521150A
Introduction to Internet

Lecture 3 – Data link layer, part I: Architectures, concepts, Medium Access Control

Erkki Harjula 2023

Schedule of the course

PART I: Basics of networking and Internet, data link layer

Mon 13.3. 10-12 L6/Zoom	Lecture 1: Introduction & motivation
Tue 14.3. 10-12 L6/Zoom	Lecture 2: Architecture & design principles
Wed 15.3. 10-12 L5/Zoom	Lecture 3: Data link layer – basics part I
Thu 16.3. 10-12 L4/Zoom	Exercise session 1A
Mon 20.3. 10-12 L6/Zoom	Lecture 4: Data link layer – basics part II
Tue 21.3. 10-12 L6/Zoom	Exercise session 1B
Tue 21.3. 14-18 AT122/Zoom	Lab exercise 1 – group 3
Wed 22.3. 10-12 L5/Zoom	Lecture 5: Data link layer – Wired networks
Wed 22.3. 14-18 AT122/Zoom	Lab exercise 1 – group 4
Thu 23.3. 14-18 AT122/Zoom	Lab exercise 1 – group 2
Fri 24.3. 12-16 AT122/Zoom	Lab exercise 1 – group 1
Mon 27.3. 14-16 L5/Zoom	Exercise session 1C
Tue 28.3. 10-12 L6/Zoom	Lecture 6: Data link layer – Wireless networks
Wed 29.3. 10-12 Moodle	Theory exam 1

PART II: Network and transport layers

Thu 30.3. 8-10 L5/Zoom	Lecture 7: Network layer part I
Mon 3.4. 14-16 L5/Zoom	Exercise session 2A
Tue 4.4. 10-12 L6/Zoom	Lecture 8: Network layer part II
Wed 5.4. 10-12 L5/Zoom	Lecture 9: Transport layer part I; Course work intro
Tue 11.4. 10-12 L6/Zoom	Lecture 10: Transport layer part II
Tue 11.4. 14-18 AT122/Zoom	Lab exercise 2 – group 3
Wed 12.4. 10-12 Moodle	Theory exam 2
Wed 12.4. 14-18 AT122/Zoom	Lab exercise 2 – group 4
Thu 13.4. 8-10 L5/Zoom	Exercise session 2B
Thu 13.4. 14-18 AT122/Zoom	Lab exercise 2 – group 2
Fri 14.4. 12-16 AT122/Zoom	Lab exercise 2 – group 1

PART III: Application layer, network security and multimedia

Fri 14.4. 10-12 L5/Zoom	Lecture 11: Networking applications
Mon 17.4. 14-16 L5/Zoom	Lecture 12: Network security
Tue 18.4. 10-12 L6/Zoom	Exercise session 3A
Wed 19.4. 10-12 L5/Zoom	Lecture 13: Multimedia and QoS
Mon 24.4. 14-16 L5/Zoom	Exercise session 3B
Tue 25.4. 14-18 AT122/Zoom	Lab exercise 3 – group 3
Wed 26.4. 10-12 L5/Zoom	Lecture 14: Challenges&Future Internet trends
Wed 26.4. 14-18 AT122/Zoom	Lab exercise 3 – group 4
Thu 27.4. 14-18 AT122 /Zoom	Lab exercise 3 – group 2
Fri 28.4. 12-16 AT122 /Zoom	Lab exercise 3 – group 1
Wed 3.5. 10-12 Moodle	Theory exam 3
Thu 25.5. 16-19 L1	Final exam

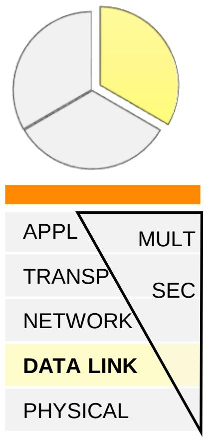
Course work (independent work)

Main learning objectives of this lecture

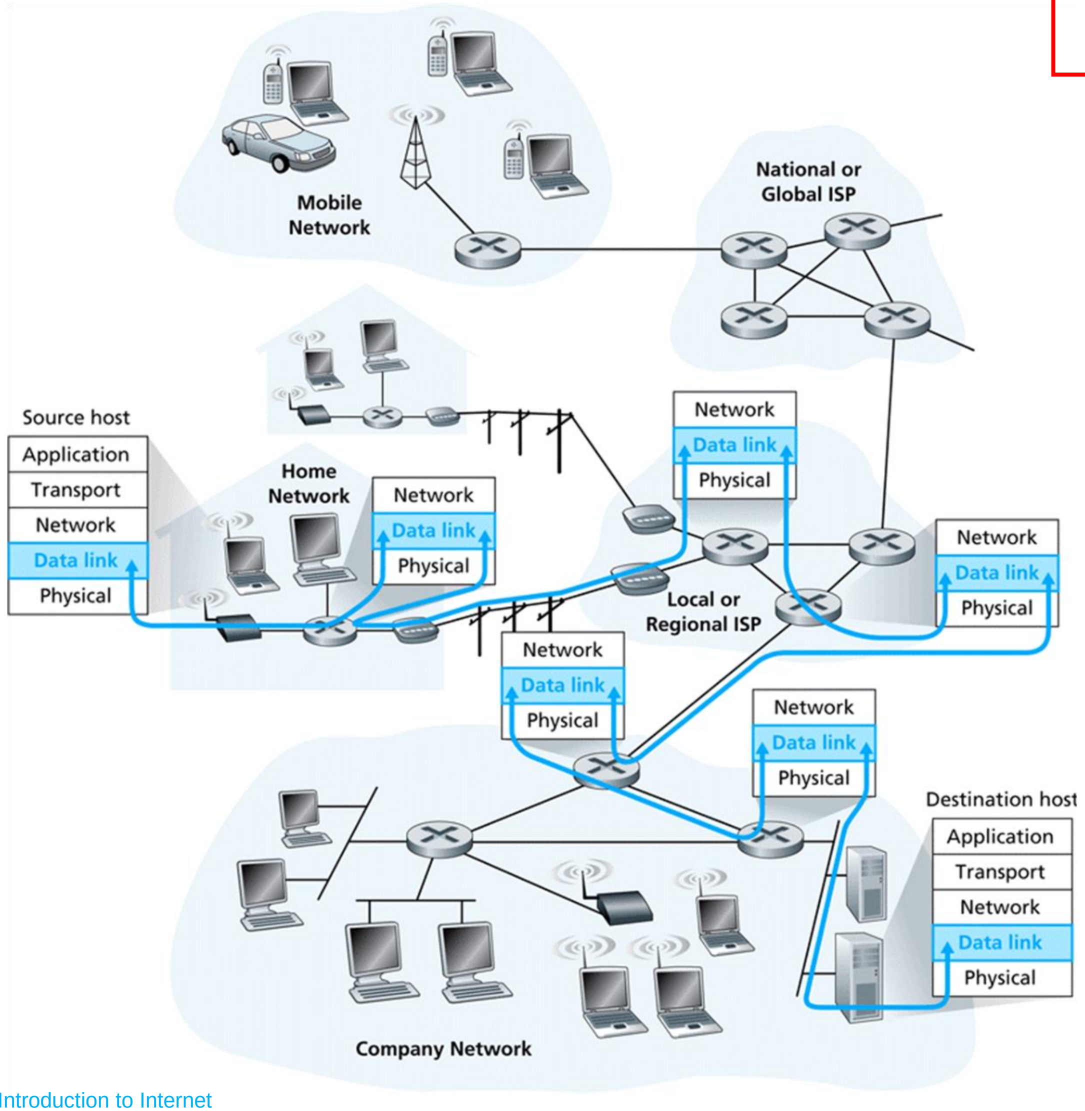
1. **Know the architecture and basic concepts of data link layer:**
   - Transmission media types
   - Synchronous vs asynchronous transmission
   - Framing
   - Error detection
2. **Know the main medium access control (MAC) protocols and understand their basic principles**
   - Multiple access protocols
     - Channel partitioning protocols
     - Random access protocols
     - “Taking turn”-protocols
   - Basics of point-to-point protocols

Architecture and Basic concepts

Data link layer (L2)

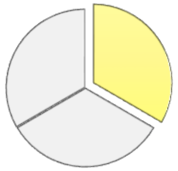


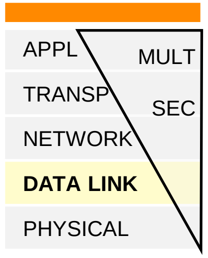
Data link layer delivers a frame (encapsulated datagram) from a node to adjacent node over a link



- **Nodes:** Hosts and routers
- **Links:** Communication channels that connect adjacent nodes along communication path
  - Wired or wireless
  - E.g. Ethernet, WLAN, LTE
  - Key feature: data rate ("bandwidth") expressed in bps (bits per second)
- **Frames:** L2 packets which encapsulate datagram provided by L3 (network layer)
- Data link layer functionality is implemented in NIC

Implementation of data link layer (1)

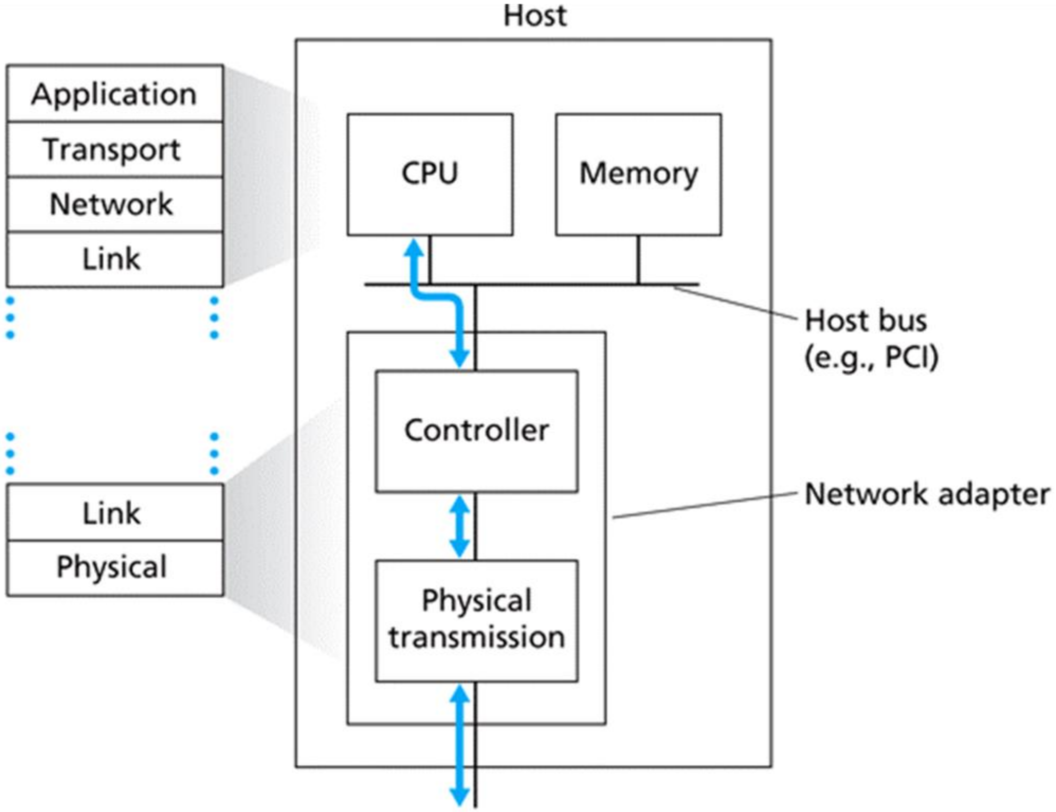




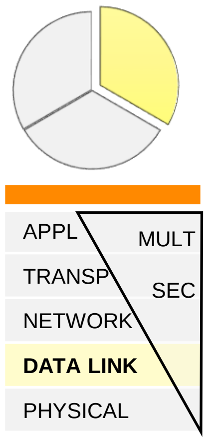
- **In each and every networked host**
  - Implemented in “adapter” (network interface card, NIC)
  - Integrated on motherboard, Ethernet card, USB stick, etc.
  - Implements data link layer and physical layer
  - Typically has unique 48-bit address (so-called MAC address) burned into its ROM (may be software modifiable)

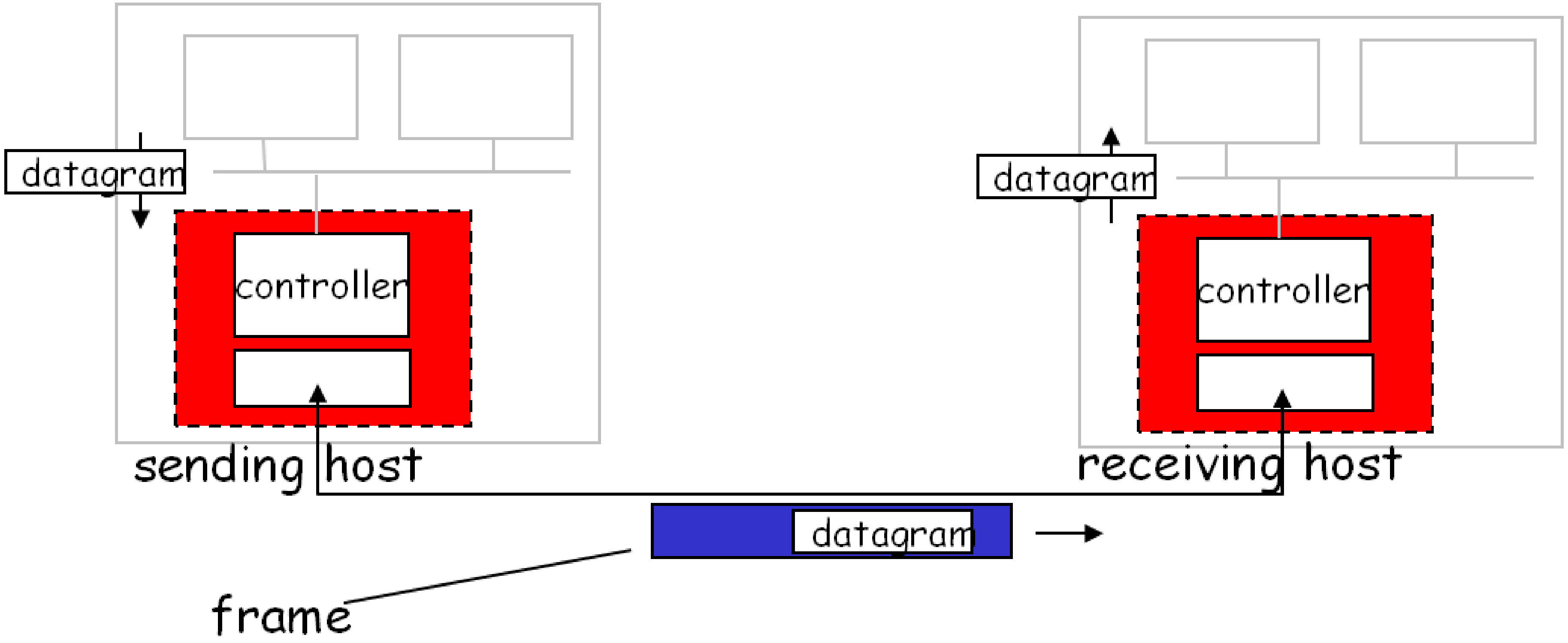
NIC is identified by a **physical (MAC) address**, e.g.
00:A0:C9:14:C8:29

- **Attaches into host’s system buses**
- **Combination of hardware, software and firmware**



Implementation of data link layer (2)





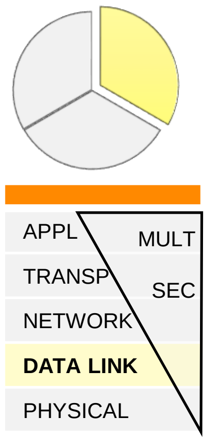
Sending host

- Encapsulates datagram in a frame
- Adds error checking bits, flow control, etc.

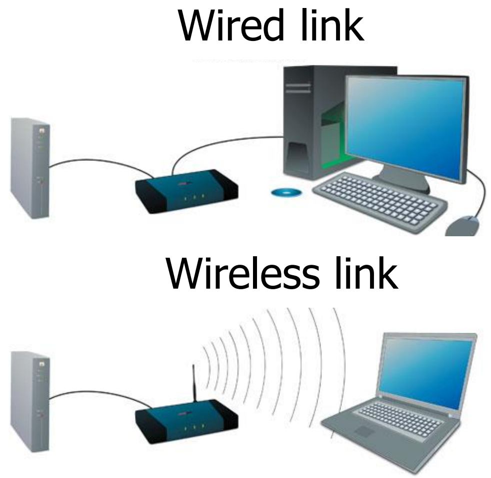
Receiving host

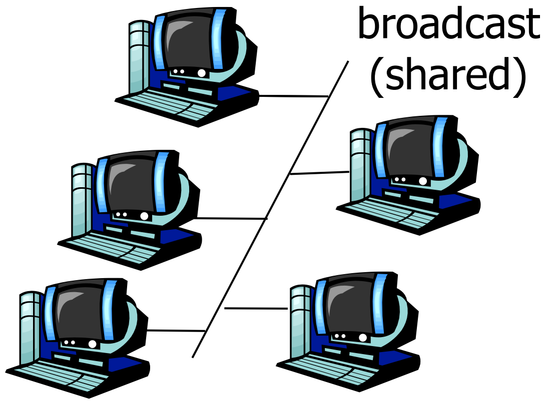
- Error control, flow control, etc.
- Extracts datagram, passes to upper layer

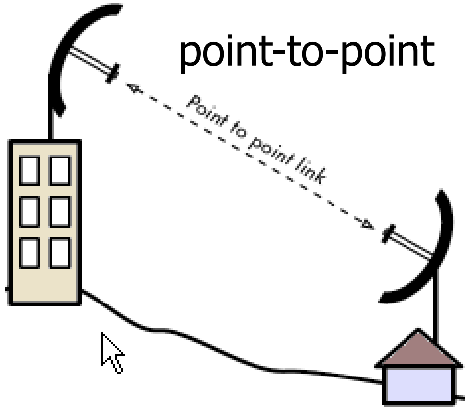
Link types

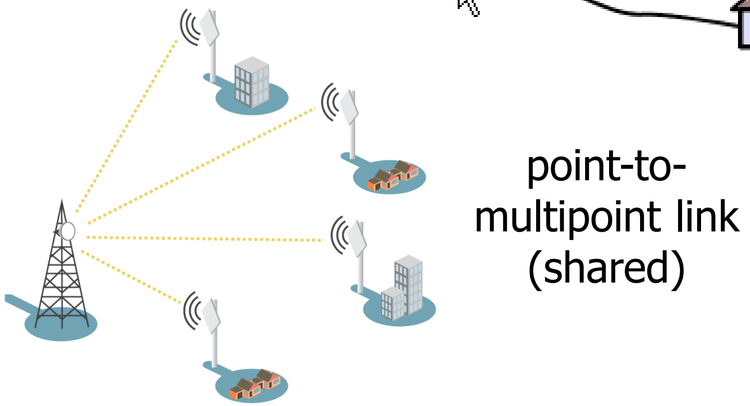


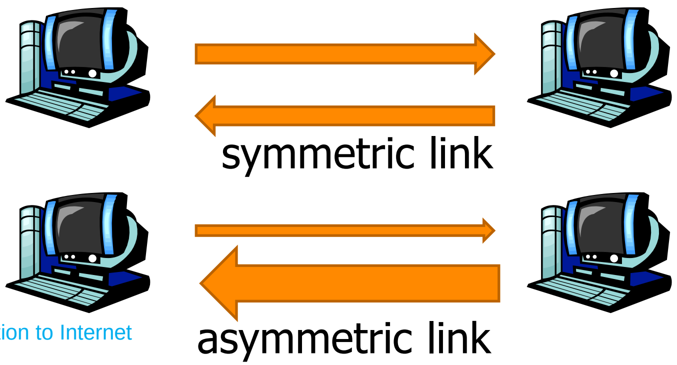
- **Guided vs unguided (wireless) links**
- **Broadcast (shared) vs point-to-point links**
- **Symmetric vs asymmetric links (up/downlink)**
  - 10/10 VDSL is symmetric
  - 8/1 ADSL is asymmetric



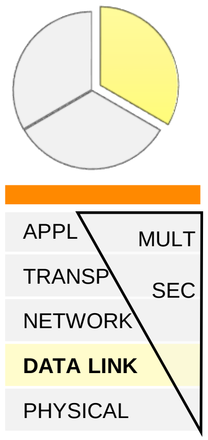






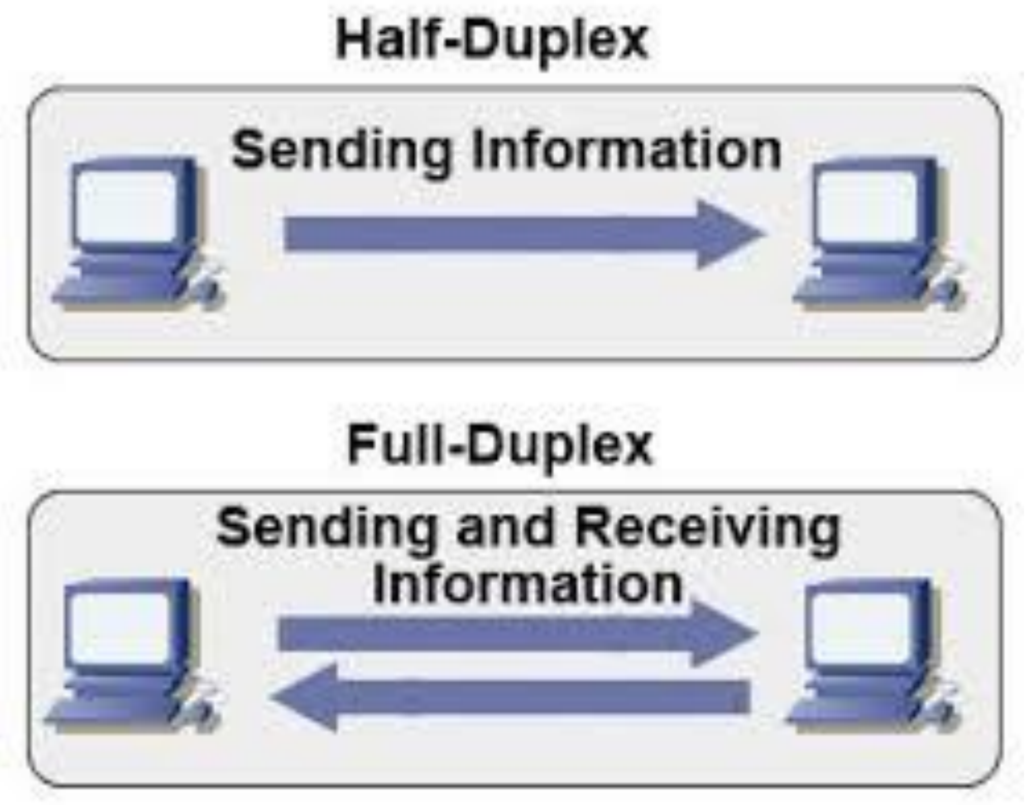


Half-duplex vs Full-duplex

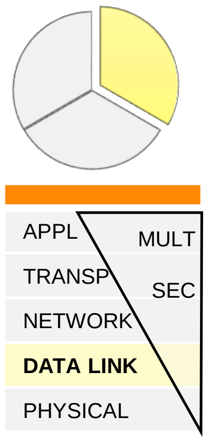


- In a **half-duplex** system, each party can communicate with the other but not simultaneously; the communication is one direction at a time. An example of a half-duplex device is a walkie-talkie two-way radio that has a "push-to-talk" button.

- In a **full-duplex** system, both parties can communicate with each other simultaneously. An example of a full-duplex device is a telephone; the parties at both ends of a call can speak and be heard by the other party simultaneously´.



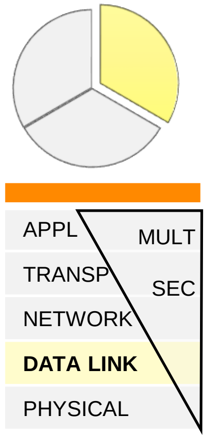
https://en.wikipedia.org/wiki/Duplex_(telecommunications)



Link media

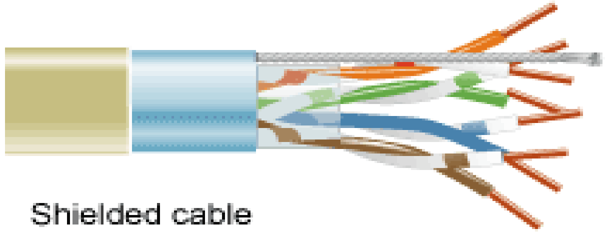
- **Guided (wire, optical fiber) vs unguided (wireless)**
- **Key concerns are data rate and distance**
- **Design factors**
  - Frequency bandwidth (higher bandwidth gives higher data rate)
  - Transmission impairments (e.g. attenuation)
  - Interference
  - Number of receivers in guided media
    - Each receiver introduces some attenuation and distortion
- **The capacity of a transmission channel is a function of**
  - Data rate (in bits per second)
  - Bandwidth (in Hz)
  - Noise (on communications link)
  - Error rate (of corrupted bits)

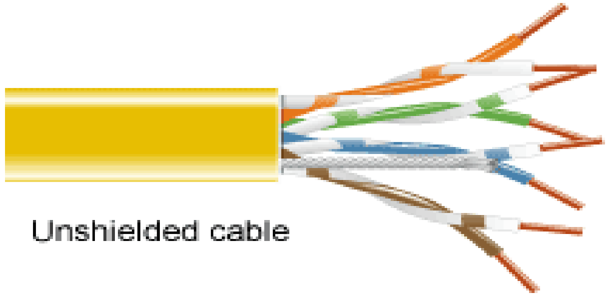
Link media: Guided (wired) media



Twisted pair

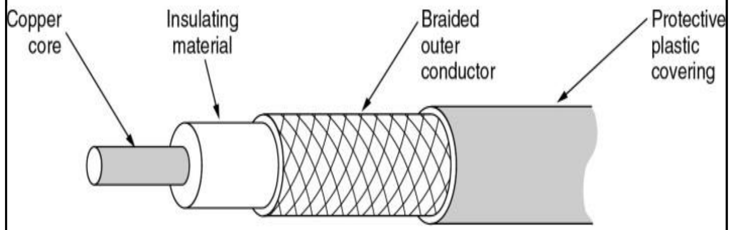
- Unshielded vs shielded
- Limited distance and data rate
- Suspect to interference and noise
- Frequency range: 0-1 MHz
- Typical attenuation: 0.7 dB/km @ 1 KHz
- Typical delay: 5 μs/km
- Typical repeater spacing: 2 km





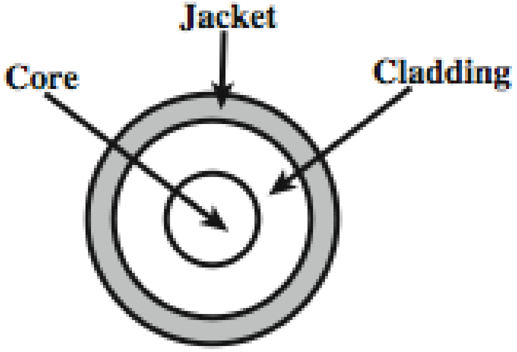
Coaxial cable

- Superior frequency properties to twisted pair
- Performance limited by attenuation and noise
- Frequency range: 0-500 MHz
- Typical attenuation: 7 dB/km @ 10 Mhz
- Typical delay: 4 μs/km
- Typical repeater spacing: 1-9 km depending on frequency

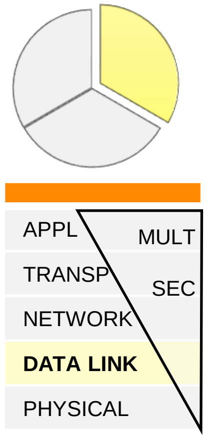


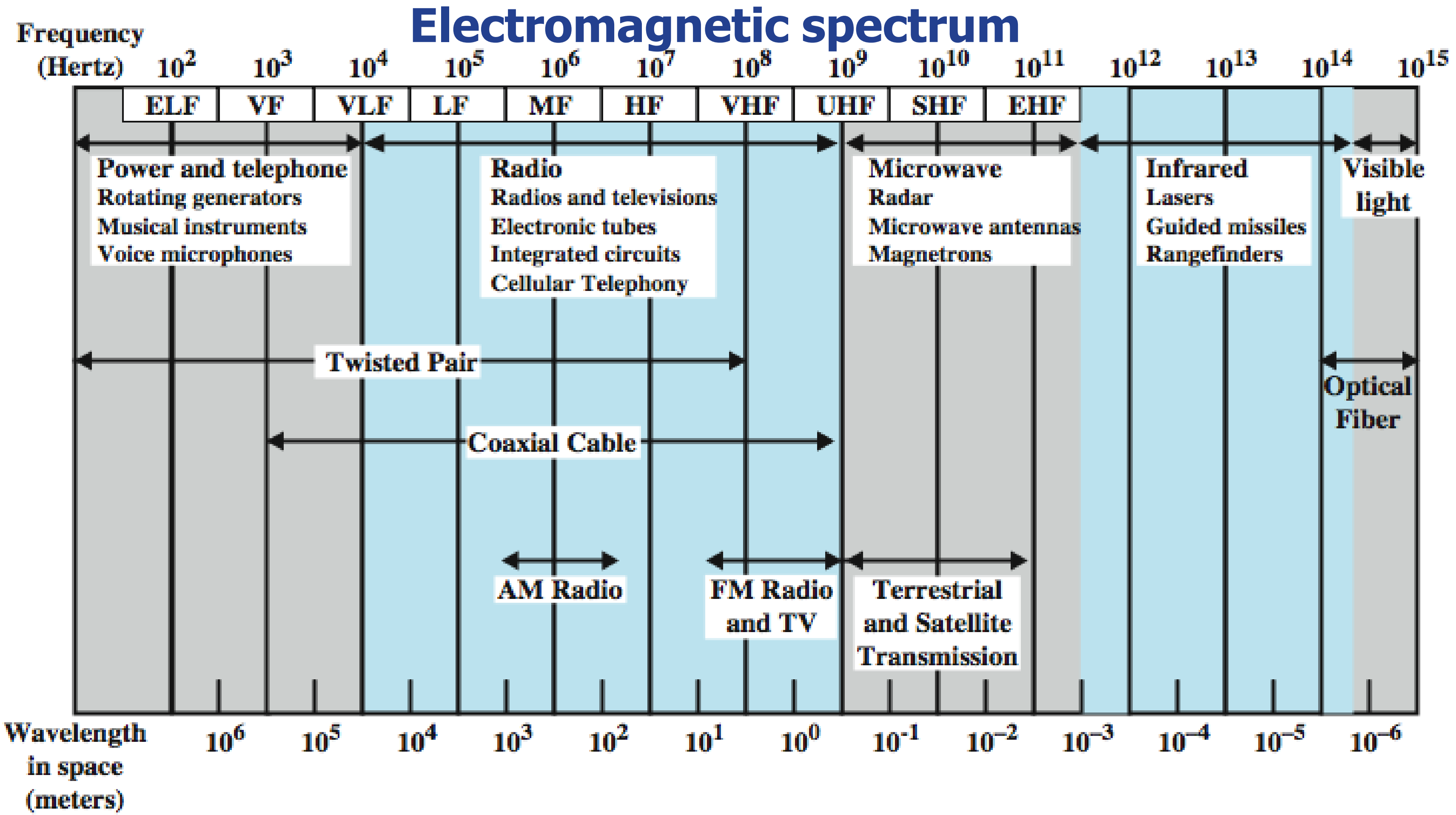
Optical fiber

- Greater capacity (data rates of hundreds of Gbps)
- Smaller size and weight
- Lower attenuation
- Electromagnetic isolation
- Greater repeater spacing (tens of kms)
- Frequency range: 186-370 THz ($T \sim 10^{12}$)
- Typical attenuation: <0.5 dB/km
- Typical delay: 5 μs/km
- Typical repeater spacing: 40 km



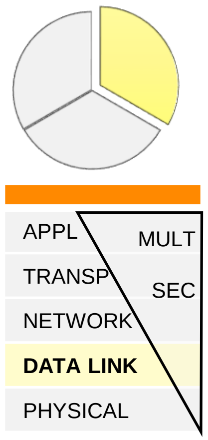
Link media: Unguided (wireless) media





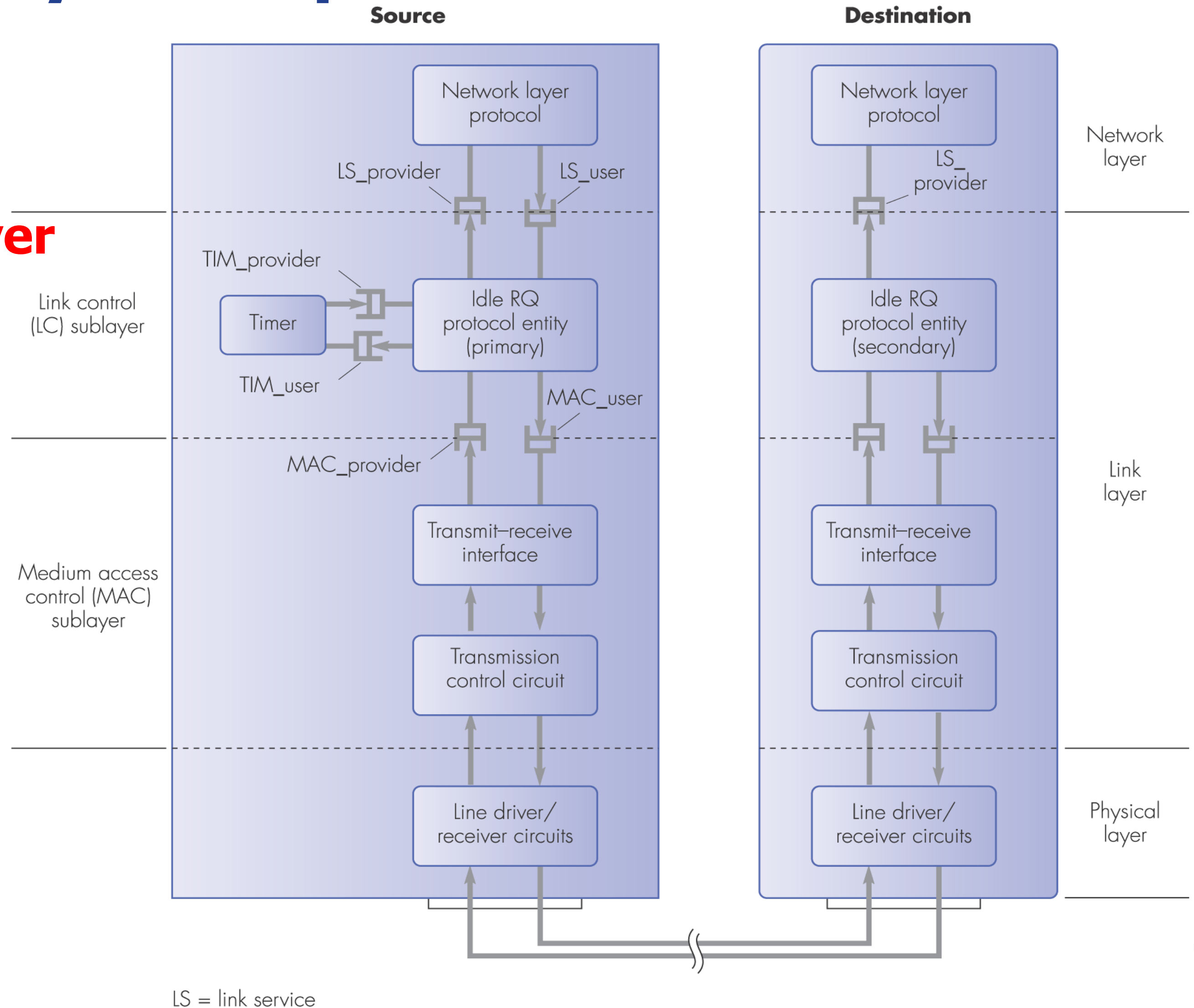
ELF = Extremely low frequency
VF = Voice frequency
VLF = Very low frequency
LF = Low frequency
MF = Medium frequency
HF = High frequency
VHF = Very high frequency
UHF = Ultrahigh frequency
SHF = Superhigh frequency
EHF = Extremely high frequency

Data link layer architecture

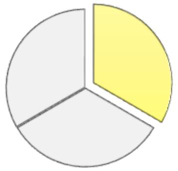


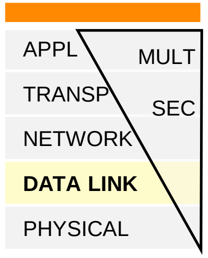
Typically data link layer comprises of two sublayers:

- **Link control (LC) sublayer**
  - Error control
  - Flow control
- **Medium access control (MAC) sublayer**
  - Framing
  - Transmission (medium access)



Digital data transmission over a link

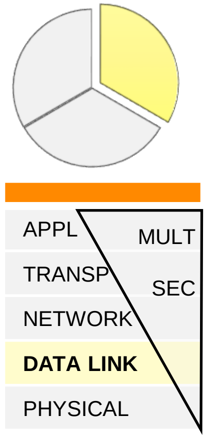




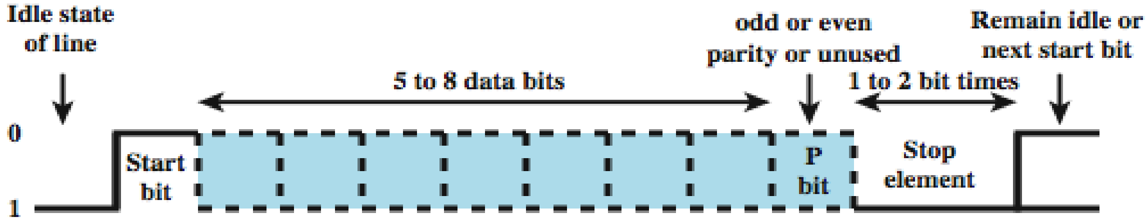
- **Need a mechanism to synchronize transmitter and receiver**
  - Receiver samples stream at bit intervals
  - If clocks not aligned and drifting, receiver will sample at wrong time after sufficient bits are sent

- **Two solutions to synchronizing clocks**
  - Asynchronous transmission
  - Synchronous transmission

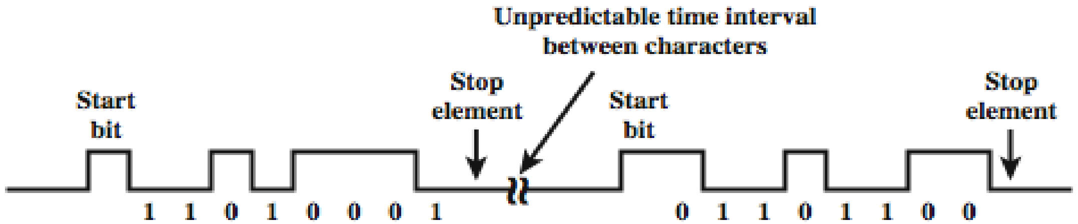
Asynchronous transmission



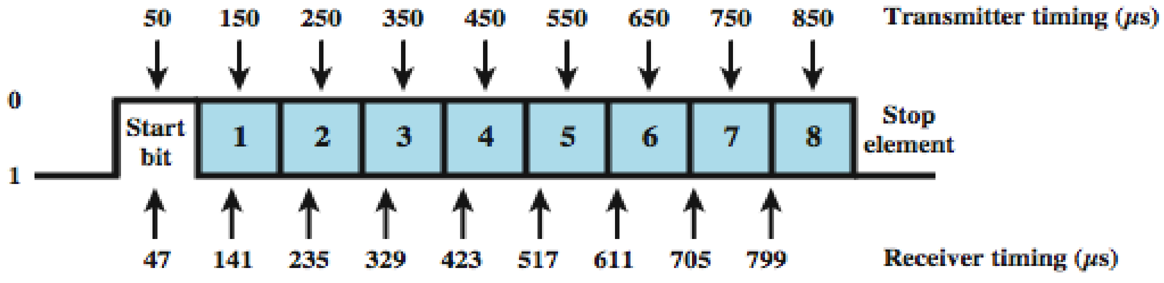
Timing/synchronization within each character separately



(a) Character format



(b) 8-bit asynchronous character stream



(c) Effect of timing error

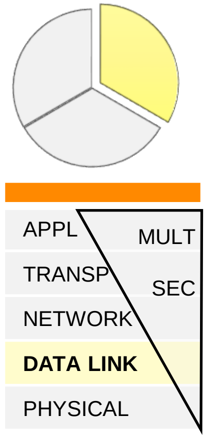
- Simple
- Cheap
- Overhead of 2 or 3 bits per char (~20%)
- Good for data with large gaps (e.g. keyboard)

Example of a sync error:

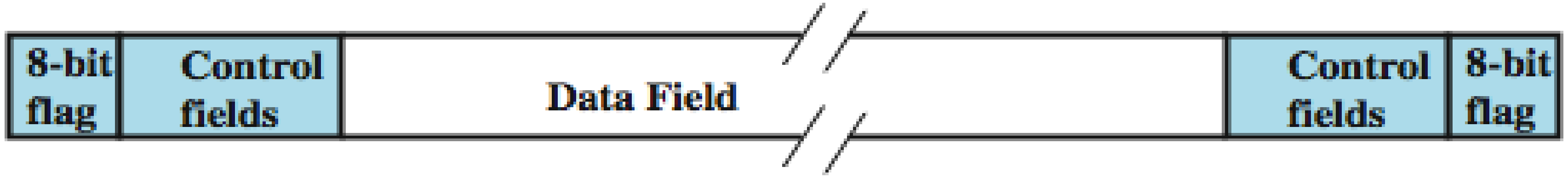
Receiver is 6% faster (samples the incoming chars every 94 μs, instead of 100 μs)

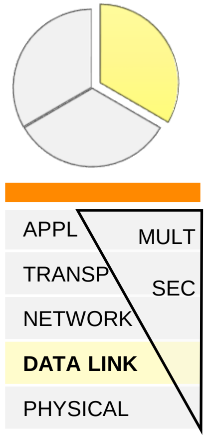
Last sample fails

Synchronous transmission



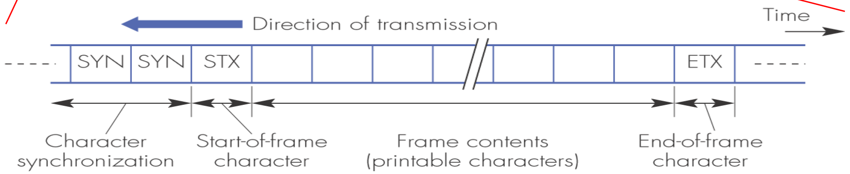
- Block of data transmitted sent as a frame
- Clocks must be synchronized, e.g. by
  - Separate clock line
  - Embedded clock signal in data
- Need to indicate start and end of block
  - Use preamble and postamble
- More efficient (lower overhead) than asynchronous

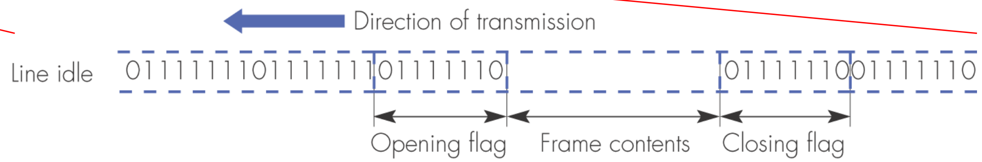




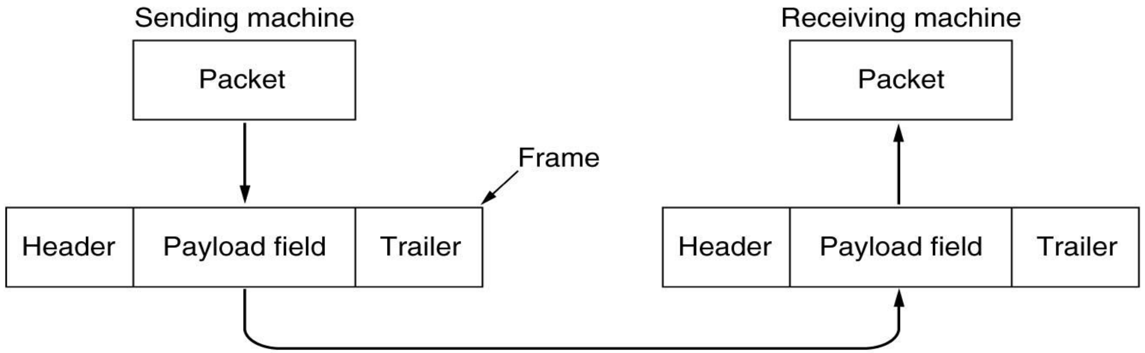
Framing

- **Encapsulate datagram into frame, adding header, trailer**
  - Frame contains synchronization flag (preamble), which allows receiver to synchronize itself
  - Character-oriented or bit-oriented

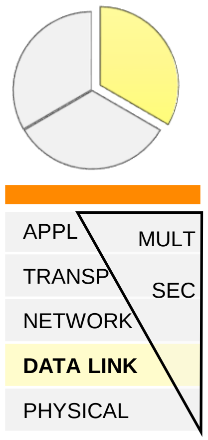




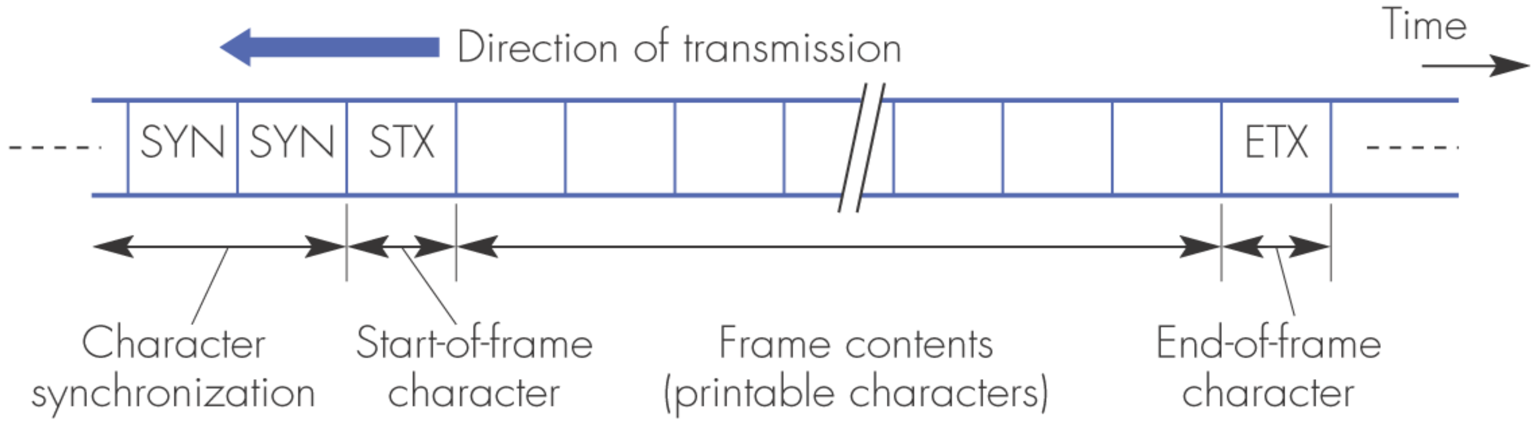
- **'Physical "MAC" addresses' used in frame headers to identify source and destination**
  - Do not mix with IP address!



Framing: Character-oriented

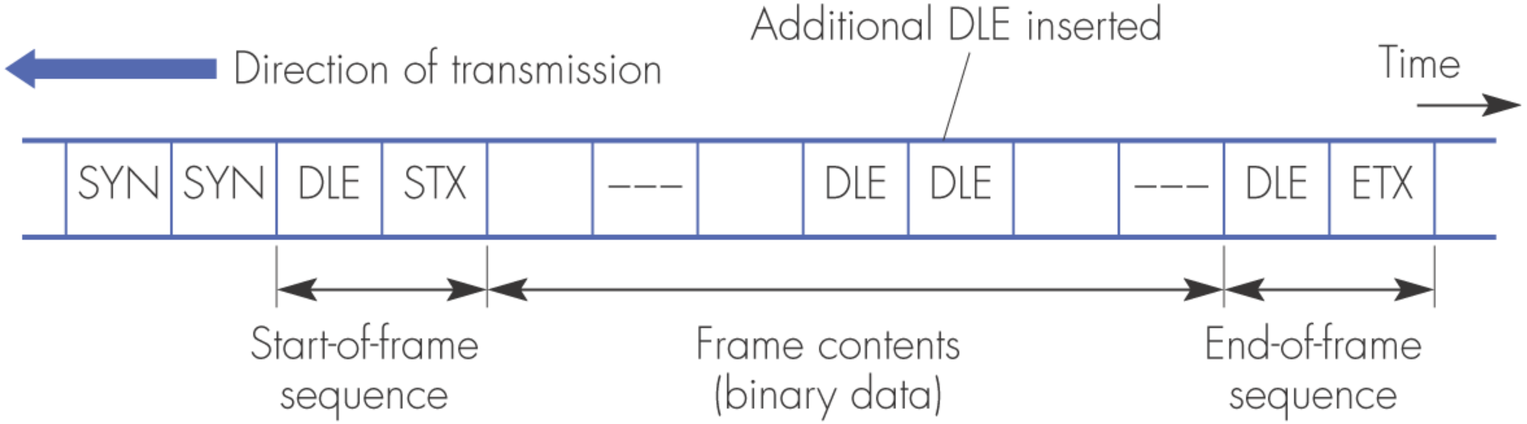


Frame structure:

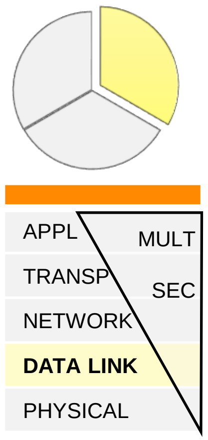


Byte stuffing to maintain sychronization:

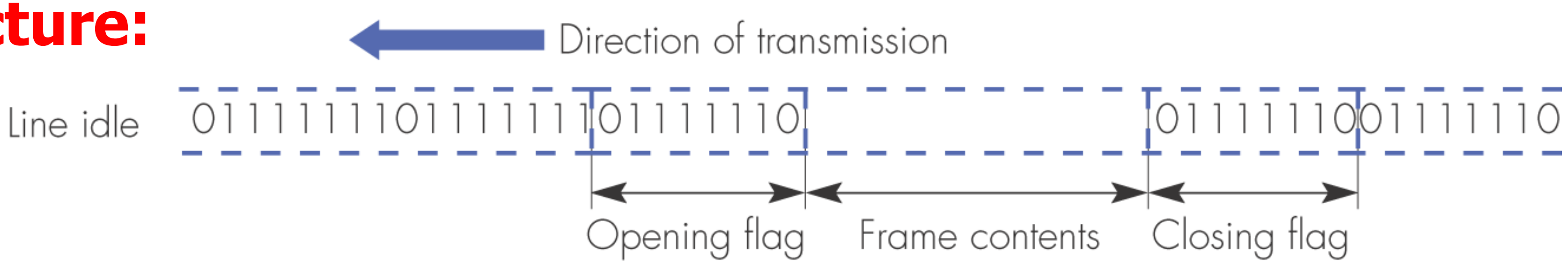
- Extra DLE (Data Link Escape) inserted after each occurrence of DLE byte in data
- Prevents flag bytes appearing in the byte stream, which would destroy synchronization
- Allows data transparency (arbitrary bytes are allowed in data frames)



Framing: Bit-oriented

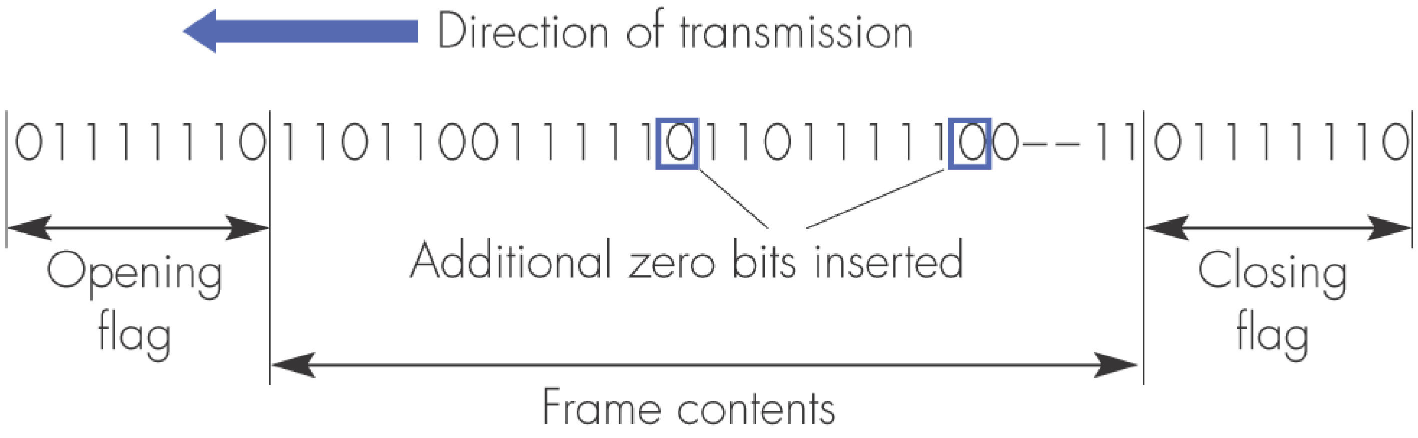


Frame structure:

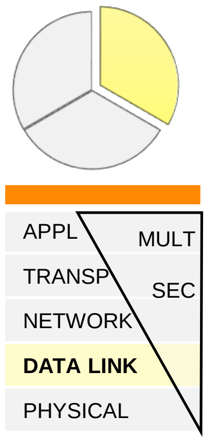


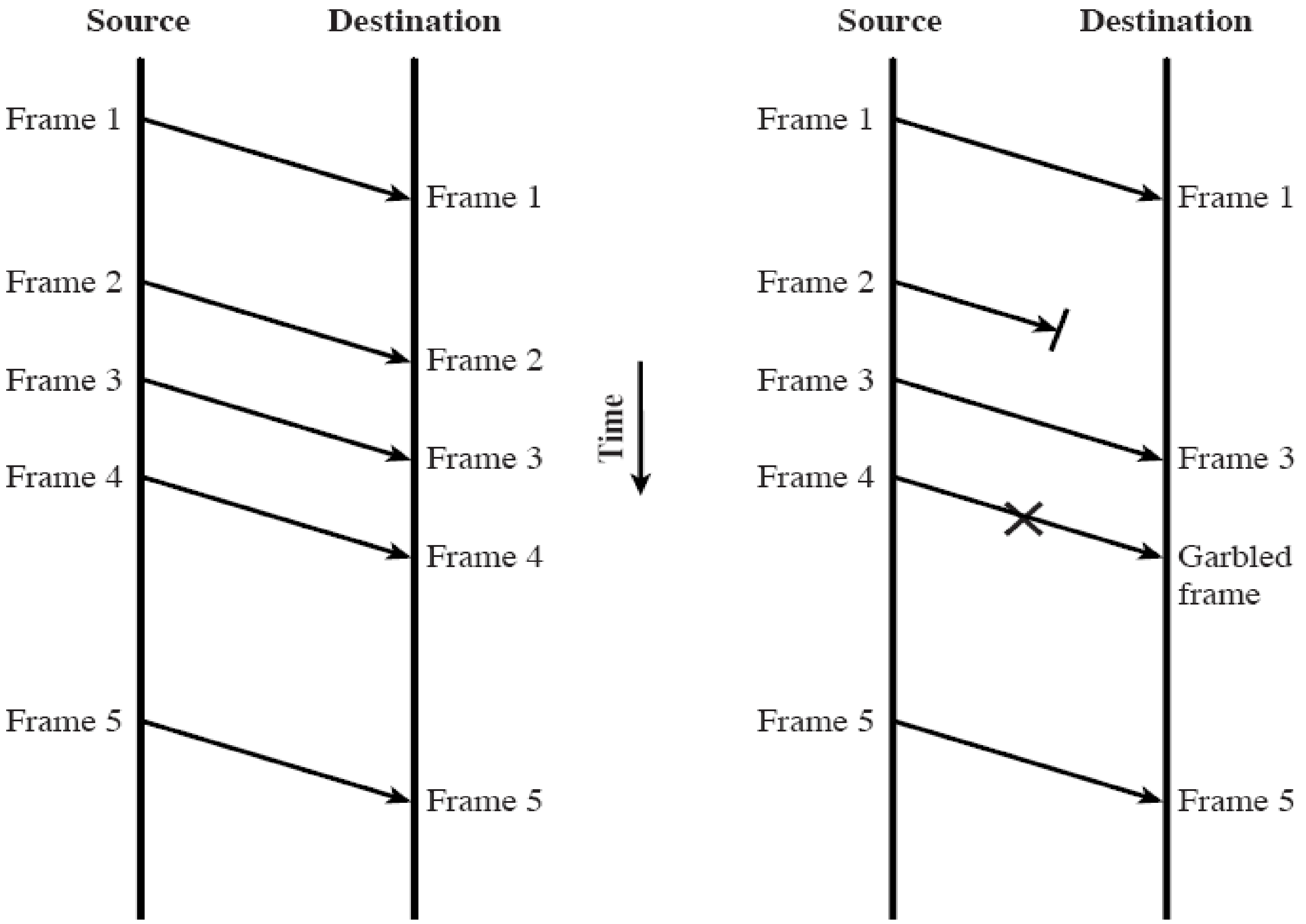
Bit stuffing to maintain synchronization (zero bit insertion):

- Extra 0 bit is inserted after each occurrence of five 1's
- Prevents flag fields (01111110) appearing in the bit stream, which would destroy synchronization
- Allows data transparency (arbitrary bit patterns allowed in data frames)



Transmission may introduce errors:

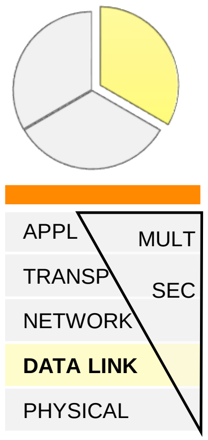




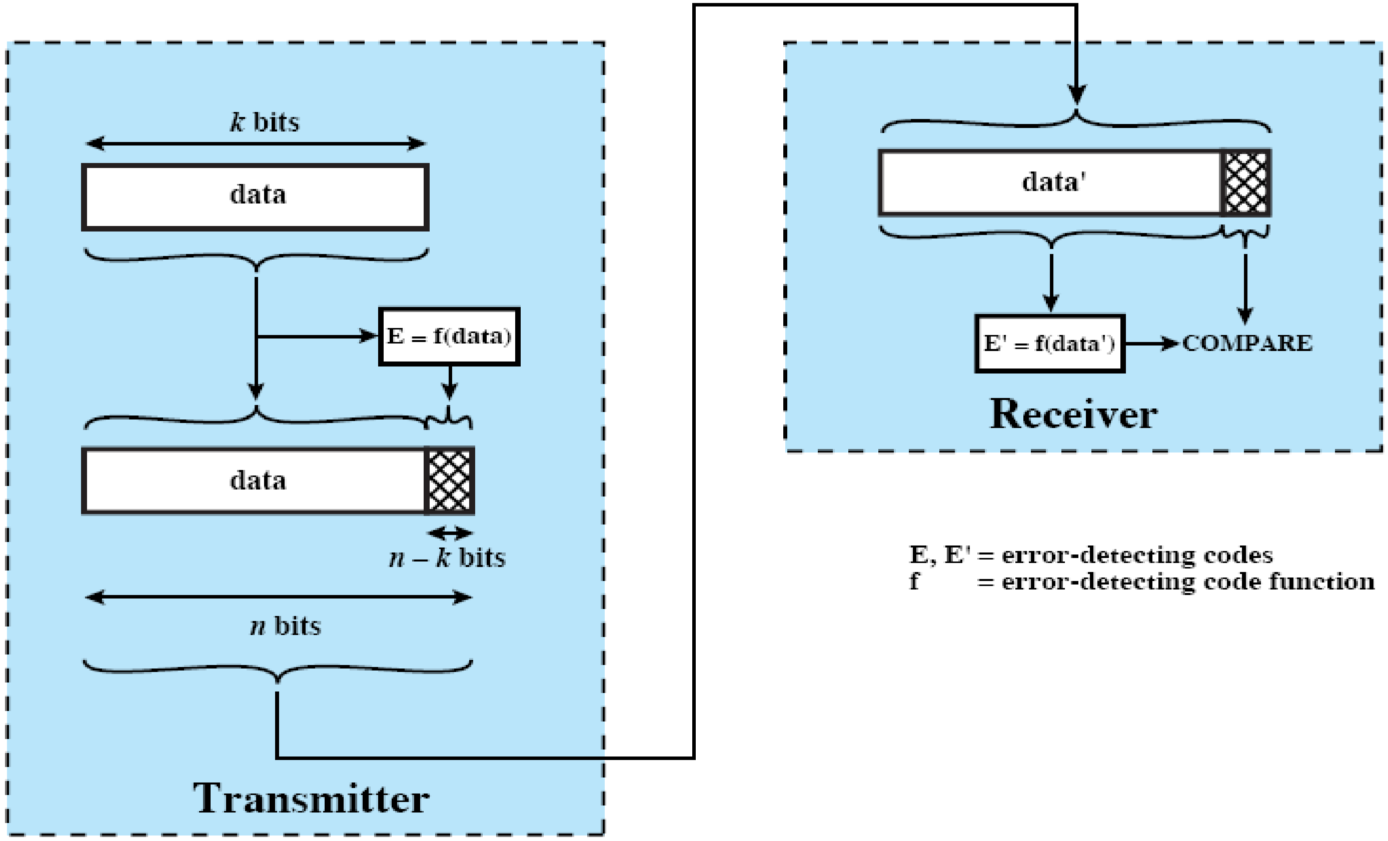
(a) Error-free transmission

(b) Transmission with losses and errors

Error detection

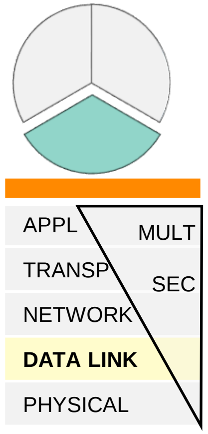


- Transmission error types:
  - Single-bit errors: a single bit changes, e.g. due to white noise
  - Burst errors: cluster of bits where errors occur, although not necessarily all the bits in the cluster suffer an error



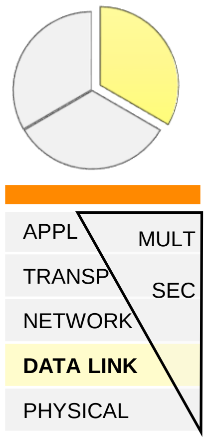
Error detection: receiver detects presence of errors with error-detecting codes

Transmission error example



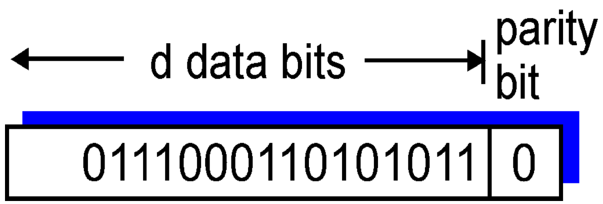
- **Burst errors**
  - Example of the impact of data rate on burst errors
    - An impulse noise event or a fading event of 1 µs occurs
    - At a data rate of 10 Mbps, there is a resulting error burst of 10 bits
    - At a data rate of 100 Mbps, there is a resulting error burst of 100 bits
- Let P_b be the probability that a bit is received in error (BER, bit error rate)
  - If we transmit a frame of F bits, the probability that the frame arrives with no bit errors is $(1-P_b)^F$

Error detection example: Parity check



Single bit parity

Detect single bit errors

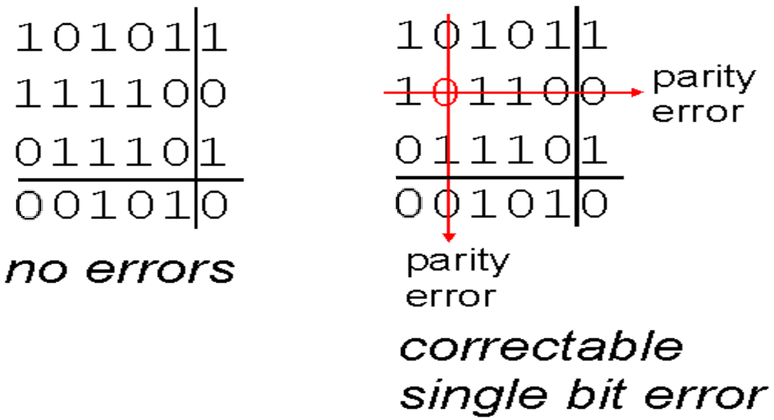


One-bit odd parity ~ total number of 1's in <data,parity bit> is odd

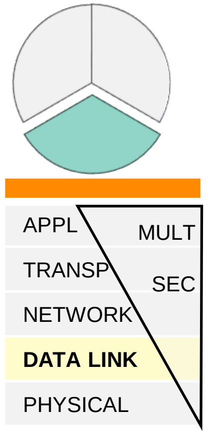
One-bit even parity ~ total number of 1's in <data, parity bit> is even

Two dimensional bit parity

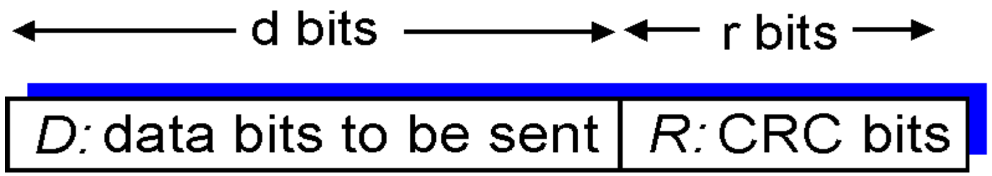
Detect *and correct* single bit errors



Error detection: Cyclic Redundancy Check (CRC)



- **View data bits, D, as a binary number**
- **Choose generator G (r+1 bit pattern)**
- **Goal: choose r CRC bits, R, such that**
  - <D,R> exactly divisible by G
    - Modulo 2 arithmetic: 1+1=0, 0+0=0, 1+0=1, 0+1=1
  - 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**
  - IEEE 802: CRC-32: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x^{1}+1$

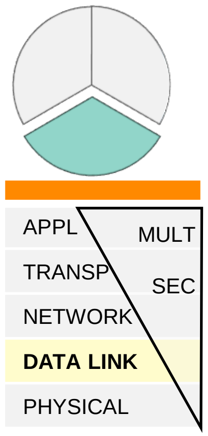


bit pattern

$D * 2^r$ XOR R

mathematical formula

CRC: Example (1)



Data <D>: <101110>

Want:

$D \cdot 2^r \text{ XOR } R = nG$

Equivalently:

$D \cdot 2^r = nG \text{ XOR } R$

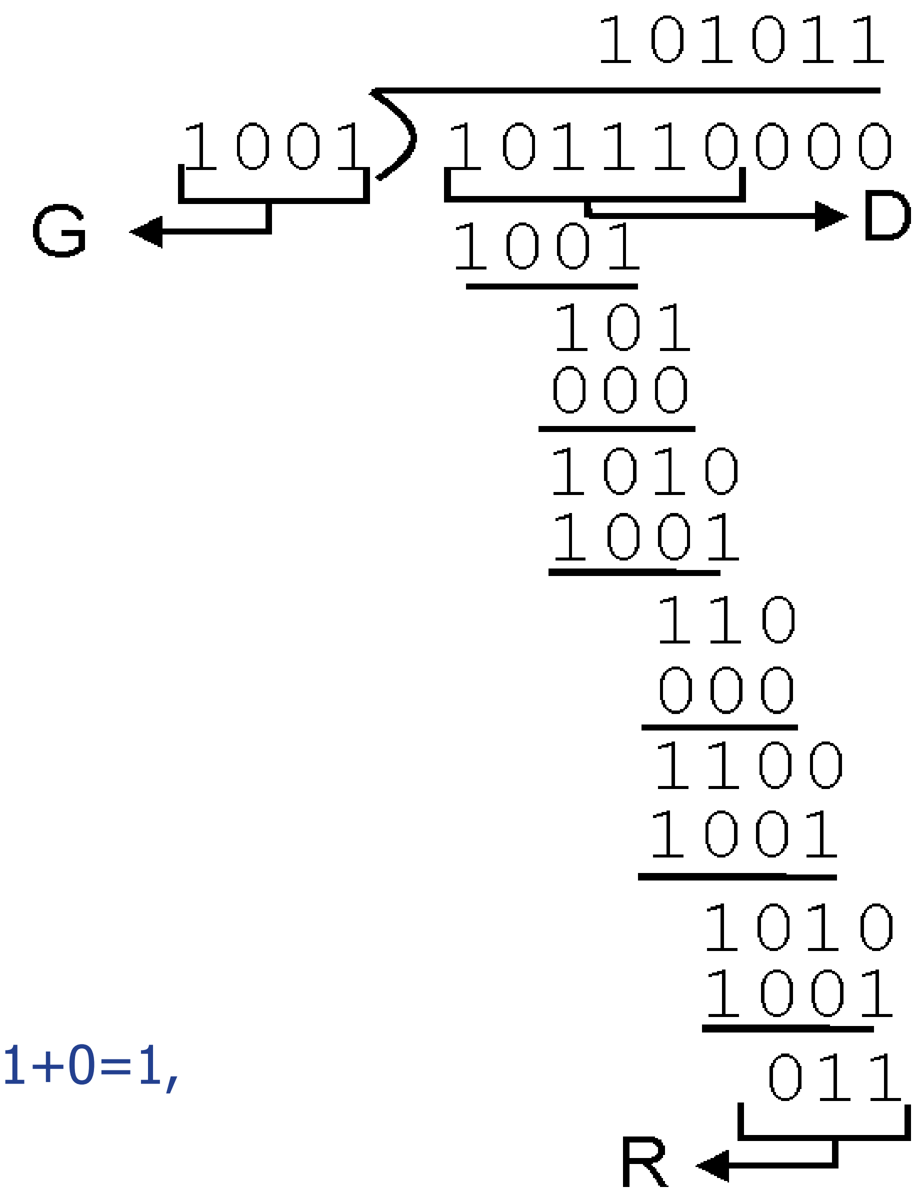
Equivalently:

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

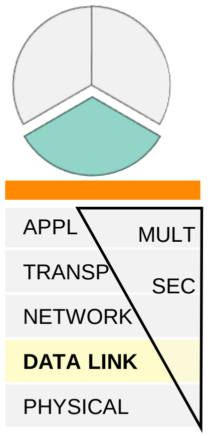
$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$

Transmitted frame <D,R>: 101110011

Obs! Modulo 2 arithmetic: 1+1=0, 0+0=0, 1+0=1, 0+1=1



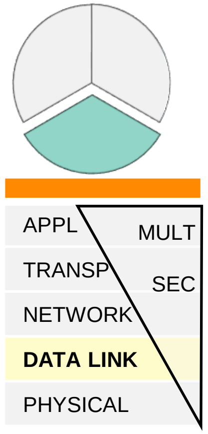
CRC: Example (2)



- Transmitted frame <D,R>: 101110011
- Received frame: 101110011

```
                101011
     1001   ) 101110011
              1001
               0101
     0000
                1010
     1001
                 0110
                 0000
                  1101
                  1001
        1001
                   1001
                   0000
```

Successful transmission



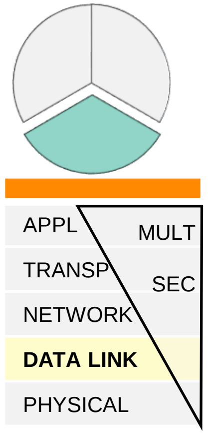
CRC: Example (3)

- Transmitted frame <D,R>: 101110011
- Received frame: 101110010

```
               101011
1001        101110010
            1001
             0101
             0000
              1010
              1001
               0110
               0000
                1101
                1001
                 1000
                 1001
                 0001
```

Transmission error!

CRC: Example (4)



A simplified example:

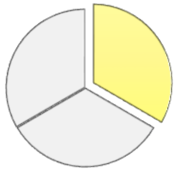
– **Say, I want to transmit an information/number 3000:**

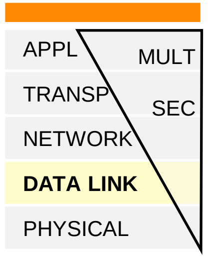
1. I divide it by a chosen number - like 7 for instance ...
   – Gives me 428 and a remainder of **4** (3000=428x7+4)
2. I take the remainder as my **checksum**
3. I transmit 3000 (information) followed by 4 (checksum).
4. A receiver gets it, takes the number 3000 divides it by 7
   – If remainder is 4 (checksum) then it assumes that no error has occurred
   – Otherwise, retransmit.

Medium Access Control (MAC)

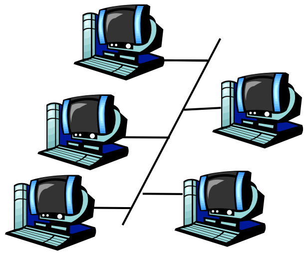
Medium access control

Two basic types of links and protocols:

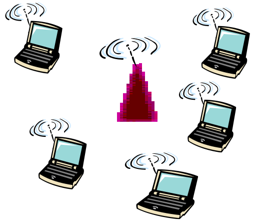




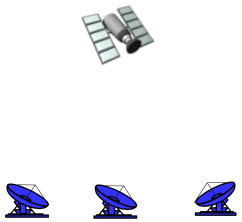
- **Broadcast links (shared wire or medium)**
  - Shared Ethernet
  - Wireless networks
    - → **Multiple access protocols**
- **Point-to-point links**
  - Dial-up access
  - xDSL (Digital Subscriber Line)
  - Switched Ethernet
    - → **Point-to-point protocols**



shared wire (e.g., cabled Ethernet)



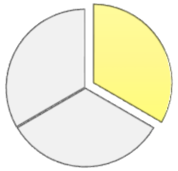
shared RF (e.g., 802.11 WiFi)

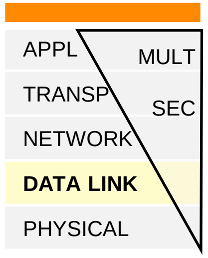


shared RF (satellite)

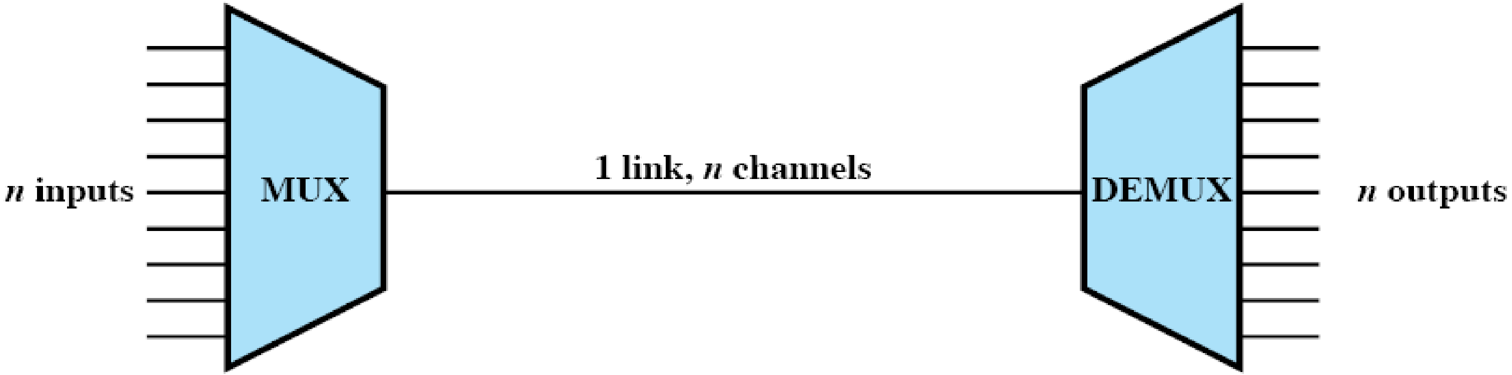
Analogy: humans at a cocktail party (shared air, acoustical)

Multiple access

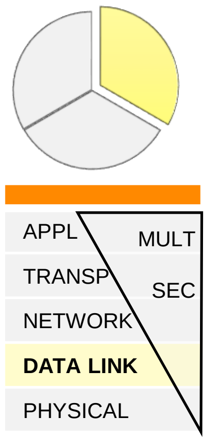




- **Most of today's wired and wireless technologies use multiple access technology**
- **Several nodes share a single shared communication channel**
  - Only one node can successfully send at a time
  - => Channel sharing principles need to be agreed
- **Multiple access protocol**
  - Distributed algorithm that determines how nodes share channel (i.e. when node can transmit)
  - Channel sharing communication must use the channel itself
    - No out-of-band channel for coordination



Multiple access protocols: Assumptions



1. Station model

- N independent stations (terminals, nodes), which generate frames for transmission

2. Single channel

- All stations transmit on and receive from the same single channel

3. Collisions

- Multiple transmitted frames overlapping in time interfere with each other resulting in garbled signal = collision
- All stations can detect collisions
- A collided frame must be retransmitted later

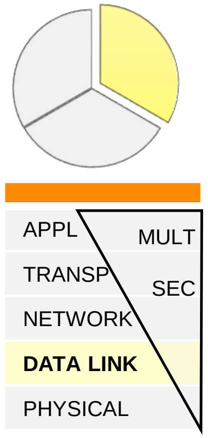
4. Time

- Continuous time: frame transmission can begin at any instant
- Slotted time: time is divided into discrete intervals

5. Carrier sense

- Yes: Stations can tell if the channel is in use before trying to use it
- No: Stations cannot sense the channel before trying to use

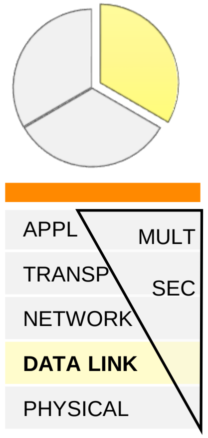
Multiple access protocols



– **Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit**
– **Communication about channel sharing must use the channel itself!**
  - No out-of-band channel for coordination

– **Ideal multiple access protocol:**
- Broadcast channel of rate R bps
  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 protocols: Three broad classes



- **Channel partitioning (static allocation) with multiplexing**
  - Divide channel into smaller "pieces"
    - Time
    - Frequency
    - Code
    - Wavelength
  - Allocate piece to node for exclusive use

Slides 35-39

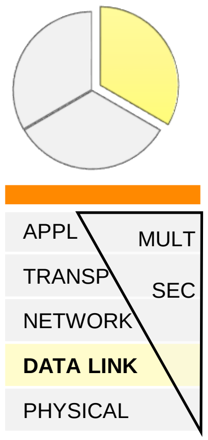
- **Random access (dynamic allocation)**
  - Channel not divided, allow collisions
  - "Recover" from collisions

Slides 40-49

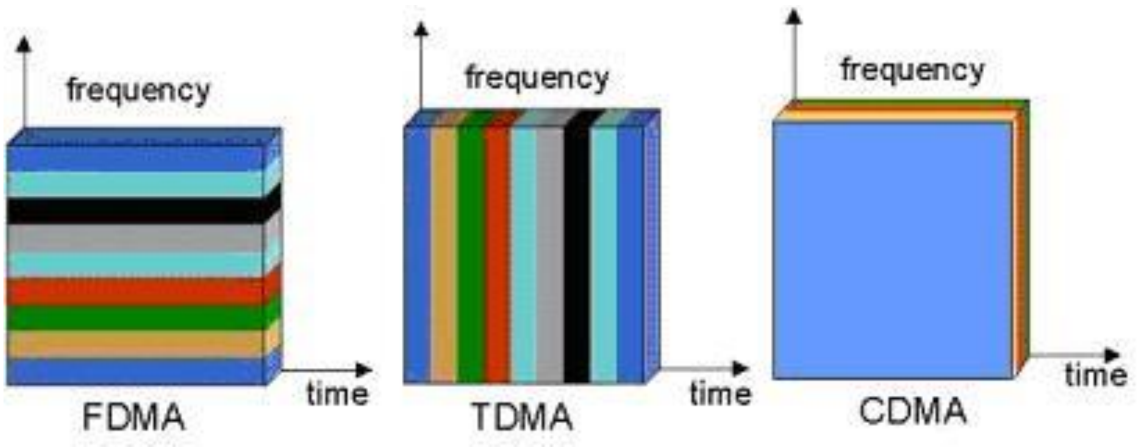
- **Dynamic allocation (taking turns)**
  - Nodes take turns
  - Nodes with more to send can take longer turns

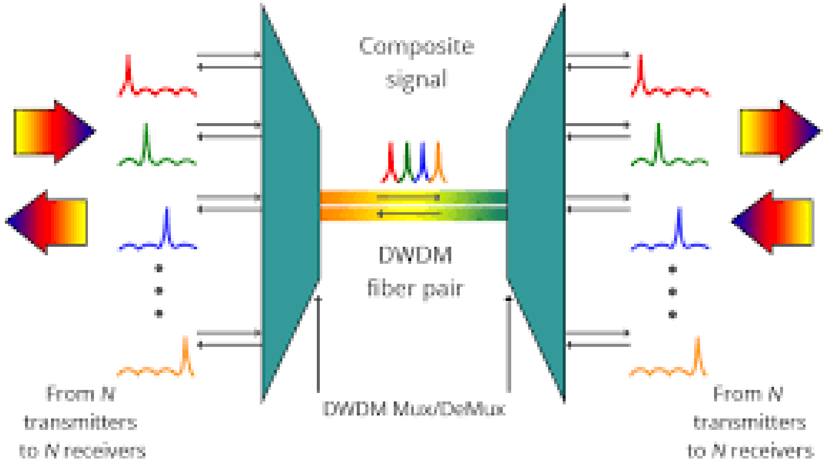
Slides 50-51

Channel partitioning MAC protocols



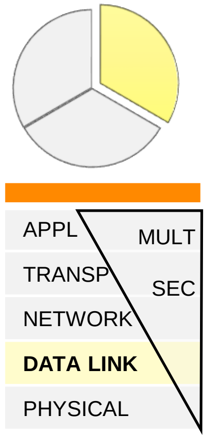
- **Widely used multiplexing techniques**
  - TDM (Time Division Multiplexing)
  - FDM (Frequency Division Multiplexing)
  - WDM (Wavelength Division Multiplexing)
  - CDM (Code Division Multiplexing)





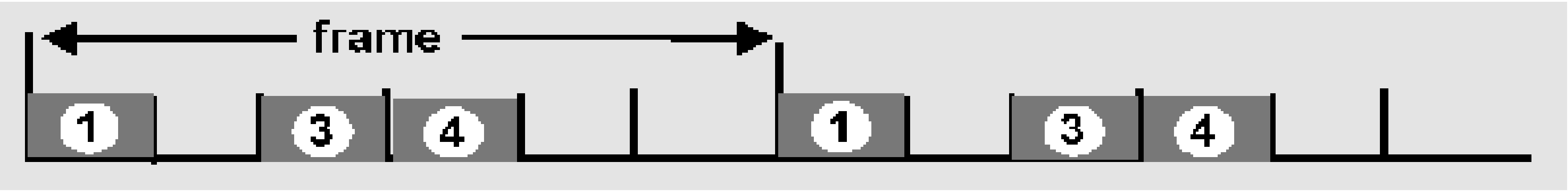
WDM

Channel partitioning MAC protocols: TDM

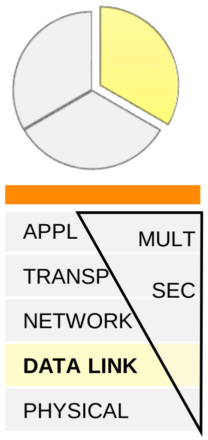


TDM (Time Division Multiplexing)

- Access to channel in "rounds"
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle
- Example: 6-station LAN
  - 1,3,4 have packet
  - Slots 2,5,6 idle

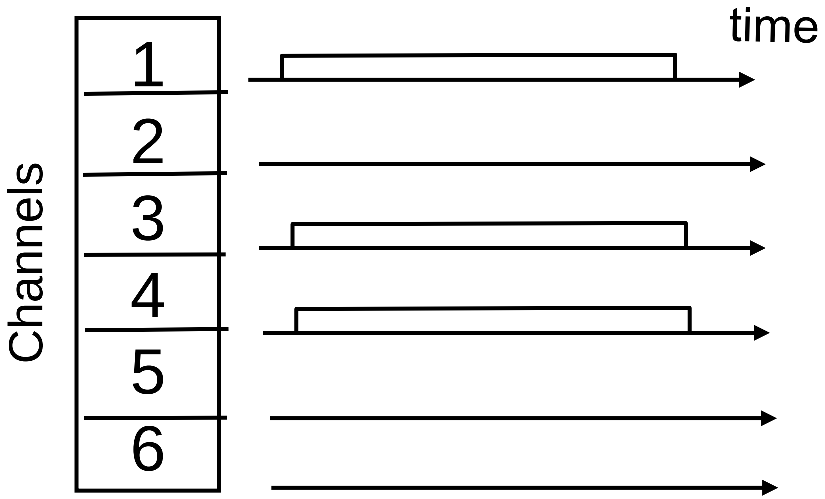


Channel partitioning MAC protocols: FDM

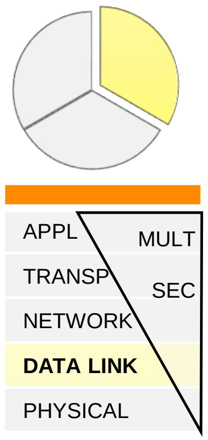


FDM (Frequency Division Multiplexing)

- **Available spectrum divided into channels**
- **Each station assigned fixed channel**
- **Unused transmission time in channels go idle**
- **Example: 6-station LAN**
  - Channels 1,3,4 have packet
  - Channels 2,5,6 idle

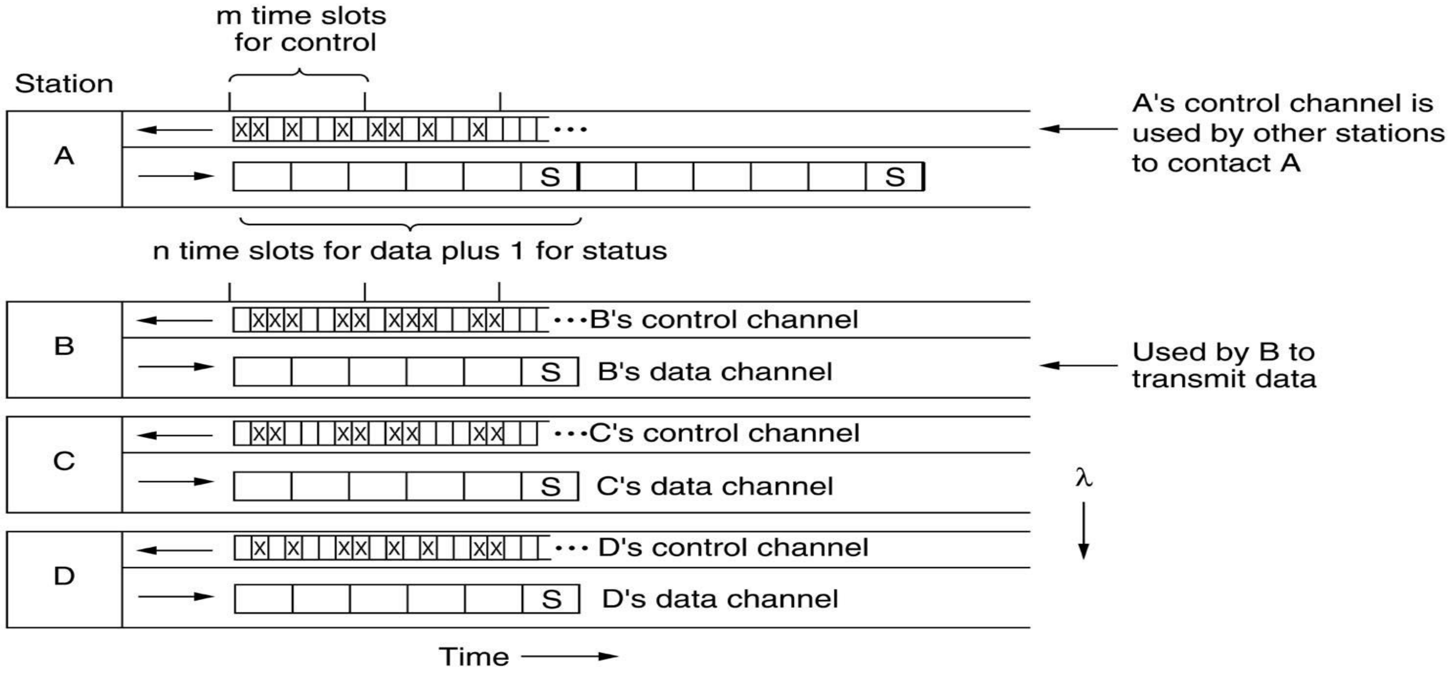


Channel partitioning MAC protocols: WDM

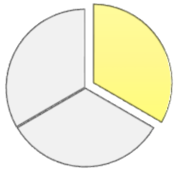


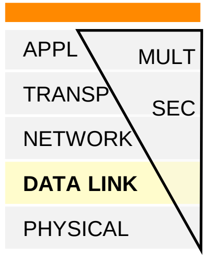
WDM (Wavelength Division Multiplexing)

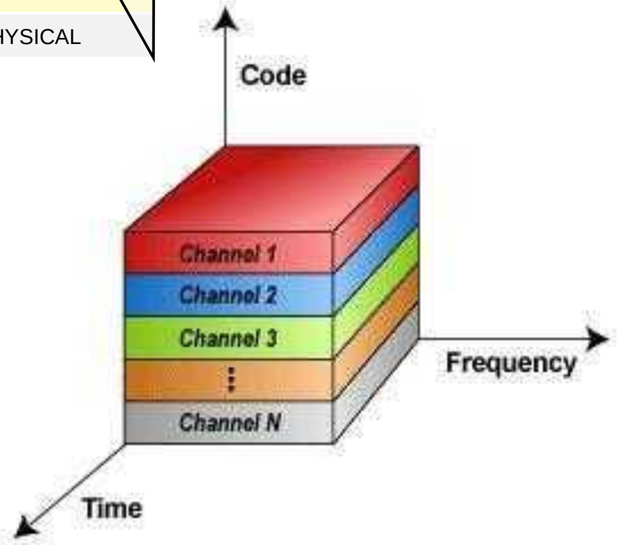
- Spectrum is divided into channels (wavelength bands)
- Each station is assigned two channels (control, data)
- Each channel is divided into groups of time slots
- Used in optical networks



Channel partitioning MAC protocols: CDMA





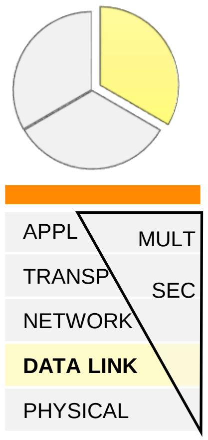


CDMA in which each channel is assigned a unique code which is orthogonal to codes used by other users.

CDMA (Code Division Multiple Access)

- **Unique "code" assigned to each user; i.e., code set partitioning**
- **Used mostly in wireless broadcast channels (cellular, satellite, etc.)**
- **All users share same frequency, but each user has own "chipping" sequence (i.e., code) to encode data**
- ***Encoded signal* = (original data) X (chipping sequence)**
- ***Decoding:* inner-product of encoded signal and chipping sequence**
- **Allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")**

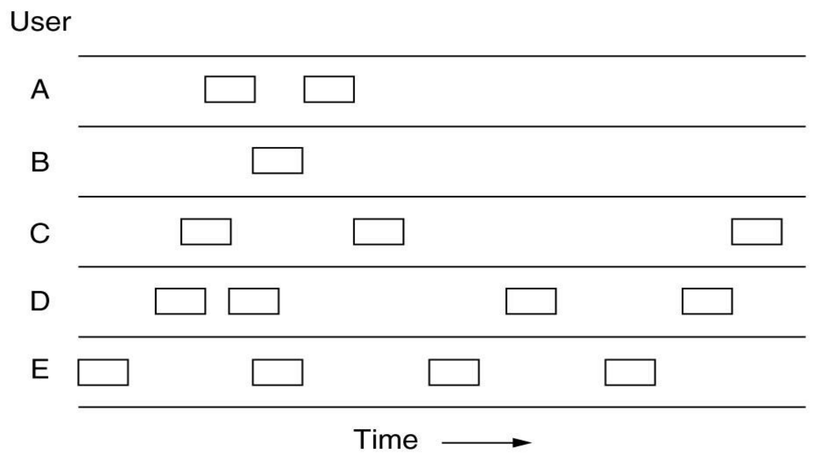
Random access protocols

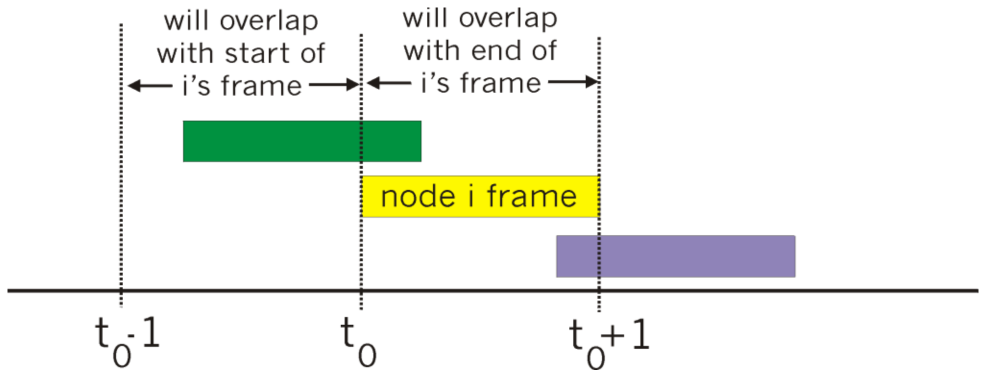


- **When node has packet to send**
  - Transmit at full channel data rate R
  - No *a priori* coordination among nodes
- **Two or more transmitting nodes → "collision"**
  - Resend after a random waiting time ("contend" for the channel)
- **Random access MAC protocol specifies**
  - How to detect collisions
  - How to recover from collisions (e.g., via delayed retransmissions)
- **Examples of random access MAC protocols**
  - Pure (unslotted) ALOHA and slotted ALOHA
  - CSMA, CSMA/CD (used in IEEE 802.3 Ethernet)
  - CSMA/CA (used in IEEE 802.11 WLAN)

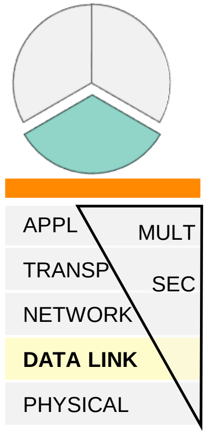
Random access protocols: Pure ALOHA (1)

- **Abramson 1970**
- **When user has frame to send, transmit immediately**
- **Collisions**
  - Frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$
  - If collision, resend after random waiting time
- **Simple**
- **No global synchronization between users**





Random access protocols: Pure ALOHA (2)



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

- **Suppose N nodes with many frames to send, each transmits with probability *p***
- **P (success by given node)**

= P (node transmits)

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

$\times$ P (no other node transmits in $[t_0, t_0+1]$)

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

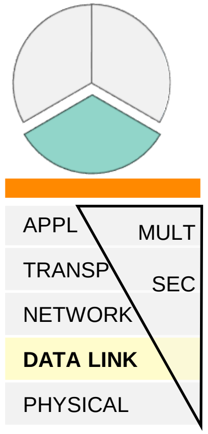
$$= p \times (1-p)^{2(N-1)}$$

… choosing optimum *p* and then letting *n* $\rightarrow$ infinity ...

$= 1/(2e) \approx$ 0.18

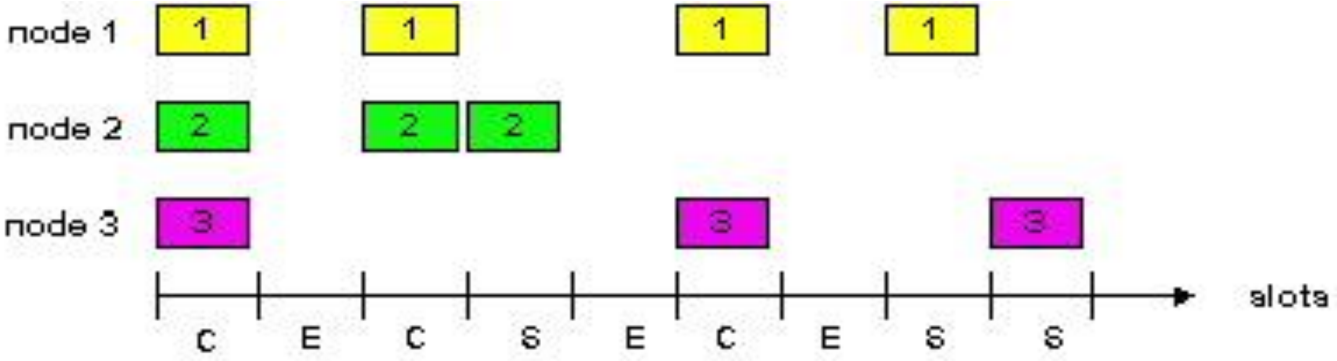
At best: channel used for useful transmissions only 18% of time!

Random access protocols: Slotted ALOHA (1)

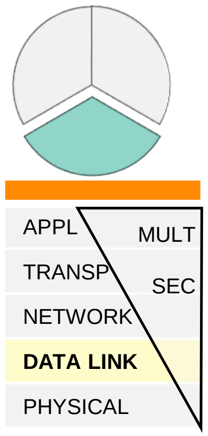


- **Assumptions**
  - All frames of same size
  - Time is divided into equal size slots, time to transmit 1 frame
  - Nodes start to transmit frames only at beginning of slots
  - Nodes are synchronized
  - If 2 or more nodes transmit in slot, all nodes detect collision

- **Operation**
  - When node obtains fresh frame, it transmits in next slot
  - No collision, node can send new frame in next slot
  - If collision, node retransmits frame in each subsequent slot with probability *p* until succe

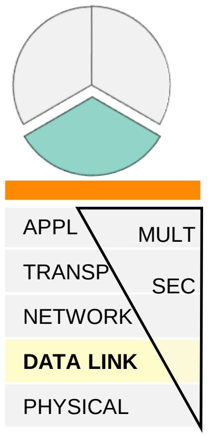


Random access protocols: Slotted ALOHA (2)



- **Pros**
  - Single active node can continuously transmit at full rate
  - 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 between nodes

Random access protocols: Slotted ALOHA (3)



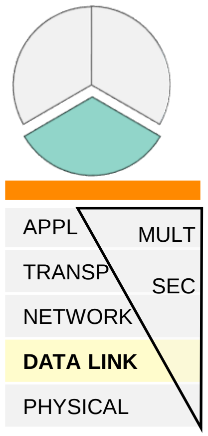
Efficiency is the long-run fraction of successful slots when there are many nodes, each with many frames to send

- **Suppose N nodes with many frames to send, each transmits in slot with probability *p***
- **Probability that node 1 has success in a slot = $p(1-p)^{N-1}$**
- **Probability that any node has a success = $Np(1-p)^{N-1}$**
- **For max efficiency with N nodes, find p* that maximizes $Np(1-p)^{N-1}$**
- **For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity**

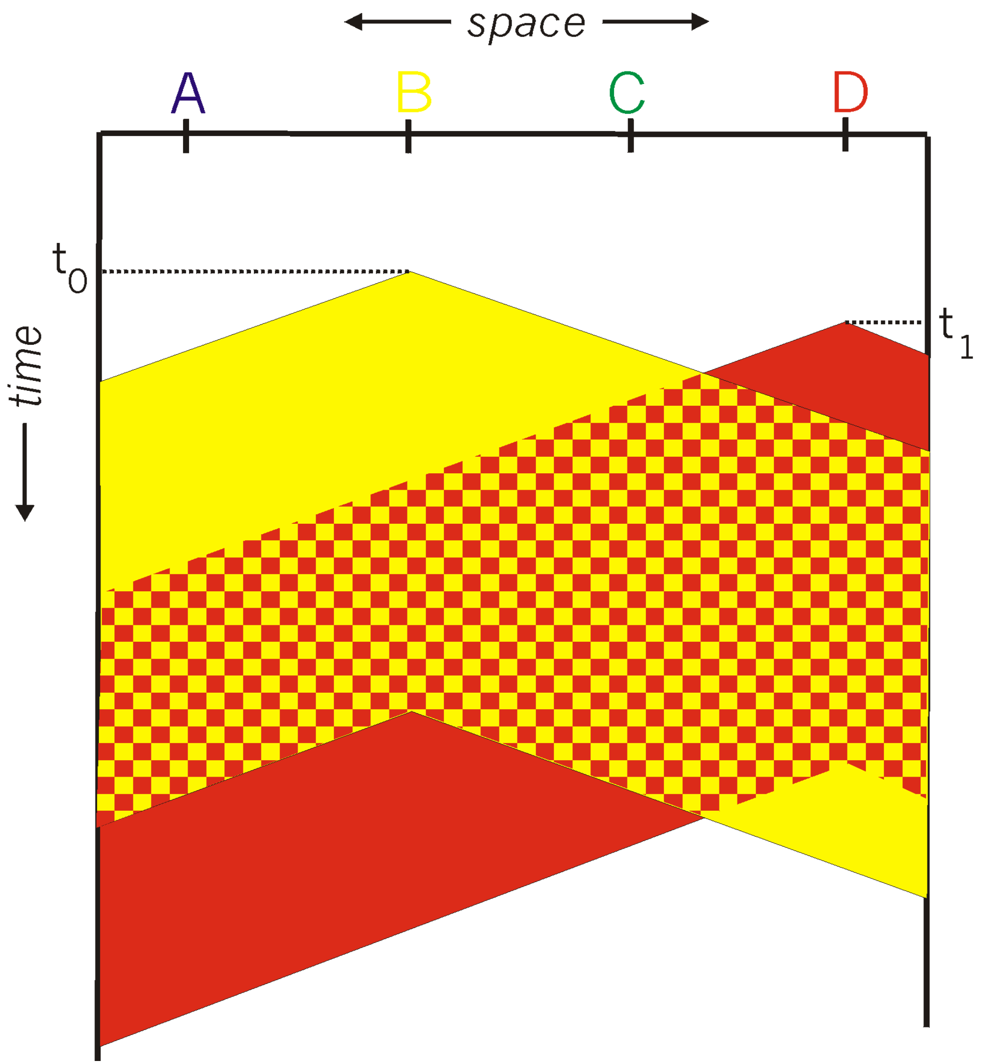
... gives $1/e \approx$ 0.37

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

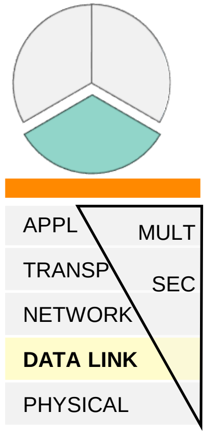
Random access protocols: CSMA (1)



- **CSMA (Carrier Sense Multiple Access): Listen before transmit**
  - If channel sensed idle, transmit entire frame
  - If channel sensed busy, defer transmission for random waiting time
  - Human analogy: don’t interrupt others!
- **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 node distance and propagation delay in determining collision probability

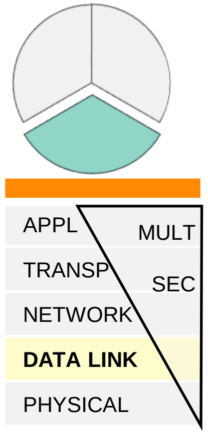


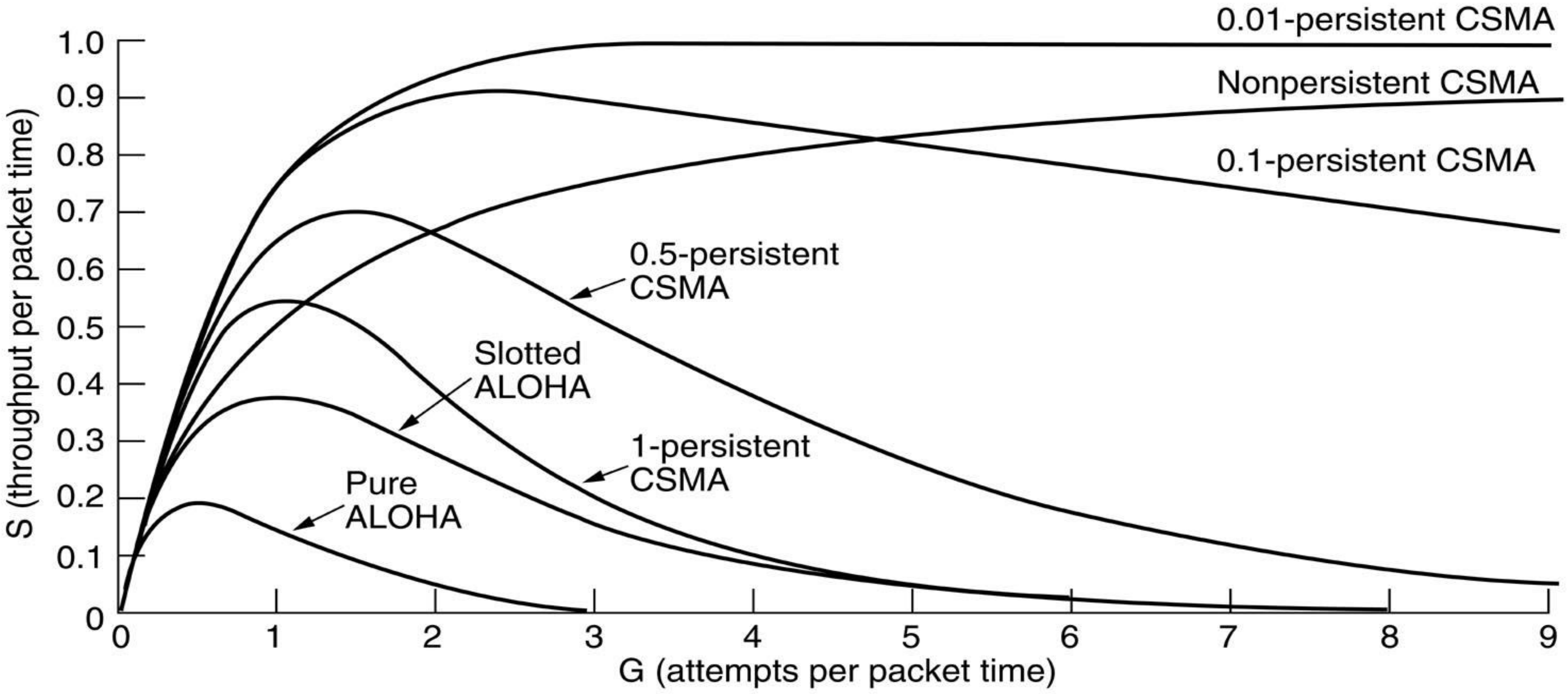
Random access protocols: CSMA (2)



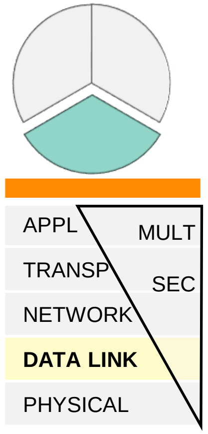
- **1-persistent CSMA**
  - Transmits with probability of 1 if the channel is sensed idle
  - If channel sensed busy, waits until free and sends immediately
- **Nonpersistent CSMA**
  - Similar to 1-persistent in case of idle channel
  - If channel sensed busy, fixed wait-time before re-sensing
- **p-persistent CSMA (for slotted channels)**
  - Channel sensed idle $\rightarrow$ transmit with probability p
    (i.e. defer to next slot with probability q=1-p)
  - Channel sensed busy $\rightarrow$ random wait before re-sensing

Random access protocols:Performance comparison between ALOHA & CSMA



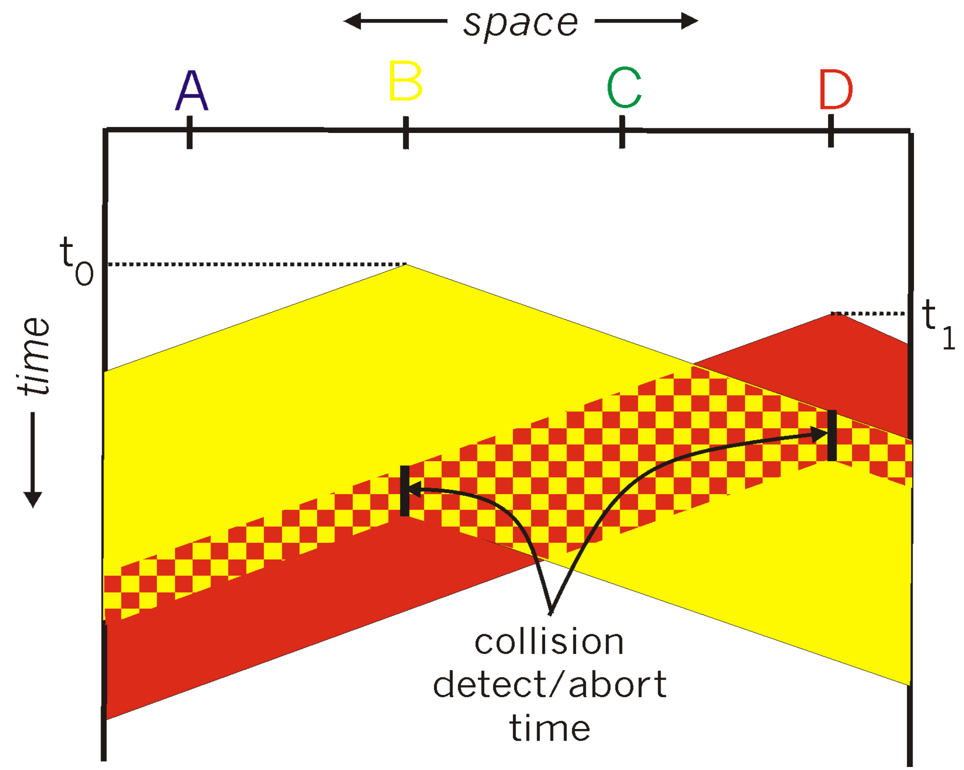


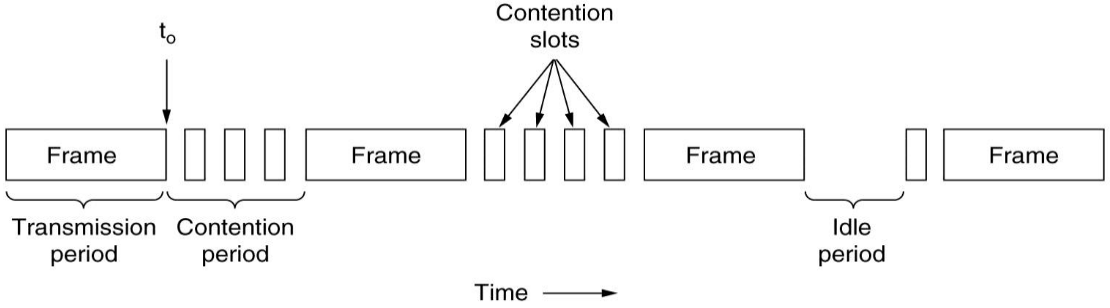
Random access protocols: CSMA/CD



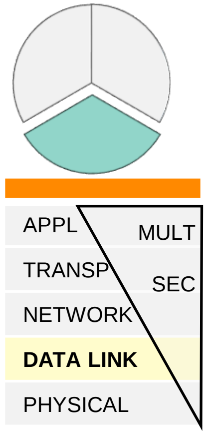
- **Carrier sensing with Collision Detection, deferral as in CSMA**
  - Collisions *detected* within short time
  - Colliding transmissions aborted (reducing channel wastage)
  - Human analogy:
    the polite conversationalist

- **Collision detection**
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs: receiver shut off while transmitting



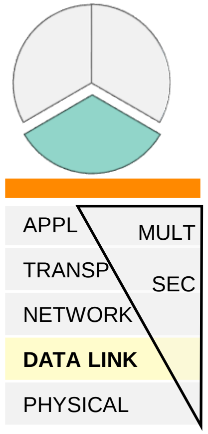


“Taking turns” MAC protocols (1)



- **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 (2)

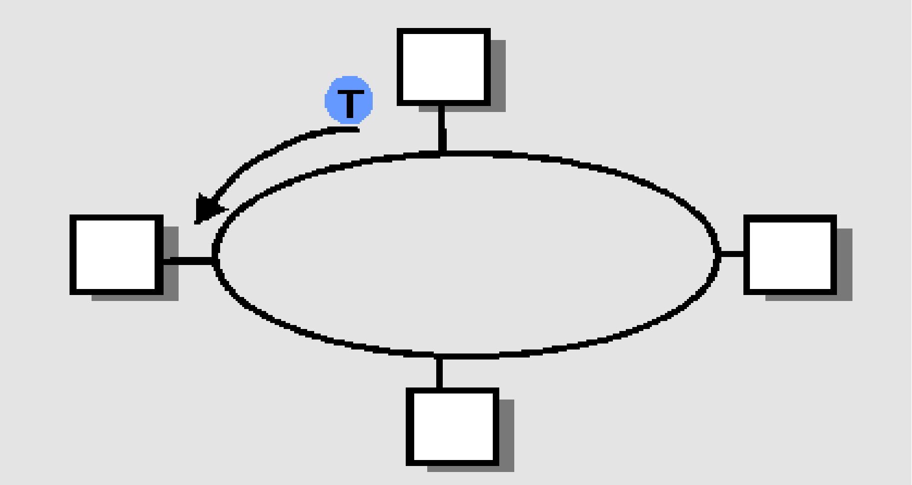


Polling

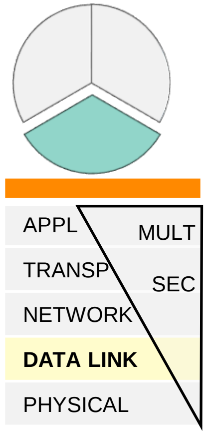
- Master node “invites” slave nodes to transmit in turn
- Concerns
  - Polling overhead
  - Latency
  - Single point of failure (master)

Token passing

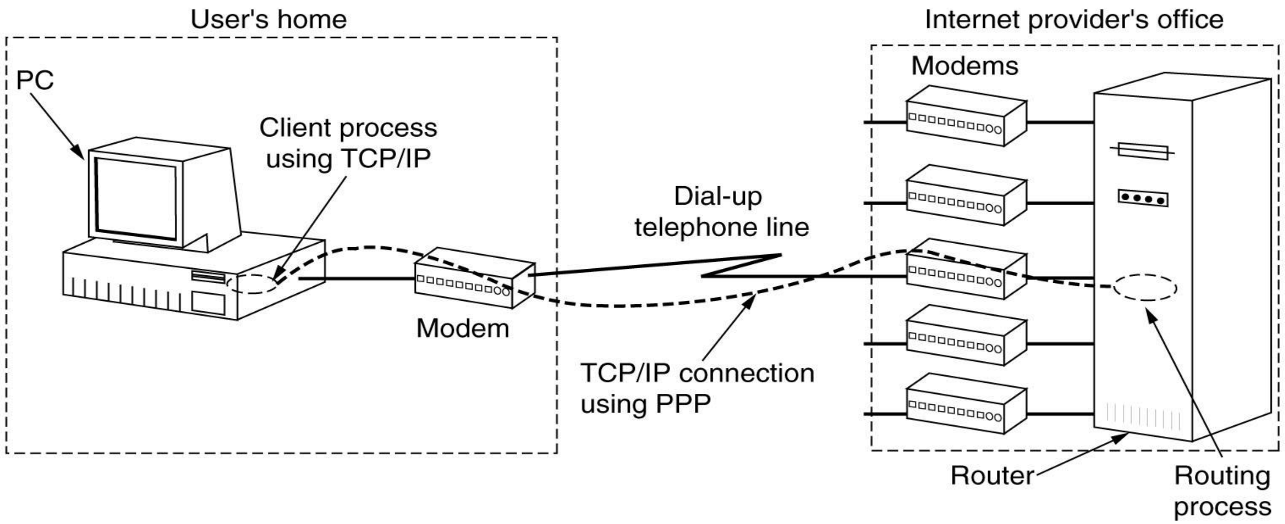
- Control token passed from one node to next sequentially
- Token message
- Concerns
  - Token overhead
  - Latency
  - Single point of failure (token)



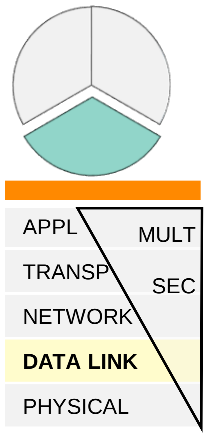
PPP: Point-to-point protocols



- **One sender, one receiver, one link**
- **Easier than shared broadcast link**
  - No medium access control
  - No need for explicit MAC addressing
  - E.g., dialup link, ISDN line



PPP: Variants



- **PPP over SONET/SDH (RFC 2615)**
  - Used for POS (Packet over SONET/SDH) communication
  - E.g. in FUNET 2.5 Gbps and 622 Mbps backbone links
- **PPPoE (PPP over Ethernet)**
  - Encapsulate PPP frames inside Ethernet frames
  - Used mainly with xDSL services
- **PPPoA (PPP over ATM)**
  - Encapsulate PPP frames inside ATM frames
  - Used mainly with xDSL services

Key points to remember

1. **Architecture and basic concepts of data link layer:**
   - Transmission media types
   - Synchronous vs asynchronous transmission
   - Framing
   - Error detection

2. **Medium access control (MAC) protocols and their basic principles**
   - Multiple access protocols
     - Channel partitioning protocols
     - Random access protocols
     - “Taking turn”-protocols
   - Basics of point-to-point protocols

Thank you!