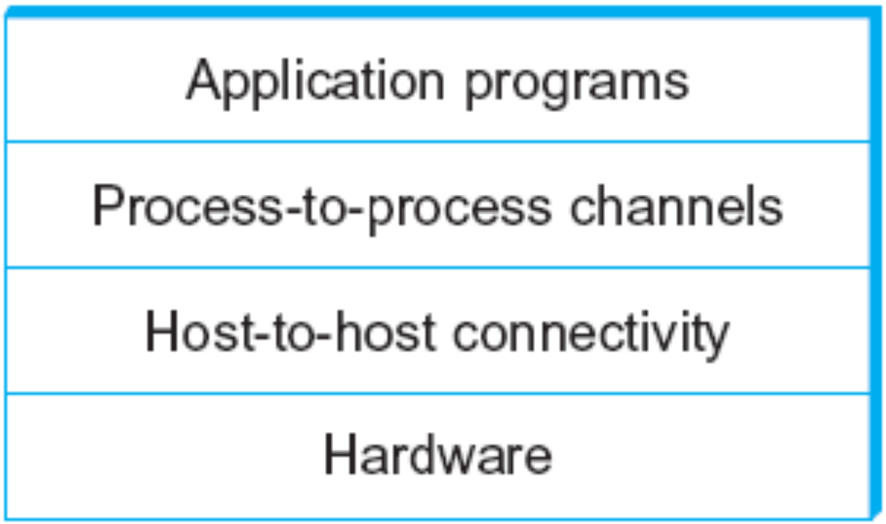
Networks and Distributed Systems

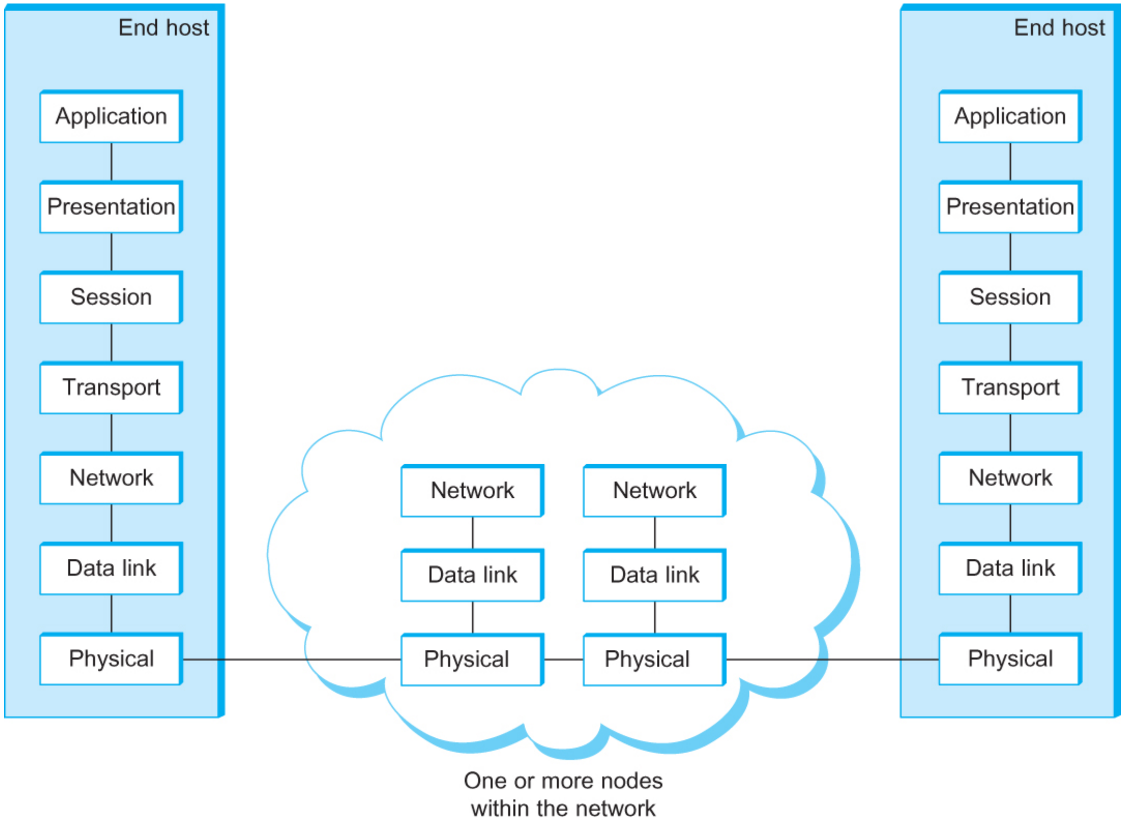
Lecture 13 – Review

Network Architecture



Example of a layered network system

OSI Architecture



The OSI 7-layer Model

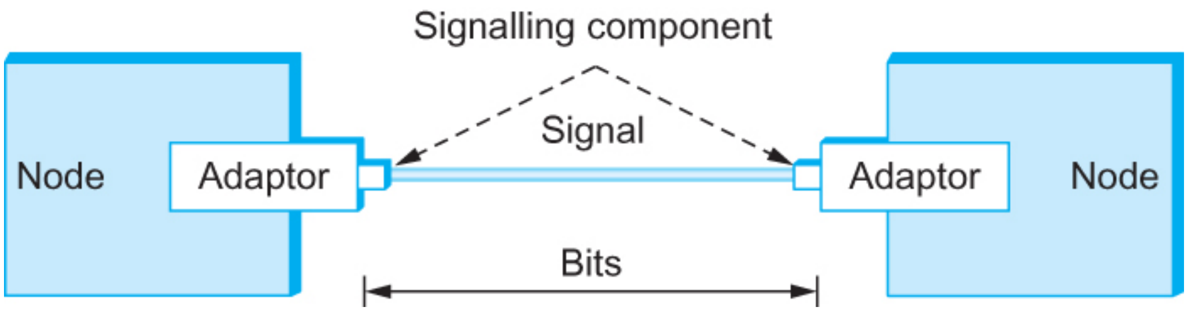
OSI – Open Systems Interconnection

TWO CONCEPTS REQUIRED FOR PHYSICAL LAYER

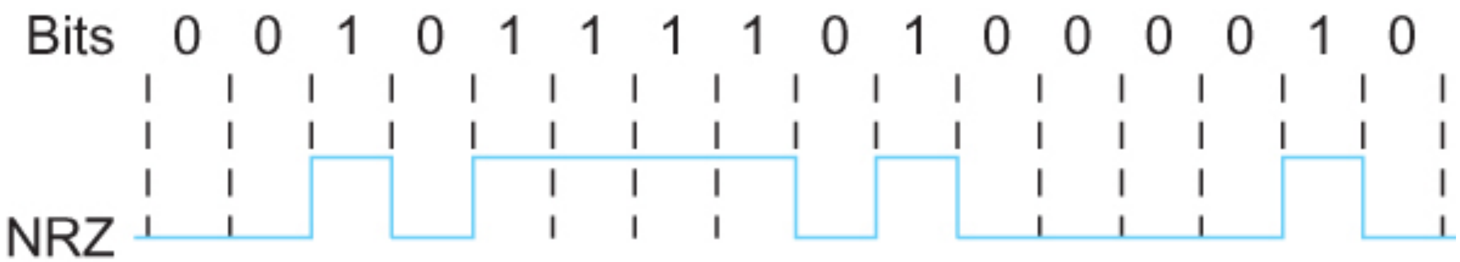
Two concepts required for physical layer

- Encoding
- Framing

Encoding



Signals travel between signaling components; bits flow between adaptors



NRZ encoding of a bit stream

PROBLEMS WITH NRZ

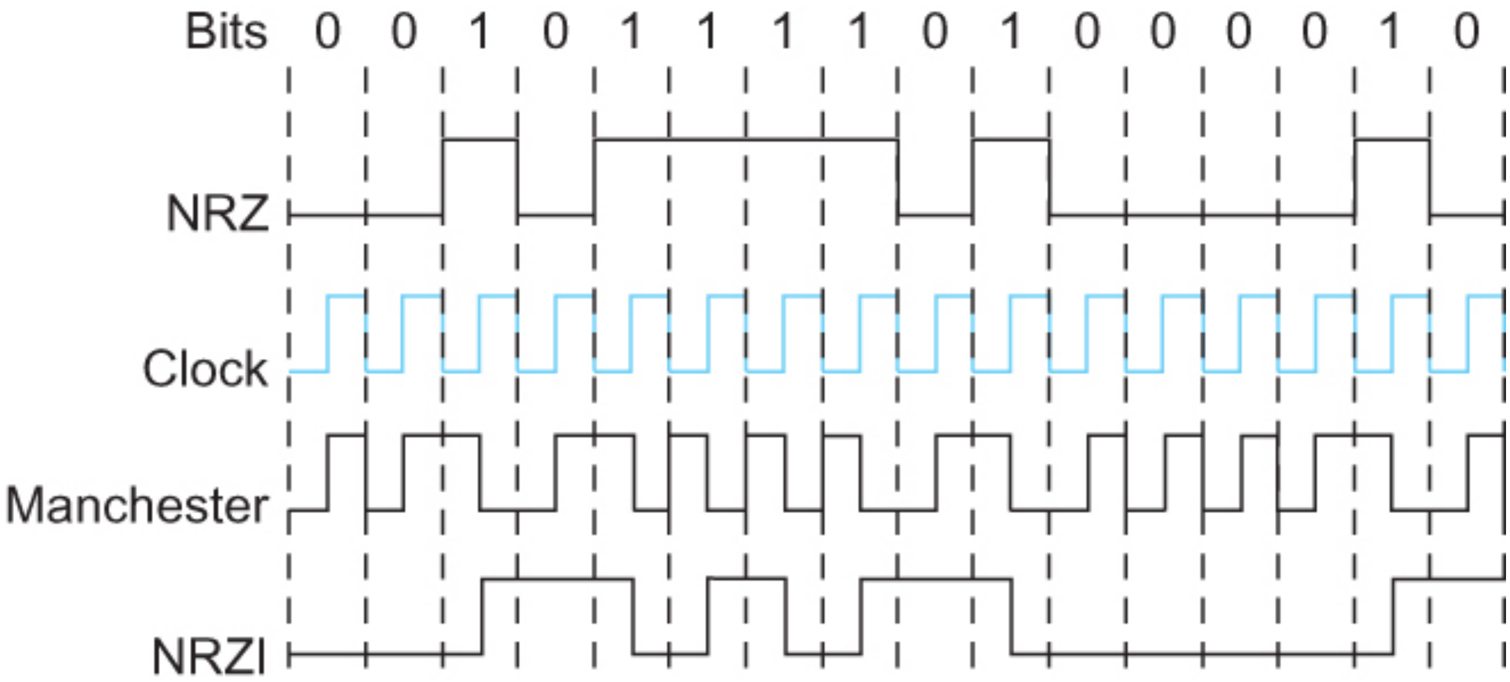
Encoding

- Problem with NRZ
  - Baseline wander
    - The receiver keeps an average of the signals it has seen so far
    - Uses the average to distinguish between low and high signal
    - When a signal is significantly low than the average, it is 0, else it is 1
    - Too many consecutive 0's and 1's cause this average to change, making it difficult to detect

Encoding

- Problem with NRZ
  - Clock recovery
    - Frequent transition from high to low or vice versa are necessary to enable clock recovery
    - Both the sending and decoding process is driven by a clock
    - Every clock cycle, the sender transmits a bit and the receiver recovers a bit
    - The sender and receiver have to be precisely synchronized

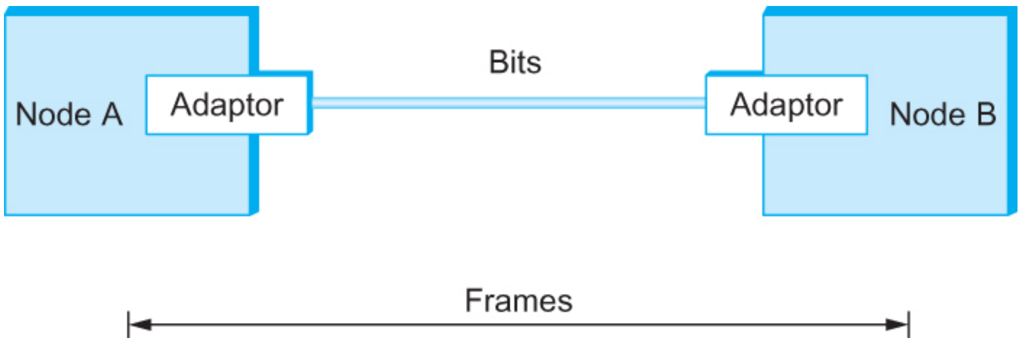
Encoding



Different encoding strategies

Framing

- We are focusing on packet-switched networks, which means that blocks of data (called *frames* at this level), not bit streams, are exchanged between nodes.
- It is the network adaptor that enables the nodes to exchange frames.



Bits flow between adaptors, frames between hosts

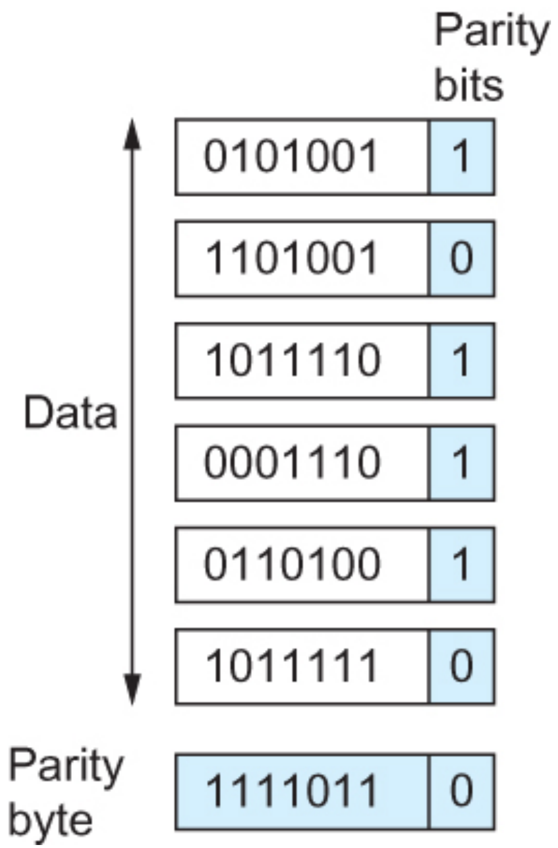
Framing

- Byte-Oriented
- Bit-Oriented

Error Detection

- Common technique for detecting transmission error
  - CRC (Cyclic Redundancy Check)
    - Used in HDLC, DDCMP, CSMA/CD, Token Ring
  - Other approaches
    - Two Dimensional Parity (BISYNC)
    - Checksum (IP)

Two-dimensional parity



Two Dimensional Parity

Internet Checksum Algorithm

- Consider the data being checksummed as a sequence of 16-bit integers.
- Add them together using 16-bit ones complement arithmetic and then take the ones complement of the result.
- That 16-bit number is the checksum

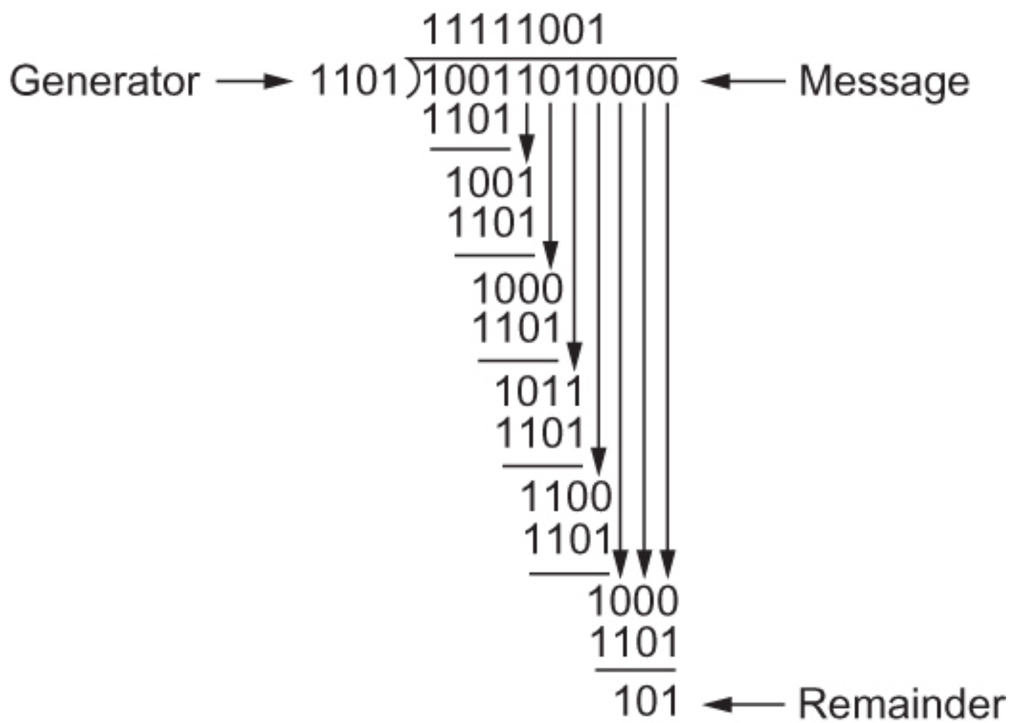
Cyclic Redundancy Check (CRC)

- Reduce the number of extra bits and maximize protection
- Given a bit string 110001 we can associate a polynomial on a single variable *x* for it.

  $1.x^5+1.x^4+0.x^3+0.x^2+0.x^1+1.x^0 = x^5+x^4+1$ and the degree is 5.

  A *k*-bit frame has a maximum degree of *k*-1
- Let M(x) be a message polynomial and C(x) be a generator polynomial.

Cyclic Redundancy Check (CRC)



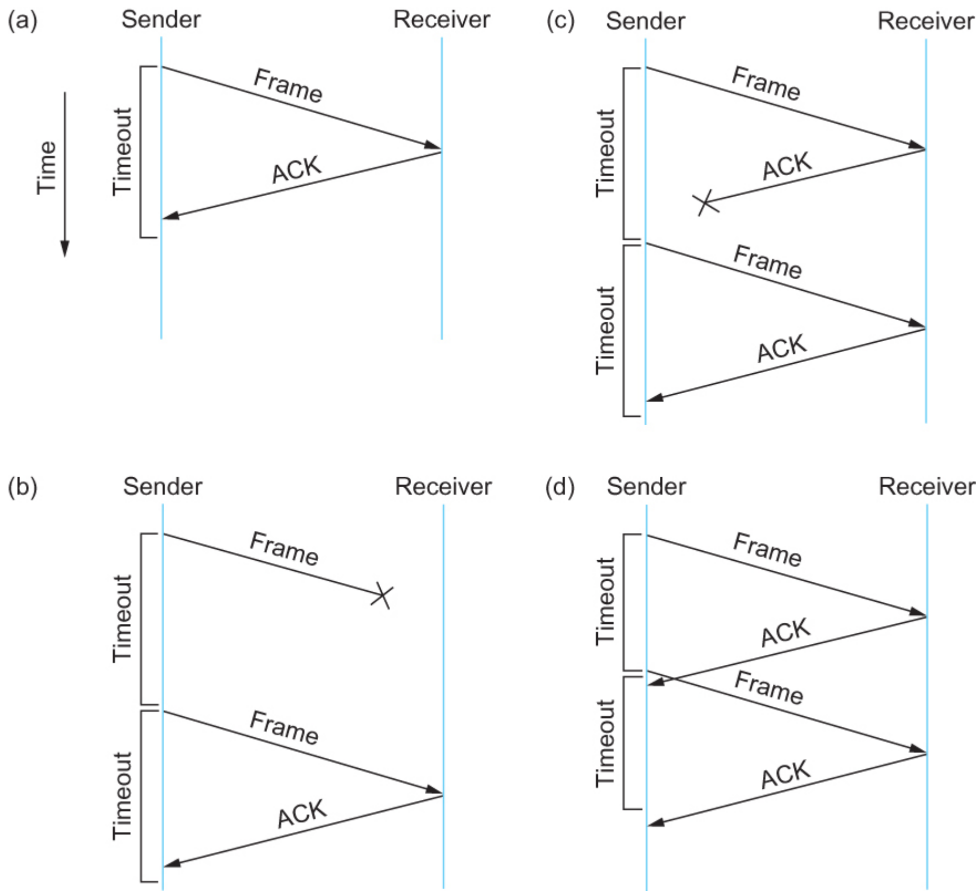
CRC Calculation using Polynomial Long Division

Cyclic Redundancy Check (CRC)

- Properties of Generator Polynomial
  - In general, it is possible to prove that the following types of errors can be detected by a C(x) with the stated properties
    - All single-bit errors, as long as the x^k and x^0 terms have nonzero coefficients.
    - All double-bit errors, as long as C(x) has a factor with at least three terms.
    - Any odd number of errors, as long as C(x) contains the factor (x+1).
    - Any "burst" error (i.e., sequence of consecutive error bits) for which the length of the burst is less than *k* bits. (Most burst errors of larger than *k* bits can also be detected.)

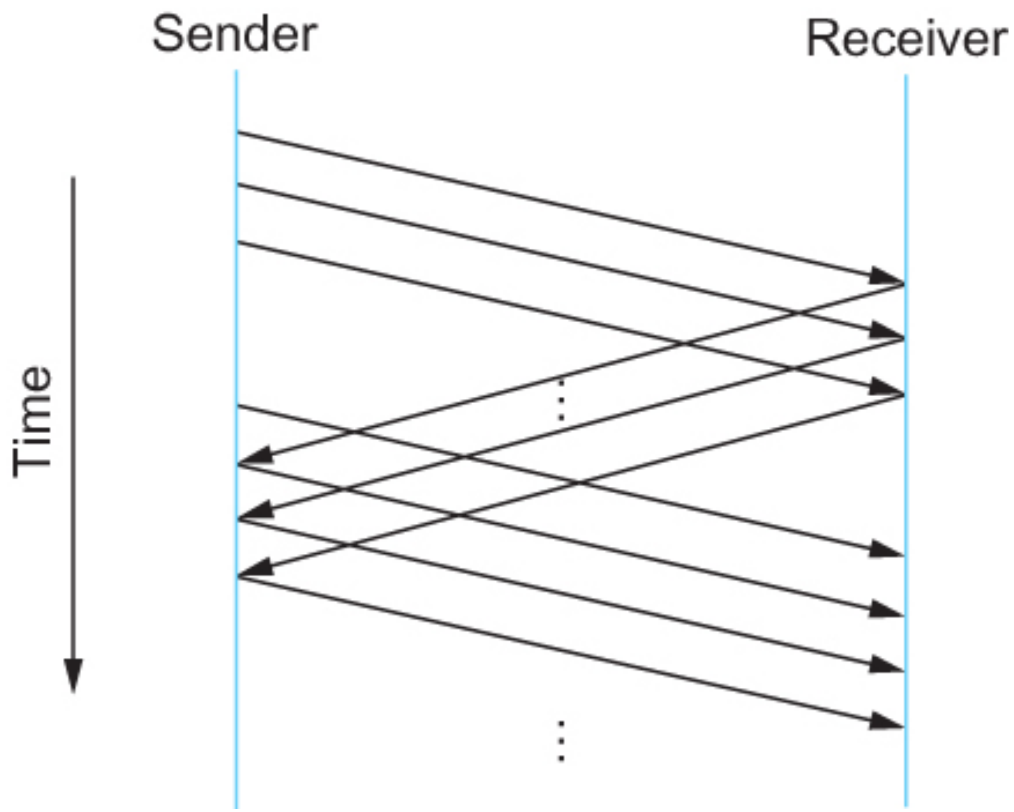
RELIABLE TRANSMISSION

Stop and Wait Protocol



Timeline showing four different scenarios for the stop-and-wait algorithm.
(a) The ACK is received before the timer expires; (b) the original frame is lost; (c) the ACK is lost; (d) the timeout fires too soon

Sliding Window Protocol



Timeline for Sliding Window Protocol

DATA LINK LAYER

Ethernet

- Any signal placed on the Ethernet by a host is broadcast over the entire network
  - Signal is propagated in both directions.
  - Repeaters forward the signal on all outgoing segments.
  - Terminators attached to the end of each segment absorb the signal.

- Ethernet uses Manchester encoding scheme.

Ethernet Addresses

- Each host on an Ethernet (in fact, every Ethernet host in the world) has a unique Ethernet Address.
- The address belongs to the adaptor, not the host.
  - It is usually burnt into ROM.
- Ethernet addresses are typically printed in a human readable format
  - As a sequence of six numbers separated by colons.
  - Each number corresponds to 1 byte of the 6 byte address and is given by a pair of hexadecimal digits, one for each of the 4-bit nibbles in the byte
  - Leading 0s are dropped.
  - For example, 8:0:2b:e4:b1:2 is
    - 00001000 00000000 00101011 11100100 10110001 00000010

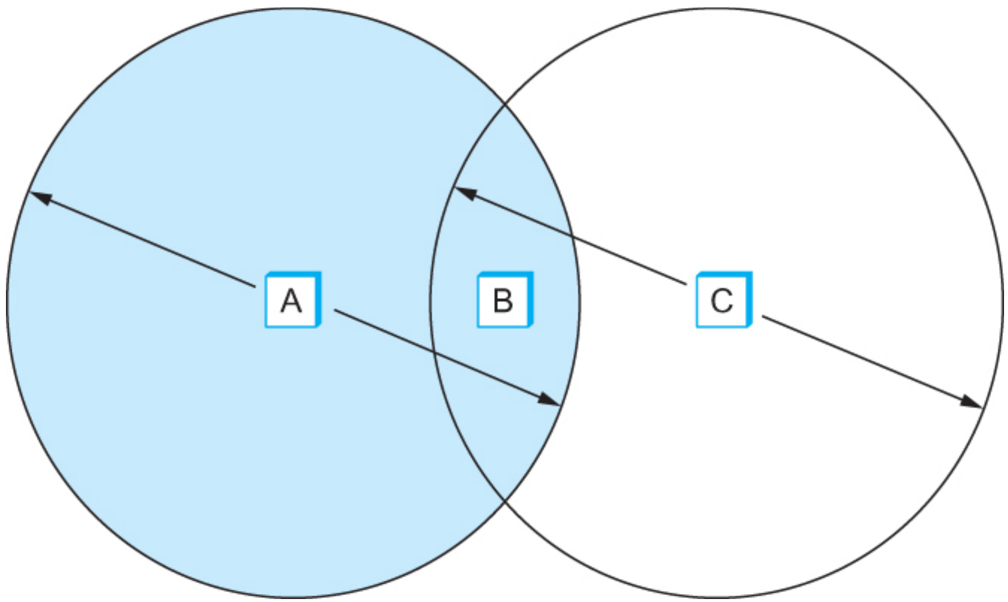
Ethernet Addresses

- To summarize, an Ethernet adaptor receives all frames and accepts
  - Frames addressed to its own address
  - Frames addressed to the broadcast address
  - Frames addressed to a multicast addressed if it has been instructed

IEEE 802.11

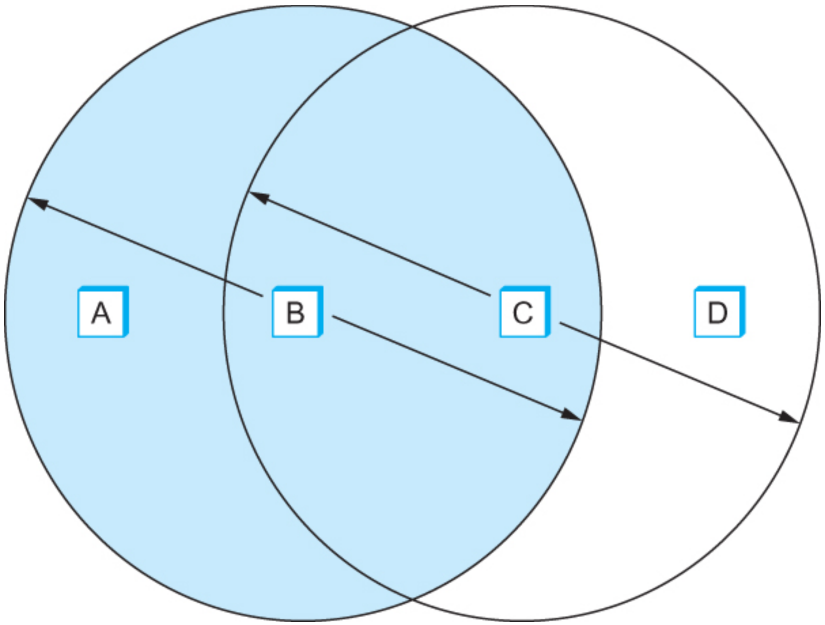
- Also known as Wi-Fi
- Like its Ethernet and token ring siblings, 802.11 is designed for use in a limited geographical area (homes, office buildings, campuses)
  - Primary challenge is to mediate access to a shared communication medium – in this case, signals propagating through space
- 802.11 supports additional features
  - power management and
  - security mechanisms

Hidden Terminal



The “Hidden Node” Problem. Although A and C are hidden from each other, their signals can collide at B. (B’s reach is not shown.)

Exposed Terminal



Exposed Node Problem. Although B and C are exposed to each other's signals, there is no interference if B transmits to A while C transmits to D. (A and D's reaches are not shown.)

IEEE 802.11 – Collision Avoidance

- 802.11 addresses these two problems with an algorithm called Multiple Access with Collision Avoidance (MACA).
- Key Idea
  - Sender and receiver exchange control frames with each other before the sender actually transmits any data.
  - This exchange informs all nearby nodes that a transmission is about to begin
  - Sender transmits a *Request to Send* (RTS) frame to the receiver.
    - The RTS frame includes a field that indicates how long the sender wants to hold the medium
      - Length of the data frame to be transmitted
  - Receiver replies with a *Clear to Send* (CTS) frame
    - This frame echoes this length field back to the sender

IEEE 802.11 – Collision Avoidance

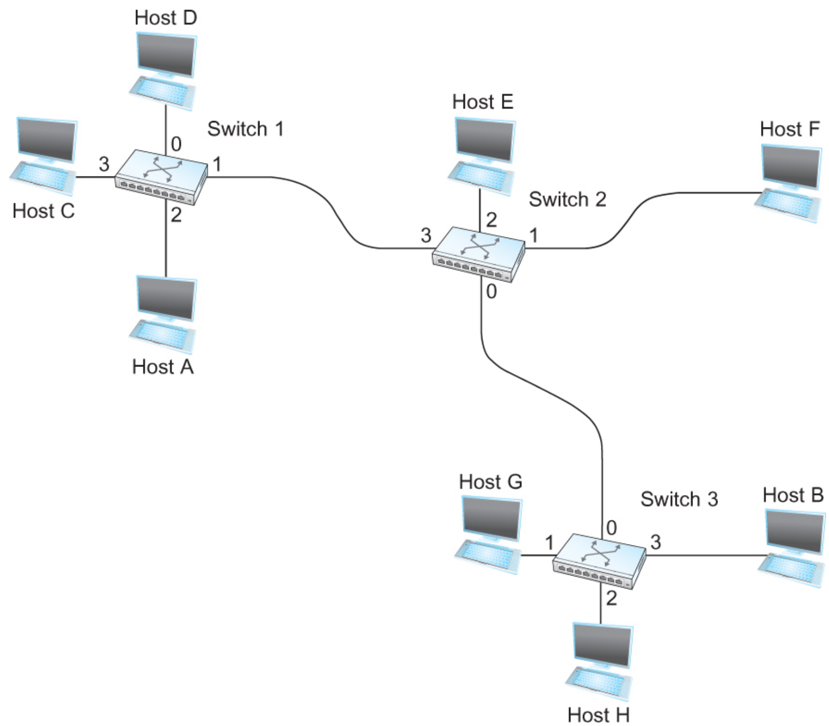
- Any node that sees the CTS frame knows that
  - it is close to the receiver, therefore
  - cannot transmit for the period of time it takes to send a frame of the specified length
- Any node that sees the RTS frame but not the CTS frame
  - is not close enough to the receiver to interfere with it, and
  - so is free to transmit

SWITCHING AND FORWARDING

Switching and Forwarding

- How does the switch decide which output port to place each packet on?
  - It looks at the header of the packet for an identifier that it uses to make the decision
  - Two common approaches
    - *Datagram or Connectionless approach*
    - *Virtual circuit or Connection-oriented approach*
  - A third approach *source routing* is less common

Connectionless (Datagram)



Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Forwarding Table for Switch 2

Connection-oriented (Virtual circuit)

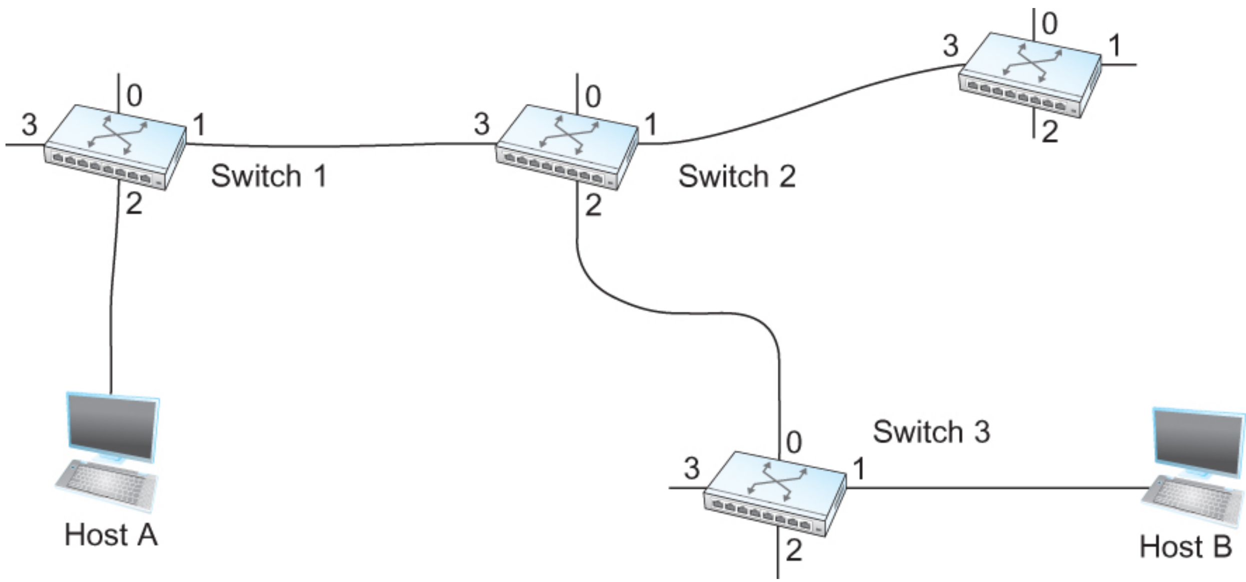
Two-stage process

- Connection setup
- Data Transfer

- Connection setup
  - Establish "connection state" in each of the switches between the source and destination hosts
  - The connection state for a single connection consists of an entry in the "VC table" in each switch through which the connection passes

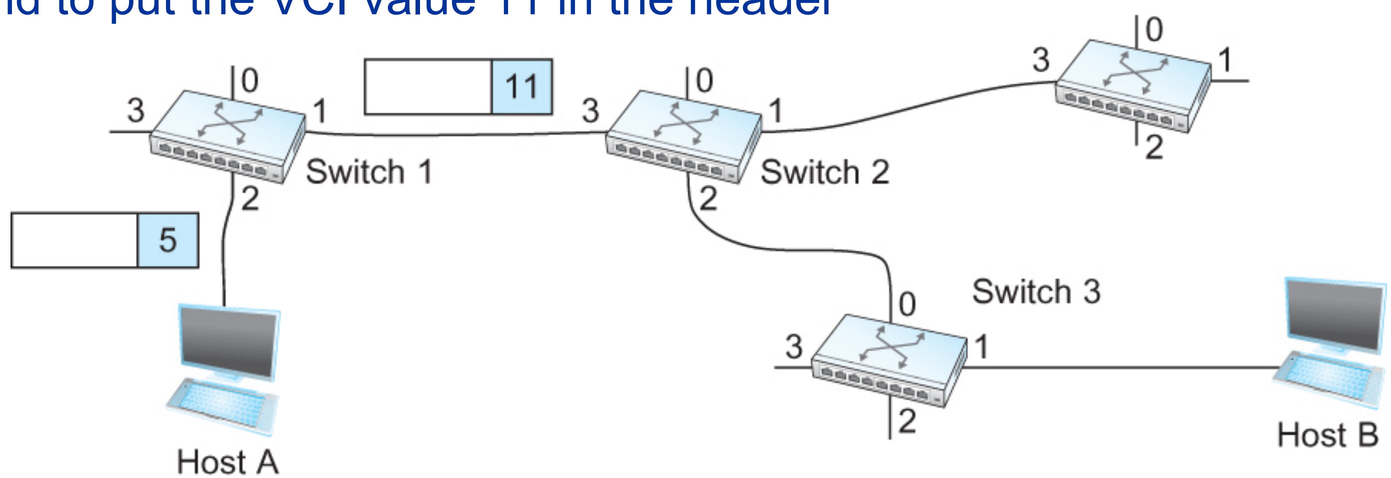
Connection-oriented (Virtual circuit)

- Host A wants to send packets to host B



Connection-oriented (Virtual circuit)

- For any packet that A wants to send to B, A puts the VCI value 5 in the header of the packet and sends it to switch 1
- Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry.
- The table entry on switch 1 tells the switch to forward the packet out of interface 1 and to put the VCI value 11 in the header



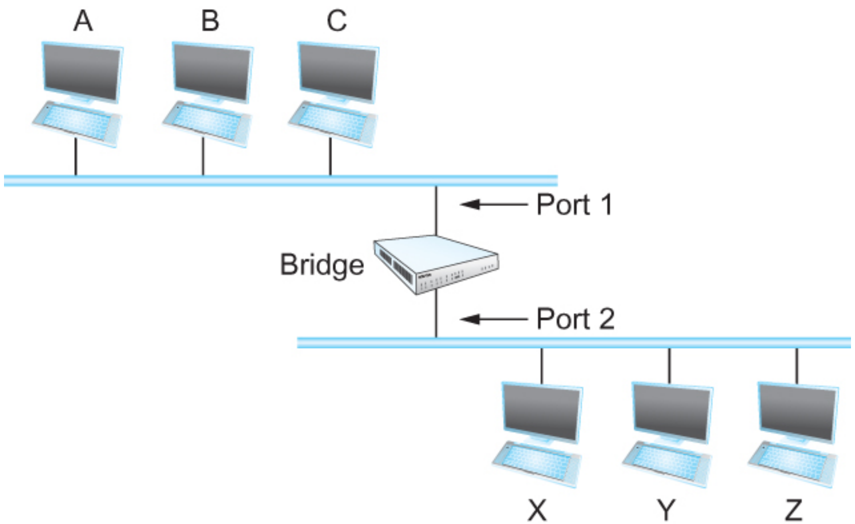
Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
2	5	1	11

Characteristics of VC

- There is at least one RTT of delay before data is sent
  - Host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet
- Data packet contains only a small identifier, which is only unique on one link.
  - The per-packet overhead caused by the header is reduced relative to the datagram model
- If a switch or a link in a connection fails, the connection is broken and a new one will need to be established.
  - Also the old one needs to be torn down to free up table storage space in the switches
- The issue of how a switch decides which link to forward the connection request on has similarities with the function of a routing algorithm

Bridges and LAN Switches

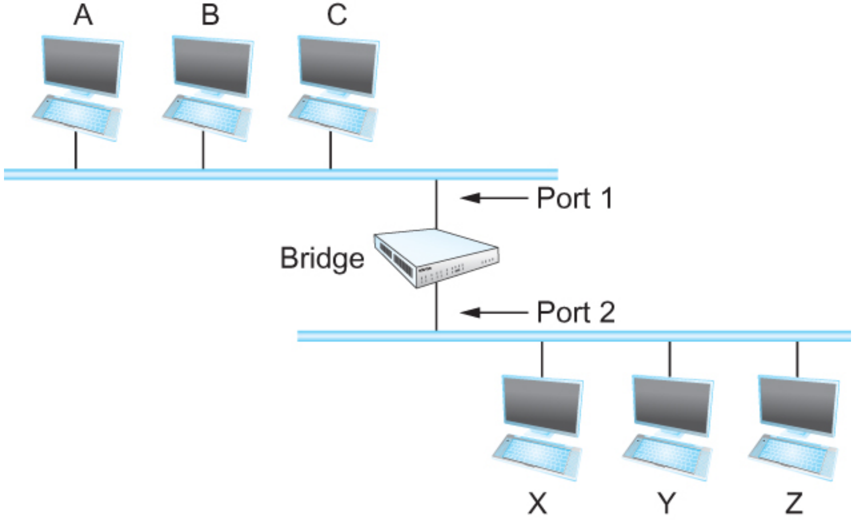
- Consider the following figure
  - When a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.



  - How does a bridge come to learn on which port the various hosts reside?

Bridges and LAN Switches

- Solution
  - Download a table into the bridge

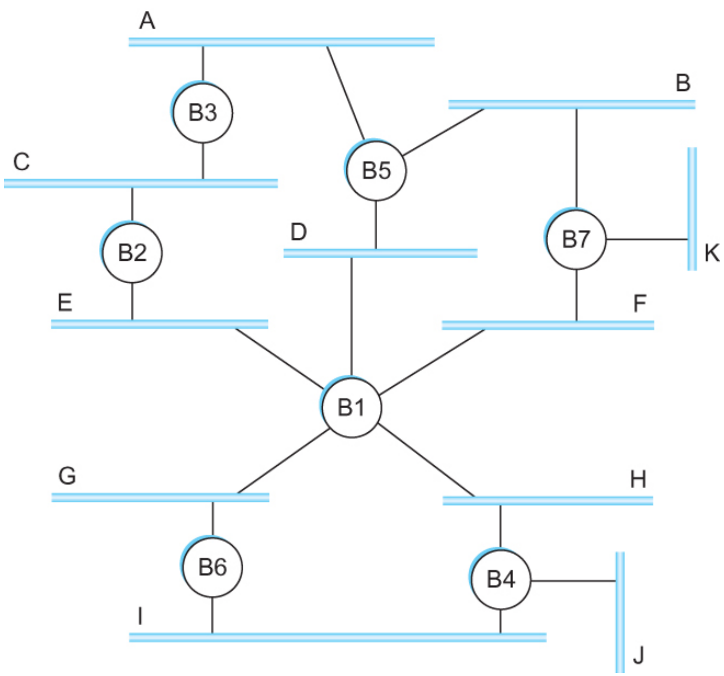


Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

  - Who does the download?
    - Human
      - Too much work for maintenance

Spanning Tree Algorithm

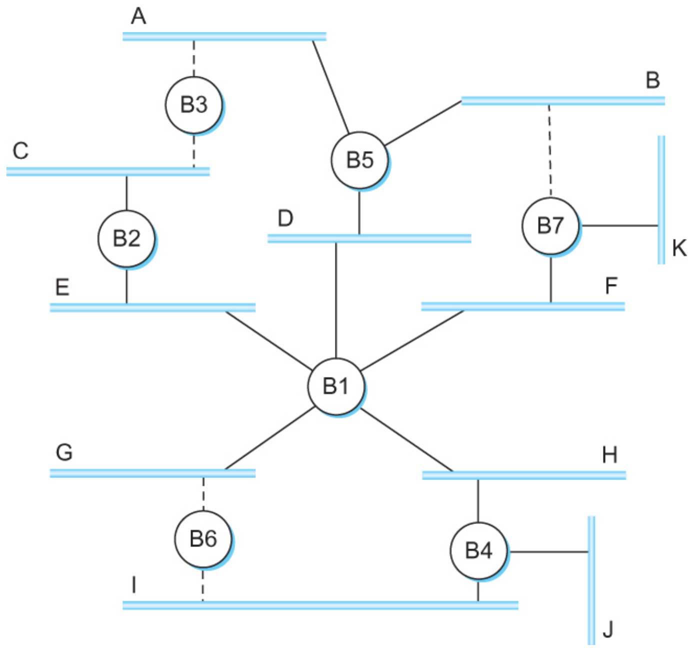
- Consider the situation when the power had just been restored to the building housing the following network



- All bridges would start off by claiming to be the root

Spanning Tree Algorithm

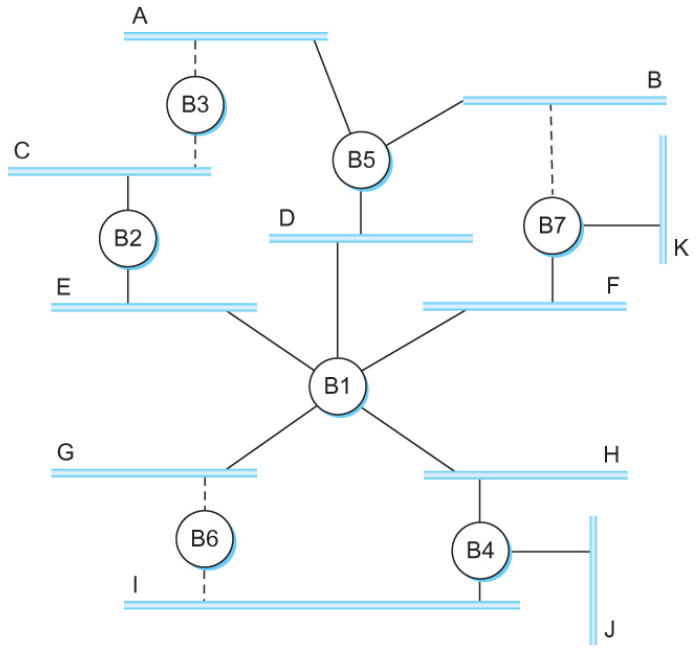
- Denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



- Consider the activity at node B3

Spanning Tree Algorithm

- B3 receives (B2, 0, B2)
- Since 2 < 3, B3 accepts B2 as root
- B3 adds 1 to the distance advertised by B2 and sends (B2, 1, B3) to B5
- Meanwhile B2 accepts B1 as root because it has the lower id and it sends (B1, 1, B2) toward B3
- B5 accepts B1 as root and sends (B1, 1, B5) to B3
- B3 accepts B1 as root and it notes that both B2 and B5 are closer to the root than it is.
  - Thus B3 stops forwarding messages on both its interfaces
  - This leaves B3 with both ports not selected



INTERNET PROTOCOL (IP)

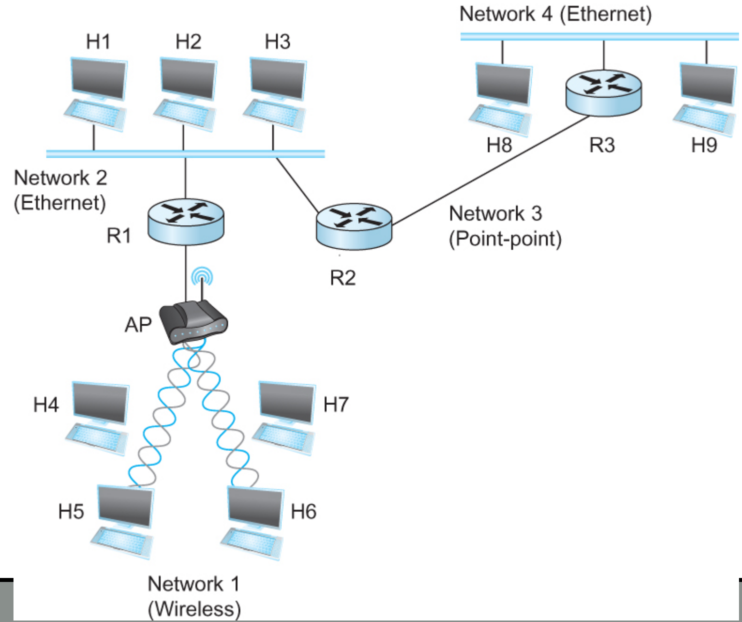
IP Service Model

- Packet Delivery Model
  - Connectionless model for data delivery
  - Best-effort delivery (unreliable service)
    - packets are lost
    - packets are delivered out of order
    - duplicate copies of a packet are delivered
    - packets can be delayed for a long time
- Global Addressing Scheme
  - Provides a way to identify all hosts in the network

IP Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table
- Example (router R2)

NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3



IP

- IP address classes?
  - Subnetting
  - Classless adressing
- Address Resolution Protocol (ARP)
- DHCP
- ICMP

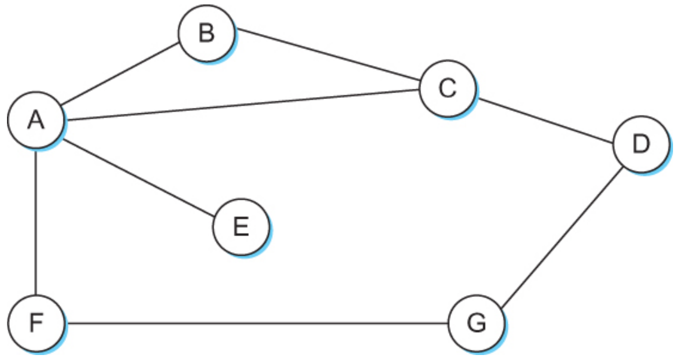
ROUTING

Routing

- For a simple network, we can calculate all shortest paths and load them into some nonvolatile storage on each node.
- Such a static approach has several shortcomings
  - It does not deal with node or link failures
  - It does not consider the addition of new nodes or links
  - It implies that edge costs cannot change

- What is the solution?
  - Need a distributed and dynamic protocol
  - Two main classes of protocols
    - Distance Vector
    - Link State

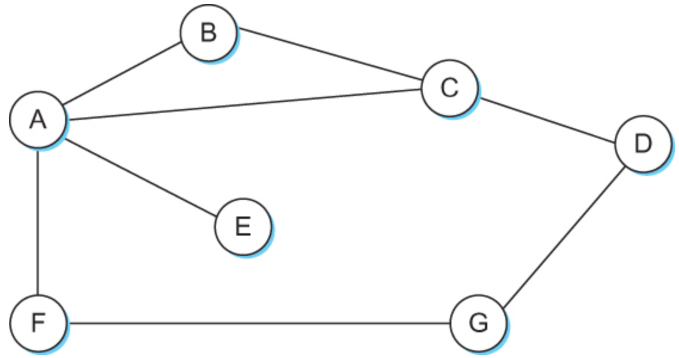
Distance Vector



Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Initial distances stored at each node (global view)

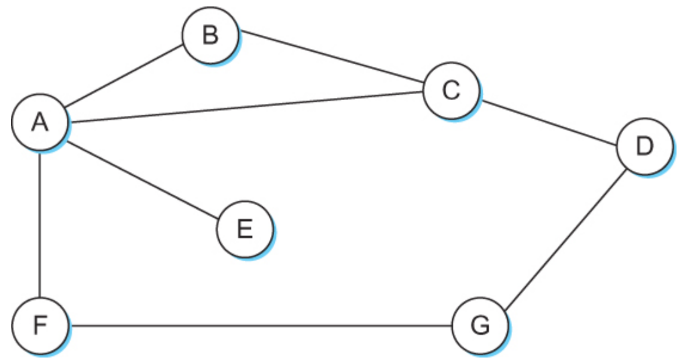
Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Initial routing table at node A

Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Final routing table at node A

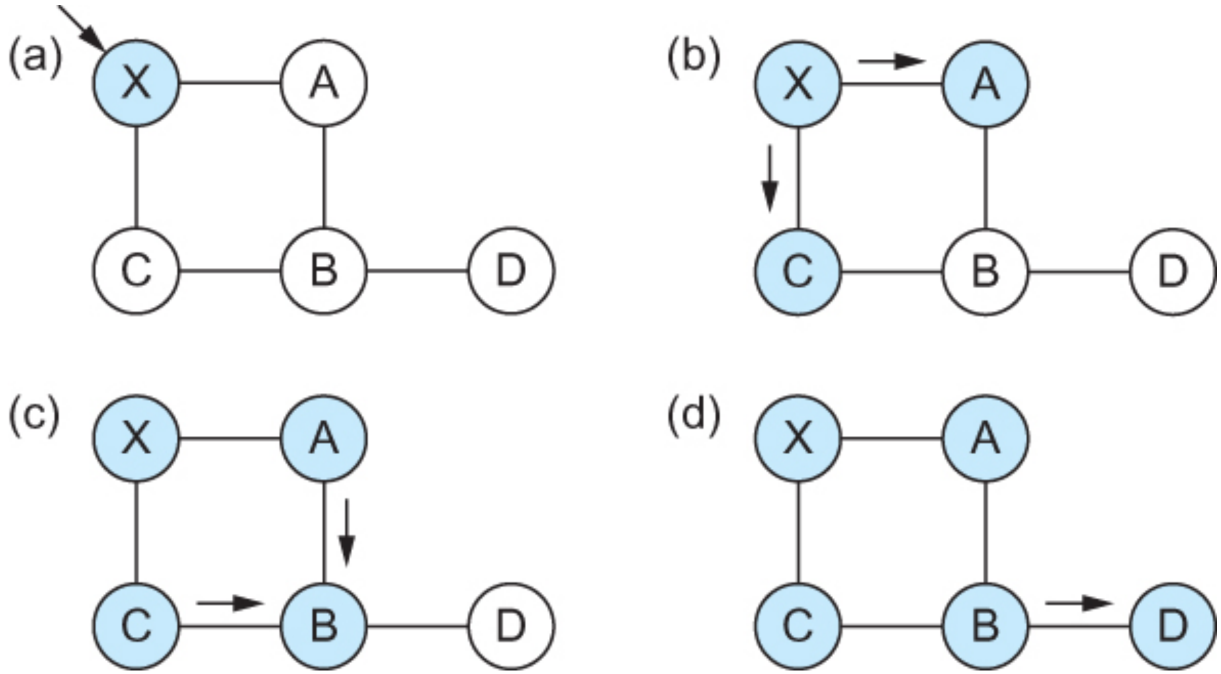
Link State Routing

Strategy: Send to all nodes (not just neighbors) information about directly connected links (not entire routing table).

- Link State Packet (LSP)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet
- Reliable Flooding
  - store most recent LSP from each node
  - forward LSP to all nodes but one that sent it
  - generate new LSP periodically; increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP; discard when TTL=0

Link State

Reliable Flooding

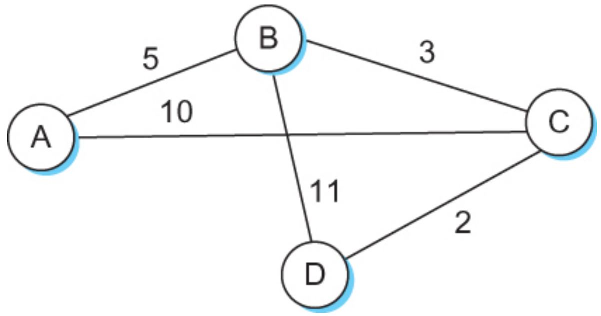


Flooding of link-state packets. (a) LSP arrives at node X; (b) X floods LSP to A and C; (c) A and C flood LSP to B (but not X); (d) flooding is complete

Shortest Path Routing

- The algorithm (based on Dijkstra's Algorithm)
  - Initialize the **Confirmed** list with an entry for myself; this entry has a cost of 0
  - For the node just added to the **Confirmed** list in the previous step, call it node **Next**, select its LSP
  - For each neighbor (Neighbor) of **Next**, calculate the cost (Cost) to reach this Neighbor as the sum of the cost from myself to Next and from Next to Neighbor
    - If Neighbor is currently on neither the **Confirmed** nor the **Tentative** list, then add (Neighbor, Cost, Nexthop) to the **Tentative** list, where Nexthop is the direction I go to reach Next
    - If Neighbor is currently on the **Tentative** list, and the Cost is less than the currently listed cost for the Neighbor, then replace the current entry with (Neighbor, Cost, Nexthop) where Nexthop is the direction I go to reach Next
  - If the **Tentative** list is empty, stop. Otherwise, pick the entry from the **Tentative** list with the lowest cost, move it to the **Confirmed** list, and return to Step 2.

Shortest Path Routing



Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

BACKUP

What are Protocols?

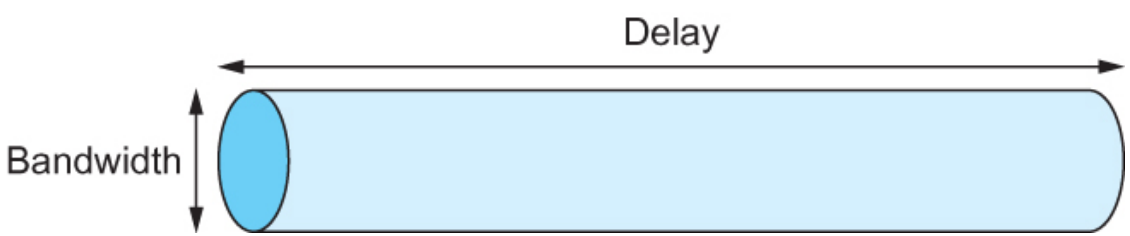
- Protocol defines the interfaces between the layers in the same system and with the layers of peer system
- Building blocks of a network architecture
- Each protocol object has two different interfaces
  - service interface: operations on this protocol
  - peer-to-peer interface: messages exchanged with peer
- Term "protocol" is overloaded
  - specification of peer-to-peer interface
  - module that implements this interface

Delay X Bandwidth

- We think the channel between a pair of processes as a hollow pipe
- Latency (delay) length of the pipe and bandwidth the width of the pipe
- Delay of 50 ms and bandwidth of 45 Mbps

⇒ 50×10^{-3} seconds x 45×10^{6} bits/second

⇒ 2.25×10^{6} bits = 280 KB data.



Network as a pipe

Encoding

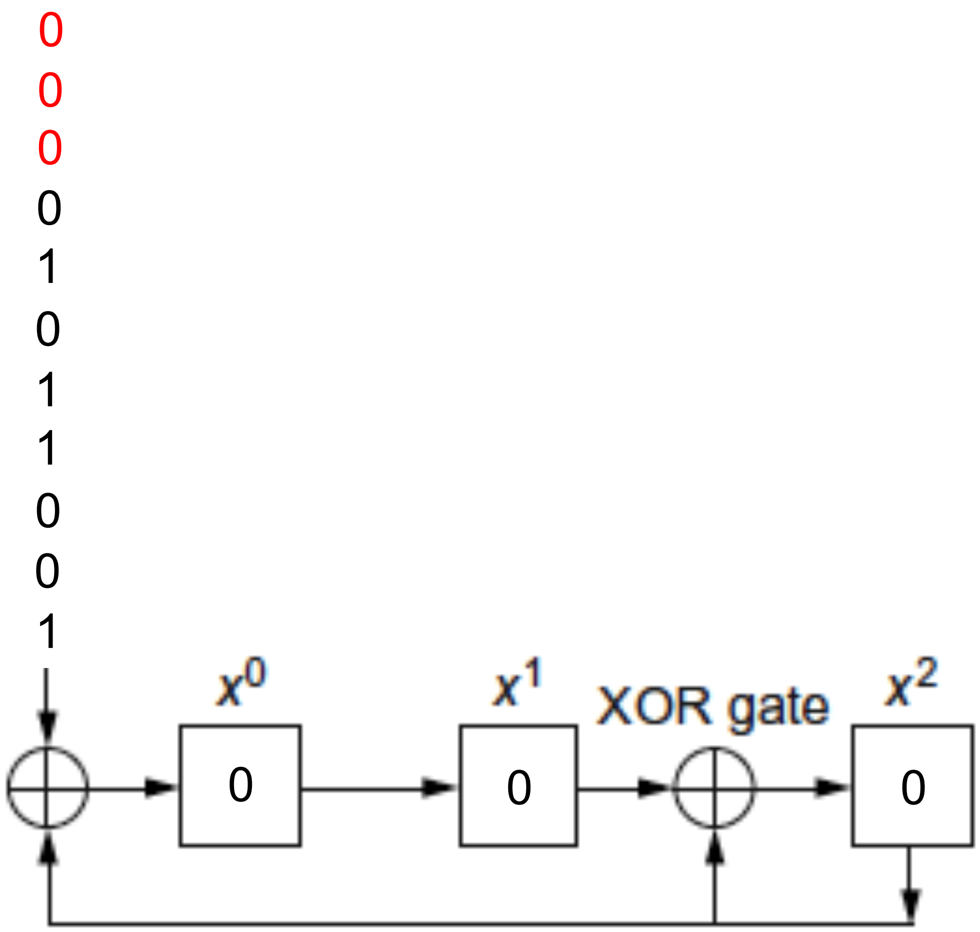
- 4B/5B encoding
  - Insert extra bits into bit stream so as to break up the long sequence of 0's and 1's
  - Every 4-bits of actual data are encoded in a 5- bit code that is transmitted to the receiver
  - 5-bit codes are selected in such a way that each one has no more than one leading 0(zero) and no more than two trailing 0's.
  - No pair of 5-bit codes results in more than three consecutive 0's

Encoding

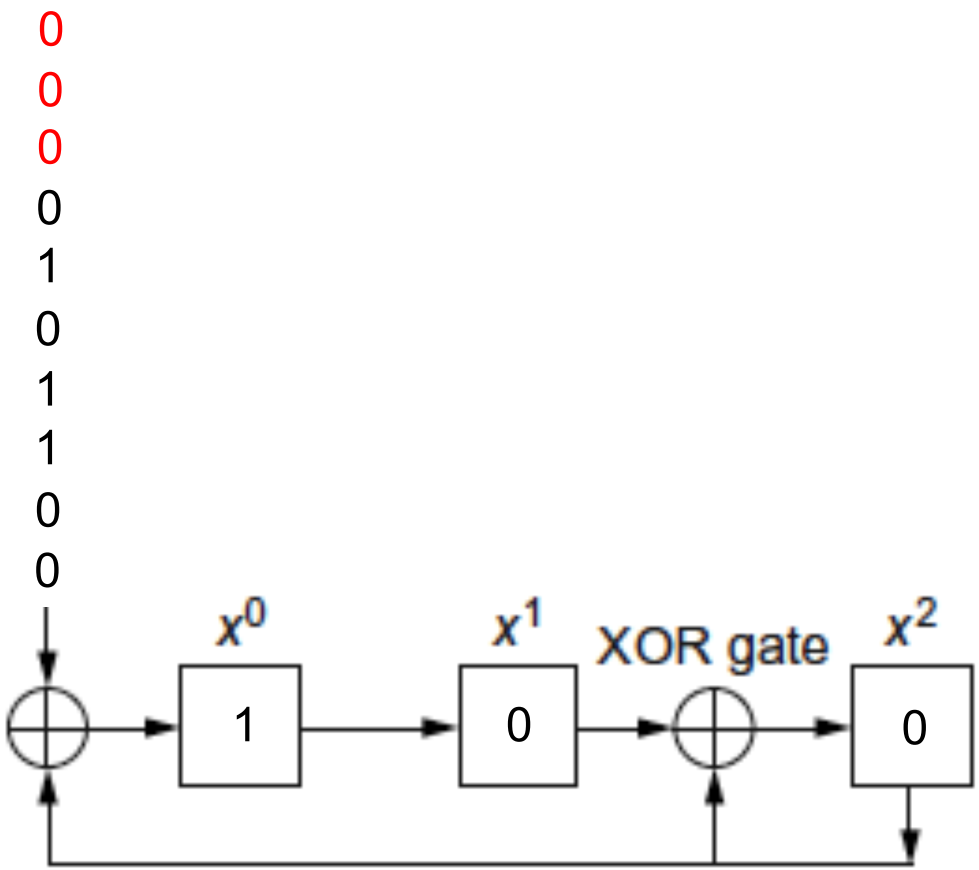
- 4B/5B encoding

0000 → 11110	16 left
0001 → 01001	11111 – when the line is idle
0010 → 10100	00000 – when the line is dead
..	00100 – to mean halt
..	
1111 → 11101	13 left : 7 invalid, 6 for various control signals

Cyclic Redundancy Check (CRC)

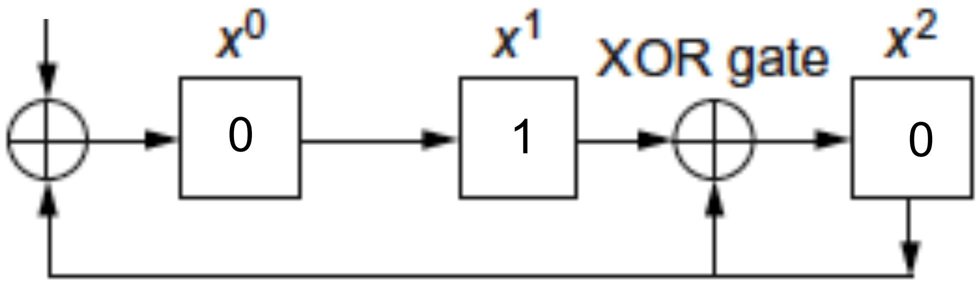


Cyclic Redundancy Check (CRC)

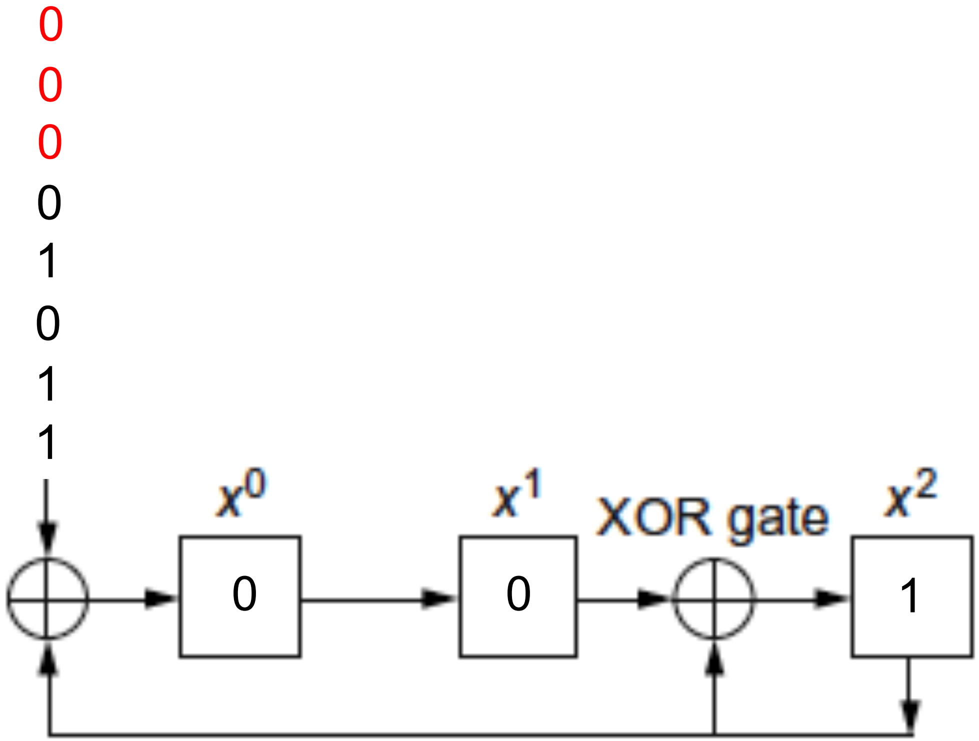


Cyclic Redundancy Check (CRC)

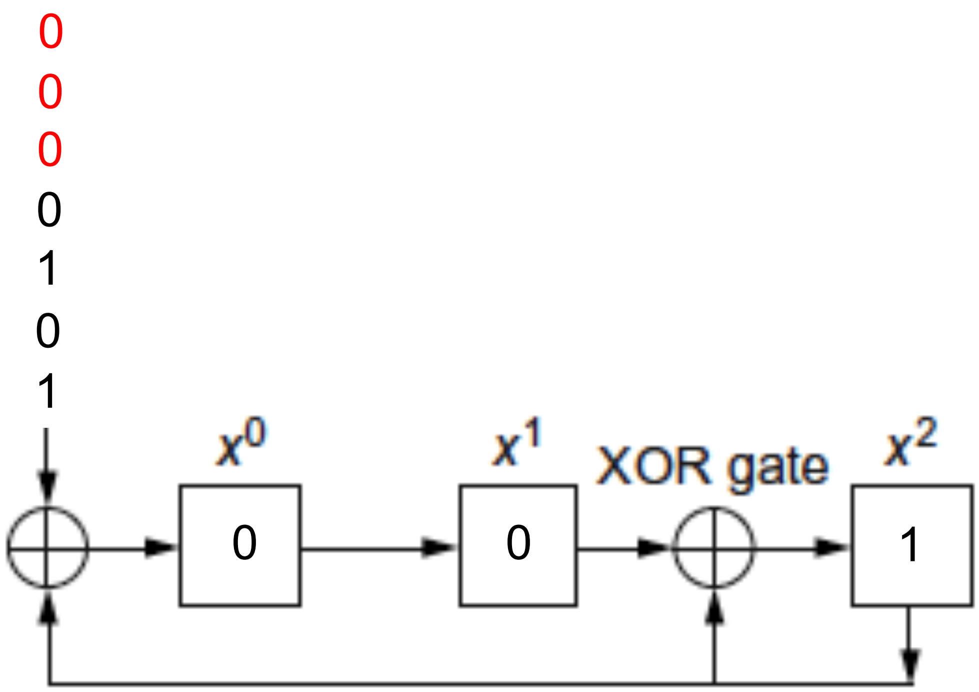
0
0
0
0
1
0
1
1
0



Cyclic Redundancy Check (CRC)

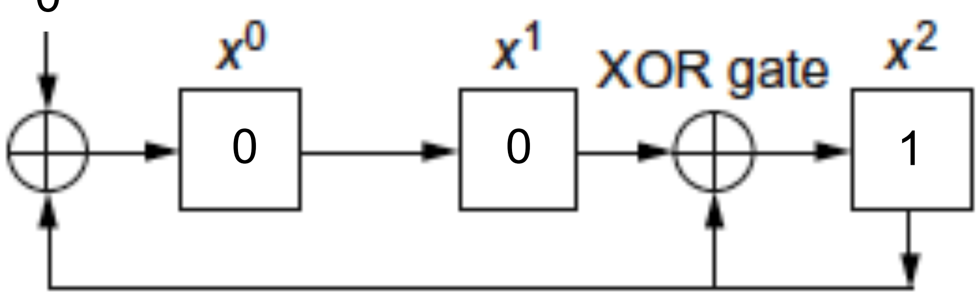


Cyclic Redundancy Check (CRC)

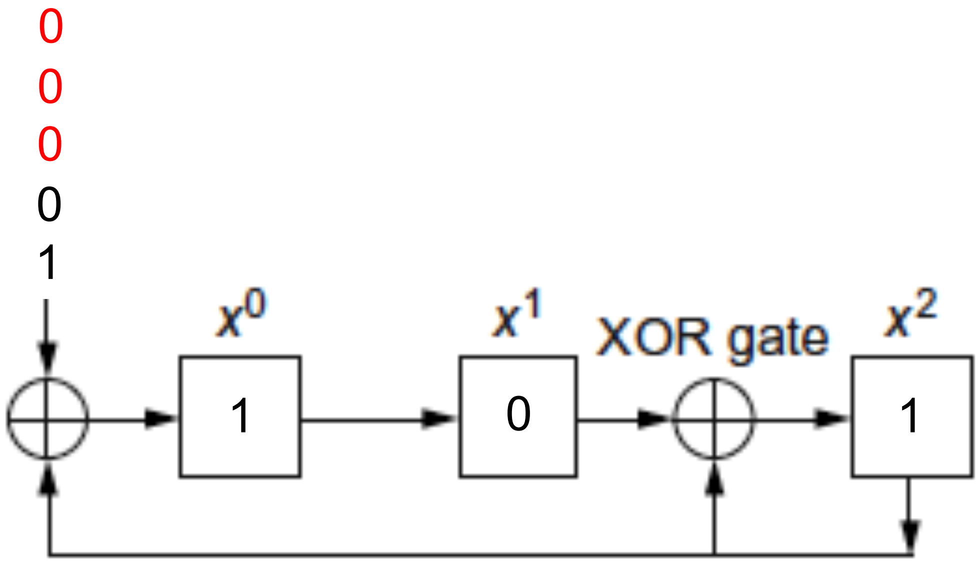


Cyclic Redundancy Check (CRC)

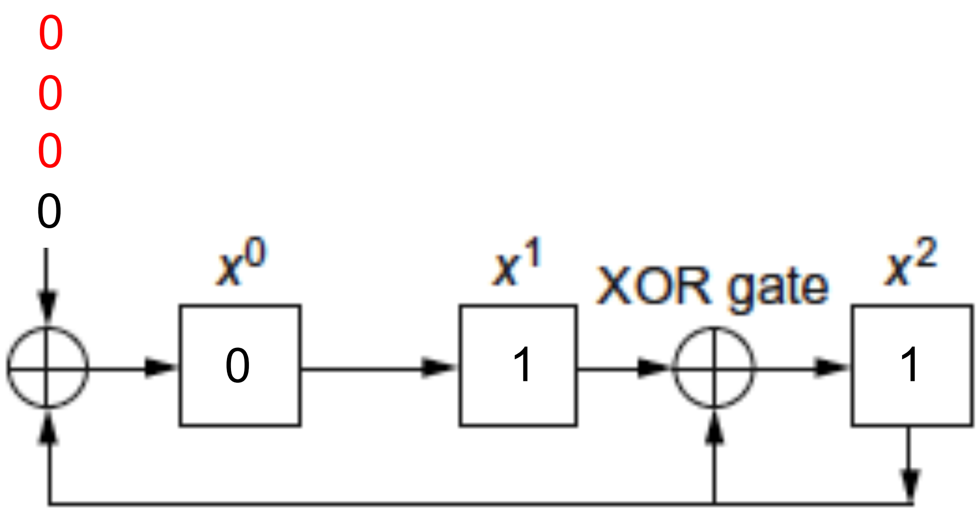
0
0
0
0
1
0



Cyclic Redundancy Check (CRC)

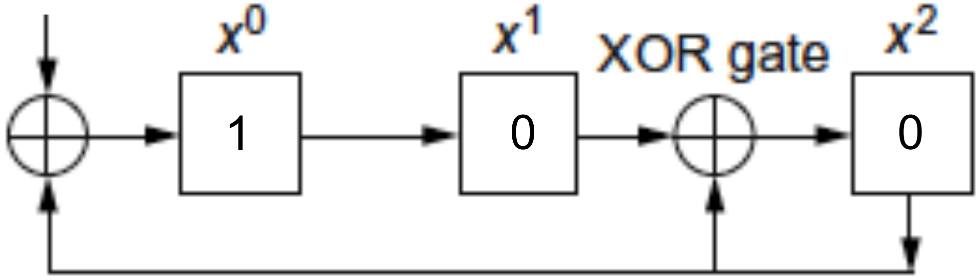


Cyclic Redundancy Check (CRC)

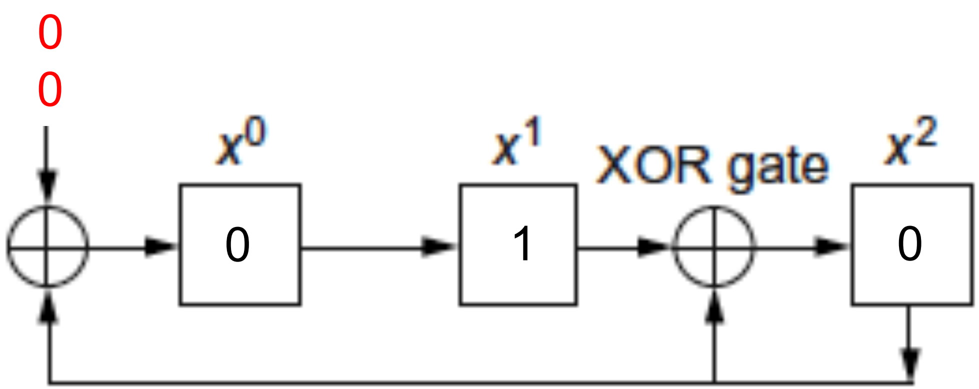


Cyclic Redundancy Check (CRC)

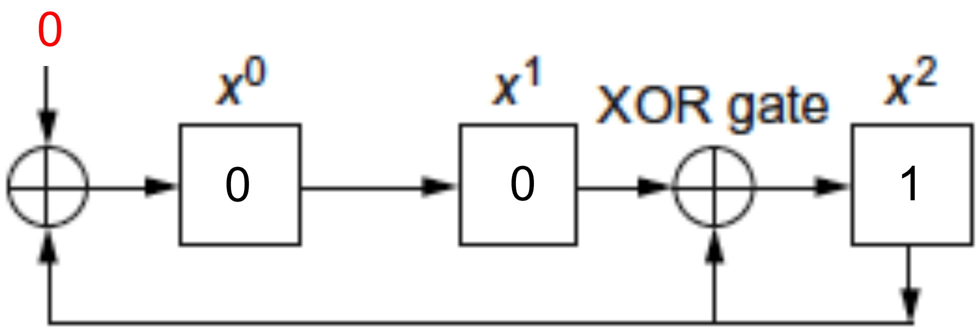
0
0
0



Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)