Data Communication and Computer Networks

9. Link Layer
PART-B

Dr. Aiman Hanna

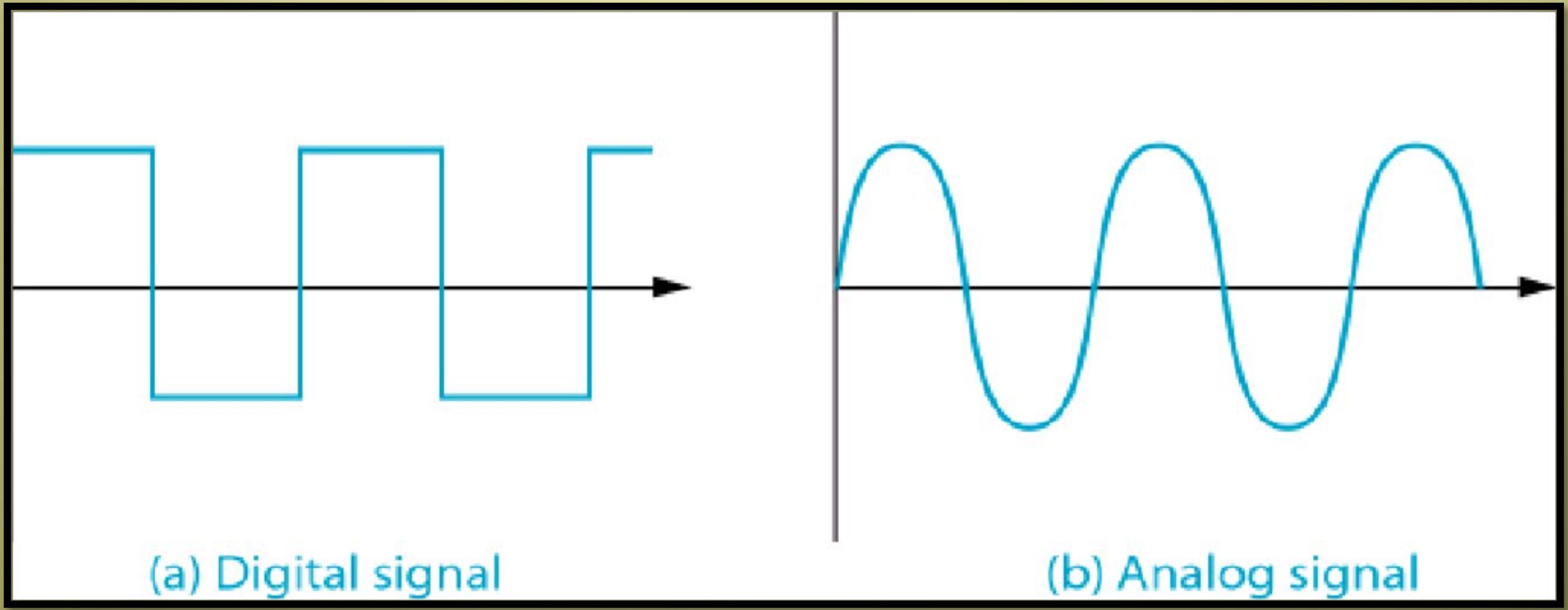
Department of Computer Science & Software Engineering
Concordia University, Montreal, Canada

These slides have mainly been extracted, modified and updated from original slides of :
Computer Networking: A Top Down Approach, 6th edition Jim Kurose, Keith Ross
Addison-Wesley, 2013

Additional materials have been extracted, modified and updated from:
Understanding Communications and Networking, 3e by William A. Shay 2005

Recall: Analog & Digital Signals

- first, let us recall:
  - analog and digital signals
  - digital signal encodings



Digital & Analog Signals

Digital Encoding Schemes

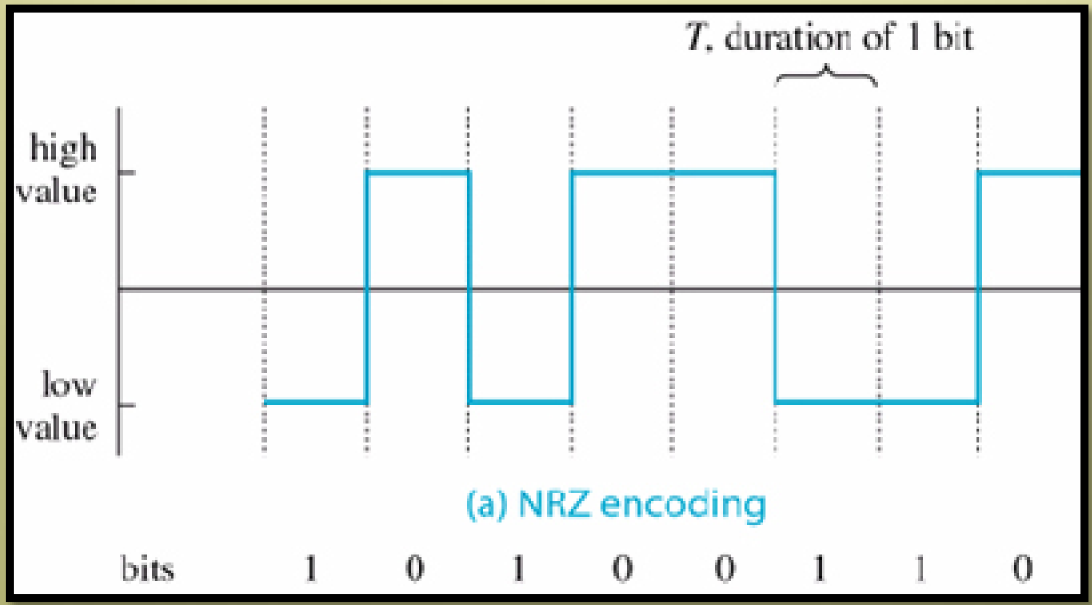
There are 10 types of people;
those who know binary and those who do not.

- digital data are represented by a sequence of 1s & 0s
- 1 refer to a high electrical voltage, and 0 refers to a low electrical voltage
- two major digital encoding schemes exist:
  - **Non-Return to Zero *(NRZ)* Encoding**
  - **Manchester Encoding**

Digital Encoding Schemes (continue...)

NRZ Encoding

- a 0 voltage is transmitted by raising the voltage level high, while 1 is transmitted by using a low voltage

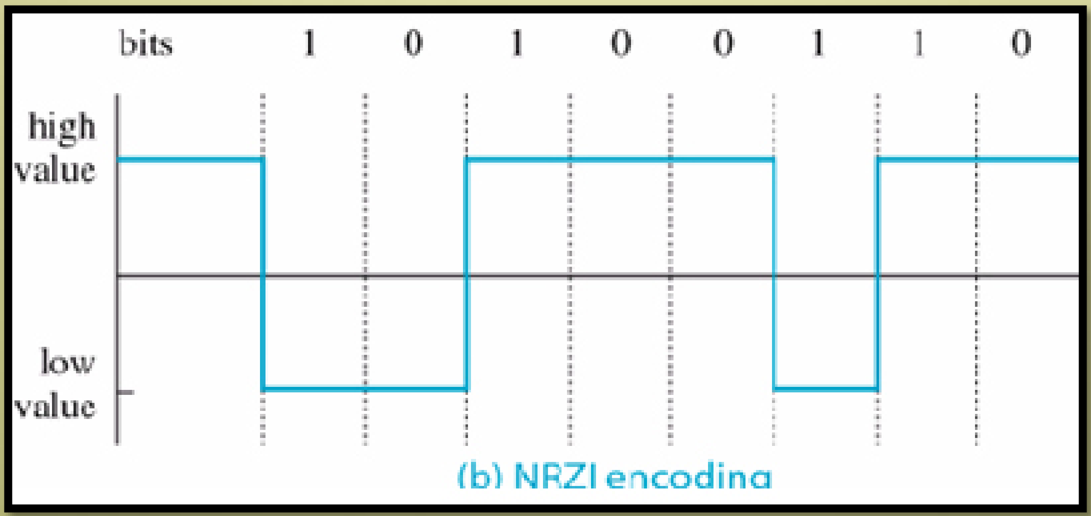


NRZ Encoding

Digital Encoding Schemes (continue...)

***NRZI* Encoding**

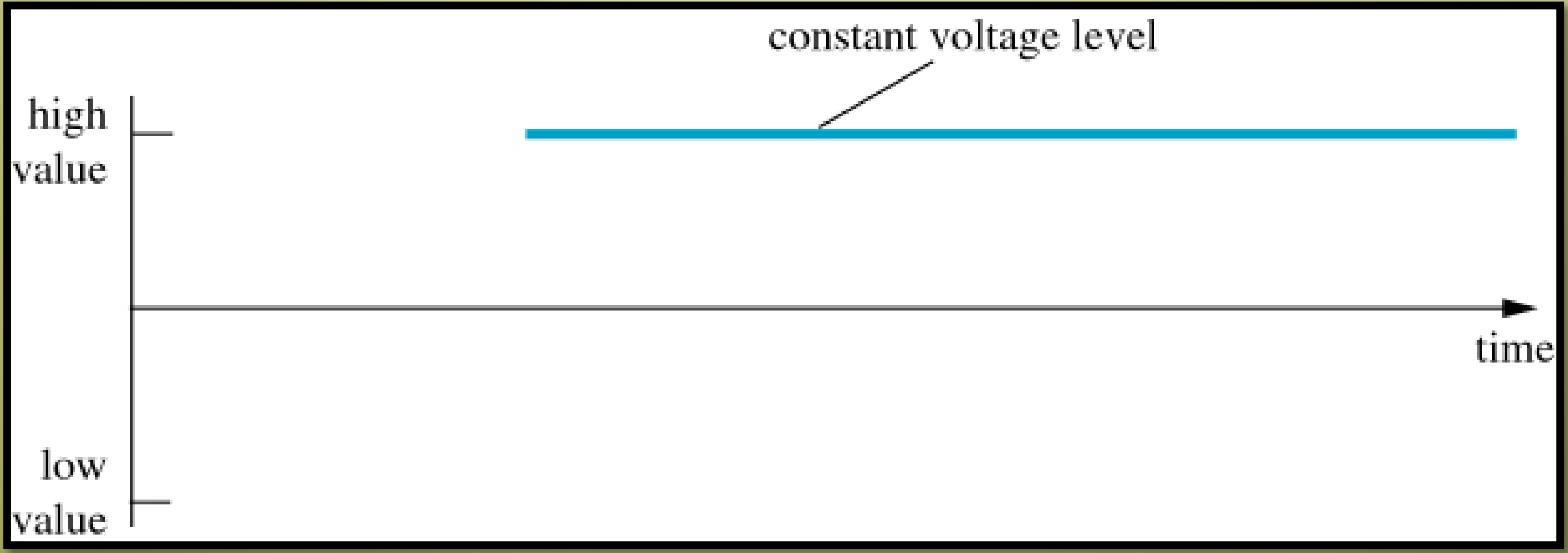
- An alternative to NRZ is NRZI (Inverted)
- The voltage changes only when a 1 is to be sent



NRZI Encoding

Digital Encoding Schemes (continue...)

- Both NRZ and NRZI have problems; for example what is the exact sequence being transmitted in the sequence below?
- Is time synchronization possible?

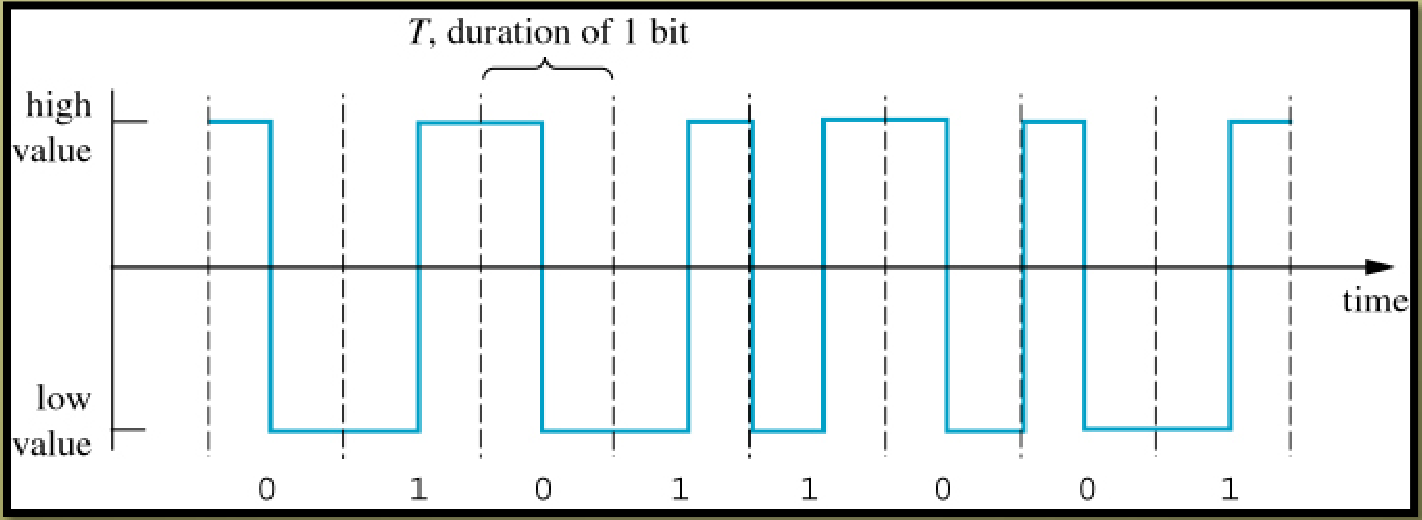


NRZ Encoding of a Sequence of 0s

Digital Encoding Schemes (continue...)

Manchester Encoding

- also called ***Self-Synchronizing Code***
- uses signal changes to keep the sending and receiving devices synchronized
- 0 is represented by a change from high to low in the middle of transmission and 1 is represented by a low to high change in the middle of transmission
- are there any disadvantages?

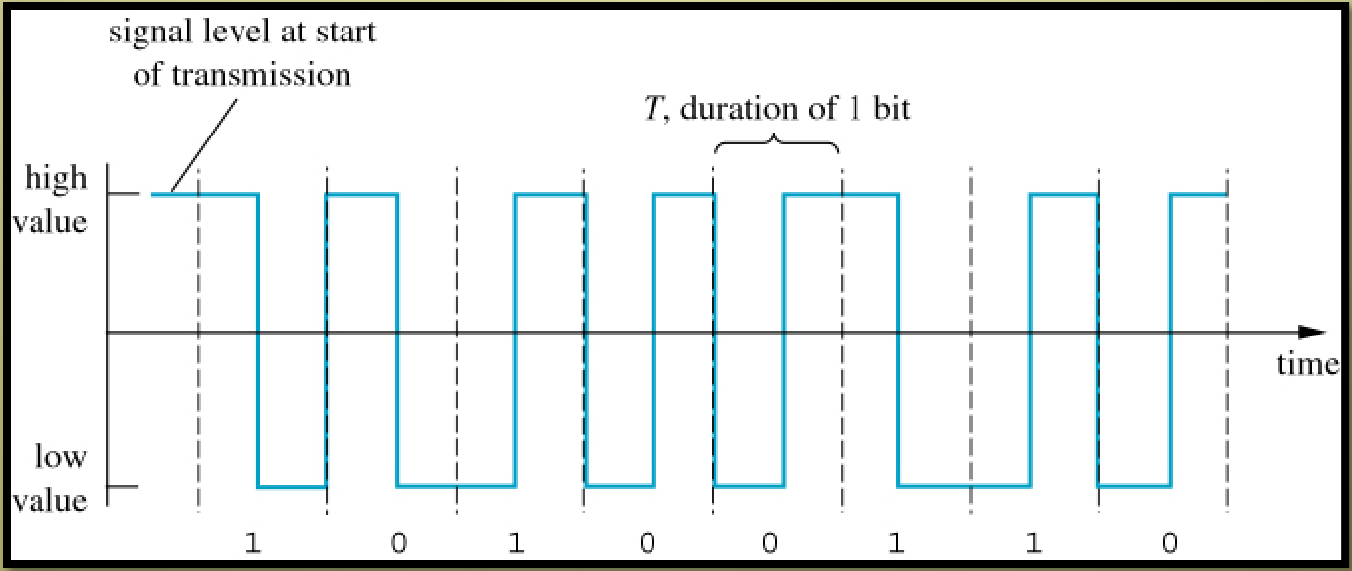


Manchester Encoding

Digital Encoding Schemes (continue...)

Differential Manchester Encoding

- similar to Manchester encoding, the signal will change in the middle, however
- 1 causes the signal to remain the same, while 0 causes the signal to change

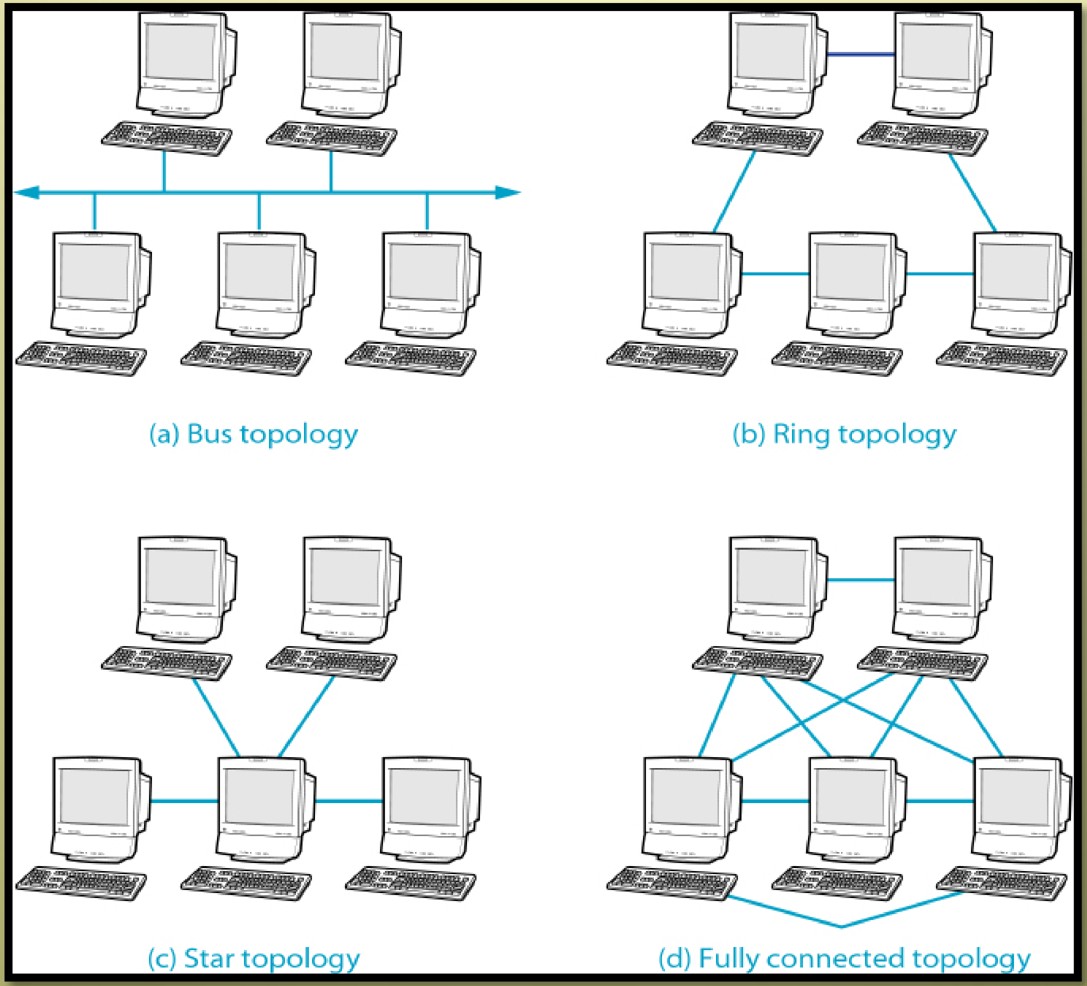


Differential Manchester Encoding

Local Area Networks (LANs)

- Local Area Network (LAN) covers limited geographic area, e.g. 1 or 2 buildings
- in contrast, a Wide Area Network (WAN) covers large area from cities, states, countries to the entire world
- LAN protocols & cabling are different than those of WAN
- Stations, or nodes, are typically PCs, printers, file servers,..
- LAN topologies are:
  - bus,
  - ring,
  - star,
  - fully connected

Local Area Networks (LANs)



LAN Topologies

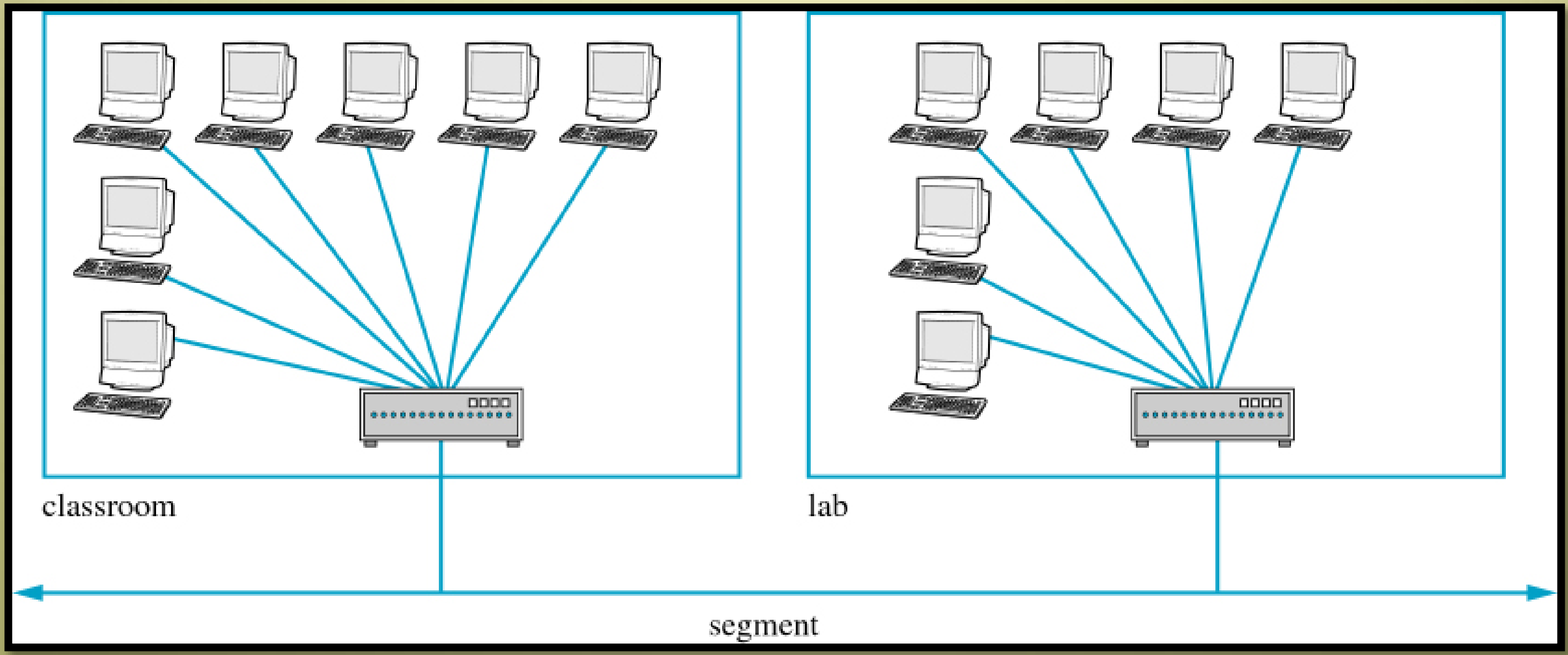
Local Area Networks (LANs)

Bus Topology

- the medium, referred to as a ***segment***, is a single communication line, typically a coaxial cable or optical fiber
- devices uses a contention protocol to send over the segment
- only one device can send at a time for collision not to occur

Local Area Networks (LANs)

Bus Topology



Bus Topology Connecting Multiple Locations

Local Area Networks (LANs)

Ring Topology

- devices are arranged into a ring, where each device is connected directly to its two neighbors
- for two devices to communicate, frames must be passed through all the devices in between
- a ring can be unidirectional or bidirectional

Local Area Networks (LANs)

Star Topology

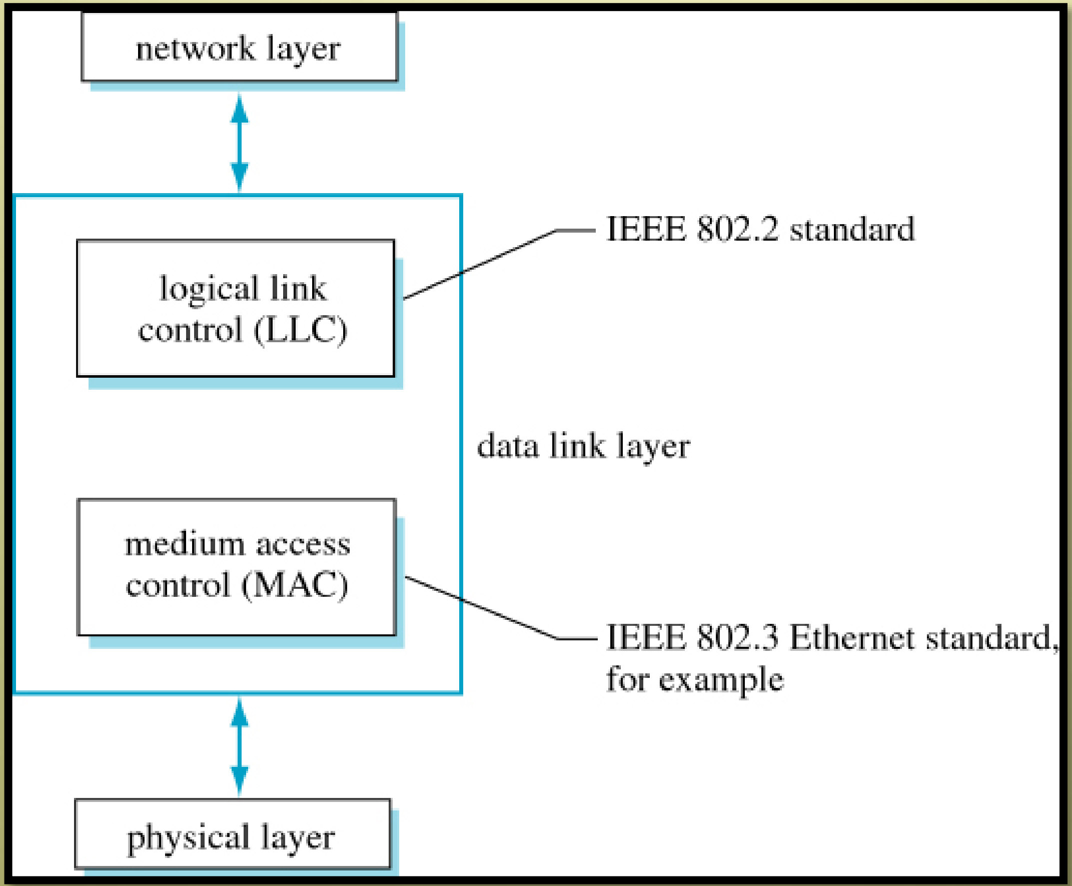
- a central devices is connected to all other devices
- communication must go through that central device

Fully Connected Topology

- a direct connection is there between any two devices
- this topology is rarely used as the general topology of a network; however it may still be utilized for a part of a network (i.e. as in data centers), which will be discussed shortly

Data Link Control

- Where LAN standers fit in a layered protocol?



Data Link Refinement

Data Link Control

- IEEE 802.3 Ethernet & IEEE 802.5 Token Ring standards are MAC protocols
- many different data link protocols can sit above the MAC
- all of these protocols however have a common ancestor, the **Synchronous Data Link Control** (**SDLC**) protocol
- SDLC was designed by IBM in the early 1970s
- SDLC was designed as a bit-oriented protocol
- prior to SDLC, protocols were byte oriented; that is frames are interpreted as sequence of bytes

Data Link Control

- IBM submitted SDLC to ISO for approval, however
  → ISO created their own standard out of it, which is called **High-level Data Link Control** (**HDLC**)

- IBM also submitted SDLC to ANSI for acceptance, however
  → ANSI modified and renamed it **Advanced Data Communication Control Procedure** (**ADCCP**)

- ITU adopted and modified SDLC to **Link Access Protocol** (**LAP**) and later to **LAPB**; B for Balanced

- IEEE has then created the **Logical Link Control** (**LLC**) protocol, for LANs, out of HDLC

- LLC also allows LANs to connect to other LANs and WANs

Mac Layer

- DHLC and similar protocols describes how devices can exchange frames, independent of the medium

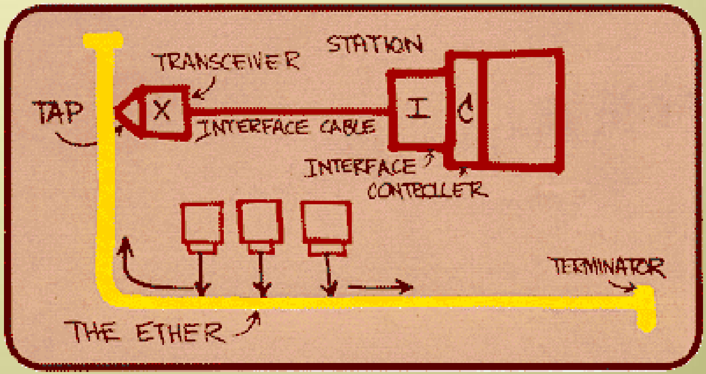
- MAC layer techniques consider the physical medium

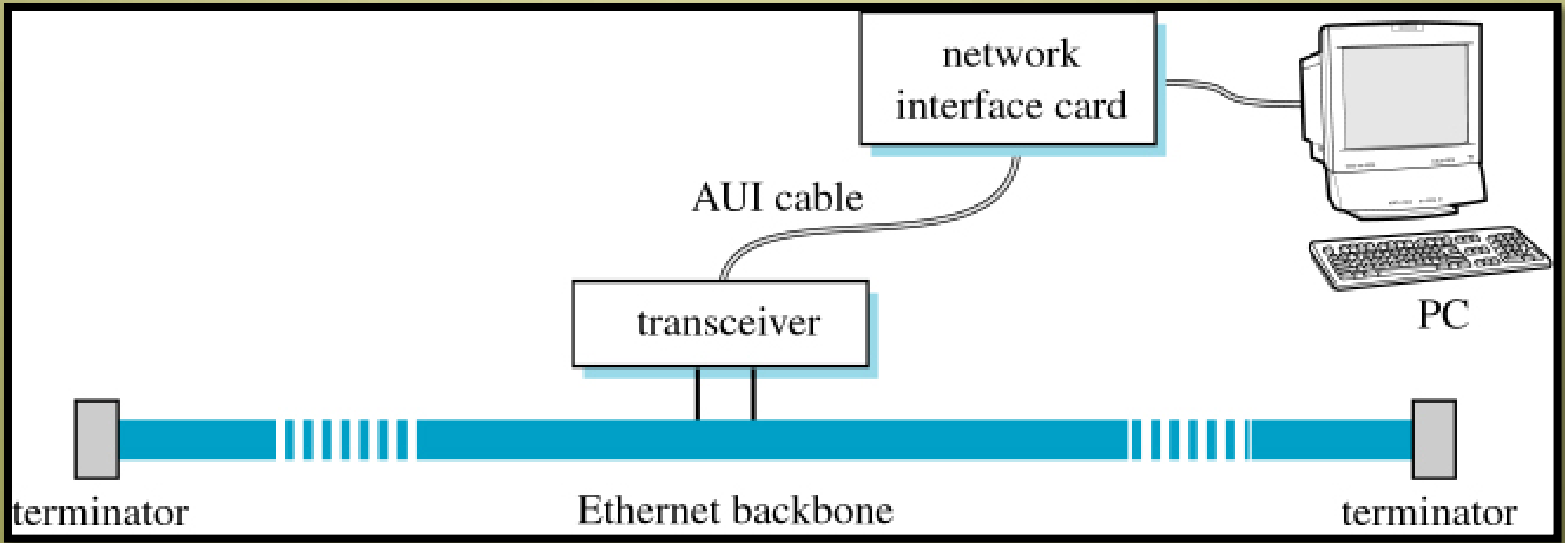
- three formal standards were defined by IEEE:
  - IEEE standard 802.3 – **The Ethernet**
    - proposed by Xerox, Intel & DEC
  - IEEE standard 802.4 – **Token Bus**
    - proposed by General Motors
  - IEEE standard 802.5 – **Token Ring**
    - proposed by IBM

Ethernet

- designed as a bus topology with some form of CSMA/CD contention protocol
- today, it is considered as the dominant LAN standard

Metcalfe's Ethernet sketch



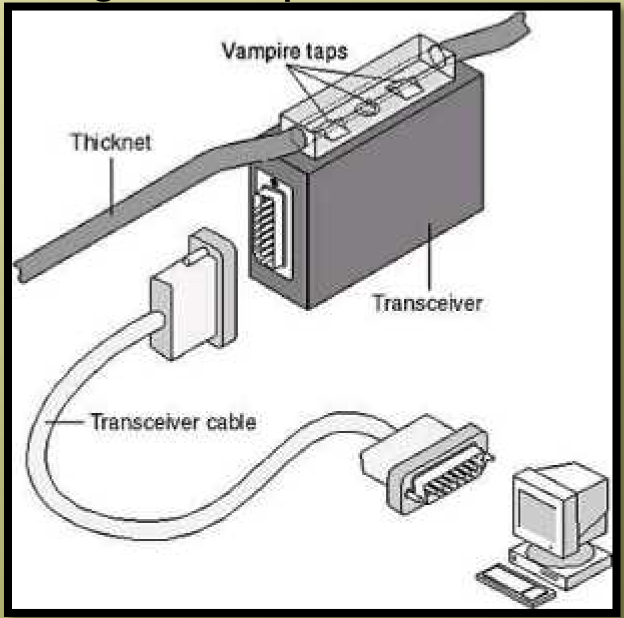


Possible Ethernet Connection

Ethernet

- the **transceiver's** primary purpose is to create an interface between the computer and the cable
- one of its main functions is to transmit bits onto the cable using CSMA/CD contention
- the connection to the network interface card is through a ***transceiver cable***; sometimes referred to as ***Attachment Unit Interface*** (**AUI**) cable
- can communicate with several devices through a multiplexer

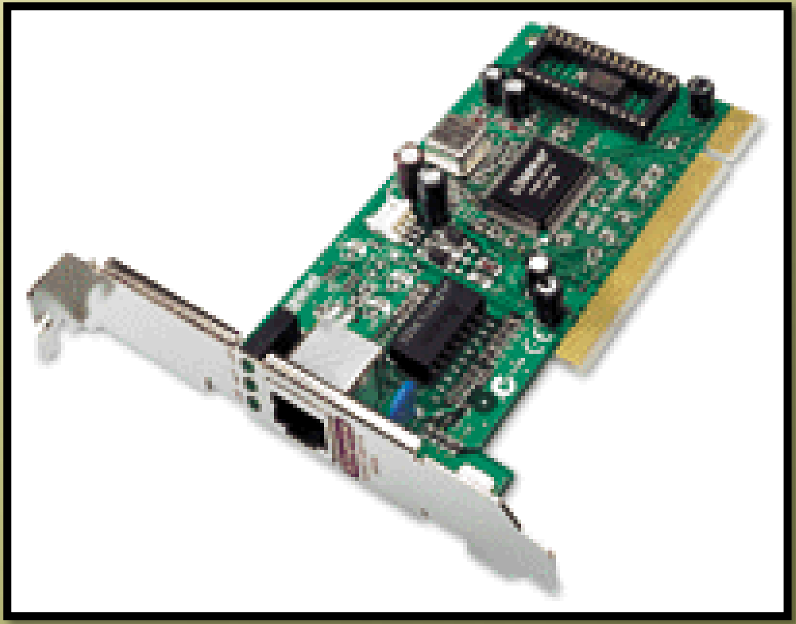
Ethernet Transceiver



Ethernet

- the **network interface card (NIC)** contains the logic necessary to buffer data and move it between the transceiver and the computer's memory

- it also recognizes frames on the LAN that is destined for its computer



Network Interface Cards

Ethernet

- activities of sending from one machine to another:
  - ➔ ***At the sending end:***
  - put packet into memory and signal NIC through internal bus
  - NIC creates correct frame format & stores packet into data field
  - NIC waits for a transceiver signal that the segment is clear
  - once this signal is received, NIC send the frame to the transceiver, which forwards it, as bits, onto the cable then listens for collision
  - if no collision, transceiver informs NIC
  - if collision occurs, NIC uses binary exponential backoff algorithm to determine when it should try again
  - if number of retry attempts is exhausted, NIC signals network software, which in turn signals the error to the user

Ethernet

- activities of sending from one machine to another (continue…):

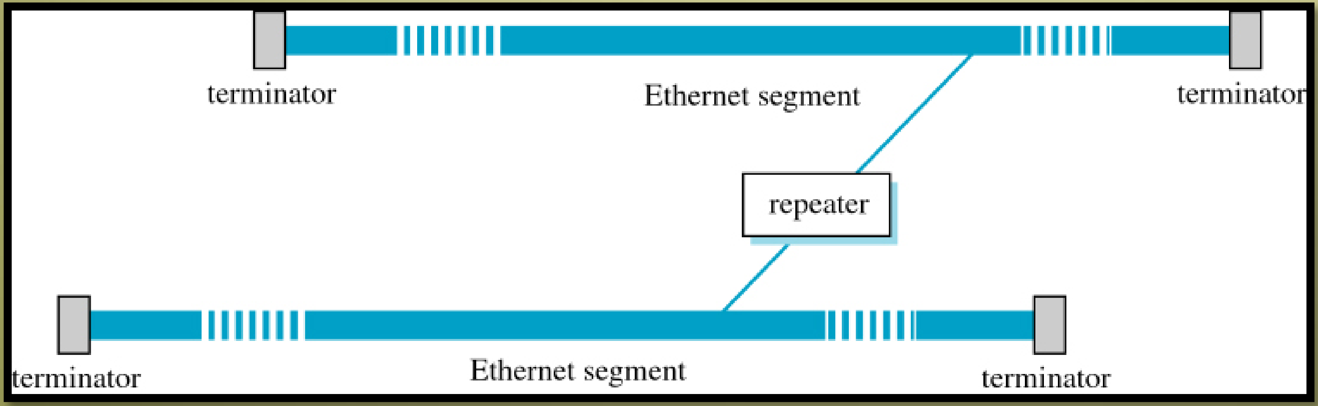
  At the receiving end:

  - transceiver monitors cable traffic, gets the frames and route them to NIC
  - NIC performs CRC error check and if the frame is error free, then check the destination address
  - if the destination address matches the machine of that NIC, then NIC extracts the packet from the frame, buffer it and sends an interrupt to the CPU
  - the machine executes network software and determines whether the packet should be accepted according to the used flow control algorithm
  - if all okay, the computer receives the packet; otherwise the network software responds according to the protocol at the next higher layer

Ethernet

Repeaters

- since signals degrade over distance, the original standard set the maximum length of the segment to 500 meters
- in many situations, this limit is smaller than the distance needed
- this problem was resolved by allowing multiple segments to be connected through repeaters
- a repeater receives the signal, regenerates and retransmits it
- the standard states that the maximum number of repeaters between any two devices is 4



Connecting Two Segments

Repeater

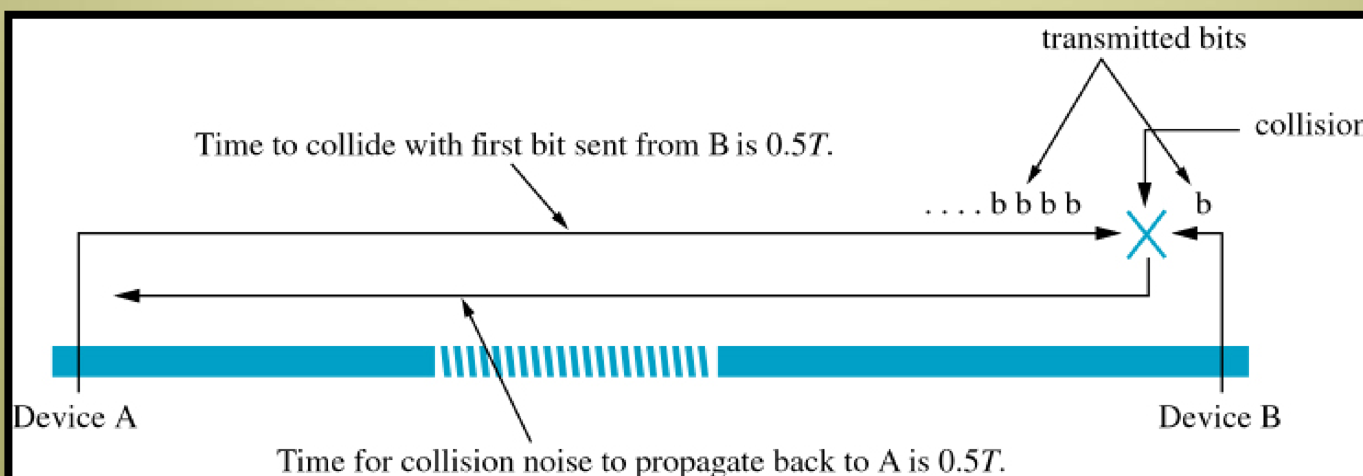
Ethernet

Frame Format

number of bytes

7	1	6	6	2	46–1500		4
preamble	start of frame delimiter	destination address	source address	data field length	data	pad	frame check sequence

Ethernet Frame Format



Maximum Time to Detect a Collision

Ethernet

What is the lower bound of data bytes?

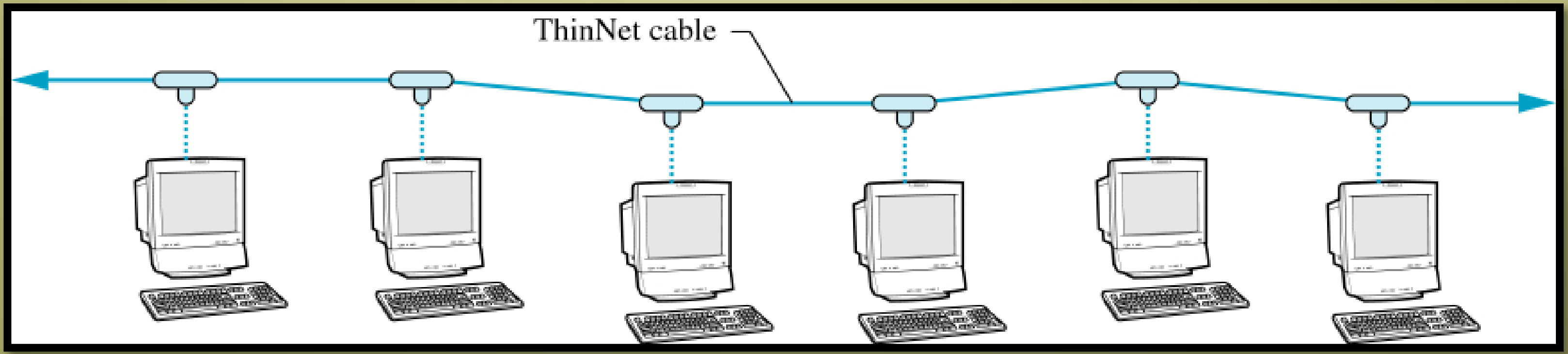
- maximum segment size with maximum repeaters: 500 * (4 + 1) = 2500 m
- signals travel on a copper wire with a speed of about 200 m / µsec
- it takes about 12.5 µsec to travel from one end to another
- total time to travel and come back is 25 µsec
- considering the different delays due to collision and repeaters processing for both the signal and the noise, worst case was set as double that time, which is 50 µsec
- each frame must take at least 50 µsec to send
- at 10Mbps (10 bits/µsec) rate, the device needs to send 500 bits in 50 µsec
- the was rounded, for safety, to 512 bits or 64 bytes

- finally, the data lower bound was set to 46 bytes to make sure that the frame will be large enough for CSMA/CD to work correctly

Ethernet

ThickNet, ThinNet & Hubs

- original implementation of the Ethernet used **10Base5** cable (a wide cable of 10 mm diameter); this was referred to as ***ThickNet***
- ThickNet had the advantage of allowing one segment to be up to 500 m
- however, the size of the wire represented a major disadvantage
- alternatively, transceiver logics were placed on NIC, and **10Base2** (a much thinner wire, referred to as ***ThinNet***) replaced 10Base5 wire
- ThinNet however has a higher resistance and so it allows a maximum segment size of 185 m



ThinNet Connections Using T-Connector

Ethernet

ThickNet, ThinNet & Hubs

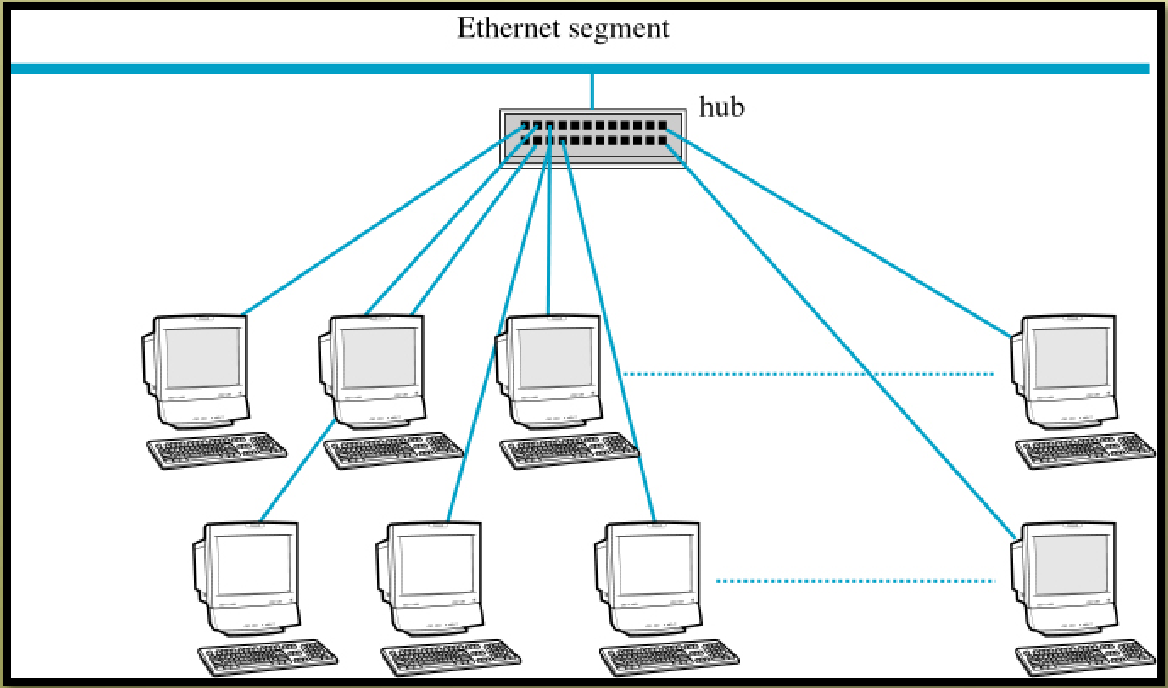
10Base2 T-Connector

Possible Transceiver Settings

Ethernet

ThickNet, ThinNet & Hubs

- a **Hub**, sometimes referred to as ***multiport* repeaters**, is a device with many ports; each connects to a device using 10BaseT cable
- a hub can also be connected to another hub
- with that, Ethernet has no longer a *physical* bus topology; so how does this affect the NICs, CSMA/CD mechanism, …etc.?



Connecting PCs with a Hub

10BaseT Cable

Fast Ethernet (100Mbps)

- IEEE 802.3u
- no change in the MAC layer details from 10Mbps Ethernet
- 10Basex runs mainly over coaxial cables
- 100Basex however runs over optical fibers, UTP or STP and uses star topology
- some of the fast Ethernet standards are:
  - 100BaseTX
  - 100BaseT4
  - 100BaseFX

Fast Ethernet (100Mbps)

100BaseTX

- designed to run over category 5 UTP
- 10Basex used Manchester coding
- using same Manchester coding but with a higher frequency would result in higher rate
- the higher frequency however over UTP produced a lot of interference
- using NRZI was an option that was finally ruled out due to its synchronization problems
- instead, 100BaseTX used **4B/5B Encoding**

Fast Ethernet (100Mbps)

- 4B/5B encoding replaces every ½ byte (4 bits) with 5 bits
- A string such as: 1010-0010-0000-0000-0000-0000 is hence replaced by: 10110-10100-11110-11110-11110-11110
- What is the advantage of that 4B to 5B transformation?

Coding Using 4B/5B

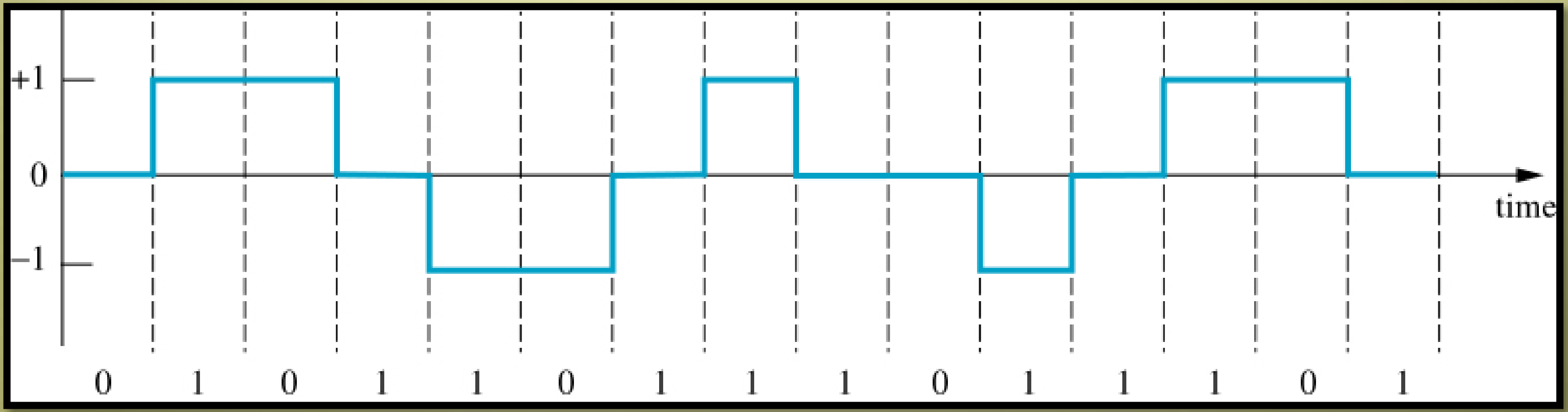
4-bit data	5-bit symbol
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Fast Ethernet (100Mbps)

- with 4B/5B, it was possible to use NRZI instead of Manchester
- however NRZI still produced noise over UTP even with lower-frequency signal
- to reduce the signal, a new signaling scheme, called **Multilevel Line Transmission-Three Levels (MLT-3)**, was used
- MLT-3 defines 3 state signals: -1, 0 & +1
- if bit is 0 → MLT-3 remains at current state
- if bit is 1 → MLT-3 moves to the next state

Fast Ethernet (100Mbps)

- How good is MLT-3 compared to Manchester coding?



Multilevel Line Transmission–Tree Levels (MLT-3)

Fast Ethernet (100Mbps)

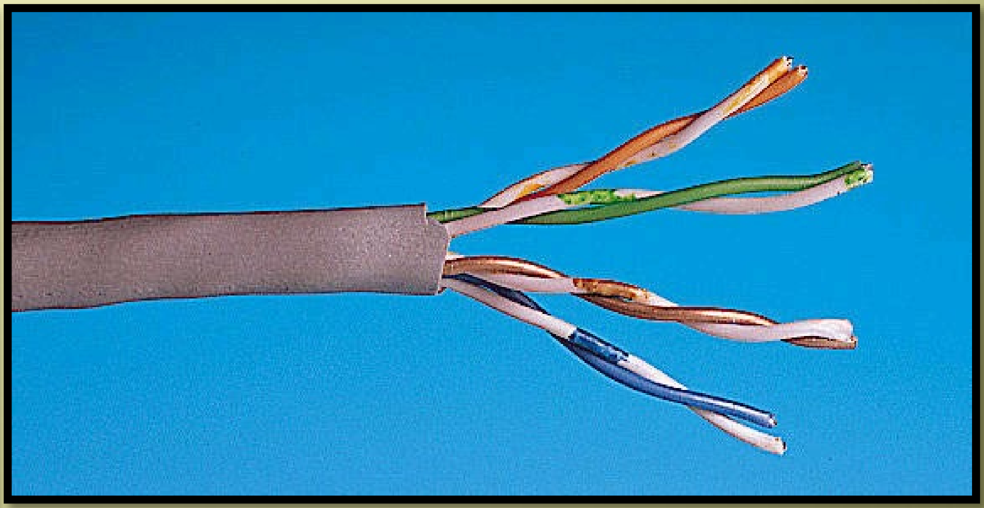
100BaseFX

- designed to run over optical fiber
- 100BaseTX, using UTP, has a maximum length of 100 meter
- 100BaseFX has a maximum length of 2 KM
- still uses 4B/5B
- NRZI is used instead of MLT-3 since optical fiber does not have the frequency constraint of UTP

100BaseT4

- designed to run over category 3 UTP (voice-grade wire)

Category 3 UTP



Category 5 UTP

Fast Ethernet (100Mbps)

100BaseT4

- the utilization of cat 3 UTP facilitated upgrades from 10Basex to Fast Ethernet without requiring new wiring
- however, cat 3 UTP is even more susceptible to noise than cat 5 UTP
- to overcome the problem, 100BaseT4 continue to use MLT-3 encoding but over **8B/6T** encoding scheme (rather than 4B/5B)

Fast Ethernet (100Mbps)

100BaseT4

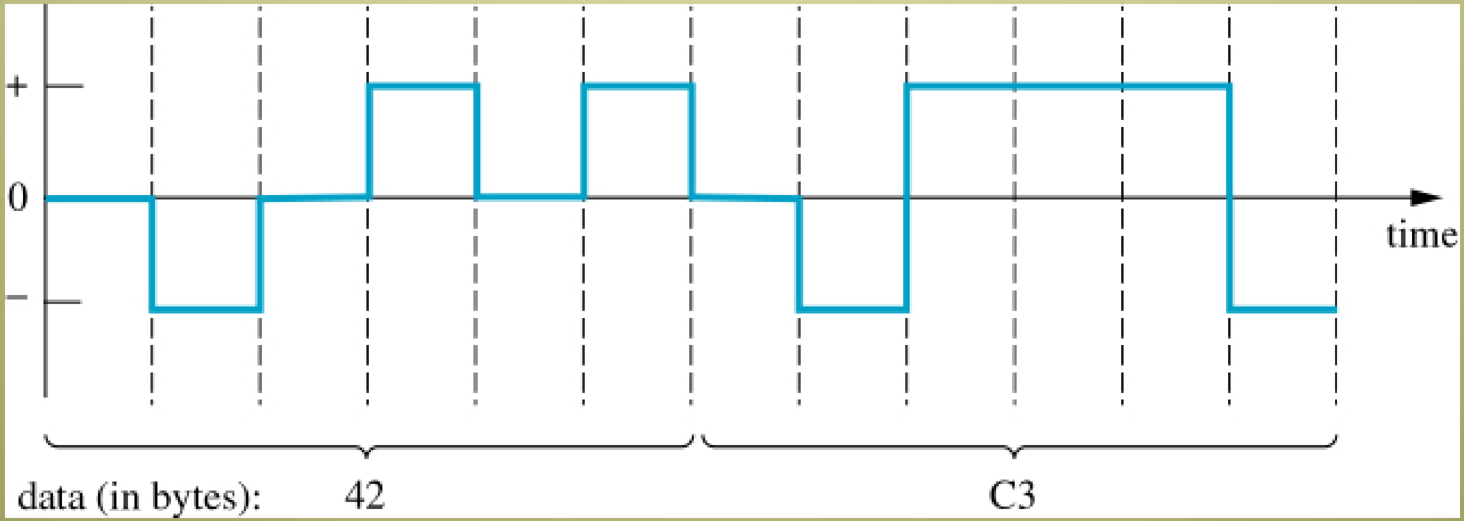
- 8B/6T associates each byte (8 bits) with a unique string of 6 ternary values, called **trits**
- 8 bits → 2^8 = 256 possible strings
- 6 trits → 3^6 = 729 possible trits
- each of the 256 strings can then be associated with a unique trit
- a trit is then represented by a signal of a +, 0 & - combination

Fast Ethernet (100Mbps)

100BaseT4

Data (Hex)	(Binary)	8B/6T Code
00	0000 0000	+-00+-
01	0000 0001	0+-+-0
....		
0E	0000 1110	-+0-0+
....		
FE	1111 1110	-+0+00
FF	1111 1111	+0-+00

Partial 8B/6T Encoding Table



8B/6T Encoding

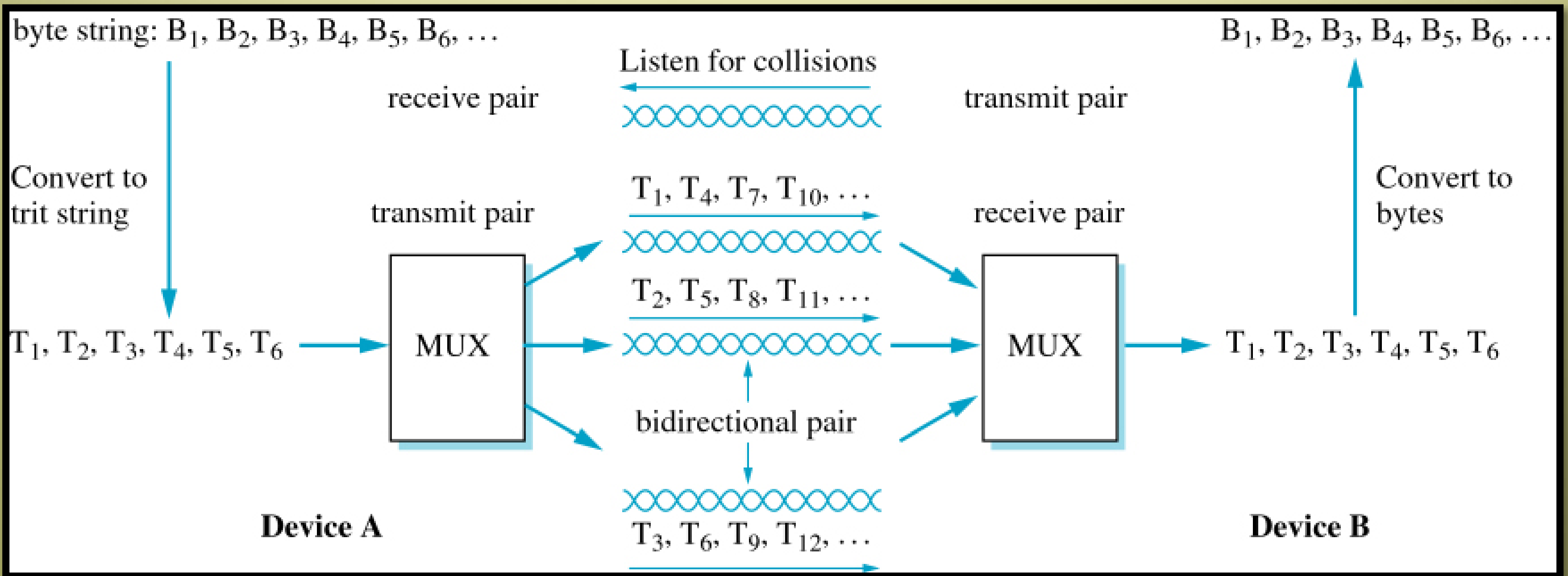
Fast Ethernet (100Mbps)

100BaseT4

- with 8B/6T, 8 bits are transmitted using 6 intervals
- although this is a frequency reduction of 25%, this is not enough to send without noise of cat 3 UTP
- to allow 100Mbps, 3 of the 4 UTP pairs are used for parallel transmission while the last one is used to sense collision
- each of the wires carries less trits (less frequency), so cat 3 UTP can handle
- using three pairs to send allows the needed 100Mbps (actually 75 M trits/second)
- the disadvantage is that 100BaseT4 can not operate in full-duplex mode

Fast Ethernet (100Mbps)

100BaseT4



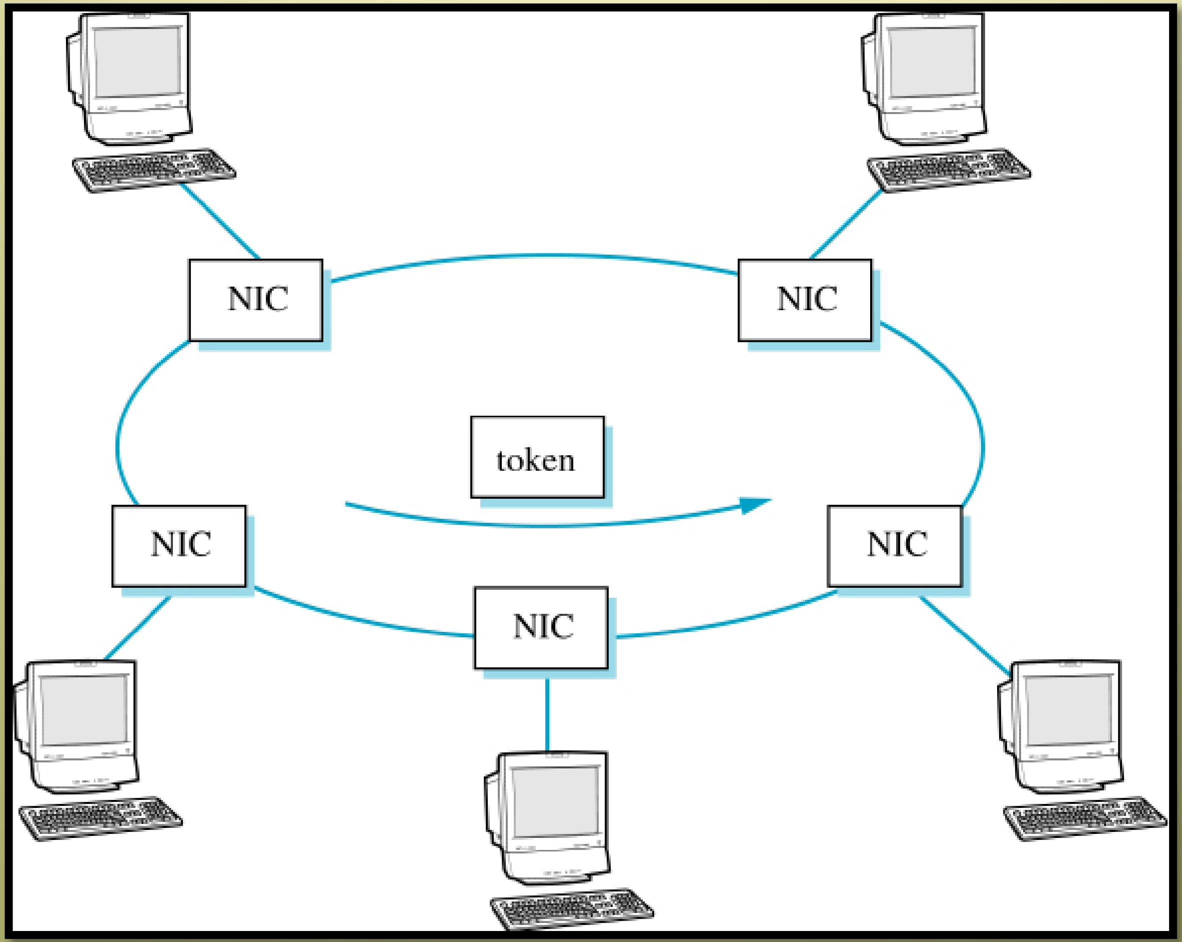
Sending Data on 100BaseT4 over Four Wire Pairs

Gigabit Ethernet

- 1000 Mbps rate
- designed to run over both fiber optics and copper
- supports both full-duplex and half-duplex
- 1000BaseSX & 1000BaseLX run over optical fiber
- 1000BaseT & 1000BaseCX run over copper wires
- in 2002, **10 Gigabit Ethernet** was developed by IEEE802.3ae task force

Token Ring

- IEEE standard 802.5



Token Ring Network & Circulating Token

Token Ring

- uses Differential Manchester encoding
- date rates are listed at 1Mbps & 4 Mbps (although IBM token rings support 4, 16 & 100 Mbps rates)
- issues:
  - how frames are transmitted
  - how rings are claimed and released
  - what happen if a device fails
  - how tokens and data frames can be distinguished

Token Ring

Token & Frame Formats

SD	AC	ED

(a) 3-byte token

number of bytes

1	1	1	2 or 6	2 or 6	0–5000	4	1	1
SD	AC	FC	destination address	source address	. . . data . . .	frame check sequence	ED	FS

(b) Variable-byte frame

```
SD (starting delimiter):  J K 0 J K 0 0 0
    AC (access control):  p p p t m r r r
                          p p p: priority bits
                              t: token bit
                              m: monitor bit
                          r r r: reservation bits
  ED (ending delimiter):  J K 1 J K 1 I E
     FC (frame control):  f f z z z z z z
                          f: frame type bits
                          z: control bits
      FS (frame status):  a c x x a c x x
                          a: address recognized bit
                          c: frame copied bit
                          x: undefined bit
```

Token and Frame Formats

Token Ring

Reserving & Claiming Tokens

- token can be passed from the one that just used it to its neighbor
- this scheme has its advantages and disadvantages
- each device is assigned an internal priority
- the token is also assigned a priority level; a device can claim the token if its priority is greater than the token priority level
- initially, the token priority is set to 0. The priority then changes by the **reservation system**, which is responsible for reserving tokens and assigning priorities

Token Ring

Ring Maintenance

- token problems are possible, for example,
  - Token may be damaged due to noise
  - Token may be lost if the device that has it crashes
- one of the devices is defined as a **monitor station**
- some of the problems, such as detection of an orphan frame or detection of a lost token, can be handled by the monitor station
- some other problems cannot be handled by the monitor station, such as a break in the ring or if the device that malfunctioning is the monitor itself
- these problems are handled using ***control frames***

Token Ring

Ring Maintenance

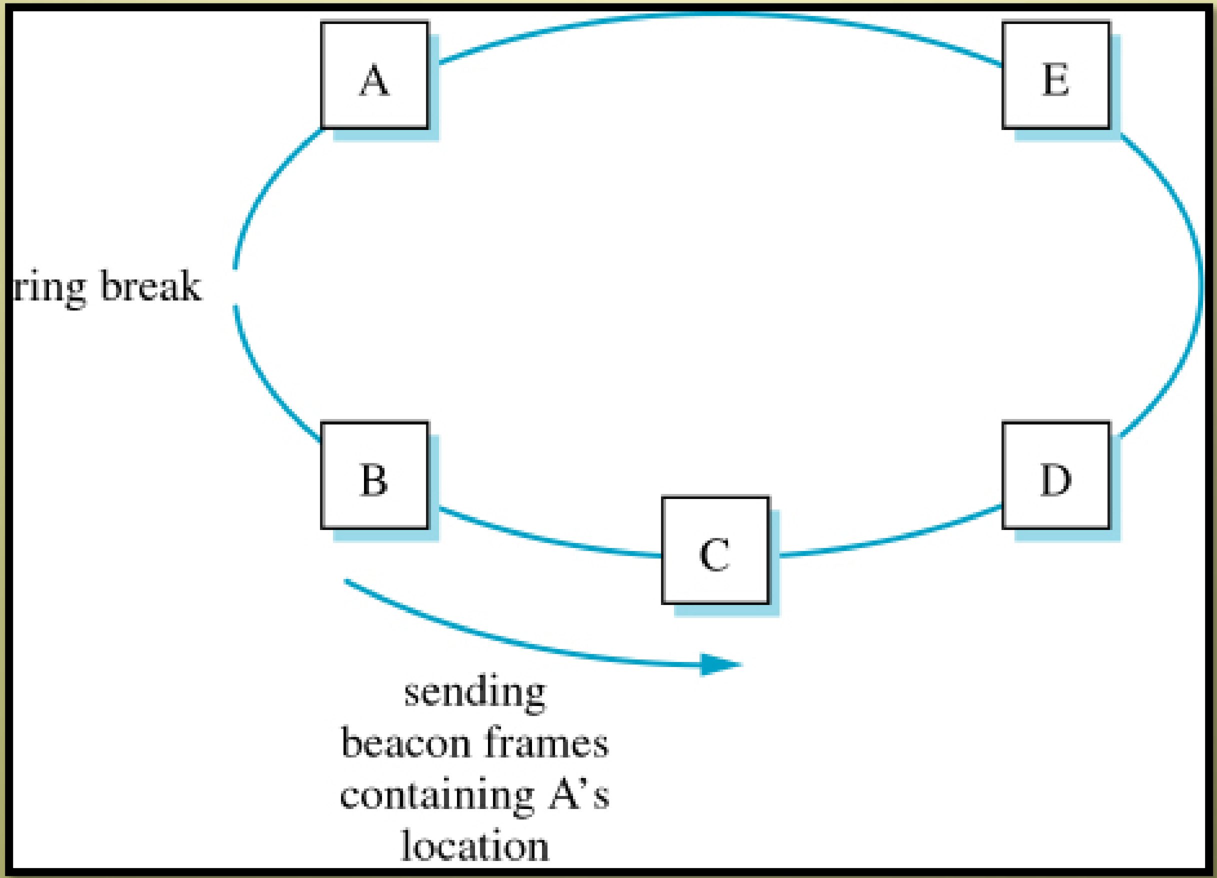
- the FC byte defines the frame's function

0001	Express buffer
0010	Beacon
0011	Claim token
0100	Ring purge
0101	Active monitor present
0110	Standby monitor present

Token Ring Control Frames

Token Ring

Ring Maintenance



Locating a Ring Break

MAC addresses

- 32-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)

IP Routing

- a device connected to an Ethernet would sense an Ethernet address on the segment to know which packet is destined for it

- however, this Ethernet address is a 48-bit address that has no significance on a global IP scale

- so, how can the device then recognize a packet containing an IP address?

IP Routing

- how the router determines the physical address from the IP address when the packet is embedded into a LAN frame

- when the router receives an IP packet, there are two possibilities:
  - The packet's destination machine is in a network where the router is attached, or
  - It is not

- if the destination machine belongs to the same network, then the router can directly send the packet to the destination; that is called ***Direct Routing***

- the router will know that since the network part of the IP address is the same as its own network part

IP Routing

- but still, how can the router determine the physical address from the IP address?
- one approach is the ***Dynamic Binding***, also called ***Address Resolution Protocol***
- the router transmits a broadcast request to all devices in the LAN, specifying the IP address
- the device with the specified IP responds with its physical address
- the router can then sends the packet to the proper device; it also stores this information on a local cache for future requests

IP Routing

- what if the destination is not directly reachable through one of the router's networks?
- the router then uses hierarchical routing, as discussed in previous lectures, to determine another router to send the packet to
- the packet will then travel from one router to another until it reaches a router connected through the same network to the destination machine

IP Routing

- Example

IP Routing