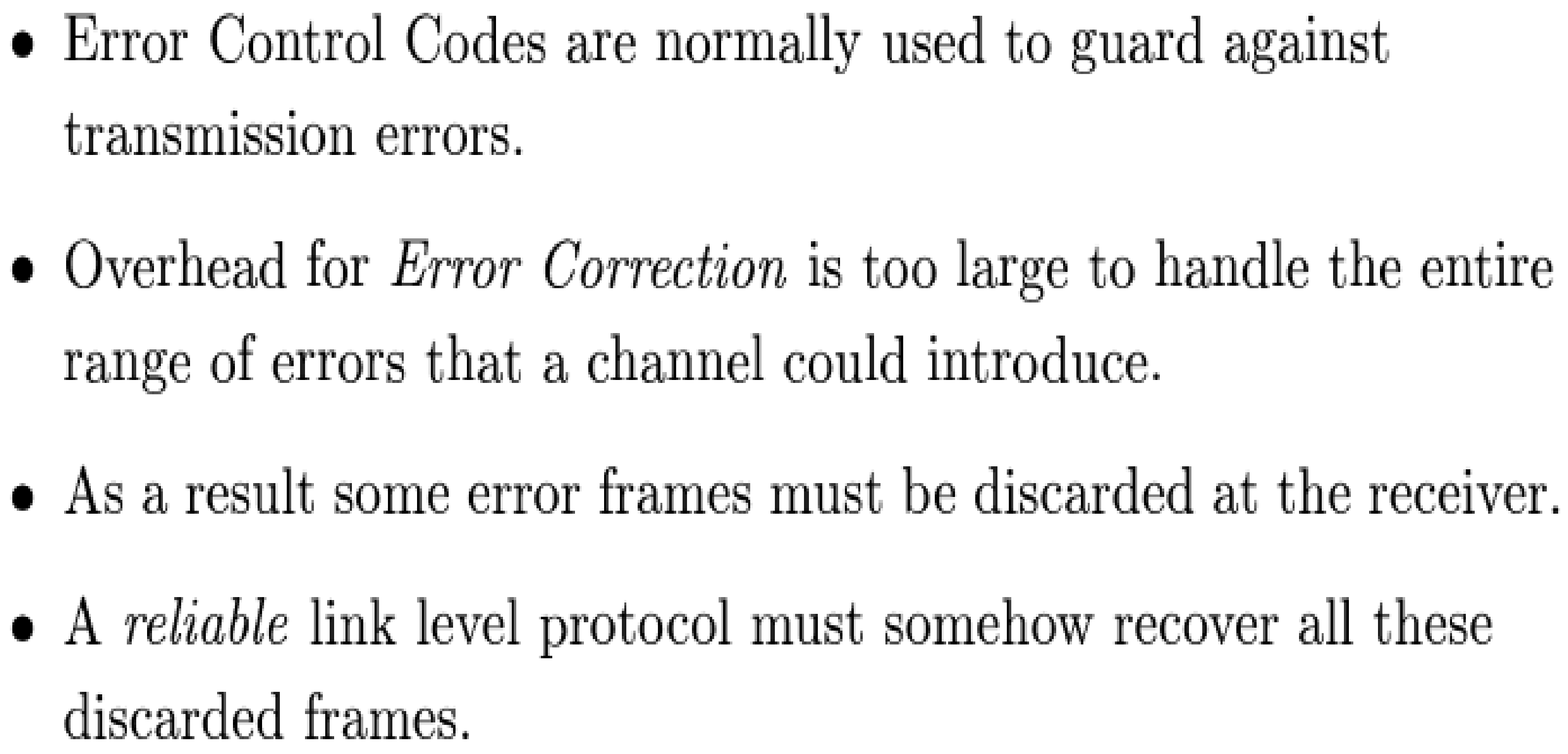
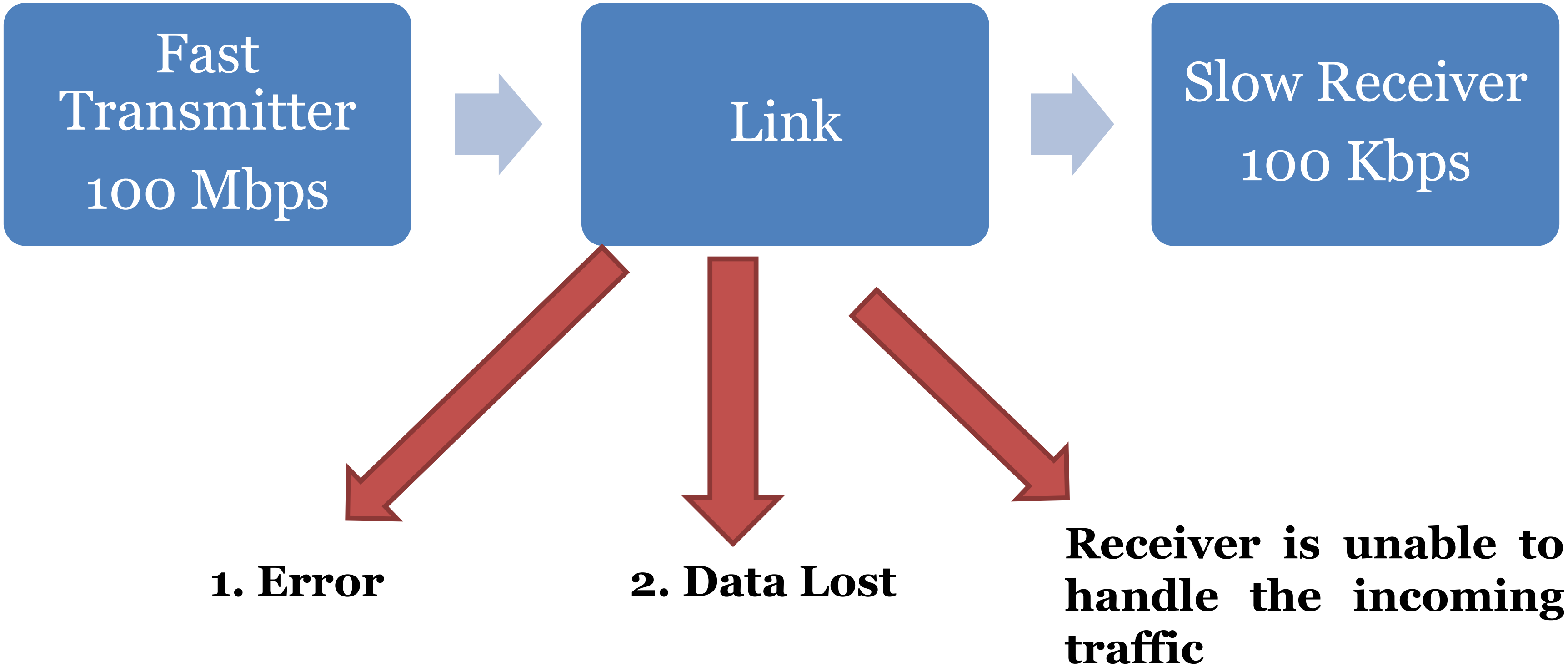
Data Link Layer
Part 2

Flow Control (Motivation)



- Error Control Codes are normally used to guard against transmission errors.
- Overhead for *Error Correction* is too large to handle the entire range of errors that a channel could introduce.
- As a result some error frames must be discarded at the receiver.
- A *reliable* link level protocol must somehow recover all these discarded frames.

Flow Control



Fast Transmitter

100 Mbps

Link

Slow Receiver

100 Kbps

1. Error

2. Data Lost

Receiver is unable to handle the incoming traffic

Solution is to use flow control techniques.

Flow Control

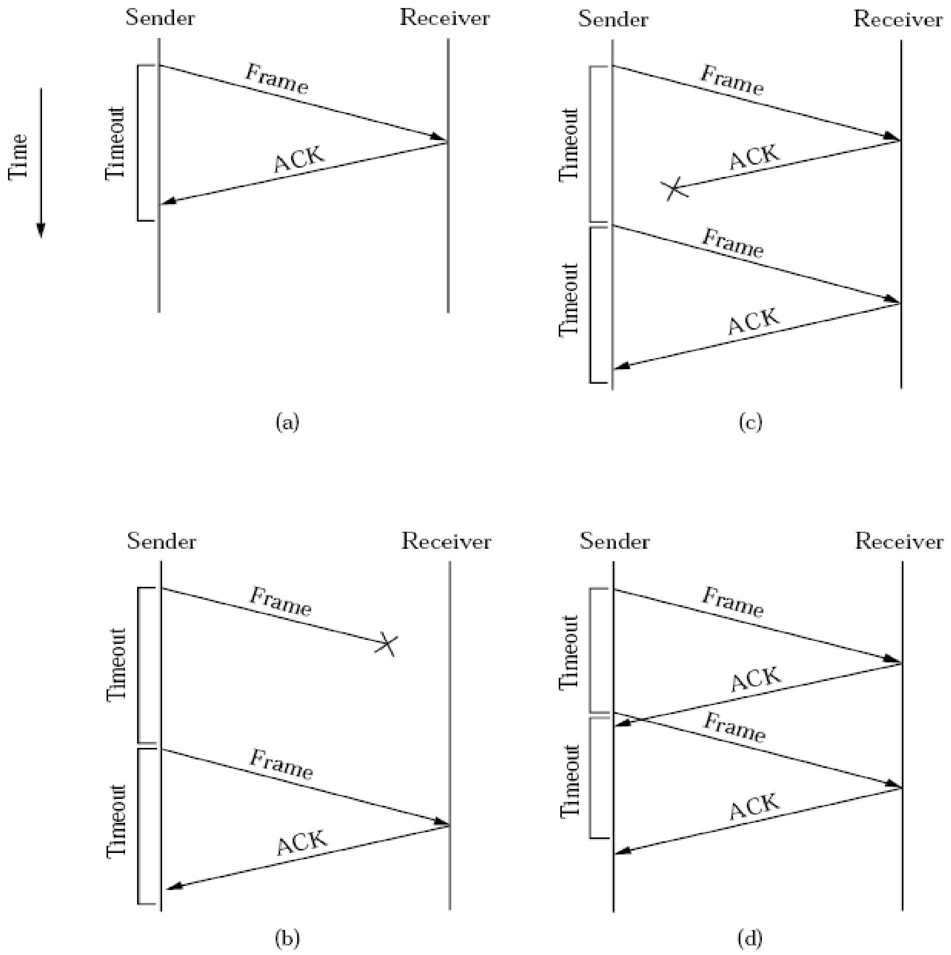
- In practice, errors cause frames to be discarded.
- Link level protocols requires to deliver frames reliably and need some recovery mechanism.
- It is usually implemented by acknowledgements and time outs.
- Algorithm:
  - Receiver can send acknowledge within a control frame (header without data) or piggybacked in the data frame it is about to send.
  - If the sender doesn't receive the acknowledgement within a given time, it retransmit the message again.
  - The retransmission interval is called the timeout period.
- This scheme is known as Automatic Repeat Request (ARQ).

Flow Control Protocols

- Stop and Wait Protocol: Send the next frame after the acknowledgement of the previous sent frame is arrived the sender.

- Sliding Window Protocol (SWP): Send multiple frames without waiting to receive the acknowledgements of previously sent frames to keep the pipe full.

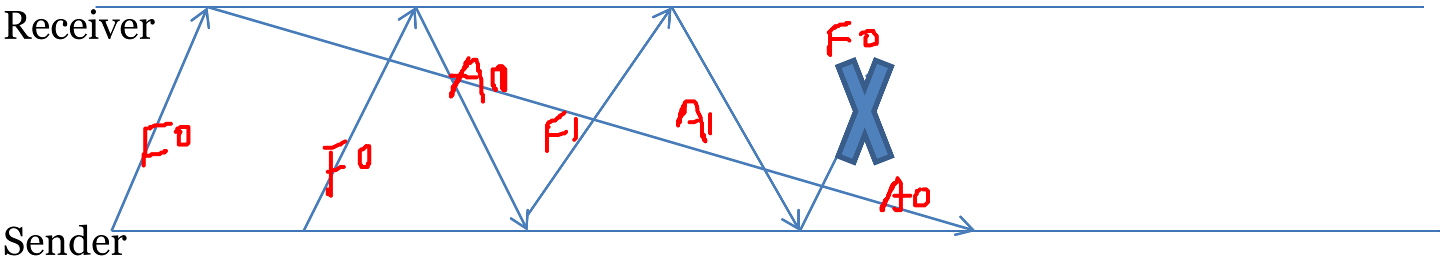
Stop and Wait Protocol Issues



Four different scenarios of stop and wait protocol

Solutions

- S/W protocol with 1 bit sequence number alternating on every consecutive frame [0, 1].
  - If consecutive frames at the receiver have same sequence number → duplicate frames.
  - This is called the alternate bit protocols.
  - This do not solve the problem of delayed acknowledgement.



- Use n number of bits for sequence number.
- Time to live constraint should be added with the acknowledgements.

Disadvantages

- S/W protocol allows only one frame on the link at any given point of time → unable to utilize the link capacity efficiently.
- Example:
- Link Speed: 1.5 Mbps
- Retransmission Time: 45 ms.
- Frame Size: 1 KB

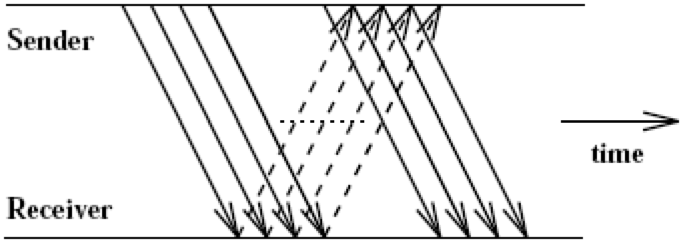
- This implies that to keep the pipe full we can send 67.5 KB data in retransmission time.

$$1.5\ Mbps\ *\ 45\ ms\ =67.5\ Kb\ data$$

- If we divide the above with the frame size we get 8.4375 → The sender can send 8 frames without waiting for the acknowledgements to keep the pipe full (Full utilization of link capacity).

- **This idea is implemented in Sliding Window Protocol (SWP).**

SWP Approach



- Basic Idea
    - Sender transmits up to a given number of frames before expecting an ACK.
    - In essence more packets are in transmission on the channel.
    - Therefore, allows multiple outstanding (un-ACKed) frames.
    - The maximum number of un-ACKed frames allowed on the channel is known as sender's window size.
- Receiver sends back ACKs on receipt of individual frame or sequence of frames.
- Allows better use of link capacity - 'keeping the pipe full'.
- Provides frame ordering and flow control utility.

Window Size

- Sender’s window size can be determined by the following equation:

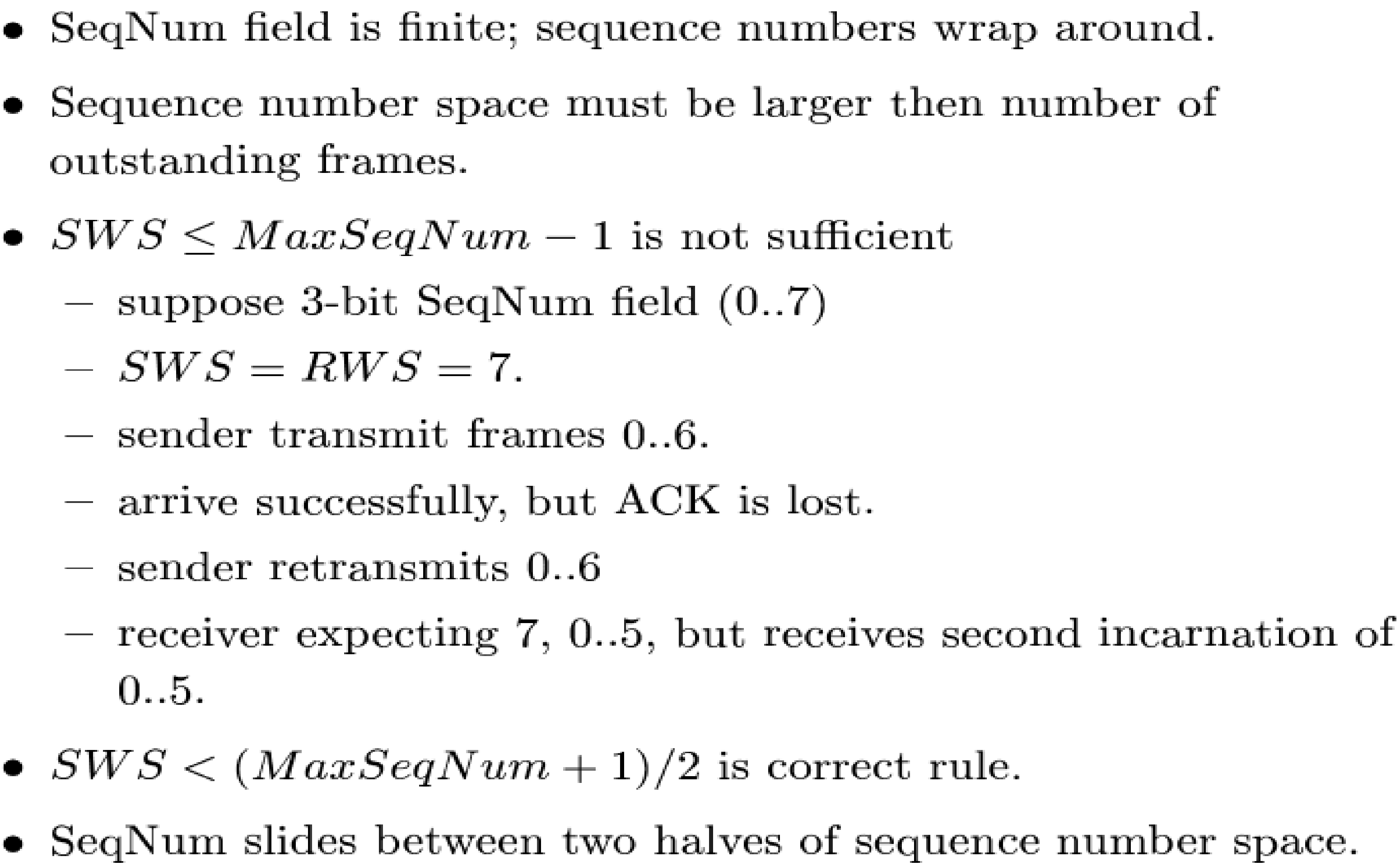
$$SWS = \left\lfloor \frac{Link\ Capacity\ * RTT}{Frame\ Size} \right\rfloor$$

- Receiver’s window size do not matter but ideally is should be the following:

$$1 \leq RWS \leq SWS$$

- RWS ≥ SWS has no significance.

Number of Bits required for the sequence number (Contd.)



- SeqNum field is finite; sequence numbers wrap around.
- Sequence number space must be larger then number of outstanding frames.
- $SWS \leq MaxSeqNum - 1$ is not sufficient
  - suppose 3-bit SeqNum field (0..7)
  - $SWS = RWS = 7$.
  - sender transmit frames 0..6.
  - arrive successfully, but ACK is lost.
  - sender retransmits 0..6
  - receiver expecting 7, 0..5, but receives second incarnation of 0..5.
- $SWS < (MaxSeqNum + 1)/2$ is correct rule.
- SeqNum slides between two halves of sequence number space.

Number of Bits required for the sequence number

- $Maximum\ value\ of\ sequence\ number$ (+1) shall be $greater\ than\ or\ equal\ to$ 2$*SWS$

$$MaxSeqNum + 1\ \geq 2 * SWS$$

- Minimum number of bits required for the sequence number will be:

$$\log_2 MaxSeqNum + 1$$

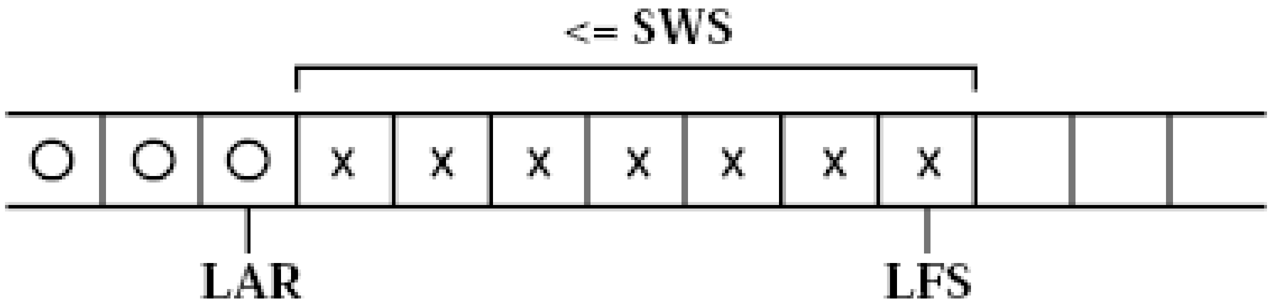
- This means that the sequence number will slide between the two halves of all the frames in transit.

Example: Suppose 3 bits are used as sequence number and the sender's window size is 8.

SWP Algorithm (Sender’s Side)

- Sequence number (SeqNum), assigned to each frame.
- Sender maintains the following variables:
  - SWS, (sender window size) : max number of unACKed frames.
  - LAR, sequence number of last acknowledgement received.
  - LFS, sequence number of last frame sent.

  Ensuring: $LFS - LAR \leq SWS$

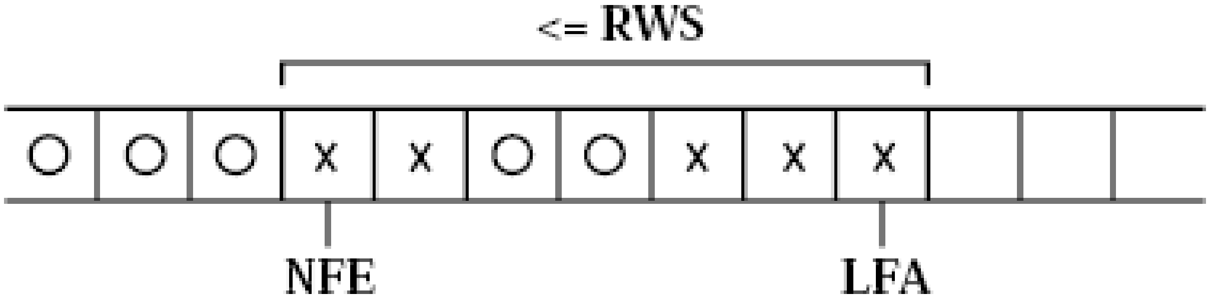


SWP Algorithm (Sender’s Side)

- Sender transmits frames until LFS - LAR ≤ SWS, ie full window.
- After each transmission - increment LFS.
- On receipt of ACK:
  - sender increments LAR.
  - allowing transmission of another frame.
- Timer attached to each frame.
- Sender retransmits frame if ACK not received within timeout.
- Requires buffering of up to SWS frames.

SWP Algorithm (Receiver's Side)

- Receiver maintains 4 variables:
  - RWS, receive window size : max number of out-of-order frames.
  - LFA, sequence number of last frame acceptable.
  - NFE, sequence number of next frame expected.
  - SeqNumToAck, Sequence number of out of order frame received.
- Ensuring: $LFA - NFE + 1 \leq RWS$.



SWP Algorithm (Receiver’s Side)

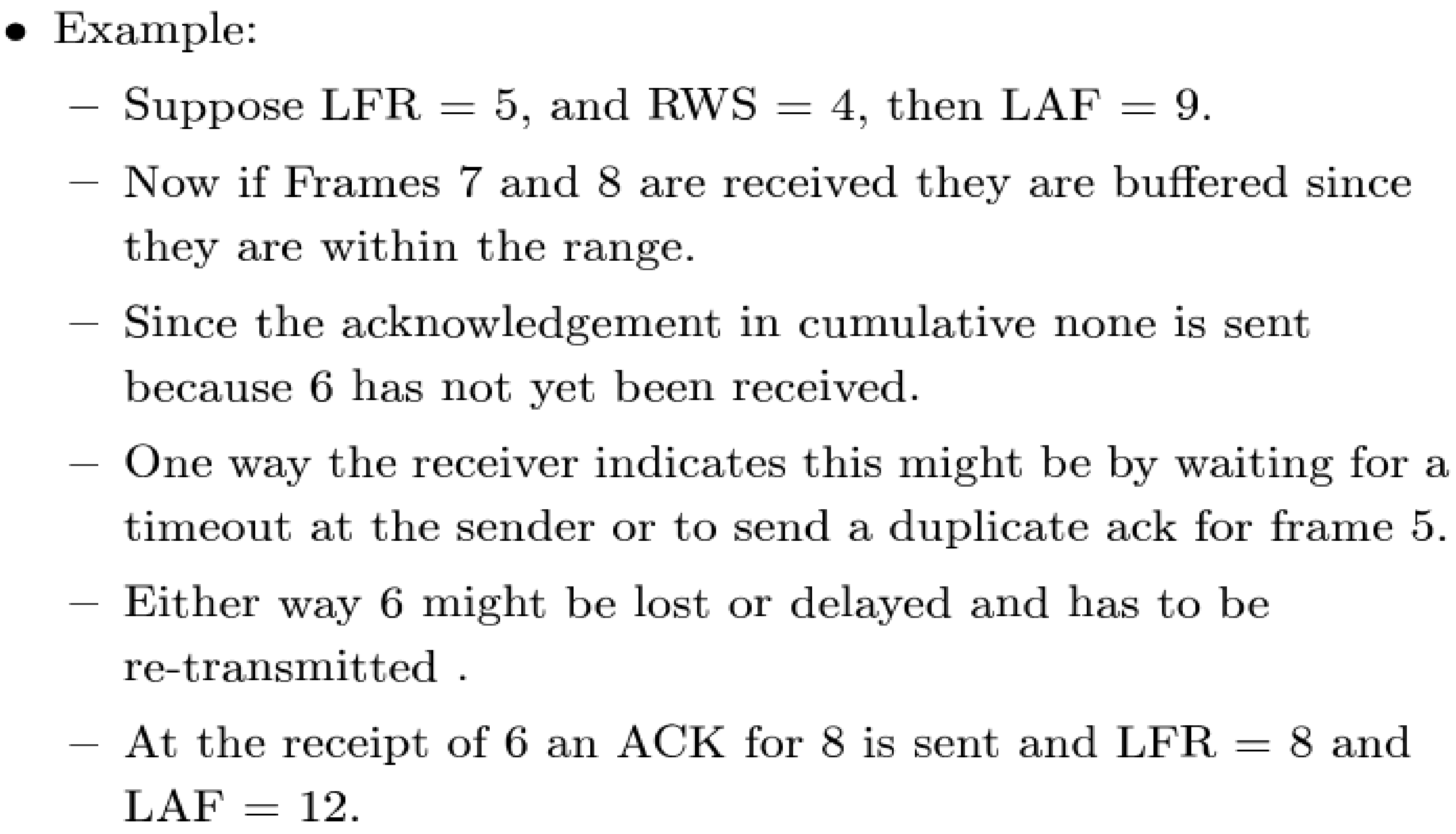
Frame arrives with sequence number, SeqNum. Check whether within receive window, discard otherwise.

```
if $NFE \le SeqNum \le LFA$
        {
        Then accept.
        if $SeqNum = NFE$
        then send Ack.
        //If out of order frame received
        if $SeqNum \neq NFE$
        //This means, out-of-order frame, so buffer SeqNum and await remaining frames.
        {        then set $SeqNumToAck = SeqNum$ of out of order frame received
                 If $SeqNum = NFE$ and all frames $\le SeqNumToAck$ received
                 ACK SeqNumToAck (cumulative ACK).
                 Set $NFE = SeqNumToAck + 1$ and $LFA = SeqNumToAck + RWS$
        }}
If $SeqNum < NFE$ or $SeqNum > LFA$
Then discard.
```

Advantages of SWP

- Sliding window protocol serves three roles.
- Reliable transmission - main role.
- Frame order preservation:
  - receiver buffers out-of-order frames
  - frame passed up only after passing frames with smaller SeqNum.
  - ensures higher protocol receives frames in correct order
- Flow control:
  - SWP can include 'feedback' to ensure receiver isn't overloaded
  - with ACK, receiver sends 'buffer spaces remaining'
  - avoids sender transmitting 'out-of-window' traffic

Sliding Window Protocol Example



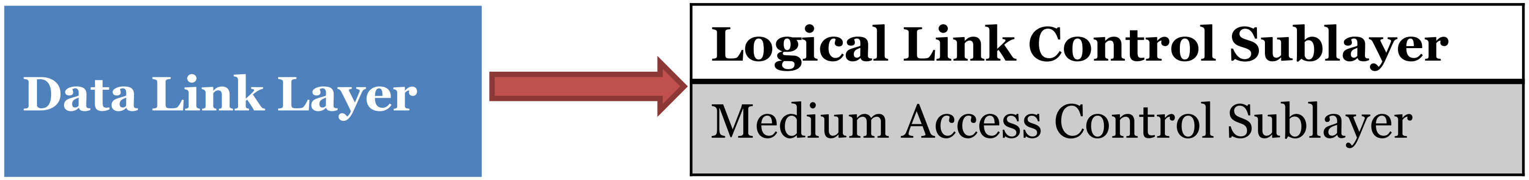
- Example:
  - Suppose LFR = 5, and RWS = 4, then LAF = 9.
  - Now if Frames 7 and 8 are received they are buffered since they are within the range.
  - Since the acknowledgement in cumulative none is sent because 6 has not yet been received.
  - One way the receiver indicates this might be by waiting for a timeout at the sender or to send a duplicate ack for frame 5.
  - Either way 6 might be lost or delayed and has to be re-transmitted .
  - At the receipt of 6 an ACK for 8 is sent and LFR = 8 and LAF = 12.

Variants of SWP

- Duplicate Ack (DAK)
- Negative ACK (NAK)
- Selective ACK (SAK)
- Cumulative ACK (CAK)

Medium Access Control Sublayer

Motivation

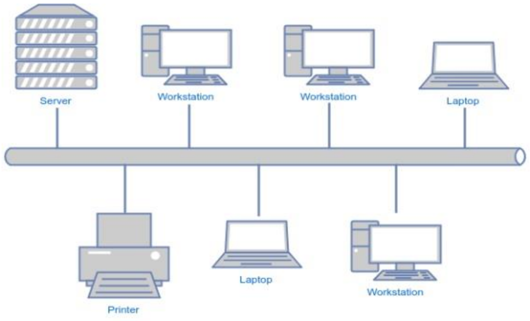


Data Link Layer

Logical Link Control Sublayer

Medium Access Control Sublayer

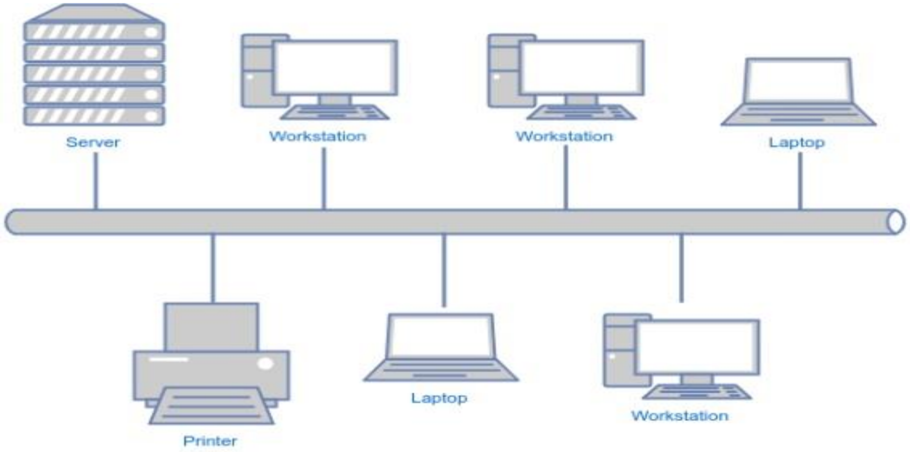
- Medium access control sublayer is required when multiple stations are sharing the medium.
- It is possible that two stations start transmitting their data at a particular point of time → Collision
- Collision is a phenomena when two signal overlap in time on the same communication channel.
- To deal with the collision MAC protocol is required.



Bus Topology Network

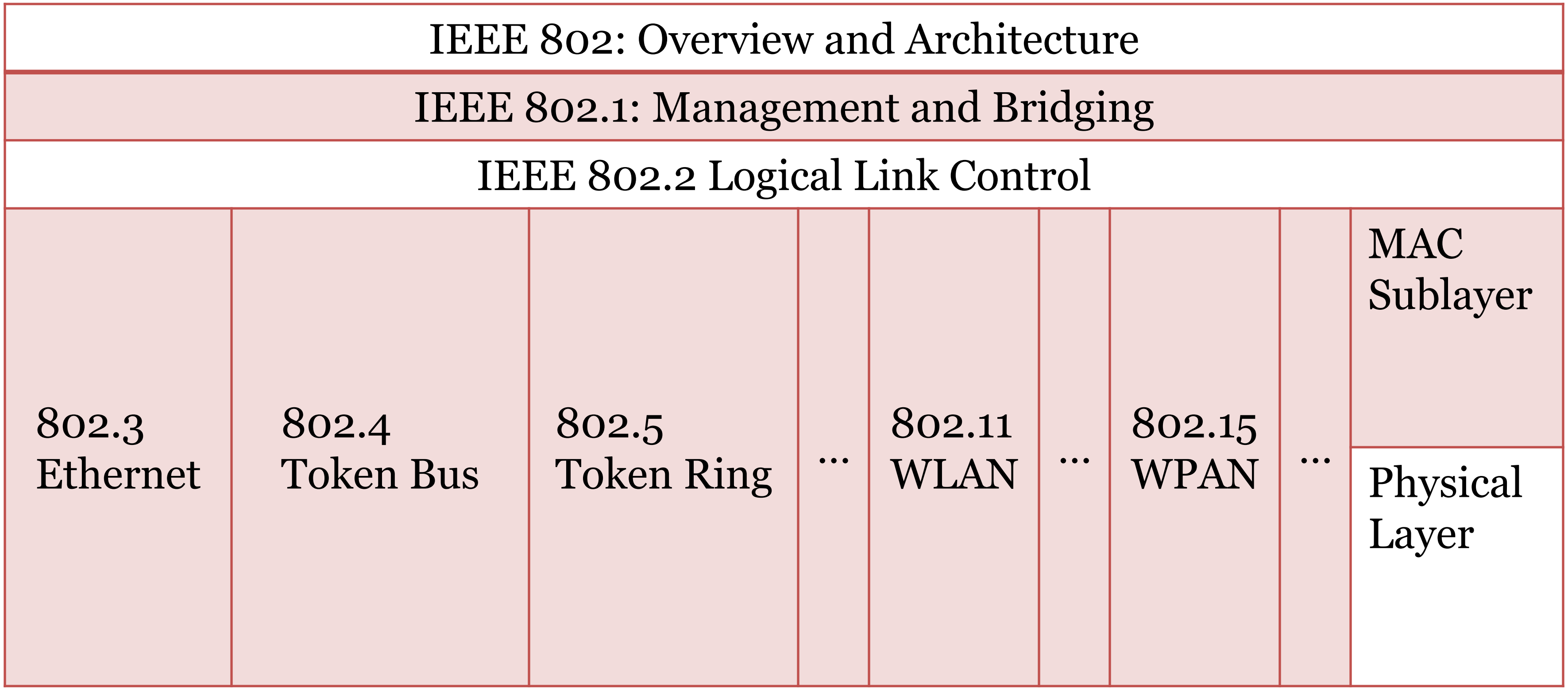
Motivation

- MAC sublayer is not important in point to point network.
- But in real world, links are shared by the multiple hots. There is no dedicated unique channel between the two shots.
- Any pair of hosts can borrow the link for direct communication.
- MAC sublayer is mandatory in all the shared link (multi-point) network.
- Example: Ethernet, Token Ring, Token Bus, WLAN, WPAN etc.
- Most of the MAC protocols are defined by IEEE 802.x standards. Where x= 1, 2, 3 ..



Bus Topology Network

IEEE 802.x standards



IEEE 802: Overview and Architecture

IEEE 802.1: Management and Bridging

IEEE 802.2 Logical Link Control

802.3
Ethernet

802.4
Token Bus

802.5
Token Ring

…

802.11
WLAN

…

802.15
WPAN

…

MAC
Sublayer

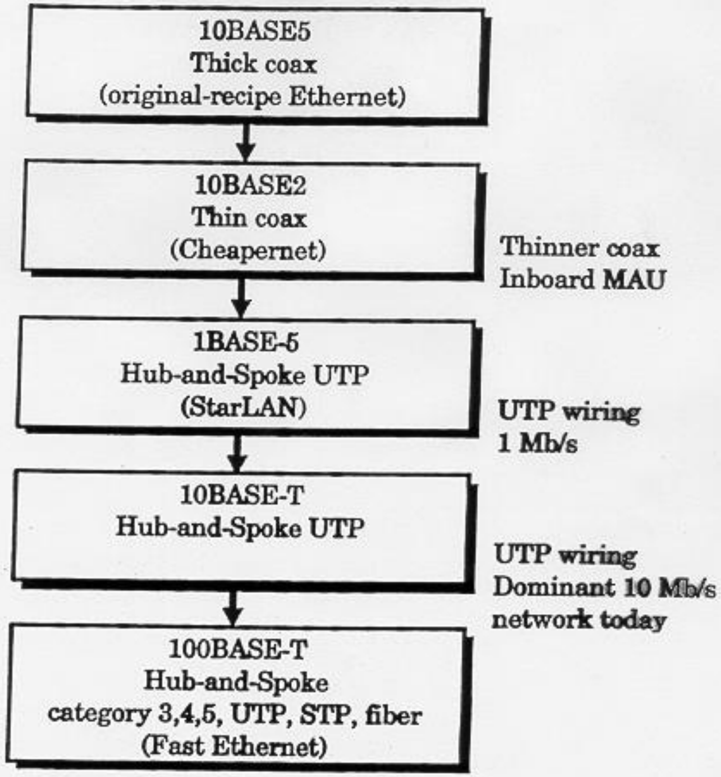
Physical
Layer

Note: IEEE 802.x standards are coving the MAC sublayer and Physical layer of TCP/IP reference Model.

Ethernet (IEEE 802.3)

- History
  - Developed by Xerox PARC in mid of 1970.
  - Roots in Aloha packet radio network
  - Initially 3 Mbps baseband coaxial cable (thick Ethernet) was proposed in 1974.
  - Standardized by Xerox, DEC and Intel in 1978.
  - IEEE Adopted the standard and named it IEEE802.3 (Ethernet) in 1980s.
  - Many variants of IEEE 802.3 is available in the market like Fast Ethernet, Gigabit Ethernet etc.

Ethernet variants



10BASE5
Thick coax
(original-recipe Ethernet)

10BASE2
Thin coax
(Cheapernet)

Thinner coax
Inboard MAU

1BASE-5
Hub-and-Spoke UTP
(StarLAN)

UTP wiring
1 Mb/s

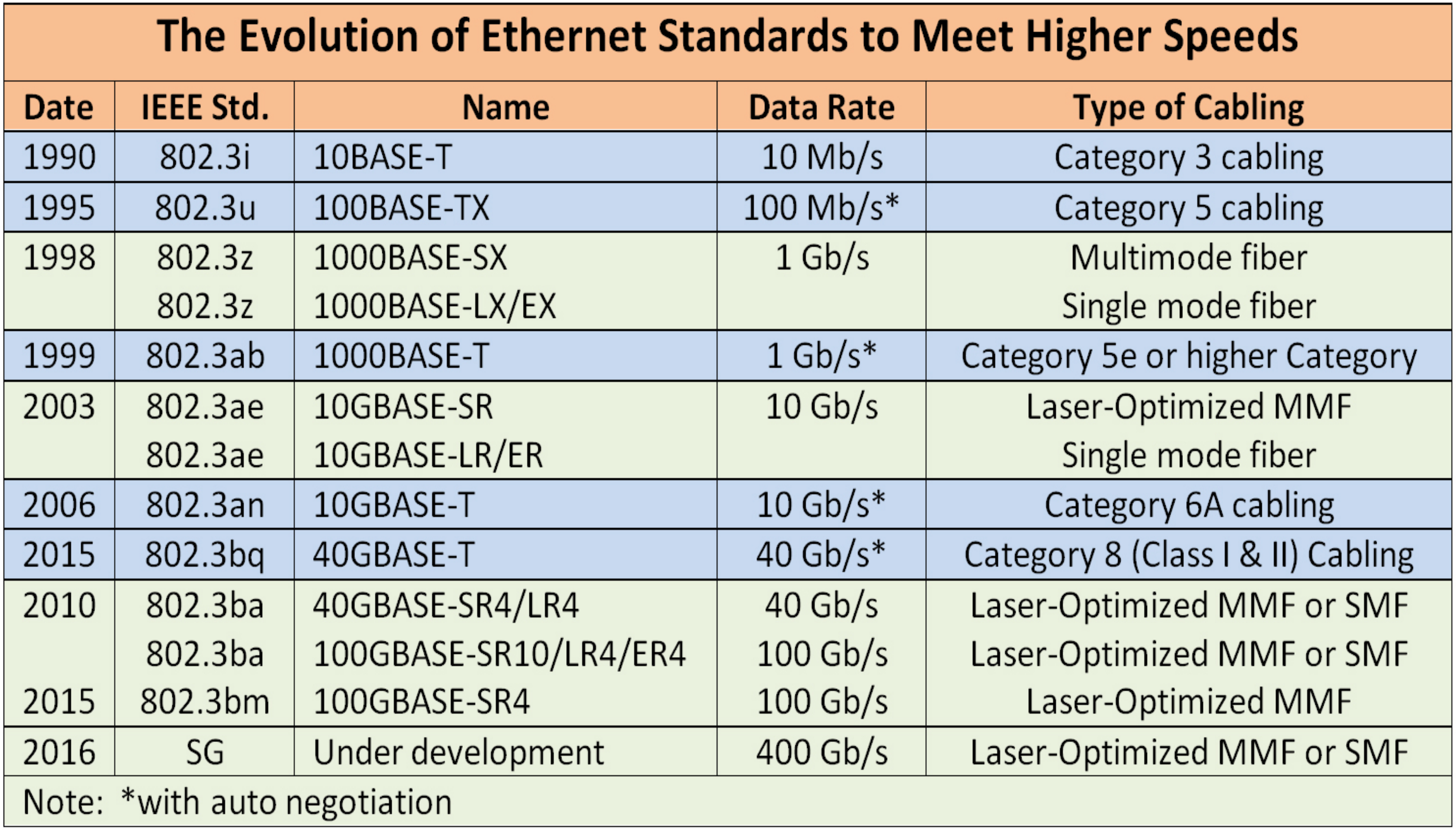
10BASE-T
Hub-and-Spoke UTP

UTP wiring
Dominant 10 Mb/s
network today

100BASE-T
Hub-and-Spoke
category 3,4,5, UTP, STP, fiber
(Fast Ethernet)

Figure 1.4 Lineage of Fast Ethernet

Ethernet variants



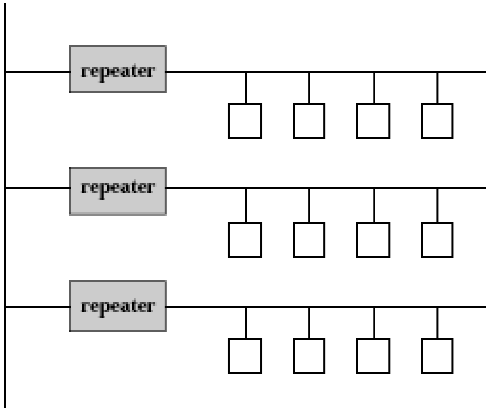
The Evolution of Ethernet Standards to Meet Higher Speeds

Date	IEEE Std.	Name	Data Rate	Type of Cabling
1990	802.3i	10BASE-T	10 Mb/s	Category 3 cabling
1995	802.3u	100BASE-TX	100 Mb/s*	Category 5 cabling
1998	802.3z	1000BASE-SX	1 Gb/s	Multimode fiber
	802.3z	1000BASE-LX/EX		Single mode fiber
1999	802.3ab	1000BASE-T	1 Gb/s*	Category 5e or higher Category
2003	802.3ae	10GBASE-SR	10 Gb/s	Laser-Optimized MMF
	802.3ae	10GBASE-LR/ER		Single mode fiber
2006	802.3an	10GBASE-T	10 Gb/s*	Category 6A cabling
2015	802.3bq	40GBASE-T	40 Gb/s*	Category 8 (Class I & II) Cabling
2010	802.3ba	40GBASE-SR4/LR4	40 Gb/s	Laser-Optimized MMF or SMF
	802.3ba	100GBASE-SR10/LR4/ER4	100 Gb/s	Laser-Optimized MMF or SMF
2015	802.3bm	100GBASE-SR4	100 Gb/s	Laser-Optimized MMF
2016	SG	Under development	400 Gb/s	Laser-Optimized MMF or SMF

Note: *with auto negotiation

Physical Properties (initial Version)

- Ethernet segment - up to 500m of 50 coaxial cable.
- Hosts tap into cable for connection, > 2.5 m apart.
- Transceiver detects ‘idle’ line and transmits signals onto link.
- Transceiver connects to Ethernet adaptor (plugs into host) - implements the Ethernet protocol.
- Signals placed on link are broadcast over entire network.
- Terminators at end of segments prevent reflections.
- Manchester encoding used. Multiple segments joined by repeaters.
- Repeaters ‘forward’ signals, avoiding problems with signal decay.
- But, no more than four repeaters between any pair of hosts.
- Hence max length = 2500m, eg using 2 repeaters between hosts:



MAC Protocol

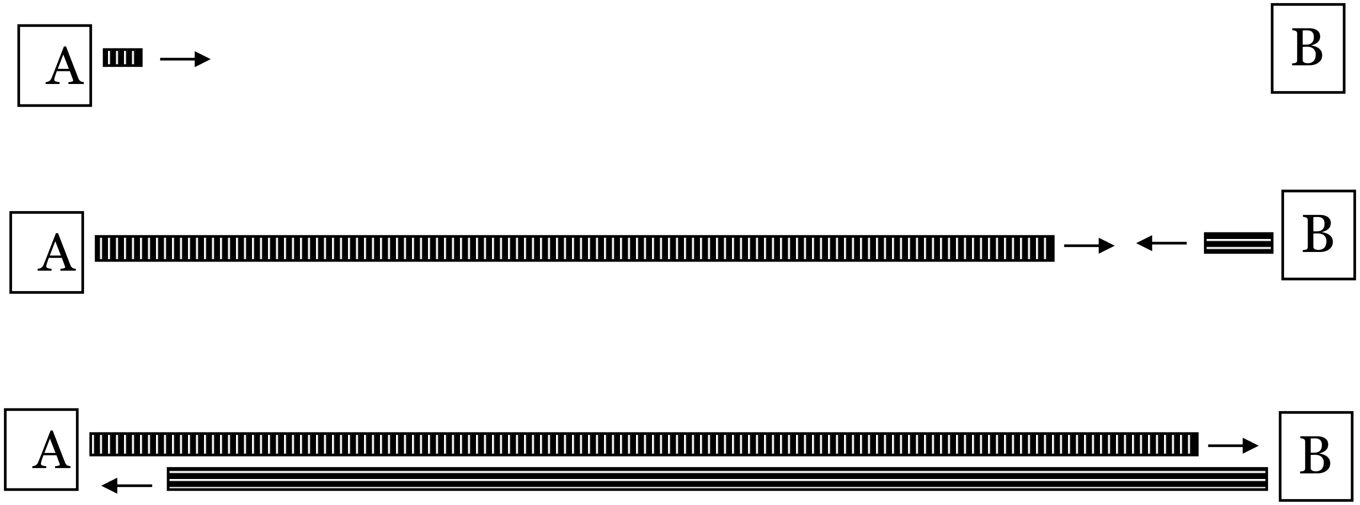
- Multiple access network - nodes send and receive frames over a shared link.

MAC Protocol: 1-persistent, CSMA-CD with Binary Exponential Backoff

- Carrier sense - all nodes are able to determine when link is ‘idle’ or ‘busy’.
- Multiple Access: More than one node can access the medium at any point of time.
- Collision detect - nodes able to detect if frame being transmitted has interfered with frame being transmitted by another node.

Collision Detection (Worst Case)

A begins to transmit at t=0



B begins to transmit at $t= t_{prop}-\delta$;
B detects collision at $t= t_{prop}$

A detects collision at $t= 2\ t_{prop}-\delta$

It takes **2 t_{prop}** to find out if channel has been captured

Collision Detection (Worst Case)

- Frame *seizes the channel after 2* t_{prop}
- On 1 km Ethernet, t_{prop} is approximately 51.2 microseconds.
- Contention interval = *2* t_{prop}
- ***Interframe gap = 9.6 microseconds***
- Modeled as *slotted scheme* with slot = 2 t_{prop}

MAC Protocol

1. Assume an adaptor has a frame to transmit.
2. If link ‘idle’ - transmit immediately at the start of next slot.
3. If link ‘busy’ - wait for ‘idle’, then transmit immediately.
4. Two adaptors may transmit simultaneously - collision occurs.
5. On detecting collision, adaptor transmits 32-bit jamming sequence to warn other transmitting nodes of collision.
6. But - to ensure every node detects jamming sequence:
   - all nodes need to at least ‘fill the pipe’
   - max Ethernet RTT is 51.2 us, at 10 Mbps →512 bits
   - ensures jamming sequence gets to each node before end of transmission.
   - Hence data field must be ≥ 46 bytes:
   - header (14) + data (46) + CRC (4) = 64 bytes = 512 bits →Minimum 46 bytes required in the data part of a frame.

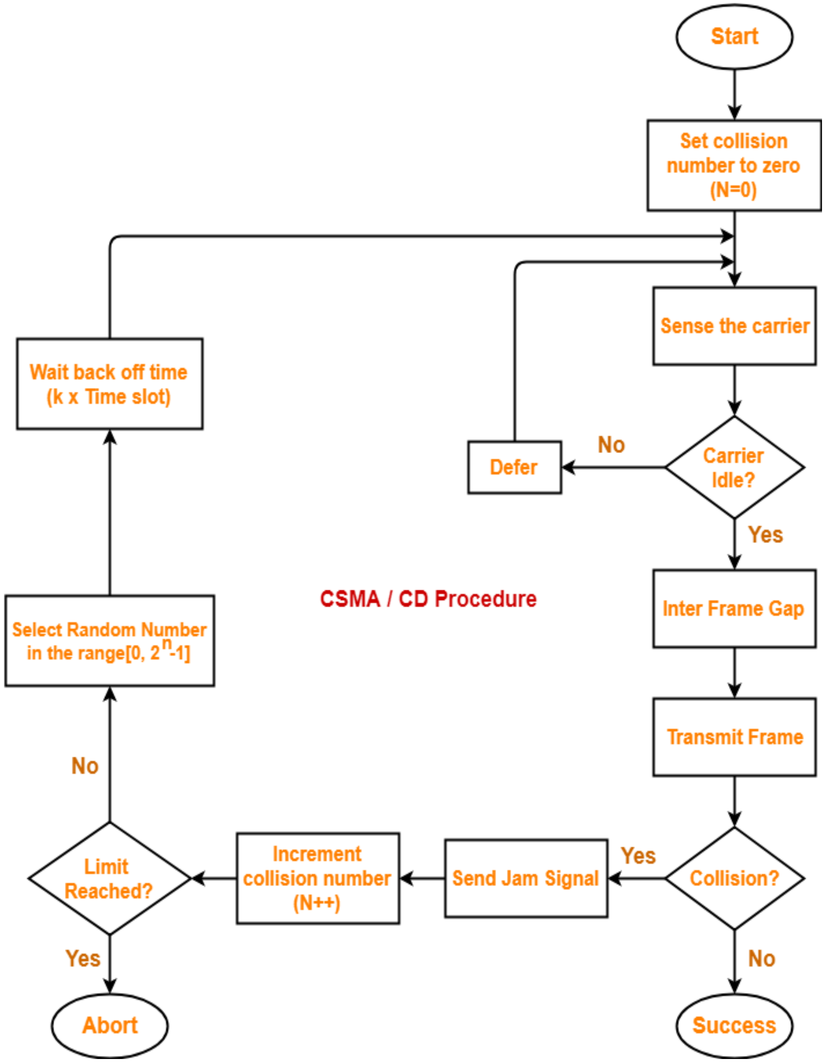
MAC Protocol

7. Both transceivers detect collision and stop transmission.
8. Each adaptor then waits for random amount of time (up to given maximum) before retransmitting based on the binary backoff algorithm.
9. An adaptor gives up after given number of attempts.

Binary Exponential Backoff Algorithm

- Upon a collision, the *sending stations* increment a local counter K.
- The backoff interval is randomly selected using a uniform distribution over the $L = 2^K$ slots.
- K is initially set to 0.
- Thus upon collision, the value of L is doubled locally for each *sending station*.
- Maximum number of retransmission is restricted to 16.

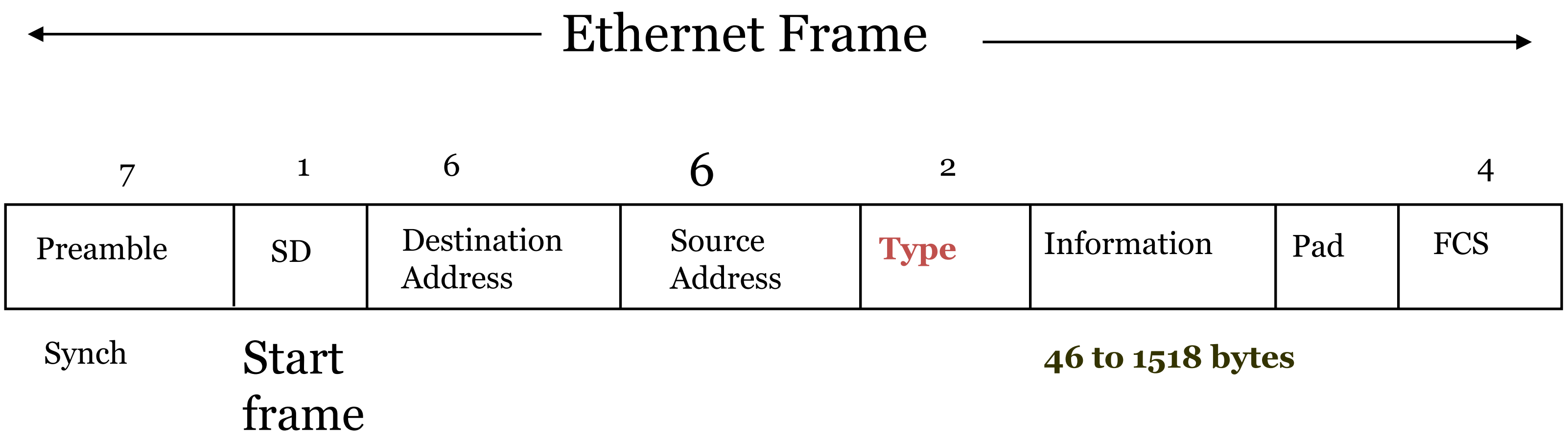
Flowchart of the Algorithm



802.3 MAC Frame

7	1	2 or 6	2 or 6	2	Min:46 Max:1500		4
Preamble	SD	Destination Address	Source Address	**Length**	Information	Pad	FCS
Synch	Start frame						

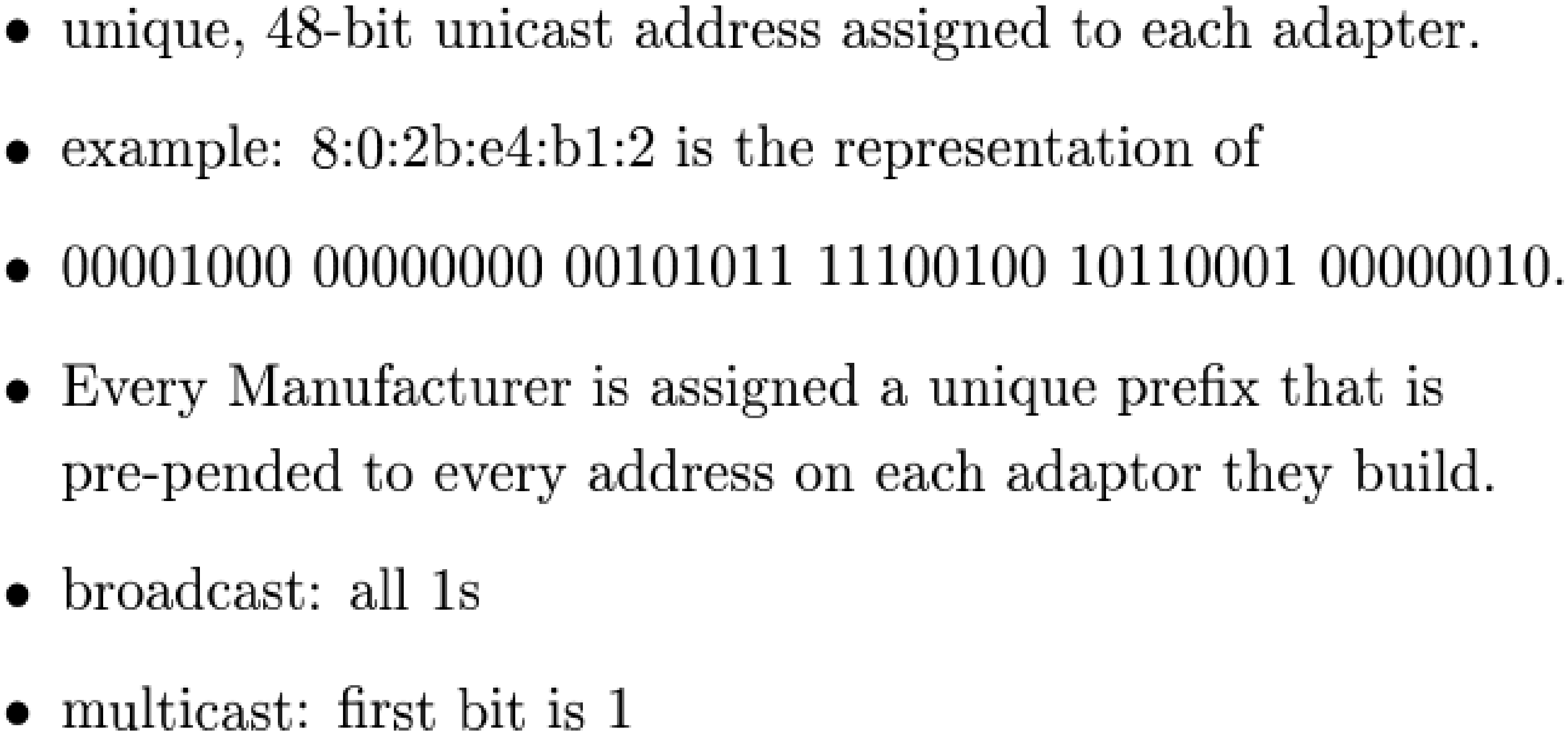
46 to 1518 bytes



Ethernet Frame

7	1	6	6	2			4
Preamble	SD	Destination Address	Source Address	Type	Information	Pad	FCS
Synch	Start frame				46 to 1518 bytes		

Ethernet Address

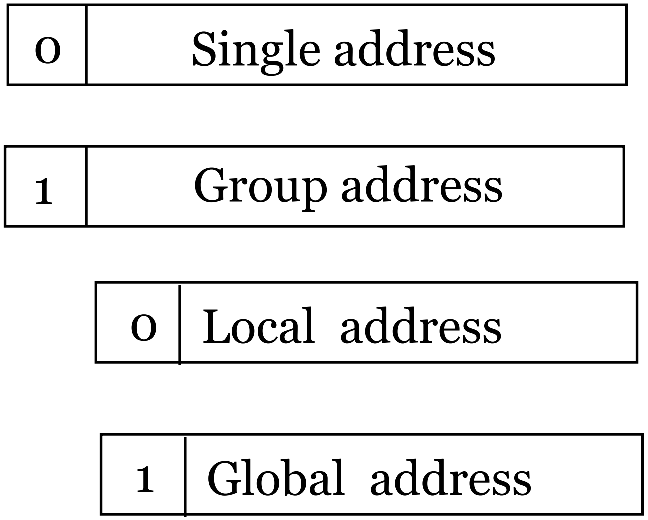


- unique, 48-bit unicast address assigned to each adapter.
- example: 8:0:2b:e4:b1:2 is the representation of
- 00001000 00000000 00101011 11100100 10110001 00000010.
- Every Manufacturer is assigned a unique prefix that is pre-pended to every address on each adaptor they build.
- broadcast: all 1s
- multicast: first bit is 1

Ethernet Address

11001100 11001100 11001100 11001100 11001100 11001100

2a:3b:fe:4c:5b:88



0 Single address

1 Group address

0 Local address

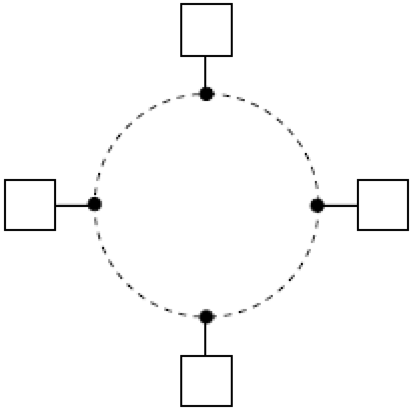
1 Global address

- Destination address is either single address
- or group address (broadcast = 111...111)
- Addresses are defined on local or universal basis
- 2^{46} possible global addresses

Ethernet: Performance and Popularity

- Ethernet performs best under light load (<30%).
- Heavy use - too many collisions.
- But, most Ethernets are used well under capacity, eg fewer than max hosts, shorter lengths, etc.
- Why is Ethernet successful?
  - easy to administer and maintain.
  - easy to add extra hosts
  - inexpensive: cable + adaptors

- Main alternative to Ethernet for shared access network.
- Most general: Fiber Distributed Data Interface (FDDI).
- Nodes connected in ring and operate as ring, ie not as pt-to-pt links configured in loop.



Token Rings: Sharing Access

- Token (bit pattern) circulates around ring: nodes receive from upstream and pass on downstream.
- Node wishing to transmit:
  - takes token off ring, ie does not pass token downstream.
  - forwards frame to downstream neighbour (inserts data into ring).
- Nodes pass frame around ring, with destination node keeping copy.
- Freeing the ring:
  - when frame reaches sender (one round trip), sender keeps frame and forwards token to downstream neighbour.
  - downstream nodes can then grab token - round robin sharing.

Device Driver

- a collection of OS routines that anchor the protocol stack.
- initializes device, transmits frames and handles interrupts.
- consists of :
  - interrupt handler
  - data processor.
- *data processor* handles the copy of data to and from the adaptor.
- *interrupt handler* handles all interrupts sent to the device.