Computer Networks & IOT (18B11CS311)

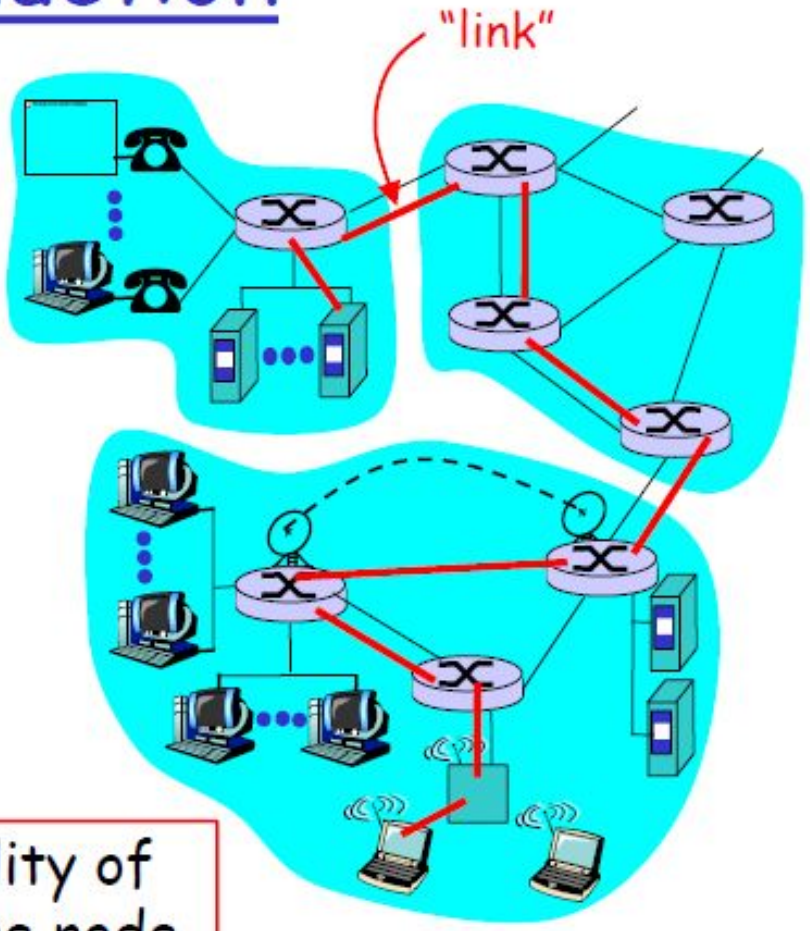
Even Semester_2024

DataLink Layer

Link Layer: Introduction

Some terminology:

- hosts and routers are **nodes** (bridges and switches too)
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- 2-PDU is a **frame**, encapsulates datagram



data-link layer has responsibility of transferring datagram from one node to adjacent node over a link

Link layer: context

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide rdt over link

transportation analogy

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

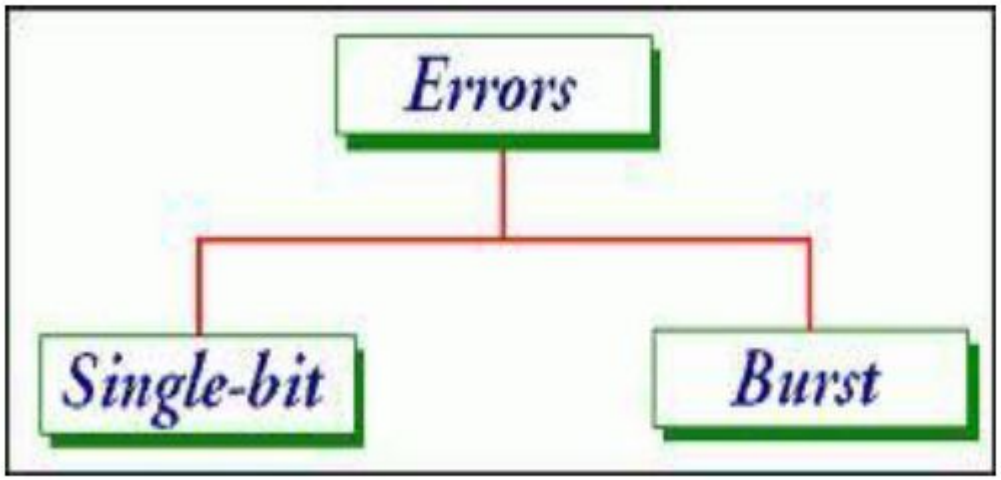
Link Layer Services

- Framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - 'physical addresses' used in frame headers to identify source, dest
    - different from IP address!
- Reliable delivery between adjacent nodes
  - we learned how to do this already (chapter 3)!
  - seldom used on low bit error link (fiber, some twisted pair)
  - wireless links: high error rates
    - Q: why both link-level and end-end reliability?

Link Layer Services (more)

- *Flow Control:*
  - pacing between adjacent sending and receiving nodes
- *Error Detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- Error Correction:
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *Half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

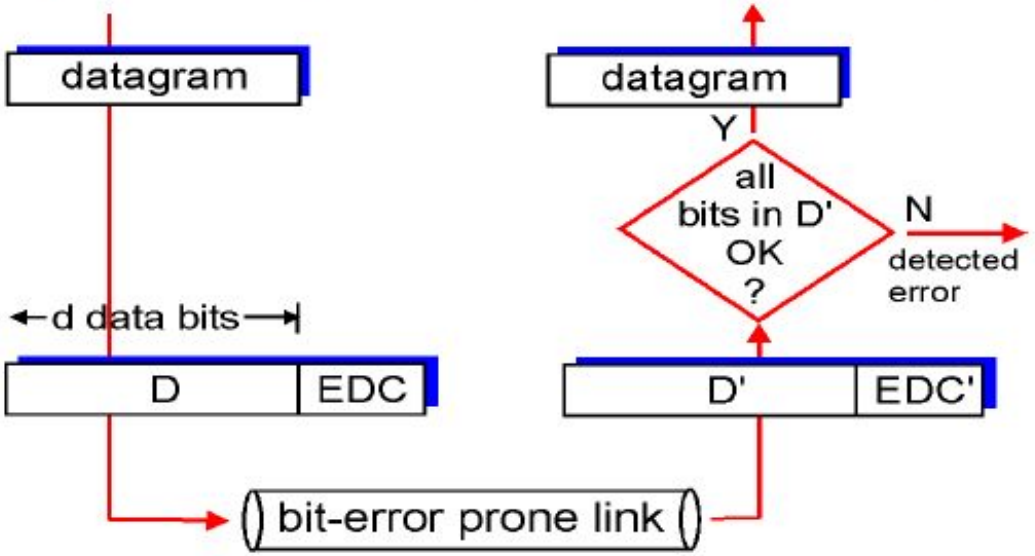
Error detection and correction


Errors
Single-bit
Burst

Error Detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

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

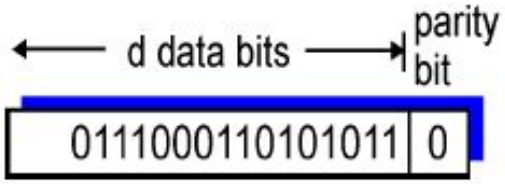


- Some popular techniques for error detection are:
  1. Simple Parity check
  2. Two-dimensional Parity check
  3. Checksum
  4. Cyclic redundancy check

Parity Checking

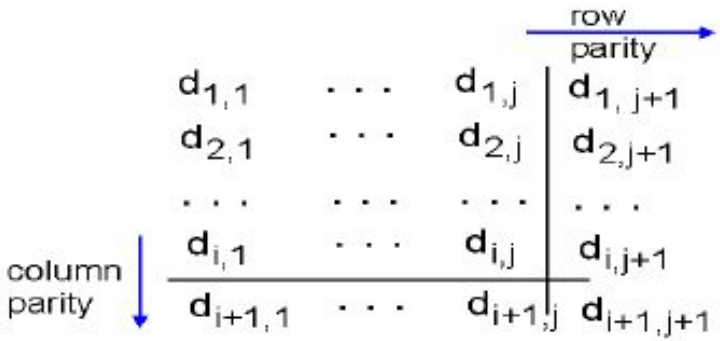
Single Bit Parity:

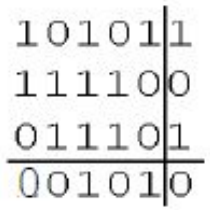
Detect single bit errors



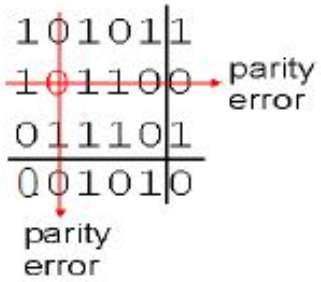
Two Dimensional Bit Parity:

Detect *and correct* single bit errors





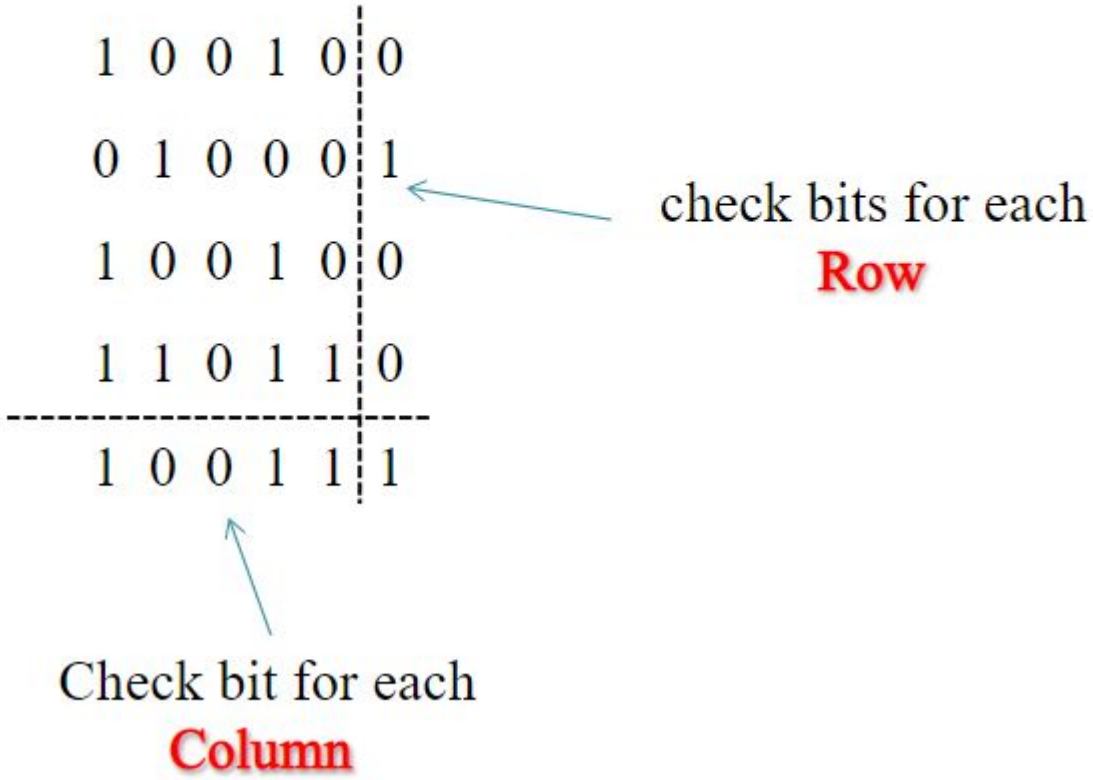
no errors



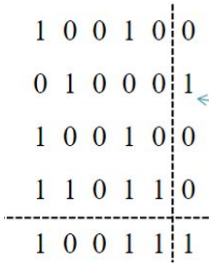
correctable single bit error

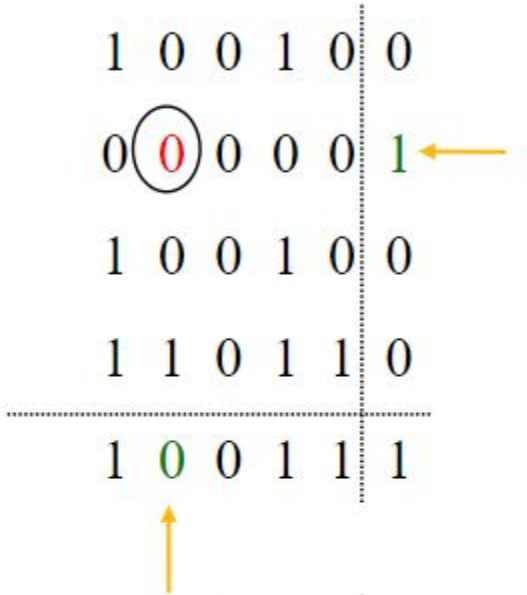
Two-dimensional Parity Check Code

Horizontal Vertical Parity Check

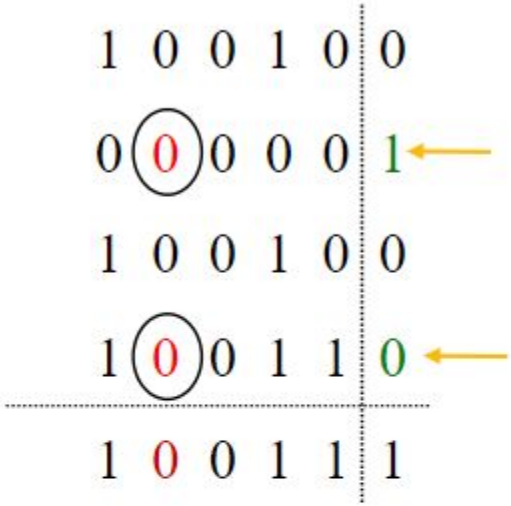


Multiple Errors

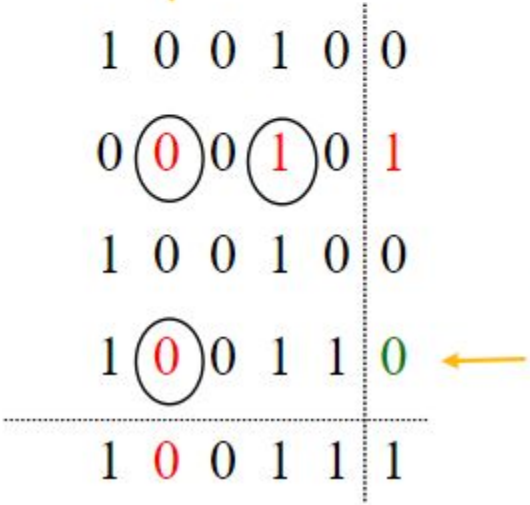




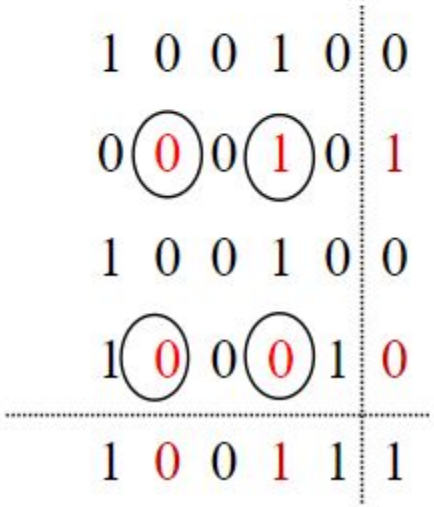
One error



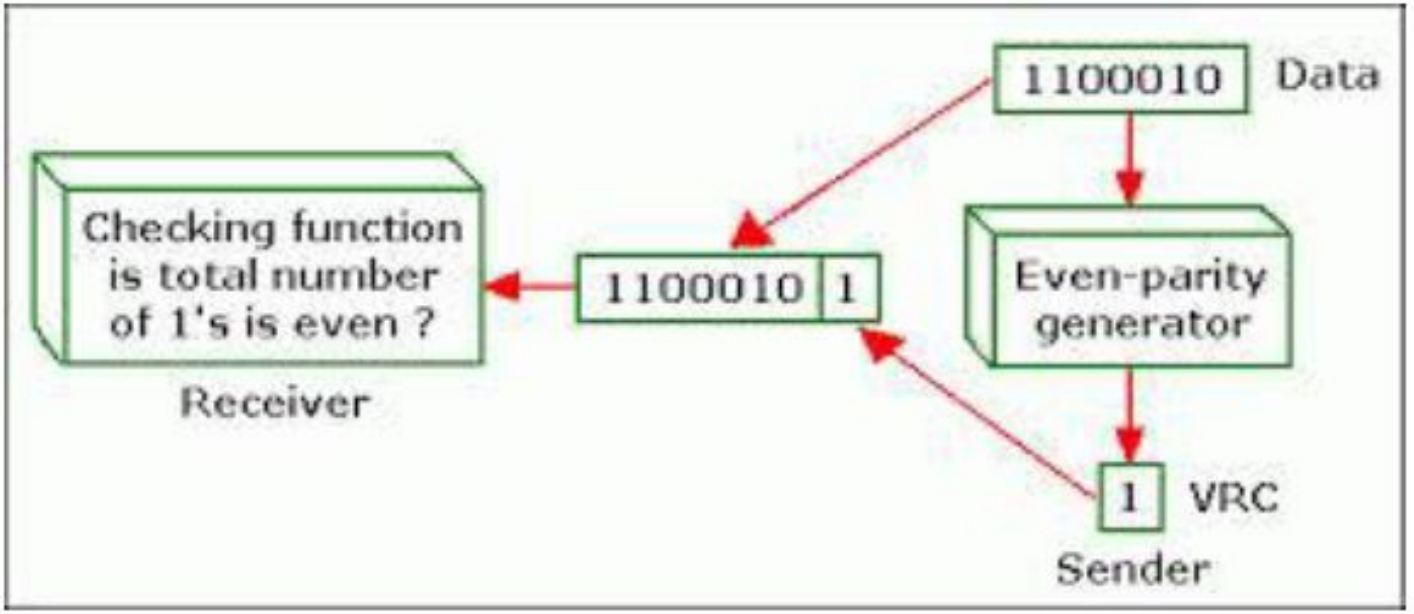
Two errors

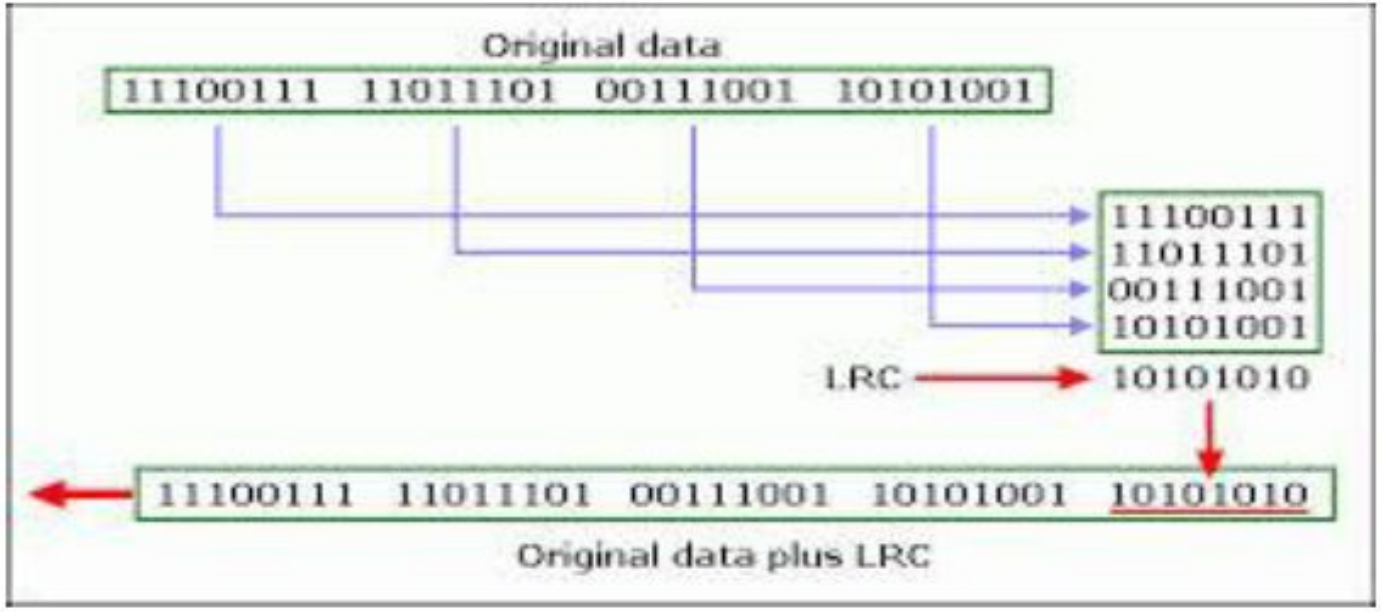


Three errors



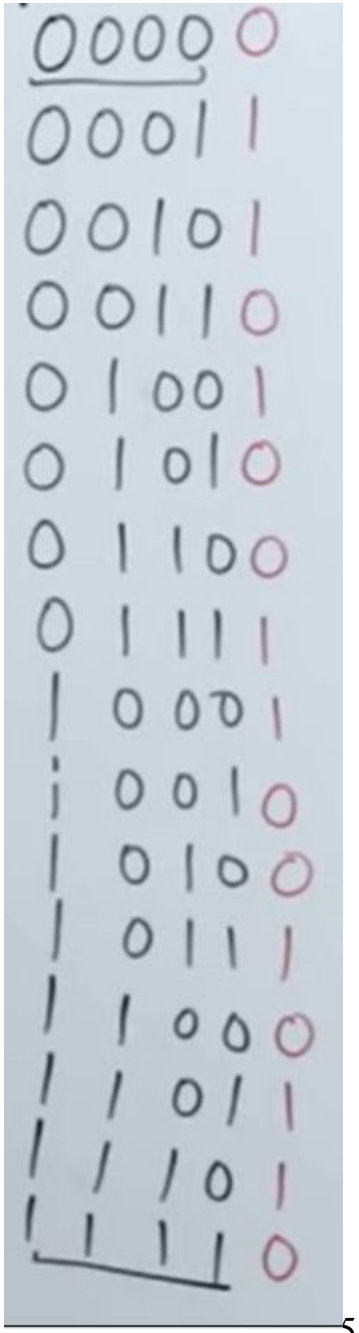
Four errors


1100010
Data
Checking function is total number of 1's is even ?
1100010 1
Even-parity generator
Receiver
1
VRC
Sender


Original data
11100111 11011101 00111001 10101001
11100111
11011101
00111001
10101001
LRC
10101010
11100111 11011101 00111001 10101001 10101010
Original data plus LRC

Example: C(5,4)

- Consider data words
- Compute hamming distance (XOR)
- Minimum hamming distance h
- (h-1) bits of error can be detected.



Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

checksum

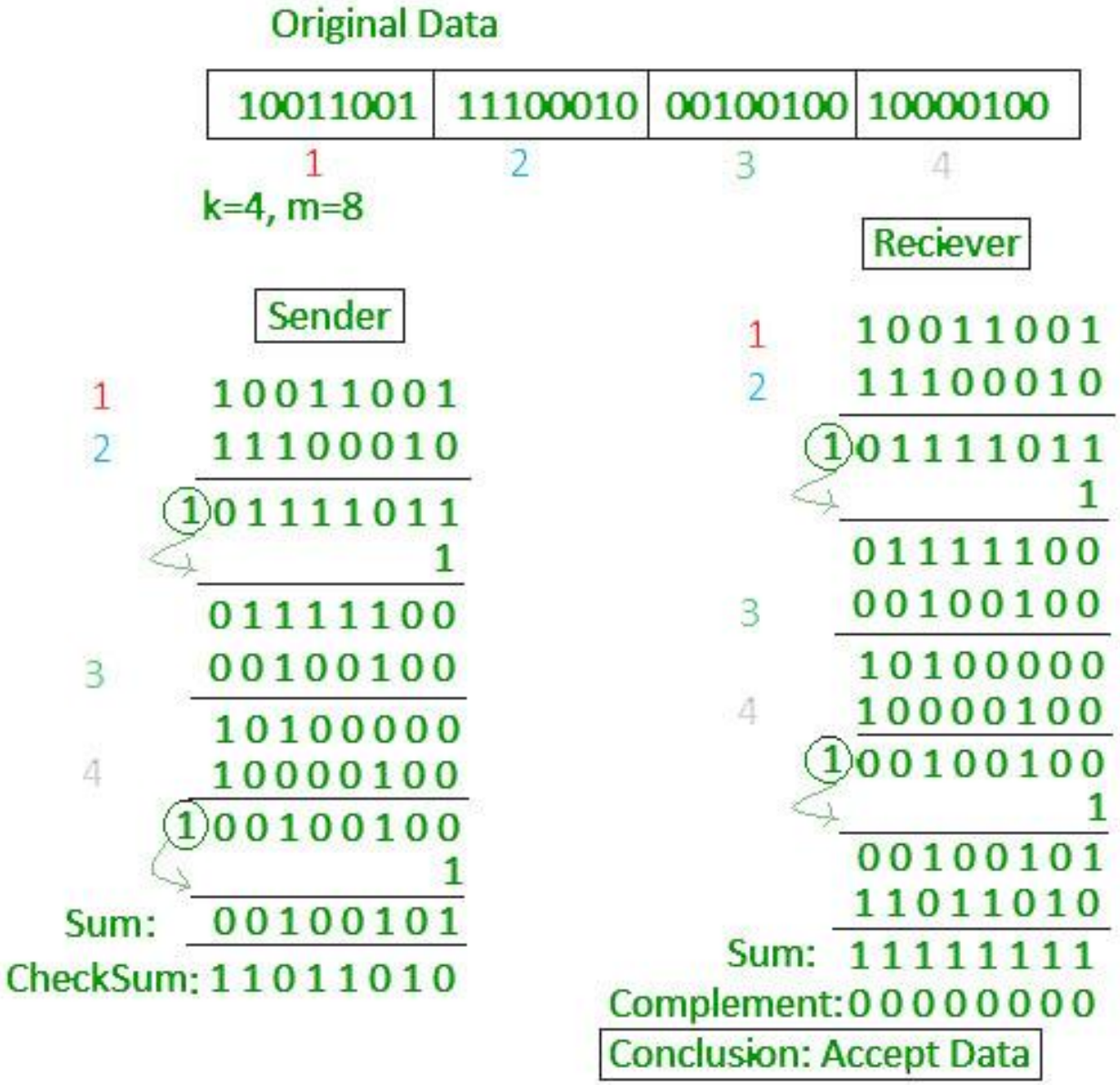
Goal: detect "errors" (e.g., flipped bits) in transmitted segment (note: used at transport layer *only*)

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

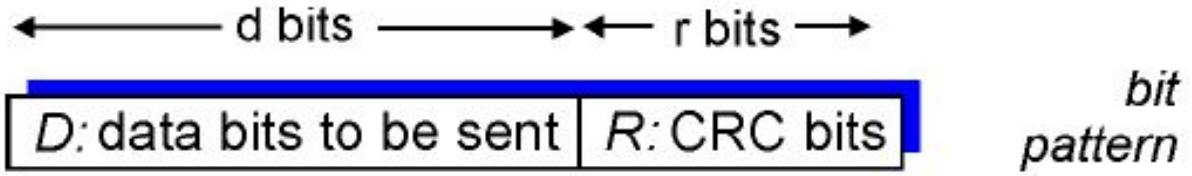
Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?* More later ....


Original Data
10011001
11100010
00100100
10000100
1
2
3
4
k=4, m=8
Sender
1 10011001
2 11100010
1 01111011
1
01111100
3 00100100
10100000
4 10000100
1 00100100
1
Sum: 00100101
CheckSum: 11011010
Reciever
1 10011001
2 11100010
1 01111011
1
01111100
3 00100100
10100000
4 10000100
1 00100100
1
00100101
11011010
Sum: 11111111
Complement: 00000000
Conclusion: Accept Data

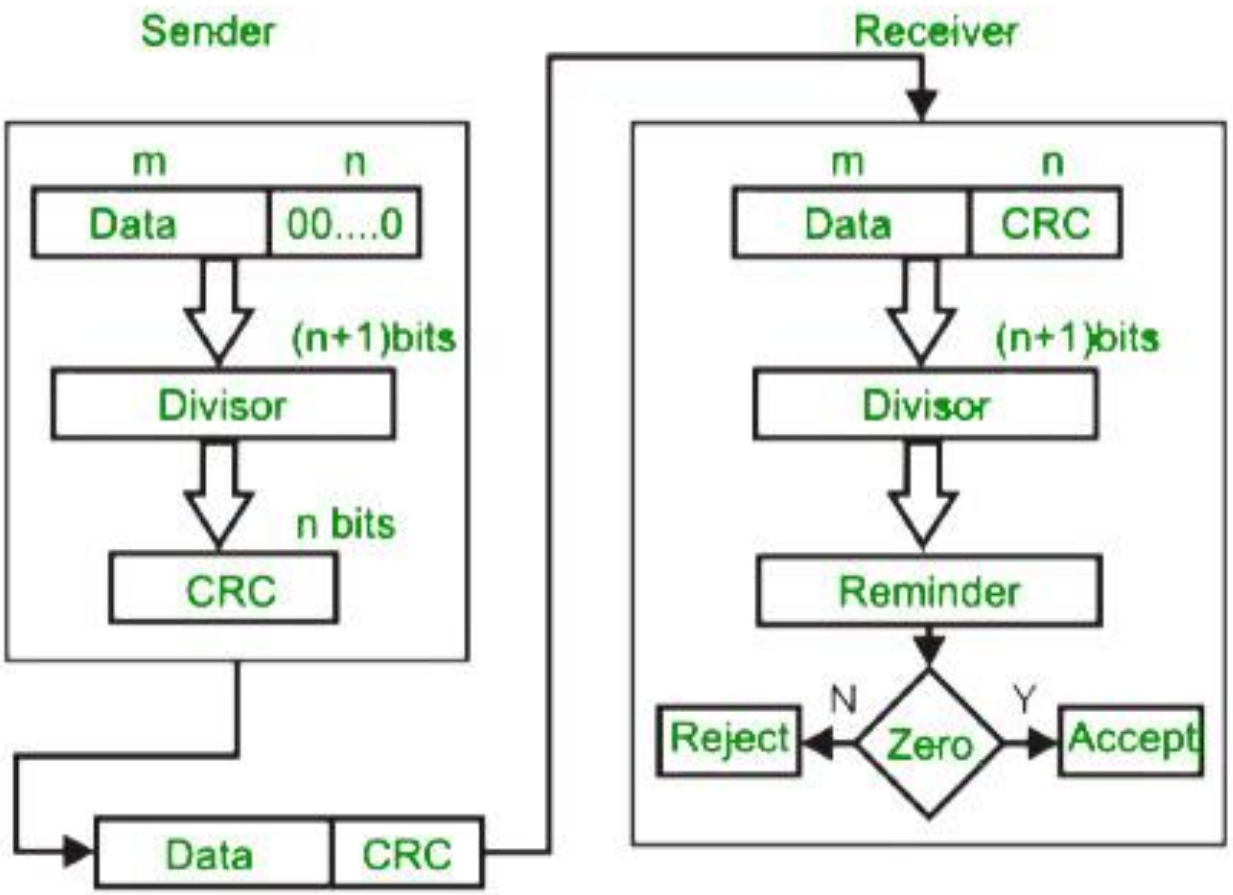
Cyclic Redundancy Check

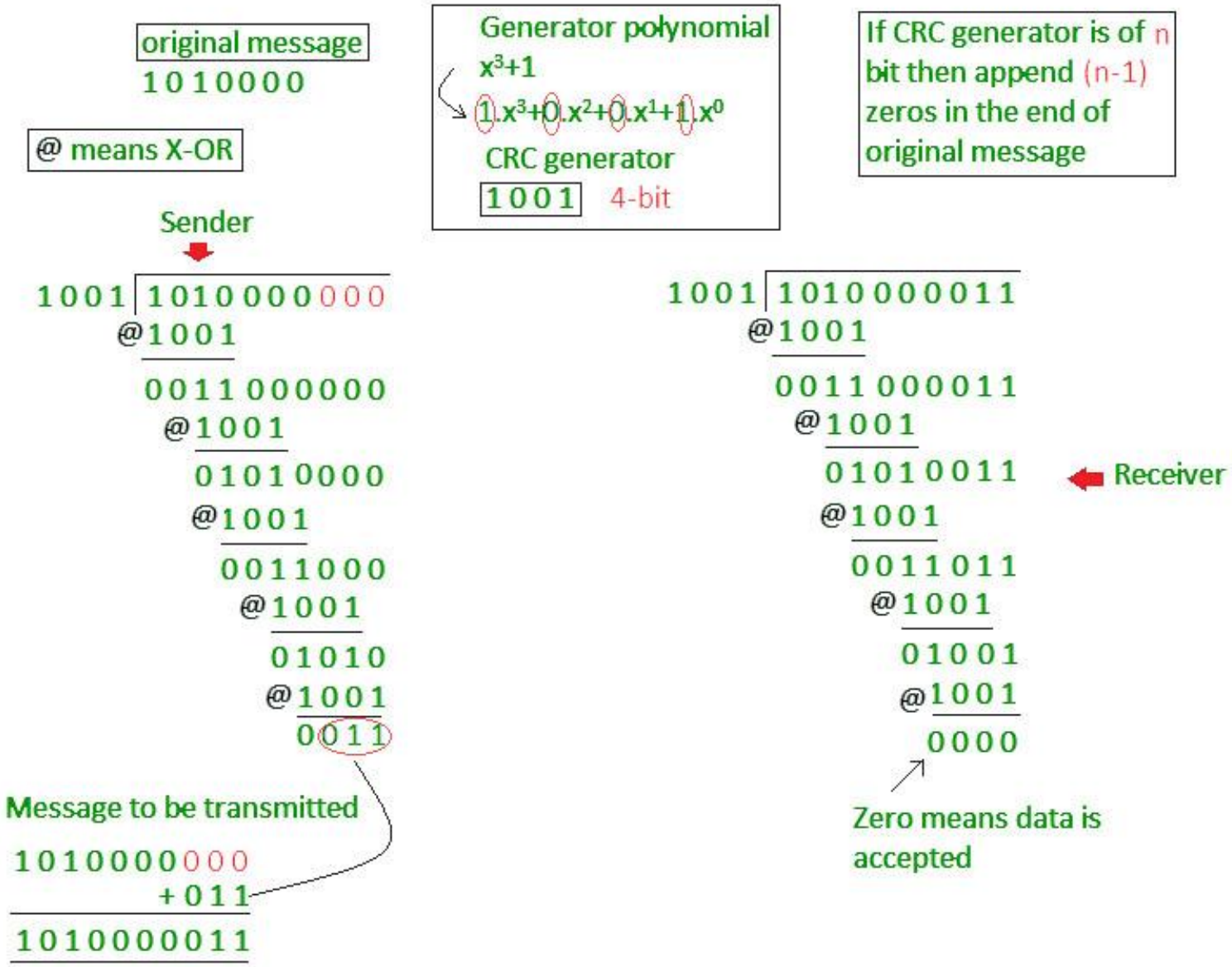
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (ATM, HDCL)



$D * 2^r$ XOR R

mathematical formula


Sender
Receiver
m
n
Data
00....0
(n+1)bits
Divisor
n bits
CRC
Data
CRC
m
n
Data
CRC
(n+1)bits
Divisor
Reminder
N
Y
Reject
Zero
Accept


original message
1 0 1 0 0 0 0
@ means X-OR
Generator polynomial
x^3+1
$1.x^3+0.x^2+0.x^1+1.x^0$
CRC generator
1 0 0 1 4-bit
If CRC generator is of n bit then append (n-1) zeros in the end of original message
Sender
1 0 0 1 | 1 0 1 0 0 0 0 0 0 0
@1 0 0 1
0 0 1 1 0 0 0 0 0 0
@1 0 0 1
0 1 0 1 0 0 0 0
@1 0 0 1
0 0 1 1 0 0 0
@1 0 0 1
0 1 0 1 0
@1 0 0 1
0 0 1 1
Message to be transmitted
1 0 1 0 0 0 0 0 0 0
+ 0 1 1
1 0 1 0 0 0 0 0 1 1
1 0 0 1 | 1 0 1 0 0 0 0 0 1 1
@1 0 0 1
0 0 1 1 0 0 0 0 1 1
@1 0 0 1
0 1 0 1 0 0 1 1
@1 0 0 1
0 0 1 1 0 1 1
@1 0 0 1
0 1 0 0 1
@1 0 0 1
0 0 0 0
Receiver
Zero means data is accepted

Polynomial Codes

- A polynomial of degree k-1 can be represented by a k-bit bit string
- At each position *i*
- Bit 0=Coefficient 0x
- Bit 1=Coefficient $x^{(i-1)}$
- $i(x) = i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \ldots + i_1x + i_0$

Example:

$i(x) = x^6 + x^4 + x^3$

1 0 1 1 0 0 0

CRC Example

Information: (1,1,0,0) ⟶ $i(x) = x^3 + x^2$

Generator polynomial: $g(x) = x^3 + x + 1$

Encoding: $i(x).x^3 = x^6 + x^5$

$$
\begin{array}{r}
x^3 + x^2 + x \\
\hline
x^3 + x + 1 \;\big)\; x^6 + x^5 \\
x^6 + \quad x^4 + x^3 \\
\hline
x^5 + x^4 + x^3 \\
x^5 + \quad x^3 + x^2 \\
\hline
x^4 + \quad x^2 \\
x^4 + \quad x^2 + x \\
\hline
x
\end{array}
$$

$$
\begin{array}{r}
1110 \\
\hline
1011 \;\big)\; 1100000 \\
1011 \\
\hline
1110 \\
1011 \\
\hline
1010 \\
1011 \\
\hline
010
\end{array}
$$

Transmitted codeword:

$b(x) = i(x).x^3 + r(x) = x^6 + x^5 + x$

⟶ $\underline{b} = (1,1,0,0,0,1,0)$

Dataword $d(x)$: 1001 $= x^3 + 1$
Appending three 0's. Multiplying by x^3.
We get $x^6 + x^3 = 1001000$
Generator $g(x) = x + 1$

$$
\begin{array}{r|l}
 & x^5 + x^4 + x^3 \\
\hline
x + 1 & \big) x^6 + x^3 \\
 & x^6 + \quad + x^5 \\
\hline
 & x^5 + x^3 \\
 & x^5 + \quad + x^4 \\
\hline
 & x^4 + x^3 \\
 & x^4 + x^3 \\
\hline
 & 0 \quad \text{Remainder}
\end{array}
$$

Codeword $c(x)$ = Dataword + Remainder
= 1001000 $= x^6 + x^3$

Lets say, **single bit error** $e(x) = x^5$ i.e. 6th LSB is inverted: 0100000. It can be seen that $e(x)$ is not divisible by $g(x)$ i.e. the error is caught.

$$
\begin{array}{r|l}
 & x^4 + x^3 + x^2 + x^1 + 1 \\
\hline
x + 1 & \big) x^5 \\
 & x^5 + x^4 \\
\hline
 & x^4 \\
 & x^4 + x^3 \\
\hline
 & x^3 \\
 & x^3 + x^2 \\
\hline
 & x^2 \\
 & x^2 + x^1 \\
\hline
 & x \\
 & x + 1 \\
\hline
 & 0
\end{array}
$$

Remainder not zero, thus error occurred and is caught.

It can also be seen that distorted codeword $c(x) + e(x) = x^6 + x^5 + x^3$ is also not divisible by $g(x)$ i.e. the error is caught

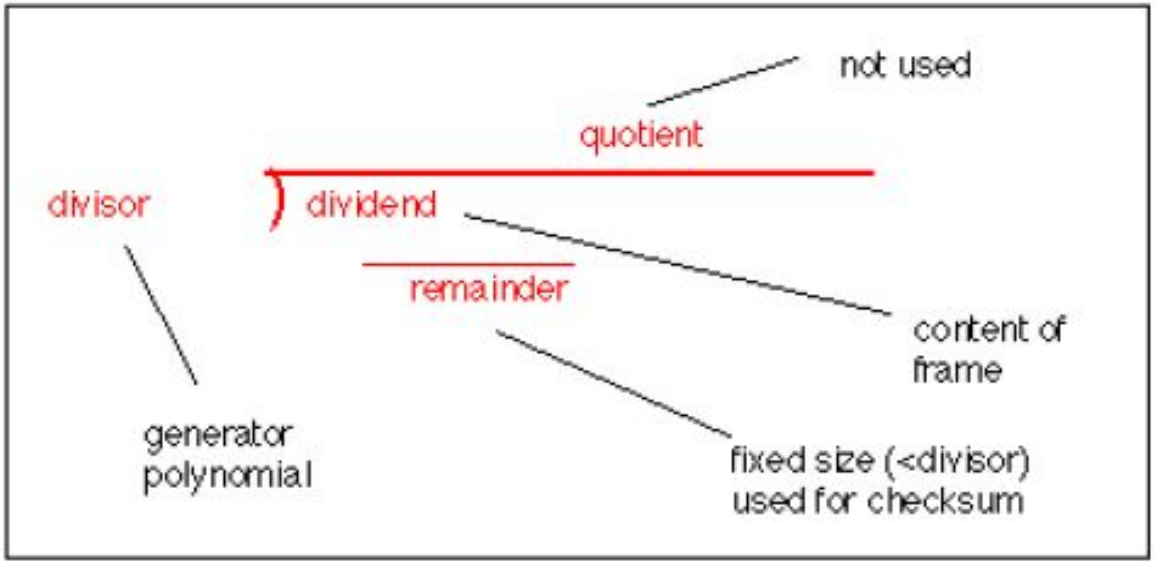
$$
\begin{array}{r|l}
 & x^5 + x^2 + x^1 + 1 \\
\hline
x + 1 & \big) x^6 + x^5 + x^3 \\
 & x^6 + x^5 \\
\hline
 & x^3 \\
 & x^3 + x^2 \\
\hline
 & x^2 \\
 & x^2 + x^1 \\
\hline
 & x \\
 & x + 1 \\
\hline
 & 0
\end{array}
$$

Remainder not zero, thus error occurred and is caught.

• There are several different standard polynomials used by popular protocols for CRC generation. These are:

Name	Polynomial	Application
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

Cyclic Redundancy Check (CRC)



CRC Example

Want:

$D \cdot 2^r$ XOR R = nG

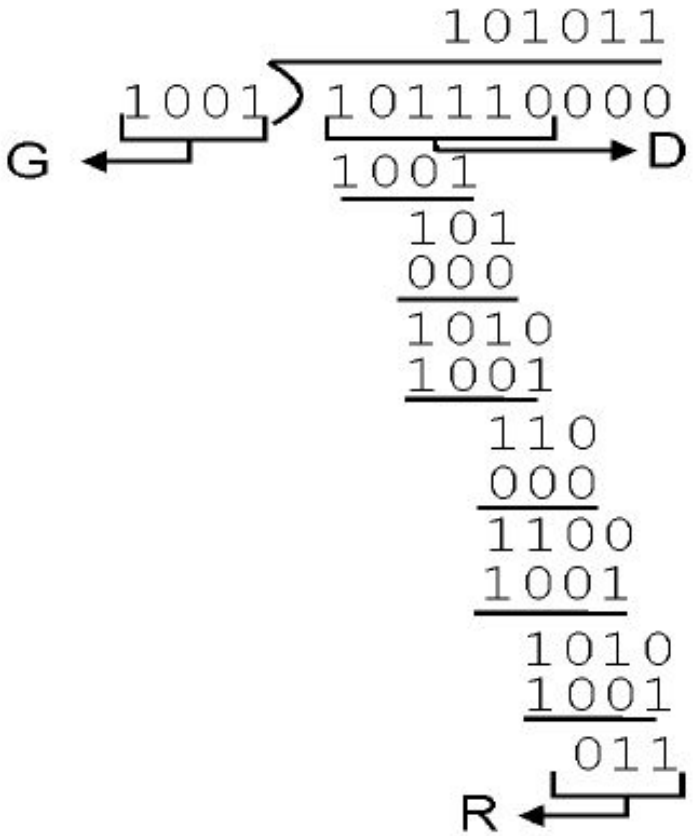
equivalently:

$D \cdot 2^r$ = nG XOR R

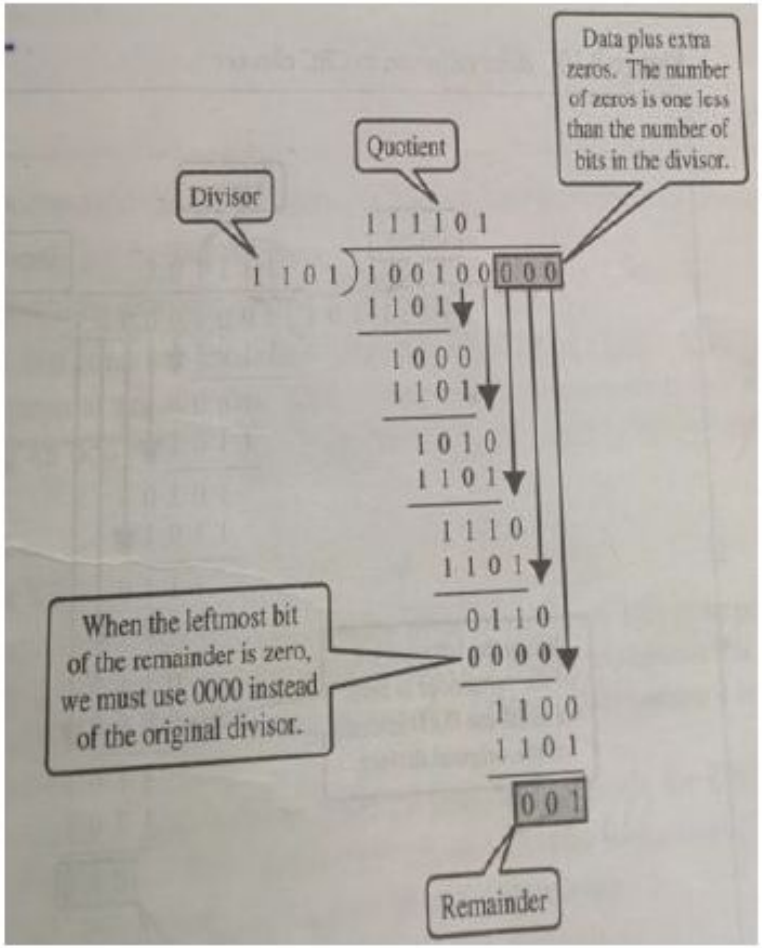
equivalently:

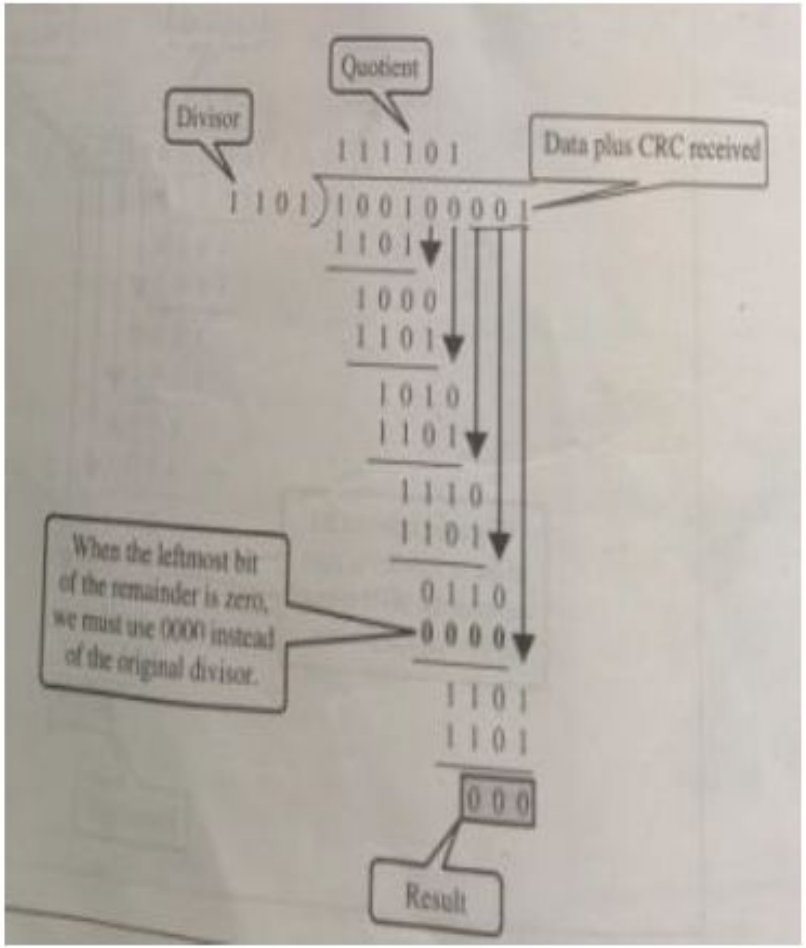
if we divide $D \cdot 2^r$ by G, want remainder R

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



Example: CRC Generator and Checker



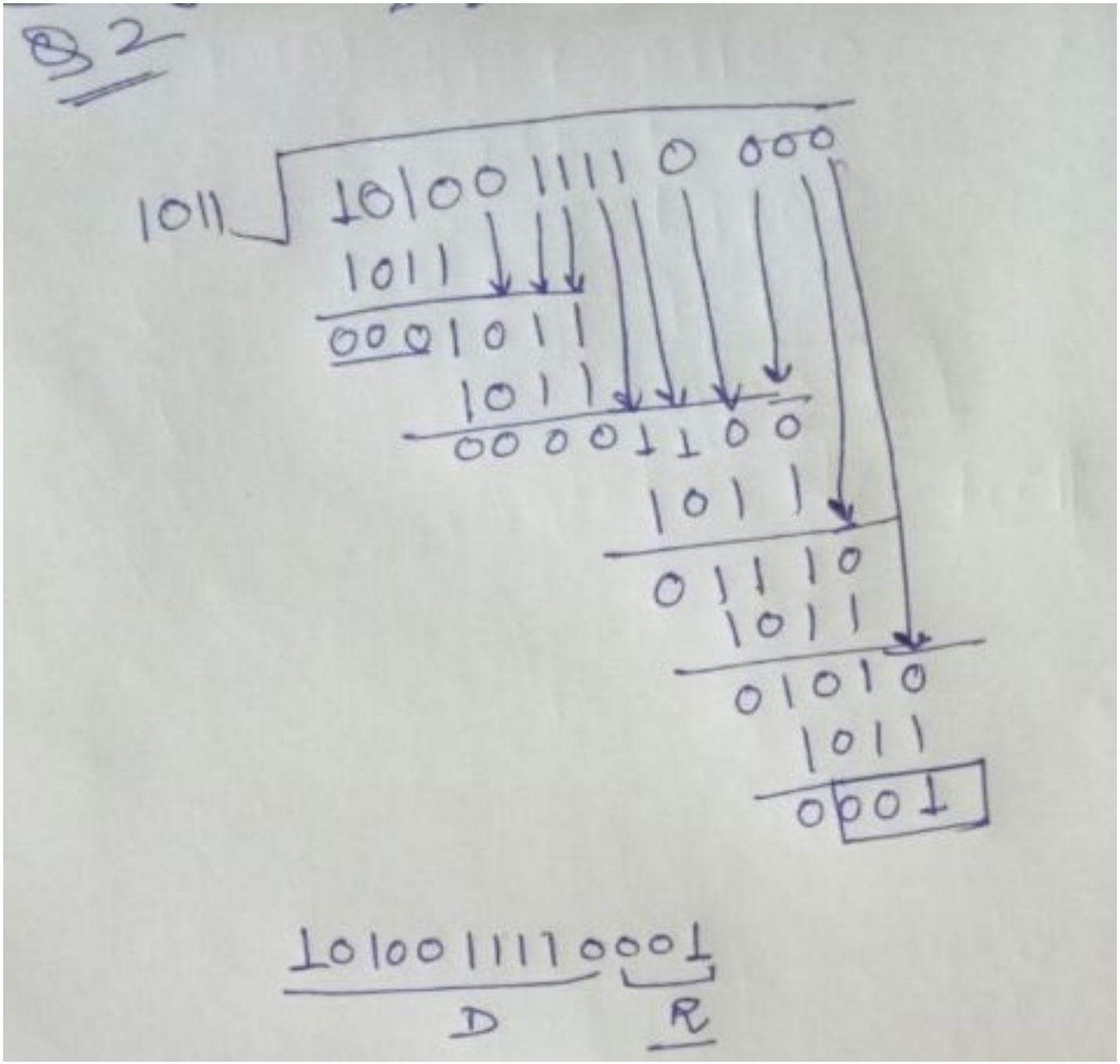


CRC polynomials

- CRC-16 = $x^{16} + x^{15} + x^2 + 1$ (used in HDLC)
- CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$
- CRC-32 = $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (used in Ethernet)

CRC Problems

1. If a divisor is 101101. how many bits long is the CRC?
2. Given 10 bit sequence 1010011110, and a divisor of 1011, find the CRC.
3. Given a remainder of 111, a data unit of 10110011, and a divisor of 1001, is there an error in the data unit?
4. Let G(x) be the generator polynomial used for CRC checking. What is the condition that should be satisfied by G(x) to detect odd number of bits in error?
5. If the CRC Code is 10100010111100 and the generator polynomial is , check if there is any error in the code word.
6. If $G(x) = x^{20} + x^{7} + 1$,how many check bits in the code word


Q2
1011
10100 1111 0 000
1011
00010 1 1
1011
00001100
1011
01110
1011
01010
1011
0001
10100 1111 0001
D
R

Hamming Code

- Consider only a single-bit error in k bits of data
  - k possibilities for an error
  - One possibility for no error
  - #possibilities = $k + 1$
- Add r redundant bits to distinguish these possibilities; we need

$$2^r \geq k+1$$

- But the r bits are also transmitted along with data; hence

$$2^r \geq k+r+1$$

Number of Redundant Bits

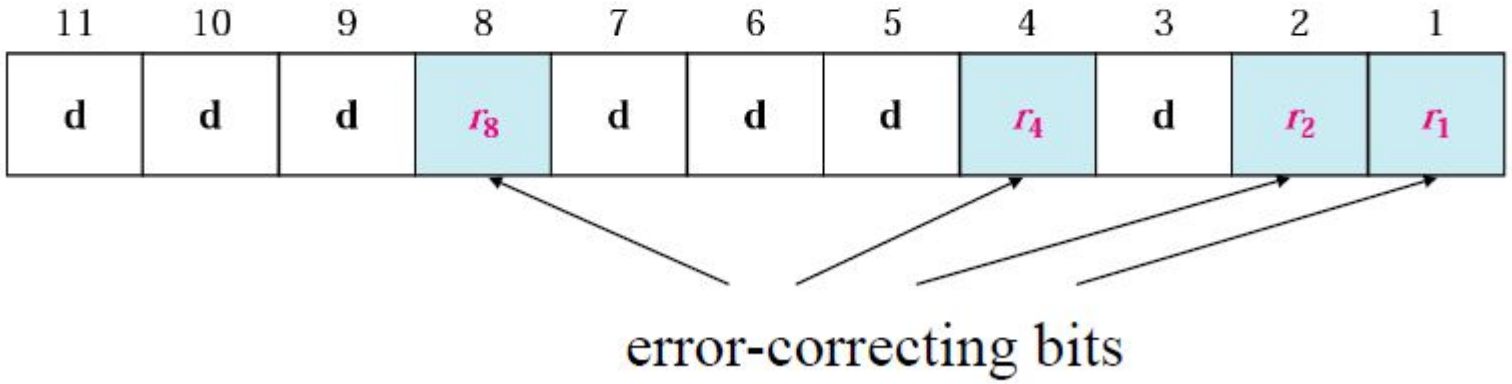
Number of data bits k	Number of redundancy bits r	Total bits $k + r$
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

General Algorithm of Hamming code

The Hamming Code is simply the use of extra parity bits to allow the identification of an error.

1. Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc).
2. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.
   a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant position (1, 3, 5, 7, 9, 11, etc).
   b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from the least significant bit (2, 3, 6, 7, 10, 11, etc).
   c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from the least significant bit (4–7, 12–15, 20–23, etc).
   d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit bits (8–15, 24–31, 40–47, etc).
   e. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.
5. Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is odd.
6. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Hamming Code



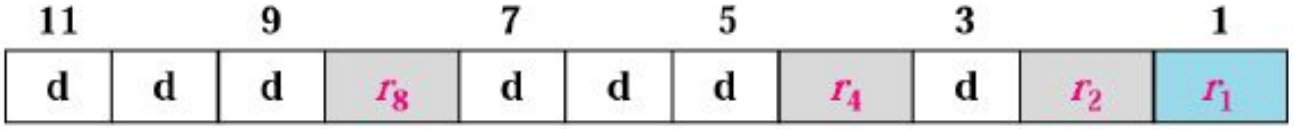
Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Position	R8	R4	R2	R1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1

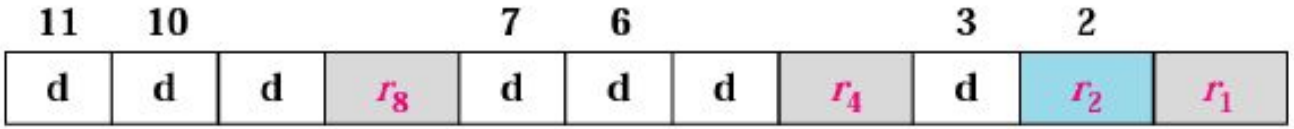
R1 -> 1,3,5,7,9,11
R2 -> 2,3,6,7,10,11
R3 -> 4,5,6,7
R4 -> 8,9,10,11

Redundant Bit Calculation

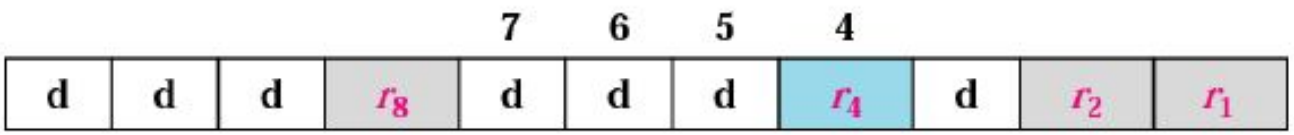
r_1 will take care of these bits.



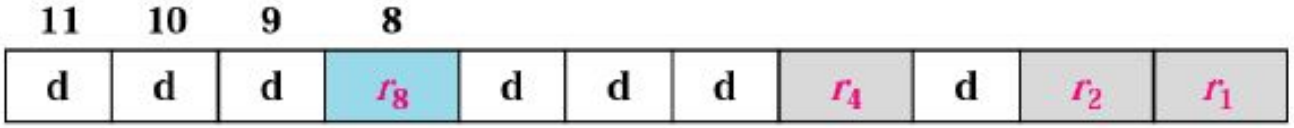
r_2 will take care of these bits.



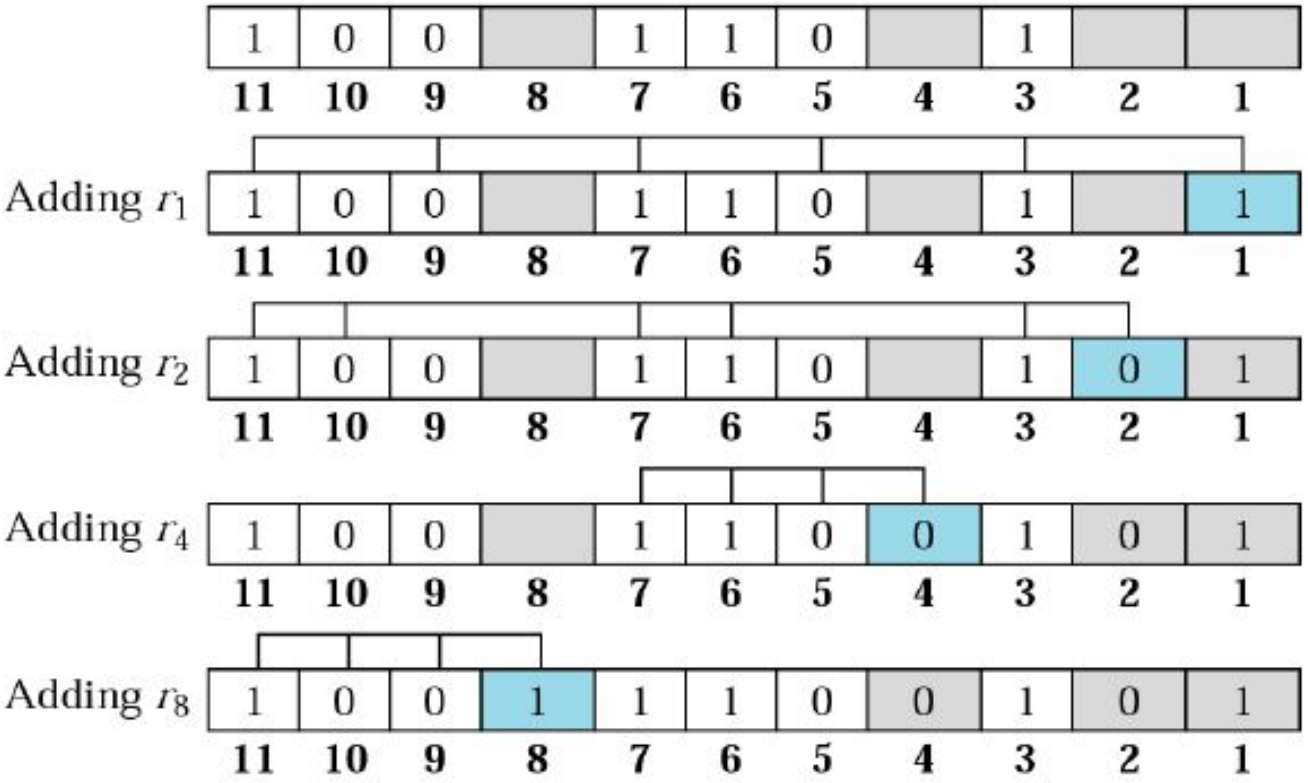
r_4 will take care of these bits.



r_8 will take care of these bits.



Example: *Hamming Code*

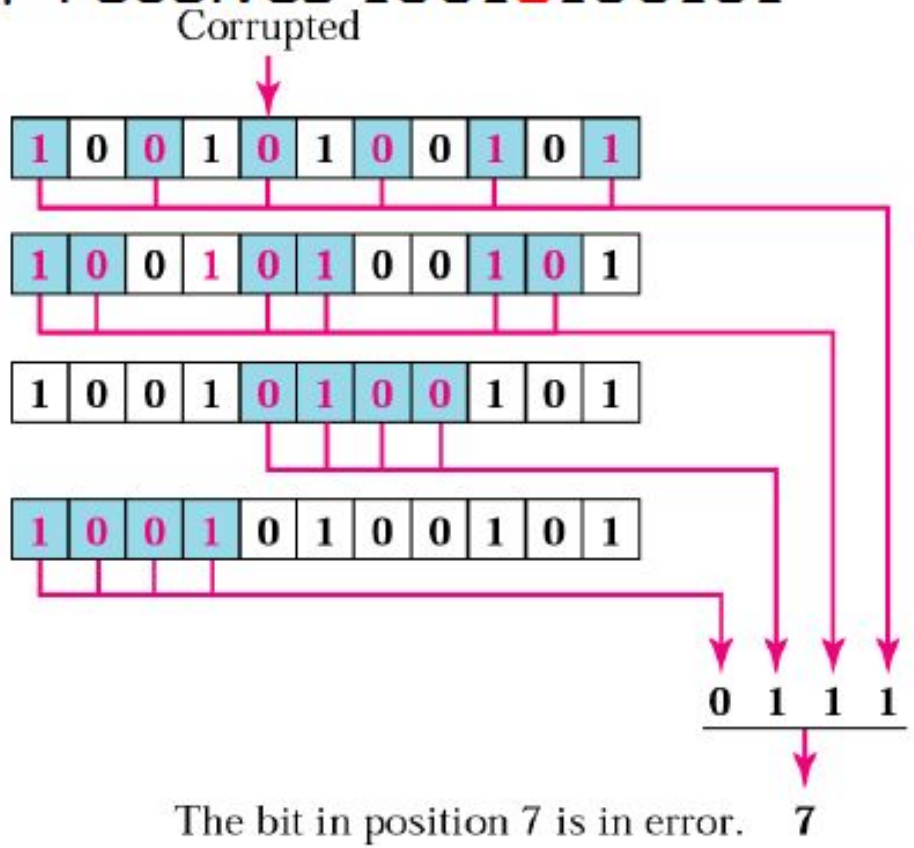


Data:
1 0 0 1 1 0 1

Code:
1 0 0 1 1 1 0 0 1 0 1

Example: *Correcting Error*

- Receiver receives 10010100101

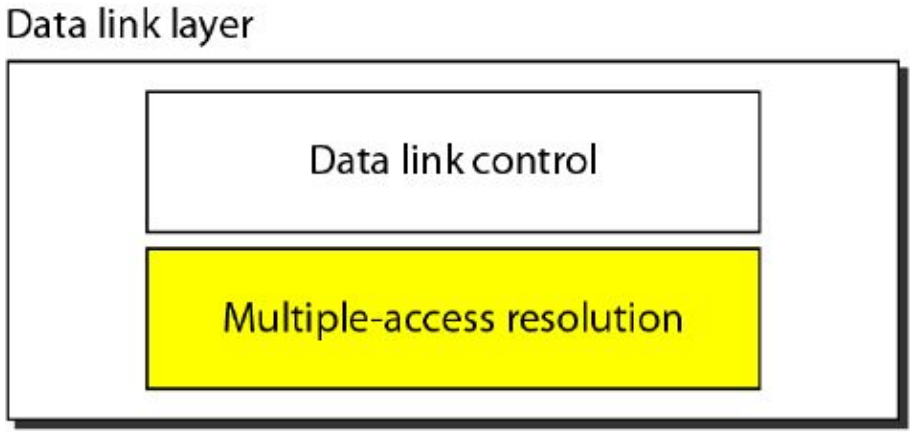


Strength of Hamming Code

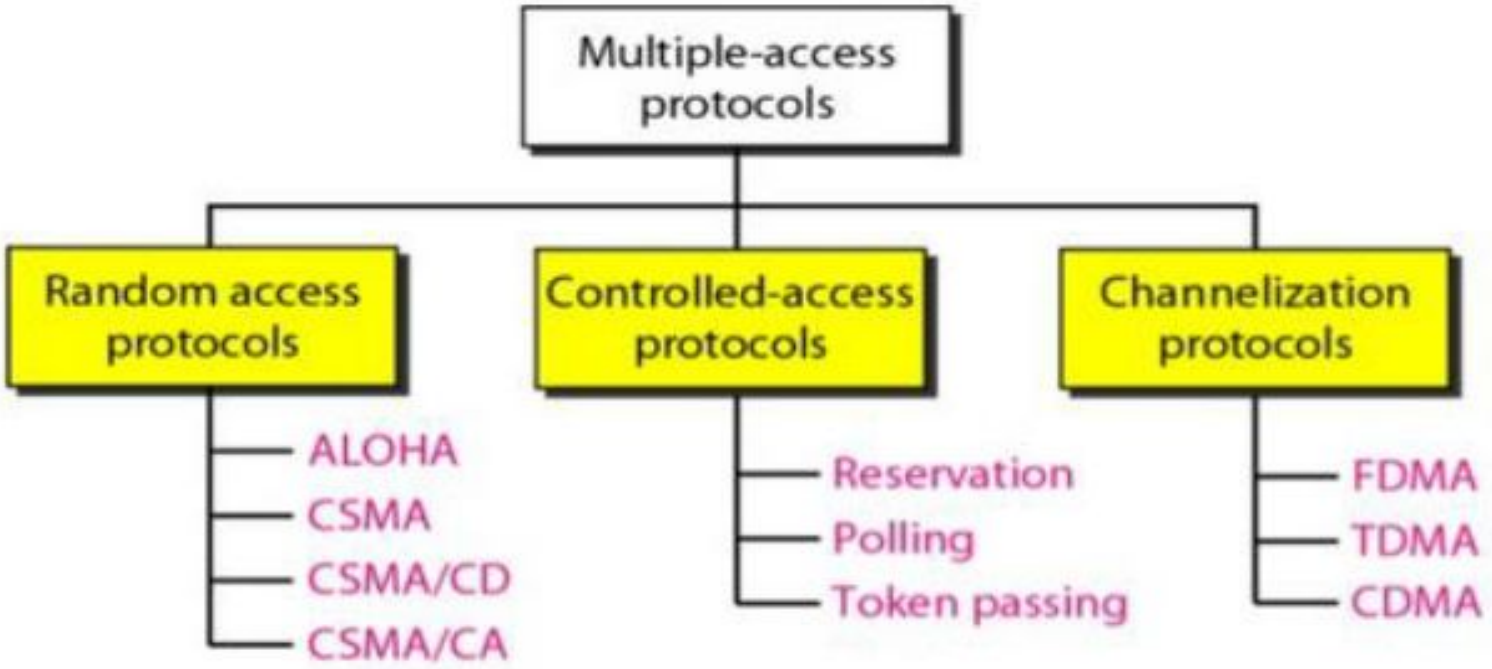
- Minimum Hamming Distance is 3
  - It can correct at most 1 bit error
  - It can detect at most 2 bit error
  - But… not both!!! (Why?)
- SECDED – Extended Hamming code with one extra parity bit
  - Achieves minimum Hamming distance of 4
  - Can distinguish between one bit and two bit errors

Multiple Access Protocols

Data link layer divided into two functionality-oriented sublayers



- The data link control is responsible for reliable transmission of message over transmission channel **by using techniques like framing, error control and flow control**.
- If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is **no dedicated link present then multiple stations** can access the channel simultaneously. Hence **multiple access protocols** are required to **decrease collision and avoid crosstalk.**


Multiple-access protocols
Random access protocols
Controlled-access protocols
Channelization protocols
ALOHA
CSMA
CSMA/CD
CSMA/CA
Reservation
Polling
Token passing
FDMA
TDMA
CDMA

Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- two or more transmitting nodes -> "collision",
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

- Two features give this method its name.
- First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access.*
- Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.

ALOHA

There are two versions of Aloha system which differ with respect to whether or not time is divided up into discrete slots into which all frames must fit. :

- *PURE ALOHA*
- *SLOTTED ALOHA*

Pure ALOHA

- The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol.
- **The idea is that each station sends a frame whenever it has a frame to send.**
- However, since there is only one channel to share, there is the possibility of collision between frames from different stations.
- The pure **ALOHA protocol relies on acknowledgments** from the receiver.
- Pure ALOHA dictates that when **the time-out period passes**, each station **waits a random amount of time before resending its frame**. The randomness will help **avoid more collisions**. We call this time the **back-off time *TB***

Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives

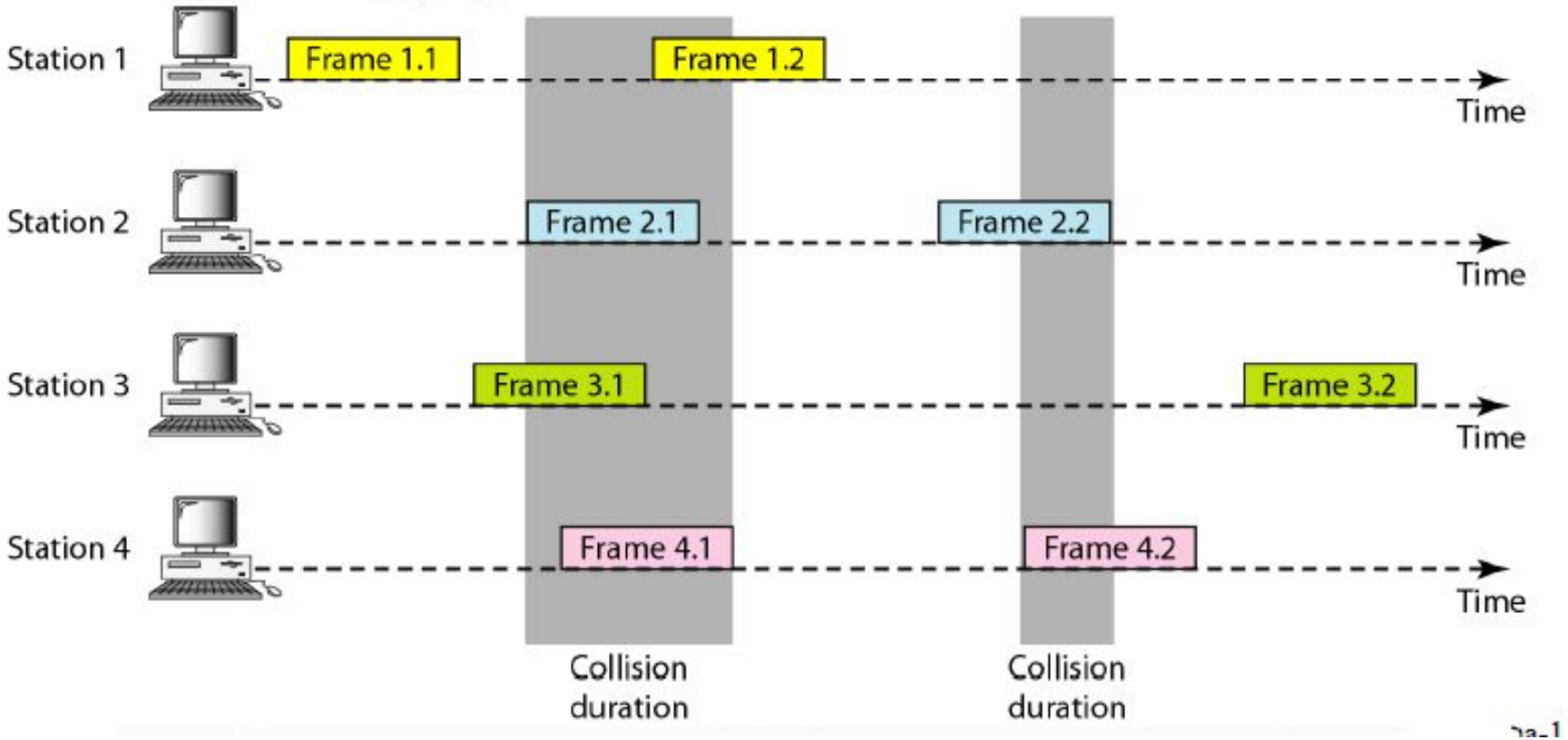
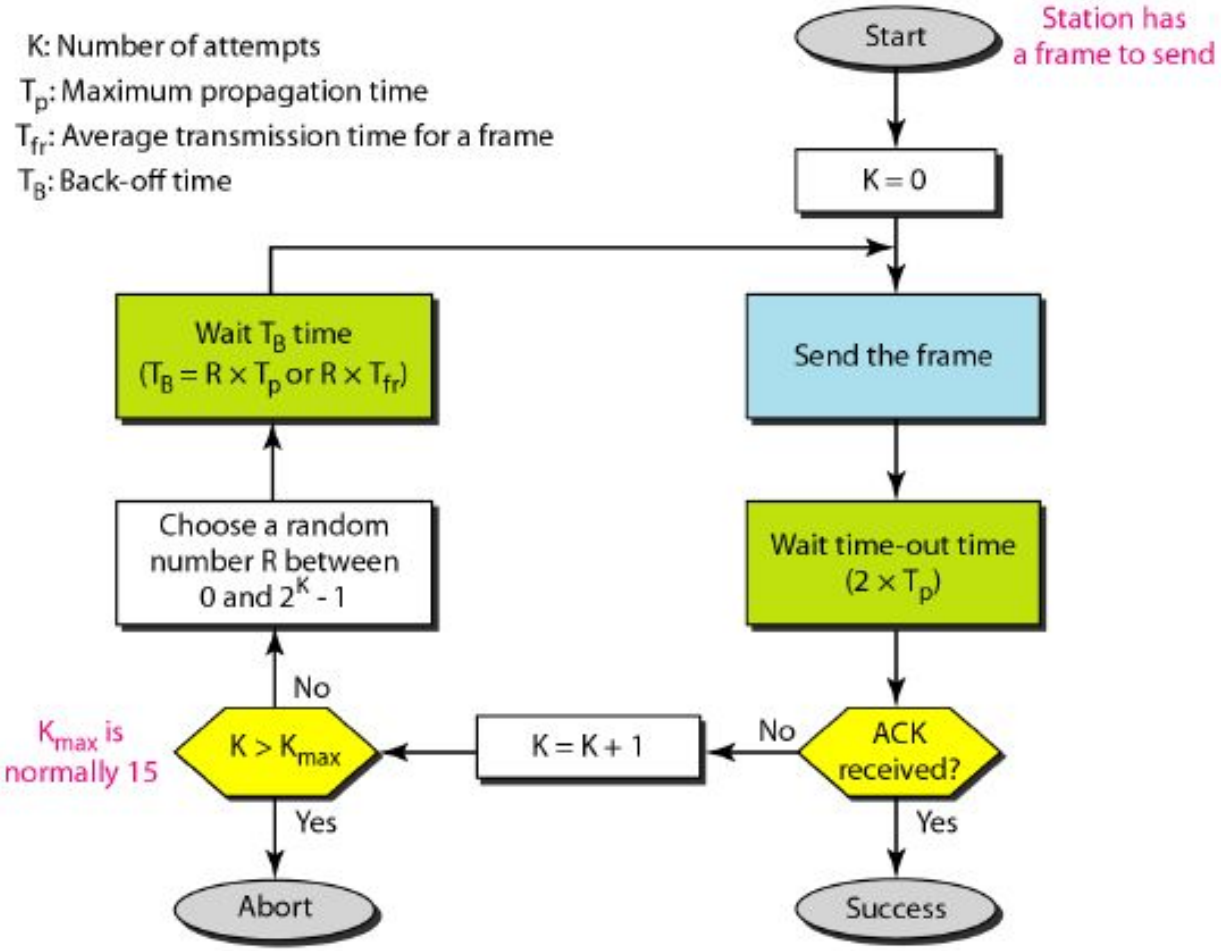


Figure Procedure for pure ALOHA protocol



Example

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, we find

$$T_p = (600 \times 10^5) / (3 \times 10^8) = 2 \text{ ms}.$$

Now we can find the value of T_B for different values of K.

a. For K = 1, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0 × 2) or 2 ms (1 × 2), based on the outcome of the random variable.

Example (continued)

b. For K = 2, the range is {0, 1, 2, 3}. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

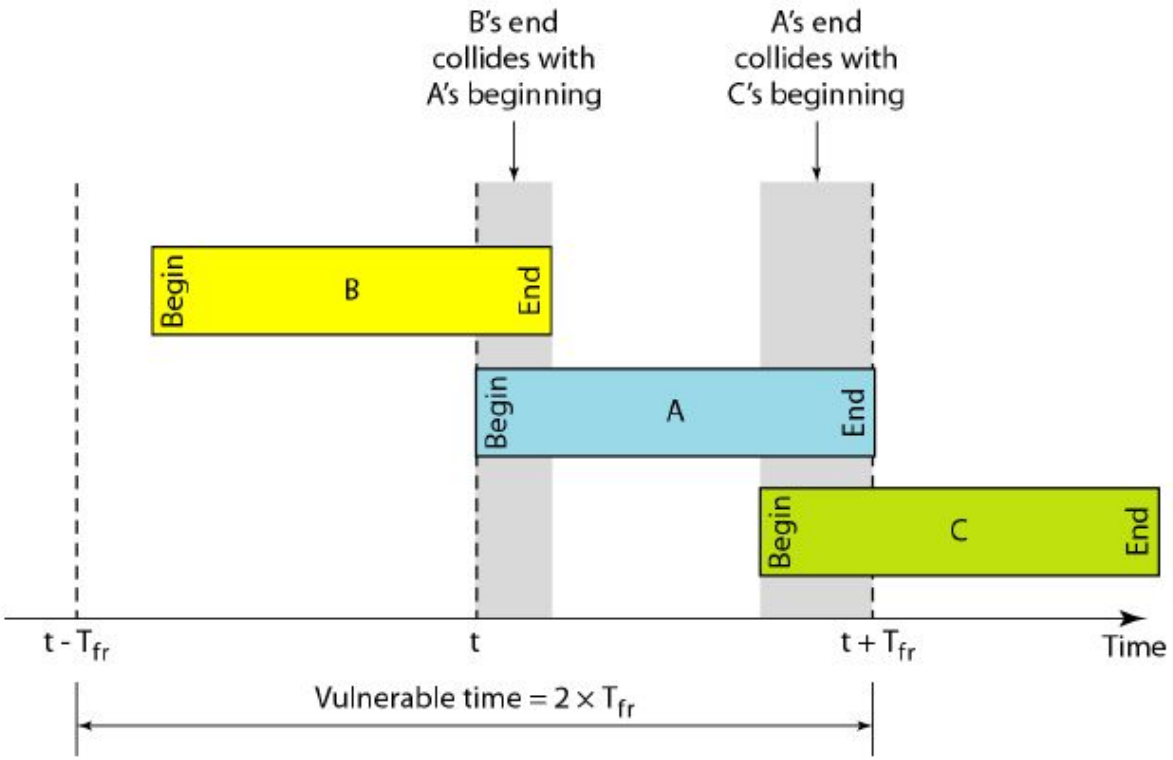
c. For K = 3, the range is {0, 1, 2, 3, 4, 5, 6, 7}. This means that T_B can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.

d. We need to mention that if K > 10, it is normally set to 10.

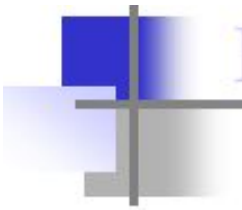
Vulnerable time

- Let us find the length of time, the **vulnerable time,** in which there is a possibility of collision.
- We assume that the stations send fixed-length frames with each frame taking T_{fr} *sec* to send.

Vulnerable time for pure ALOHA protocol



Pure ALOHA vulnerable time = 2 x T_{fr}



Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is 2×1 ms = 2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

Throughput

- Let G be the average number of frames generated by the system during one frame transmission time. Then average number of successful transmissions for it:

The throughput for pure ALOHA is
$S = G \times e^{-2G}$.
The maximum throughput
$S_{max} = 0.184$ when G= (1/2).

In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully

Throughput

- Efficiency n=G*e^-2G
- For finding maximum efficiency

Differentiate n wrt G

dn/dG = (-2G)G *e^-2G+ e^-2G

e^-2G(-2G+1)=0

G=1/2

Max efficiency=1/2 * e^-1

=0.184 i.e. 18.4%

One half of the frame is generated over 1 frame transmission time and 18.4% of frames reach destination successfully.

Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second b. 500 frames per second

c. 250 frames per second.

Solution

The frame transmission time is 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-2G}$ or $S = 0.135$ (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

Example (continued)

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentagewise.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-2G}$ or $S = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

Pure ALOHA efficiency

P(success by given node) = P(node transmits) ·

P(no other node transmits in $[t_0-1,t_0]$ ·

P(no other node transmits in $[t_0-1,t_0]$

$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$

$= p \cdot (1-p)^{2(N-1)}$ $\xrightarrow[\infty]{}$

... choosing optimum p and then letting n

$= 1/(2e) = .18$

even worse than slotted Aloha!

Q:

A group of N stations share a 56kbps pure ALOHA channel. Each station outputs a 1000bit frame on an average of once every 100 sec, even if the previous one has not yet been send (e.g. the stations are buffered). What is the maximum value of N ?

There are N Stations Sharing 56kbps Pure ALOHA Channel

so with pure ALOHA Usable Bandwidth = 0.184* 56kbps=10.3kbps

1 Station Outputs 1000 bits in every 100sec

so in 1sec One station will ouput at rate 1000/100=10bits/sec

so For N stations in 1 sec Total Output Data is 10*N bits this should be equal to the Channel Capacity in pure ALOHA

N*10=10300

N=1030 it is the maximum value of Number of Station Possible.

Soln.

Recall that ALOHA achieves an average throughput of appr 18%, when operating at reasonable load. In an ALOHA network with channel capacity 56 kbps, only 18% of this capacity will be used to deliver meaningful data.

With pure ALOHA the usable bandwidth is 0.184 * 56kbps = 10.3kbps.

This 10 kbps must be divided among N hosts, each of which is transmitting

an average of 1000 bits every 100 seconds. This corresponds to a transmission rate of 10 bits per second per host. If the channel can support

10 kbps of data, then it can support up to N users, each transmitting at 10

Bps Each station requires 10 bps (1000bit/100sec), so

N = 10300 /10 = ***1030 stations***.

Slotted ALOHA

Assumptions

- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation

- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. p until success

- **Pure ALOHA has a vulnerable time of 2 x T_{fr} .** This is so because there is **no rule** that defines when the station can send.
- A station may send soon after another station has started or soon before another station has finished.
- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA we divide the time into slots of $\mathbf{T_{fr}}$ **sec** and **force the station to send only at the beginning of the time slot**.

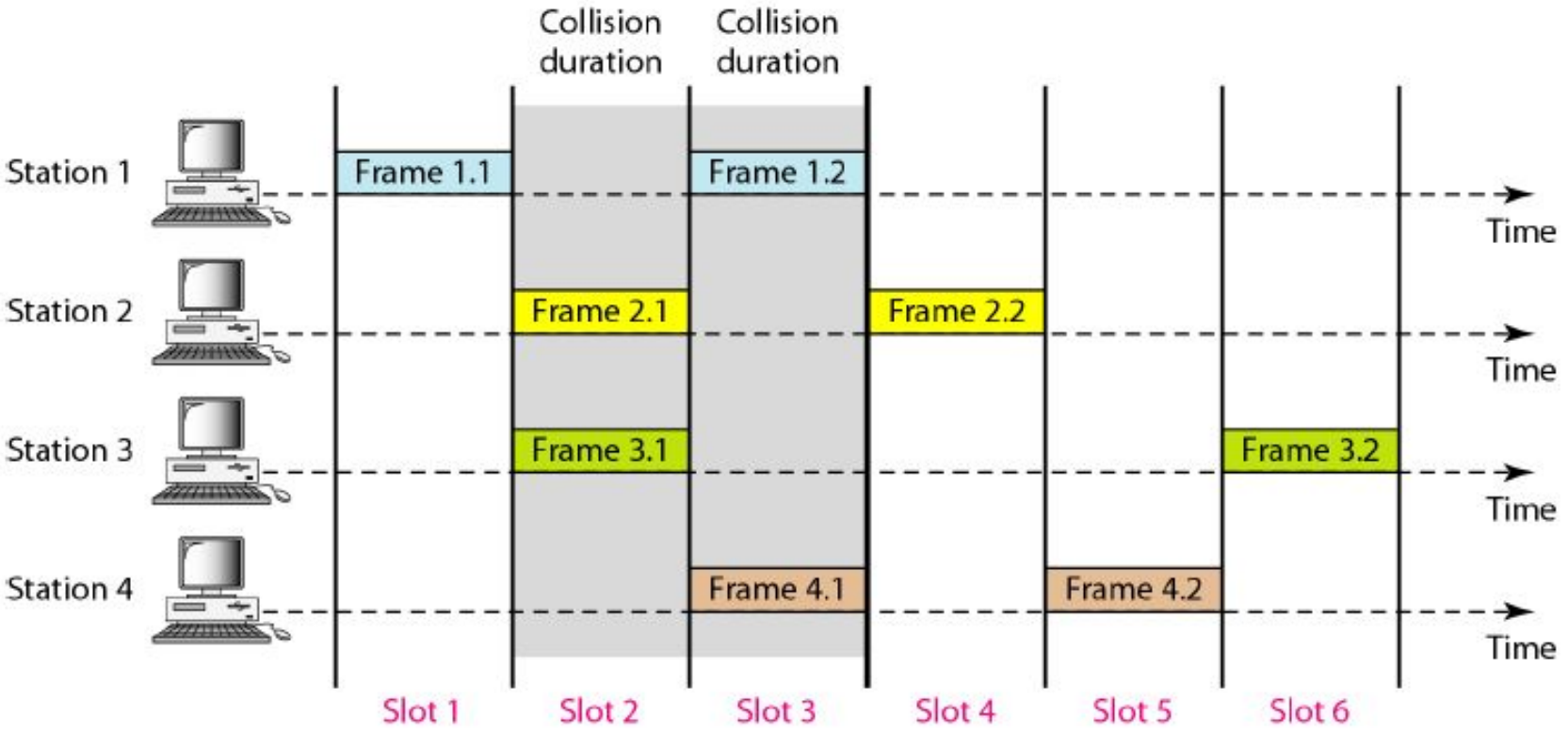
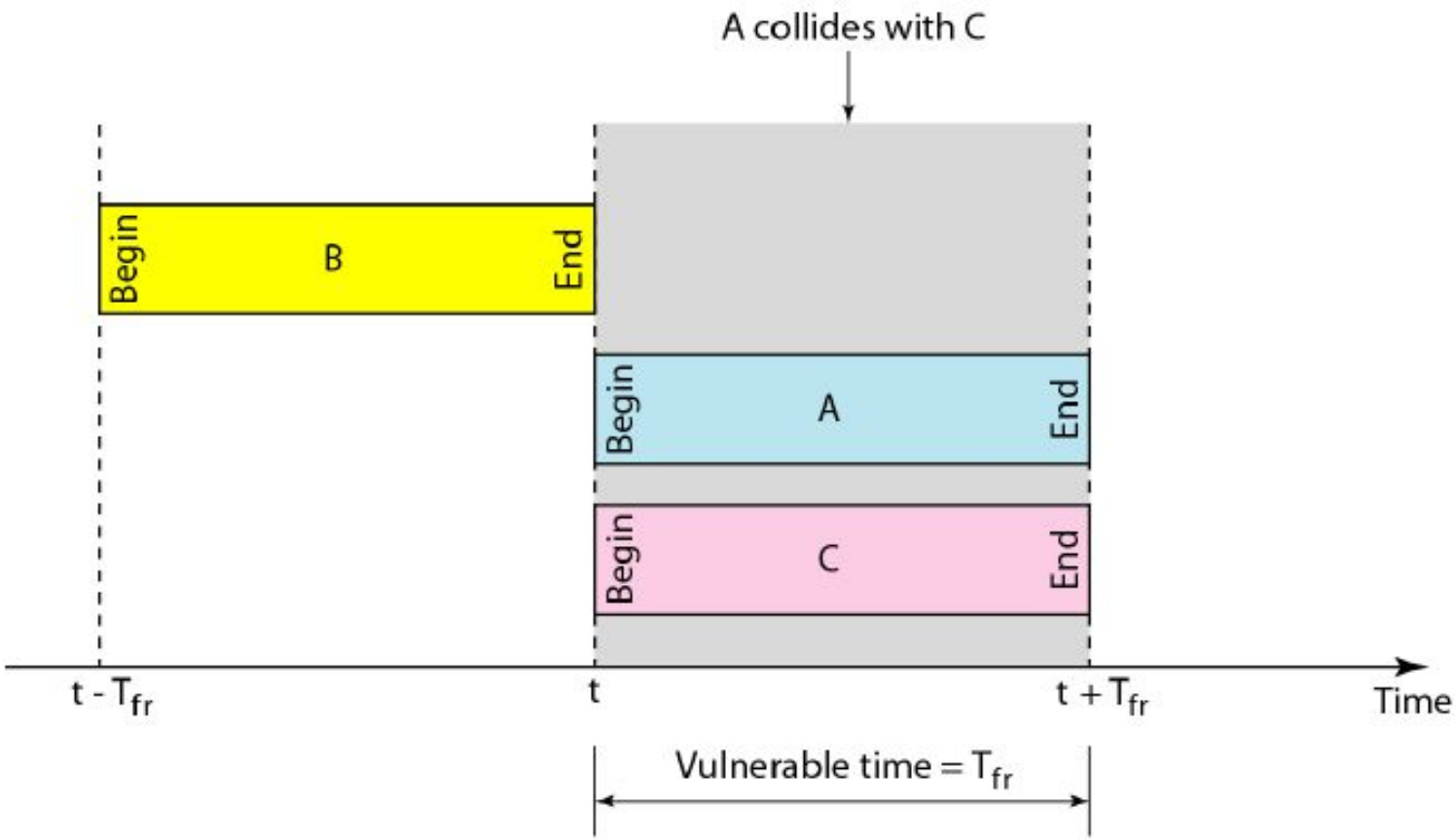

Collision duration
Collision duration
Station 1
Station 2
Station 3
Station 4
Frame 1.1
Frame 1.2
Frame 2.1
Frame 2.2
Frame 3.1
Frame 3.2
Frame 4.1
Frame 4.2
Time
Time
Time
Time
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Slot 6

Figure Vulnerable time for slotted ALOHA protocol



SlottedALOHA vulnerable time = T_{fr}

Note

The throughput for slotted ALOHA is $S = G \times e^{-G}$.

The maximum throughput $S_{max} = 0.368$ when $G = 1$.

In other words, if **a frame is generated during one frame transmission time,** then 36.8 percent of these frames reach their destination successfully.

This result can be expected because the vulnerable time is equal to the frame transmission time.

Therefore, if a station generates only one frame in this vulnerable time (and no other station generates a frame during this time), the frame will reach its destination successfully.

Example

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second b. 500 frames per second

c. 250 frames per second.

Solution

The frame transmission time is 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.0368 = 368$ frames. Only 386 frames out of 1000 will probably survive.

Example (continued)

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-G}$ or $S = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

Slotted ALOHA

Pros

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet

Slotted Aloha efficiency

Efficiency is the long-run fraction of successful slots when there's many nodes, each with many frames to send

- Suppose N nodes with many frames to send, each transmits in slot with probability *p*
- prob that 1st node has success in a slot = $p(1-p)^{N-1}$
- prob that any node has a success = $Np(1-p)^{N-1}$

- For max efficiency with N nodes, find p* that maximizes $Np(1-p)^{N-1}$
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives 1/e = .37

At best: channel used for useful transmissions 37% of time!

Question 2: *(Slotted Aloha) Suppose four active nodes—nodes A, B, C and D—are competing for access to a channel using slotted ALOHA. Assume each node has an infinite number of packets to send. Each node attempts to transmit in each slot with probability p. The first slot is numbered slot 1, the second slot is numbered slot 2, and so on.*

a. What is the probability that node A succeeds for the first time in slot 5?

b. What is the probability that some node (either A, B, C or D) succeeds in slot 4?

c. What is the probability that the first success occurs in slot 3?

d. What is the efficiency of this four-node system?

a) $(1 - p(A))^4 \, p(A)$

where, $p(A)$ = probability that A succeeds in a slot

$p(A) = p(A \text{ transmits and B does not and C does not and D does not})$

$= p(A \text{ transmits}) \, p(B \text{ does not transmit}) \, p(C \text{ does not transmit}) \, p(D \text{ does not transmit})$

$= p(1 - p) \, (1 - p)(1-p) = p(1 - p)^3$

Hence, $p(A \text{ succeeds for first time in slot 5}) = (1 - p(A))^4 \, p(A) = (1 - p(1 - p)^3)^4 \, p(1 - p)^3$

b) $p(A \text{ succeeds in slot 4}) = p(1-p)^3$

$p(B \text{ succeeds in slot 4}) = p(1-p)^3$

$p(C \text{ succeeds in slot 4}) = p(1-p)^3$

$p(D \text{ succeeds in slot 4}) = p(1-p)^3$

$p(\text{either A or B or C or D succeeds in slot 4}) = 4 \, p(1-p)^3$

(because these events are mutually exclusive)

c) $p(\text{some node succeeds in a slot}) = 4p(1-p)^3$
$p(\text{no node succeeds in a slot}) = 1 - 4p(1-p)^3$

Hence, $p(\text{first success occurs in slot 3}) = p(\text{no node succeeds in first 2 slots})\, p(\text{some node succeeds in } 3^{rd} \text{ slot}) = (1 - 4p(1-p)^3)^2\, 4p(1-p)^3$

d) efficiency = $p(\text{success in a slot}) = 4p(1-p)^3$

Question 1:

Ten thousand airline reservation stations are coming for the use of a single slotted ALOHA channel. The average station makes 18 request/hour. A slot is 125 micro sec. What is the approximate total channel load.

Solution:

Average requests for 10000 stations

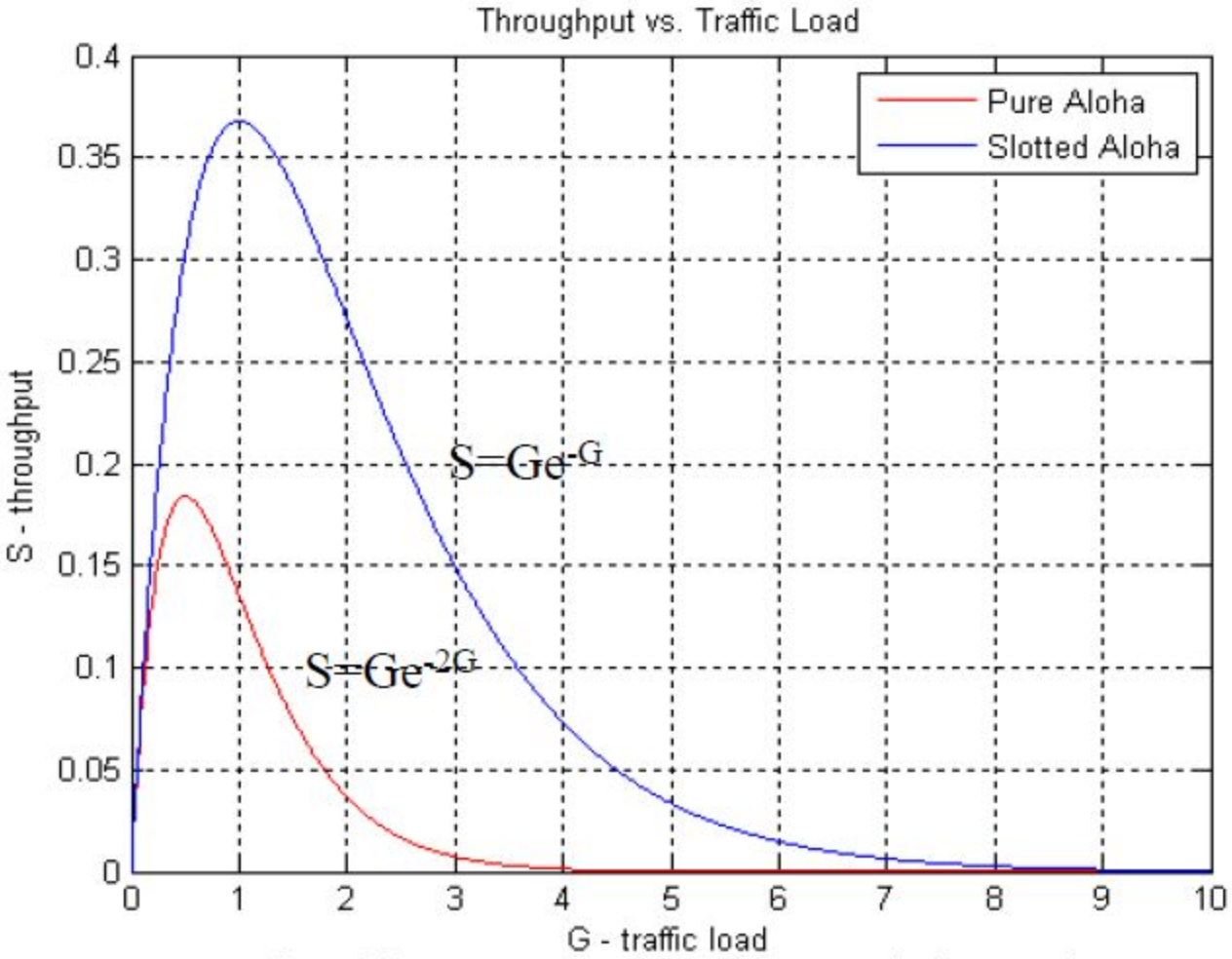
= 10^4 x 18 / (60 x 60) = 50 requests/sec

Average slots number = 1 / (125 x 10^-6) = 8000 slots/sec.

Total channel load = average requests / average slots number

= 50 / 8000 = 0.0625

Hence, the total channel load is 0.0625 request/slot.



G : average number of frames generated by the system during one frame transmission time.

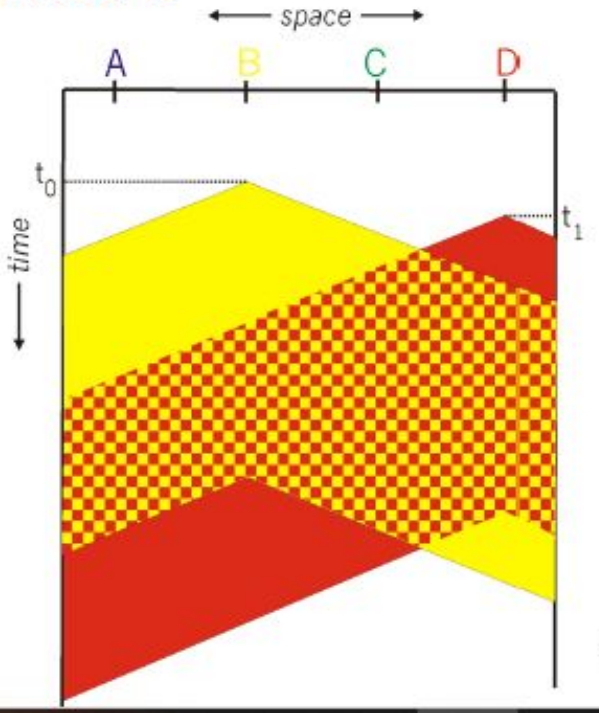
CSMA and CSMA/CD

- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, **CSMA is based on the principle "sense before transmit" or "listen before talk.“**
- **CSMA can reduce the possibility of collision,** but it **cannot eliminate it.**
- **The possibility of collision still exists because of propagation delay;** when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it. In other words, **a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.**

CSMA (Carrier Sense Multiple Access)

CSMA: listen before transmit:

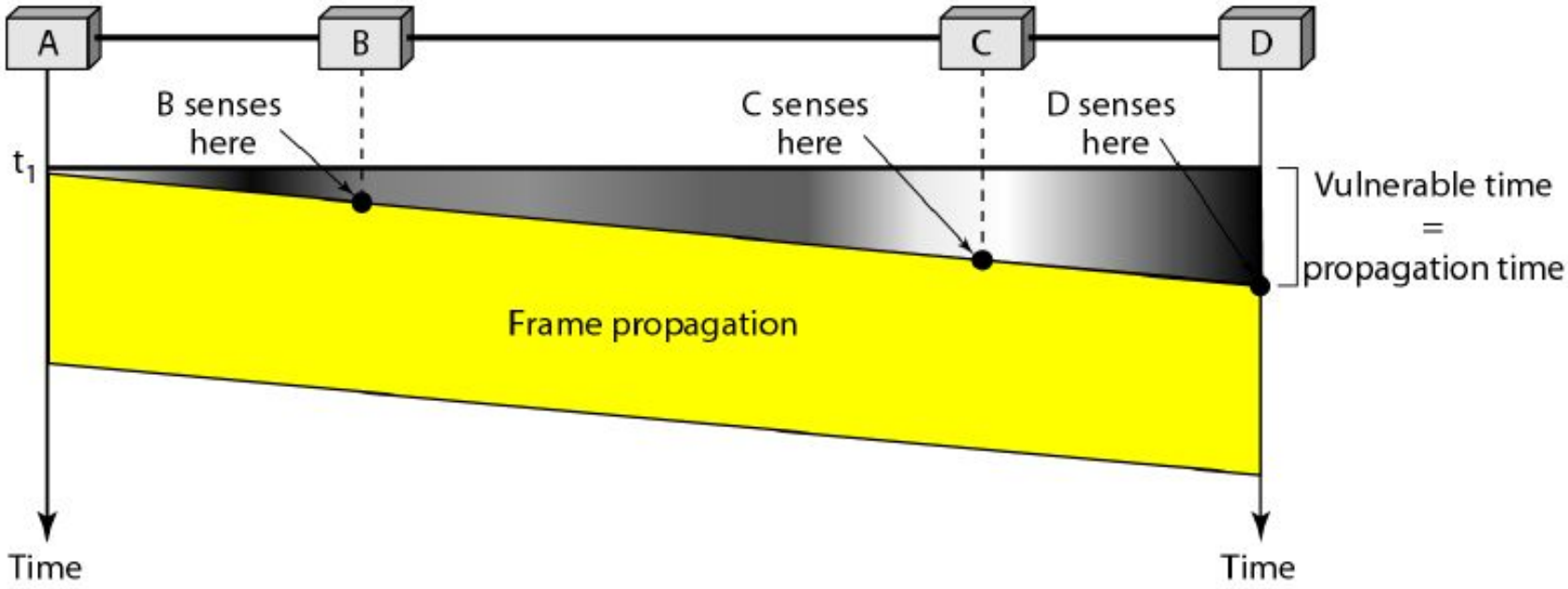
- If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission
- Human analogy: don't interrupt others!



vulnerable time

- **The vulnerable time for CSMA is the propagation time *Tp* .**
- This is the time needed for a **signal to propagate from one end of the medium to the other.**
- When a station sends a frame, and any other station tries to send a frame during this time, a collision will result.
- But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.

Vulnerable time in CSMA



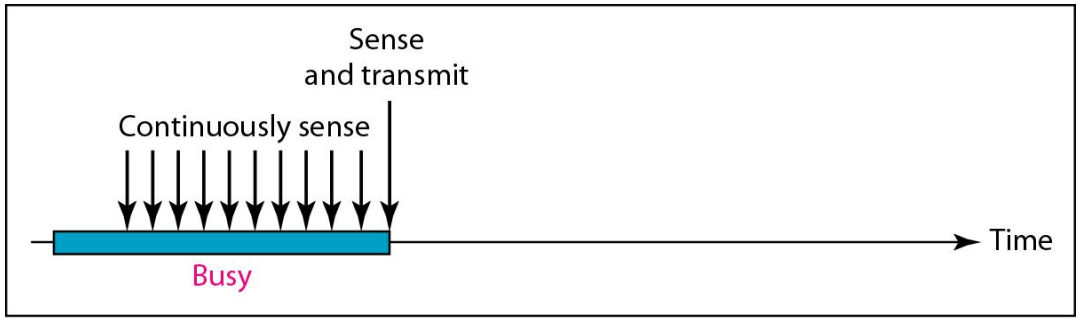
- What should a station do if the channel is busy?
- What should a station do if the channel is idle?
- Three methods have been devised to answer these questions:
  - the I-persistent method,
  - the nonpersistent method, and
  - the p-persistent method.

- **I-Persistent:** The **I-persistent method** is simple and straightforward. **In** this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.
- **Nonpersistent: In** the **nonpersistent method,** a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, **it waits a random amount of time and then senses the line again**. The nonpersistent approach **reduces the chance of collision** because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method **reduces the efficiency** of the network because the **medium remains idle** when there may be stations with frames to send.

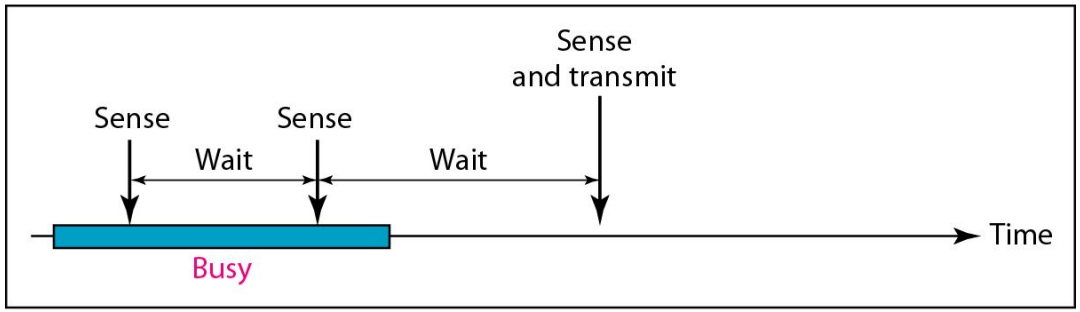
p-Persistent

- **p-Persistent :The p-persistent method** is used if the channel has **time slots with a slot duration equal to or greater than the maximum propagation time**. The p-persistent approach combines the advantages of the other two strategies. **It reduces the chance of collision and improves efficiency.** In this method, after the station finds the line idle it follows these steps:
- 1. With probability *p,* the station sends its frame.
- 2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.
  - a. If the line is idle, it goes to step 1.
  - b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

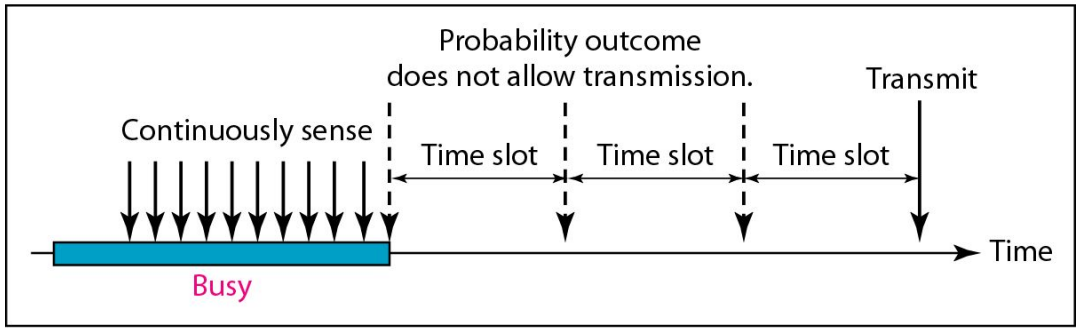
Figure 12.10 Behavior of three persistence methods



a. 1-persistent

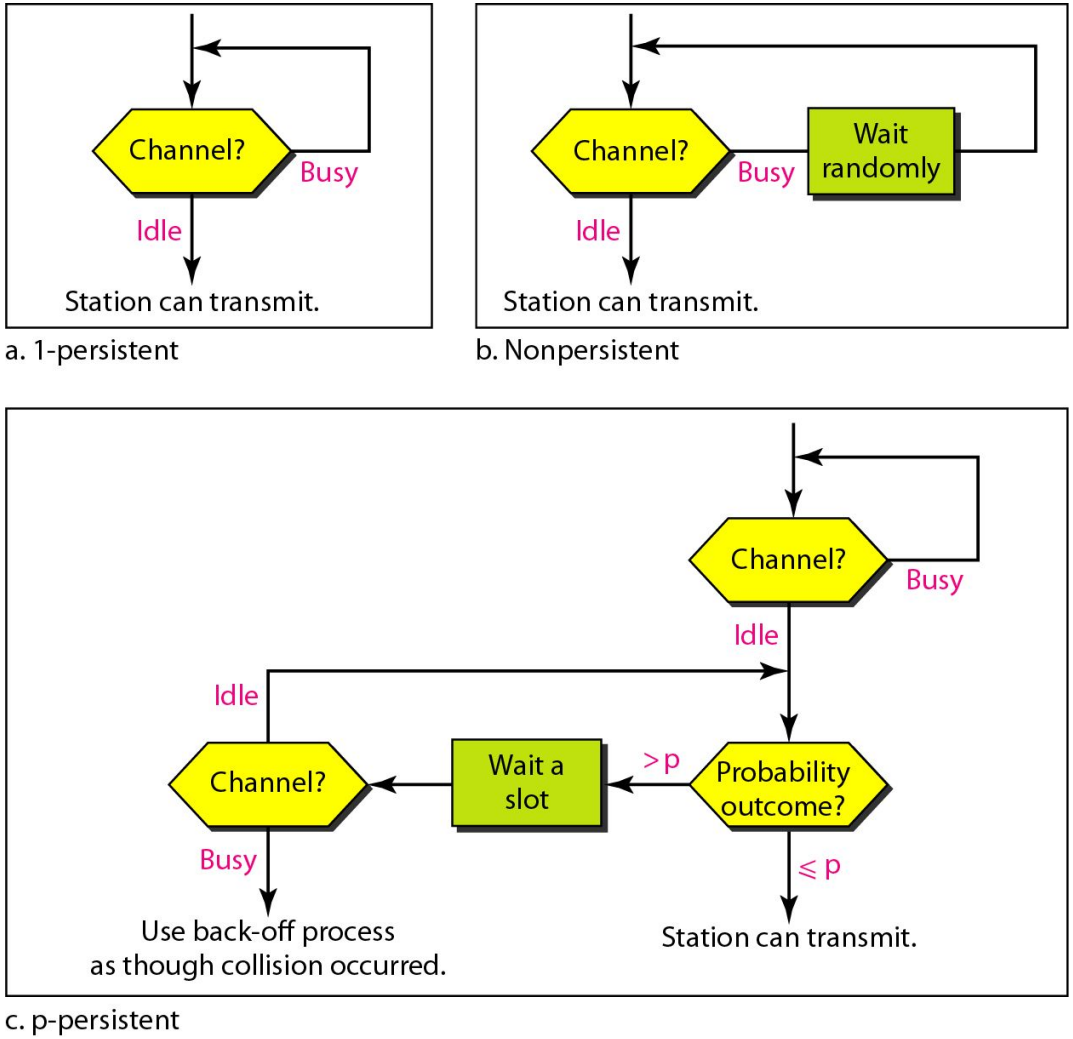


b. Nonpersistent



c. p-persistent

Figure 12.11 Flow diagram for three persistence methods



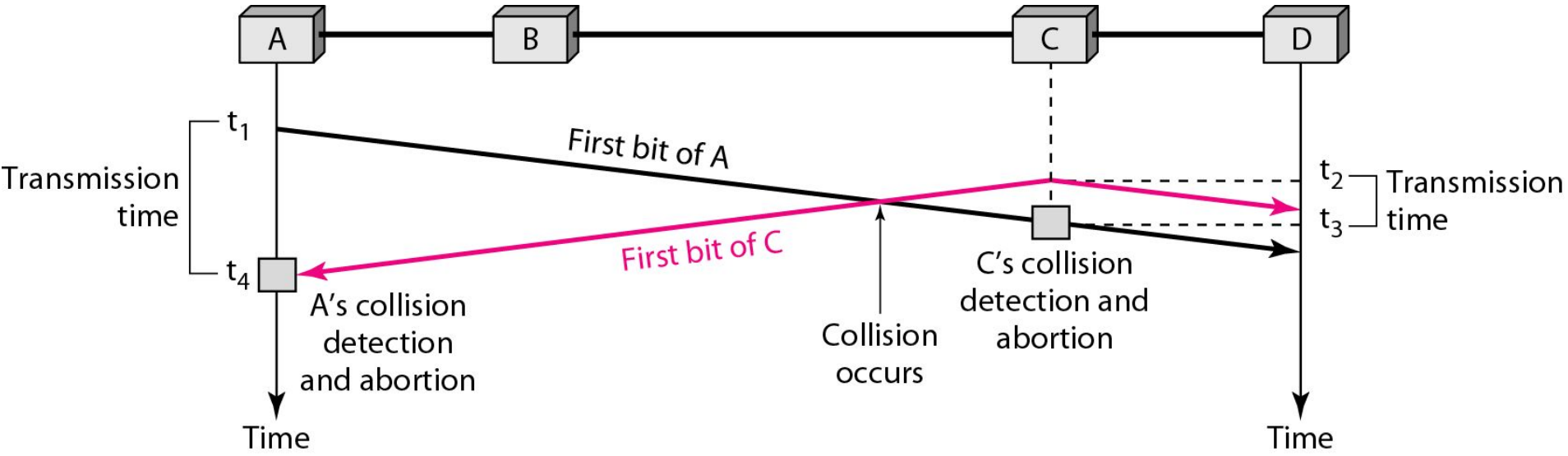
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

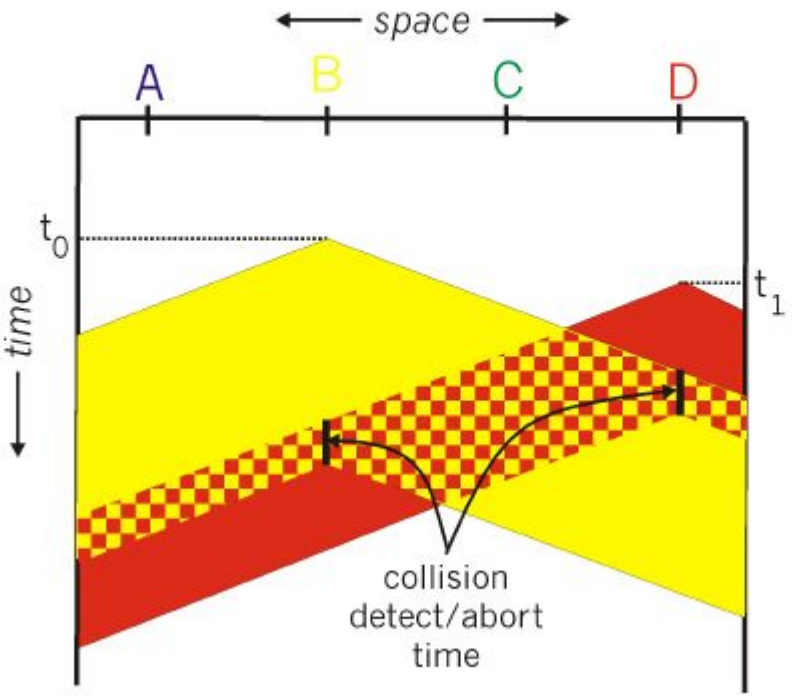
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting
- human analogy: the polite conversationalist

Figure 12.12 Collision of the first bit in CSMA/CD



CSMA/CD collision detection



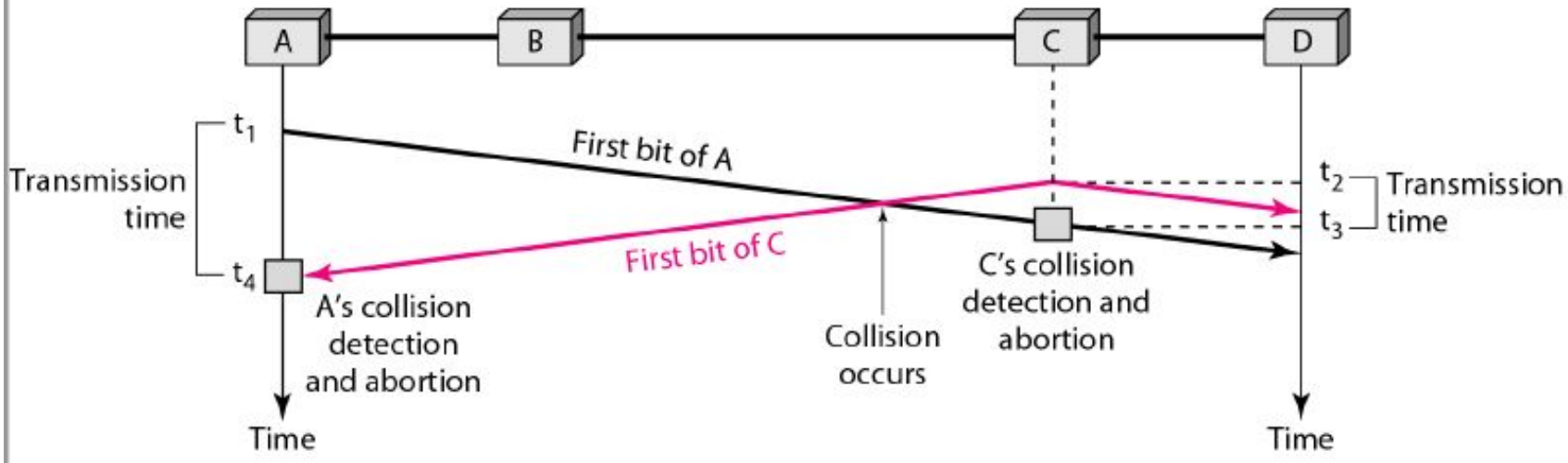
In Figure below, the data rate is 10 Mbps, the distance between station A and C is 2000 m, and the propagation speed is 2×10^8 *mls.Station A starts sending a long* frame at time *t1 =0; station C starts sending a long frame at time t2 =3* μs. *The* size of the frame is long enough to guarantee the detection of collision by both stations. Find:

a. The time when station C hears the collision *(t3)'*

b. The time when station A hears the collision *(t4)'*

c. The number of bits station A has sent before detecting the collision.

d. The number of bits station C has sent before detecting the collision.



Solution:

We have $t_1 = 0$ and $t_2 = 3\ \mu s$

a. $t_3 - t_1 = (2000\ m) / (2 \times 10^8\ m/s) = 10\ \mu s \rightarrow t_3 = 10\ \mu s + t_1 = \mathbf{10\ \mu s}$

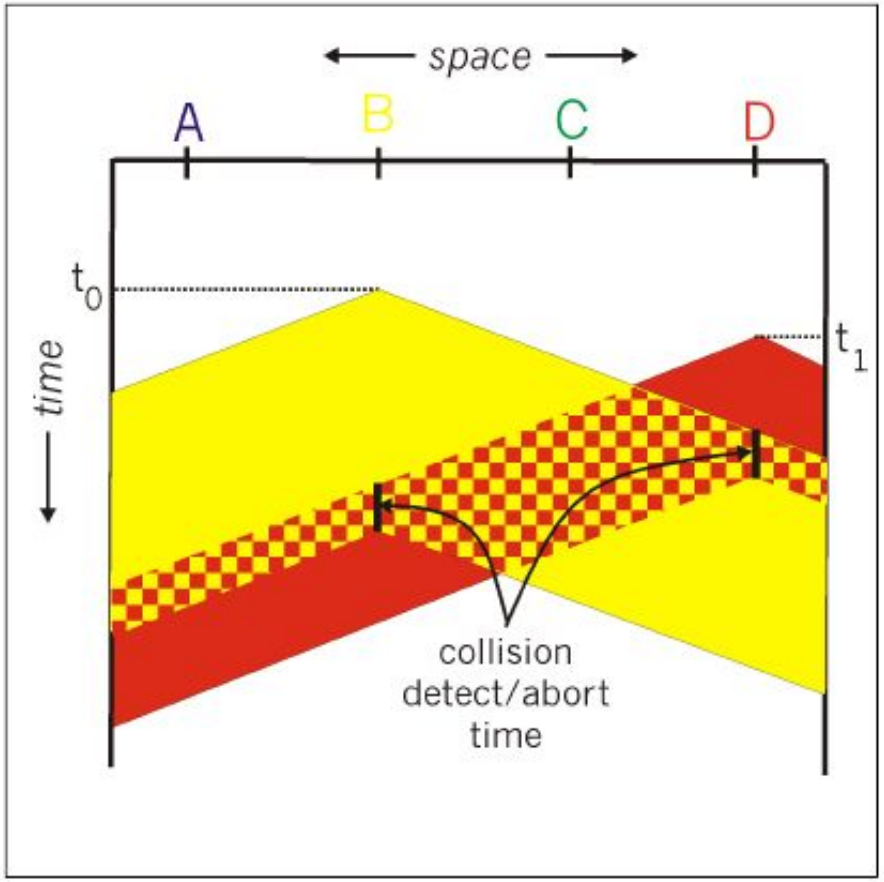
b. $t_4 - t_2 = (2000\ m) / (2 \times 10^8\ m/s) = 10\ \mu s \rightarrow t_4 = 10\ \mu s + t_2 = \mathbf{13\ \mu s}$

c. $T_{fr(A)} = t_4 - t_1 = 13 - 0 = 13\ \mu s \rightarrow Bits_A = 10\ Mbps \times 13\ \mu s = \mathbf{130\ bits}$

d. $T_{fr(C)} = t_3 - t_2 = 10 - 3 = 07 \mu s \rightarrow Bits_C = 10\ Mbps \times 07\ \mu s = \mathbf{70\ bits}$

Ethernet CSMA/CD and Packet Size

- What if two people sent really small packets
  - How do you find collision?
  - Must have a minimum packet size
- Min packet length > 2x max prop delay
  - If A, B are at opposite sides of link, and B starts one link prop delay after A



Example

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μs. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

8.a. (6 points) In a CSMA/CD network with a data rate of 10 Mbps, the maximum distance between any station pair is found to be 2500 m for the correct operation of the collision detection process. What should be the maximum distance if we increase the date rate to 100 Mbps, 1 Gbps, and 10 Gbps?

Answer: Let us find the relationship between the collision domain (maximum length of the network) and the data rate. We know that

Tfr = (frame size) / (data rate) = 2 × Tp = 2 × distance / (propagation speed)

or

distance = [(frame size) (propagation speed)] / [2 × (data rate)]

or

distance = K / (data rate)

This means that distance is inversely proportional to the data rate (K is a constant). When the data rate is increased, the distance or maximum length of network or collision domain is decreased proportionally. In Example 12.5, we mentioned that the maximum distance for a data rate of 10 Mbps is 2500 meters. We calculate the maximum distance based on the above proportionality relationship.

Data rate = 10 Mbps → maximum distance = 2500 m Data rate = 100 Mbps → maximum distance = 250 m Data rate = 1 Gbps → maximum distance = 25 m Data rate = 10 Gbps → maximum distance = 2.5 m

This means that when the data rate is very high, it is almost impossible to have a network using CSMA/CD.

Q: In a *CDMA/CD network with a data rate of 10 Mbps, the minimum frame size is* found to be 512 bits for the correct operation of the collision detection process. What should be the minimum frame size if we increase the data rate to 100 Mbps? To 1 Gbps? To 10 Gbps?

$$T_b = (\text{frame size}) / (\text{data rate}) = 2 \times T_p = 2 \times \text{distance} / (\text{propagation speed})$$

or

$$(\text{frame size}) = [2 \times (\text{distance}) / (\text{propagation speed})] \times (\text{data rate})$$

or

$$\textbf{(frame size) = K} \times \textbf{(data rate)}$$

Data rate =	10 Mbps	→	minimum frame size = **512 bits**
Data rate =	100 Mbps	→	minimum frame size = **5120 bits**
Data rate =	1 Gbps	→	minimum frame size = **51,200 bits**
Data rate =	10 Gbps	→	minimum frame size = **512,000 bits**

Question:

Suppose the round trip propagation delay for a 10 Mbps Ethernet having 48-bit jamming signal is 46.4 ms. The minimum frame size is:

(a) 94
(b) 416
(c) 464
(d) 512

Answer (c)

Transmission Speed = 10Mbps.
Round trip propagation delay = 46.4 ms
The minimum frame size = (Round Trip Propagation Delay) * (Transmission Speed) = 10*(10^6)*46.4*(10^-3) = 464 * 10^3 = 464 Kbit

Types of CSMA protocols

- There are several types of CSMA protocols:
  - 1-Persistent CSMA
  - Non-Persistent CSMA
  - P-Persistent CSMA

1-Persistent CSMA

- Sense the channel.
  - If busy, keep listening to the channel and transmit immediately when the channel becomes idle.
  - If idle, transmit a packet immediately.
- If collision occurs,
  - Wait a random amount of time and start over again.
- The protocol is called 1-persistent because the host transmits with a probability of 1 whenever it finds the channel idle.

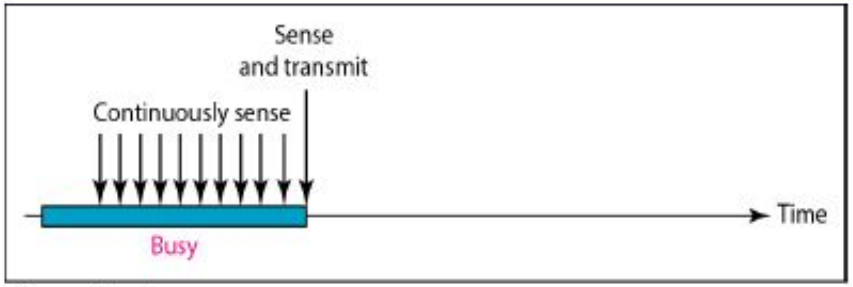
Non-Persistent CSMA

- Sense the channel.
  - If busy, wait a random amount of time and sense the channel again
  - If idle, transmit a packet immediately
- If collision occurs
  - wait a random amount of time and start all over again

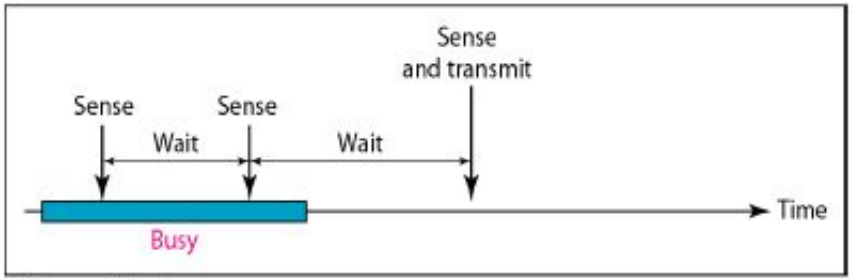
P-Persistent CSMA

- Optimal strategy: use P-Persistent CSMA
- Assume channels are slotted
- One slot = contention period (i.e., one round trip propagation delay)
- 1. Sense the channel
  - If channel is idle, transmit a packet with probability p
    - if a packet was transmitted, go to step 2
    - if a packet was not transmitted, wait one slot and go to step 1
  - If channel is busy, wait one slot and go to step 1.
- 2. Detect collisions
  - If a collision occurs, wait a random amount of time and go to step 1

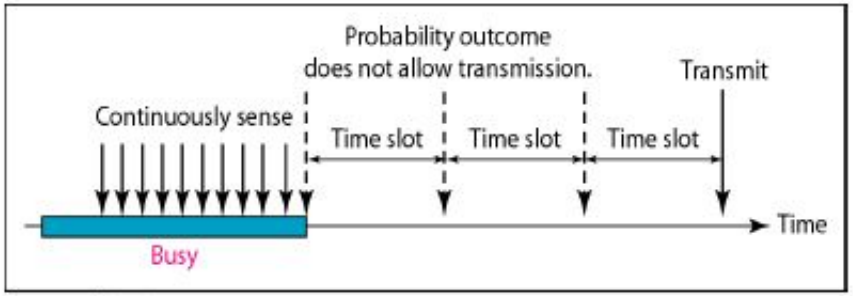
Behavior of three persistence methods



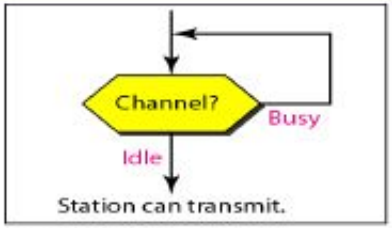
a. 1-persistent



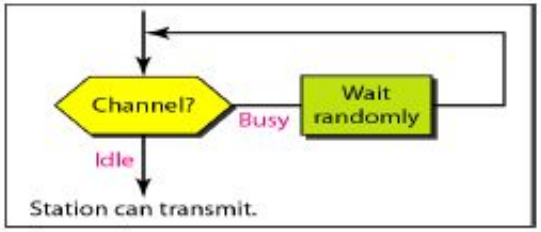
b. Nonpersistent



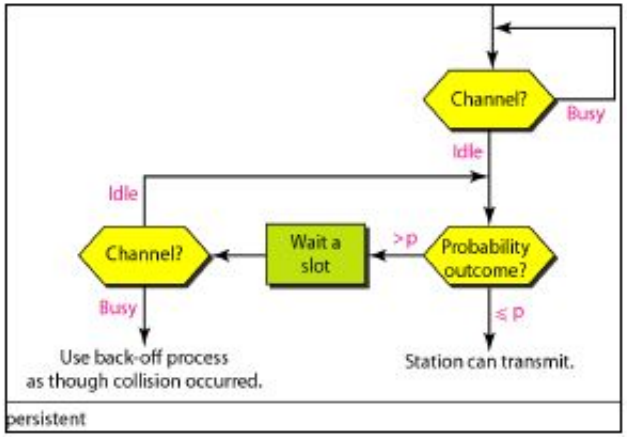
c. p-persistent



a. 1-persistent

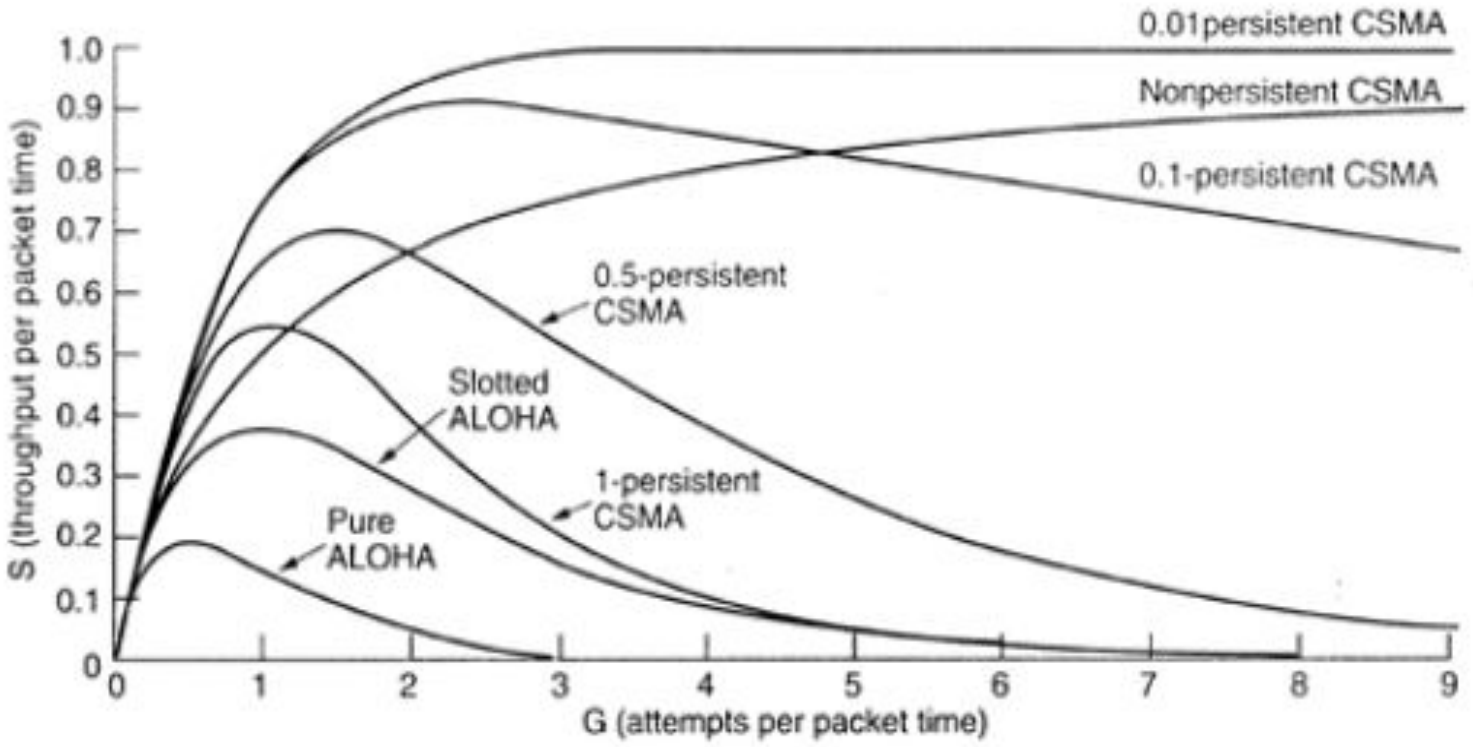


b. Nonpersistent



persistent

Comparison of CSMA and ALOHA Protocols



(Number of Channel Contenders)

Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame (96-bit time
2. Carrier sensing: If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. Collision detection: If adapter detects another transmission while transmitting, aborts and sends jam signal (make sure all adapters see collision: 48 bits)
5. Random access: After aborting, adapter enters **exponential backoff** before returning to Step 2
   - after *m* th collision, choose *K* randomly out of {0,1,2,…2^m-1}. Wait *K**512 bit times
   - 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}

CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

Suppose two nodes, *A* and *B*, are attached to opposite ends of a 900 m cable, and that they each have one frame of 1,000 bytes (including all headers and preambles) to send to each other. Both nodes attempt to transmit at time t=0. Suppose there are four hubs between *A* and *B*, each inserting a 20-bit delay. Assume the transmission rate is 10 Mbps, and CSMA/CD with backoff intervals of multiples of 512 bits is used. After the 1st collision, *A* draws K=0 and *B* draws K=1 in the exponential backoff protocol. Ignore the jam signal and the 96 bit-time delay.

(a) What is the one-way propagation delay (including hub delays) between *A* and *B* in seconds? Assume that the signal propagation speed is 2×10^8 m/sec.

(b) At what time (in seconds) is *A*'s packet completely delivered to *B*?

(c) Now suppose that only *A* has a packet to send and that the hubs are replaced with switches. Suppose that each switch has a 20-bit processing delay in addition to a store-and-forward delay. At what time, in seconds, is *A*'s packet delivered at *B*?

bits

bits

a)

$$\frac{900m}{2 \cdot 10^8 m/\sec} + 4 \cdot \frac{20bits}{10 \times 10^6 bps}$$

$$= (4.5 \times 10^{-6} + 8 \times 10^{-6}) \sec$$

$$= 12.5 \mu \sec$$

b)

- At time $t = 0$, both A and B transmit.
- At time $t = 12.5 \mu \sec$, A detects a collision.
- At time $t = 25 \mu \sec$ last bit of B's aborted transmission arrives at A. (A picks K = 0 and (ignoring jam signal) waits for the channel to be idle. The channel becomes idle at time t = 25μs. This is the time last bit from B arrives at A.)
- A retransmits at t = 25μs (ignoring 96-bit time delay). Thus At time $t = 25 + 12.5 = 37.5 \mu \sec$ first bit of A's retransmission arrives at B.
- At time $t = 37.5 \mu \sec + \frac{1000bits}{10 \times 10^6 bps} = 137.5 \mu \sec$ A's packet is completely delivered at B. → A finishes transmission at time t = 125μs. Last bit from A arrives at B at time 137.5μs.

c) $12.5\mu\text{sec} + 5 \cdot 100\mu\text{sec} = 512.5\mu\text{sec}$ →

Each switch introduces additional 1000-bit store-and-forward delay and 20-bit processing delay. Total delay introduced is 4080-bit time or 408 μs. Transmission delay is 1000-bit time or 100 μs. Propagation delay is 4.5 μs. A's packet reaches B at time

$$408 + 100 + 4.5 = 512.5\mu\text{s}$$

Question:

Suppose two nodes, A and B, are attached to opposite ends of a 900 m cable, and that they each have one frame of 1000 bits (including all headers and preambles) to send to each other. Both nodes attempt to transmit at time t=0. Suppose there are four repeaters between A and B, each inserting a 20 bit delay. Assume the transmission rate is 10 Mbps, and CSMA/CD with backoff intervals of multiples of 512 bits is used. After the first collision, A draws K=0 and B draws K=1 in the exponential backoff protocol. Ignore the jam signal.

a) What is the one-way propagation delay (including repeater delays) between A and B in seconds. Assume that the signal propragation speed is $2 * 10^8$m/sec.

b) At what time (in seconds) is A's packet completely delivered at B.

c) Now suppose that only A has a packet to send and that the repeaters are replaced with bridges. Suppose that each bridge has a 20 bit processing delay in addition to a store-and-forward delay. At what time in seconds is A's packet delivered at B?

Solution:

a)

$$\frac{900m}{2\cdot 10^8 m/\sec} + 4\cdot\frac{20bits}{10\times 10^6 bps}$$

$$= (4.5\times 10^{-6} + 8\times 10^{-6})\sec$$
$$= 12.5\mu\sec$$

b)

- At time $t = 0$, both A and B transmit.
- At time $t = 12.5\mu\sec$, A detects a collision.
- At time $t = 25\mu\sec$ last bit of B's aborted transmission arrives at A.
- At time $t = 37.5\mu\sec$ first bit of A's retransmission arrives at B.
- At time $t = 37.5\mu\sec + \frac{1000bits}{10\times 10^6 bps} = 137.5\mu\sec$ A's packet is completely delivered at B.

c) $12.5\mu\sec + 5\cdot 100\mu\sec = 512.5\mu\sec$

Question:

Suppose nodes A and B are on the same 10 Mbps Ethernet segment, and the propagation delay between the two nodes is 225 bit times. Suppose A and B send frames at the same time, the frames collide, and then A and B choose different values of K in the CSMA/CD algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide? For our purposes, it suffices to work out the following example. Suppose A and B begin transmission at t=0 bit times. They both detect collisions at t=225 bit times. They finish transmitting jam signal at t= 225+48= 273 bit times. Suppose K_A=0 and

K_B=1. At what time does B schedule its retransmission? At what time does A begin transmission? (Note, the nodes must wait for an idle channel after returning to Step 2-- see protocol.) At what time does A's signal reach B? Does B refrain from transmitting at its scheduled time?

Solution..

Time, t	Event
0	A and B begin transmission
225	A and B detect collision
273	A and B finish transmitting jam signal
273+225 = 498	B's last bit arrives at A; A detects an idle channel
498+96=594	A starts transmitting
273+512 = 785	B returns to Step2 B must sense idle channel for 96 bit times before it transmits
594+225=819	A's transmission reaches B

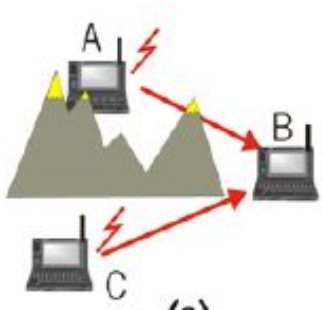
Because A's retransmission reaches B before B's scheduled retransmission time, B refrains from transmitting while A retransmits. Thus A and B do not collide. Thus the factor 512 appearing in the exponential backoff algorithm is sufficiently large.

CSMA/CD problems

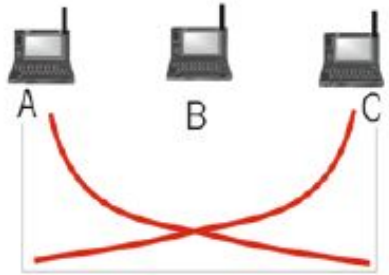
- Can CSMA/CD work over wireless LANs?
  - Collision detection difficult in wireless LANs: receiver shut off while transmitting
  - Hidden terminal problem

Hidden Terminal effect

- A, C cannot hear each other
  - obstacles, signal attenuation
  - Neither A nor C can tell if they collide at B



(a)

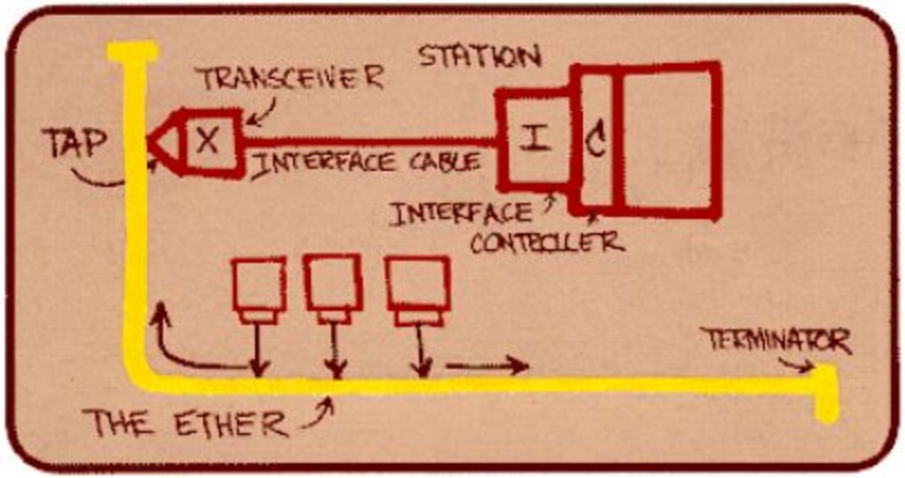


(b)

Ethernet

"dominant" LAN technology:

- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps

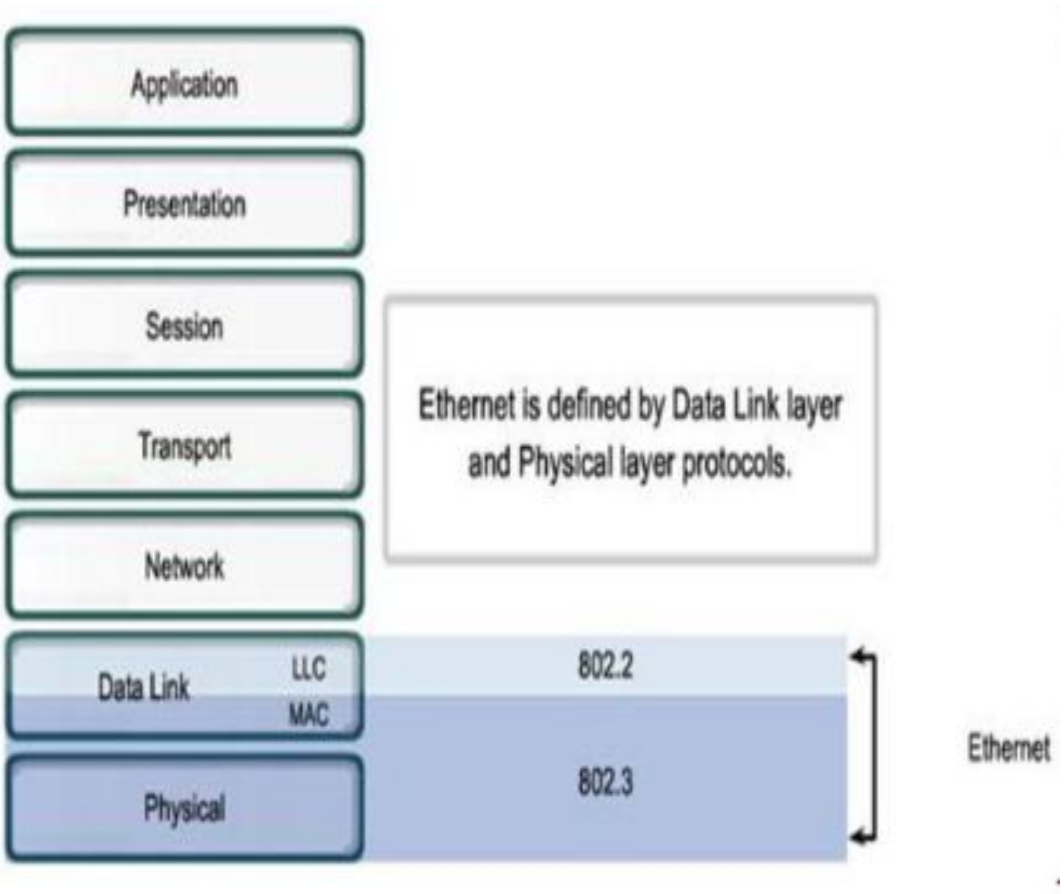


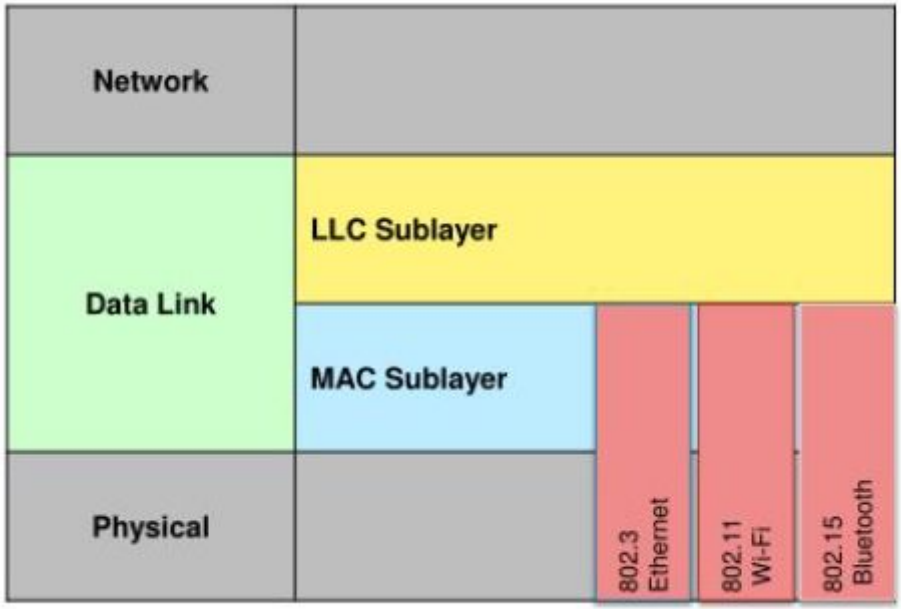
Metcalfe's Ethernet sketch

Ethernet

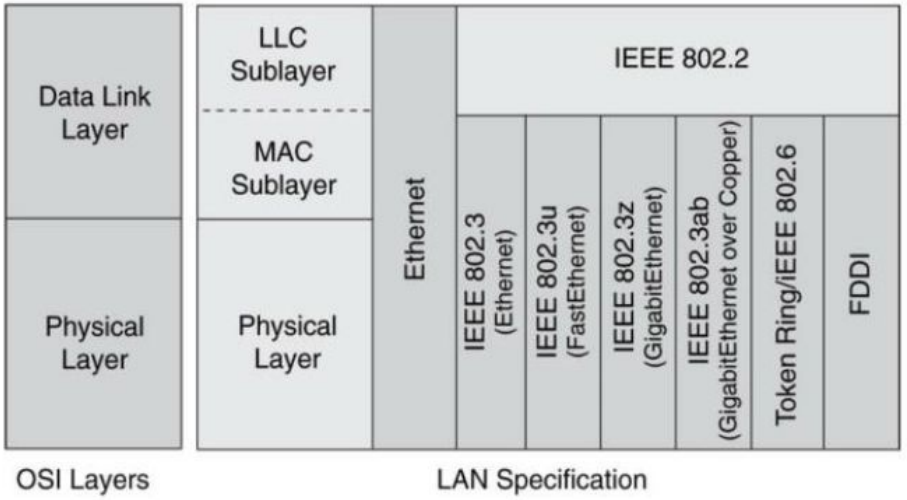
Ethernet standards –

- Most widely used LAN technology
- Operates in the data link layer (**Layer 2** protocols) and the physical layer (**Layer 1** technologies)
- Supports data bandwidths of **10, 100, 1000, 10,000, 40,000, and 100,000 Mbps** (100 Gbps)
- Define and Two separate sub layers of the **data link layer** to operate - **Logical link control (LLC)** and the **MAC sublayers**
- Family of networking technologies that are defined in the **IEEE 802.2 - LLC** and **802.3 standards-MAC+Physical**


Application
Presentation
Session
Transport
Network
Data Link
LLC
MAC
Physical
Ethernet is defined by Data Link layer and Physical layer protocols.
802.2
802.3
Ethernet



Layer 2 Standards

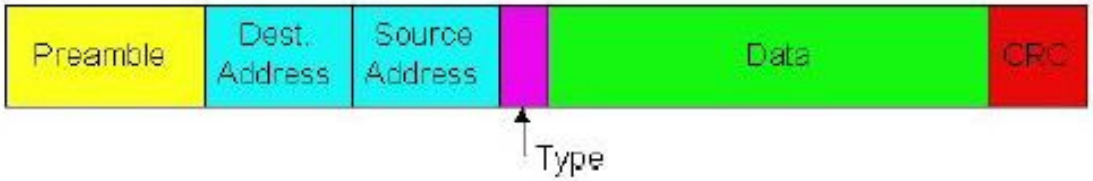


Data Link Layer Standards

Standard organization	Networking Standards
IEEE	• 802.2: Logical Link Control (LLC) • 802.3: Ethernet • 802.4: Token bus • 802.5: Token passing • 802.11: Wireless LAN (WLAN) & Mesh (Wi-Fi certification) • 802.15: Bluetooth • 802.16: WiMax
ITU-T	• G.992: ADSL • G.8100 - G.8199: MPLS over Transport aspects • Q.921: ISDN • Q.922: Frame Relay
ISO	• HDLC (High Level Data Link Control) • ISO 9314: FDDI Media Access Control (MAC)
ANSI	• X3T9.5 and X3T12: Fiber Distributed Data Interface (FDDI)

Ethernet Frame Structure

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

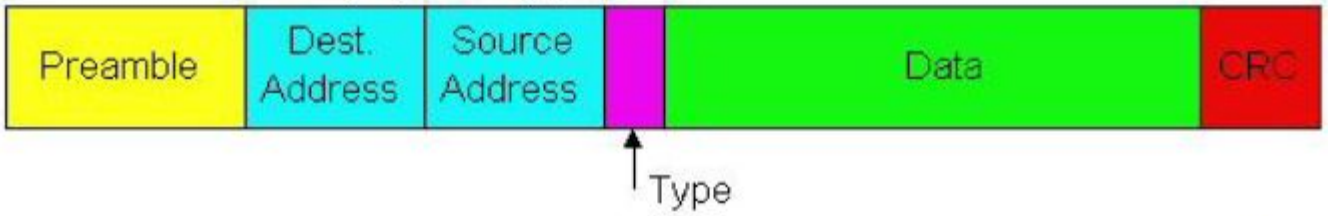


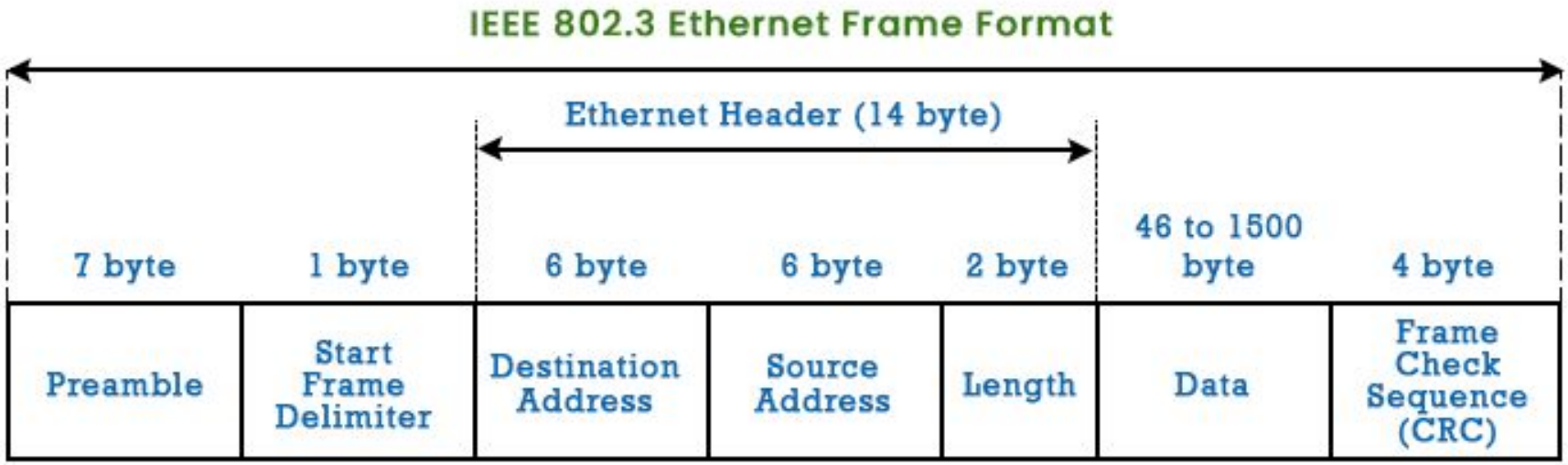
Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 for SFD
- used to synchronize receiver, sender clock rates

Ethernet Frame Structure (more)

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk)
- CRC: checked at receiver, if error is detected, the frame is simply dropped




IEEE 802.3 Ethernet Frame Format
Ethernet Header (14 byte)
7 byte
1 byte
6 byte
6 byte
2 byte
46 to 1500 byte
4 byte
Preamble
Start Frame Delimiter
Destination Address
Source Address
Length
Data
Frame Check Sequence (CRC)

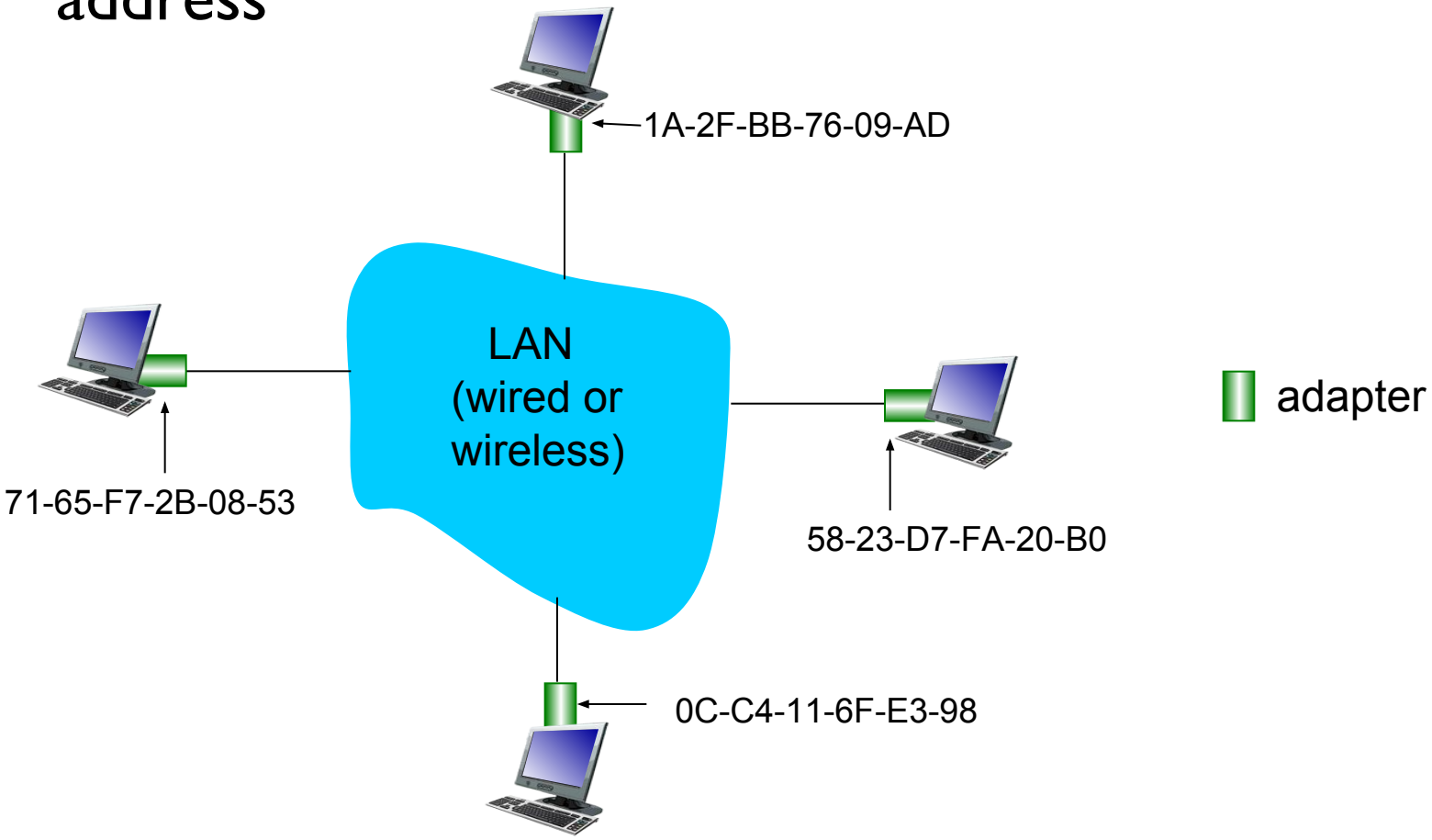
MAC addresses and ARP

- 32/128-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
  - function: *used 'locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "number" represents 4 bits)

LAN addresses and ARP

each adapter on LAN has unique LAN / MAC address

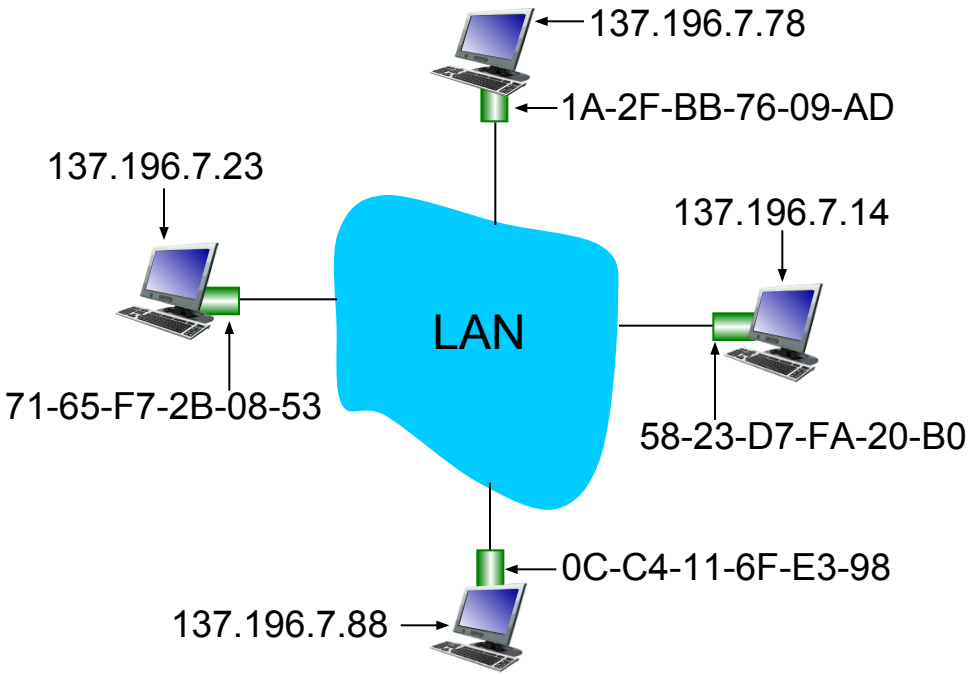


LAN addresses (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address ➜ portability
  - can move LAN card from one LAN to another
- IP hierarchical address *not* portable
  - address depends on IP subnet to which node is attached

ARP: address resolution protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

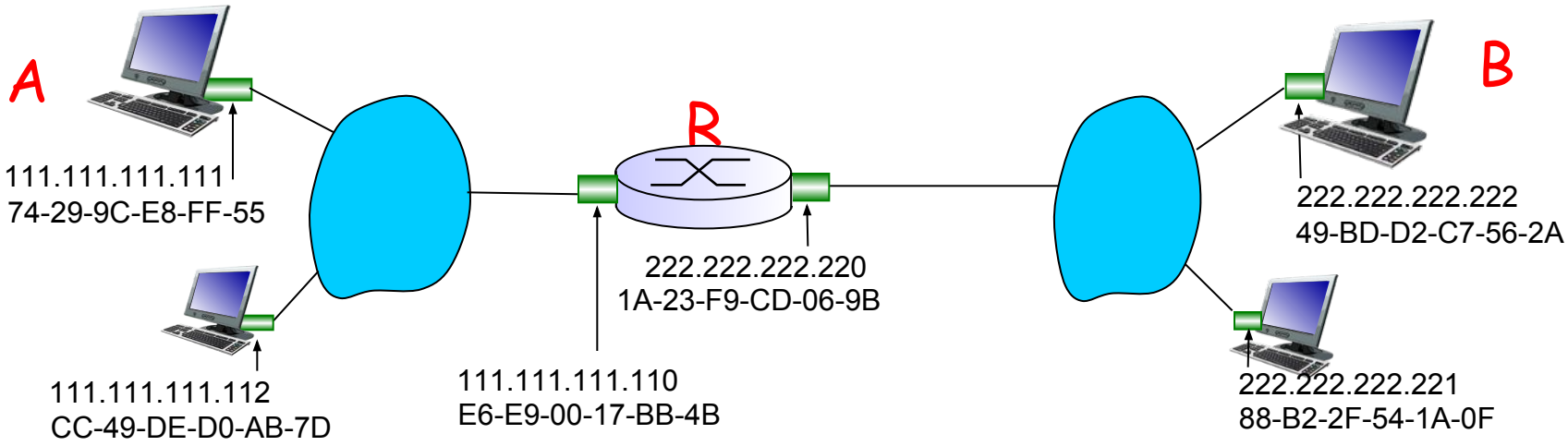
ARP protocol: same LAN

- A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables *without intervention from net administrator*

Addressing: routing to another LAN

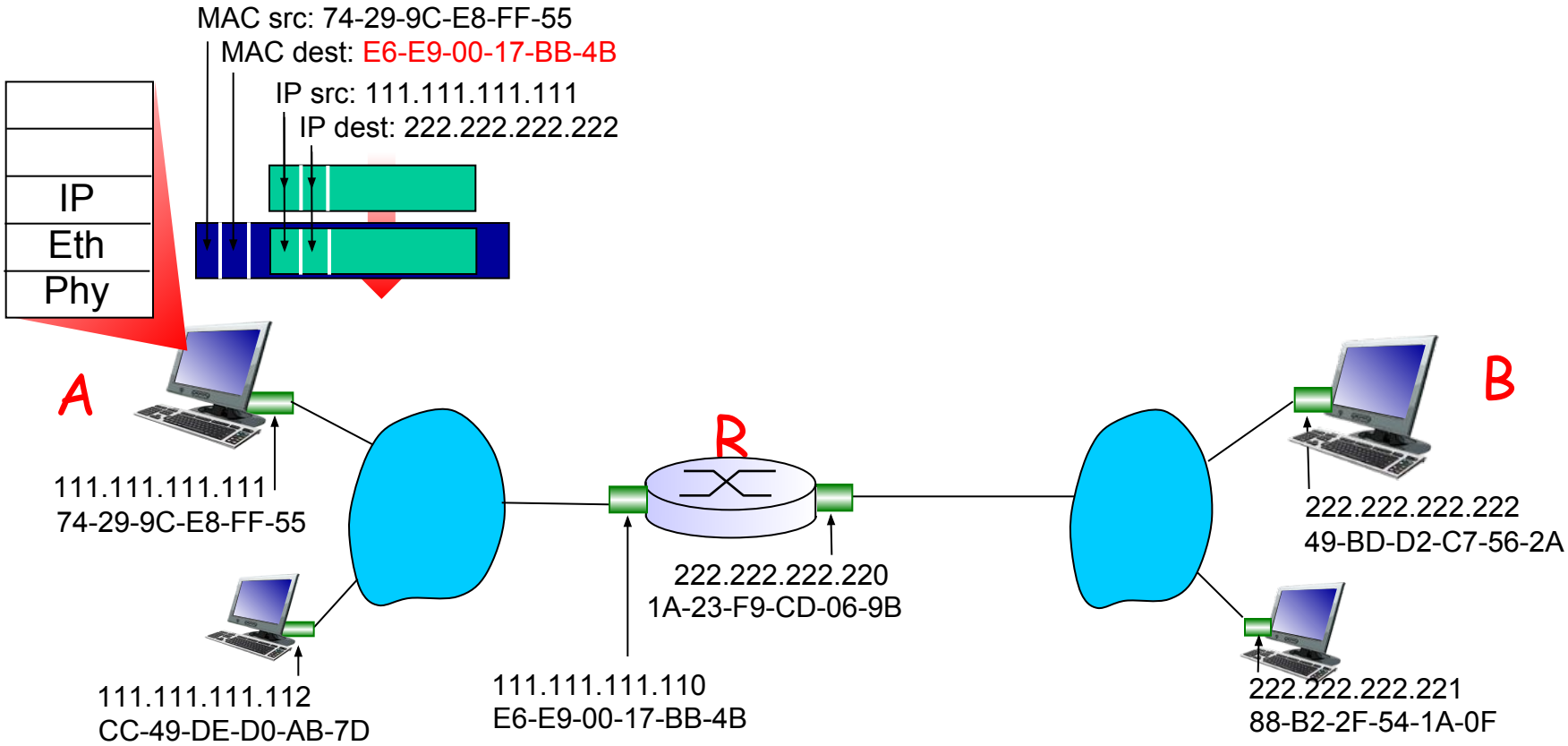
walkthrough: send datagram from A to B via R

- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)



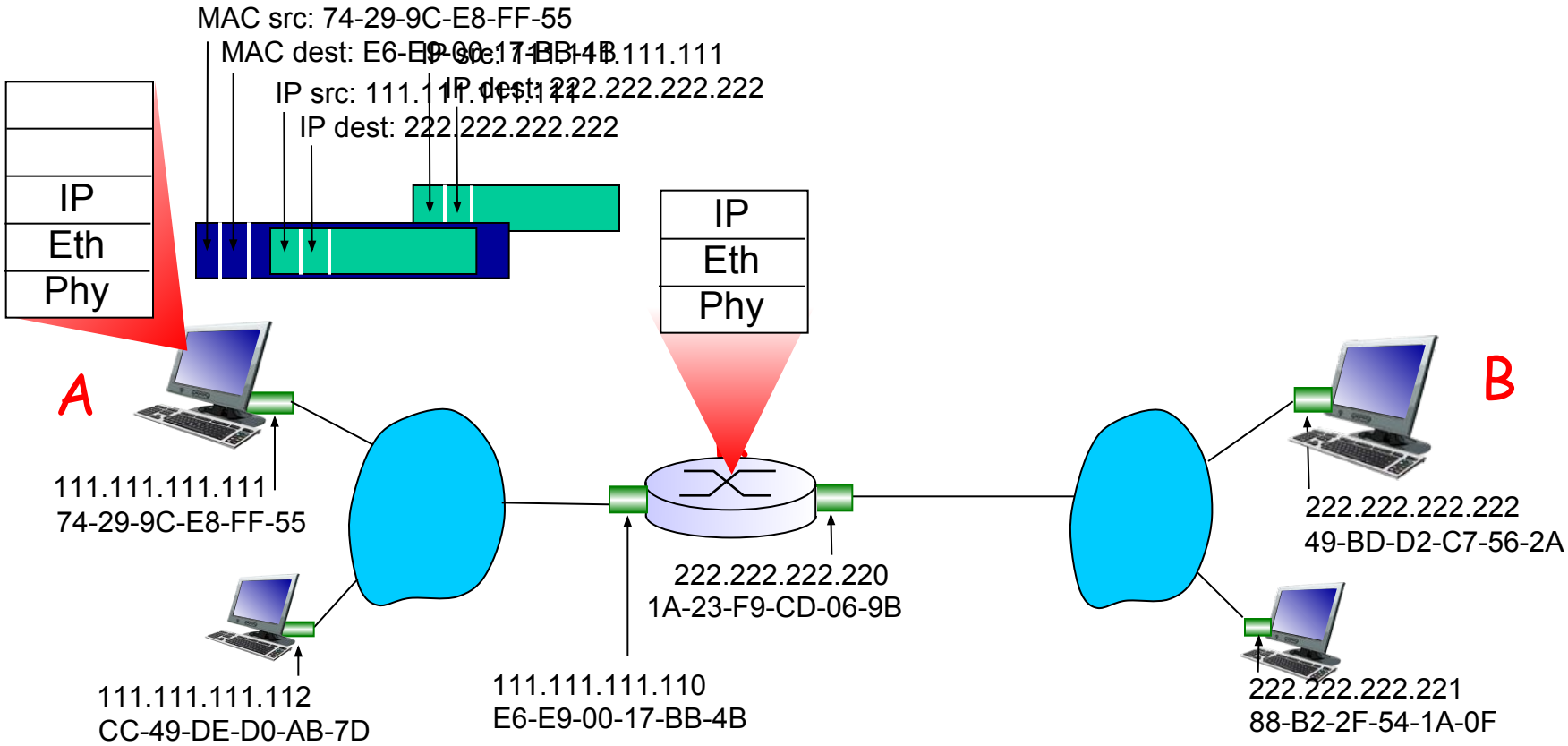
Addressing: routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



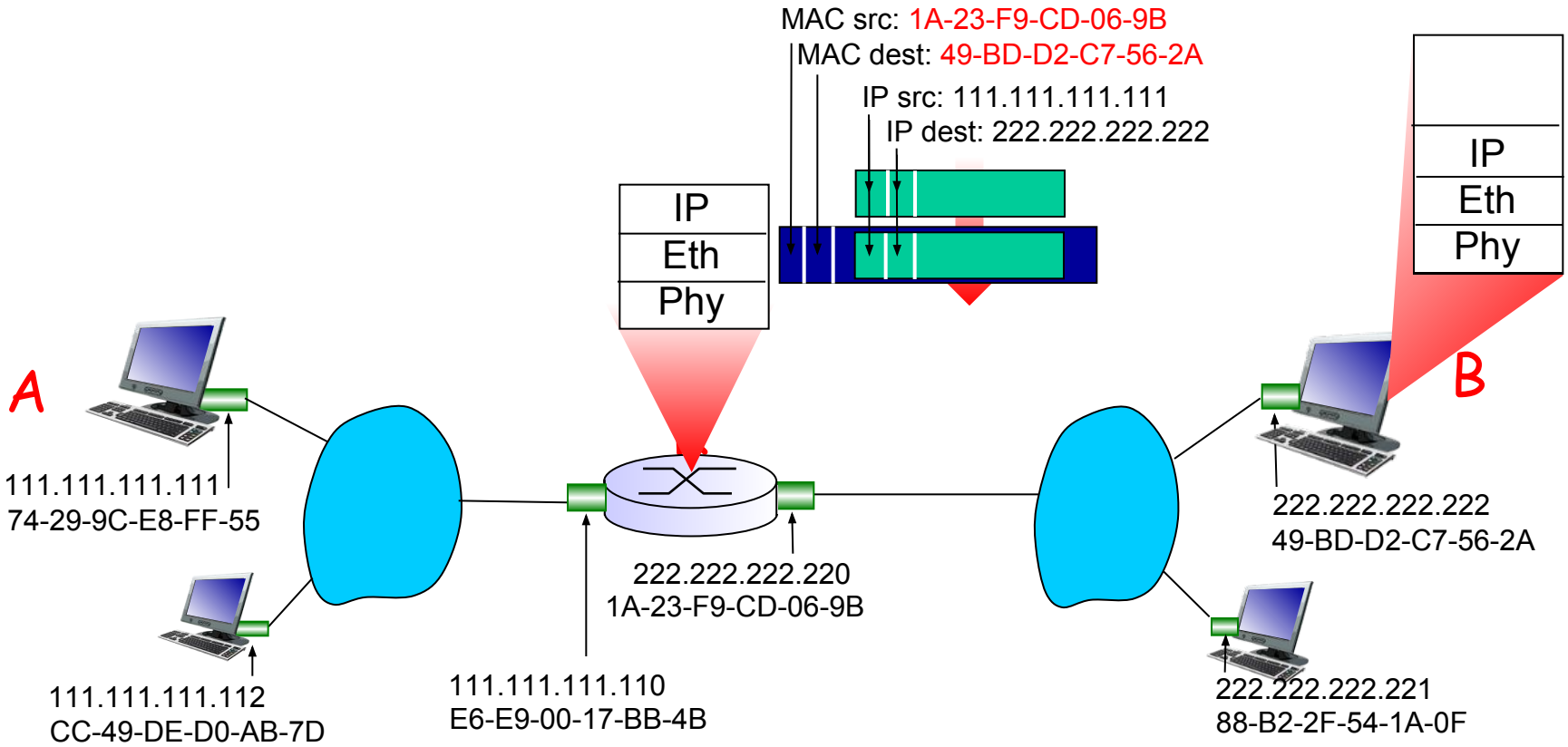
Addressing: routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



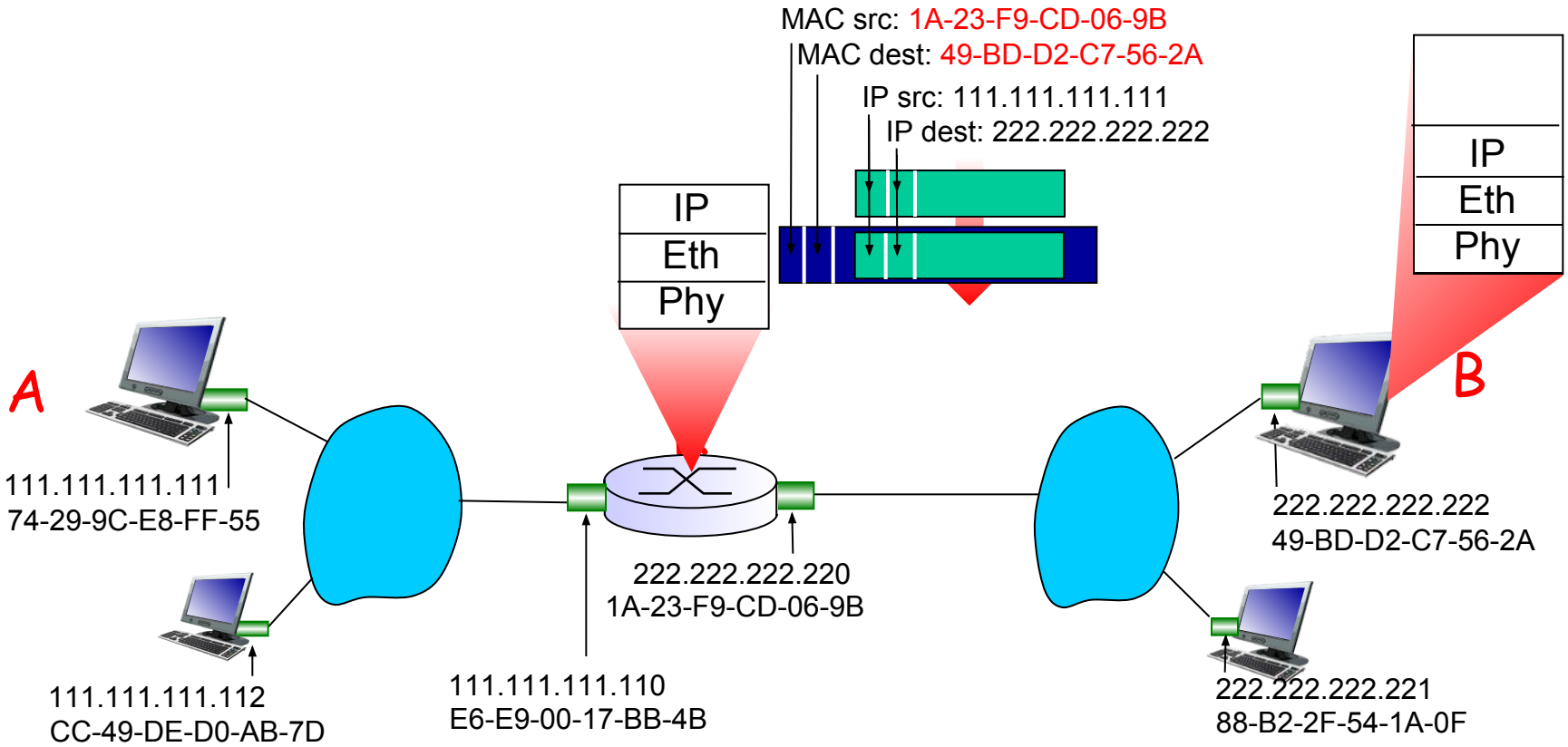
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



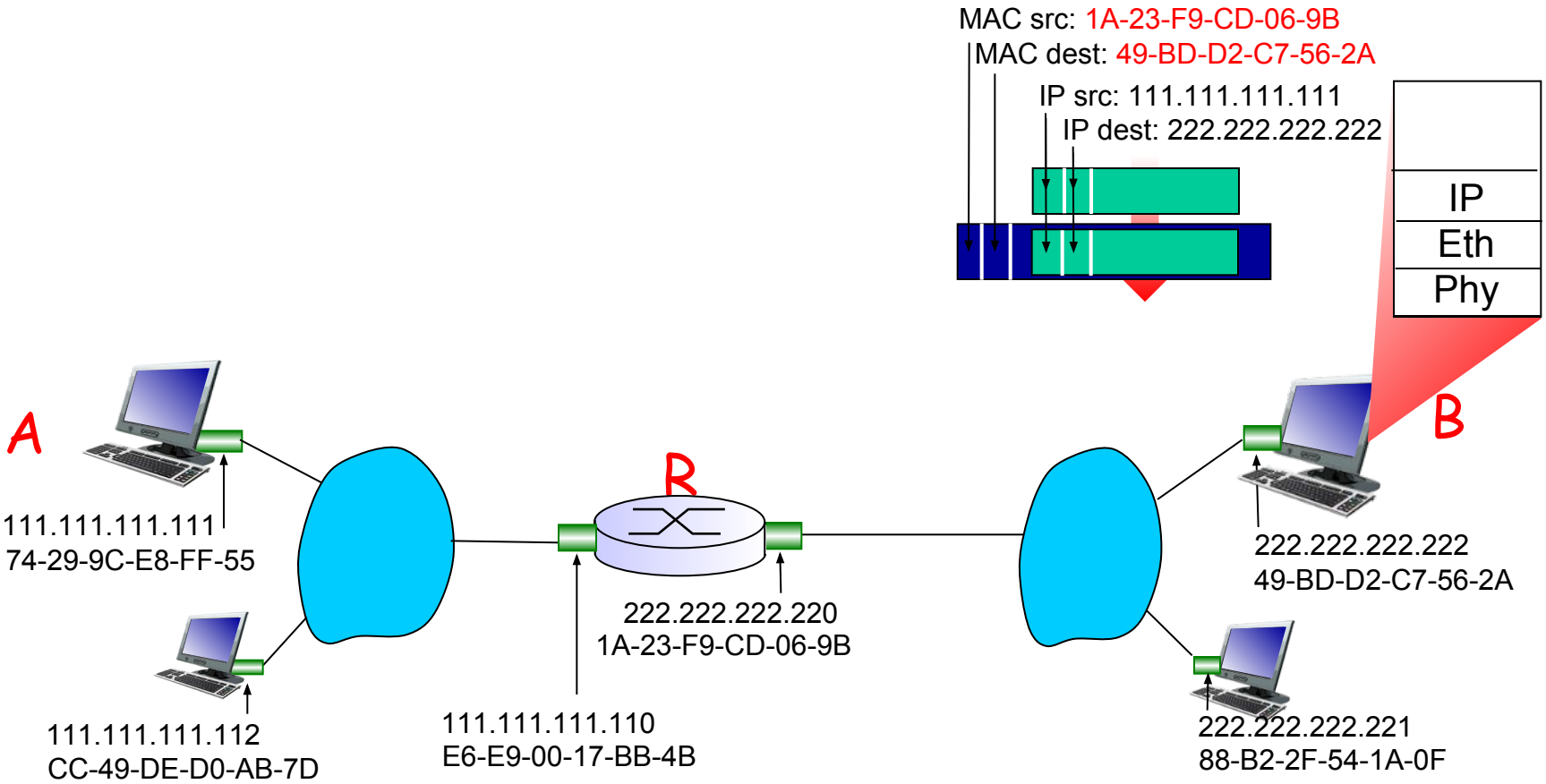
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



- https://slideplayer.com/slide/9860883/