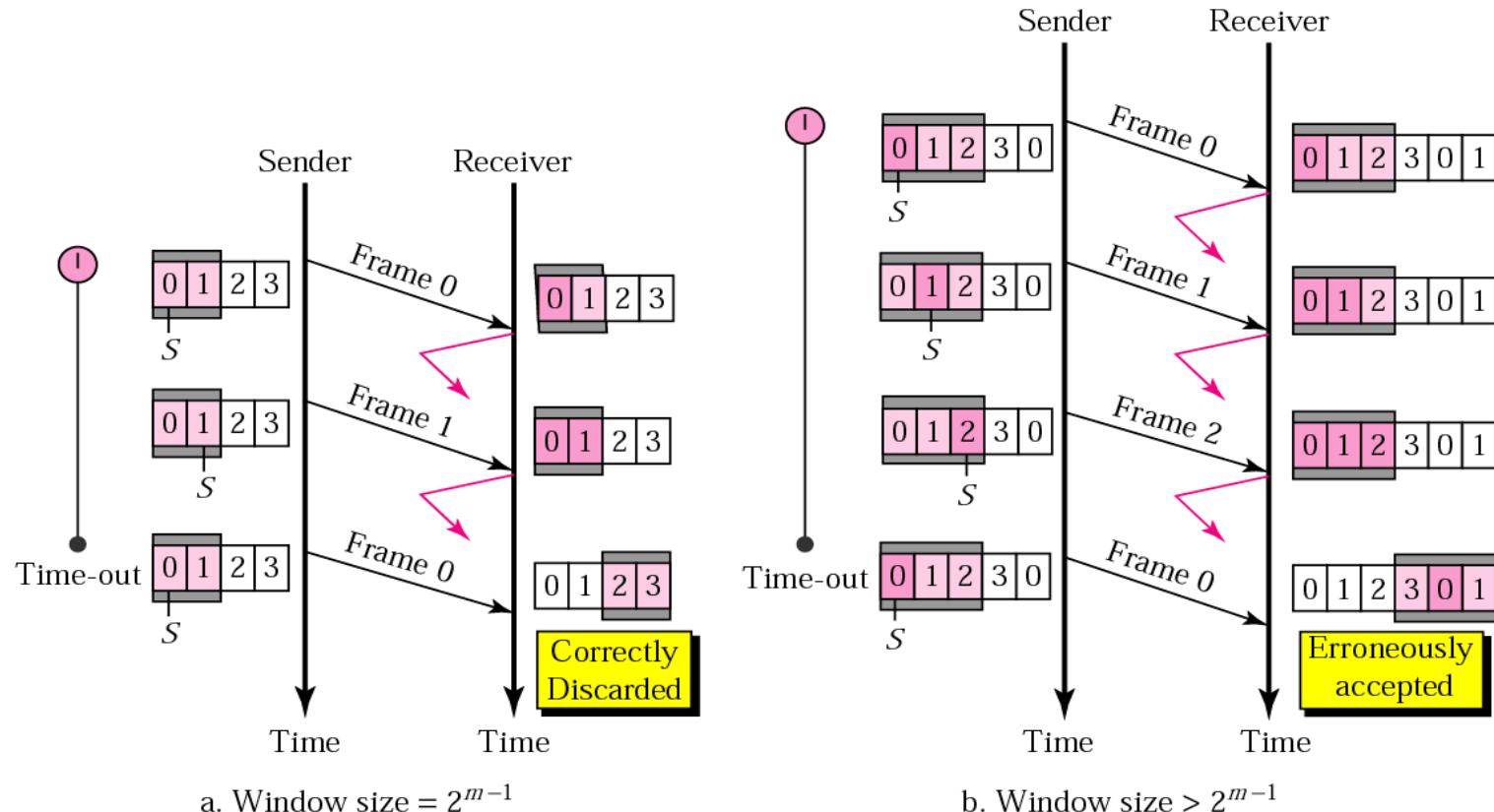
* Figure is courtesy of B. Forouzan

Selective Repeat ARQ



* Figure is courtesy of B. Forouzan

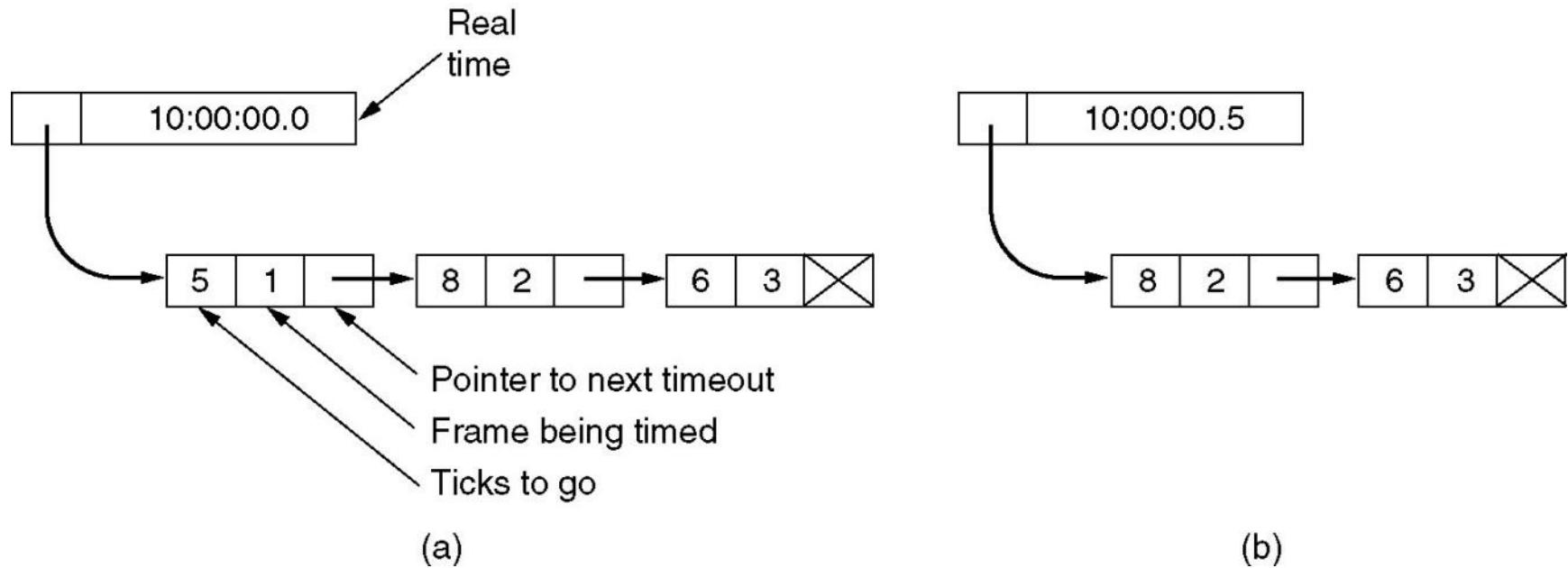
Selective Repeat ARQ: Sender Window



Size of the sender and receiver window must be at most one-half of 2^m

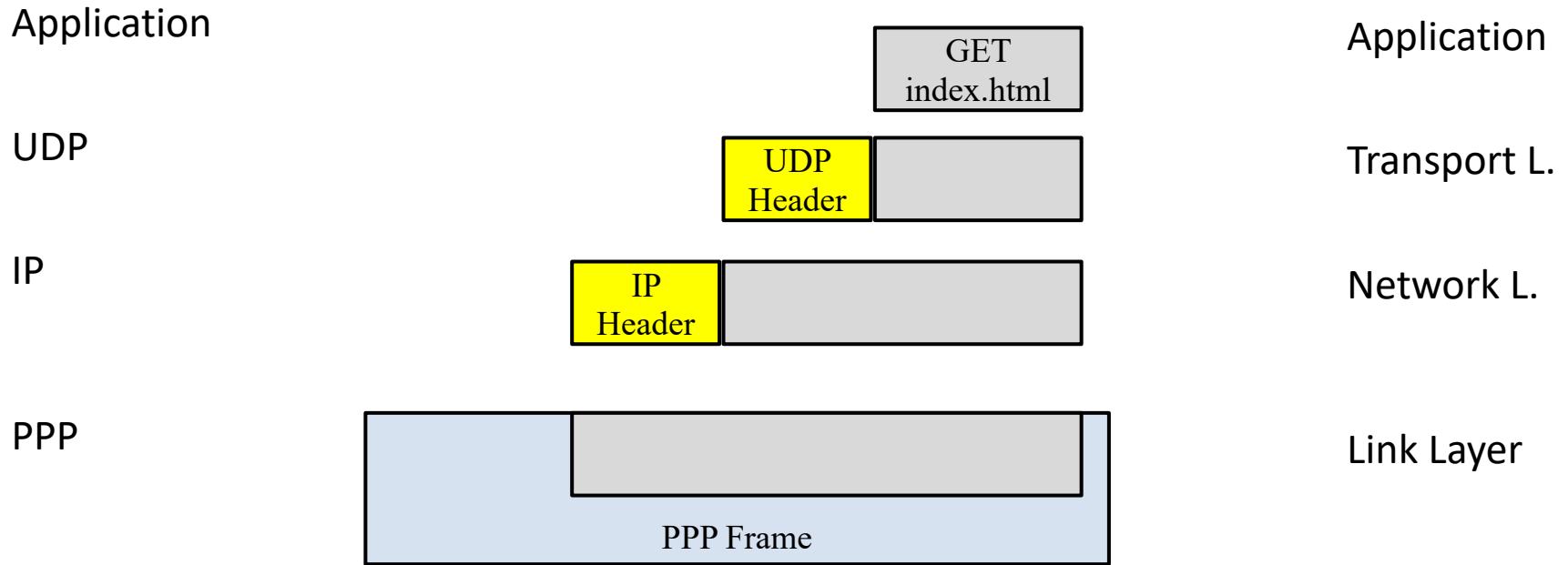
* Figure is courtesy of B. Forouzan

Simulation of Multiple Timers in Software



* Figure is courtesy of A. Tanenbaum

Encapsulation

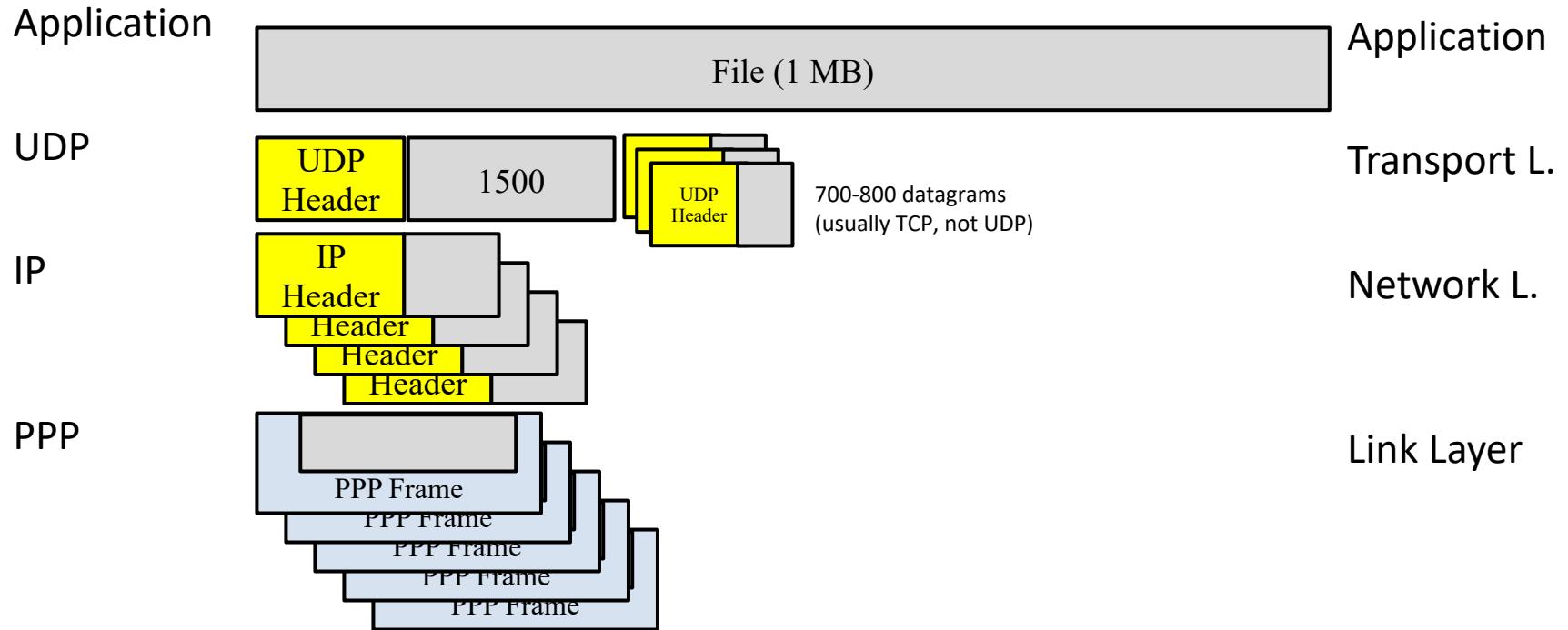


Overhead: Management information needed to deliver payload
eg. UDP header 8 bytes, IP header 20 bytes, PPP 8 bytes

* Figure is courtesy of B. Forouzan

67

Encapsulation



Overhead: Management information needed to deliver payload

eg. UDP header 8 bytes, IP header 20 bytes, PPP 8 bytes

* Figure is courtesy of B. Forouzan

Summary: Flow Control

- Flow Control:
 - Stop-and-Wait
 - Sliding Window
- Error Control
 - Stop-and-Wait ARQ
 - Go-back-N ARQ
 - Selective Repeat ARQ

Items from Today

- Bit-Stuffing/Byte-Stuffing
- Flow Control
- Stop-and-Wait ARQ
- Sliding Window
 - Go-Back-N ARQ
 - Selective Repeat ARQ

Videos for Next Week: Error Control

- 03a-03_Error_Coding_Overview.mp4
- 03a-04_Error_Detection.mp4
- 03a-05_Error_Correction.mp4



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

