

CSE 461: Midterm Review

Autumn 2022

Administrivia

- **Midterm!**
 - Wednesday, November 16
- Assignment 3 is due Monday, November 14 (extended!)
- Project 2 is due Friday, November 18 (also extended!)
- No class tomorrow (Veterans Day)

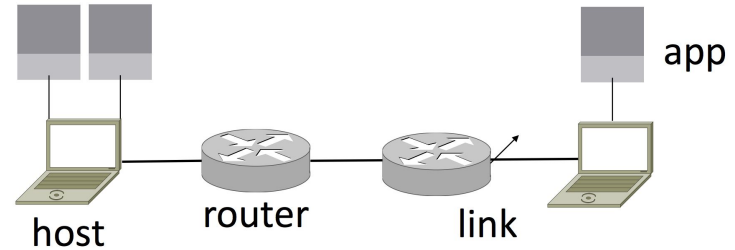


Network Components

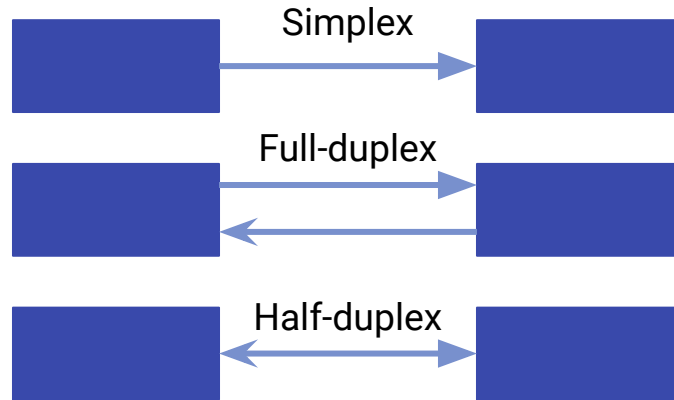
- Parts of a Network
 - Types of Links
 - Protocols and Layers
 - Encapsulation
-

Parts of a Network

- Parts of a Network



- Types of Links



Protocols and Layers

	Purpose	Protocols	Unit of Data
Application	Programs that use network service	HTTP, DNS	Message
Transport	Provides end-to-end data delivery	TCP, UDP	Segment
Network	Sends packets across multiple networks	IP	Packet
Link	Sends frames across a link	Ethernet, Cable	Frame
Physical	Transmit bits	—	Bit

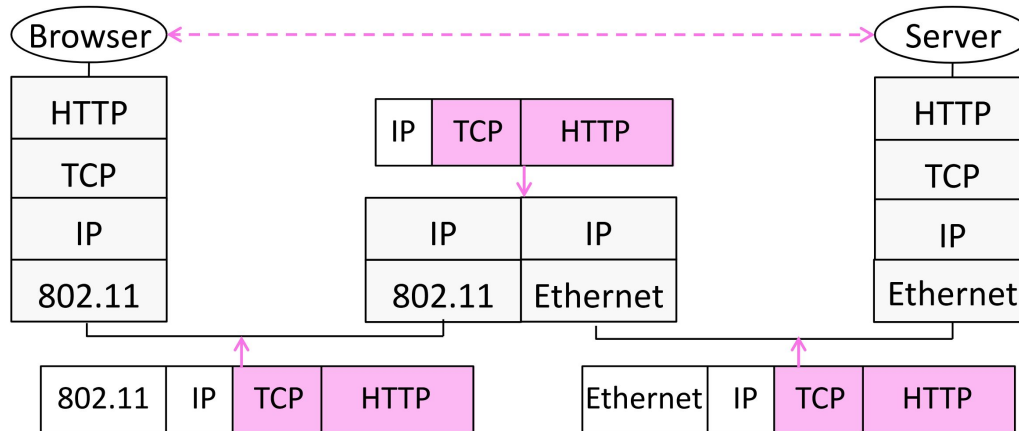
Protocols and Layers

ADVANTAGES

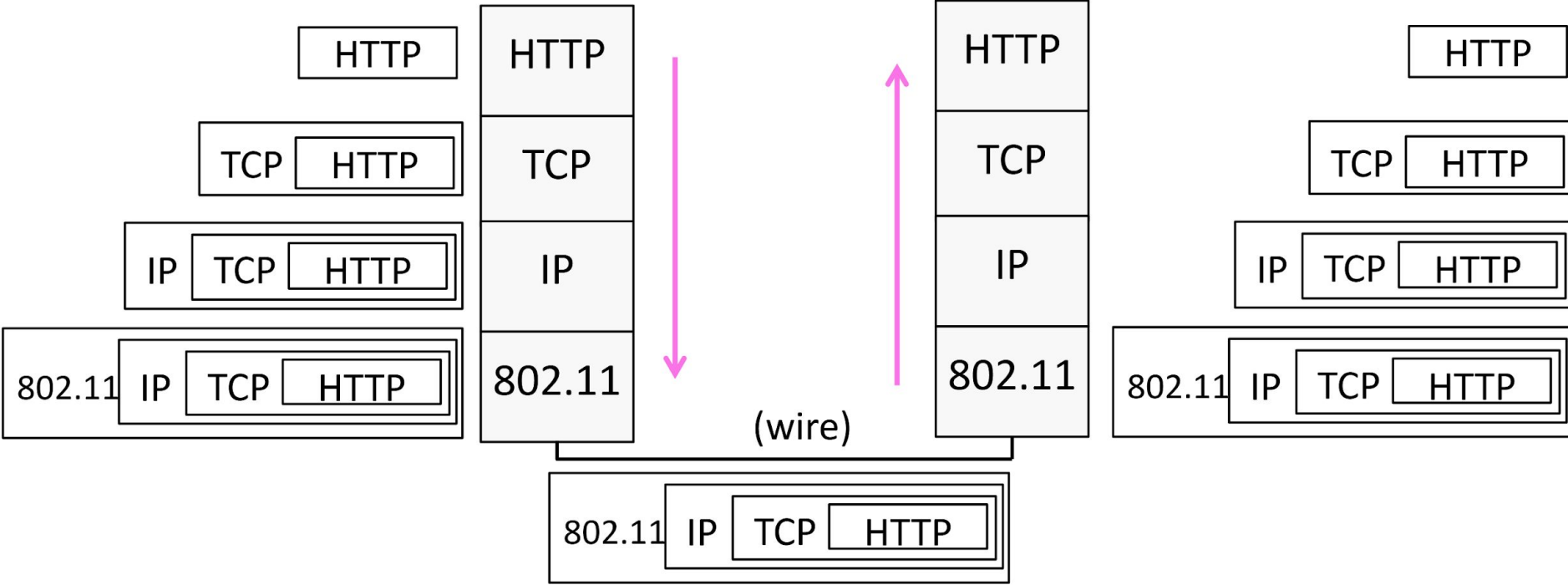
- Use information hiding to connect different systems
- Information reuse to build new protocols

DISADVANTAGES

- Adds overhead
- Hides information



Encapsulation

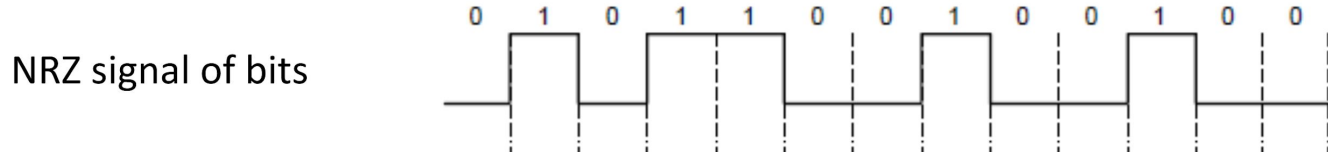


Physical Layer

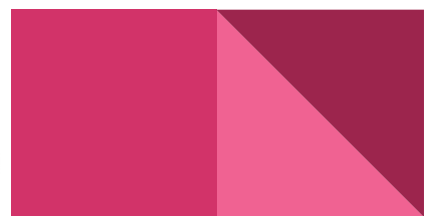
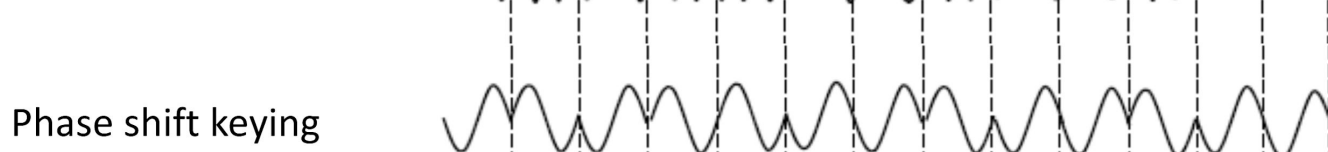
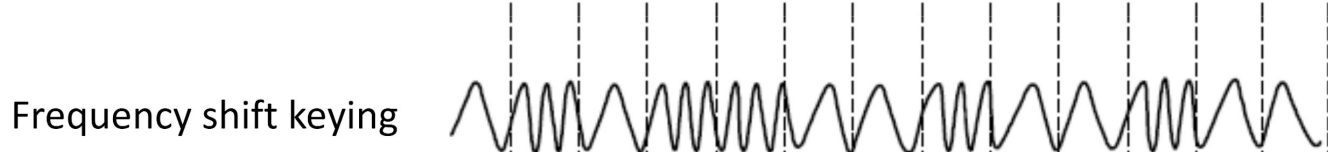
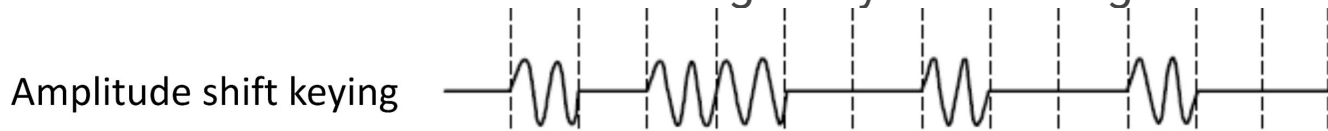
- Coding: Clock Recovery
- Modulation
- Latency
- Media and Theoretical Limits

Modulation

- Baseband modulation allows signal to be sent directly on wire

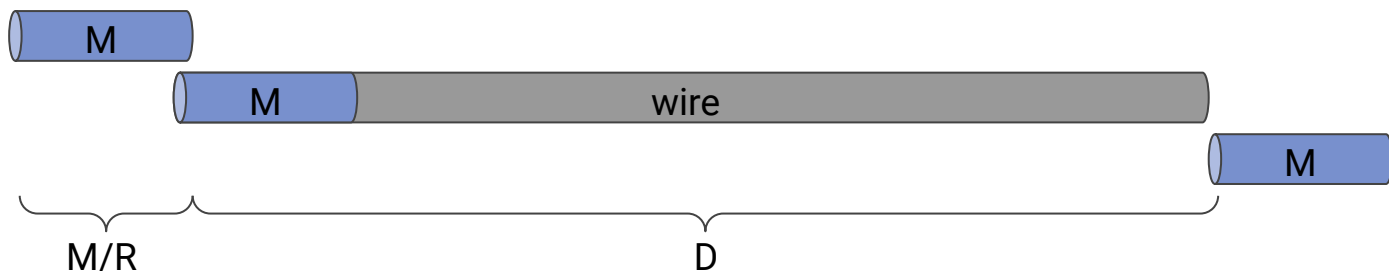


- Passband modulation carries a signal by modulating a carrier



Latency

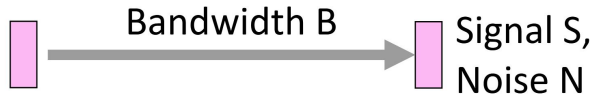
- Latency = Transmission Delay + Propagation Delay
- Transmission Delay = M (bits) / R (bits/sec) = M/R (sec)
- Propagation Delay = Length / Speed of Signals = Length / $\frac{2}{3}c$ = D (sec)



- Bandwidth-Delay Product = R (bits/sec) x D (sec) = BD (bits)
- RTT = round-trip time

Media and Theoretical Limits

- Media
 - Wire, Fiber
 - Wireless: radiates signal over a region
- Channel Limits: how rapidly can we send information over a link?
 - Bandwidth (**B**), Signal Power (**S**), Noise Power (**N**)
 - Shannon Capacity - maximum lossless info carrying rate



$$C = B \log_2(1 + S/N) \text{ bits/sec}$$

Link Layer

- Framing
 - Error detection and correction
 - Retransmissions
 - Multiple Access
 - Switching
-

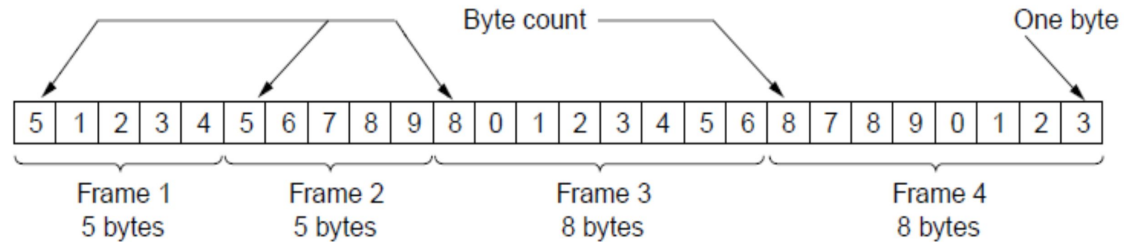
Framing



Framing Methods

- How do we know where a bit sequence (frame) begins and ends?
 - Byte count
 - Byte stuffing
 - Bit stuffing

- **Byte Count**



Framing Methods

- Byte Stuffing

- Replace ESC in data with ESC ESC, and replace FLAG in data with ESC FLAG



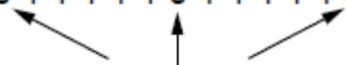
- Bit Stuffing

- Sequences of 1s as flag, and then add 0 after each flag within data

Data bits 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Transmitted bits with stuffing 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

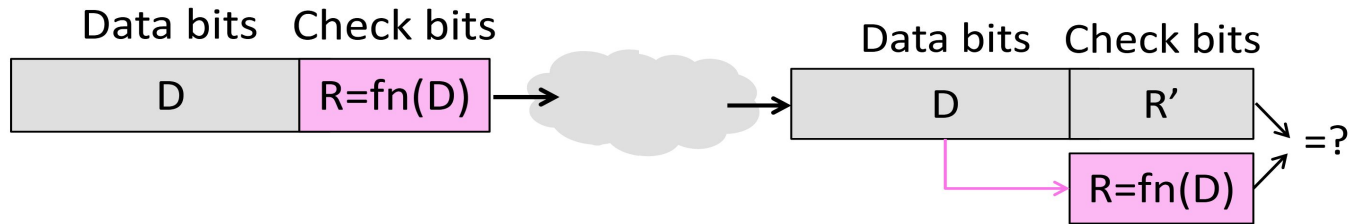




Error Detection and Correction

Error Detection and Correction

- Add check bits to the message bits to let some errors be detected
- Add more check bits to let some errors be corrected



Hamming Distance

- HD between two codes ($D1, D2$)
 - the number of bit flips needed to change $D1$ to $D2$
 - $D1 = 0110110101001$
 - $D2 = 0100000100001$
- HD of a coding
 - the minimum error distance between any pair of codewords that cannot be detected
- For a Hamming distance of $d + 1$, up to d errors will be **detected**
- For a Hamming distance of $2d + 1$, up to d errors can be **corrected**

Error Detection Methods

	Description	Hamming Distance
Parity Bit	Add 1 check bit that is sum/XOR of d data bits	2
Internet Checksum	1s complement sum of 16 bit word	2
Cyclic Redundancy Check (CRC)	For n data bits, generate n+k bits that are evenly divisible by C	4(CRC32)

HD of Internet Checksum

```
0001
f204
f4f5
f6f7
+ 220c
-----
2fffd
  ↓
 fffd
+   2
-----
 ffff
  ↓
0000
```



Error Correction - Hamming Code

Hamming Distance = 3

Suppose we want to send a message M of 4 bits: **0101**

We add $k=3$ check bits, because $(n = 2^k - k - 1 = 2^3 - 3 - 1 = 4)$

So, we will have a $n+k = 7$ bit code, with check bits in positions 1, 2, 4

Each check bit is an XOR of certain positions.



Error Correction - Hamming Code

	421	421	421
1 = 0b00 1	1 = 0b001	1 = 0b001	1 = 0b001
2 = 0b010	2 = 0b0 1 0	2 = 0b010	2 = 0b010
3 = 0b01 1	3 = 0b0 1 1	3 = 0b011	3 = 0b011
4 = 0b100	4 = 0b100	4 = 0b 1 00	4 = 0b100
5 = 0b10 1	5 = 0b101	5 = 0b101	5 = 0b101
6 = 0b110	6 = 0b1 1 0	6 = 0b110	6 = 0b110
7 = 0b11 1	7 = 0b1 1 1	7 = 0b111	7 = 0b111

0 1 0 0 1 0 1

1 2 3 4 5 6 7

$$p_1 = b_3 + b_5 + b_7 = 0 + 1 + 1 = 0$$

$$p_2 = b_3 + b_6 + b_7 = 0 + 0 + 1 = 1$$

$$p_4 = b_5 + b_6 + b_7 = 1 + 0 + 1 = 0$$

• Example, continued

→ 0 1 0 0 1 **1** 1
1 2 3 4 5 6 7

$$p_1 = 0 + 0 + 1 + 1 = 0, \quad p_2 = 1 + 0 + \mathbf{1} + 1 = \mathbf{1},$$

$$p_4 = 0 + 1 + \mathbf{1} + 1 = \mathbf{1}$$

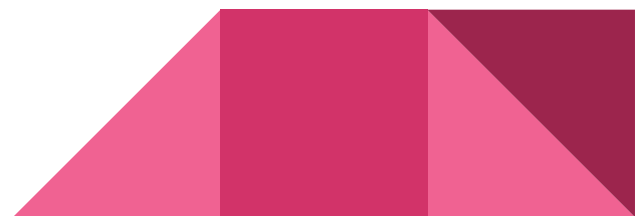
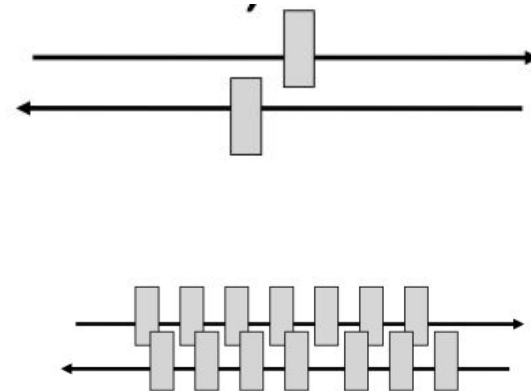
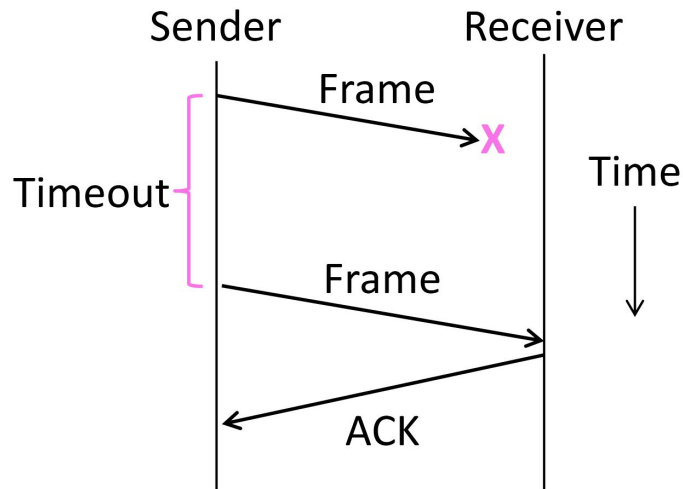
Syndrome = **1 1 0**, flip position 6

Data = 0 1 0 1 (correct after flip!)

Retransmissions

ARQ - Automatic Repeat Request

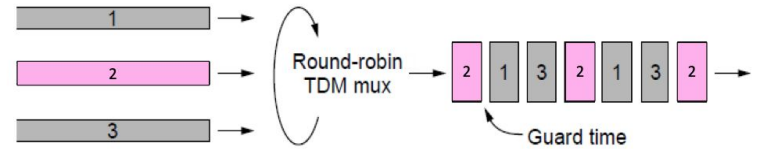
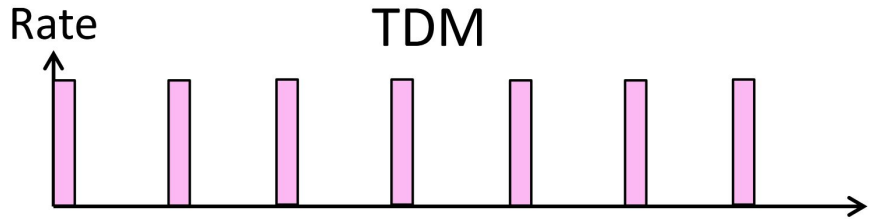
- ARQ
 - Wait-Resend
- Stop-and-wait
 - Single bit SEQ
- Sliding window



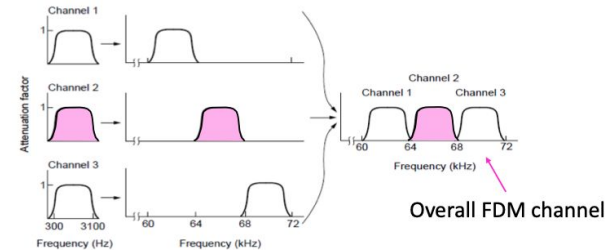
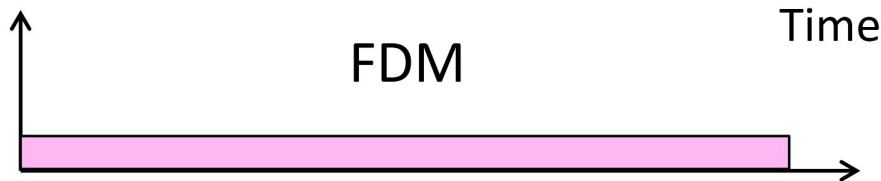
Multiple Access

Multiplexing

- Multiplexing is the network word for the sharing of a resource
- Time Division Multiplexing - high rate at some times



- FDM - low rate all the time



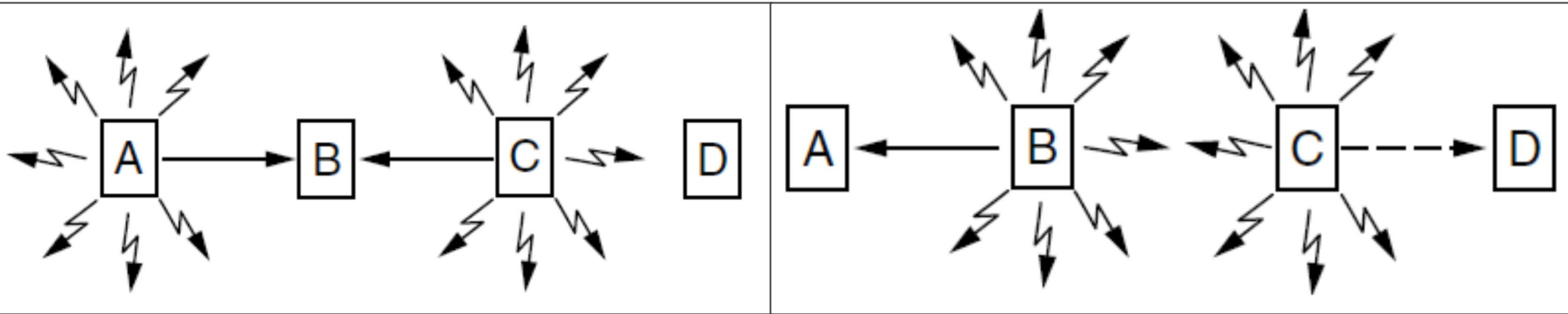
Multiple Access

- ALOHA: Node just sends when it has traffic; if collision happens, wait for a random amount of time and try again.
 - Huge amount of loss under high load
- CSMA (Carrier Sense Multiple Access): Listen before send.
 - Collision is still possible because of delay; good only when BD is small
- CSMA/CD (Carrier Sense Multiple Access with Collision Detection): CSMA + Aborting JAM for the rest of the frame time
 - Minimum frame length of $2D$ seconds
- CSMA “Persistence”: CSMA + $P(\text{send}) = 1 / N$
 - Reduce the chance of collision
- Binary Exponential Backoff (BEB): Doubles interval for each successive collision
 - Very efficient in practice

Issues with Wireless

Hidden Terminal Problem: nodes A and C are hidden terminals when sending to B

Exposed Terminal Problem: nodes B and C are exposed terminals when sending to A and D



MACA is a potential solution: Sender sends Request to Send(RTS) and receiver replies Clear To Send(CTS).

Switching

Switches

- Backward Learning
 - Learn the sender's port by looking at the packets
- Spanning Tree
 - Pick a root (Usually the switch with the lowest address)
 - Grow based on the shortest distance from the root
 - Ports not on the spanning tree are turned off

