Romain Fontugne

2023 Fall Semester

Information Network Systems

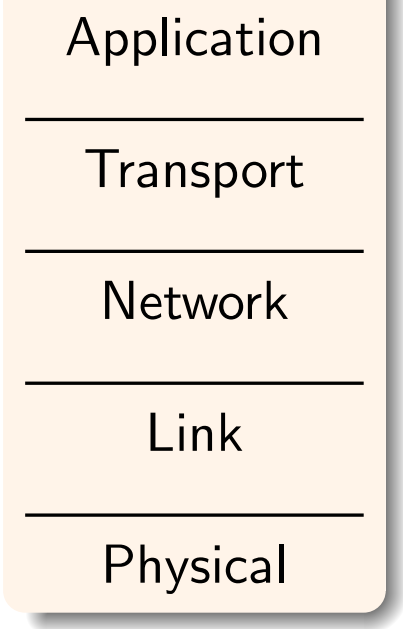
The Link Layer

Last lecture:

Introduction to the Internet

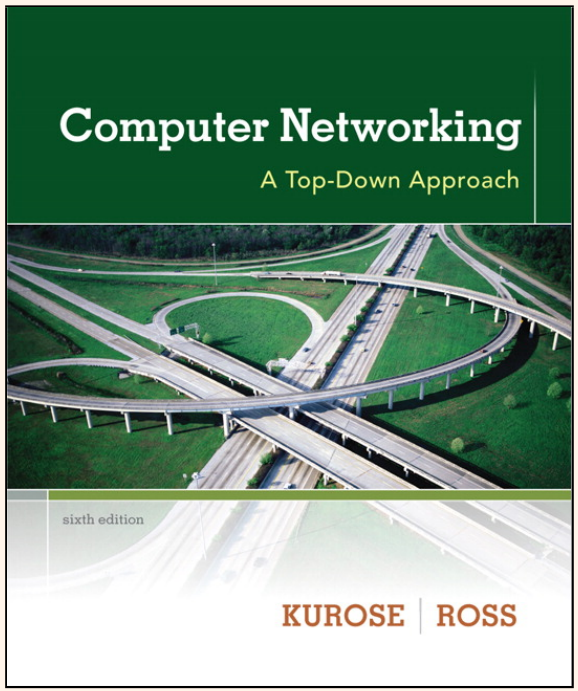
- Internet overview
- network edge
- network core
- packet-switching
- performance: loss, delay, throughput
- protocol, Internet protocol stack
- layering, service models
- history

IP Stack



Today's Lecture: Link layer

1. Introduction, services
2. Error detection, correction
3. Multiple access protocols
4. LANs
   - Ethernet
   - Switches
5. Data center networking



- Chapter 5

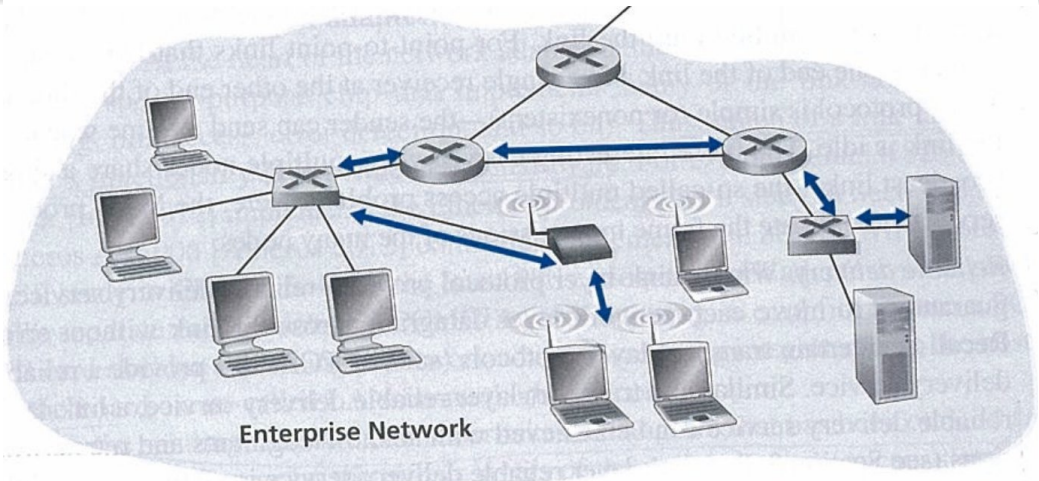
Link layer: Introduction

Role of the link layer

Link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

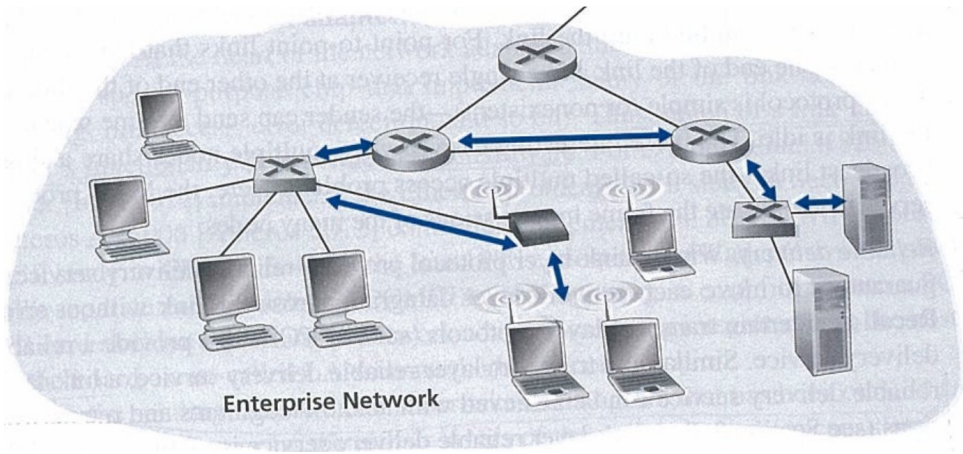
Terminology

- Nodes: hosts, WiFi access points and routers
- Links: communication channels that connect adjacent nodes along communication path (i.e. wired links, wireless links, LANs)
- Frame: layer-2 packet encapsulating datagram (layer-3 packet)



Link layer: Context

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



Transportation analogy

- Trip from Waseda to Princeton:
  Waseda $\xrightarrow{train}$ NRT $\xrightarrow{plane}$ JFK $\xrightarrow{bus}$ Princeton
- tourist = datagram, transportation mode = link layer protocol, transport segment = communication link

Link layer: Services

Framing

- Encapsulate datagram into frame, adding header, trailer
- "MAC" addresses used in frame headers to identify source, dest. (different from IP address!)

Link access

- Media Access Control (MAC): Channel access if shared medium

Reliable delivery between adjacent nodes

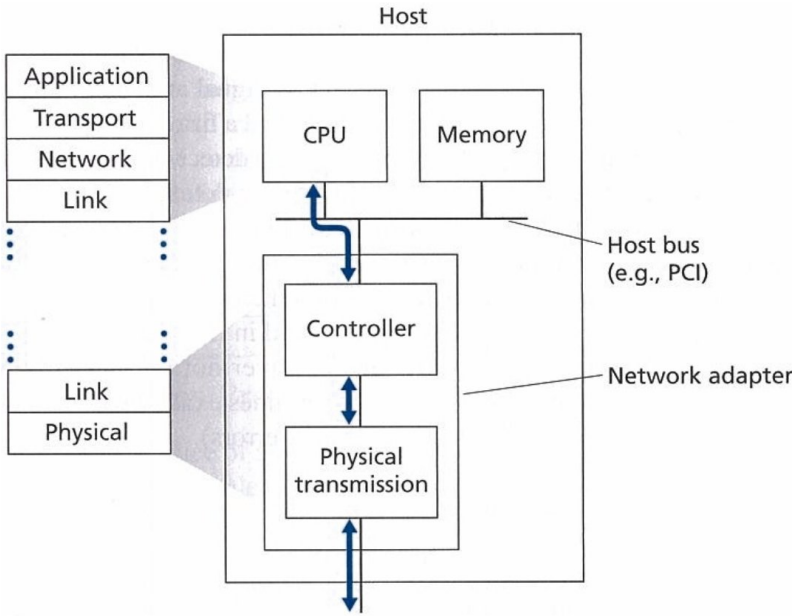
- Seldom used on low bit-error link (fiber, some twisted pair)
- Used in wireless links: high error rates

Error detection and correction

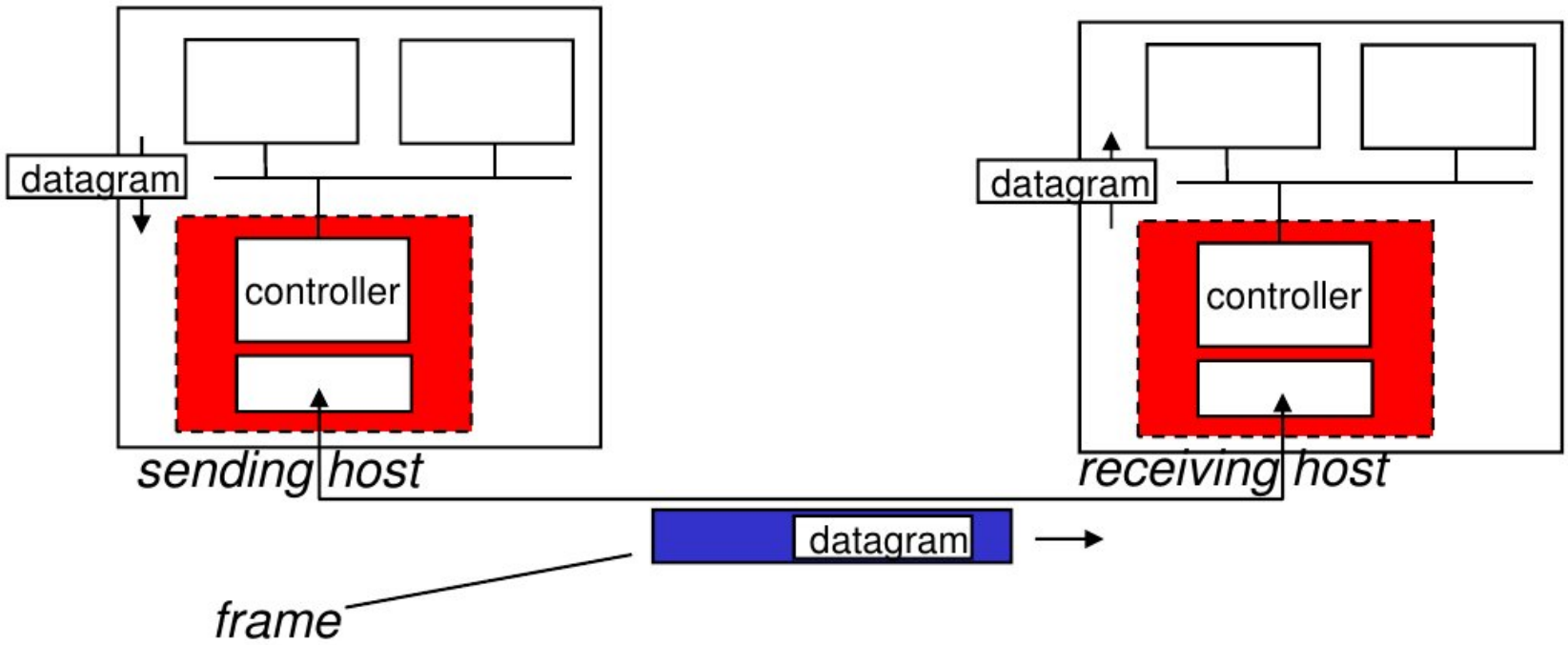
- Errors caused by signal attenuation, noise
- Receiver detects presence of errors
- Correct bit error(s) without resorting to retransmission

Where is the link layer implemented?

- In each and every host
- Link layer implemented in "adaptor" (aka network interface card NIC) e.g. Ethernet card, 802.11 card
- Most functionality implemented in hardware
- Partly also implemented in software executed on host cpu (assembling link-layer addressing information, activate controller hardware)
- Link layer is the place in the protocol stack where software meets hardware



Adaptors Communicating



Sender

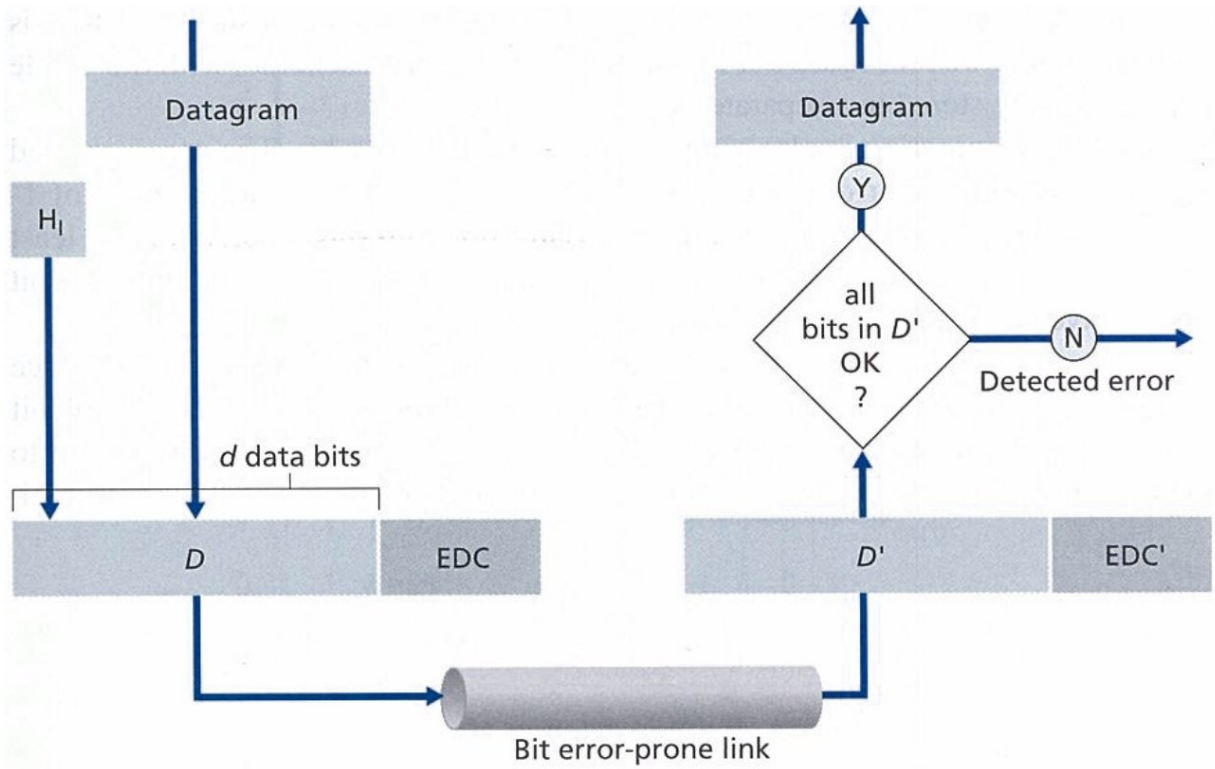
- encapsulates datagram in frame
- adds error checking bits, reliable delivery, flow control,etc.

Receiver

- looks for errors, reliable delivery, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

Error detection

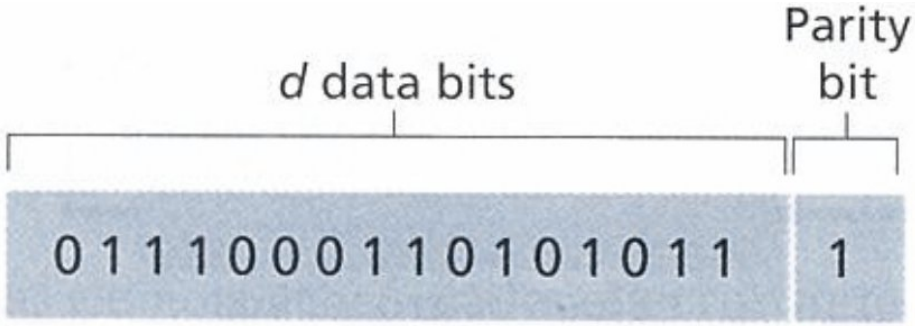
- EDC: Error Detection and Correction bits (redundancy)
- D: Data protected by error checking, may include header fields
- Error detection not 100% reliable!
  - protocol may miss some errors
  - larger EDC field yields better detection and correction



Parity Checks

Single parity bit:

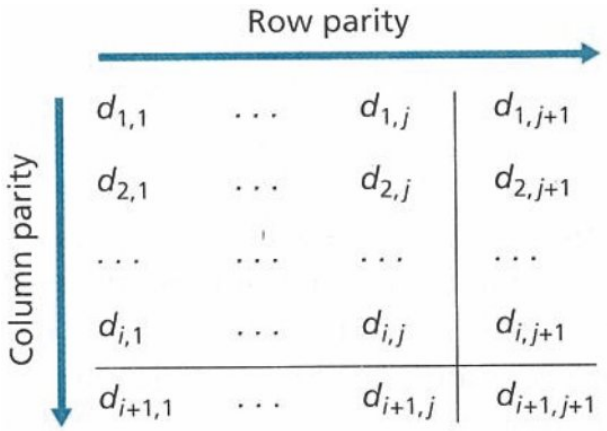
- Simplest form of error detection
- Add one additional bit such that the sum of all bits is even
- ✓ Able to detect odd number of bit errors
- ✗ Blind to even amount of errors

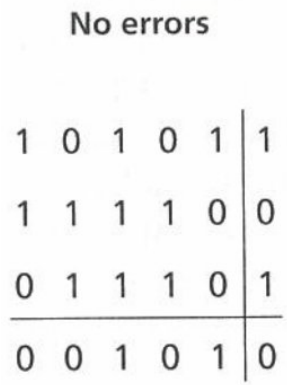


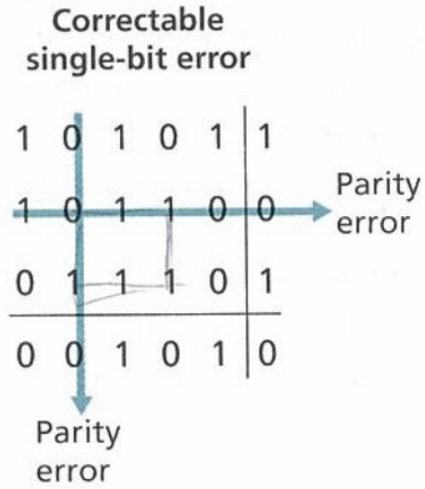
Parity Checks

Two-dimensional bit parity:

- Two dimensional parity scheme to improve detection ability
- Correction of erroneous bits possible
- Can detect and correct any combination of up to two erroneous bits
- Ability of the receiver to both detect and correct errors is called Forward Error Correction (FEC)







Internet Checksum

Bytes of data are treated as 16-bit integers

- the 1s-complement of their sum is treated as checksum
- receiver detects an error when the sum of the checksum and the 16-bit words contain any 0 bit
- used at transport layer only

Example with three 16-bit words

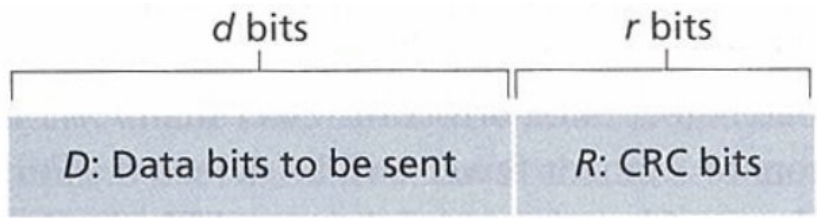
```
0010011001100000  First 16-bit word
0101010101010101  Second 16-bit word
----------------
0111101110110101  Sum of the first two 16-bit words
0010110100001100  Third 16-bit word
----------------
1010100011000001  Sum of all three 16-bit words
----------------
0101011100111110  1-complement of the sum
```

Cyclic Redundancy Check

CRC: Widely used error detection technique

- e.g Ethernet, 802.11 WiFi
- view data bits, D, as a binary number
- Sender and receiver agree on an $r+1$ bit pattern (generator), G
- goal: choose r bits, R, such that
  - $D \cdot 2^r \oplus R$ exactly divisible by G (modulo 2)
  - receiver knows G, divides $D \cdot 2^r \oplus R$ by G
  - If non-zero remainder: error detected!
  - $\rightarrow$ can detect all burst errors less than $r+1$ bits

Bit pattern:



Mathematical formula: $D \cdot 2^r \oplus R$
($\oplus$ means XOR)

CRC calculation

Find R?

- such as
  $D \cdot 2^r \oplus R = nG$
- $R = remainder \frac{D \cdot 2^r}{G}$
  (see textbook/wikipedia)
- $\rightarrow$ Easily implemented in hardware by bit-shift and XOR operators

Example:

$D = 11010011101100$, $r = 3$, $G = 1011$

```
11010011101100 000 <--- D right padded by 3 bits
1011               <--- divisor, G
01100011101100 000 <--- result
 1011              <--- divisor G
00111011101100 000 <--- result ...
  1011
00010111101100 000
   1011
00000001101100 000
       1011
00000000110100 000
        1011
00000000011000 000
         1011
00000000001110 000
          1011
00000000000101 000
           101 1
-----------------
00000000000000 100 <--- remainder (3 bits), R
```

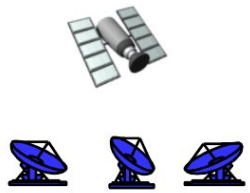
Multiple access links

Two types of links:

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch, host
- 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)



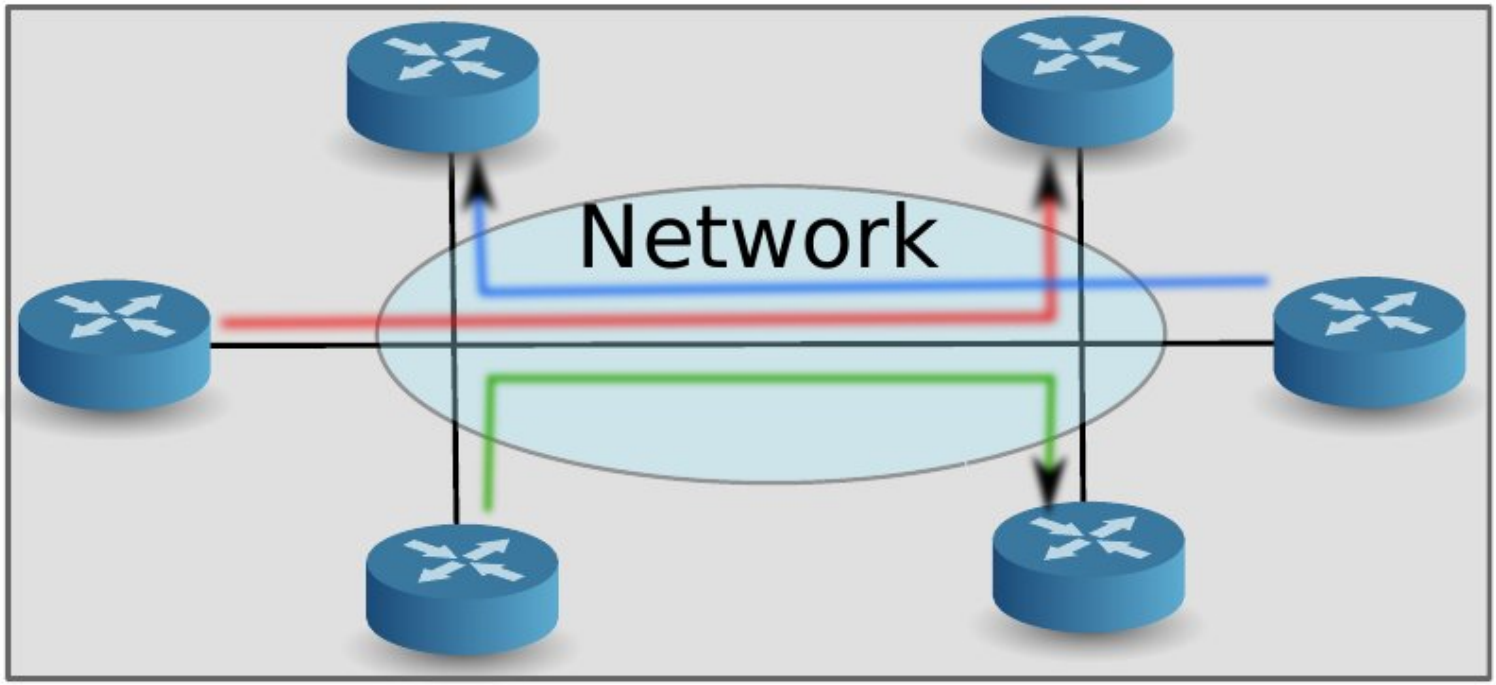
shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple access links

Single shared broadcast channel

- Two or more simultaneous transmissions by nodes: collision
- Normally, when a collision occurs, no receiver can use the data
- How to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel?



Multiple access protocol

An Ideal Multiple Access Protocol

Ideally a multiple access protocol for a broadcast channel of rate R bps should have the following characteristics:

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

Multiple Access Protocol: Taxonomy

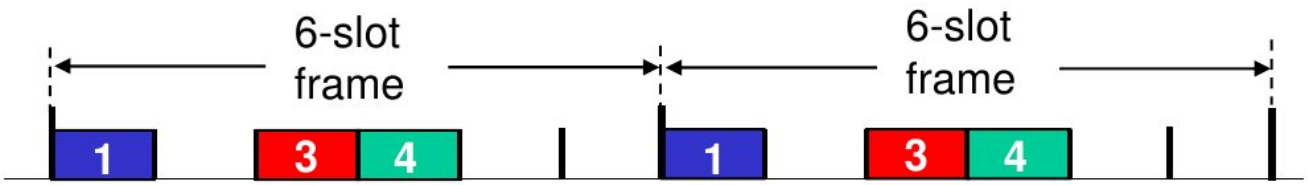
Three broad classes

- channel partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- random access
  - channel not divided, allow collisions
  - "recover" from collisions
- "taking turns"
  - nodes take turns, but nodes with more to send can take longer turns

Channel partitioning protocols: TDM

TDM: time division multiplexing

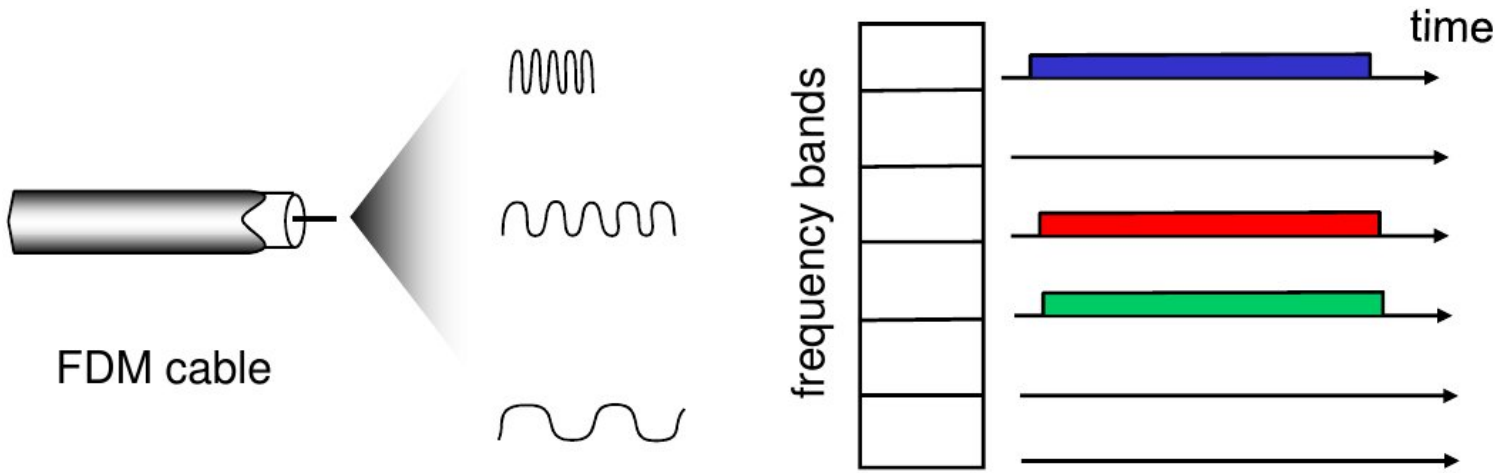
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel partitioning protocols: FDM

FDM: frequency division multiplexing

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



Random Access Protocols

Principles

- when node has packet to send
  - transmit at full channel data rate R
  - no a priori coordination among nodes
- two or more transmitting nodes $\rightarrow$ collision!
- random access protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)

Example of random access protocols

- slotted ALOHA
- ALOHA
- CSMA, CSMA/CD, CSMA/CA

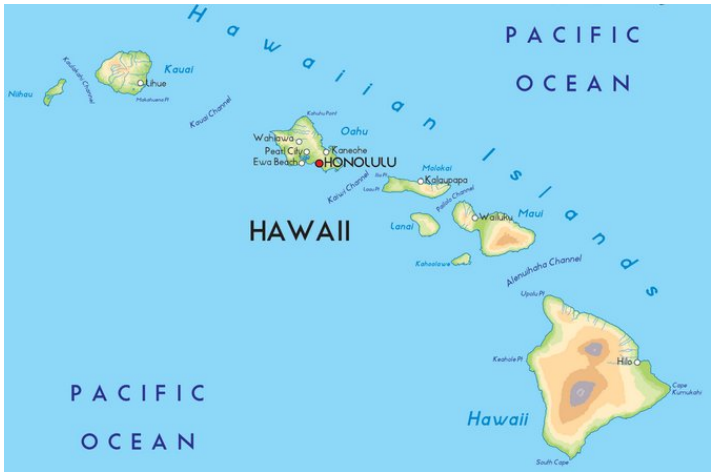
Simple Random Access Protocol: Slotted ALOHA

Assumptions:

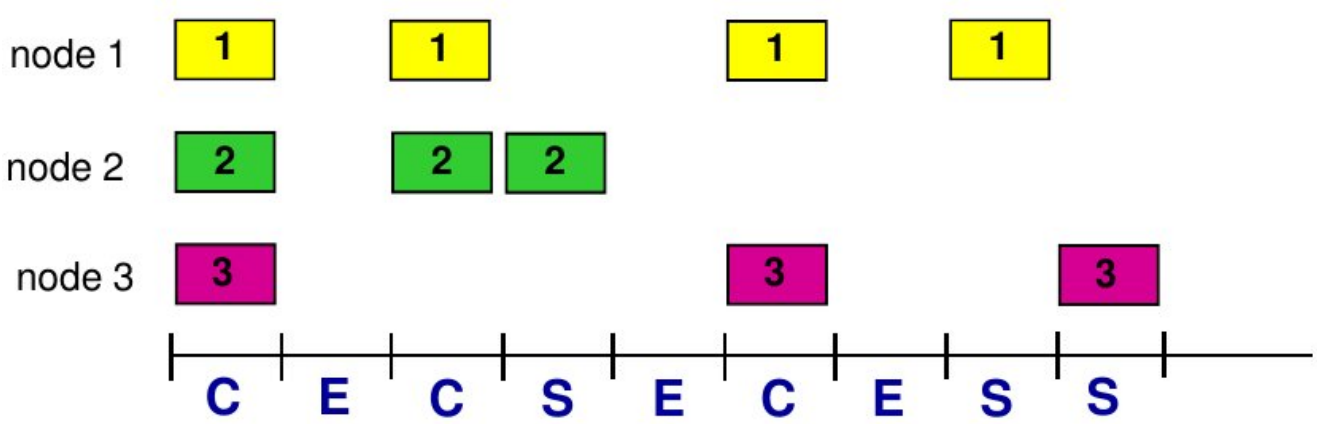
- all frames/packets same size
- nodes are synchronized
- time divided into equal size slots
  (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation:

- when node has a fresh frame to send, transmits in next slot
  - if no collision: successful transmission!
  - if collision: node retransmits frame in each subsequent slot with prob. p until success



Slotted ALOHA



C=Collision, E=Empty, S=Successful

✓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
- clock synchronization

Slotted ALOHA: efficiency

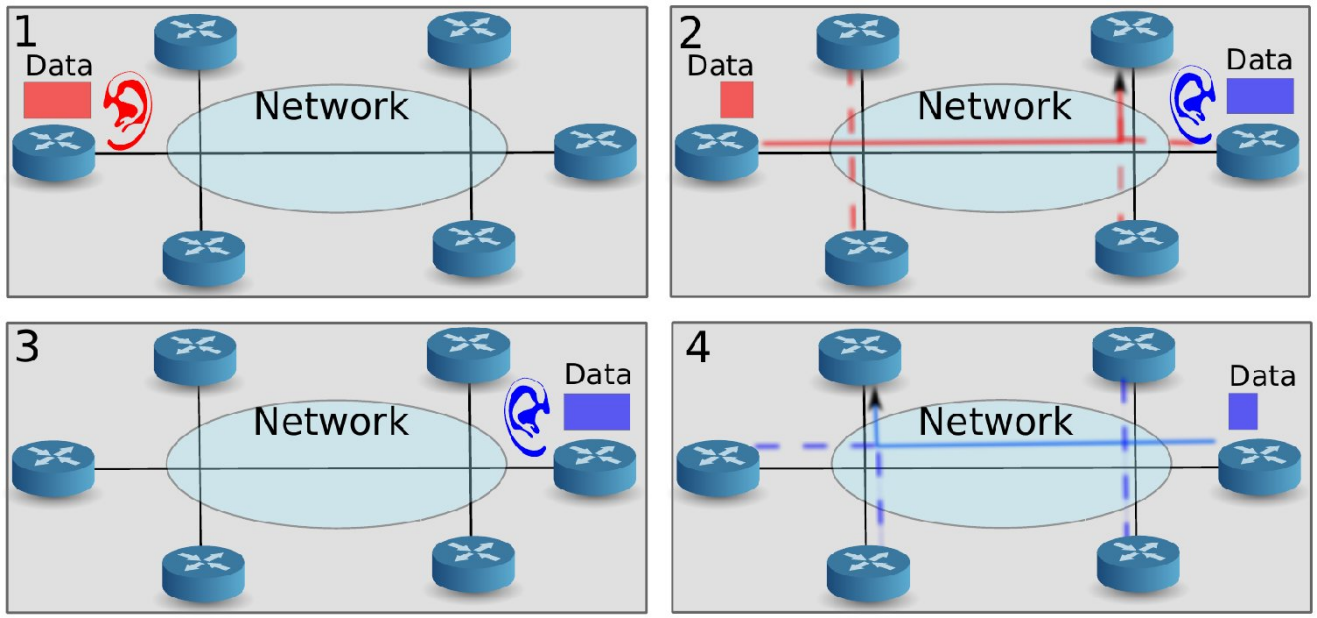
What is the effective transmission rate?

- Efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)
- suppose: N nodes with many frames to send, each transmits in slot with probability p
- prob. that given node has success in a slot $= p(1-p)^{N-1}$
- prob. that any node has a success $= Np(1-p)^{N-1}$
- max efficiency: find $p^{\star}$ that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^{\star}(1-p^{\star})N-1$ as N goes to infinity, gives: $maxefficiency = 1/e = .37$
- with the best p: successful transmissions happen 37% of time!

CSMA (carrier sense multiple access)

CSMA: listen before transmit:

- Before transmitting, a node will listen (sense) to the channel
  - if channel sensed idle: transmit entire frame
  - if channel sensed busy: defer transmission
- human analogy: don't interrupt others!



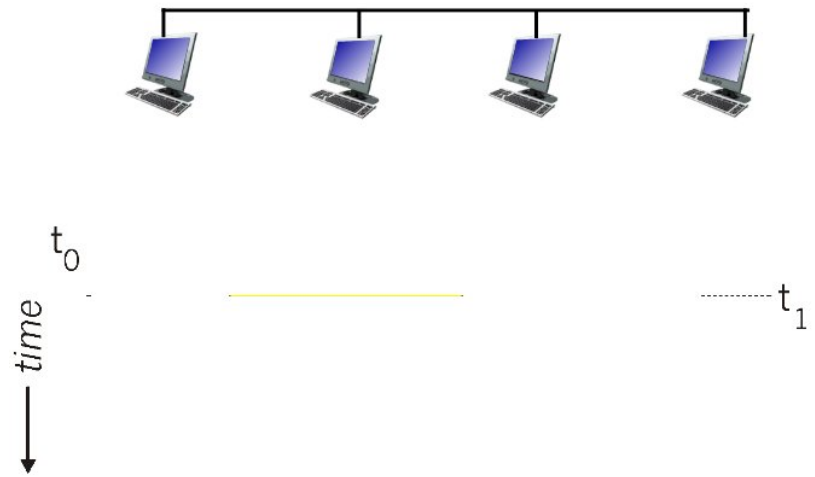
CSMA collisions

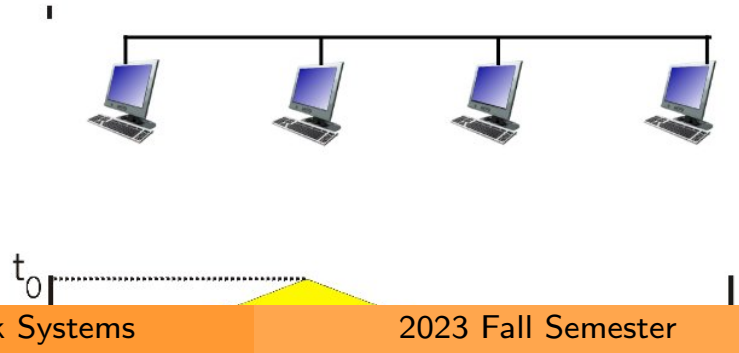
collisions can still occur

- propagation delay means two nodes may not hear each other's transmission

collision: entire packet transmission time wasted

- distance & propagation delay play role in in determining collision probability





CSMA/CD (collision detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- Sense the channel while transmitting a frame
- collisions detected within short time
- colliding transmissions aborted, reducing channel wastage

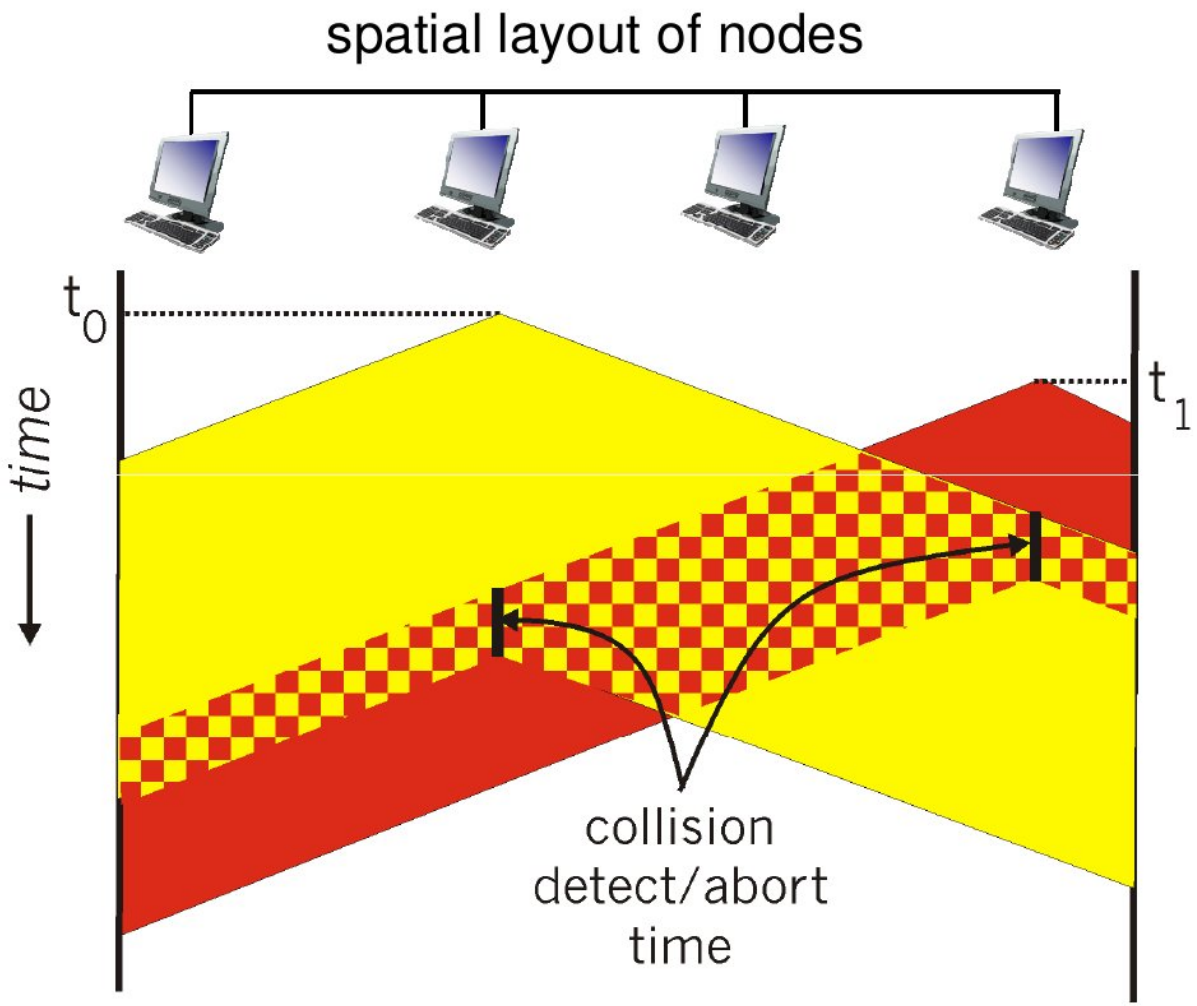
Collision detection:

- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

Human analogy:

the polite conversationalist

CSMA/CD (collision detection)



Ethernet CSMA/CD algorithm

CSMA/CD implementation in Ethernet:

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If NIC senses 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 binary (exponential) backoff:
   - after mth collision, NIC chooses K at random from 0,1,2,..., 2m-1. NIC waits K·512 bit times, returns to Step 2
   - longer backoff interval with more collisions

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!

"Taking turns" MAC protocols

channel partitioning MAC protocols:

- share channel efficiently and fairly at high load
- inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!

random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

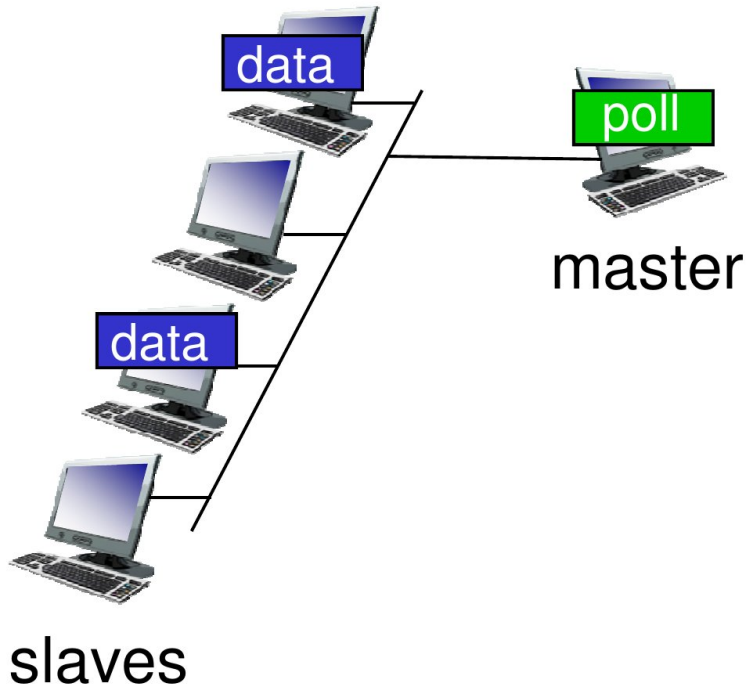
"taking turns" protocols

look for best of both worlds!

“Taking turns” MAC protocols

Polling

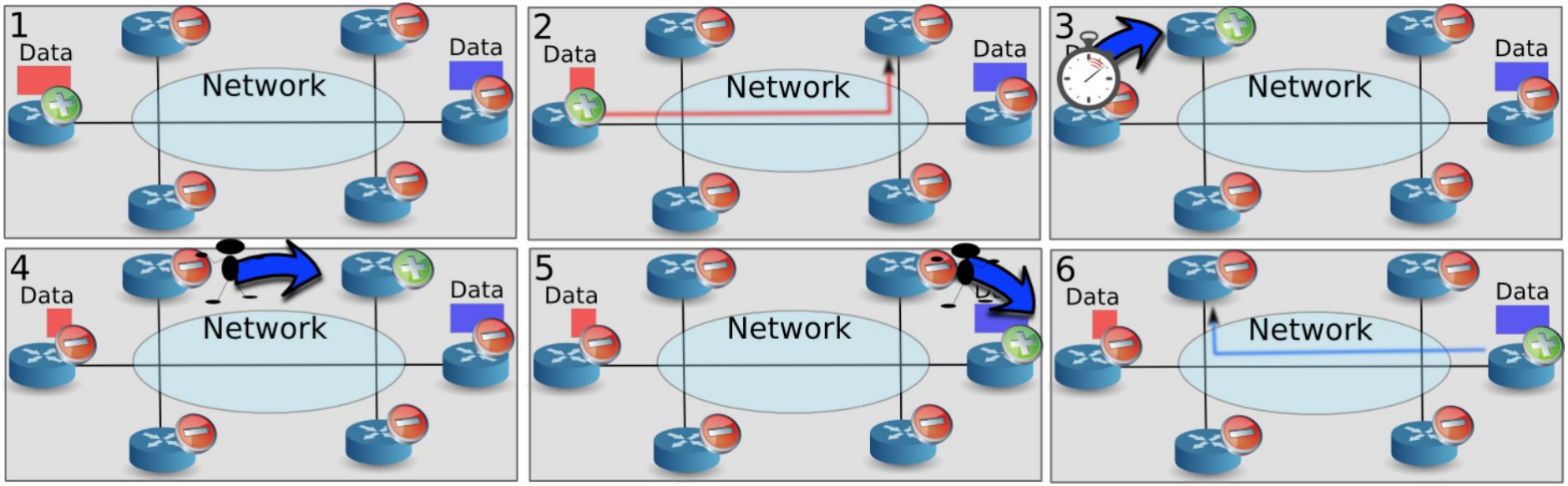
- master node “invites” slave nodes to transmit in turn
- typically used with “dumb” slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)
- e.g. Bluetooth, 802.15



"Taking turns" MAC protocols

Token passing

- control token passed from one node to next sequentially
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Summary of MAC protocols

Channel partitioning

- Time Division, Frequency Division

Random access (dynamic)

- ALOHA, S-ALOHA, CSMA, CSMA/CD
- carrier sensing: easy in some technologies (wire), hard in others (wireless)
- CSMA/CD used in Ethernet
- CSMA/CA used in 802.11

Taking turns

- polling from central site, token passing
- bluetooth, FDDI, token ring

Ethernet

"Dominant" wired LAN technology:

- cheap $20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs (e.g. FDDI) and ATM
- kept up with speed race: 10 Mbps – 10 Gbps

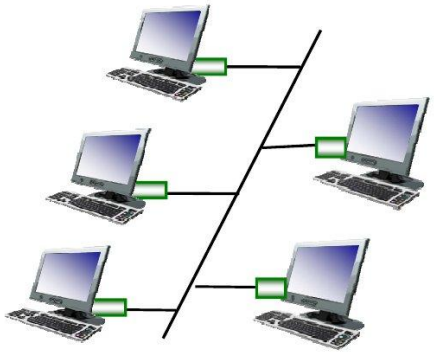
Ethernet: physical topology

Bus: popular through mid 90s

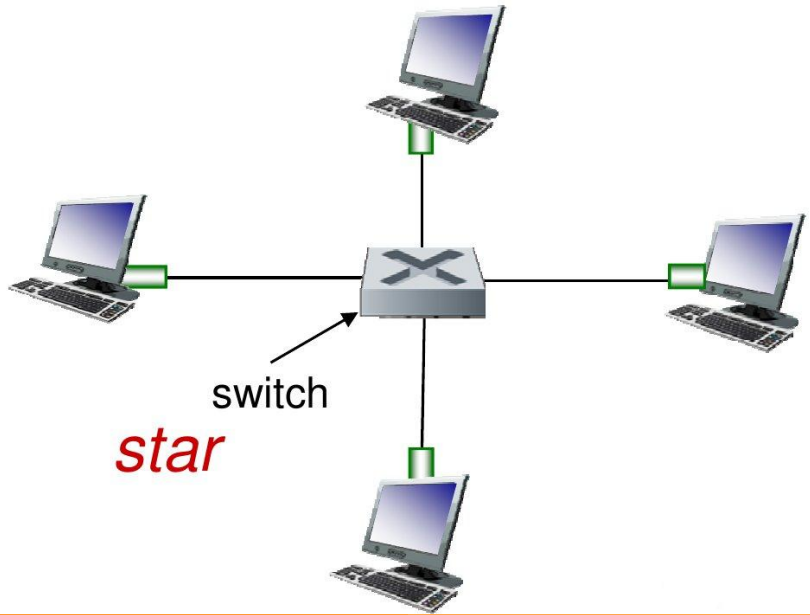
- all nodes in same collision domain (can collide with each other)

Star: prevails today

- active switch in center
- each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



bus: coaxial cable

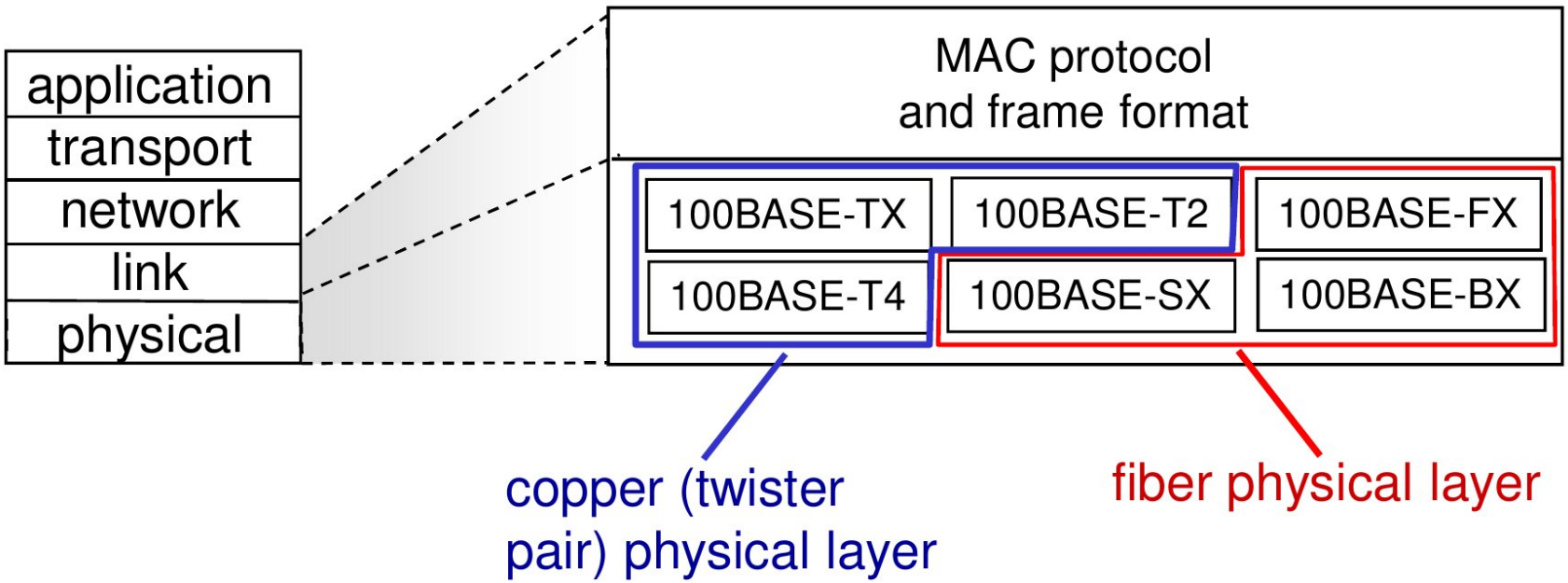


star

802.3 Ethernet standards: link & physical layers

Many different Ethernet standards

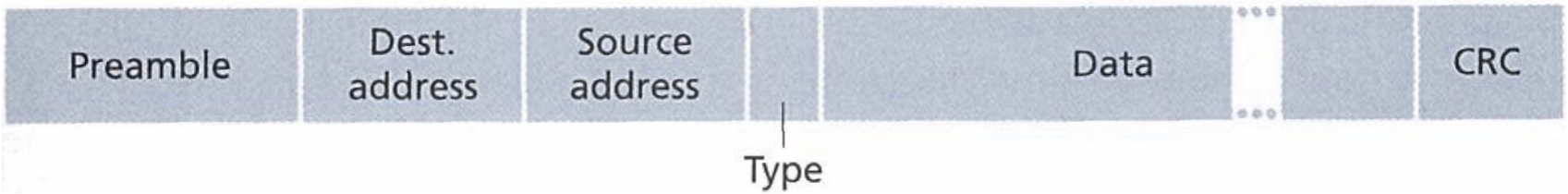
- common MAC protocol and frame format
- different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
- different physical layer media: fiber, cable



Ethernet frame structure

Ethernet frame

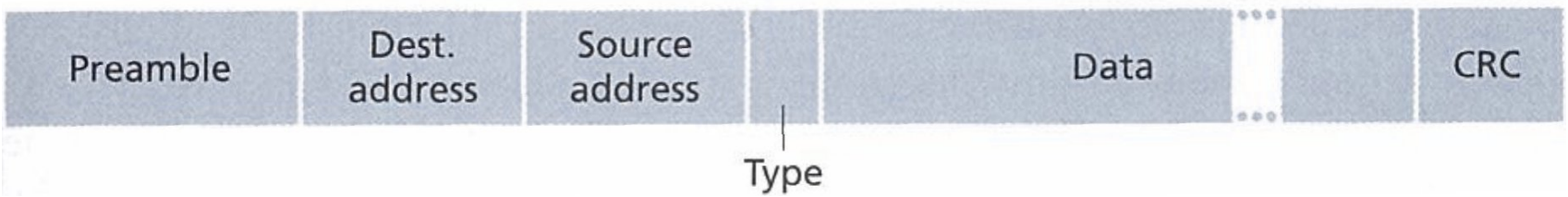
- sending adapter encapsulates network layer protocol packet (e.g. IP datagram) in Ethernet frame
- Ethernet frame: 6 fields common to all Ethernet protocols



Ethernet frame structure

Preamble (8 bytes)

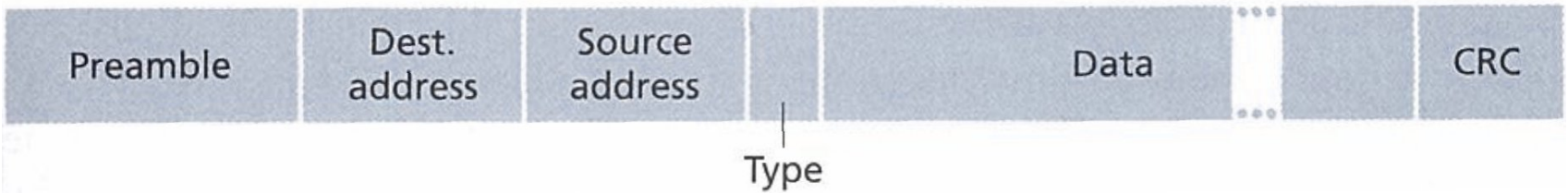
- Ethernet frame begins with an 8-byte preamble field
- Each of the first 7 bytes of the preamble has a value of 10101010
- The last byte is 10101011
- First 7 bytes serve to wake up the receiving adapters and to synchronize their clocks to that of the senders clock.
- Each adapter attempts to transmit a frame at a specific data rate (e.g. 10Mbps, 100Mbps or 1Gbps). However, there will always be some drift in the data rate. Synchronise receiver to this drift.



Ethernet frame structure

Source address (6 bytes)

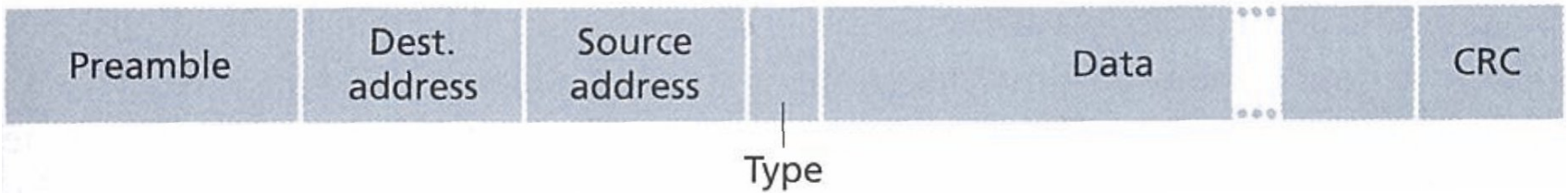
- Carries the source MAC address
- MAC address (aka LAN or physical or Ethernet address):
  - 48-bit address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD
    (hexadecimal (base 16) notation (each "number" represents 4 bits))



Ethernet frame structure

Destination address (6 bytes)

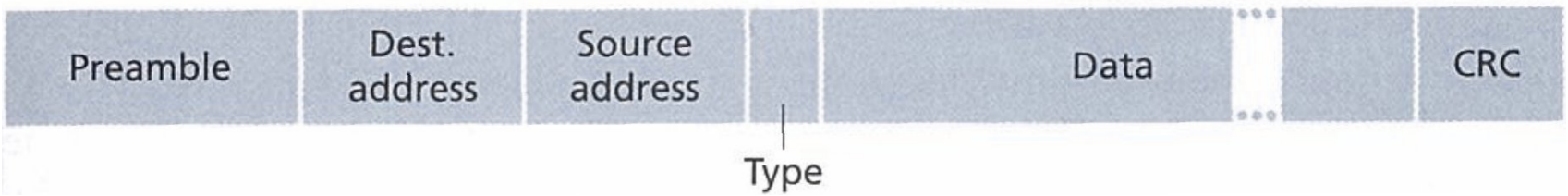
- Carries the destination MAC (aka LAN or physical or Ethernet) address
  - 48-bit address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD
    (hexadecimal (base 16) notation (each "number" represents 4 bits))
- If adapter receives frame with matching destination address, or with broadcast address, it passes data in frame to network layer protocol
- Otherwise, adapter discards frame



Ethernet frame structure

Type field (2 bytes)

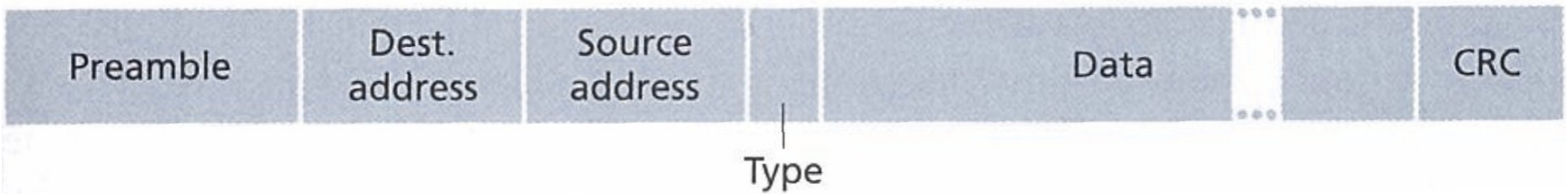
- Permits Ethernet to multiplex network-layer protocols
- Hosts can use multiple network-layer protocols (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- Ethernet needs to know for an arriving frame to which network-layer protocol it should pass the contents of the data field.
- A network layer protocol has its specific type value that can be specified in this field (e.g. 0800 for IPv4, 86DD for IPv6)



Ethernet frame structure

Data field (46 to 1500 bytes)

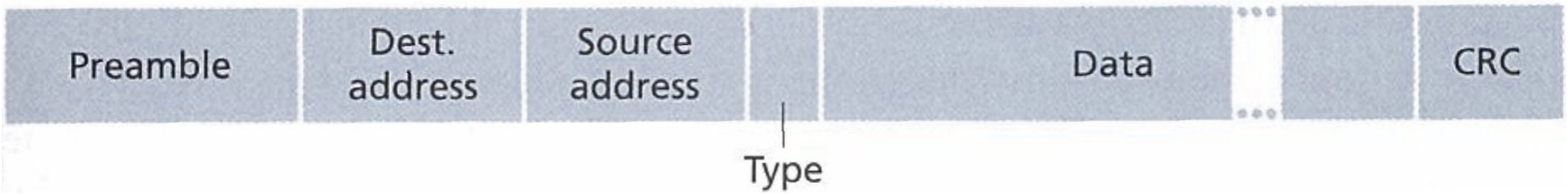
- Carries the network-layer datagram (e.g. IP datagram)
- MTU 1500 bytes (longer datagrams should be fragmented)
- Data shorter than 45 bytes is stuffed



Ethernet frame structure

Cyclic redundancy check (CRC) (4 bytes)

- Contains the CRC bits
- Allows the receiving adapter to detect bit errors in the frame
- The frame is dropped if an error is detected



Ethernet: unreliable, connectionless

Unreliable

- receiving NIC sometimes drop data
- receiving NIC does not send acknowledgments to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer reliable delivery transmission (e.g., TCP), otherwise dropped data lost

Connectionless

- no handshaking between sending and receiving NICs

Ethernet's MAC protocol

- unslotted CSMA/CD with binary backoff

Ethernet switch

link-layer device: takes an active role

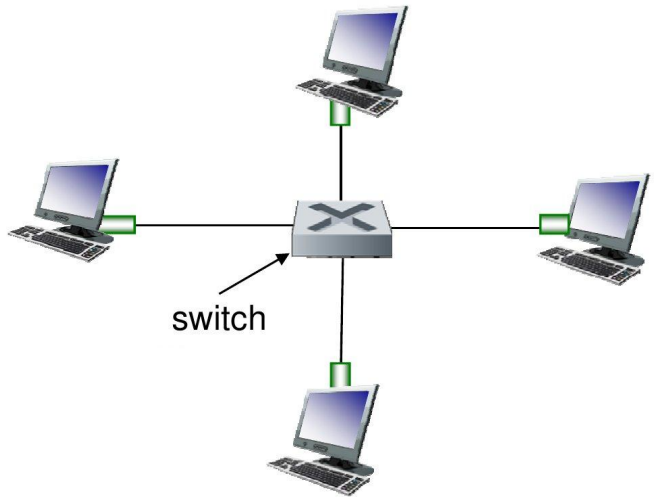
- store, forward Ethernet frames
- examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment

Transparent

- hosts are unaware of presence of switches

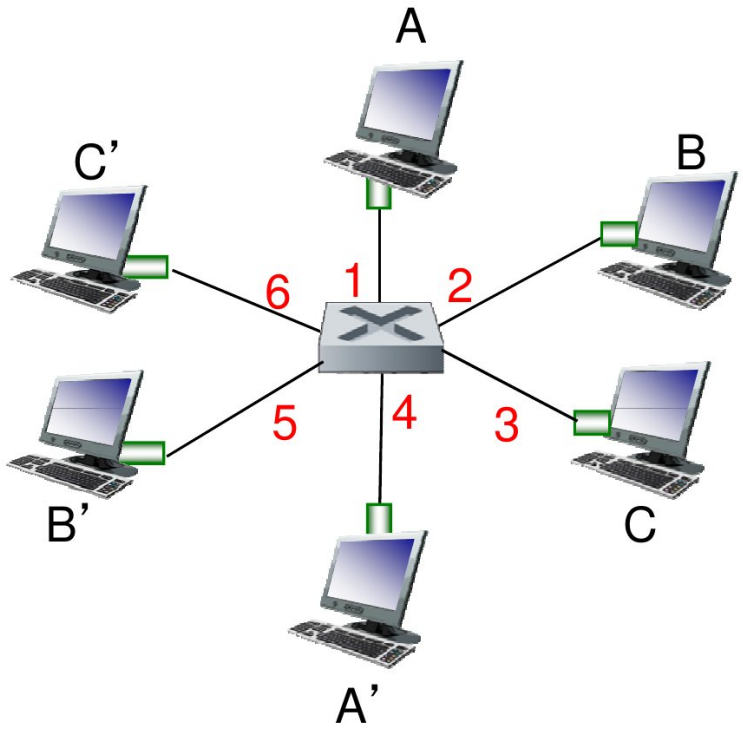
plug-and-play, self-learning

- switches do not need to be configured



Switch: 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' can transmit simultaneously, without collisions



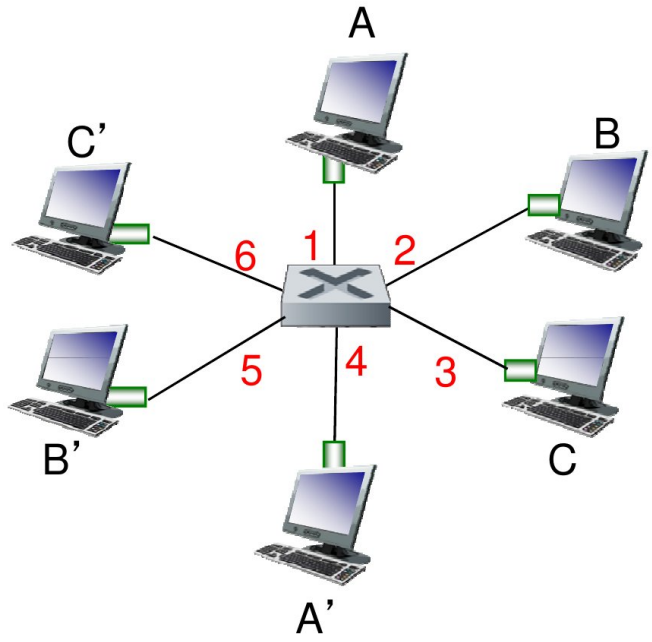
switch with six interfaces (1,2,3,4,5,6)

switch vs. hubs (common in 1990s)

- Hubs re-transmit all bits they receive on all their interfaces
- Collision occurs when two hosts transmit data

Switch forwarding table

- Question: How does switch know A' reachable via interface 4, B' reachable via interface 5?
- Answer: each switch has a switch table, each entry:
  - MAC address of host
  - interface to reach host
  - time stamp
- Question: How are entries created, maintained in switch table?



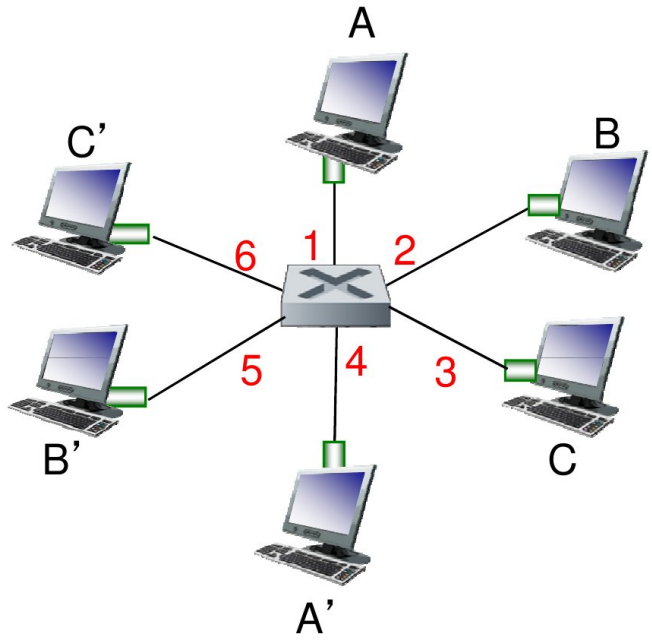
switch with six interfaces (1,2,3,4,5,6)

MAC address	interface	Time
A	1	14:05:12
B	2	14:35:43
...	...	...

Switch: self-learning

switch learns which hosts can be reached through which interfaces

- when frame received switch "learns" location of sender
- records send/location pair in switch table
- if destination unknown broadcast on all other interfaces



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

For example:

- A sends a frame to B
- the frame is received by the switch and a new entry is added

MAC address	interface	Time

Switch table (initially empty)

MAC address	interface	Time
A	1	14:05:12

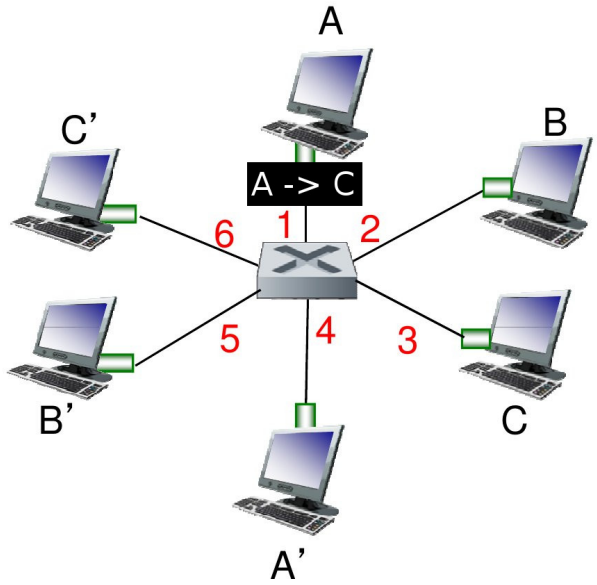
Switch table (initially empty)

Switch: frame filtering/forwarding

when frame received at switch:

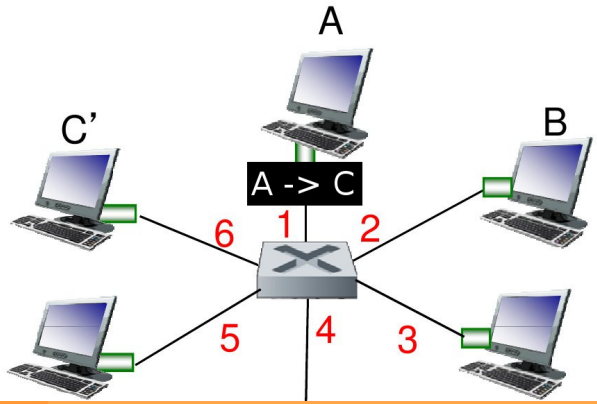
1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
   {
   if destination on segment from which frame arrived
   drop frame
   else forward frame on interface indicated by entry
   }
   else flood /*forward on all interfaces except arriving interface*/

Self-learning, forwarding: example



MAC addr.	interface	Time

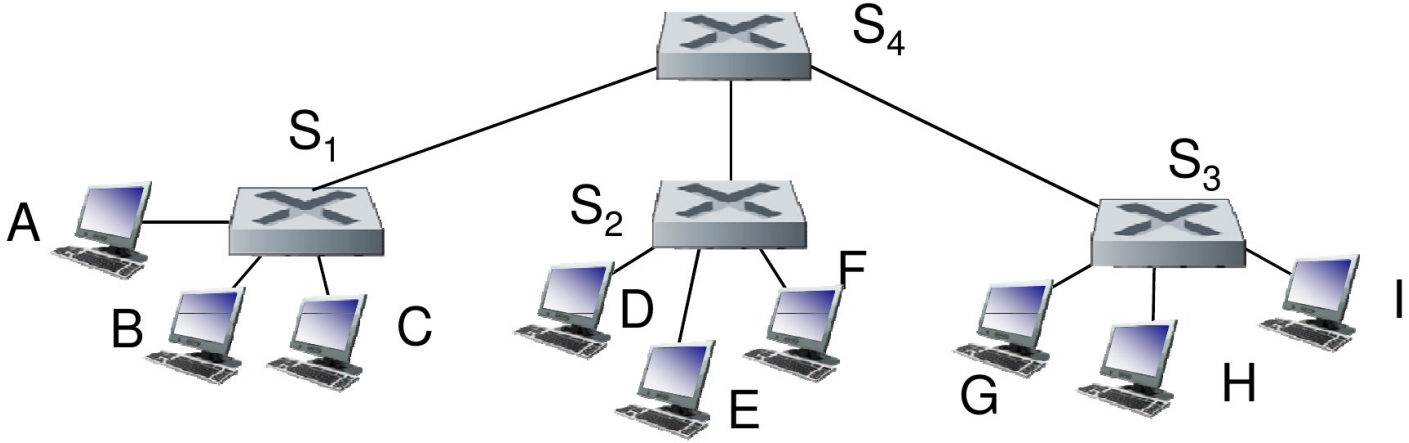
Switch table (initially empty)



Interconnecting switches

switches can be connected together

- Question: sending from A to G - how does S1 know to forward frame destined to F via S_4 and S_3?
- Answer: self learning! (works exactly the same as in single-switch case!)



Ethernet switch: Summary

Properties of Link-Layer Switching

- Elimination of collisions: full duplex switches never transmit more than one frame on a segment
- Throughput: max throughput is the sum of all interfaces rate
- Heterogeneous links: isolates links thus they can run at different speed and over different media
- Management: isolate problems, enhance security, gather statistics to debug, correct problems and plan better LAN

Study case: Data center networks

10's to 100's of thousands of hosts, often closely coupled, in close proximity

- e-business (e.g. Amazon)
- content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
- search engines, data mining (e.g., Google)

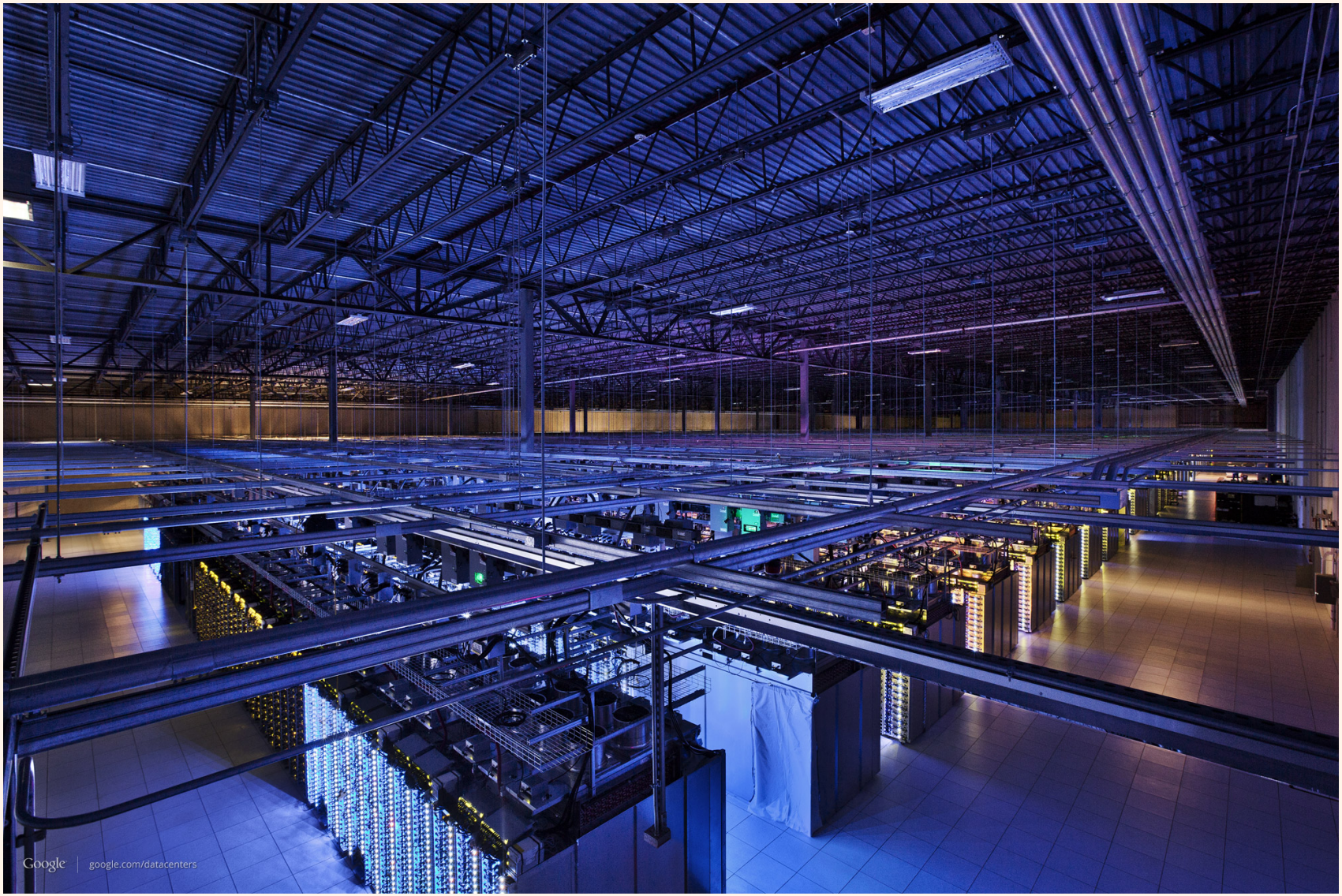
Challenges

- multiple applications, each serving massive numbers of clients
- managing/balancing load, avoiding processing, networking, data bottlenecks

Study case: Data center networks

Inside Google Data Center, Lynhaven Drive, Lenoir, NC

Study case: Data center networks

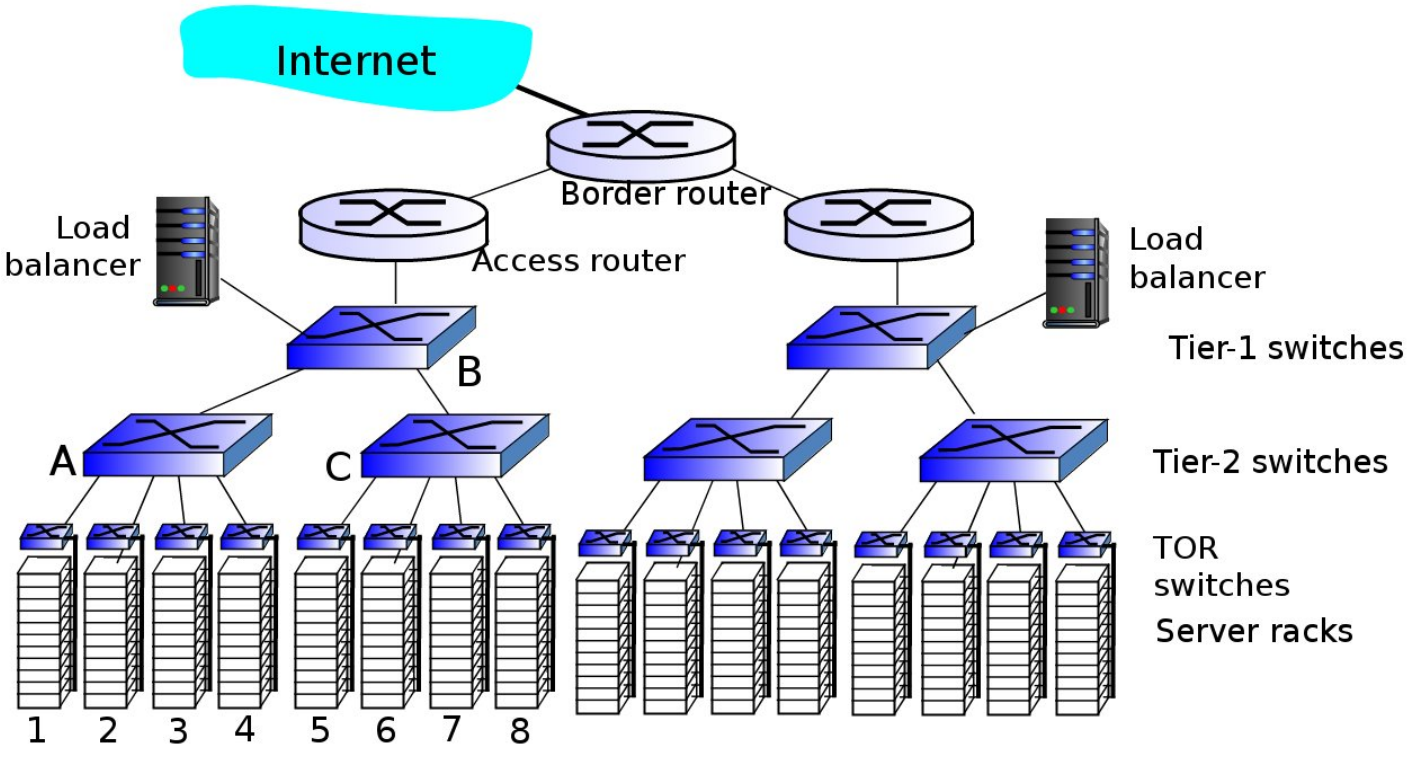


Inside Google Data Center, Lynhaven Drive, Lenoir, NC

Data center networks: Load balancing

load balancer: application-layer routing

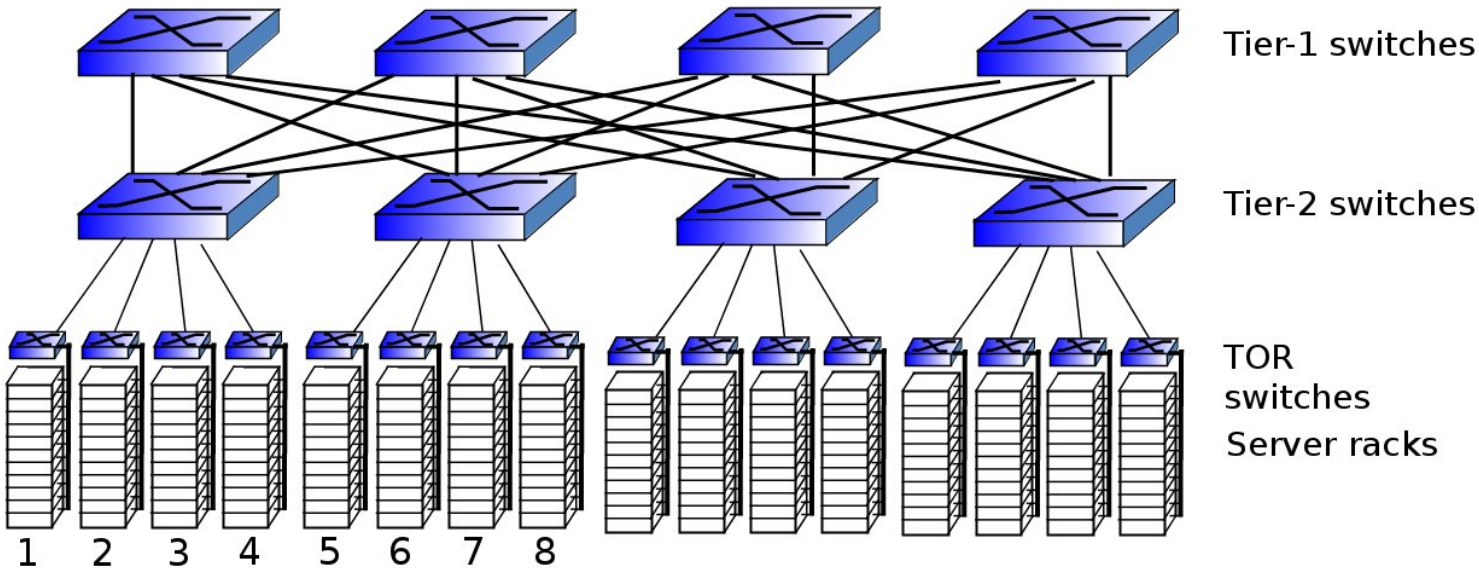
- receives external client requests
- directs workload within data center
- returns results to external client (hiding data center internals from client)



Data center networks: Fully connected topology

Rich interconnection among switches, racks:

- increased throughput between racks (multiple routing paths possible)
- increased reliability via redundancy
- open problem: many recent papers on data center design



The Link Layer: Summary

Today's lecture covered the link layer

- Link layer services
- Error detection, correction
- Multiple access protocols
- Ethernet
- Switches
- Data center networking

In the next 2 lectures

Network layer: one of the most interesting/complex layers in the protocol stack. You will learn how IP and routers work

IP Stack

Application

Transport

Network

Link

Physical

Today's important points

- Checksum
- CRC
- Ethernet (CSMA/CD)