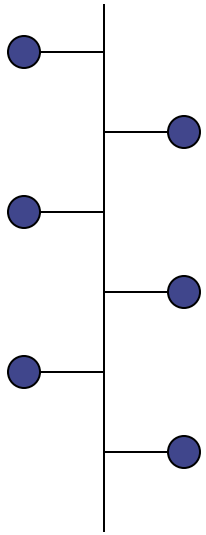
Rechnernetze und Verteilte Systeme

Introduction to Communication Networks and Distributed Systems



Unit 10: Physical and Data Link Layer

Prof. Dr.-Ing. Adam Wolisz

TKN Telecommunication Networks Group

Physical and Data Link Layer

Physical layer

- Encoding
- Duplexing

Data link layer

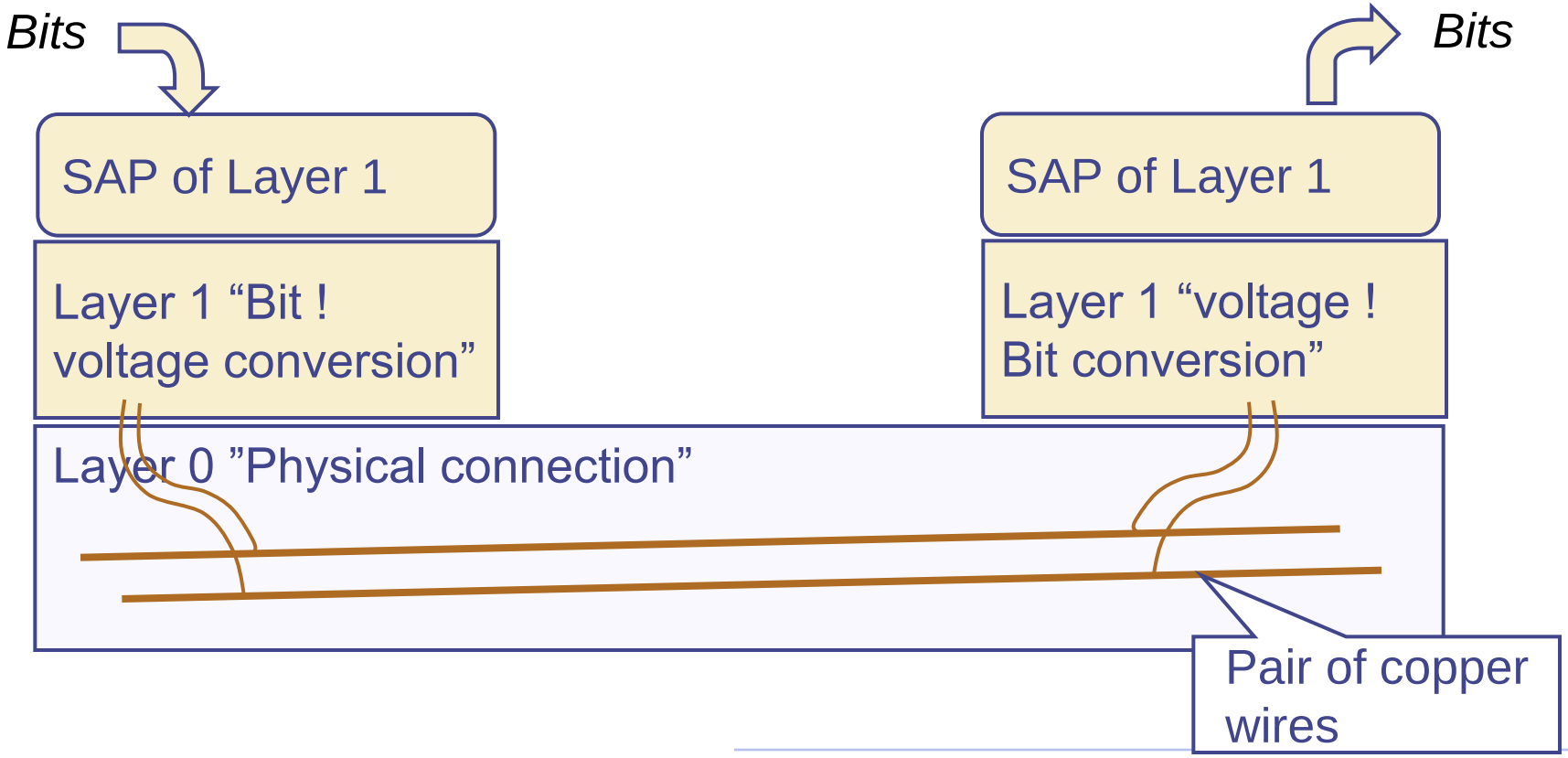
- Error detection, correction
- Sharing a broadcast channel: multiple access
- Addressing
- Reliable data transfer, flow control

Physical Layer (Layer 1)

- Tasks for the physical layer
  - Transmits raw bits over a physical link connecting nodes
  - Creates frames around the code words and symbols
  - Provides an electrical, mechanical, and procedural interface to the transmission medium
- Receiver creates a bit stream and delivers to the layer above for further processing
- Properties
  - No guarantees for reliable transmission
  - Auto negotiation = procedure for choosing common transmission parameters such as
    - Speed
    - Duplex mode
    - Flow control between participating network devices

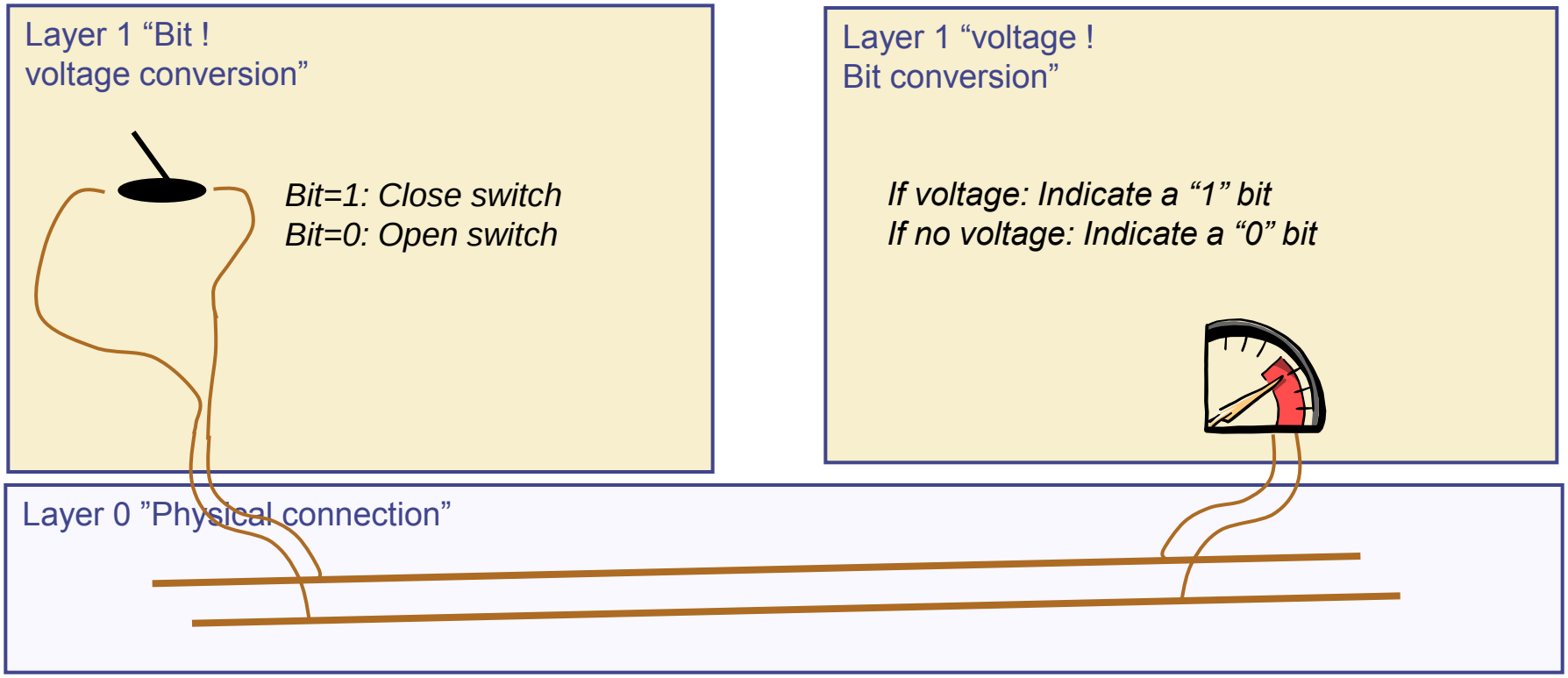
Basic service of physical layer: transport bits

- Physical layer should enable the transport of bits between two locations A and B
- Abstraction: Bit sequence – correct, in order delivery



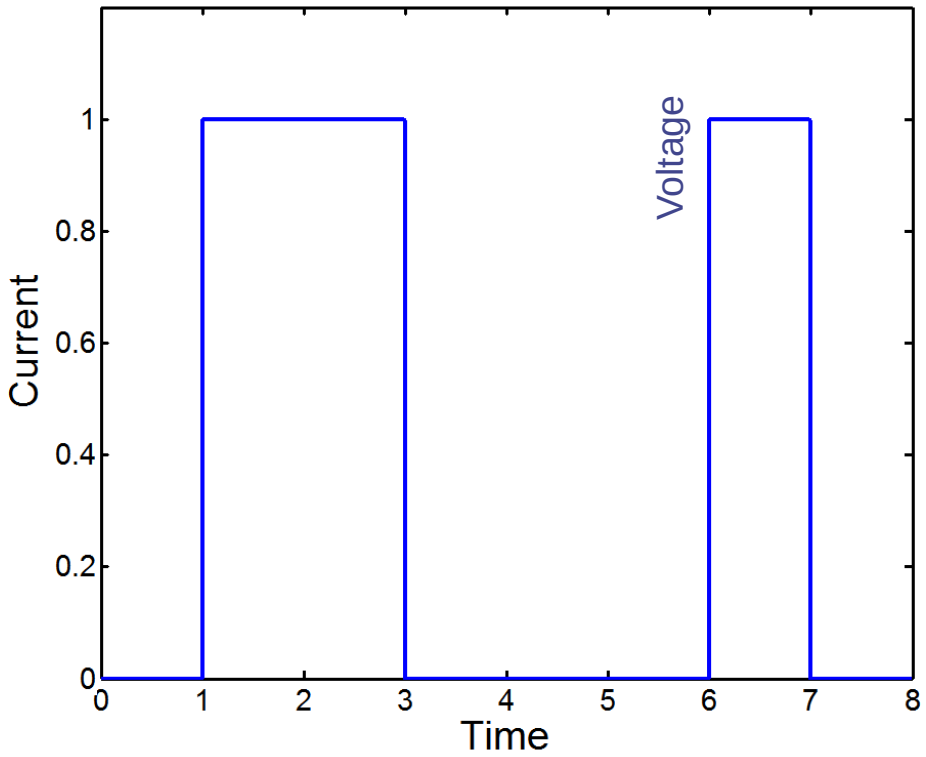
A bit to signal conversion rule

- A simple conversion rule
  - For a “1” bit, apply voltage to the pair of wires
  - For a “0” bit, no voltage



Example: Transmit bit pattern for character “b”

- Character “b” needs a representation as sequence of bits
- Option 1: Use the ASCII code of “b”, 98, as a binary number 01100010
- Resulting voltage put on the wire



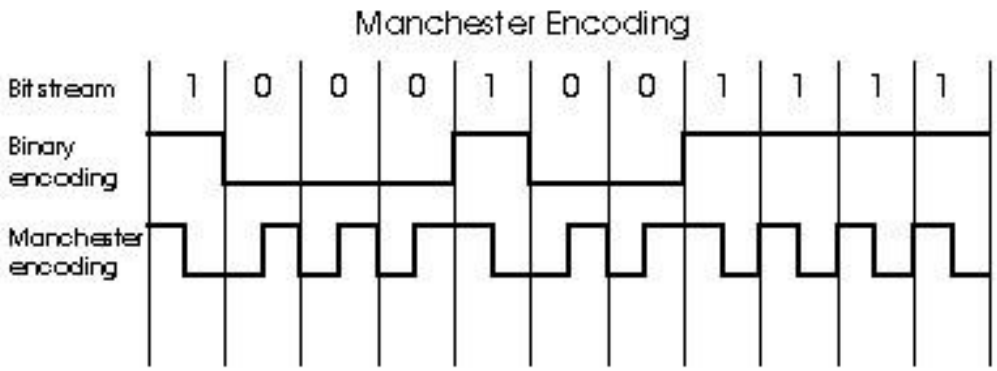
Note: Abstract data is represented by physical signals – changes of a physical quantity in time or space!

Problems

- Differentiation between no activity and tramsmitted zero needed
  - ⇒Change the coding to 5V for 1 and -5V for 0
- How many bits are transmitted in a row
  - ⇒Timer needed that is perfectly syncronized between sender and receiver (initial synchronization and regular “correction of deviation”
  - ⇒Expensive solution

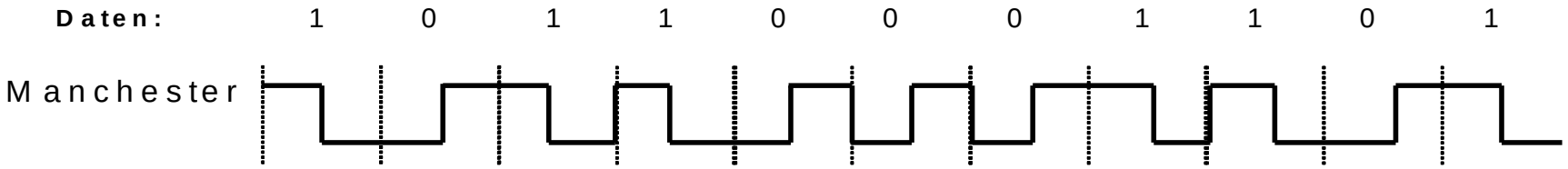
⇒Next idea to avoid timers

- Count the transition from low to high and vice versa



Manchester encoding

- Idea: At each bit, provide indication to receiver that this is where a bit {starts/stops/has its middle}
  - Example: Manchester encoding
  - For a 0 bit, have the signal change in the middle of a symbol (=bit) from low to high
  - For a 1 bit, have the signal change in the middle of a symbol (=bit) from high to low



- Ensures sufficient number of signal transitions
  - Independent of what data is transmitted!
  - Used in Ethernet signal coding

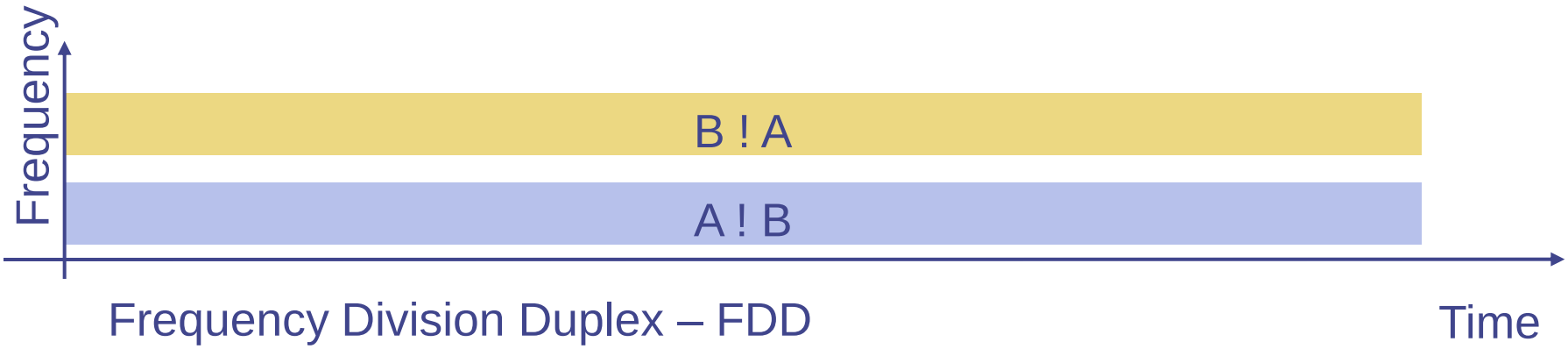
Duplexing

- Direction of data transfer
  - One-way (simplex) or
  - Two-way (duplex)
- Simplex operation: trivial
- Half duplex
  - Two pairs of cables, one for each direction – wasteful
  - Use one cable intelligently – participants alternatively transmit, wait their time until it is their turn
    - Both sending at the same time would not work, signals interfere
  - Problem: How can one node decide that the other is done sending?

Time division duplex – TDD

How to realize duplexing?

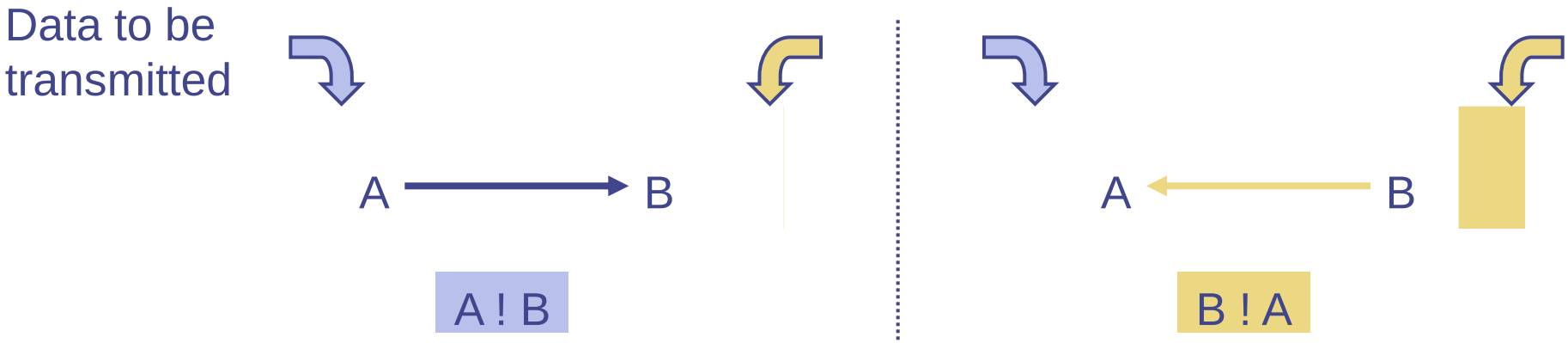
- Full duplex
  - Two pairs of cables would work, but still overhead (installation, maintenance, …) – does it work with one cable also?
  - Exploit some properties of the physical medium
    - Here: transmissions in different frequencies do not interfere
    - Idea: use different frequencies for transmission in different directions



Frequency Division Duplex – FDD

How to realize duplexing?

- Full duplex by time division duplexing?
  - Sounds like a contradiction: both A and B always have data to send, but have to take turns?
  - “Having data to send” corresponds to a certain data rate – bits per second
  - How about intermediately storing data when the other station is currently sending? Then quickly send all stored & new data



Time division duplexing can realize full duplex if transmission over medium is at least twice as fast as data is to be transmitted

Lessons learned from duplexing

- It is useful to distinguish between
  - *Requirements* on what should be possible
  - *Rules and methods* how to implement such requirements
  - Example: Implement a “full duplex” requirement using TDD

- Buffering is an important means to decouple different dynamics in time
  - Questions of buffer overflow have to be considered

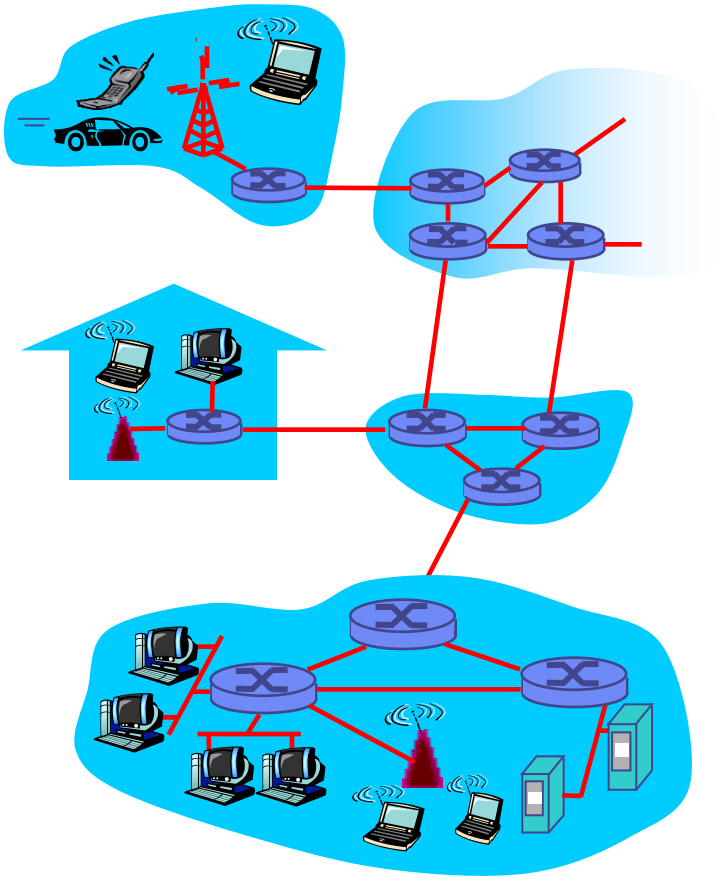
Physical Layer (Layer 1)- summary.

- Tasks for the physical layer
  - Transmits raw bits over a physical link connecting nodes
  - Creates frames around the code words and symbols
  - Provides an electrical, mechanical, and procedural interface to the transmission medium
- Receiver creates a bit stream and delivers to the layer above for further processing
- Properties
  - No guarantees for reliable transmission
  - Auto negotiation = procedure for choosing common transmission parameters such as
    - Speed
    - Duplex mode
    - Flow control between participating network devices

7.2 Data link layer (Layer 2): Introduction

- Terminology
  - Hosts and routers = nodes
  - Communication channels that connect adjacent nodes along communication path are links
    - wired links
    - wireless links
    - LANs
- Layer-2 packet is a frame, encapsulates datagram

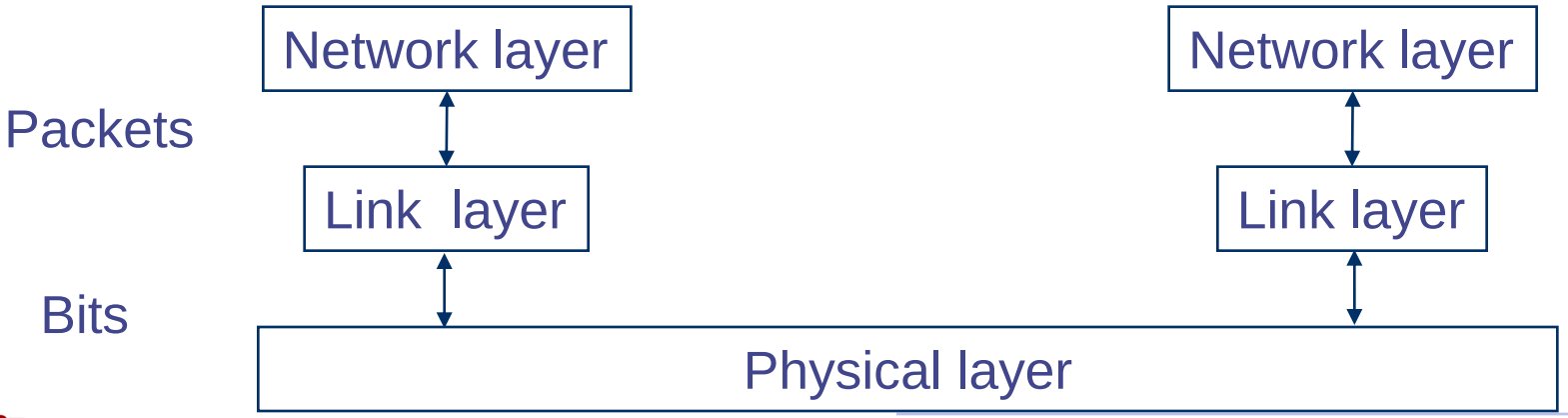
data-link layer has at least the responsibility of transferring datagrams from one node to adjacent node over a link



The link layer's service

- Link layer sits on top of the physical layer
  - Can use a bit stream transmission service
  - But: this service might have incorrect bits

- Expectations of the higher layer (networking layer)
  - Wants to use either a packet service or, sometimes, a bit stream service (rather unusual)
  - Does not really want to be bothered by errors
  - Does not really want to care about issues at the other end

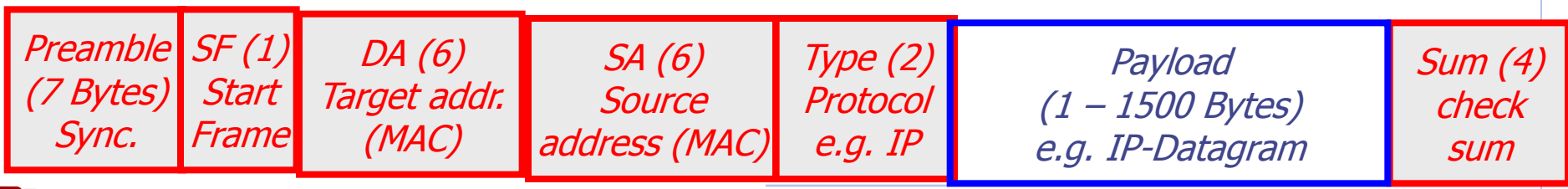


Options for link layer service

- Reliable (dependable) service – yes/no
  - Reliability has many facets
    - A delivered packet should have the same content as the transmitted packet
    - All packets have to be delivered
    - Eventually, packets have to be delivered in order
  - Error control may be required
    - Forward error control, acknowledgements
- Connection-oriented – yes/no
  - Should a context be setup to/with the peer entity?
- Packet or bit stream abstraction
  - Usually in computer networks: packets
  - What about a maximal packet length?

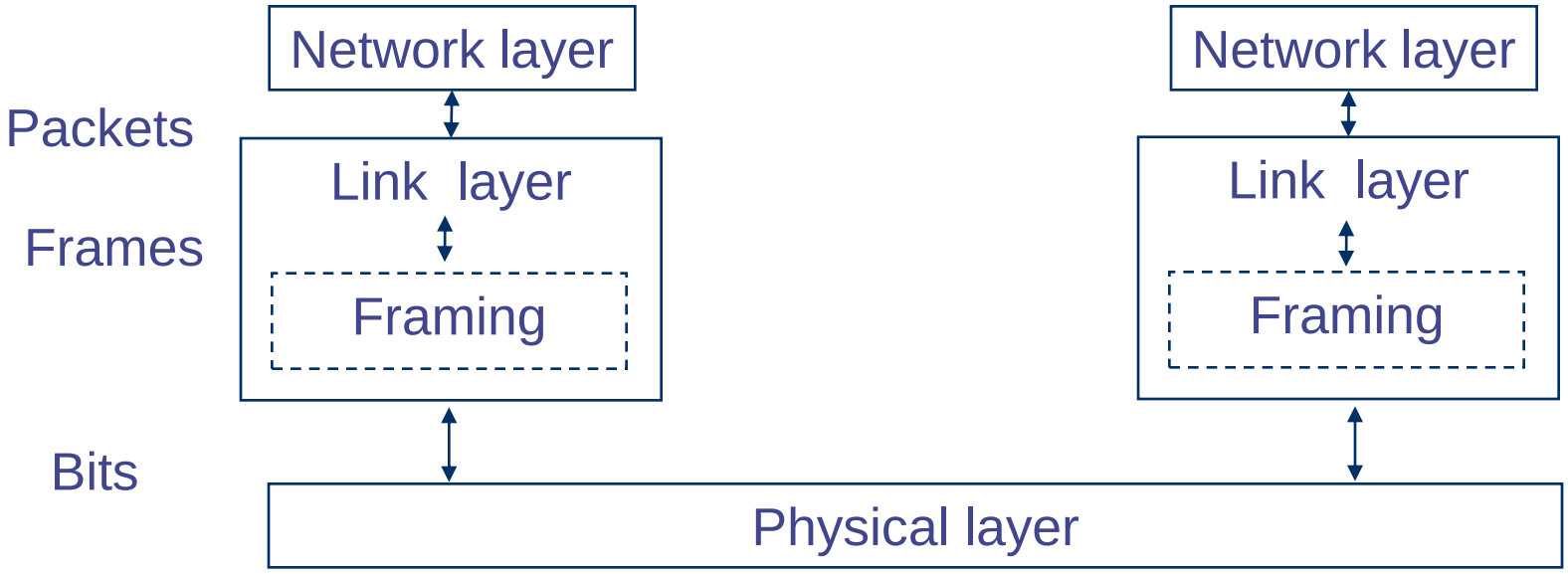
Datagrams: Packets in conectionless transmission!

- Definition
  - Datagram = basic transfer unit associated with a packet-switched network
  - No guarantees regarding delivery, arrival time, and order of arrival
- Datagram analogous to post packet
  - Header = all information needed to deliver the packet (no other knowledge required) or to return the packet to the sender
  - Payload = content of the packet
- Typical Frame Format (Datagram)
  - Here the Ethernet format …(see later..)



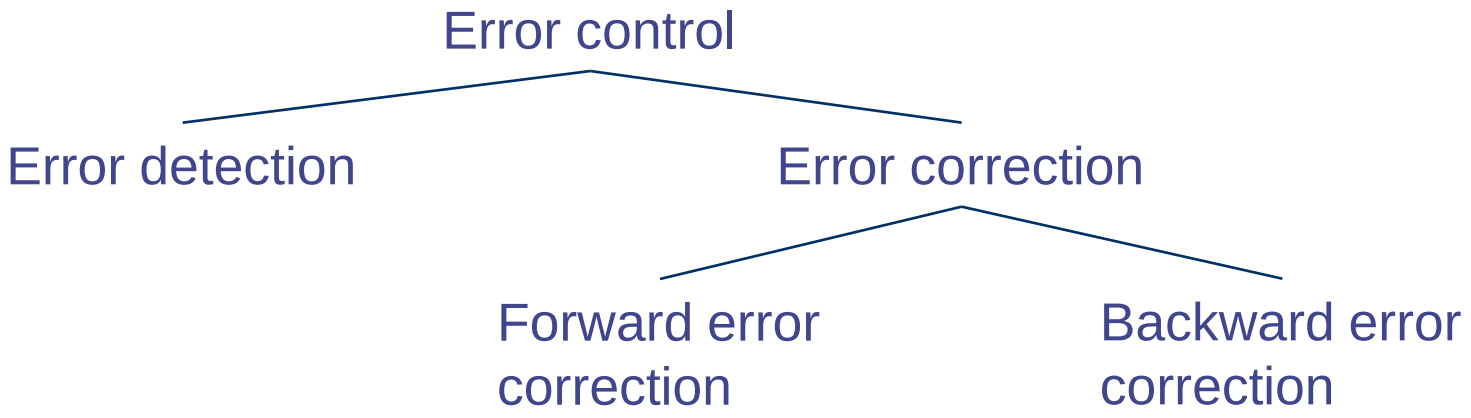
Basic link layer functions – Framing

- How to turn a physical layer's bit stream abstraction into individual, well demarcated frames?
  - Usually necessary to provide error control – not obvious how to do that over a bit stream abstraction
  - Frames and datagrams are really the same thing, only a convention to talk about "frames" in the link layer context
- In addition: Fragmentation & reassembly if network layer packets are longer than link layer packets



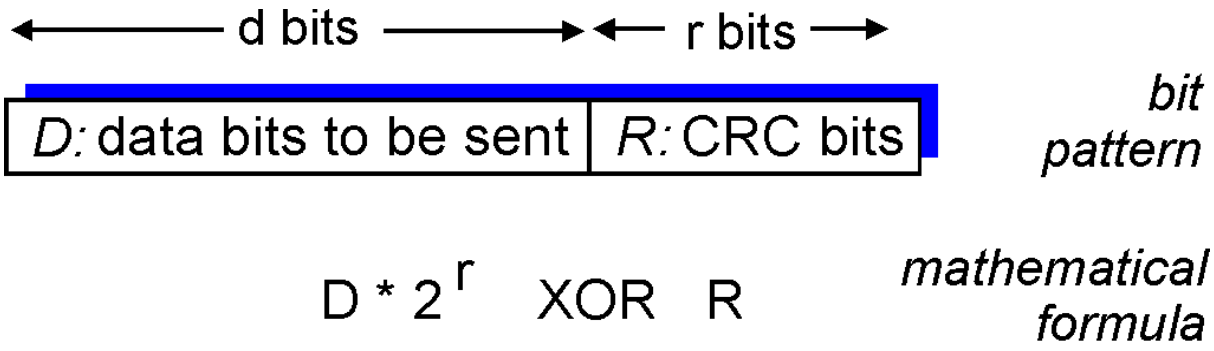
Basic link layer functions – Error control

- Error control if desired by the network layer – usually is
- Error detection – are there incorrect bits?
- Error correction – repair any mistakes that have happened
  - Forward error correction – invest effort before error happened; try to hide it from higher layers
  - Backward error correction – invest effort after error happened; try to repair it



Checksumming: Cyclic Redundancy Check

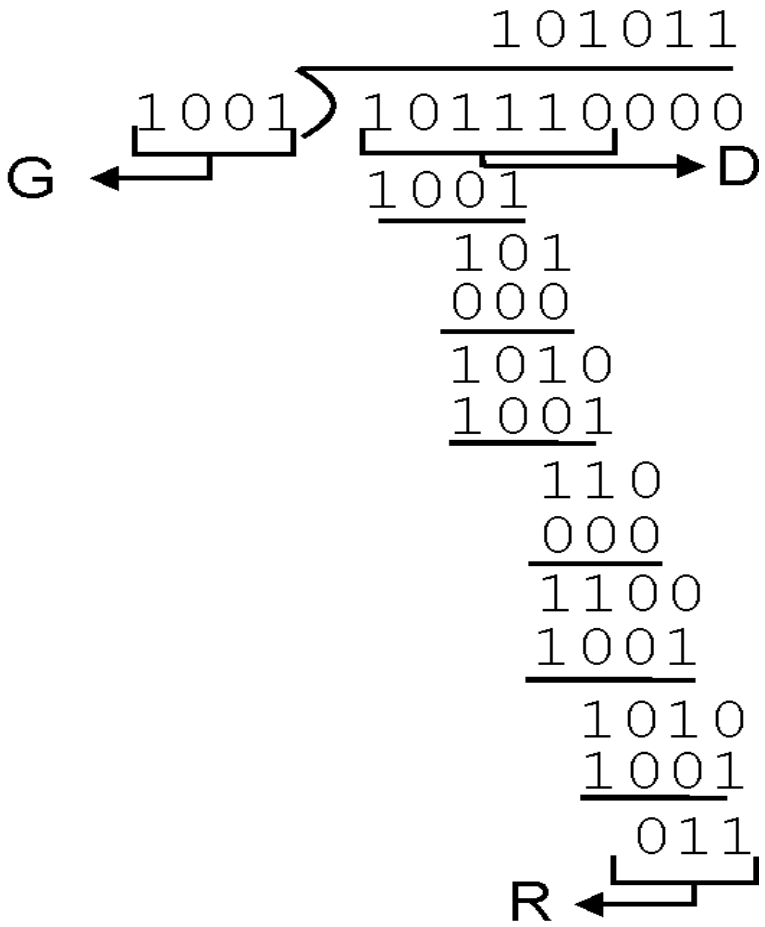
- View data bits, D, as a binary number
- Choose r+1 bit pattern (generator), G
- Goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - ⇒can detect all burst errors less than r+1 bits
- Widely used in practice (Ethernet, 802.11 WiFi, ATM)



CRC Example

- Want
  - $D*2^r$ XOR R = nG
- Equivalently
  - $D*2^r$ = nG XOR R
- Equivalently
  - if we divide $D*2^r$ by G, want remainder R

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



Basic link layer functions – Connection setup

- Connection: Shared state at sender and receiver
- Connections useful for many purposes
  - Application context
  - Error control – several error control schemes rely on a common context between sender and receiver
- Control of the Medium Access
  - Which station is allowed to use the connection at given time?
- Separating the data into frames for the transport
  - Modem: PPP (point to point protocol), SLIP (serial line IP protocol)
  - LAN: LLC (Logical Link Control) uses 802.X protocols
- The frames used in connection – oriented mode usually have additional CONTROL FIELD…

Basic link layer functions – Flow control

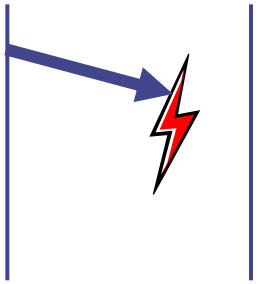
- What happens with a fast sender and a slow receiver?
  - Sender will overrun buffers faster than the receiver can process the packets in that buffer
  - Lots of transmission effort is wasted in this case

Thirsty?
Drink!

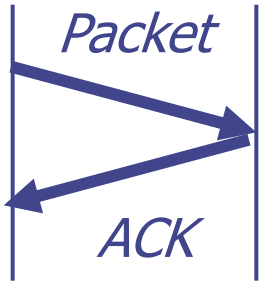
- Necessary to control the amount of frames a link layer sends per unit time, adapt to receiver's capabilities

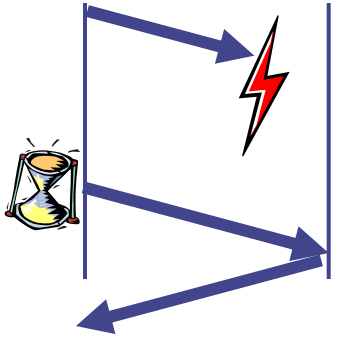
Possibly - Repairing errors: Repeat packet

- What happens if a packet is lost on the way?
- Idea 1: Have the receiver tell the sender that the packet was lost
  - But how would the receiver know that a packet was on the way in the first place?
  - ! Doesn't work!



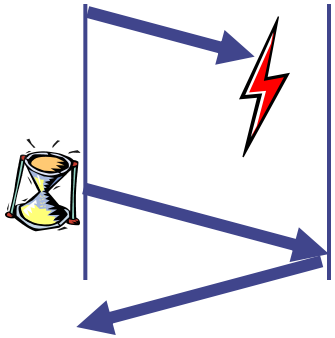
- Idea 2: Turn idea 1 upside down – receiver tells sender when a packet has arrived
  - An acknowledgement
  - When packet lost, acknowledgement will not arrive
  - Sender can wait for acknowledgment, when it not arrives at expected time, resend the packet
  - A timeout is used, forming an Automated Repeat Request (ARQ) protocol

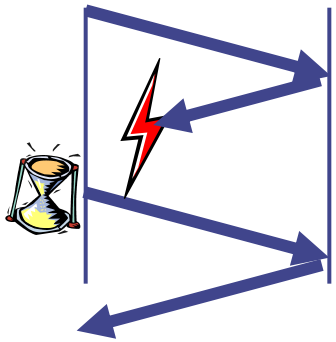




Problem: How can different error situations be distinguished?

- How to tell apart:

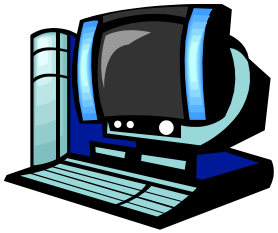


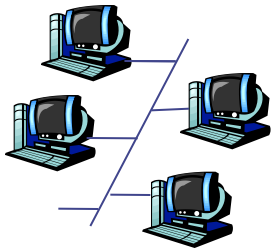


  - Looks the same for the sender, but receiver will get different sequences
- Receiver needs to know whether a packet is a new one or one that is repeated
- Introduce sequence numbers to identify packets
- Requires state _ possible only in connection –oriented style…

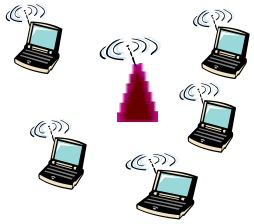
Multiple Access Links and Protocols

- Two types of “links”
  1. Point-to-point for dial-up access and as link between Ethernet switch and host
  2. Broadcast (shared wire or medium)
     - old-fashioned Ethernet
     - 802.11 wireless LAN

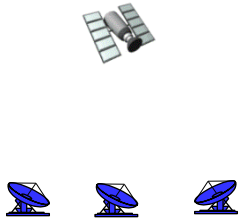




shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple access protocols

- Single shared broadcast channel
  - Two or more simultaneous transmissions by nodes with possible interference
  - ⇒Collision if node receives two or more signals at the same time
- Multiple access protocol
  - Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
  - Communication about channel sharing must use channel itself!
  - ⇒no out-of-band channel for coordination

Ideal Multiple Access Protocol

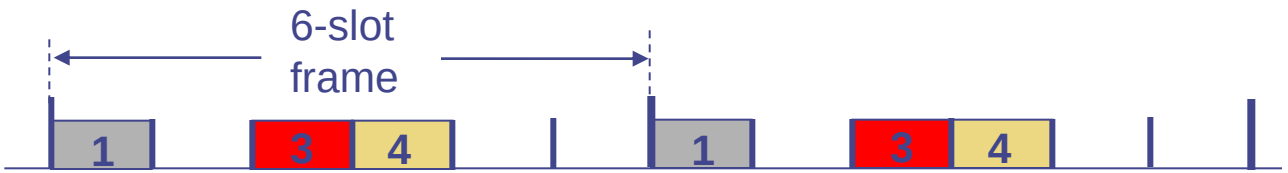
- Broadcast channel of rate R bps
  - when one node wants to transmit, it can send at rate R.
  - when M nodes want to transmit, each can send at average rate R/M
  - fully decentralized
    - no special node to coordinate transmissions
    - no synchronization of clocks, slots
  - simple

Taxonomy of MAC Protocols

- Three broad classes
  1. Channel Partitioning
     - divide channel into smaller “pieces” (time slots, frequency, code)
     - allocate piece to node for exclusive use
  2. Random Access
     - channel not divided, allow collisions
     - “recover” from collisions
  3. “Taking turns”
     - nodes take turns, but nodes with more to send can take longer turns

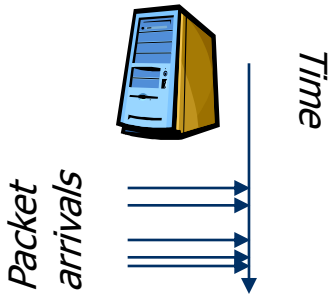
Channel Partitioning MAC protocols: TDMA

- TDMA: time division multiple access
  - access to channel in "rounds"
  - each station gets fixed length slot (length = pkt trans time) in each round
  - unused slots go idle
  - Example:
    - 6-station LAN
    - 1,3,4 have packets
    - slots 2,5,6 idle

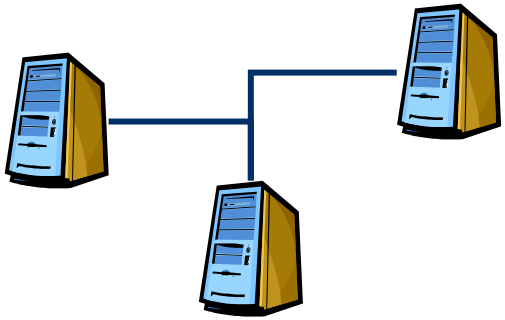


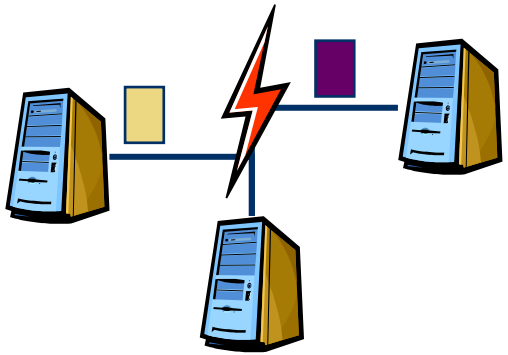
Assumptions for dynamic channel allocation

- Dynamic medium allocation
  - Assign channel/link/resource to that source that currently has data to send
  - No fixed assignments



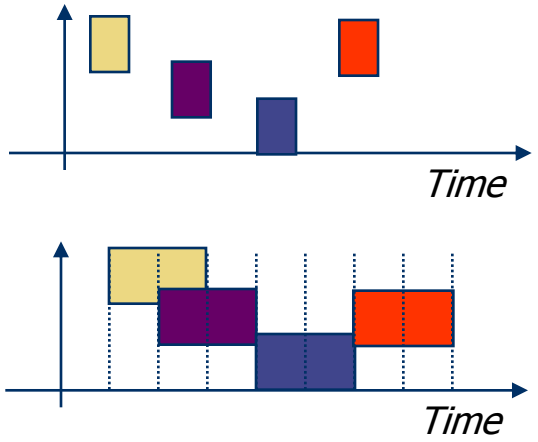
- Variations
  - Station model (or terminal model)
    - N independent stations want to share a given resource
    - Only a single channel for all stations
    - No possibility to communicate/signal anything via other means
  - Collision assumption
    - Only a single frame can be successfully transmitted at a time
    - Two (or more) frames overlapping in time will collide and are both destroyed
    - No station can receive either frame



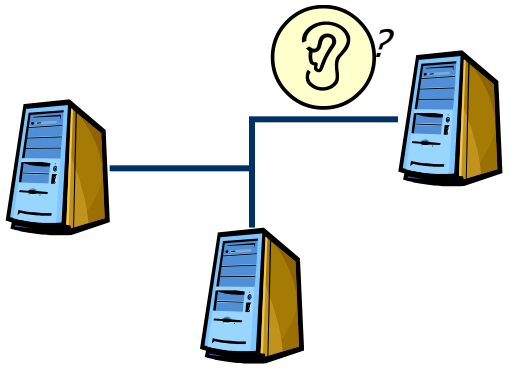


Assumptions for dynamic channel allocation

- Time model
  - Continuous time: Transmissions can begin at any time; no central clock
  - Slotted time: Time is divided in slots; transmissions can only start at a slot boundary. Slot can be idle, a successful transmission, or a collision



- Carrier Sensing
  - Stations can/cannot detect whether the channel is currently used by some other station
  - There might be imperfections involved in this detection (e.g., incorrectly missing an ongoing detection)



Figures of merit

- How to judge the efficiency of a dynamic channel allocation system?
  - Intuition: transmit as many packets as quickly as possible
- At high load (many transmission attempts per unit time): Throughput is crucial – ensure that many packets get through
- At low load (few attempts per time): Delay is crucial – ensure that a packet does not have to wait for a long time
- Fairness: Is every station treated equally? Or justifiable inequality?

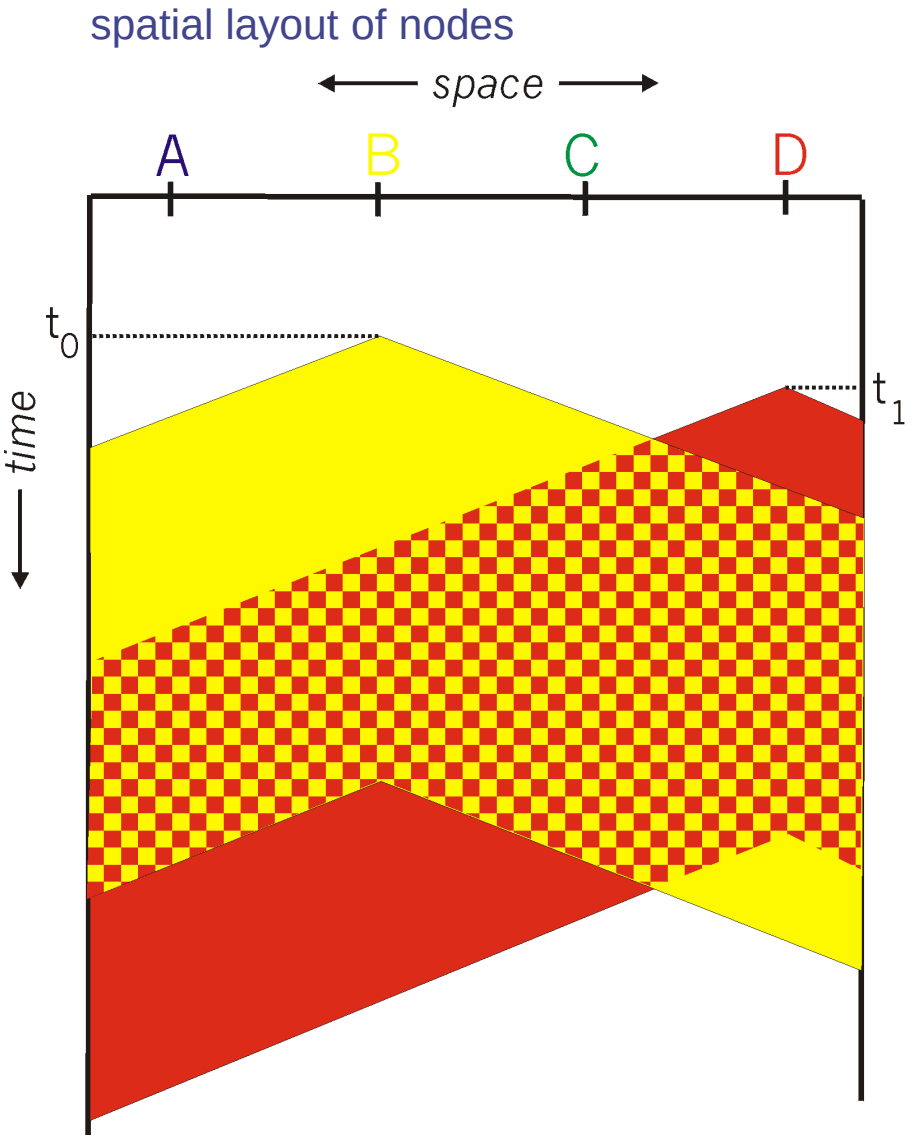
CSMA (Carrier Sense Multiple Access)

- CSMA = listen before transmit
  - If channel sensed idle, transmit entire frame
  - If channel sensed busy, defer transmission

- Human analogy: don’t interrupt others!

CSMA collisions

- Collisions can still occur
  - propagation delay means two nodes may not hear each other's transmission
- Collision
  - entire packet transmission time wasted
- Note
  - Role of distance & propagation delay in determining collision probability

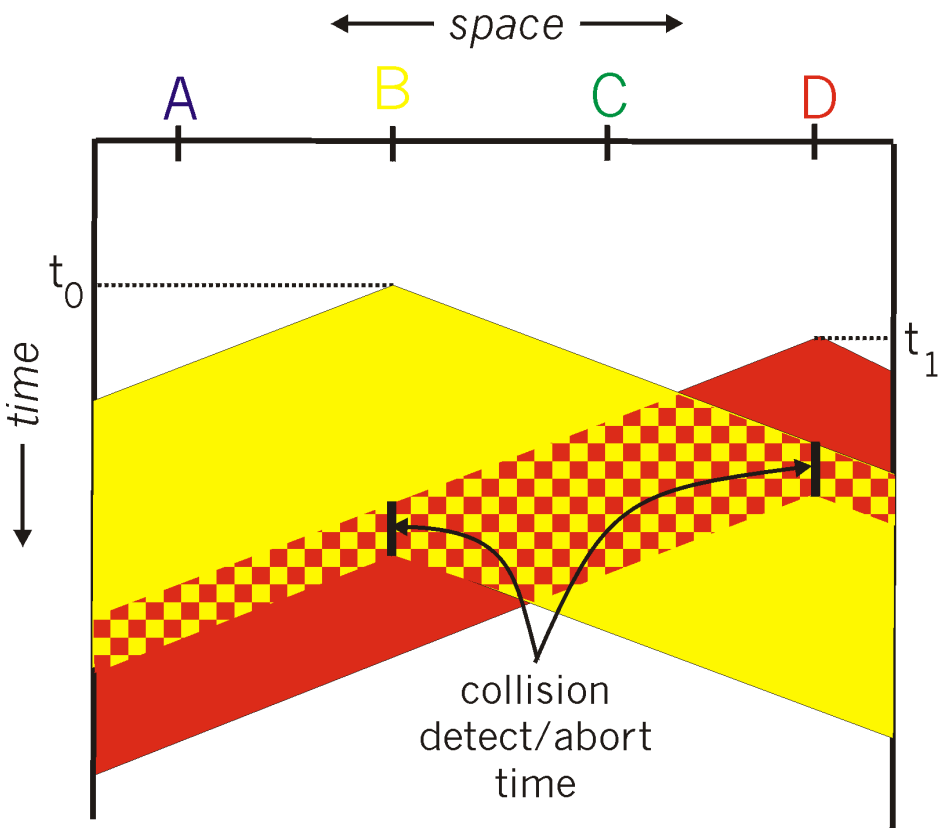


CSMA/CD (Collision Detection)

- 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: received signal strength overwhelmed by local transmission strength

CSMA/CD collision detection



- Implementation in Ethernet: few slides later

MAC Layer Addressing

- Hardware address for unique identification of every device (network card, switch, router, ….) in the network
- Example: MAC address for Ethernet
  - 48 bits, hexadecimal notation, e. g. 08-00-20-ae-fd-7e
    - First 24 bits = Vendor identification defined by IEEE (e.g. 00-50-8b-xx-xx-xx for Compaq)
    - Second 24 bits = defined by the vendor for each network interface
  - World-wide unique address ⇒ Application for automatic device configuration, e.g. with DHCP

MAC vs. IP address

- 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

MAC Addresses and ARP

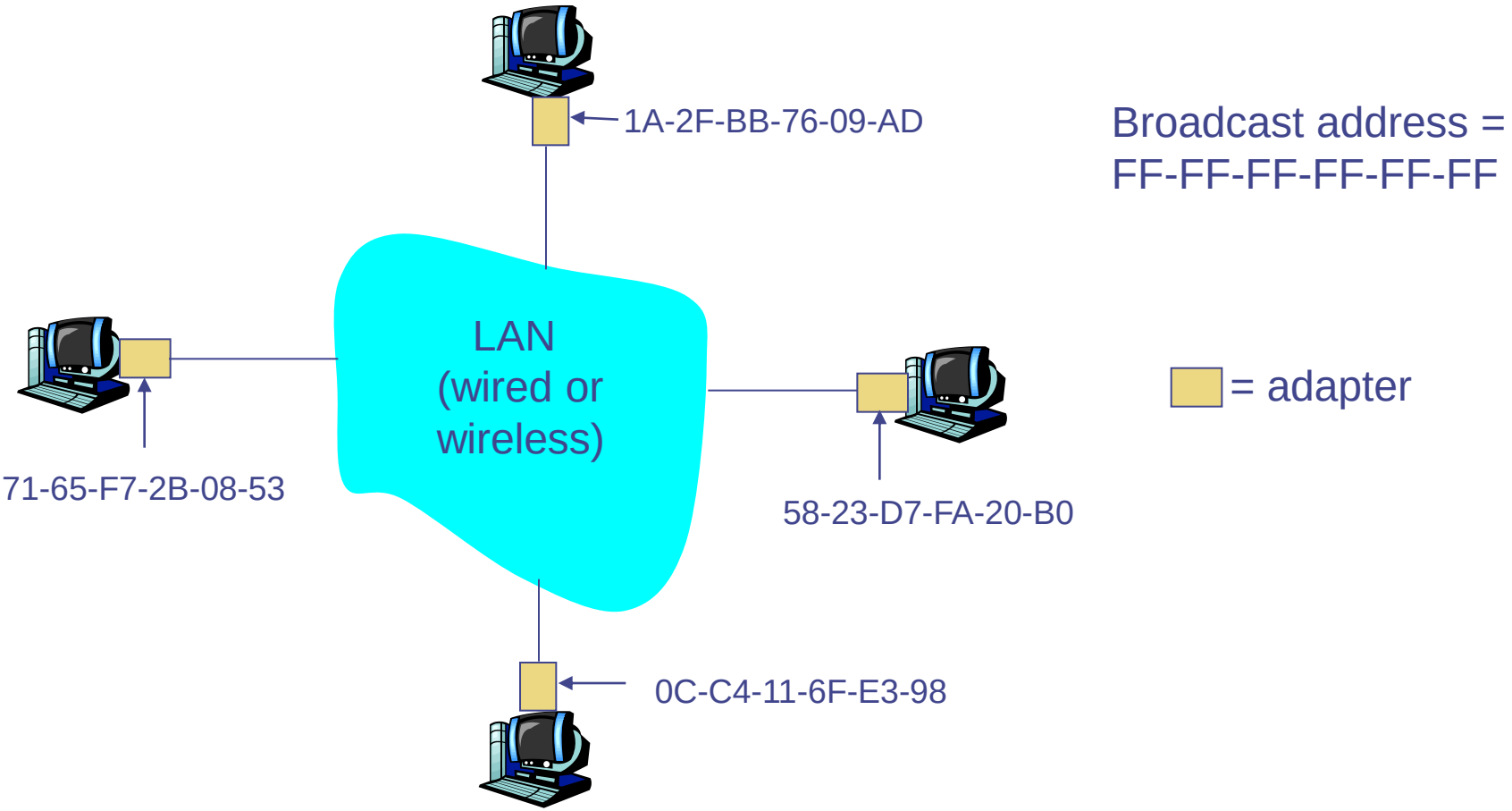
- 32-bit IP address
  - Network-layer address
  - Used to get datagram to destination IP subnet

- MAC (or LAN or physical or Ethernet) address
  - Function: get frame from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs)
    - burned in NIC ROM, also sometimes software settable

- Internet protocols use dynamic assignment for MAC addresses to Internet addresses with ARP (Address Resolution Protocol)

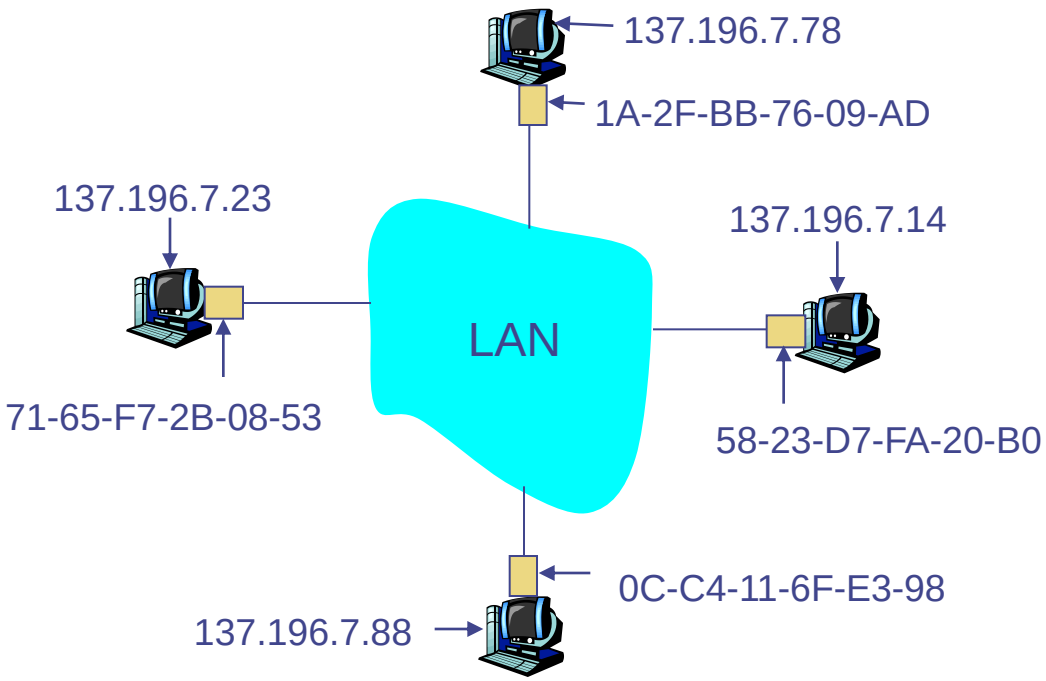
LAN Addresses and ARP

Each adapter on LAN has unique MAC address (also called LAN address)



ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?



- Each IP node (host, router) on LAN has ARP table
- ARP 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)

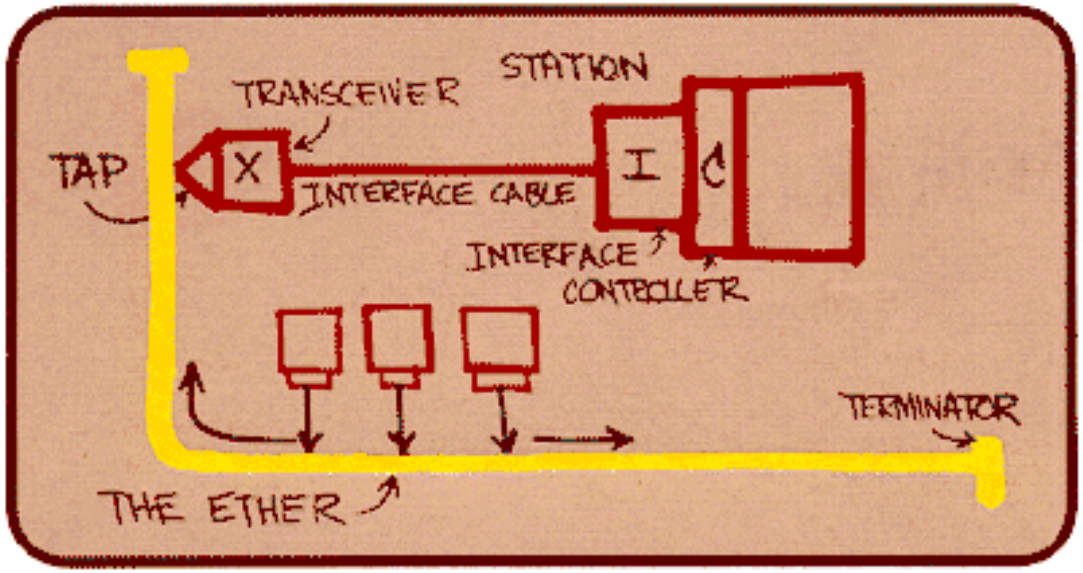
ARP protocol: Same LAN (network)

1. A wants to send datagram to B, and B's MAC address not in A's ARP table
2. A broadcasts ARP query packet, containing B's IP address
   - dest MAC address = FF-FF-FF-FF-FF-FF
   - All machines on LAN receive ARP query
3. B receives ARP packet, replies to A with its (B's) MAC address
   - frame sent to A's MAC address (unicast)
4. 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

Ethernet

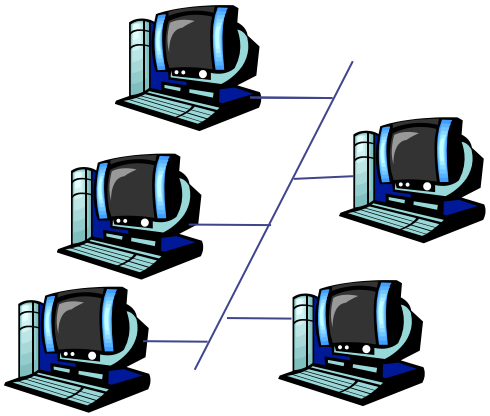
- Dominant wired LAN technology
  - cheap
  - first widely used LAN technology
  - simpler, cheaper than token LANs and ATM
  - Kept up with speed race: 10 Mbps – 10 Gbps



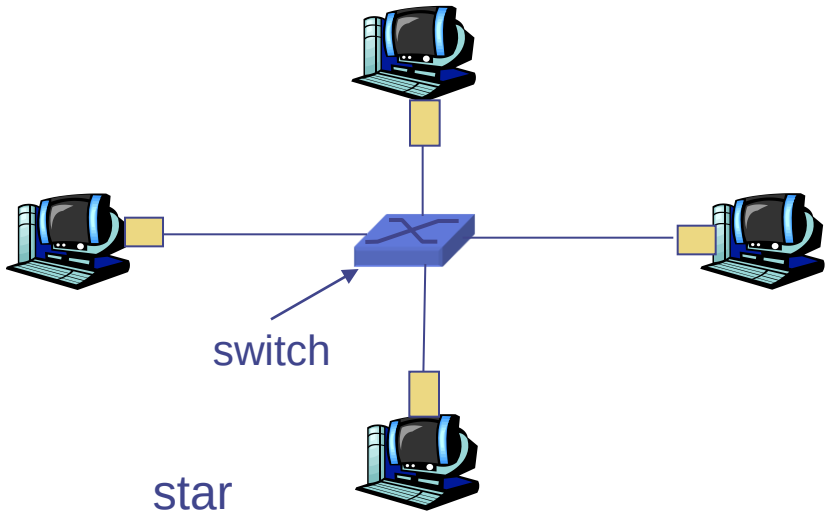
Metcalfe's Ethernet sketch

Star topology

- Bus topology popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- Today: star topology prevails
  - active switch in center
  - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)



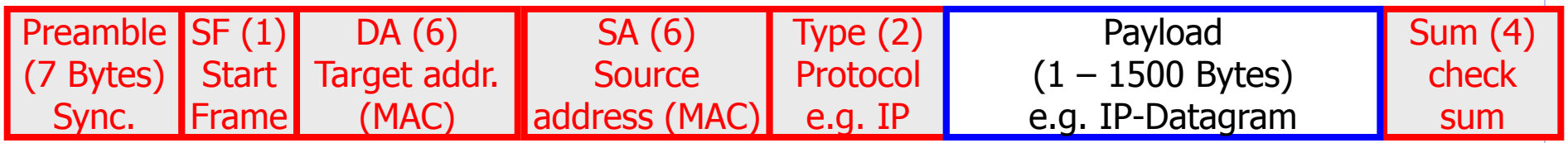
bus: coaxial cable



star

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, used to synchronize receiver, sender clock rates
- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- Type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- CRC: checked at receiver, if error is detected, frame dropped



Ethernet: Unreliable, connectionless

- Connectionless: No handshaking between sending and receiving NICs
- Unreliable: receiving NIC doesn't send acks or nacks to sending NIC
  - stream of datagrams passed to network layer can have gaps (missing datagrams)
  - gaps will be filled if app is using TCP
  - otherwise, app will see gaps
- Ethernet's MAC protocol: unslotted CSMA/CD

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame

2. If NIC senses
   - channel idle
     ⇒ starts frame transmission
   - channel busy
     ⇒ waits until channel idle, then transmits

3. If NIC transmits entire frame without detecting another transmission

   ⇒NIC is done with frame!

4. If NIC detects another transmission while transmitting, aborts and sends jam signal

5. After aborting, NIC enters exponential back-off:
   - after collision nr. X , NIC chooses K at random from {0,1,2,…,2X-1}.
   - NIC waits K·512 bit times
   - NIC returns to Step 2

Ethernet's CSMA/CD (more)

- Jam Signal
  - make sure all other transmitters are aware of collision
- 48 bits time
  - .1 microsec for 10 Mbps Ethernet
  - For K=1023, wait time is about 50 msec
- Exponential Back-off
  - Goal: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- First collision
  - choose K from {0,1};
  - delay is K· 512 bit transmission times
- After second collision
  - choose K from {0,1,2,3}…
- After ten collisions
  - choose K from {0,1,2,3,4,…,1023}

CSMA/CD efficiency

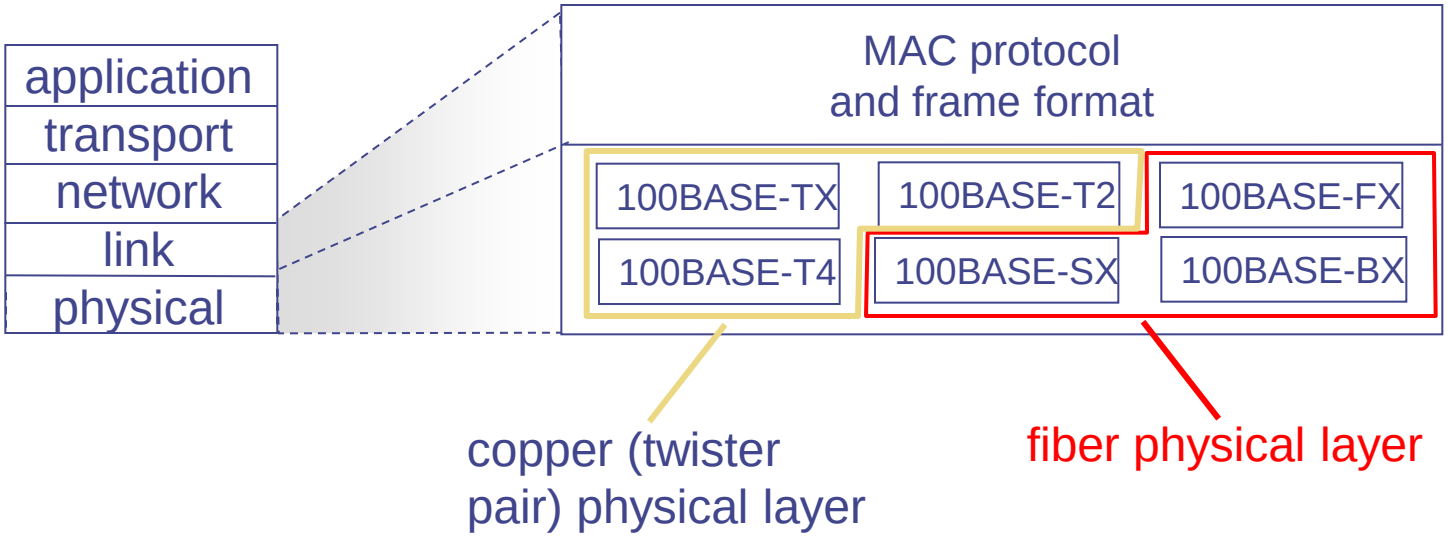
- tprop = max propagation delay between 2 nodes in LAN
- ttrans = time to transmit max-size frame

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

- Efficiency goes to 1
  - as tprop goes to 0
  - as ttrans goes to infinity

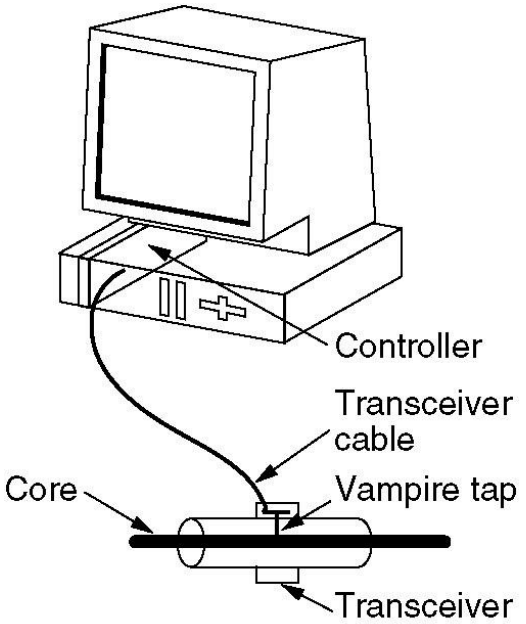
802.3 Ethernet Standards: Link & Physical Layers

- Many different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10Gbps
  - different physical layer media: fiber, cable

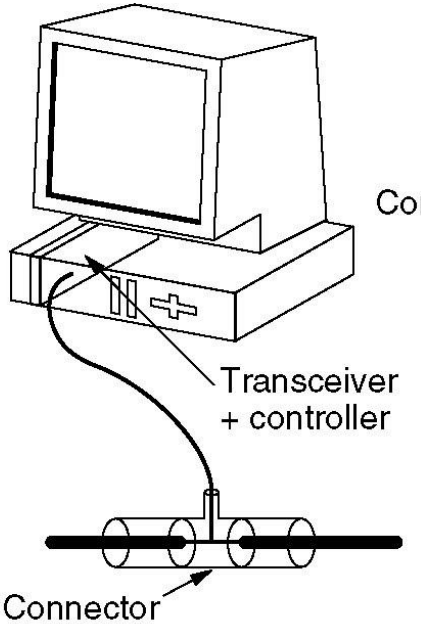


Ethernet cabling

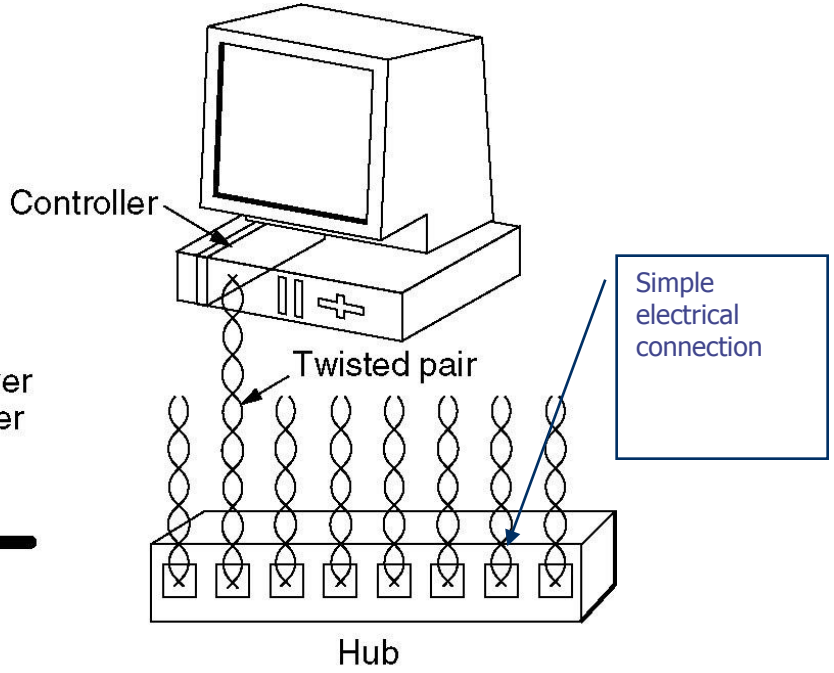
Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings



10Base5



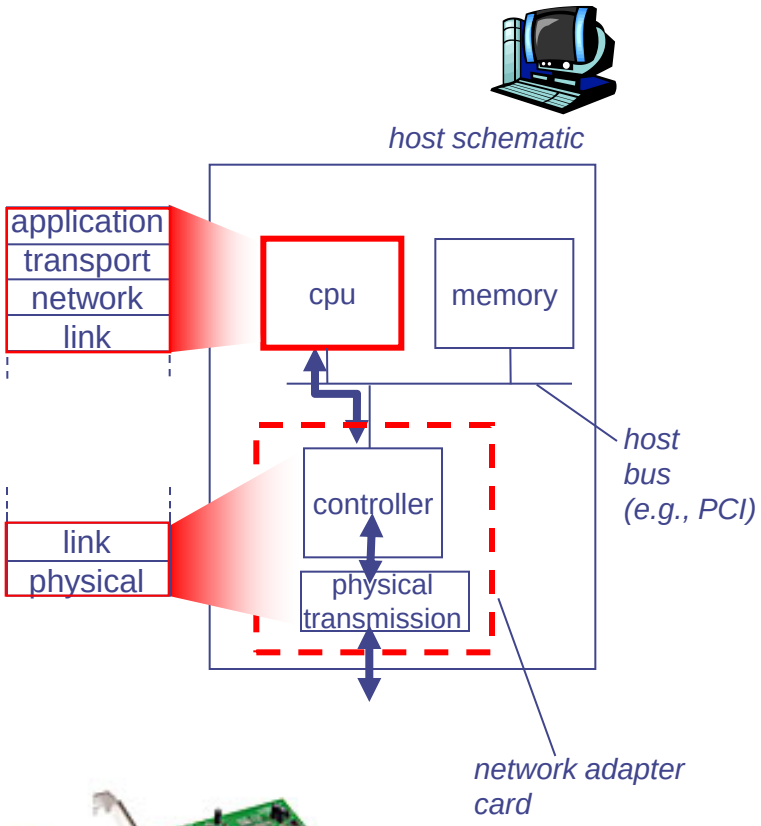
10Base2



10BaseT

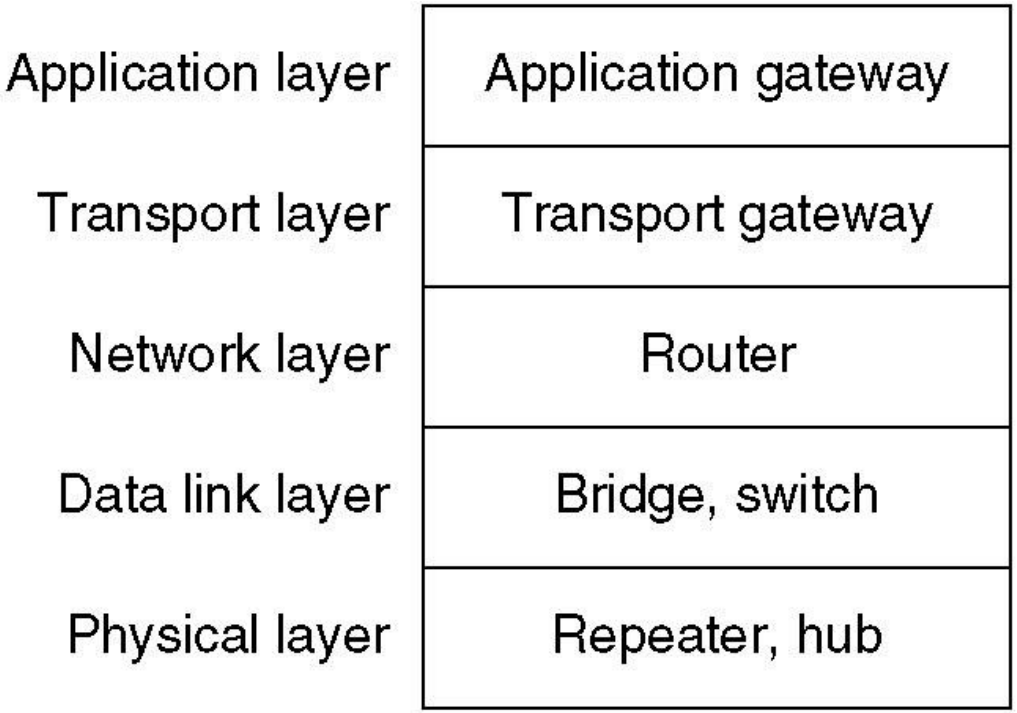
Where is the link layer implemented?

- In every host there is a network interface card (NIC) as “adaptor”
  - Ethernet card
  - PCMCI card
  - 802.11 card
- NIC
  - Implements link, physical layer
  - Attaches into host’s system buses
- Combination of
  - Hardware
  - Software
  - Firmware



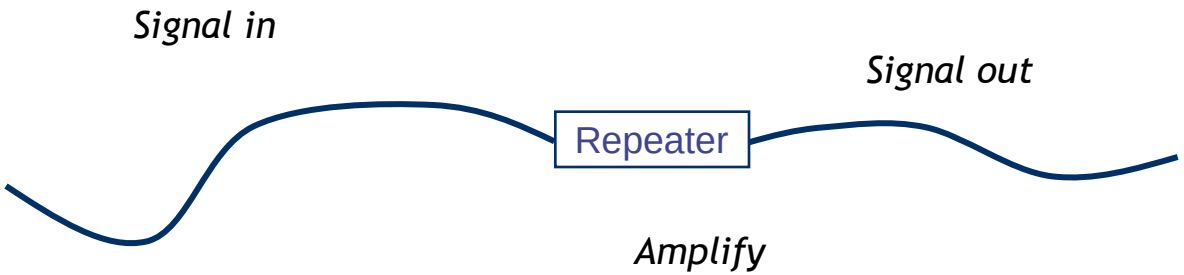
Connecting Elements

- Connecting elements on different layers



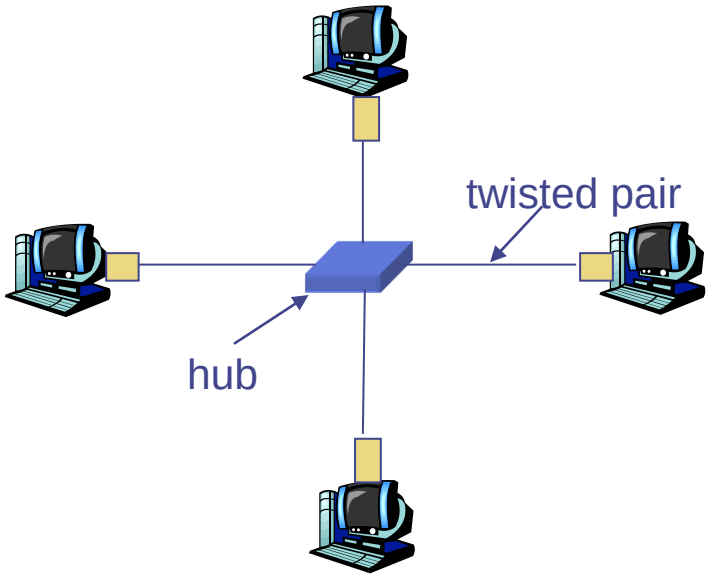
Repeaters

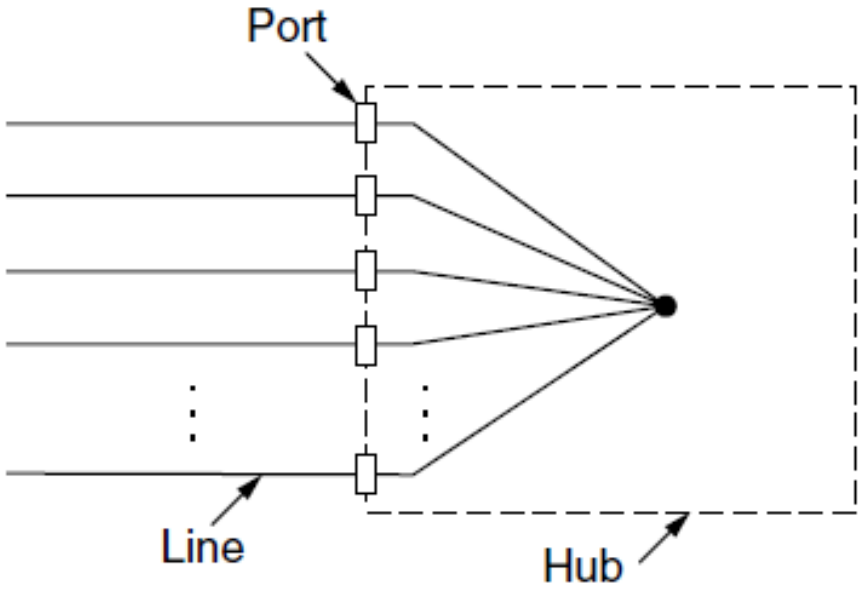
- Simplest option: Repeater
  - Physical layer device
  - Connected to two cables
  - Amplifies signal arriving on either one, puts on the other cable
  - Essentially an analog amplifier to extend physical reach of a cable
  - Combats attenuation
  - Neither understands nor cares about content (bits) of packets



Hubs

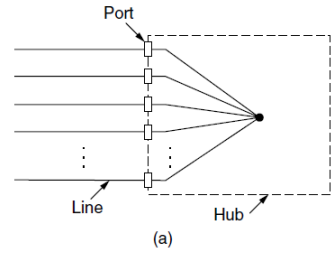
- Physical-layer (“dumb”) repeaters
  - Bits coming in one link go out all other links at same rate
  - All nodes connected to hub can collide with one another
  - No frame buffering
  - No CSMA/CD at hub: host NICs detect collisions

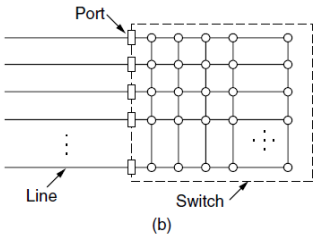


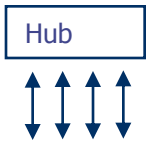


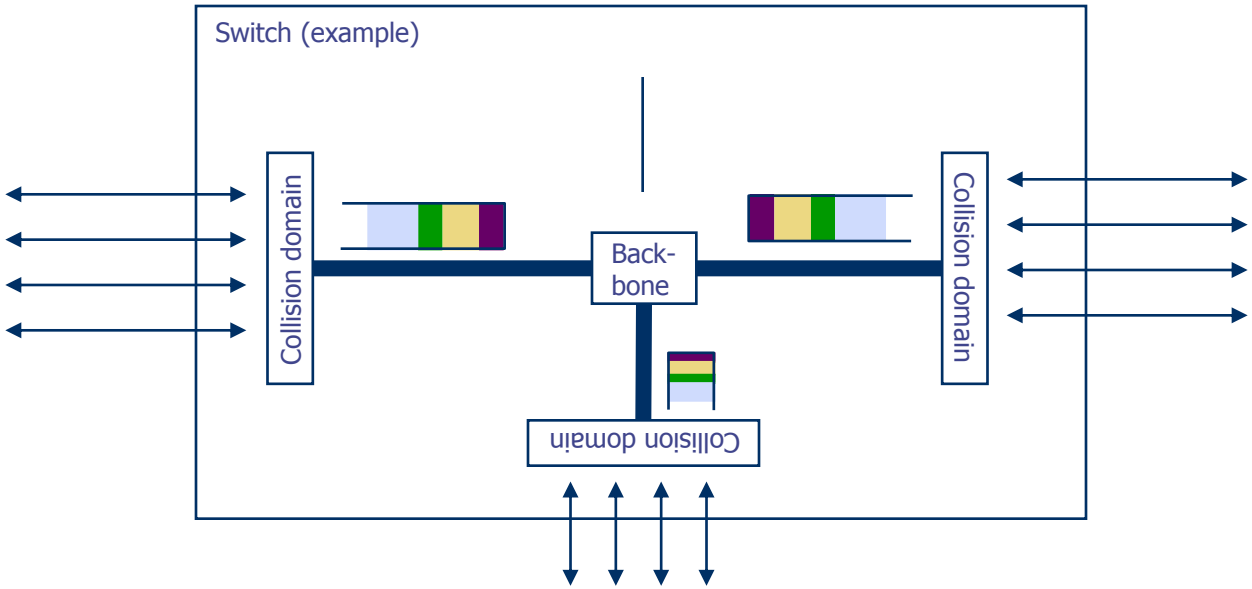
A switch…

- How to exchange packets between different Inputs?
  - Switch contains buffers to intermediately store incoming packets before forwarding them towards their destination
  - Different buffer structures possible: one per incoming link, one per group of links,…
    - Cost issue, mainly



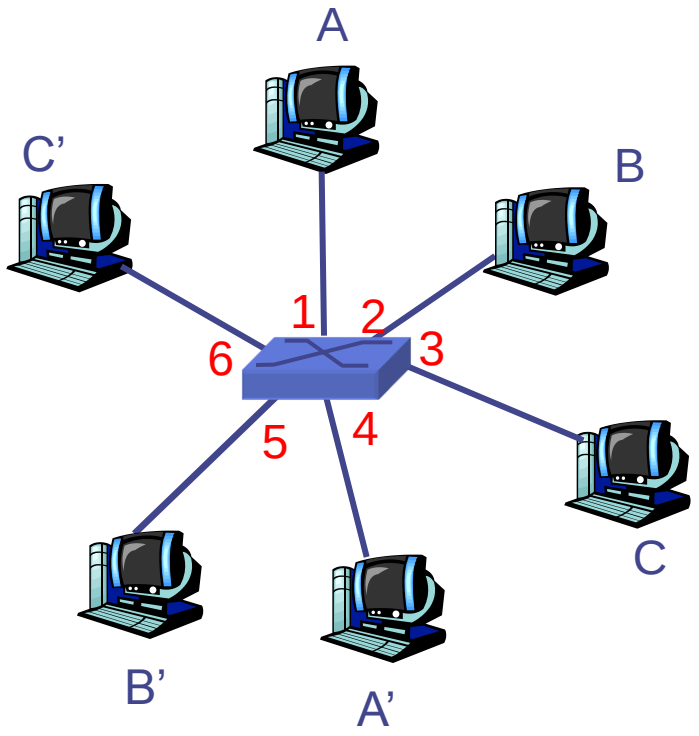






Switch: allows multiple simultaneous transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on each incoming link, but no collisions; full duplex
  - ⇒each link is its own collision domain
- Switching
  - A-to-A' and B-to-B' simultaneously, without collisions
  - Not possible with dumb hub



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

Physical layer solutions not satisfactory

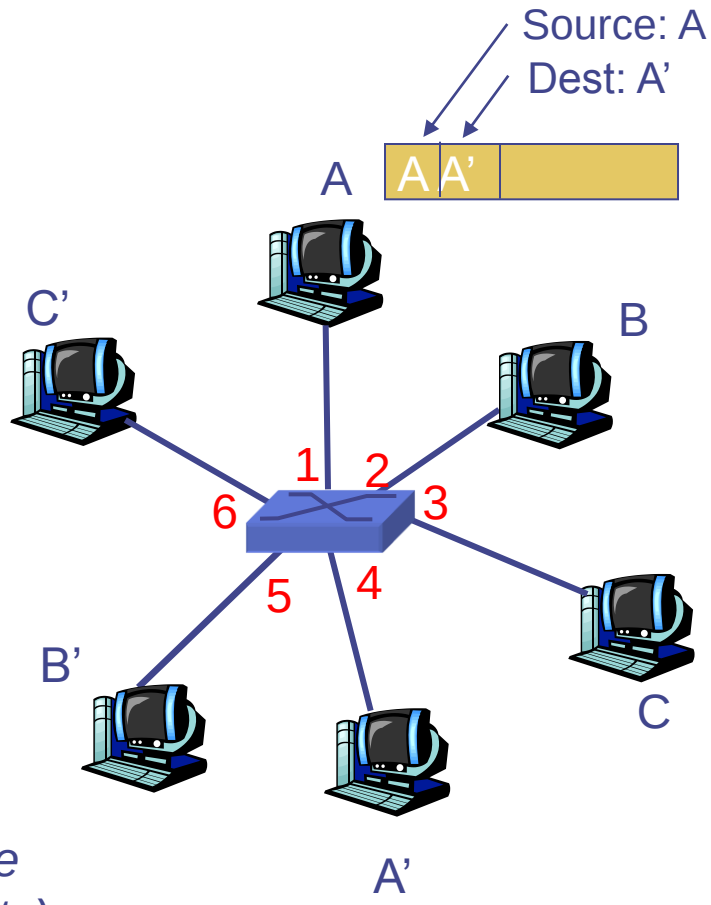
- Physical layer devices – repeater, hub – do not solve the more interesting problems
  - E.g., how to handle load
- Some knowledge of the data link layer structure is necessary
  - To be able to inspect the content of the packets/frames and do something with that knowledge
- Link-layer solutions
  - Bridge & switch
  - Historic distinction
    - Switch: An interconnecting device supporting packet switching.
    - Bridge: Interconnect several networks (Link Layer)
    - Router: A switch in NETWORK LAYER (will be discussed later)
  - *Take care: Occasionally Bridges are referred to as switches ☺*

Bridge

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

Bridge: self-learning

- Bridge learns which hosts can be reached through which interfaces
  - When frame received it “learns” location of sender: incoming LAN segment
  - Records sender/location
  - Pair in table

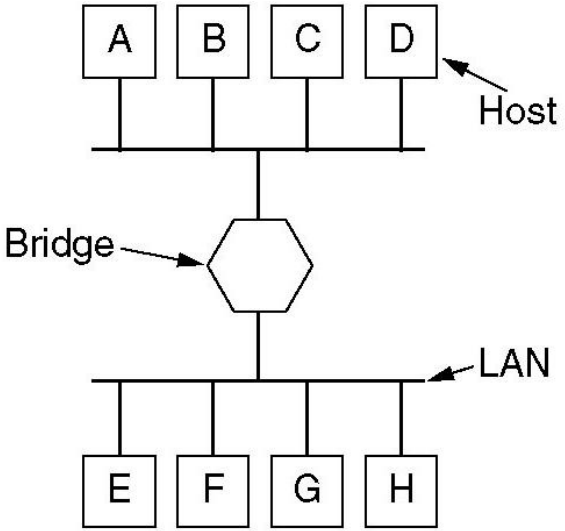


MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

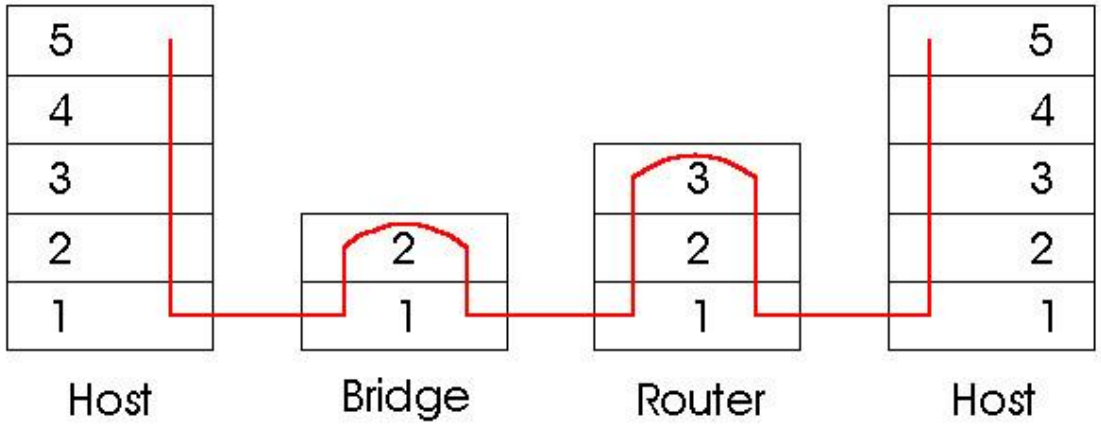
Bridges

- Switches are limited in that they connect simple terminals
- Sometimes, entire networks have to be connected: Bridges
- Bridge also inspects incoming packet and forwards only towards destination
- How to learn here where destination is? Does simple “backward” learning suffice?
- Each network connected to a bridge is a separate collision domain
  - Not possible on physical layer only



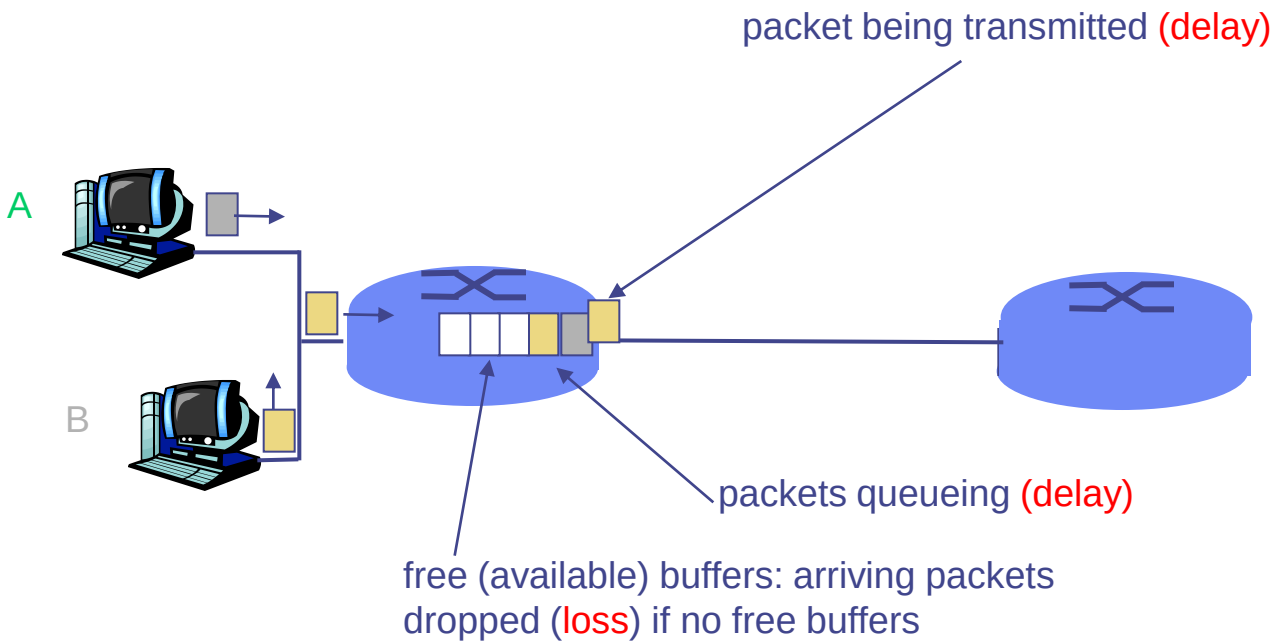
Bridges vs. Routers

- Both store-and-forward devices (switches)
  - Routers: network layer devices (examine network layer headers)
  - Bridges are link layer devices
- Routers maintain routing tables, implement routing algorithms
- Bridges maintain switch tables, implement filtering, learning algorithms



How do loss and delay occur?

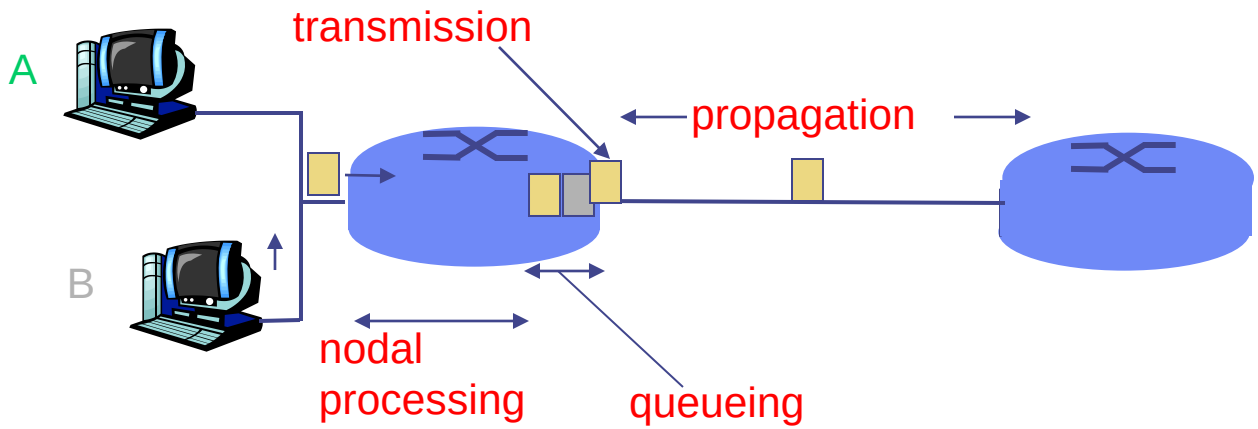
- Packets queue in buffers
- Packet arrival rate to link exceeds output link capacity
- Packets queue, wait for turn



Four sources of packet delay

1. Nodal processing
   - check bit errors
   - determine output link

2. Queuing
   - time waiting at output link for transmission
   - depends on congestion level of router



Delay in packet-switched networks [Kurose-Ross]

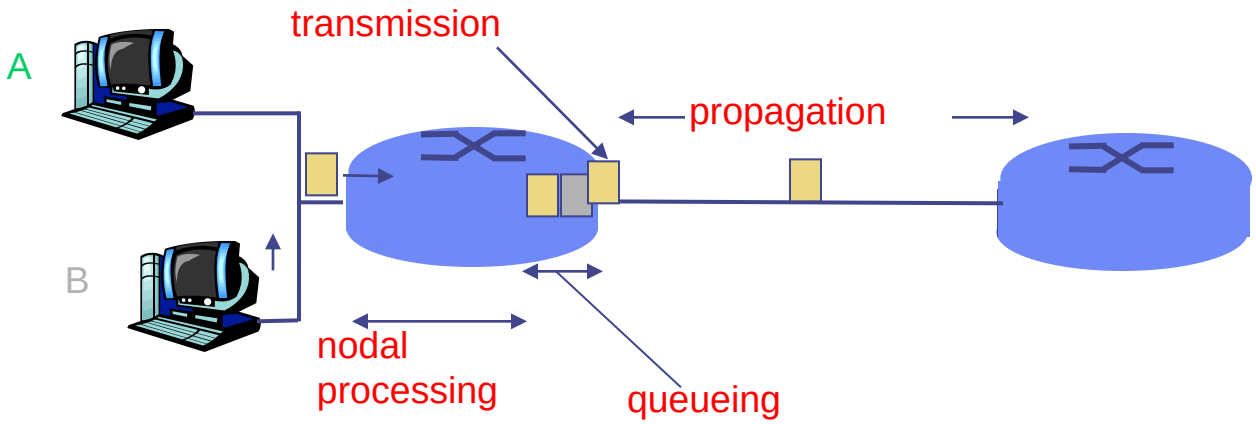
3. Transmission delay

- R=link bandwidth (bps)
- L=packet length (bits)
- time to send bits into link = L/R

4. Propagation delay

- d = length of physical link
- s = propagation speed in medium (~2x108 m/sec)
- propagation delay = d/s

Note: s and R are very different quantities!



Packet Loss ?

- Damaged bits
  - Error in transmission
  - Error in bit synchonization

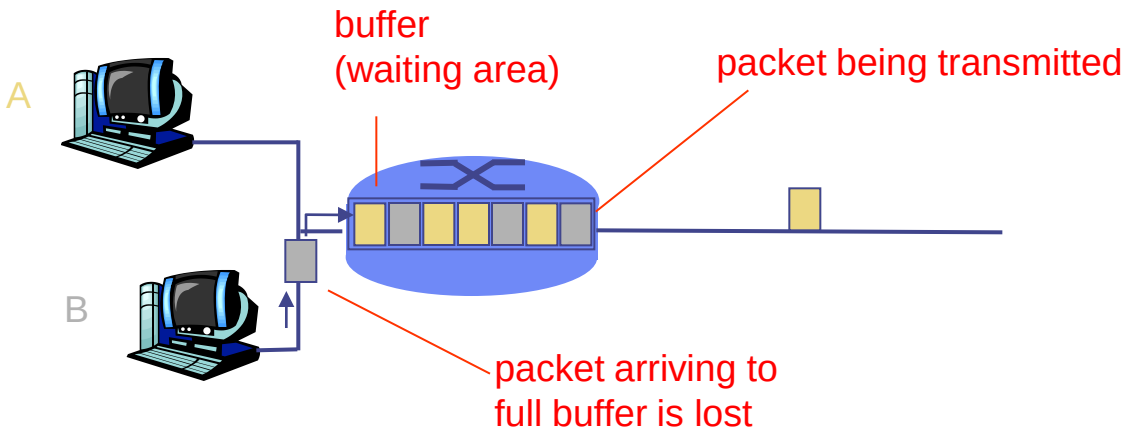
  $\Rightarrow$ detect and skip (e.g. CRC codes)

- Collisions….
  - Lost if no collision detection
  - Retransmission in case of collision detection

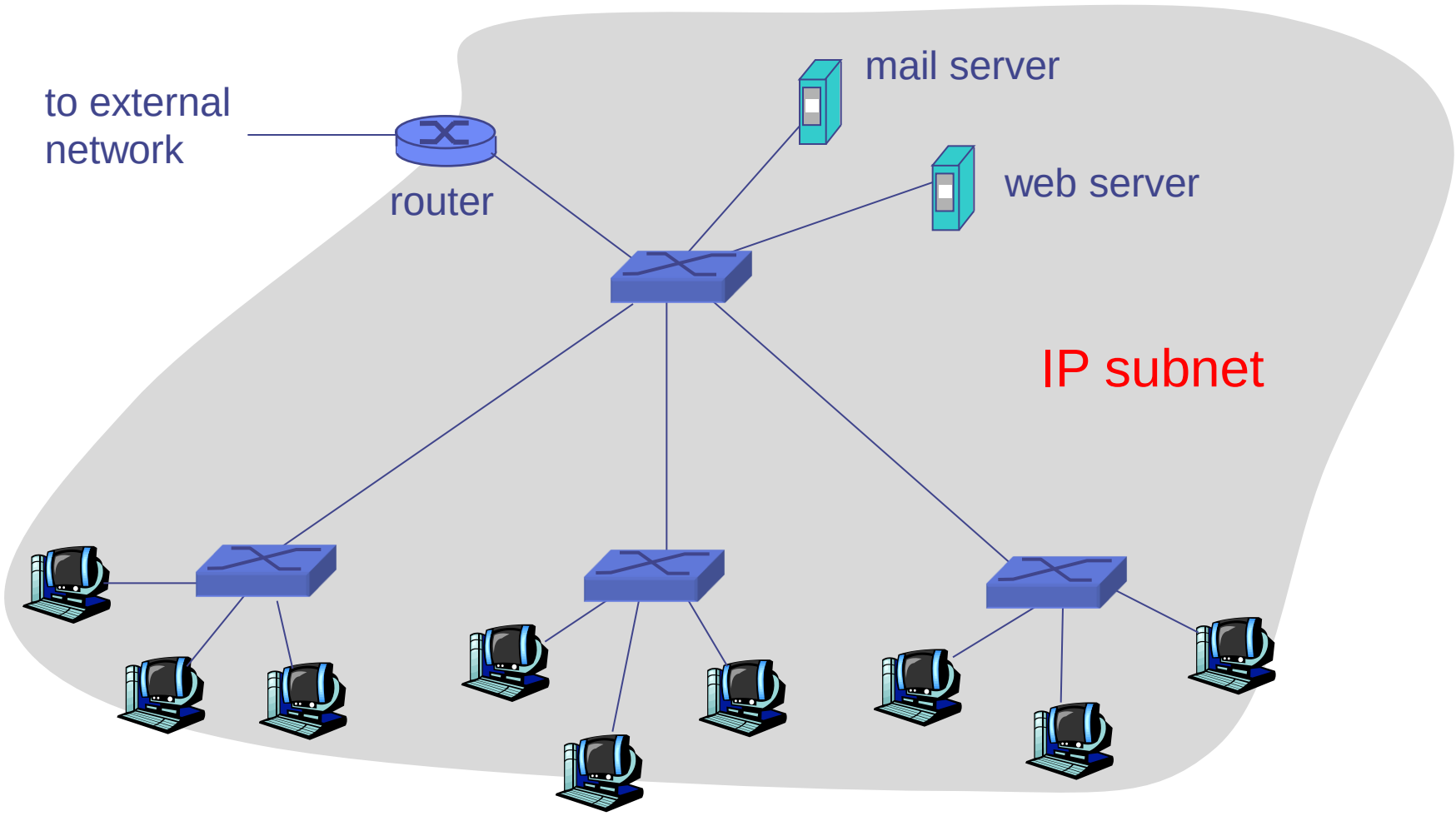
- Switch buffer overflow….

Packet loss

- Queue (aka buffer) preceding link in buffer has finite capacity
- Packet arriving to full queue dropped (aka lost)
- Lost packet may be retransmitted by previous node, by source end system, or not at all

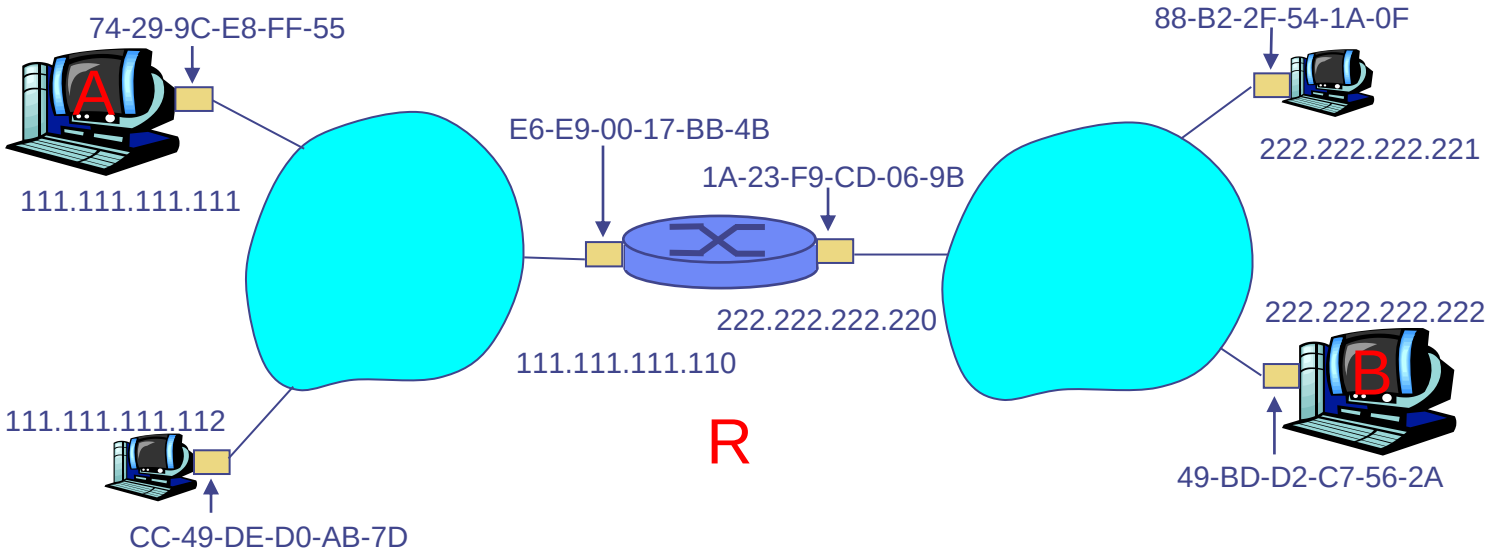


Institutional network



Addressing: routing to another LAN

- Walkthrough
  - send datagram from A to B via R assume A knows B's IP address
  - Two ARP tables in router R, one for each IP network (LAN)



Example

- A creates IP datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP datagram
- A's NIC sends frame, R's NIC receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B