Computer Networks

Data Link Layer

Nassim Kobeissy

Data Link Layer

Achieving reliable, **efficient communication** of whole units of information called **frames** (rather than individual bits, as in the physical layer) between two **adjacent** machines.

- Providing a well-defined interface to network layer
- Framing
- Error detection and correction
- DLL Protocols (CSMA)
- Ethernet

Services provided to network layer

- The function of the data link layer is to provide services to the network layer. The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine.

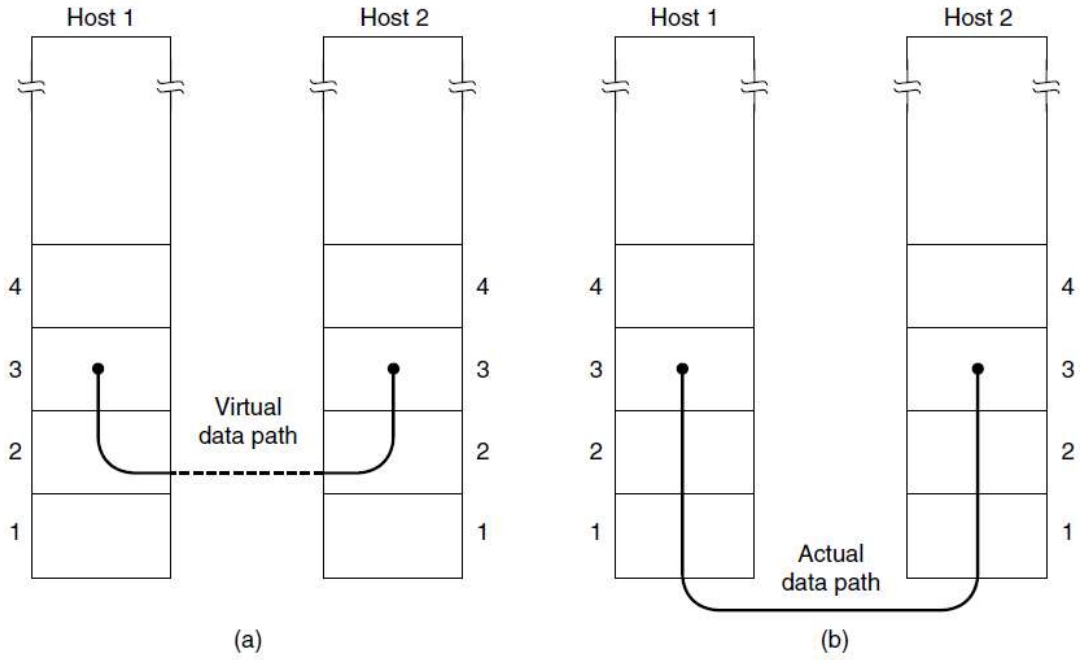


Figure 3-2. (a) Virtual communication. (b) Actual communication.

Framing

- Physical layer accepts a raw bit stream from DLL and tries to deliver it to destination
- May add some redundancy bit to reduce bit error rate (BER) to a tolerable level.
- This is not sufficient in noisy mediums like wireless. → no guarantee for error-free reception

Solution:

- Framing at DLL
- Each frame contains a check sum
- Receiver can detect/correct errors based on this checksum
- Challenge: Detect the start and the end of a frame

Framing

- the data link layer takes the packets it gets from the network layer and encapsulates them into **frames** for transmission.

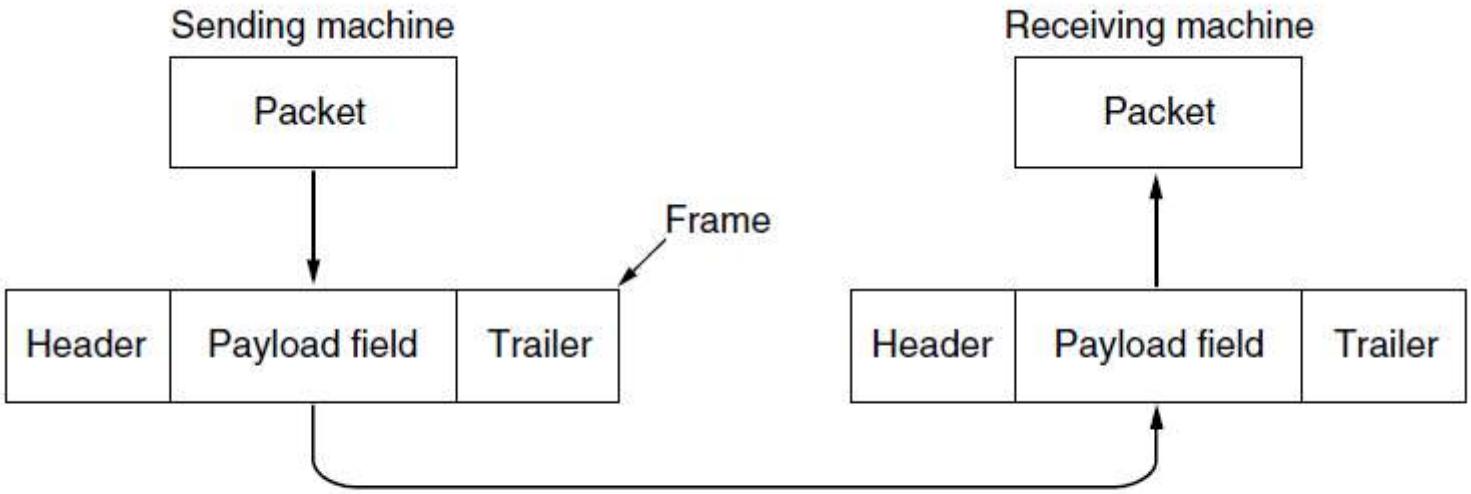


Figure 3-1. Relationship between packets and frames.

Framing

Methods

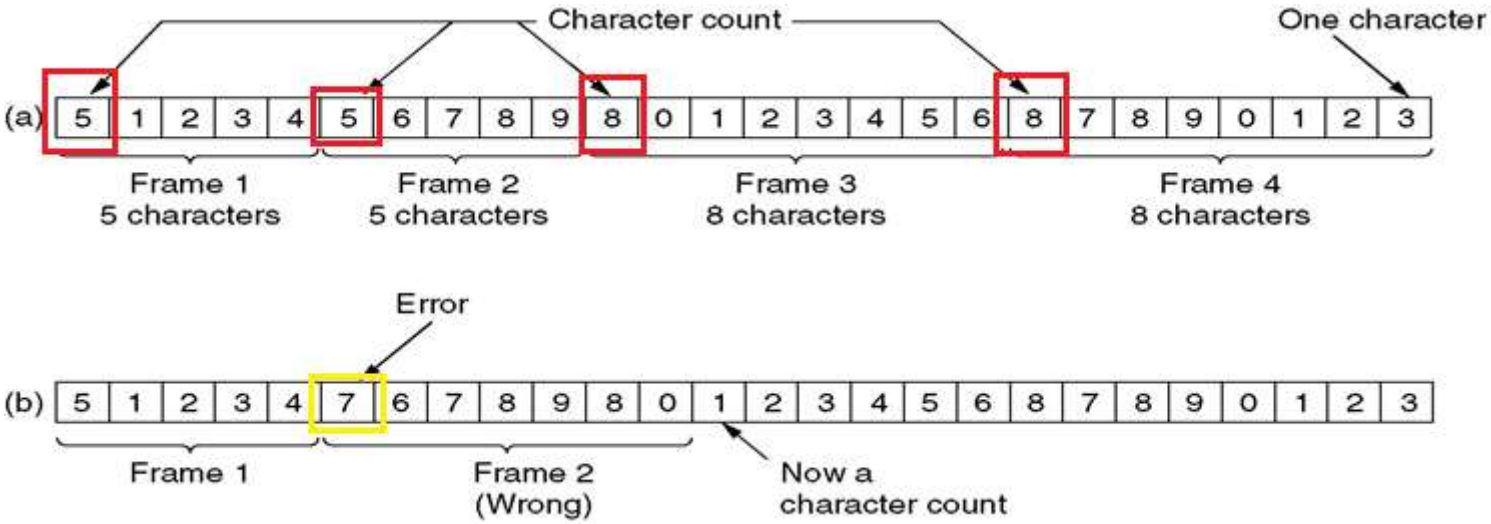
- Byte count
- Flag bytes with byte stuffing
- Flag bytes with bit stuffing

Byte count

- Disadvantages:
- Transmission error in count byte → losing synchronization
- Receiver can not locate the start of the next frame
- Rarely used

Byte count

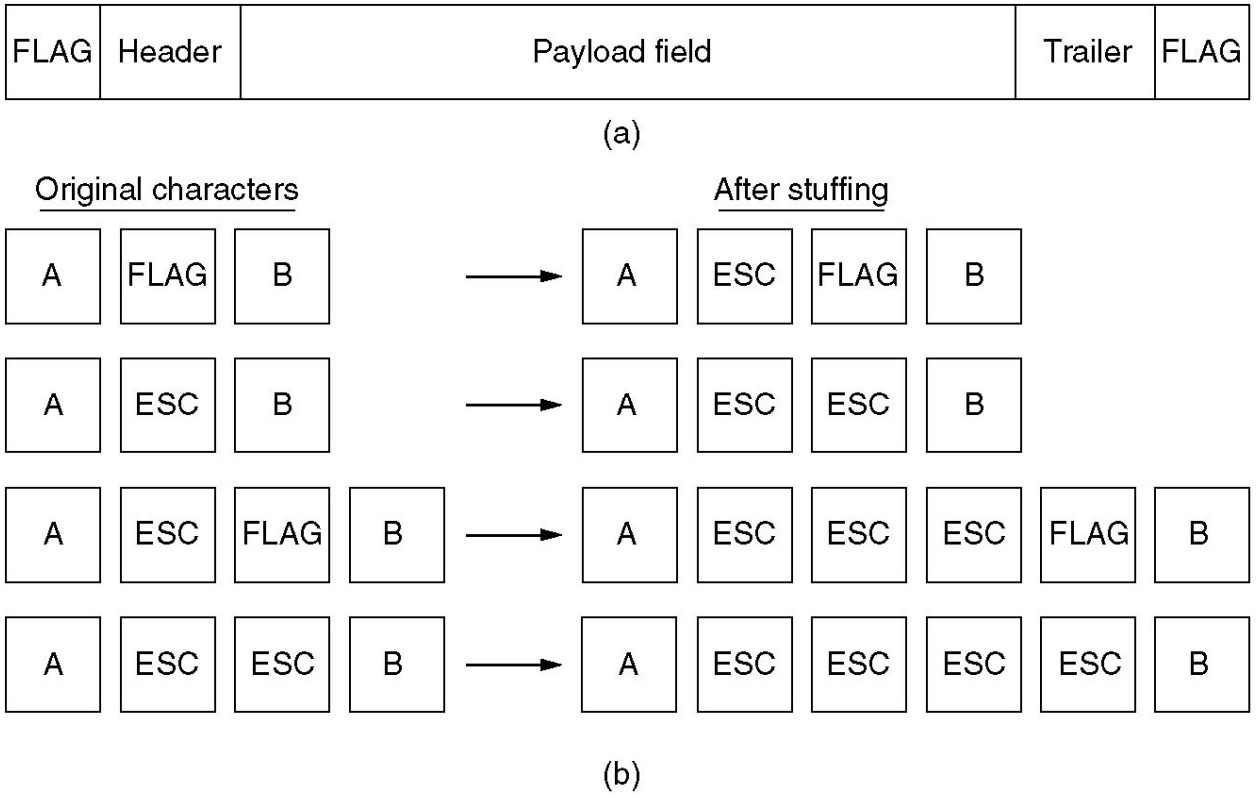
A character stream. (a) Without errors. (b) With one error.



Byte Stuffing

- Uses a special byte for delimiting the start and the end of the frame.
- This byte is called flag (***Fanion*** in french)
- If the byte flag occurs in the data stream, a special byte (for instance, ESC) is used to distinguish it from real flag
- The special byte may occur, it is used twice.
- At the receiver, the first special byte (stuffing byte) is removed.

Byte Stuffing



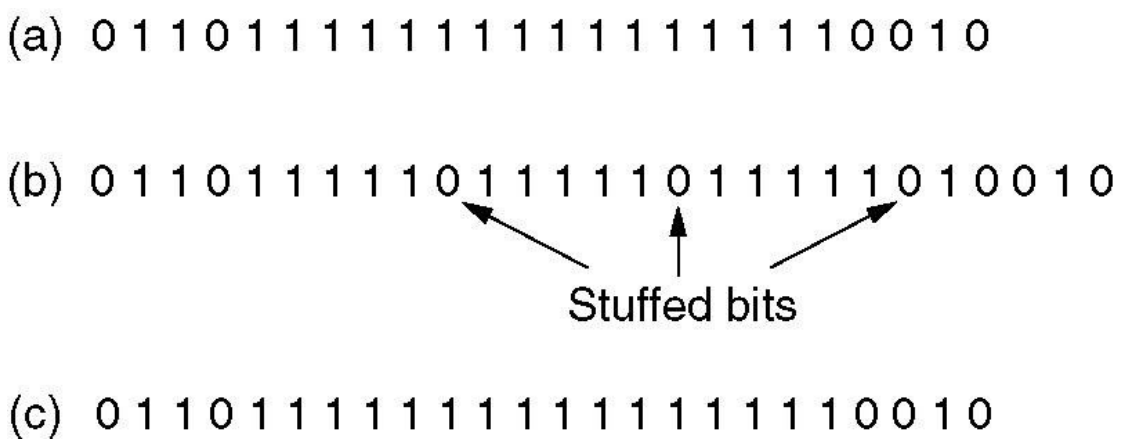
(a) A frame delimited by flag bytes.
(b) Four examples of byte sequences before and after stuffing.

Bit Stuffing

- With byte stuffing, for each byte flag in data → one additional byte

  → Additional control and hence less goodput (useful data)
- With bit stuffing, a flag byte is always used to delimit start and end of a frame
- Ex: 01111110 (0x7E) is used HDLC
- To avoid flag occurrence in data, **a bit is stuffed**.
- For instance, in HDLC, a **ZERO** bit is inserted after **5 ONES**
- At the reception it is destuffed

Bit Stuffing



Bit stuffing

(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.

Error Control

- **COST:** Error control means adding control bits to data streams
  → Overhead and loss of goodput
- **CHALLENGE:** Must decide to add enough redundant information and in function of medium quality.
- In fiber links, error rate is minimal
- In wireless links, error rate is very big
- 2 approaches:
  - **Error correcting codes** or (**FEC: Forward Error Correction**) is used in very noisy channels
  - **Error detecting codes** used in lower error-rate channels. It is used along with acknowledgements and retransmissions

Attention:

- **Neither approaches can handle all possible errors.**
- **Errors can occur on control bits as well**

Error Detecting Codes

Error Detecting Codes

- In reliable mediums such as fiber and high-quality copper, the error rate is much lower, so error correcting techniques may be expensive.

- Error detecting techniques are more efficient
- Examples:
  - Parity
  - Checksums
  - Cyclic Redundancy Checks

Parity

Single Bit Parity:

- Parity bit is appended to data.
- Even parity (number of 1s shall be even) = modulo 2 sum or XOR
- Odd parity: number of ones shall be odd
- Detect single bit errors
- Ex: 1011010
- Even parity : 10110100
- Odd parity: 10110101

Exclusive OR

- Exclusive OR is (XOR) is highly used in error control $\oplus$
- Recall
  - $0 \oplus 0 = 0$
  - $0 \oplus 1 = 1$
  - $1 \oplus 0 = 1$
  - $1 \oplus 1 = 0$
- To detect how many error-bits received, just $\oplus$ the two code words and count the number of ones
- Example:
  - Sent code: 11001100
  - Received code: 10011001
  - $\oplus$ ➔ 01010101
  - 4 bits differ !

Internet Checksum

Internet Checksum is a group of parity bits that are added to a message. It is calculated as follows.

Sender:

- **Treats segment contents as sequence of 16-bit words**
- **Calculates checksum: addition (1's complement sum) of 16-bit words**
- **Inserts the checksum at the end of IP packet**

Receiver:

- **Computes the checksum of received segment**
- **Check if computed checksum is equal to the received checksum:**
  - **NO - error detected**
  - **YES - no error detected. *But maybe errors nonetheless?***

CRC: Cyclic Redundancy Check

- M is the sequence of bits to send
- Choose a generator G(x) of degree r (known from the protocol used)
- Goal: choose r CRC bits, R, such that
  - <M,R> is exactly divisible by G (modulo 2)
  - The receiver knows G, divides the received <M,R> by G. If the remainder is non-zero: error detected!
  - It can detect all burst errors less than r+1 bits
- CRC is widely used in practice (ATM, HDLC, 802.3)

M: data bits to be sent	R: CRC bits

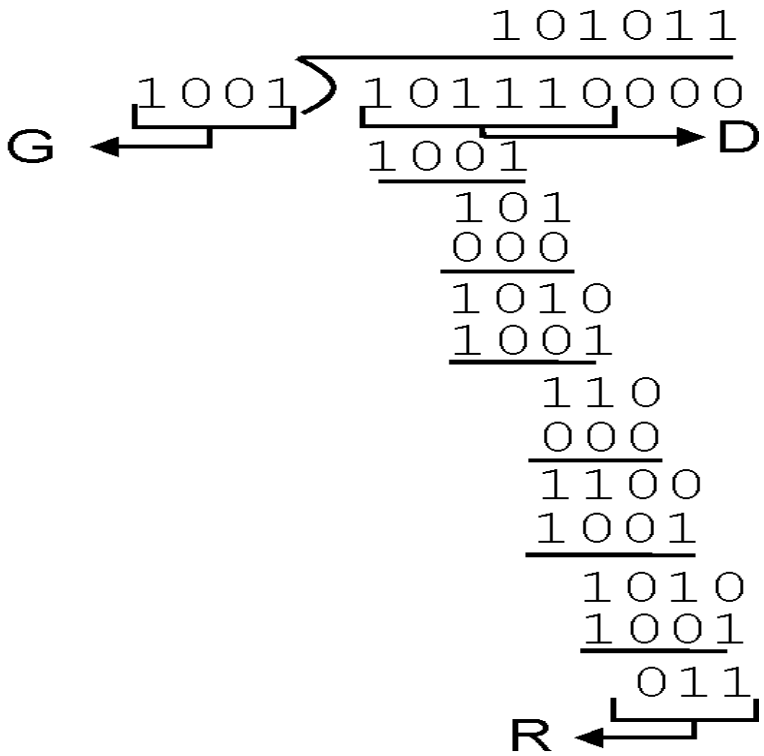
$$N = <M, R> = M*2^r \oplus R$$

CRC Example

Determine R

$$M^*2^r \oplus R = n^*G$$

R is the remainder of the division of M^*2^r by G



In the example $G(x) = x^3 + 1$
Sequence to send (data only): 101110
Sequence sent (data + CRC): 101110011

In Ethernet:
$G(x) = x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

Data Link Protocols

Multiple Access Protocols

- Single shared broadcast channel for all nodes
- If two or more simultaneous transmissions
  - interference
  - collision if node receives two or more signals at the same time

- Multiple access protocols run distributed algorithms to determine how nodes share the channel, i.e., determine when node can transmit
- Communication about channel sharing uses the channel itself!
  - no control channel for coordination

Multiple Access Protocols

Design Challenges:

Suppose that the channel rate is R bps

1. When only one node wants to transmit, it can send at rate R.
2. **When n nodes want to transmit, CAN each node send at average rate R/n ??**
   - **At least, it is the objective**
3. Fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots

DLL Protocol Classes

- Channel Partitioning
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate piece to node for exclusive use
  - Examples: TDMA, FDMA, CDMA
- “Taking turns”
  - nodes take turns, but nodes with more to send can take longer turns
  - Example: token ring
- Random Access
  - Channel not divided → Simultaneous transmissions yield collisions
  - Must “recover” from collisions
  - Examples: Ethernet, WiFi

Random Access Protocols

- When a node has a frame to send
  - It transmits at full channel data rate R.
  - No *a priori* coordination among nodes
- If two or more nodes transmit simultaneously ➜ "collision" occurs
- Random access MAC protocols specify:
  - how to detect collisions (and/or possibly avoids)
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA - ALOHA
  - Carrier Sense Multiple Access / Collision Detection: **CSMA/CD** (used in Ethernet)
  - Carrier Sense Multiple Access / Collision Avoidance: **CSMA/CA** (used by WiFi)

Carrier Sense Multiple Access: CSMA

- A node listens to the channel before transmitting:
- If channel is sensed idle, it transmits the entire frame
- If channel is sensed busy, it defers transmission

Collisions may (will) occur:

- The propagation delay on the shared medium means that two nodes may not hear each other's transmission
- Collision probability is a function of:
  - Distance
  - Propagation delay
- Collisions → Time wasted (all the frame is transmitted)
- Collisions → reduced capacity

CSMA/CD (Collision Detection)

CSMA/CD:

- A node listens to the channel before transmitting:
- If the channel is sensed idle, it transmits entire frame
- Collisions are *detected* within short time
- Colliding transmissions are aborted and thus reducing channel wastage

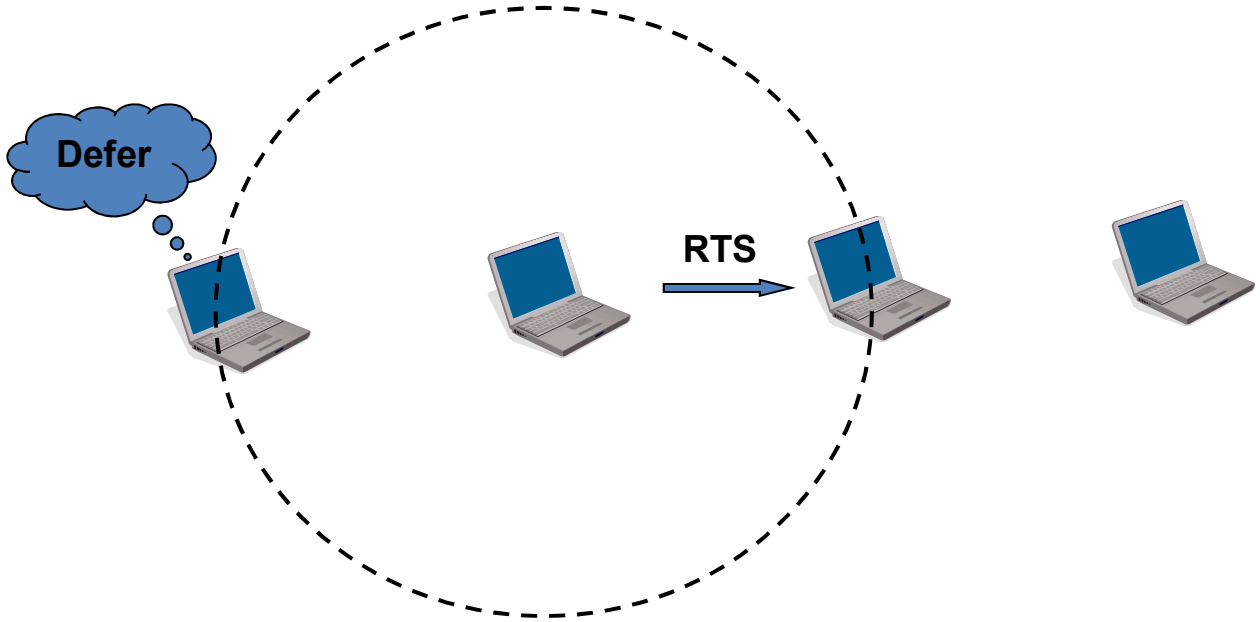
- Easy in wired LANs:
  - Can be achieved by measuring signal strengths, comparing transmitted and received signals

- Difficult in wireless LANs:
  - received signal strength combined with local transmission strength

Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)

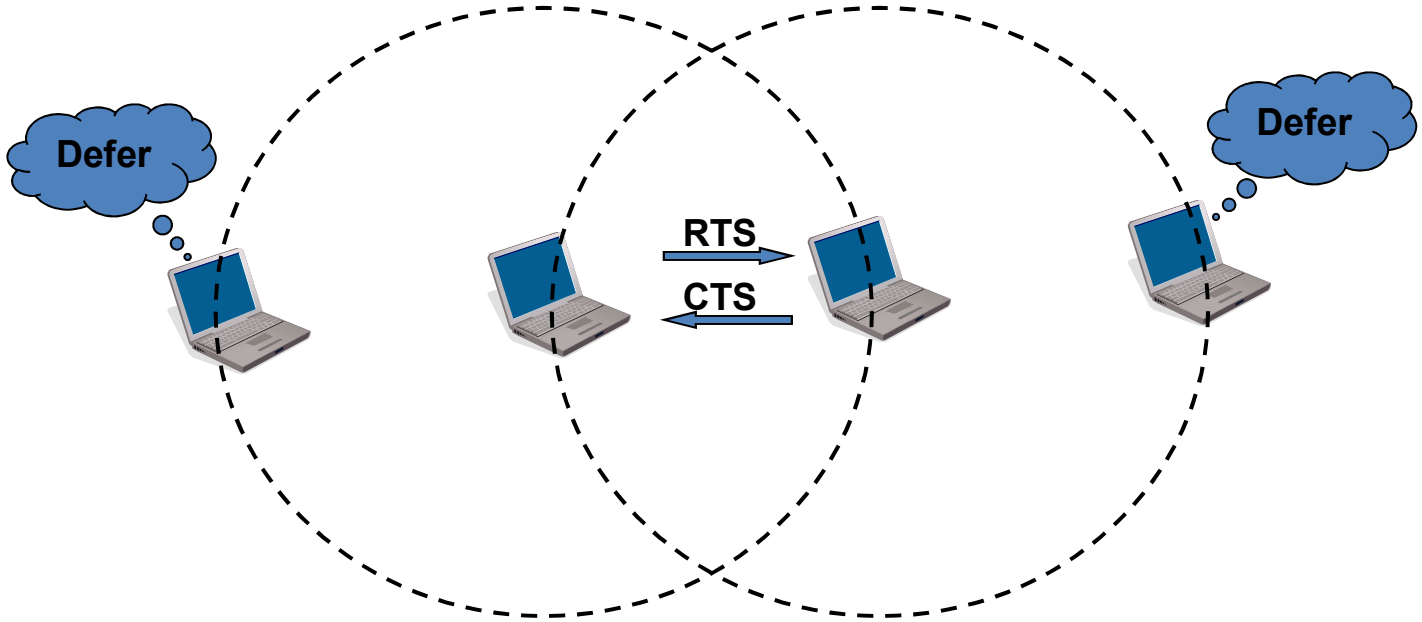
- Procedure
  - Similar to CSMA but instead of sending packets control frames are exchanged
  - RTS = request to send
  - CTS = clear to send
  - DATA = actual packet
  - ACK = acknowledgement

RTS/CTS dialog (1)



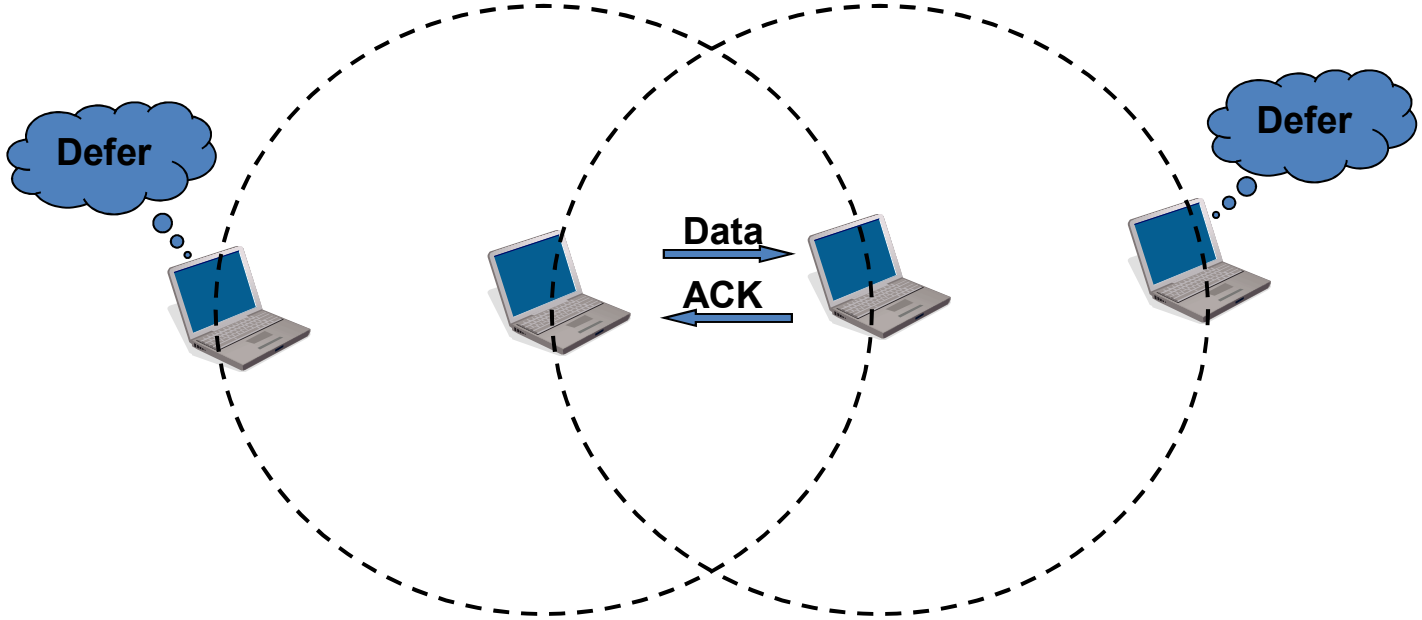
Any node hearing this RTS will defer medium access

RTS/CTS dialog (2)



Any node hearing this CTS will defer medium access

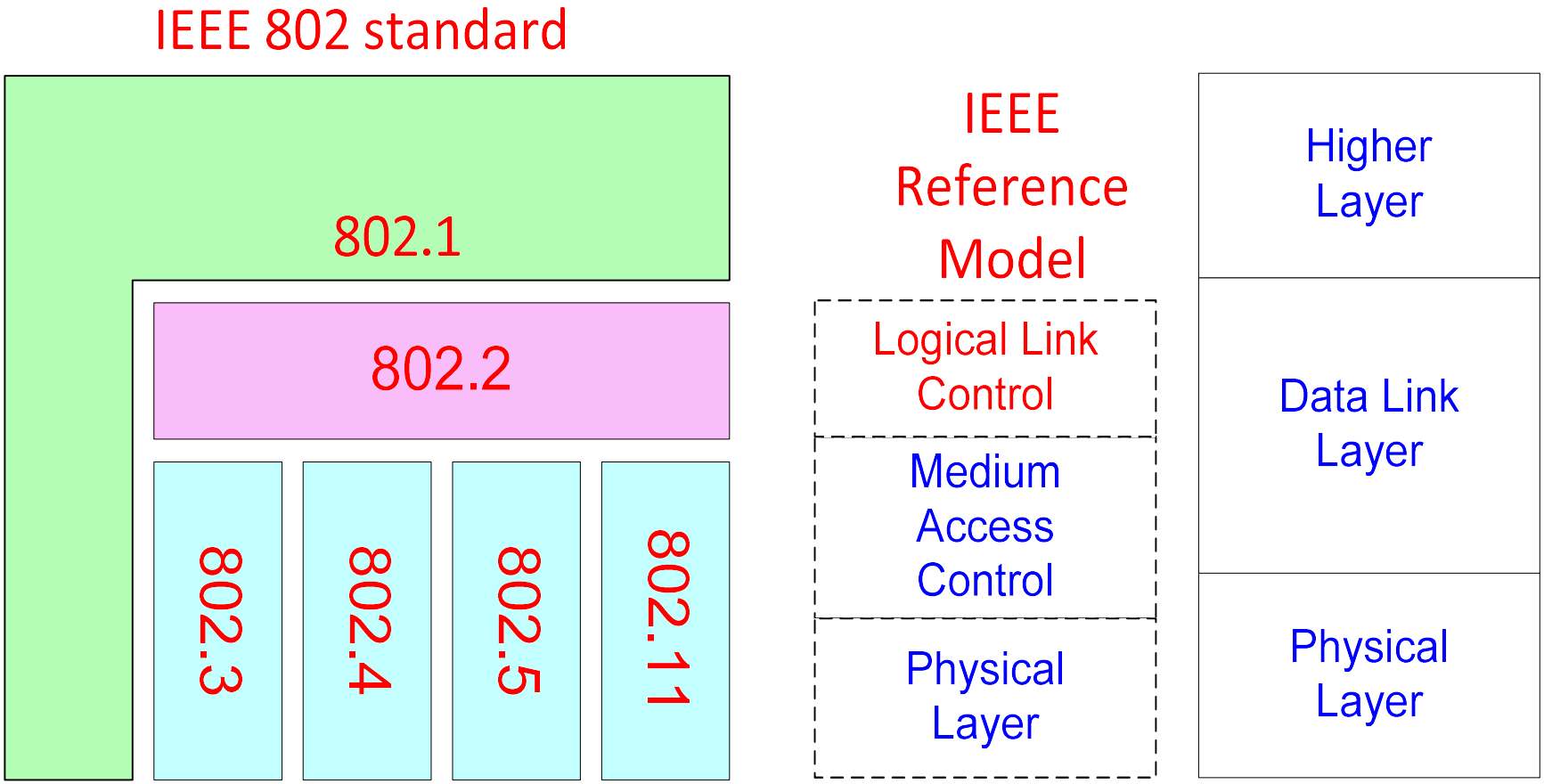
RTS/CTS/DATA/ACK dialog



Ethernet

Ethernet in IEEE 802 Standards

- IEEE 802 is a family of standards for LANs, which defines an LLC and several MAC sublayers



Ethernet in IEEE 802 Standards

- IEEE 802.1 : Interworking
- IEEE 802.2 : LLC
- MAC
  - 802.3 CSMA/CD (Ethernet)
  - 802.4 Token Bus
  - 802.5 Token Ring
  - 802.6 Metropolitan Area Network
  - 802.7 Broadband Technical Advisory Group
  - 802.8 Fiber Optic Technical Advisory Group
  - 802.9 Integrated Voice and Data Networks
  - 802.11? Wireless LAN « Wi-Fi » (plusieurs normes, ?={a,b,g})
  - 802.15 Wireless PAN « BlueTooth »

Ethernet in IEEE 802 Standards

Number	Topic
802.1	Overview and architecture of LANs
802.2 ↓	Logical link control
802.3 *	Ethernet
802.4 ↓	Token bus (was briefly used in manufacturing plants)
802.5	Token ring (IBM's entry into the LAN world)
802.6 ↓	Dual queue dual bus (early metropolitan area network)
802.7 ↓	Technical advisory group on broadband technologies
802.8 †	Technical advisory group on fiber optic technologies
802.9 ↓	Isochronous LANs (for real-time applications)
802.10 ↓	Virtual LANs and security
802.11 *	Wireless LANs (WiFi)
802.12 ↓	Demand priority (Hewlett-Packard's AnyLAN)
802.13	Unlucky number; nobody wanted it
802.14 ↓	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth, Zigbee)
802.16 *	Broadband wireless (WiMAX)
802.17	Resilient packet ring
802.18	Technical advisory group on radio regulatory issues
802.19	Technical advisory group on coexistence of all these standards
802.20	Mobile broadband wireless (similar to 802.16e)
802.21	Media independent handoff (for roaming over technologies)
802.22	Wireless regional area network

Figure 1-38. The 802 working groups. The important ones are marked with *. The ones marked with ↓ are hibernating. The one marked with † gave up and disbanded itself.

Ethernet

Ethernet is the most used wired LAN technology:

- Cheap 15 € for NIC (Network Interface Card)
- First widely used LAN technology
- Simpler and cheaper than token LANs and ATM
- Speeds: 10 Mbps – 100 Gbps
- connectionless: No handshaking between sending and receiving NICs
- unreliable: receiving NIC doesn't send ACKs
  - Stream of datagrams passed to network layer can have gaps (missing datagrams)
  - Gaps will be managed by TCP if it is used at transport layer
  - otherwise, application will see gaps (UDP case)
- Ethernet's MAC protocol: CSMA/CD

Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 (Start of Frame **SOF**)
- used to synchronize receiver, sender clock rates
- The last 2 bits indicates the start of the frame…

Field name	**Preamble**	**SOF**	**DA**	**SA**	**DL / TYPE**	**Data (LPDU)**	**Pad**	**FCS (CRC)**
Size(B)	7	1	6	6	2	0 - 1500	0 - 46	4
			Minimum frame size : 64, max. : 1518					

Ethernet Frame Structure

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
  - Destination address:
    - First bit 0 → Unicast
    - First bit 1 → Multicast (the destination is a group of stations)
    - ALL 1 → Broadcast (the destination is all stations)
- MAC addresses:
  - 3 bytes for **OUI** (Organizationally Unique identifier): Manufacturer
  - Each NIC has a unique identifier (address)

Field name	**Preamble**	**SOF**	**DA**	**SA**	**DL / TYPE**	**Data (LPDU)**	**Pad**	**FCS (CRC)**
Size(B)	7	1	6	6	2	0 - 1500	0 - 46	4
			Minimum frame size : 64, max. : 1518					

Ethernet Frame Structure

- **Type**:
  - In Ethernet (before 1997)
  - indicates higher layer protocol (mostly IP but others possible, ARP, RARP)
  - It specifies which process to five the frame to (may have multiple network protocols)
  - Ex: 0x0800 means data contains IPv4 packet, 0x806 for ARP, 0x835: RARP
  - In 802.3, the field contains the length of the frame**.** The type is included on 802 LLC header.
  - Compatibility is still ensured since all data type values are > 0X600 (1536)
- **CRC**:
  - checked at receiver, if error is detected, frame is dropped
  - CRC 32 bits calculated over all fields except preamble
- **Pad** is used to ensure a minimum frame size.

Field name	**Preamble**	**SOF**	**DA**	**SA**	**DL / TYPE**	**Data (LPDU)**	**Pad**	**FCS (CRC)**
Size(B)	7	1	6	6	2	0 - 1500	0 - 46	4
			Minimum frame size : 64, max. : 1518					

Ethernet Frame Structure

- Pad is used to ensure a minimum frame size.
- Minimun size:
  - 10 Mbps LAN
  - 2500 Meters
  - 4 repeaters

  → round-trip time = 50 micro seconds
- At 10 Mbps, a bit takes 0.1 micro sec,

→ 500 bits is the smallest frame that is guaranteed to work

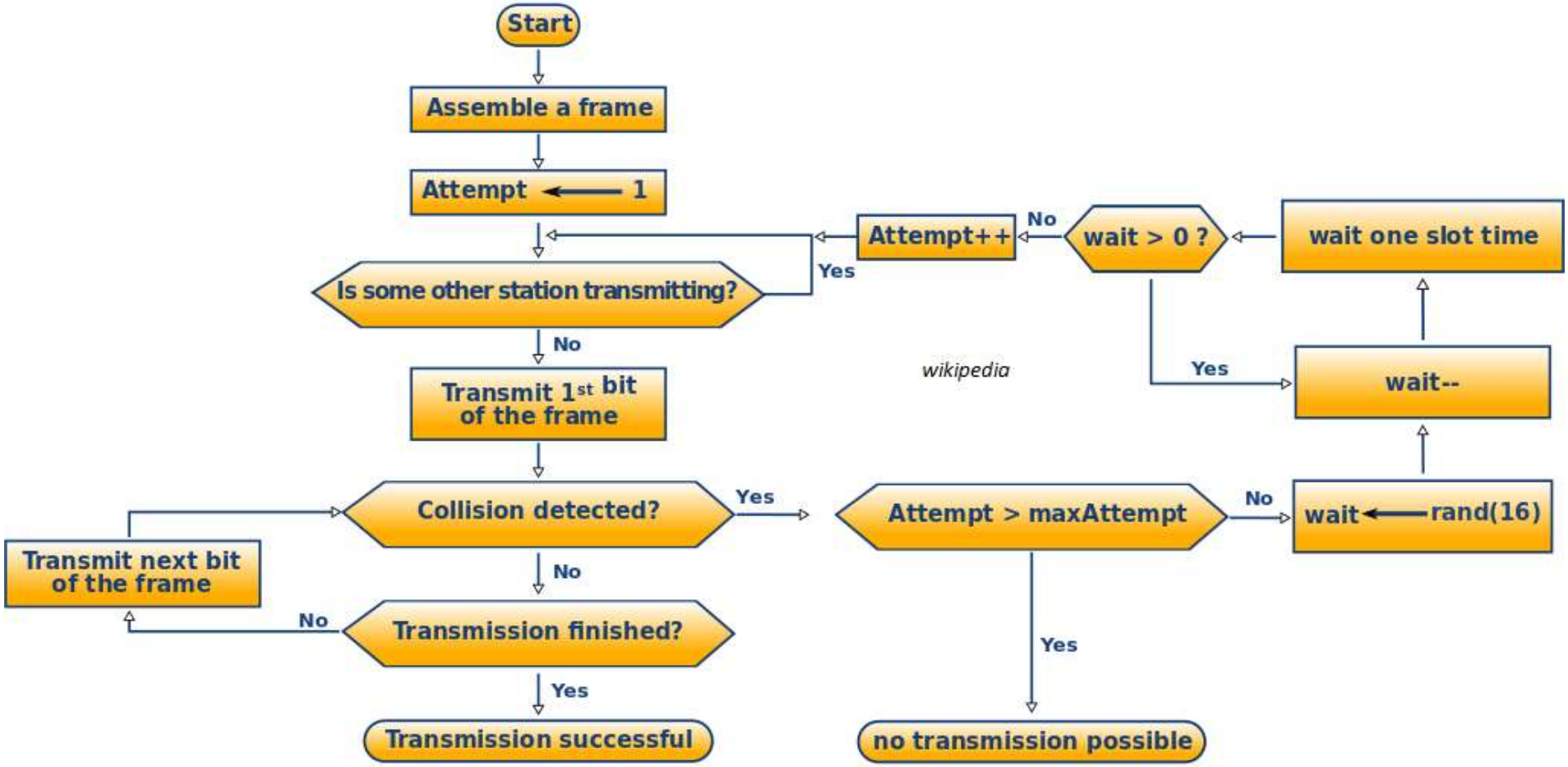
- 500 ~ 512 = 64 bytes

Field name	**Preamble**	**SOF**	**DA**	**SA**	**DL / TYPE**	**Data (LPDU)**	**Pad**	**FCS (CRC)**
Size(B)	7	1	6	6	2	0 - 1500	0 - 46	4
			Minimum frame size : 64, max. : 1518					

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If, however, NIC senses channel busy, it waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters **exponential backoff**:
   1. after *m*th collision, NIC randomly chooses *K* from {0,1,2,…,2^m-1}.
   2. NIC waits K*512 bit times, returns to Step 2

Ethernet CSMA/CD algorithm



Ethernet CSMA/CD algorithm

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- First collision: choose K from {0,1}; delay is K*512 bit transmission times
- After second collision: choose K from {0,1,2,3}…
- After ten collisions, choose K from {0,1,2,3,4,…,1023}

Ethernet CSMA/CD algorithm

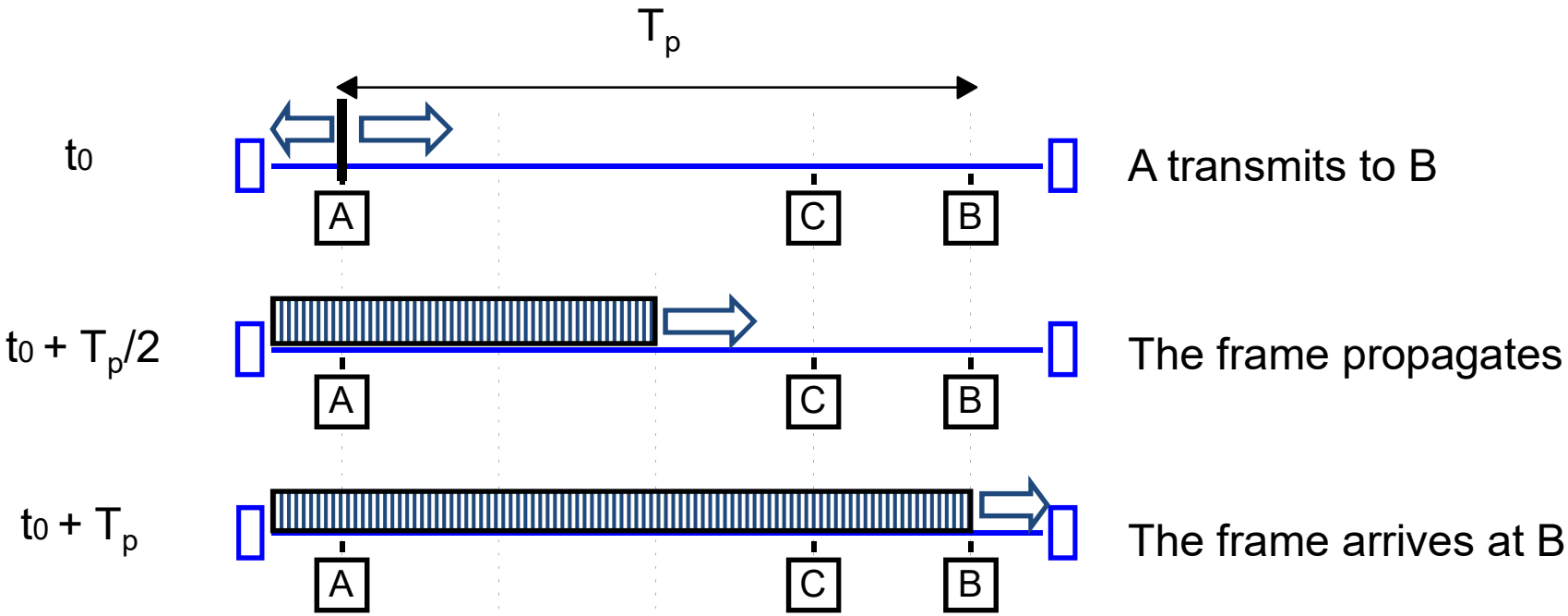
Jam Signal:

Make sure all other transmitters are aware of collision;
48 bytes

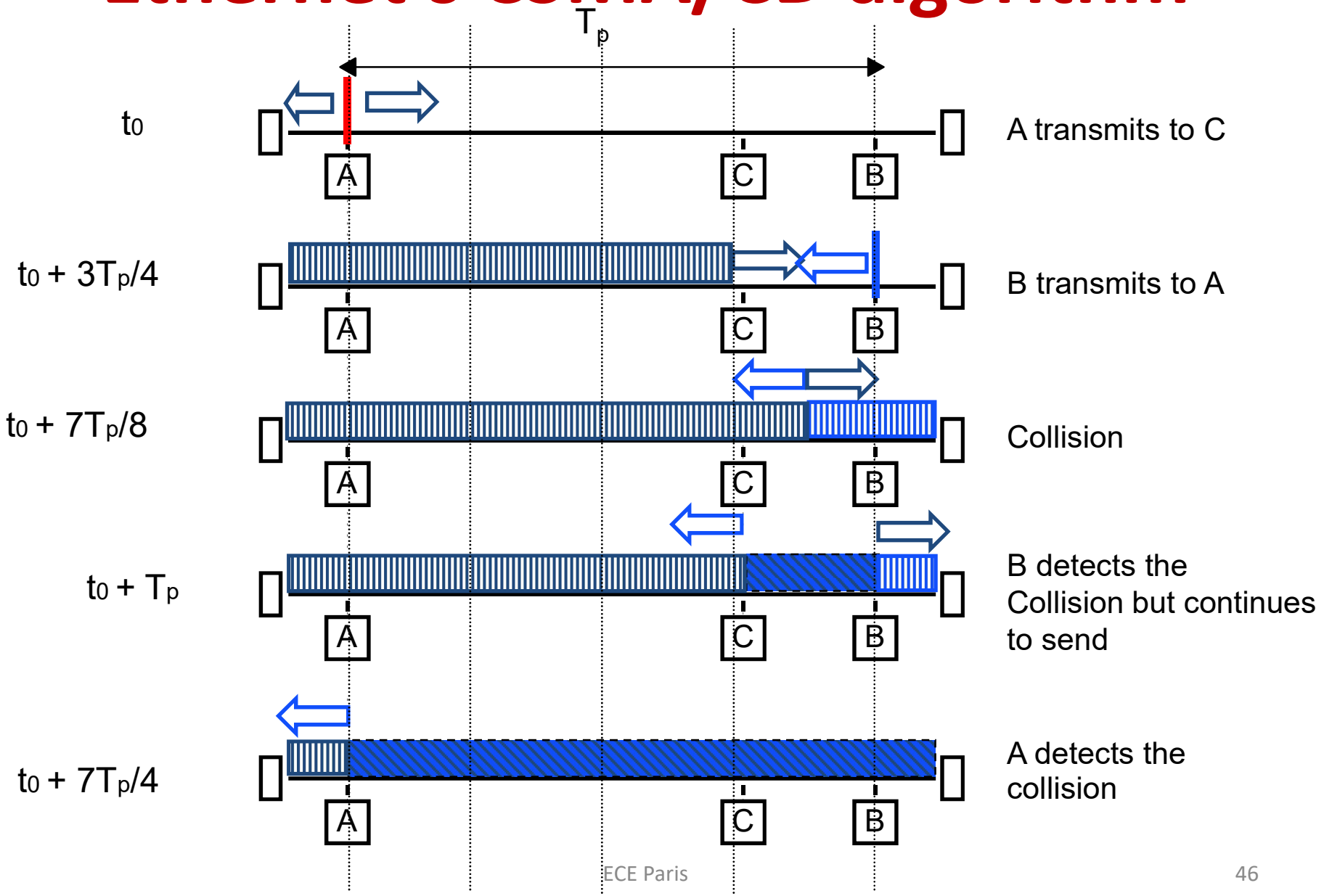
Bit time:

0.1 μs for 10 Mbps Ethernet ;
for K=1023, wait time is about 50 ms

Ethernet's CSMA/CD algorithm



Ethernet's CSMA/CD algorithm



Ethernet's CSMA/CD algorithm

- Minimal duration (size) of a frame
  - $2\ T_p$
    - Size in bits: $2T_pD$
      - Standardized : 512 bits (64 bytes)
- Waiting time before retransmission:
  - Random select of n ;
    - $0 \leq n < 2^{\min(i,10)}$
  - Wait *n*slotTime*

Metric Units

Prefix	Symbol	Multiplier Numerical	Exponential
deci	d	0.1	10^{-1}
centi	c	0.01	10^{-2}
milli	m	0.001	10^{-3}
micro	μ	0.000001	10^{-6}
nano	n	0.000000001	10^{-9}
pico	p	0.000000000001	10^{-12}
femto	f	0.000000000000001	10^{-15}
atto	a	0.000000000000000001	10^{-18}
zepto	z	0.000000000000000000001	10^{-21}
yocto	y	0.000000000000000000000001	10^{-24}

The principal metric prefixes

Metric Units

Prefix	Symbol	Multiplier Numerical	Exponential
yotta	Y	1,000,000,000,000,000,000,000,000	10^{24}
zetta	Z	1,000,000,000,000,000,000,000	10^{21}
exa	E	1,000,000,000,000,000,000	10^{18}
peta	P	1,000,000,000,000,000	10^{15}
tera	T	1,000,000,000,000	10^{12}
giga	G	1,000,000,000	10^{9}
mega	M	1,000,000	10^{6}
kilo	k	1,000	10^{3}
hecto	h	100	10^{2}
deca	da	10	10^{1}
no prefix means: 1			10^{0}

The principal metric prefixes

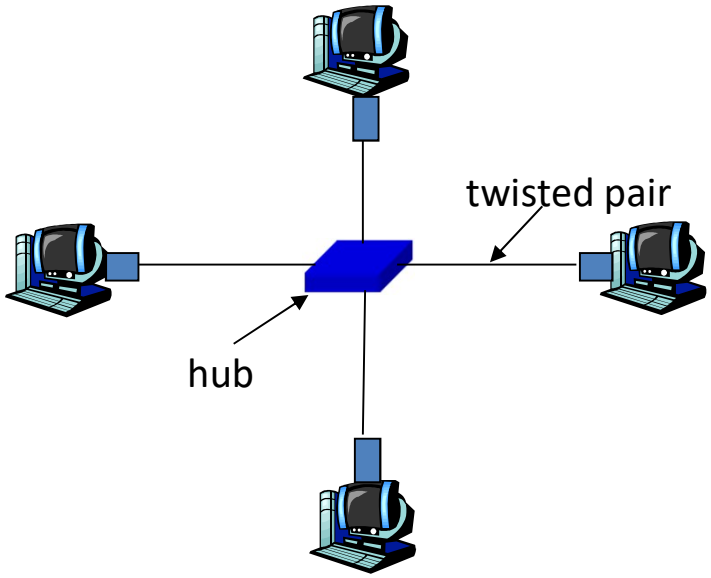
Ethernet cables

Ethernet Type	Bandwidth	Cable Type	Maximum Distance
10BASE-T	10Mbps	Cat3/Cat5 UTP	100 m
100BASE-TX	100Mbps	Cat5 UTP	100 m
100BASE-TX	200Mbps	Cat5 UTP	100 m
100BASE-FX	100Mbps	Multimode fiber	400 m
100BASE-FX	200Mbps	Multimode fiber	2 km
1000BASE-T	1Gbps	Cat5e UTP	100 m
1000BASE-TX	1Gbps	Cat6 UTP	100 m
1000BASE-SX	1Gbps	Multimode fiber	550 m
1000BASE-LX	1Gbps	Single-mode fiber	2 km
10GBASE-T	10Gbps	Cat6a/Cat7 UTP	100 m
10GBASE-SX4	10Gbps	Multimode fiber	550 m
10GBASE-LX4	10Gbps	Single-mode fiber	2 km

DLL Devices

Hub

- Acts just like a connector: bits coming in one link go out to *all* other links at same rate
- All nodes connected to hub can collide with one another → ALL in same collision domain
- No frame buffering at the hub
- No CSMA/CD at hub: host NICs detect collisions



Switch

- Link-layer device: smarter than hubs, take *active* role
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- *Transparent*
  - hosts are unaware of presence of switches
- *Plug-and-play, self-learning*
  - switches do not need to be configured

Switch advantages vs. Hub

- Almost no collisions. Each port is a collision domain.

  → more capacity

- Simultaneous transmissions. The backplane of a switch is very high speed

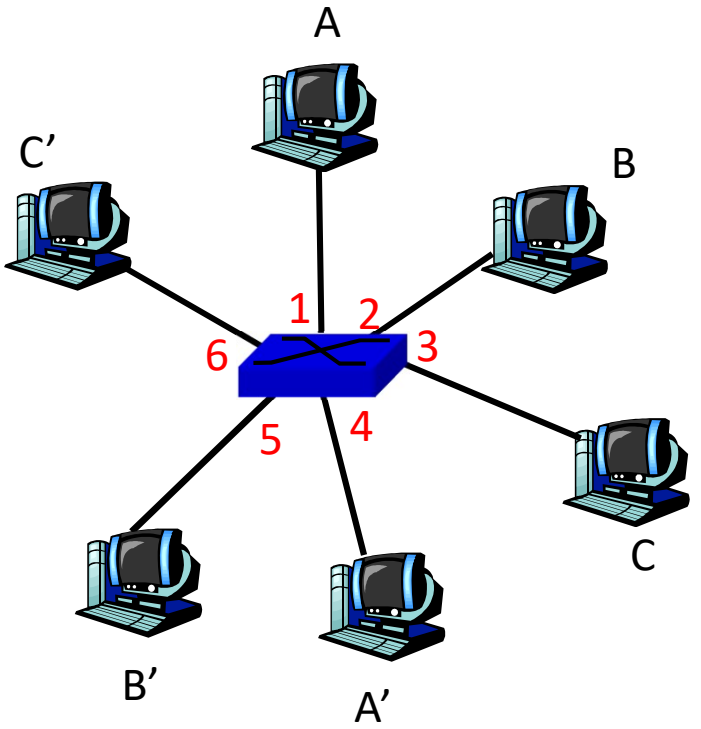
  → more capacity

- Security: Data sent from A to B can not be captured at D or C because they cannot receive it as in hub case

Data Link Frame Forwarding (Switching)

Switch Table

- Each switch has a switching table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- It looks like a routing table!
- A routing protocol-like is needed!



switch with six interfaces (1,2,3,4,5,6)

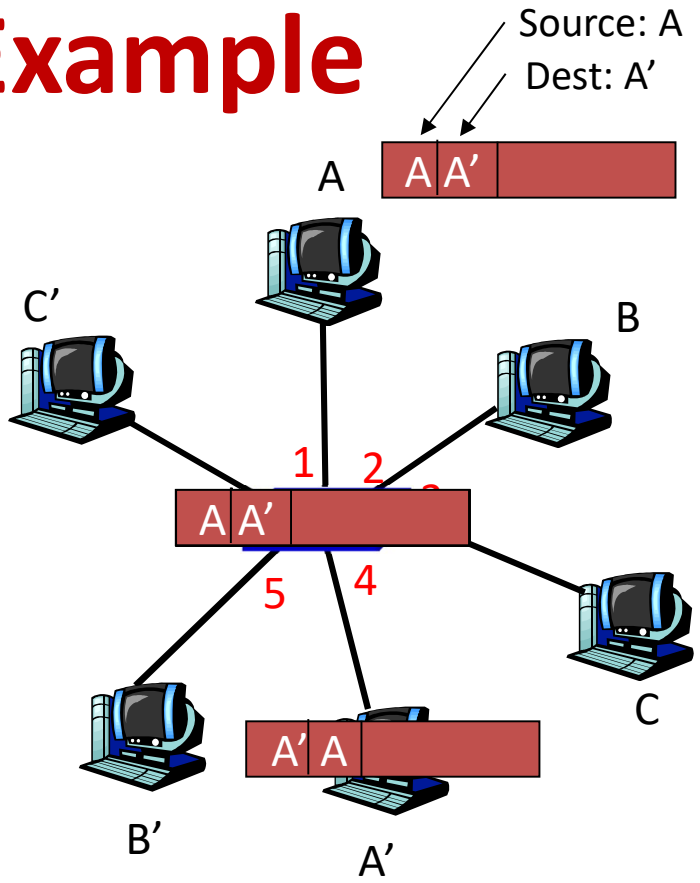
Frame forwarding Table Construction

When frame arrived at switch:

1. Record link associated with sending host
2. Index switch table using MAC destination address
3. **if** entry found for destination
   then {
      if destination on segment from which frame arrived
      then drop the frame
      else forward the frame on interface indicated
   }
   else flood

Switching Example

- frame destination unknown: flood
- destination A location known: selective send



MAC addr	interface	TTL
A	1	60
A'	4	60

Switch table (initially empty)

References

[1] Andrew S. Tanenbaum, David J. Wetherall, *Computer Network*

[2] Jim Kurose, Keith Ross, Computer Networking: A Top Down Approach