CS2105

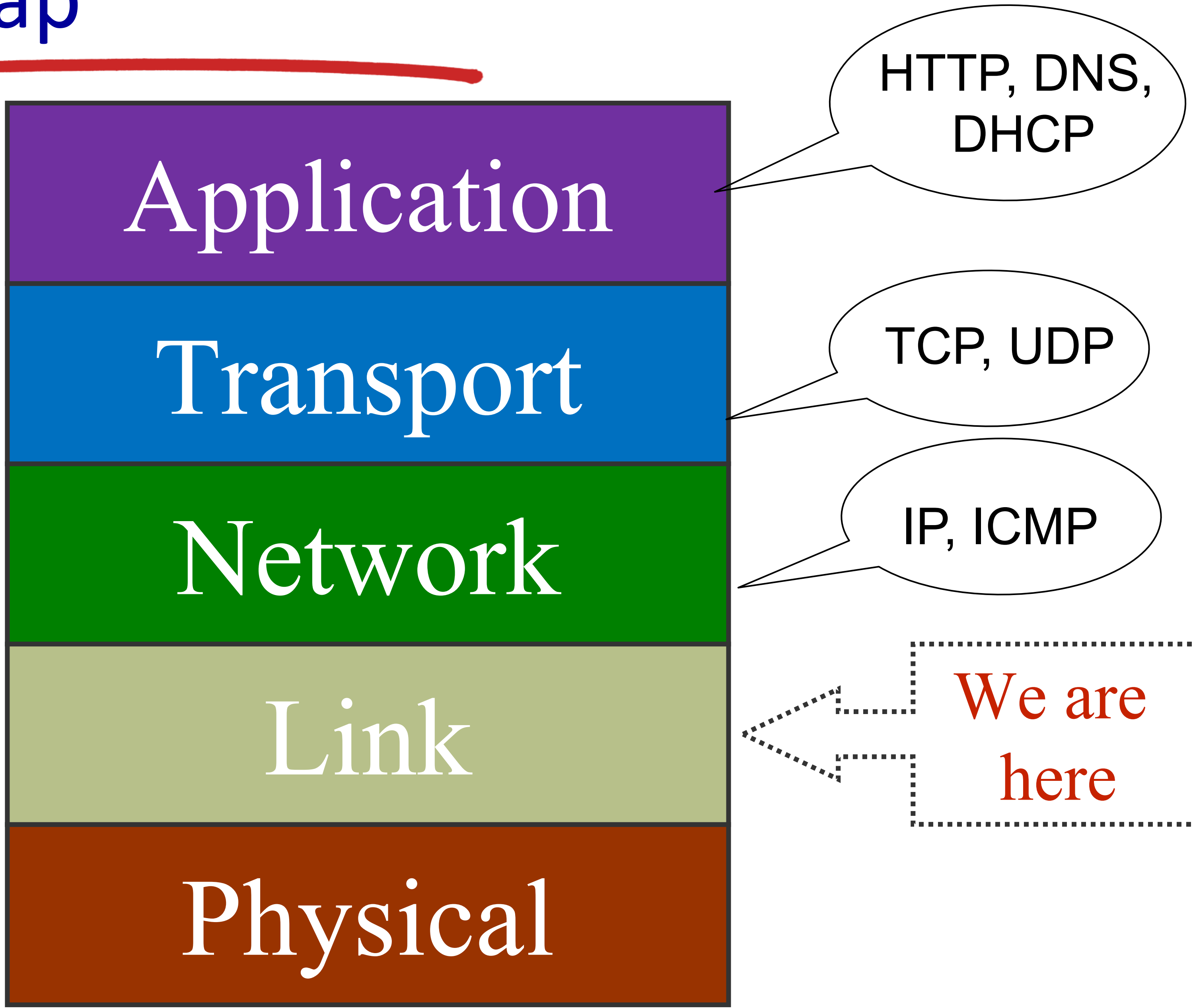
An *Awesome* Introduction to Computer Networks

The Link Layer

National University of Singapore
Department of Computer Science
School of Computing

Adopted from Prof Roger

Recap


Application
Transport
Network
Link
Physical
HTTP, DNS, DHCP
TCP, UDP
IP, ICMP
We are here

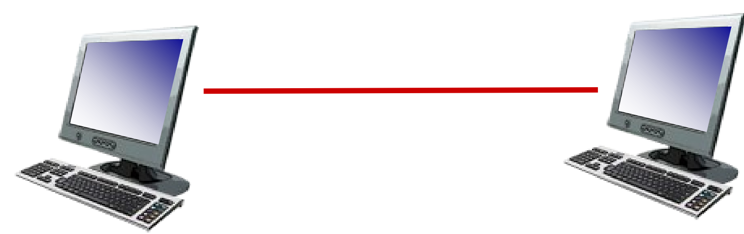
Motivation

- Let us look at networking from Bottom-up perspective
- Assuming, we have figured out the electronics of sending and receiving binary data over a *communication channel*
- *Aim:* Send data between 2 nodes via cable.
- *Solution:* Connect the 2 nodes and send data

Jargon Alert:

- **Communication channel**: the transmission medium of the data signals. E.g. copper wire, optical fiber, terrestrial radio, Satellite, etc.
- **Node**: Devices exchanging data. E.g. hosts, routers, etc.

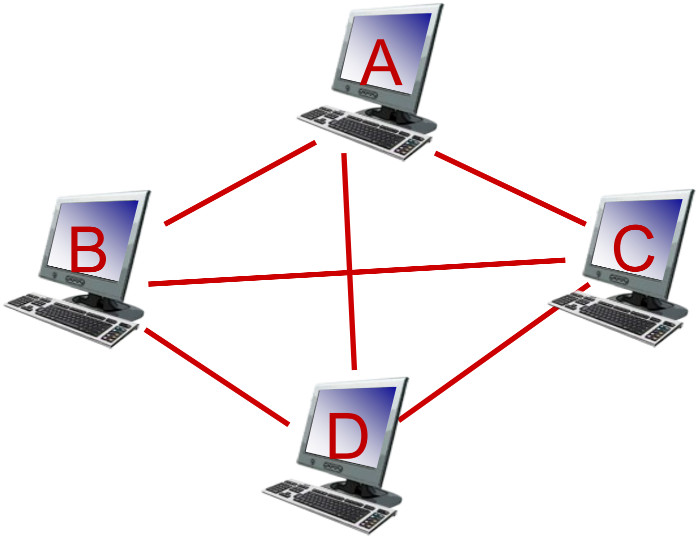
Physical



Motivation

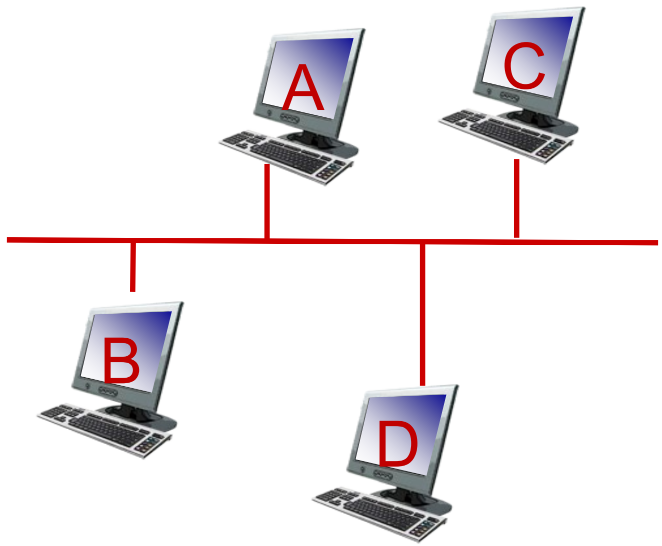
Jargon Alert:
- Link: *Communication channels that connect adjacent nodes.*

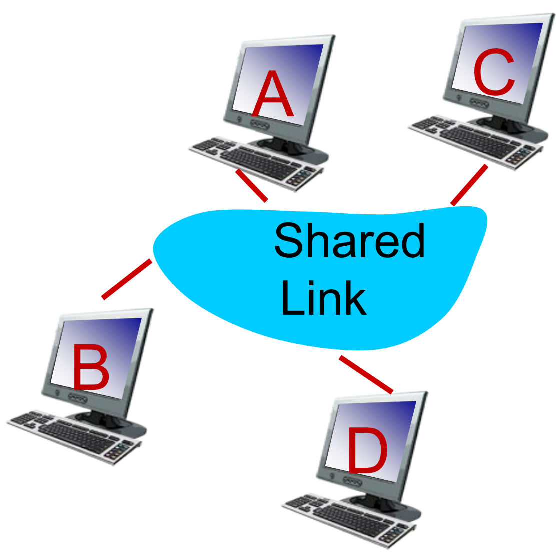
- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes and send data
  - Each link needs to be *addressed*
  - *Drawback*: Does not scale
    - N-1 links needed



Motivation

- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*





The Link Layer

After the next set of lectures, will understand:

- The role of link layer and the services it could provide.
- How parity and CRC scheme work.
- Different methods for accessing shared medium.
- How ARP allows a host to discover the MAC addresses of other nodes in the same subnet.
- The role of switches in interconnecting subnets in a LAN.

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

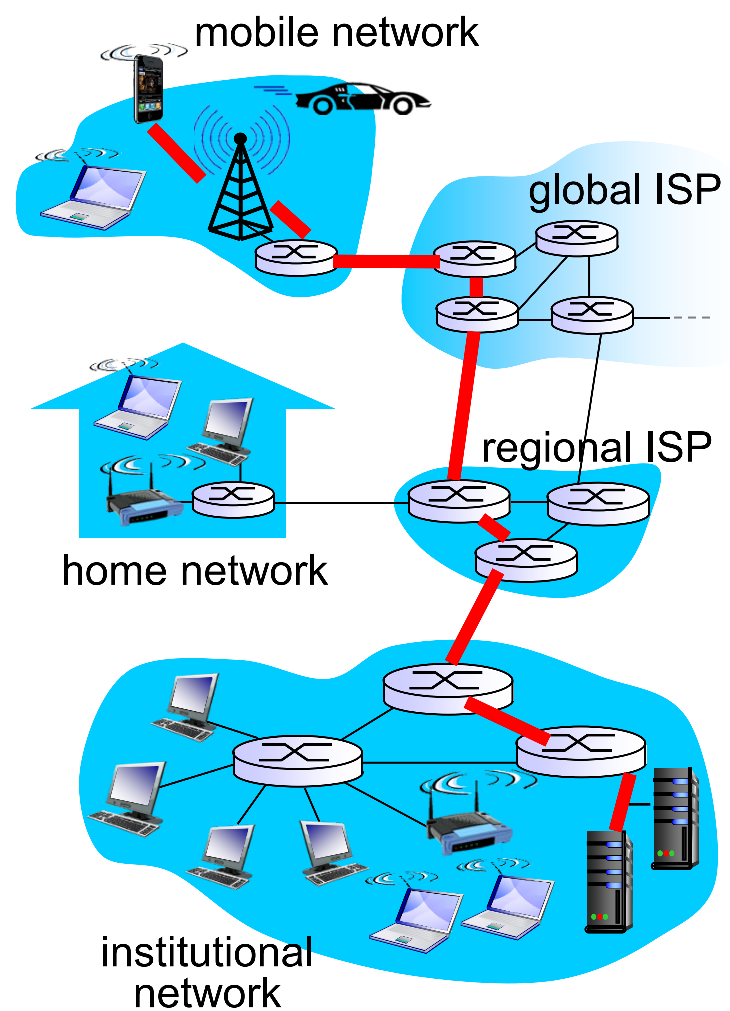
6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

Kurose Textbook, Chapter 6
(Some slides are taken from the book)

Link Layer: Introduction (1/2)

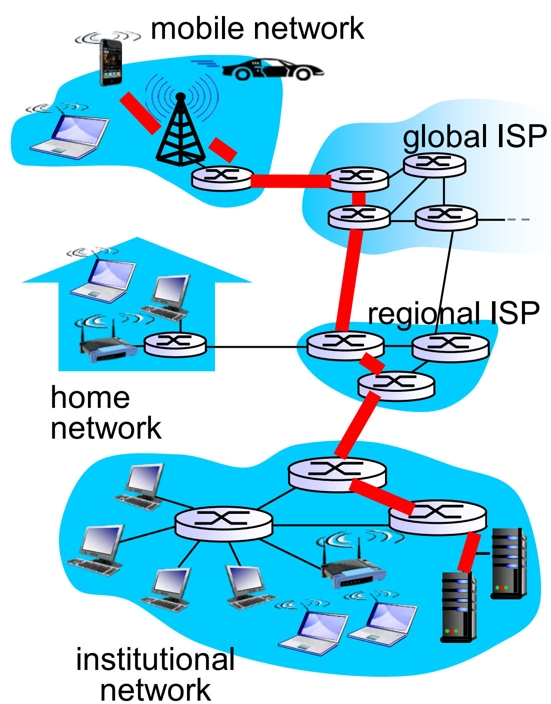
- *Network layer* provides communication service between any two hosts.
- An IP datagram may travel through multiple routers and links before it reaches destination.



Link Layer: Introduction (2/2)

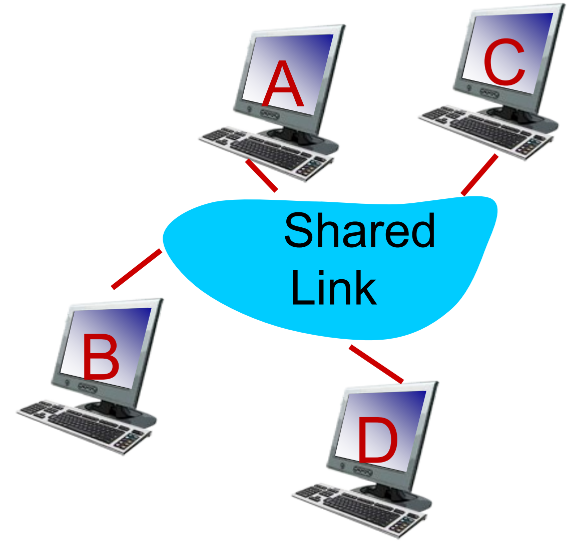
- *Link layer* sends datagram between adjacent nodes (hosts or routers) over a single link.
  - IP datagrams are encapsulated in link-layer frames for transmission.
  - Different link-layer protocols may be used on different links.
    - each protocol may provide a different set of services.

Jargon Alert:
- **Adjacent**: *A single hop connects the two nodes.*
- **Frame**: *layer-2 packet.*



data-link layer has responsibility of transferring datagram from one node to *physically adjacent* node over a link

Motivation (revisited)



- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Framing

Link Access Control

Reliability

Detection

Possible Link Layer Services (1/2)

- Framing
  - Encapsulate datagram into frame, adding header and trailer.

- Link access control
  - When multiple nodes *share* a single link, need to coordinate which nodes can send frames at a certain point of time.

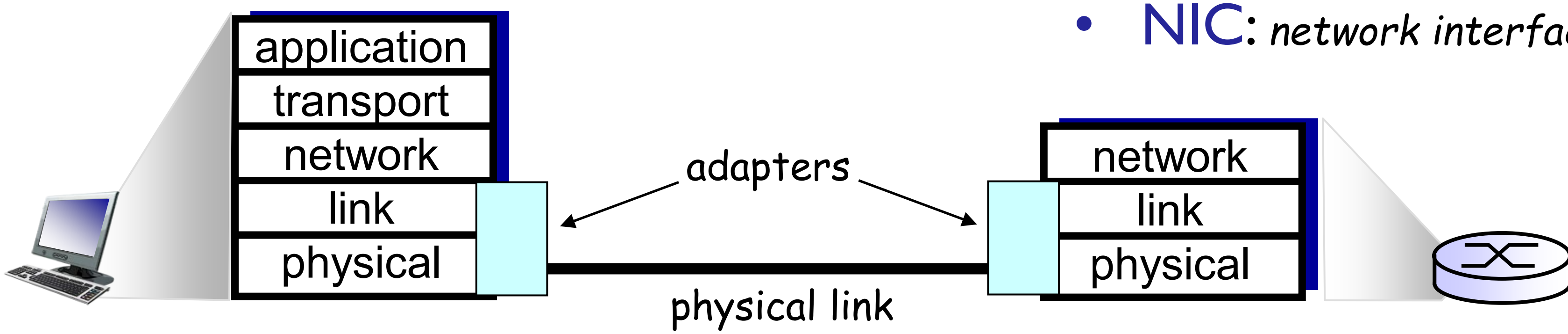
humans at a cocktail party (shared air)

Possible Link Layer Services (2/2)

- Error detection
  - Errors are usually caused by signal attenuation or noise.
  - Receiver detects presence of errors.
    - *may signal sender for retransmission or simply drops frame*
- Error correction
  - Receiver identifies and corrects bit error(s) without resorting to retransmission.
- Reliable delivery
  - Seldom used on low bit-error link (e.g., fiber) but often used on error-prone links (e.g., wireless link).

Link Layer Implementation

Jargon Alert:
- NIC: *network interface card.*



- Link layer is implemented in "adapter" (aka NIC) or on a chip.
  - E.g., Ethernet card, Wi-Fi adapter
- Adapters are semi-autonomous, implementing both link & physical layers.

Roadmap

6.1 Introduction to the Link Layer

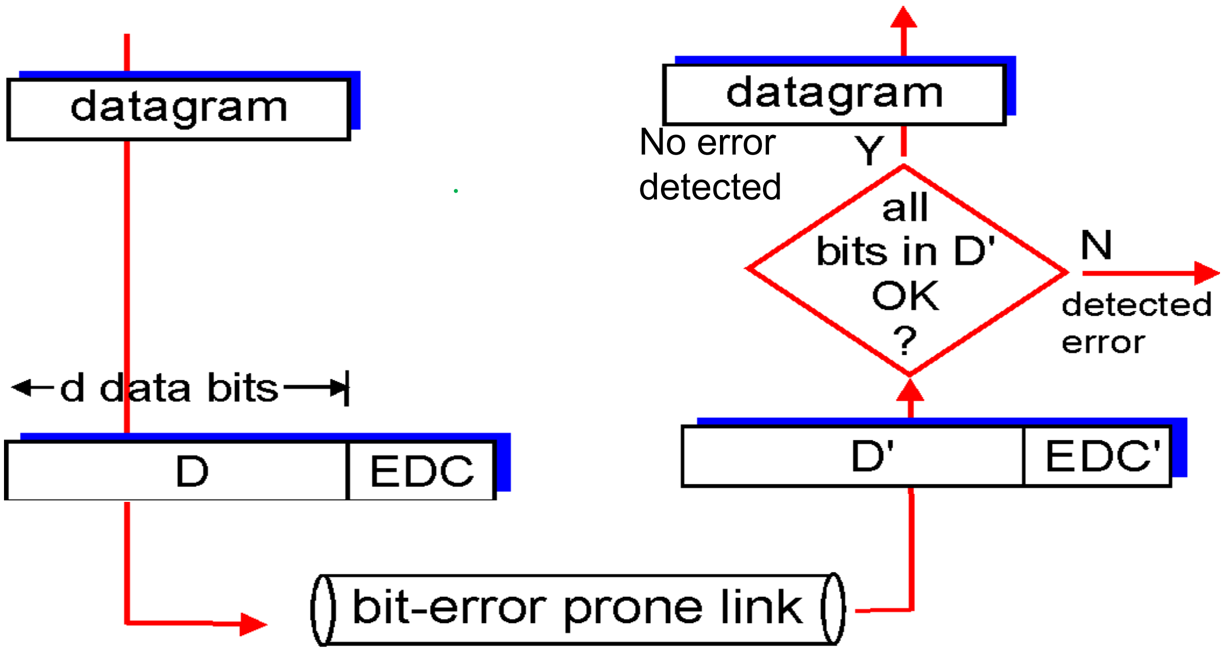
6.2 Error Detection and Correction

- 6.2.1 Parity Checks
- 6.2.3 Cyclic Redundancy Check (CRC)

6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

Error detection



Notation:

- EDC: *Error Detection and Correction bits*
- D: *Data protected by error checking, may include header fields*

- Error detection schemes are not 100% reliable!
  - may miss some errors, but rarely.
  - Usually, larger EDC field yields better detection (and even correction).

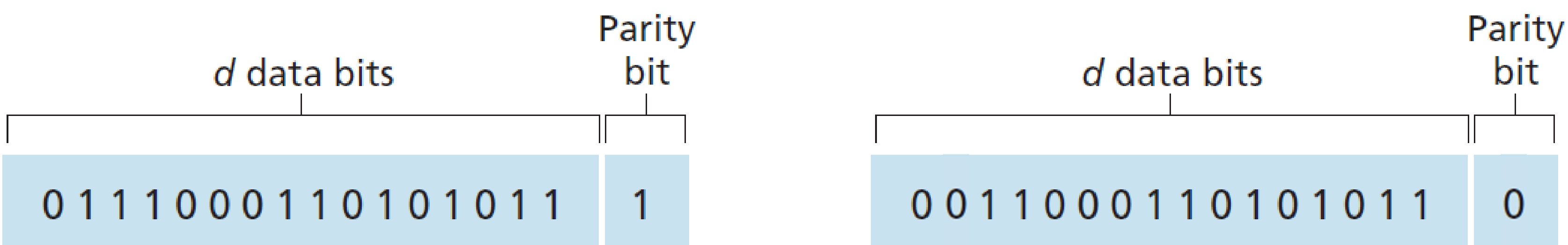
Error Detection

- Popular error detection schemes:
  - Checksum (used in TCP/UDP/IP)
  - Parity Checking
  - CRC (commonly used in link layer)

- Checksum (review)
  - treat segment contents as sequence of 16-bit integers
  - *checksum*: 1's complement of the sum of segment contents

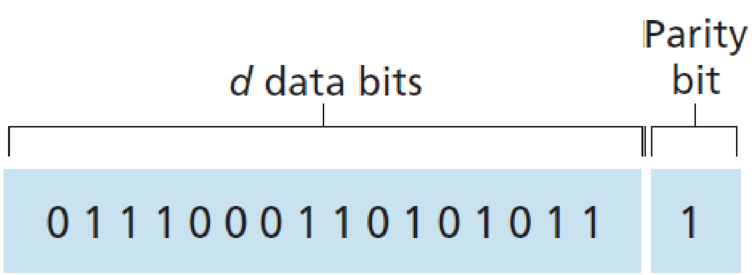
Parity Checking: Single bit

- Suppose that the information to be sent, *D*, has *d* bits.
- In an *even parity* scheme,
  - the sender simply includes one additional bit
  - chooses its value such that the total number of 1s in the *d + 1* bits is even



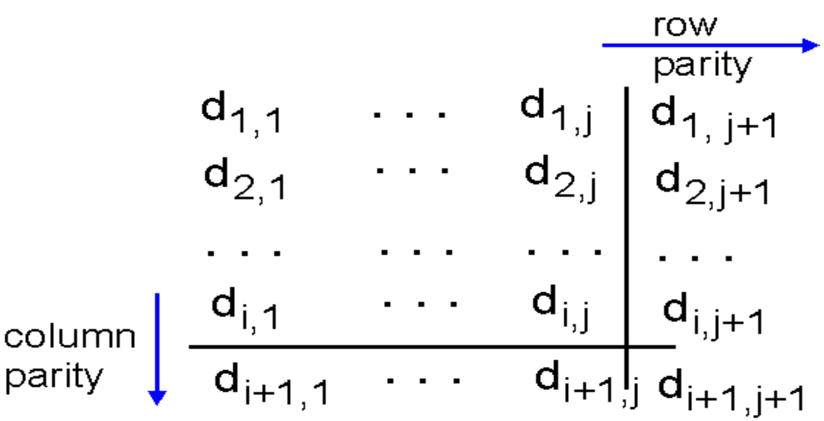
Parity Checking: Single bit

- Can *detect* single bit errors in data.
  - Actually, can detect odd number of single bit errors
  - Cannot detect even number of single bit error
- Works exceptionally well (Mathematically)
  - Probability of multiple bit errors is low (if errors are independent)
- However, errors are often clustered together in "bursts."
  - The probability of *undetected errors* in a frame can approach *50%*



Parity Checking 2-D

- the *d* bits in *D* are divided into *i* rows and *j* columns.
- A parity value is computed for each row and for each column.
  - The resulting *i + j + 1* parity bits comprise the link-layer frame's error-detection bits



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

Parity bit for the column and row parity bits

$$d_{i+1,1} \ldots d_{i+1,j} d_{1,j+1} \ldots d_{i,j+1}$$

$$d_{i+1,j+11}$$

Parity Checking 2-D

- Can *detect and correct* single bit errors in data.

$$\begin{array}{ccccc|c} 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

$$\begin{array}{ccccc|c} 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

Parity error

Parity error

- Can *detect* any two-bit error in data.

$$\begin{array}{ccccc|c} 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

Cyclic Redundancy Check: Motivation

- We want to transfer a non-binary number D without error.
- R: the r digit error detection code.

D: data to be sent	R:code
d digit number	r digits

- *Aim:* We need to generate R such that
  - The sender can compute R easily.
  - The receiver can verify the integrity of D easily.
- *Solution:* Let us use the mathematical properties of "division".
  - We shall use a special r digit number G, called the "Generator".

Cyclic Redundancy Check: Motivation

- Let $D = 21027845$, $r = 3$ and $G = 401$.
- Create a new number X by appending r $9's$ to D
  - $X = 21027845\,\mathbf{999}$
  - Mathematically, $X = D \times 10^r + (10^r - 1)$
- Find the remainder y of $\frac{X}{G}$
  - $y = X \,\%\, G$
  - $y = 281$
- The message M being transmitted is
  - $M = X - y$
  - $M = 21027845999 - 281 = 21027845718$
  - M is divisible by G

Cyclic Redundancy Check: Motivation

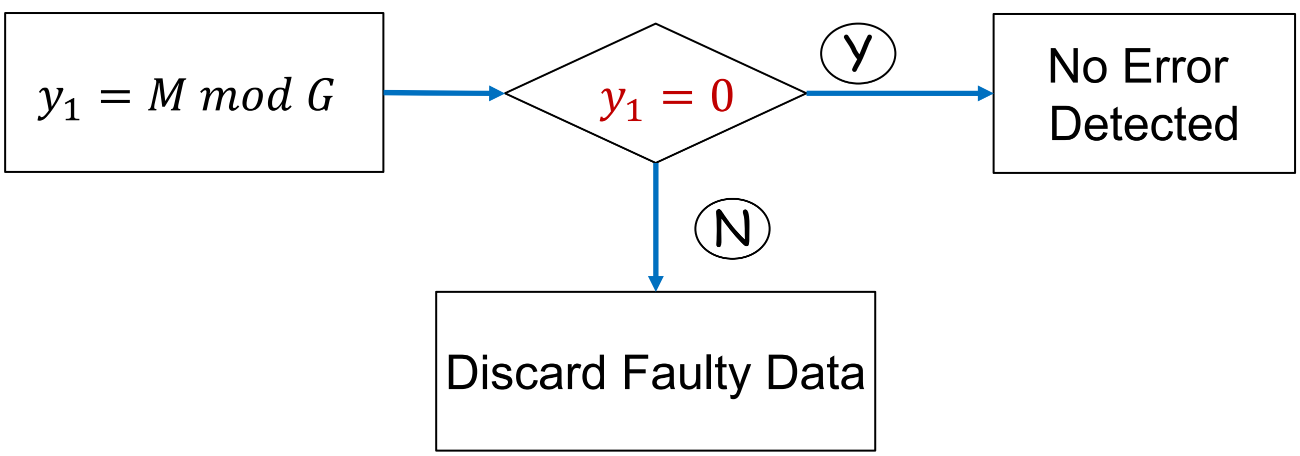
- $M = \underbrace{21027845}_{\text{8 digit number}}\ \underbrace{718}_{R\text{:3 digits}}$

21027845 | 718

8 digit number | R:3 digits

- $D = 21027845$
- $r = 3$
- $G = 401.$
- $X = 21027845\ \mathbf{999}$
- $y = 281$

- On the Receiver end, we find the remainder



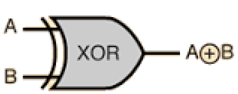
Cyclic Redundancy Check (CRC)

- D: data bits, viewed as a binary number.
- G: generator of $r+1$ bits, agreed by sender and receiver beforehand.
- R: the r bit CRC.

D: data to be sent	R: CRC bits
d bits	r bits

Cyclic Redundancy Check (CRC)

- Calculations are done modulo 2.
  - It does not have carries for addition or borrows for subtraction.
  - Both addition and subtraction are *identical* to XOR
    - $x + y = x - y = x \oplus y$
    - $0 + 1 = 0 - 1 = 0 \oplus 1 = 1$
    - $1011 \oplus 0101 = 1110$
    - $1001 \oplus 1101 = 0100$
    - $1011 - 0101 = 1110$
    - $1001 - 1101 = 0100$



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Cyclic Redundancy Check (CRC)

- For performing division, we append r 0's to D.
- Because of the properties of modulo 2 arithmetic, The remainder directly gives us R

```
           101011
      ____________
 1001 ) 101110000
        1001
        ----
          1010
          1001
          ----
            1100
            1001
            ----
             1010
             1001
             ----
              011
```

E.g.
$D = 101110,$
$r = 3$
$G = 1001$

G

D

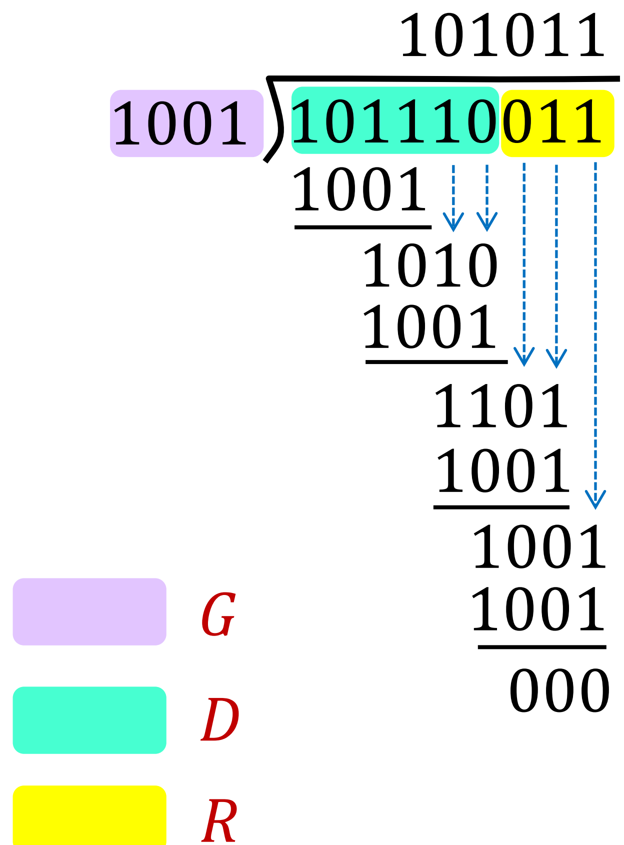
R

Cyclic Redundancy Check (CRC)

- Sender sends (D, R)

  101110011

- Receiver knows G, divides (D, R) by G.
  - If non-zero remainder: error is detected!



Cyclic Redundancy Check (CRC)

- Easy to implement on hardware
- Powerful error-detection coding that is widely used in practice (e.g., Ethernet, Wi-Fi)
  - Can detect *all odd number* of single bit errors
  - CRC of r bits can detect
    - all burst errors of less than $r + 1$ bits
    - all burst errors of greater than r bits with probability $1 - 0.5^r$
- CRC is also known as Polynomial code
  - A k-bit frame is regarded as the coefficient list for a polynomial with k terms, ranging from x^{k-1}

    E.g. 110001

    $\Rightarrow 1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0 = x^5 + x^4 + 1$

Cyclic Redundancy Check (CRC)

- E.g. $D = 101100$, $r = 3$ and $G = 1001$

$$
\begin{array}{r}
101001 \\
1001\,\overline{\big)\,101100000} \\
\underline{1001} \\
1000 \\
\underline{1001} \\
1000 \\
\underline{1001} \\
001
\end{array}
$$

G

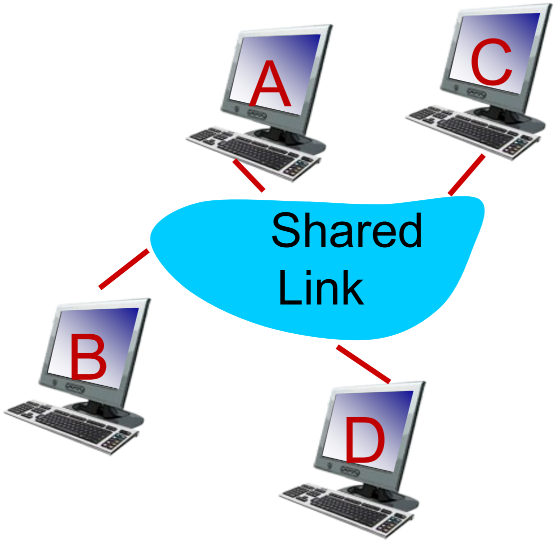
D

R

Summary

- Checksum (used in TCP/UDP/IP)
- Parity Checking
  - Single bit
  - 2-Dimensional
    - *Capable of error correction*
- Cyclic redundancy Check (CRC)
  - Commonly used in link layer
  - Efficient
  - effective

Motivation (revisited)



- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Link Access Control

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

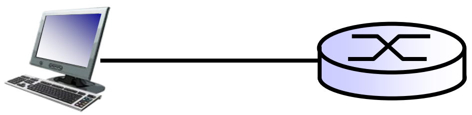
6.3 Multiple Access Links and Protocols

- 6.3.1 Channel Partitioning Protocols
- 6.3.2 Random Access Protocols
- 6.3.3 Taking-Turns Protocols

6.4 Switched Local Area Networks

Two Types of Network Links

- **Type 1: point-to-point link**
  - A sender and a receiver connected by a dedicated link



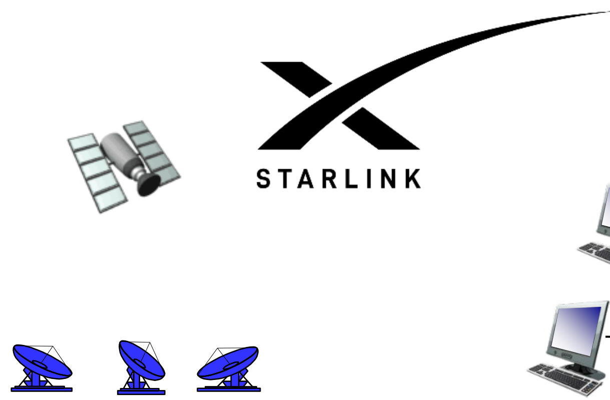
A host connects to router
through a dedicated link

  - Example protocols: Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP)
    - No need for multiple access control

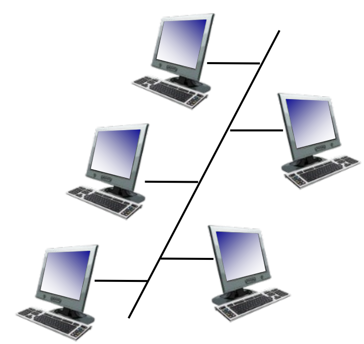
Two Types of Network Links

- **Type 2: broadcast link** (shared medium)
  - Multiple nodes connected to a shared broadcast channel.
  - When a node transmits a frame, the channel broadcasts the frame and every other node receives a copy.

802.11 Wi-Fi



Satellite



Ethernet with bus topology

Multiple Access Protocols

- In a broadcast channel, if two or more nodes transmit simultaneously
  - *collision* if node receives two or more signals at the same time.

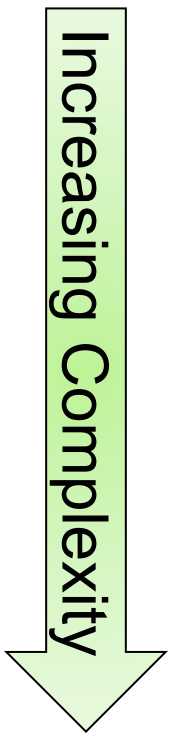
humans at a cocktail party (shared air)

Multiple Access Protocols: Motivation

- The central questions in a conversation carried in a group are *who, when* and how *long* one gets to talk.
- Desired Conversational Characteristics: *etiquettes*
  - Give everyone a chance to speak.
  - Don’t speak until you are spoken to.
  - Don’t monopolize the conversation.
  - Raise your hand if you have a question.
  - Don’t interrupt when someone is speaking.
  - Don’t fall asleep when someone is talking.

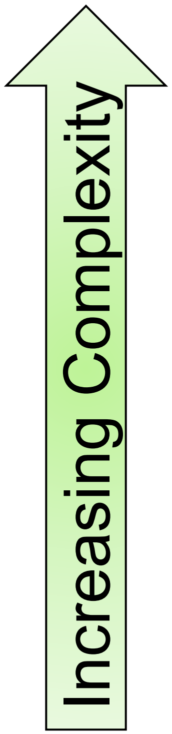
Multiple Access Protocols: Motivation

- Human Conversation Protocols can be categorized into three broad classes:



  - **Random Access**
    - No coordination, collisions are possible.
    - "recover" from collisions.
    - E.g. Most of our conversations
  - **"Taking turns"**
    - Each person take turns to talk.
    - E.g. Question answer sessions in seminars
  - **Channel partitioning**
    - divide channel into fixed, smaller "pieces" (e.g., time slots, subject).
    - allocate piece to a person for exclusive use.
    - E.g. U.S. Presidential debates (Mostly)

Multiple Access Protocols: Motivation



- Multiple access protocols can be categorized into three broad classes :
  - **Random Access**
    - channel is not divided, collisions are possible.
    - "recover" from collisions.
  - **"Taking turns"**
    - Each node take turns to transmit.
  - **Channel partitioning**
    - divide channel into fixed, smaller "pieces" (e.g., time slots, frequency).
    - allocate piece to node for exclusive use.

An ideal multiple access protocol

Given: Broadcast channel of rate R bps

Desired Properties:

1. Collision Free
2. Efficient: when only one node wants to transmit, it can send at rate R.
3. Fairness: when M nodes want to transmit, each can send at average rate R/M
4. fully decentralized:
   - no special node to coordinate transmissions

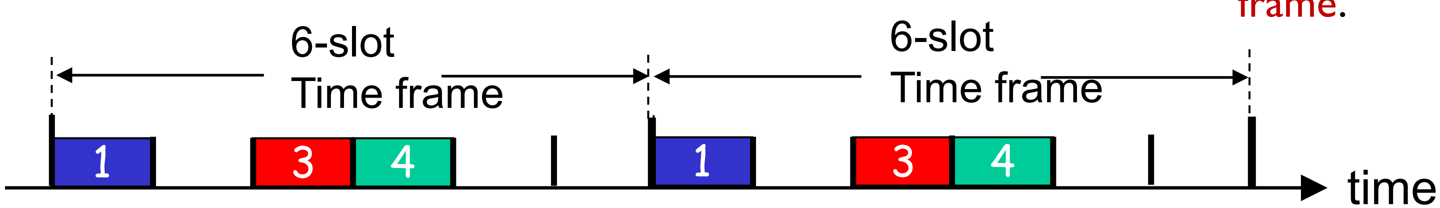
Mandatory Requirement: coordination about channel sharing must use channel itself!: no out-of-band channel signaling

Channel Partitioning Protocols: TDMA

- TDMA (time division multiple access)
  - Access to channel in "rounds".
    - Similar to US presidential debates
  - Each node gets fixed length time slots in each round.
    - Length of time slot = data frame transmission time
  - Example: 6 nodes sharing a link
    - Nodes 1, 3, 4 have data to send
    - slots 2, 5, 6 are idle.

Jargon Alert:

- **Frame**: Unfortunately, in TDMA, the collection of N time slots is called a Frame. We will disambiguate this by calling a frame as either data frame or time frame.



Channel Partitioning Protocols: TDMA

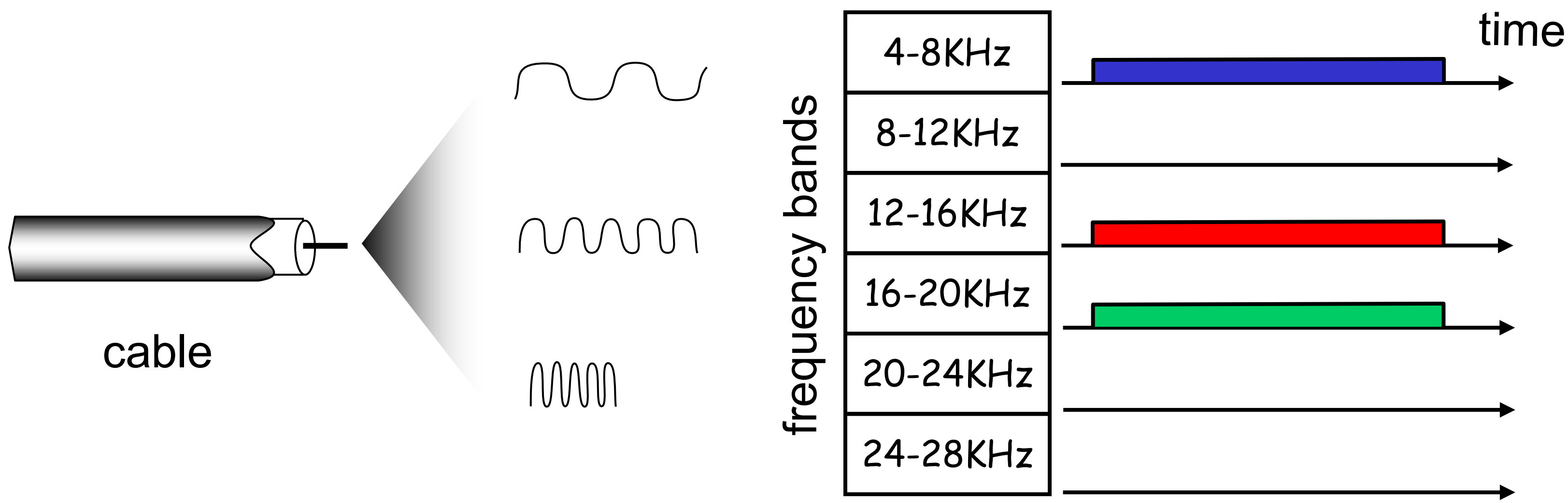
- Collision Free: Yes
- Efficiency
  - Inefficient
  - Unused slots go idle.
  - The maximum throughput for a node is R/N
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Channel Partitioning Protocols: FDMA

- FDMA (frequency division multiple access)
  - Channel spectrum is divided into frequency bands.
  - Each node is assigned a fixed frequency band.
  - Unused transmission time in frequency bands go idle.
  - Example: 6 nodes, 1, 3, 4 have frames, frequency bands 2, 5, 6 are idle.



Channel Partitioning Protocols: FDMA

- Collision Free: Yes
- Efficiency
  - Inefficient
  - Unused slots go idle.
  - The maximum throughput for a node is R/N
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

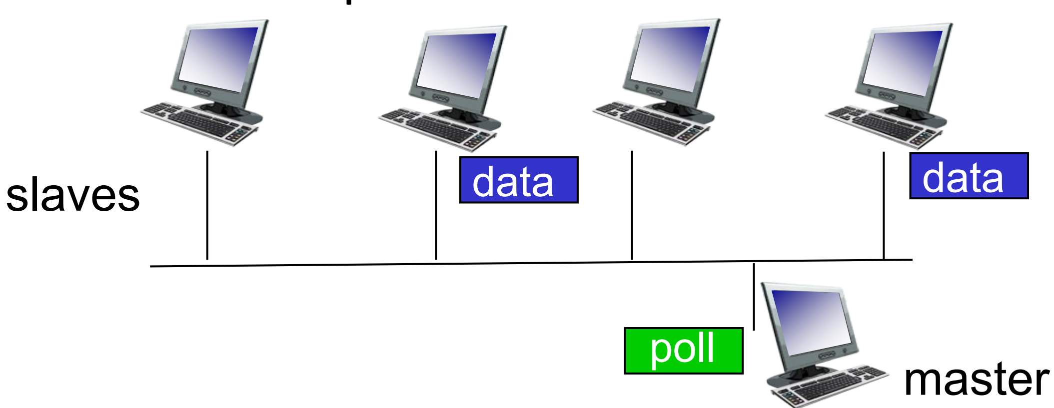
1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Multiple Access Protocols

- Multiple access protocols can be categorized into three broad classes:
  - Channel partitioning
    - divide channel into fixed, smaller "pieces" (e.g., time slots, frequency).
    - allocate piece to node for exclusive use.
  - **"Taking turns"**
    - nodes take turns to transmit.
  - Random Access
    - channel is not divided, collisions are possible.
    - "recover" from collisions.

"Taking Turns" Protocols: Polling

- The polling protocol requires one of the nodes to be designated as a *master* node.
- The master node *polls* each of the nodes in a *round-robin* fashion.
  - master informs node 1, it can transmit up to some maximum number of frames.
  - After node 1 transmits some frames, the master node tells node 2 it (node 2) can transmit up to the maximum number of frames.
  - The procedure continues in this manner



"Taking Turns" Protocols: Polling

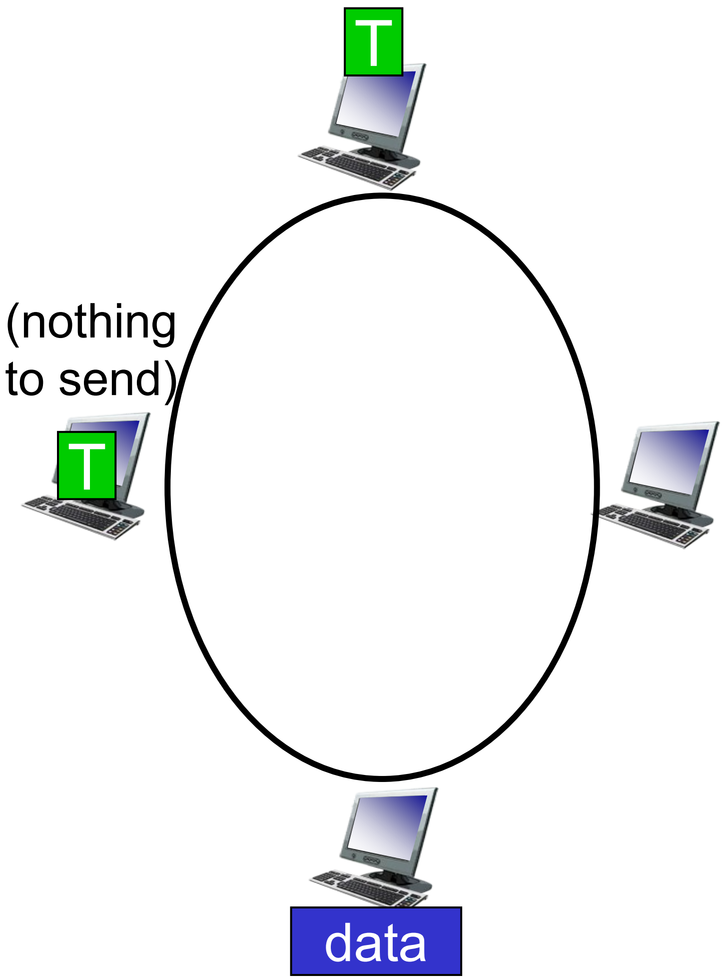
- Collision Free: Yes
- Efficiency
  - Higher efficiency.
  - Overhead of polling.
- Fairness: Perfectly Fair
- Decentralized:
  - No
  - Master node is a single point of Failure

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

"Taking Turns" Protocols: Token Passing

- Special frame, *token*, is passed from one node to next, sequentially.
- When a node receives a token
  - hold onto the token only if some frames to transmit
    - it sends up to a maximum number of frames and then forwards the token to the next node.
  - otherwise, forward the token to the next node.



"Taking Turns" Protocols: Token Passing

- Collision Free: Yes
- Efficiency
  - Higher efficiency.
  - Overhead of token passing
- Fairness: Perfectly Fair
- Decentralized: Yes
- *Downside*
  - Token loss can be disruptive
    - data frame loss
    - System bugs
  - Node failure can break the ring

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Multiple Access Protocols

- Multiple access protocols can be categorized into three broad classes:
  - Channel partitioning
    - divide channel into smaller "pieces" (e.g., time slots, frequency).
    - allocate piece to node for exclusive use.
  - "Taking turns"
    - nodes take turns to transmit.
  - **Random Access**
    - channel is not divided, collisions are possible.
    - "recover" from collisions.

Random Access Protocols

- When node has data to send
  - transmit at full channel data rate *R*.
  - no *a priori* coordination among nodes
- Two or more transmitting nodes ➜ "collision"

- Random access protocols specify:
  - how to *detect* collisions
  - how to *recover* from collisions

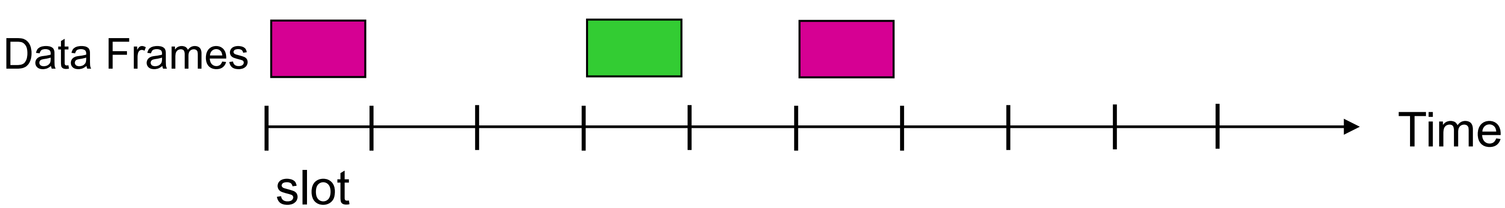
- We will explore various protocols
  - Slotted ALOHA, ALOHA
  - CSMA, CSMA/CD

Slotted ALOHA

- When node has data to send
  - transmit at full channel data rate *R*.
  - no *a priori* coordination among nodes

Design:

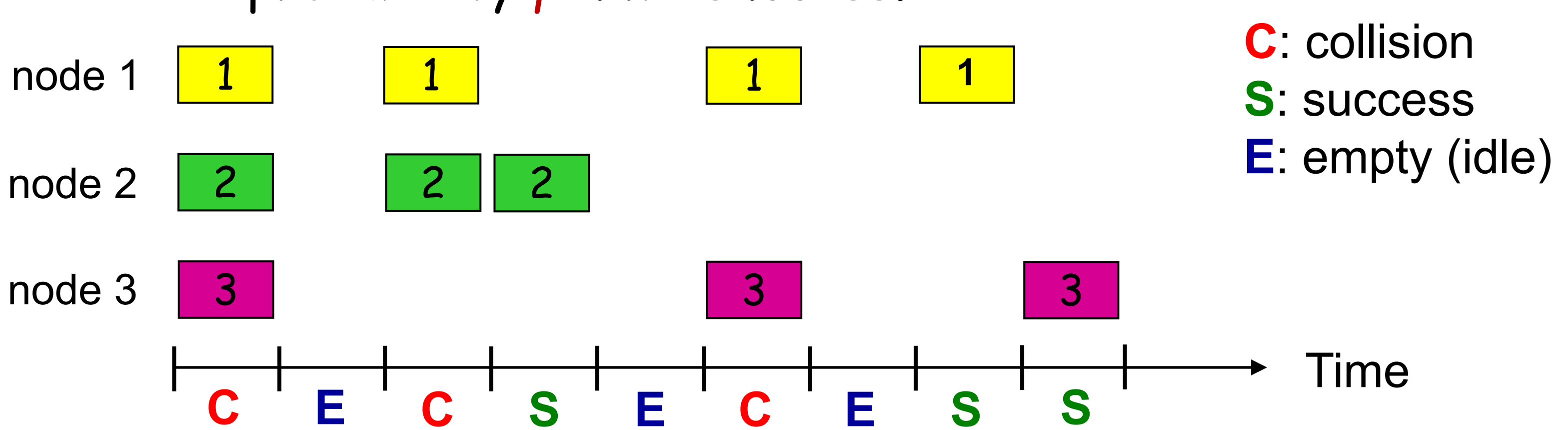
- All frames are of equal size, *L* bits.
- Time is divided into slots of equal length
  - length = time to transmit 1 frame = *L*/*R*
- Nodes start to transmit only at the beginning of a slot.
  - Time is synchronized at each node.



Slotted ALOHA

Operation:

- When the node has a fresh frame to send
  - wait until the beginning of the next slot and transmits the entire frame in the slot.
  - If *no collision*: data transmission is a success.
  - If *collision*: data transmission is a failure.
    - retransmit the frame in each subsequent slot with probability *p* until success.



Slotted ALOHA

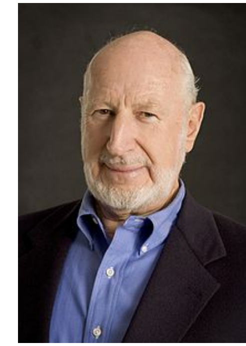
- Collision Free: No
- Efficiency
  - Yes, when only one node is active, it gets a throughput of R
  - No, when there are many active nodes the maximum efficiency is only *37%*
    - Slots are wasted due to both *collision* and because of being *empty*
    - 100 Mbps system will give only 37 Mbps
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

A Little Side Note

Figure 1. *ALOHA TCU 1971*



- Q: Why is it called ALOHA?
- A: The **ALOHAnet**, also known as the ALOHA System, or simply ALOHA, was a pioneering computer networking system developed – maybe you can guess it – at the University of Hawaii.
- Norman Abramson was the leader of the team.
- The idea was to use a radio network to connect Oahu and the other Hawaiian islands together. ALOHA made use of one, shared, inbound channel, and thus requiring a novel ***multiple access protocol***.

Pure (unslotted) ALOHA

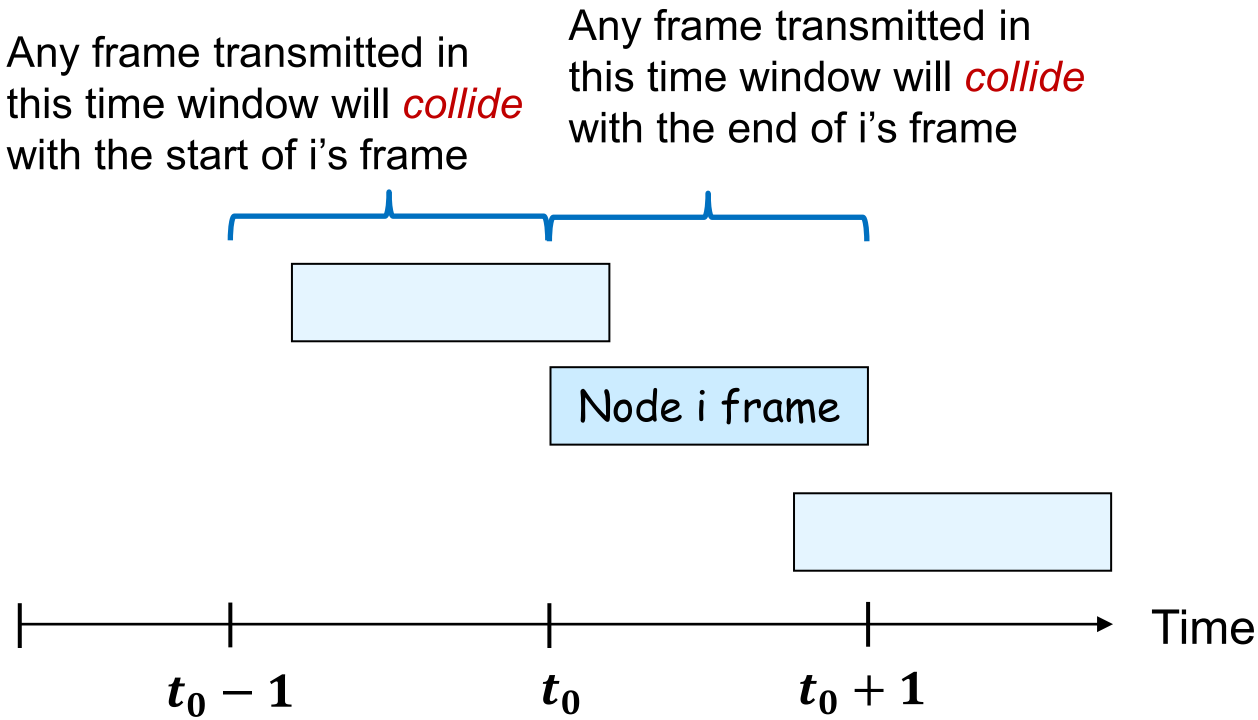
- Even simpler than Slotted ALOHA
  - *No* time slots
  - *No* Synchronization

Operation:

- When the node has a fresh frame to send
  - Transmits the entire frame *immediately*.
  - If *no collision*: data transmission is a success.
  - If *collision*: data transmission is a failure.
    - *Wait* for 1 frame transmission time
    - retransmit the frame with probability *p* until success.

Pure (unslotted) ALOHA

- Chance of collision increases:
  - frame sent at t_0 collides with other frames sent in $(t_0 - 1, t_0 + 1)$



Pure (unslotted) ALOHA

- Collision Free: No
- Efficiency
  - Yes, when only one node is active, it gets a throughput of R
  - No, when there are many active nodes the maximum efficiency is only *18%*
    - Slots are wasted due to both *collision* and because of being *empty*
    - 100 Mbps system will give only 18 Mbps
- Fairness: Perfectly Fair
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Carrier Sense Multiple Access

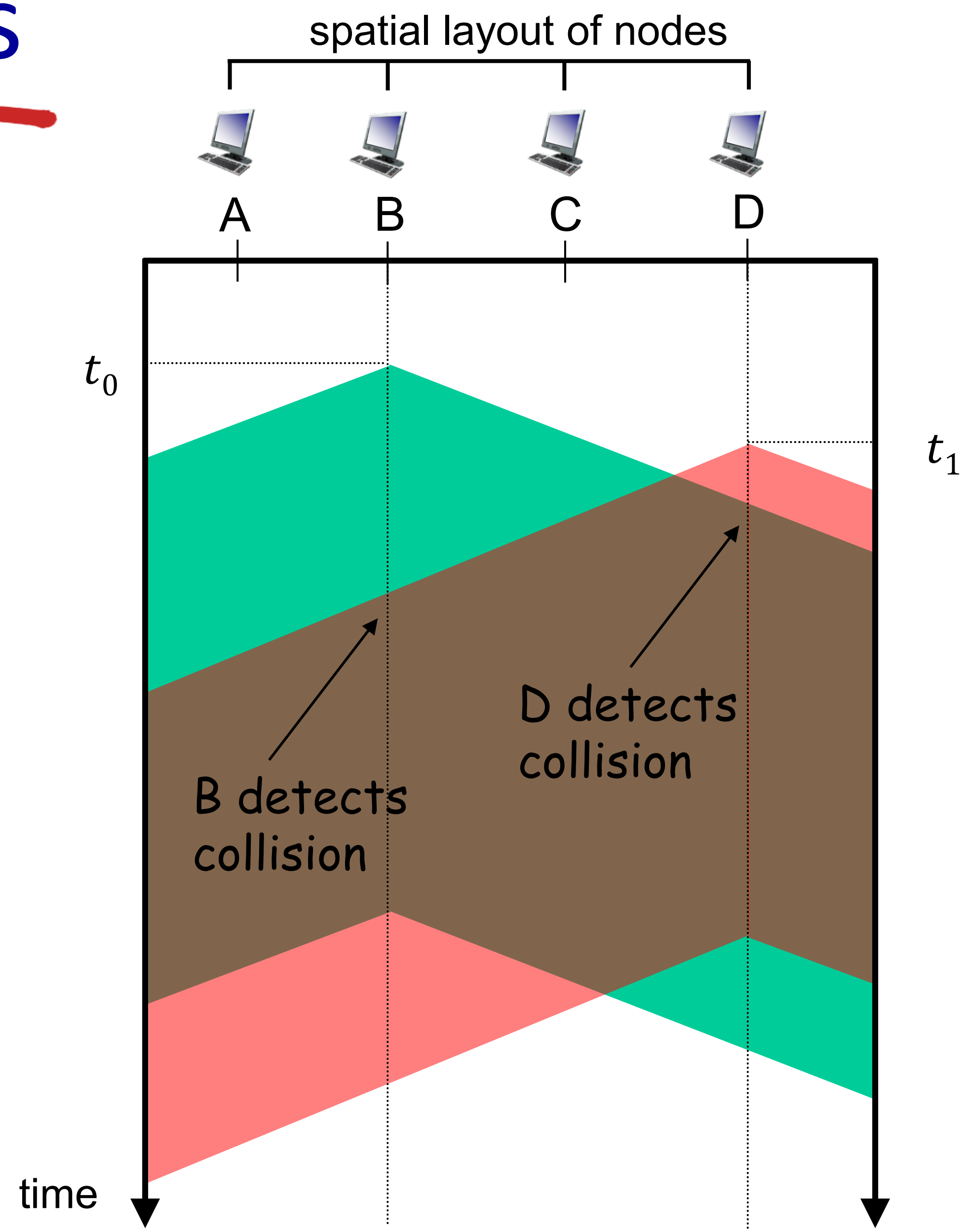
- One major *design flaw* in ALOHA
  - a node's decision to transmit is made *independently* of the activity of the other nodes attached to the broadcast channel.
  - a node *pays no attention* to whether another node happens to be transmitting when it begins to transmit
- Human analogy
  - Listen before you speak

CSMA: listen before transmit

- if channel *sensed idle*: transmit entire frame
- if channel *sensed busy*: defer transmission

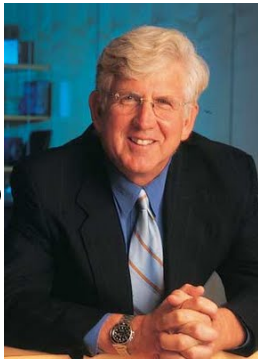
CSMA Collisions

- Collisions can still occur:
  - *propagation delay* means two nodes may not hear each other's transmission immediately.



CSMA/CD (Collision Detection)

- One major *design flaw* in ALOHA and CSMA
  - a node *does not stop* transmitting even when collision is detected
- Human analogy
  - If someone else begins talking at the same time, stop talking

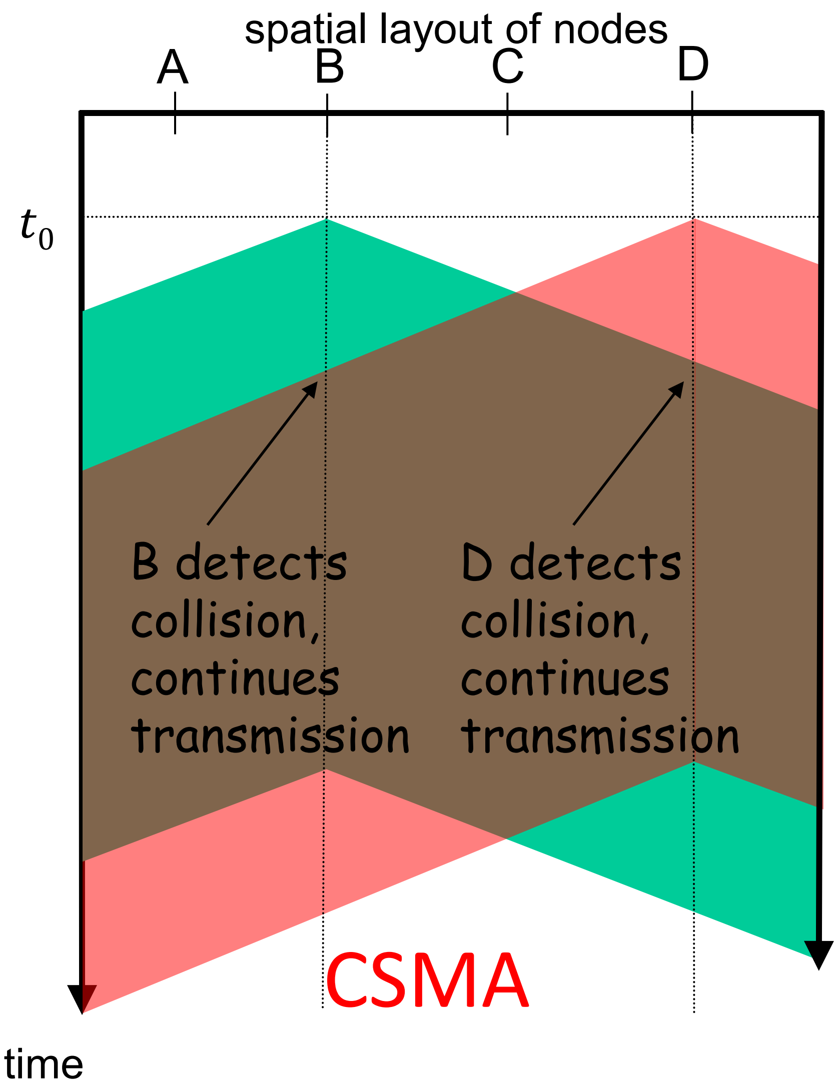


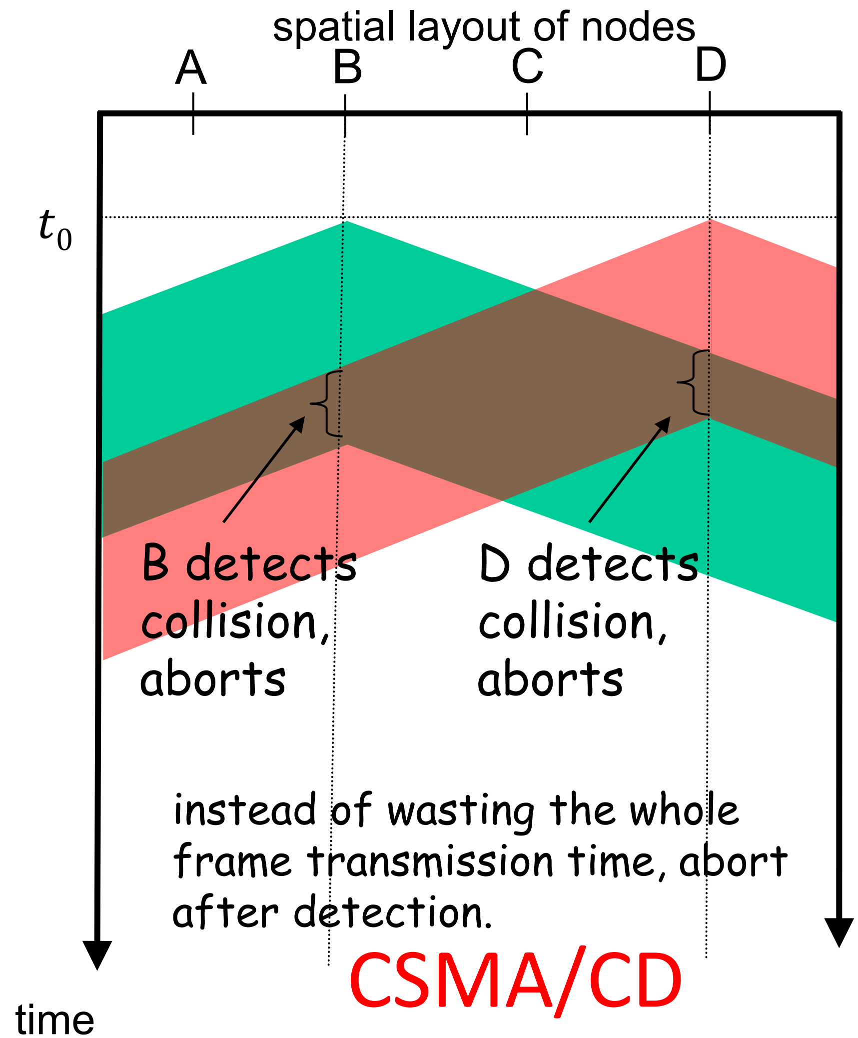
Bob Metcalfe

CSMA/CD:

- if channel *sensed idle*: transmit entire frame
- if channel *sensed busy*: defer transmission
- If *collision detected*: Abort transmission
  - Retransmit after a random delay

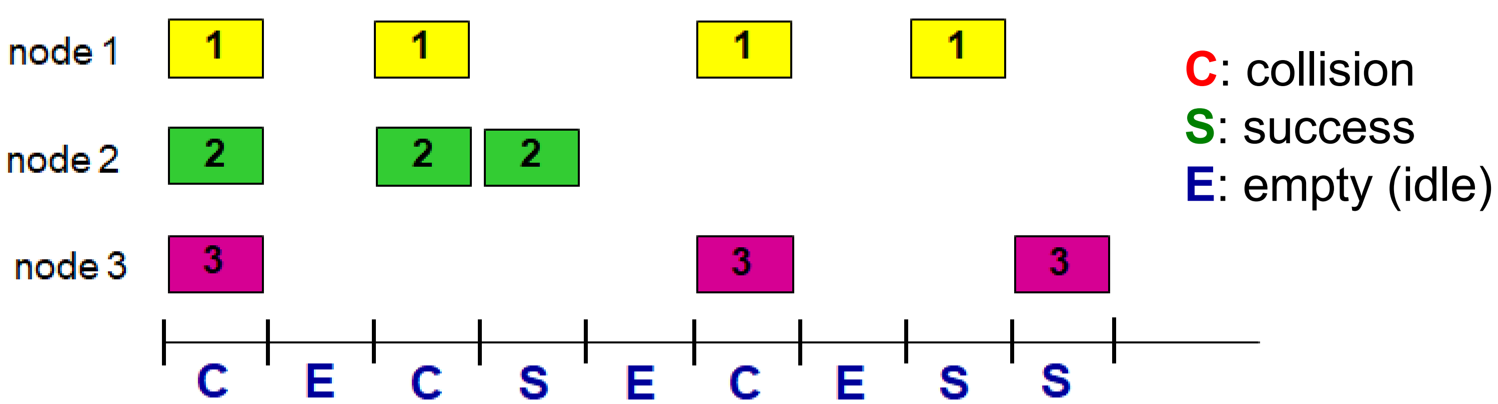
CSMA/CD (Collision Detection)





CSMA/CD Backoff Algorithm

- If *collision detected*: Abort transmission
  - Retransmit after a random delay
- *Motivation:* ALOHA
  - If *collision*: data transmission is a failure.
    - *Wait* for 1 frame transmission time
    - retransmit the frame with probability *p* until success.



CSMA/CD Backoff Algorithm

- If *collision detected*: Abort transmission
  - Retransmit after a random delay
- *Motivation:* ALOHA
  - If *collision*: data transmission is a failure.
    - *Wait* for 1 frame transmission time
    - retransmit the frame with probability *p* until success.
  - Major Drawback:
    - The probability of collision in all subsequent time slots remain the same
      - It can even increase if a new node starts transmitting
- *Goal*: adapt retransmission attempts to estimated current load
  - More collisions implies heavier load.
  - longer back-off interval with more collisions.

CSMA/CD Backoff Algorithm

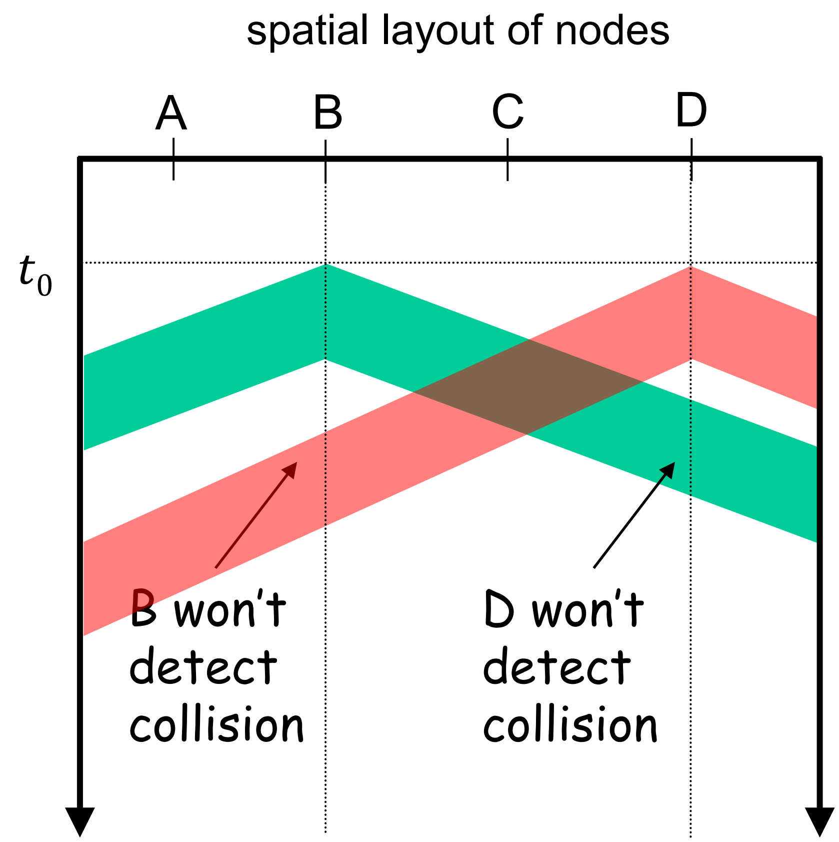
For Ethernet 1 time unit is set as 512 bit transmission times

Binary Exponential backoff:

- After 1st collision:
  - choose K at random from {0, 1}; $p = 1/2$
  - wait K time units before retransmission.
- After 2nd collision:
  - choose K from {0, 1,2, 2^2-1}. $p = 1/4$
  - wait K time units before retransmission.
- After m^{th} collision $p = 1/2^m$
  - choose K at random from {0, 1, …, 2^m-1}
- *Property*: retransmission attempts to estimates current load
  - More collisions implies heavier load.
  - longer back-off interval with more collisions.

Minimum Frame Size

- What if the frame size is too small?
  - Collision happens but may not be detected by sending nodes.
    - No retransmission!



- For example, Ethernet requires a minimum frame size of 64 bytes.

CSMA & CSMA/CD

- Collision Free: NO
- Efficiency: Yes
- Fairness: Yes
- Decentralized: Yes

Desired Properties:

1. Collision Free
2. Efficient
3. Fair
4. fully decentralized

Summary

- Channel partitioning
  - Divide channel by time, used in GSM
  - Divide channel by frequency, commonly used in radio, satellite systems
- Taking turns
  - polling from central site, used in Bluetooth
  - token passing, used in FDDI and token ring
- Random access
  - ALOHA wireless packet switched network.
  - CSMA/CD used in Ethernet