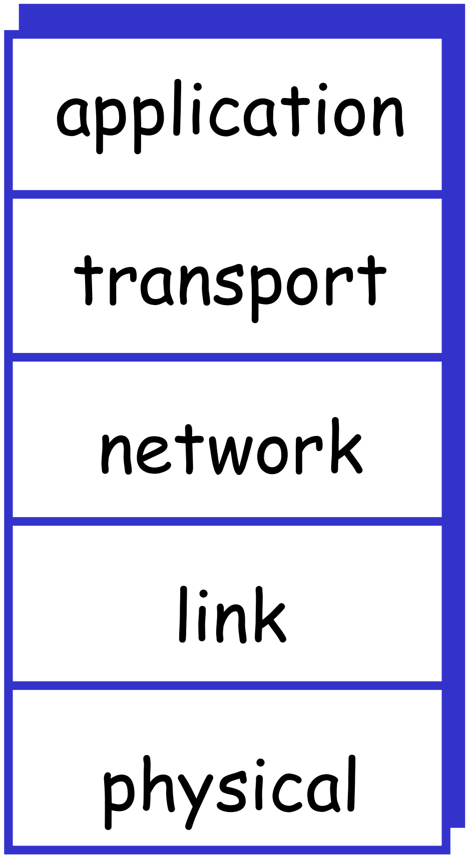
Mid-term Review

Spring 2026

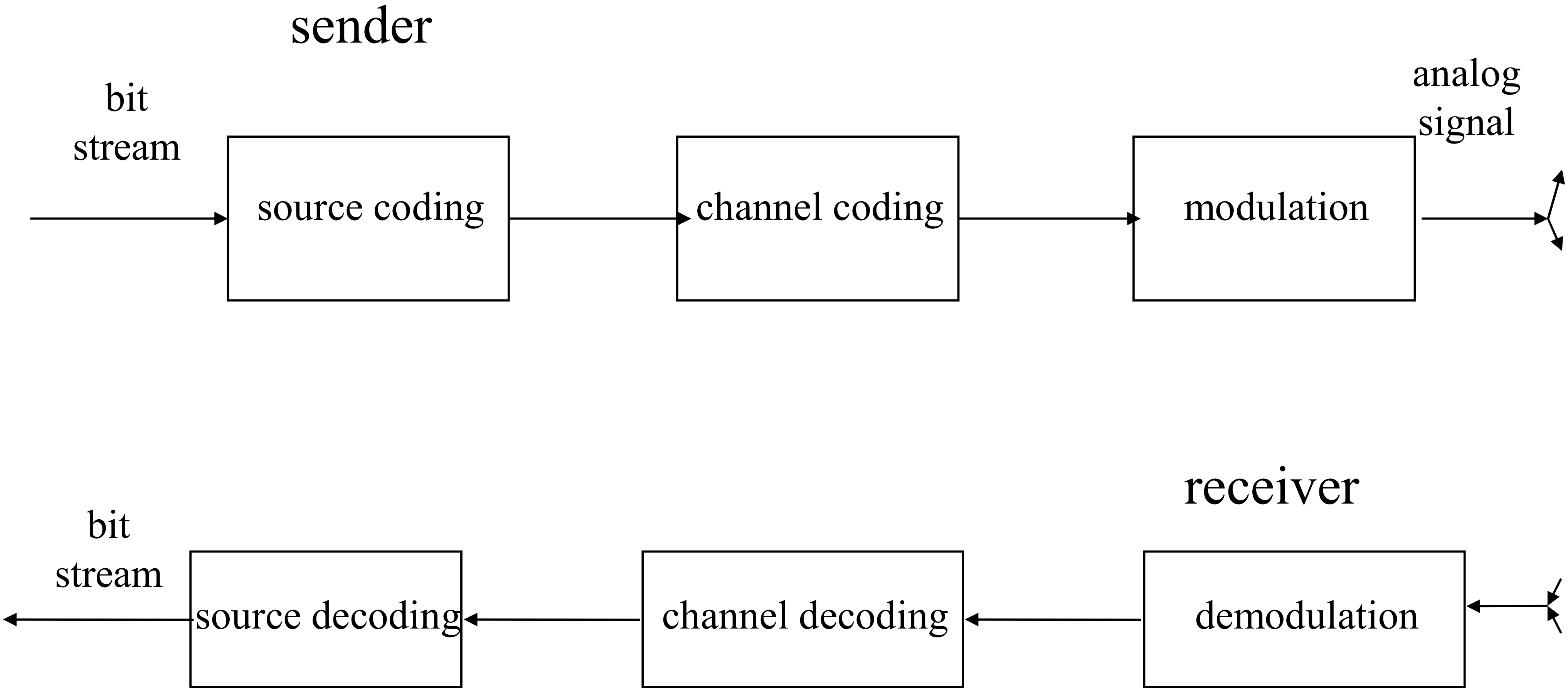
Internet Protocol Stack

- Application: supporting network applications
  - FTP, SMTP, HTTP
- Transport: host-host data transfer
  - TCP, UDP
- Network: routing of datagrams from source to destination
  - IP, routing protocols
- Link: data transfer between neighboring network elements
  - WiFi, Ethernet
- Physical: bits "on the wire"
  - Radios, coaxial cable, optical fibers



Physical Layer

Overview of Wireless Transmissions



Shannon Channel Capacity

- The maximum number of bits that can be transmitted per second by a physical channel is:

$$\boxed{W\log_2(1+\tfrac{S}{N})}$$

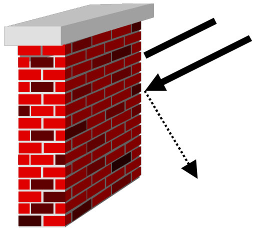
where W is the frequency range that the media allows to pass through, S/N is the signal noise ratio

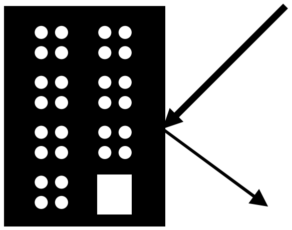
Signal, Noise, and Interference

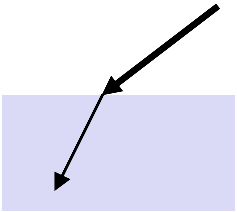
- Signal (S)
- Noise (N)
  - Includes thermal noise and background radiation
  - Often modeled as additive white Gaussian noise
- Interference (I)
  - Signals from other transmitting sources
- SINR = S/(N+I) (sometimes also denoted as SNR)

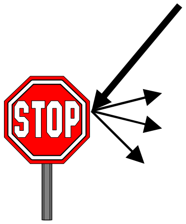
Signal Propagation

- Path loss
- Shadowing
- Reflection at large obstacles
- Refraction depending on the density of a medium
- Scattering at small obstacles
- Diffraction at edges
- Fading (depending on the frequency)


shadowing


reflection


refraction



scattering

diffraction

Path Loss (Cont.)

- Free space model

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

- Two-ray ground reflection model

$$P_r(d) = \frac{P_t G_t G_r h_t^{\,2} h_r^{\,2}}{d^4 L} \qquad d_c = (4\pi h_t h_r)/\lambda$$

- Log-normal shadowing

$$P(d)[dB] = \overline{P}(d)[dB] + X_\sigma$$

- Indoor model

$$P_r(d)[dBm] = P_t(d)[dBm] - 10n\log(\frac{d}{d_0}) - \begin{cases} nW * WAF & nW < C \\ C * WAF & nW \geq C \end{cases}$$

- P = 1 mW at d0=1m, what's Pr at d=2m?

Multiplexing

- Goal: multiple use of a shared medium

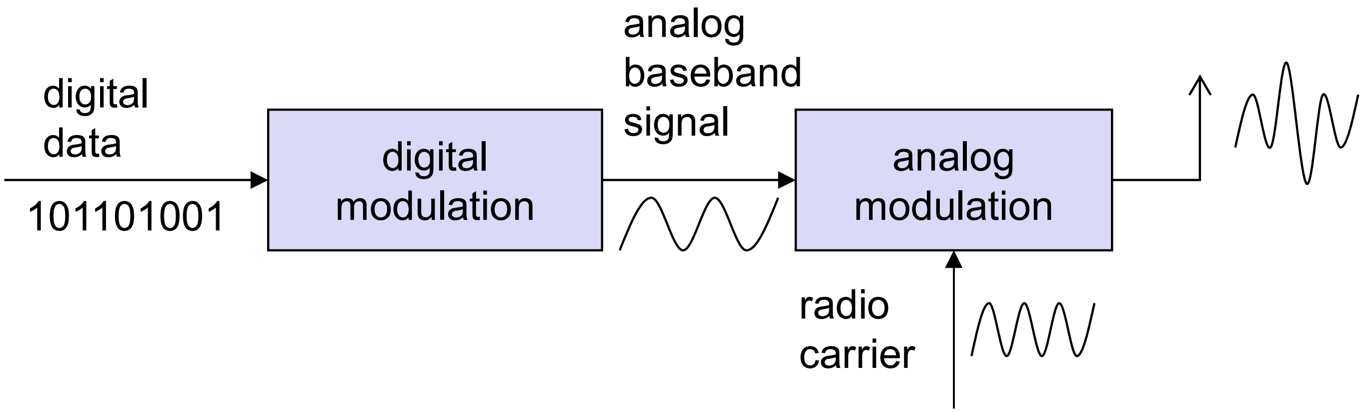
- Multiplexing in 4 dimensions

Multiplexing

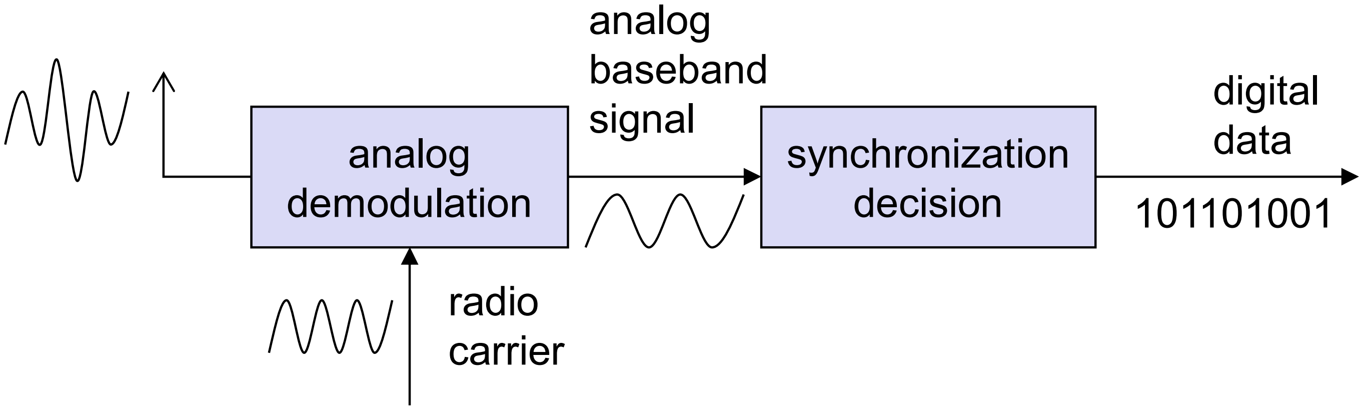
- Goal: multiple use of a shared medium

- Multiplexing in 4 dimensions
  - space (s_i)
  - time (t)
  - frequency (f)
  - code (c)

Modulation and Demodulation



radio transmitter



radio receiver

Modulation Schemes

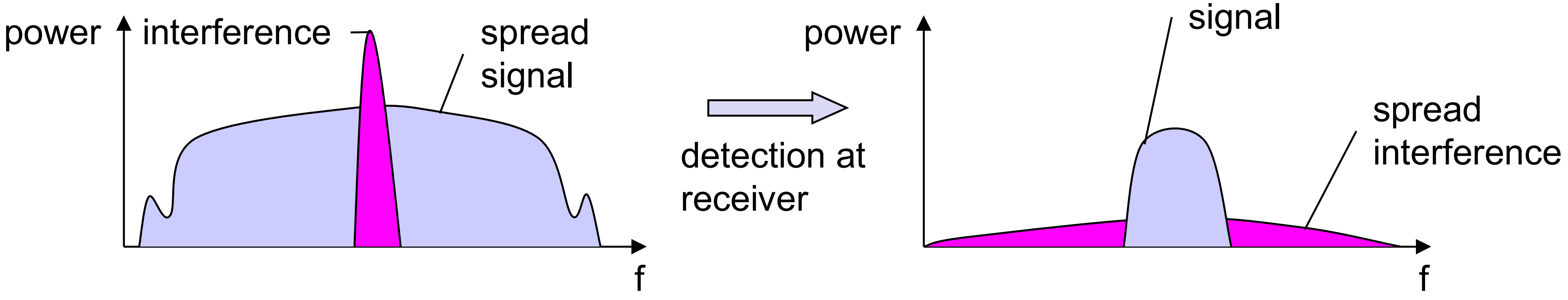
Modulation Schemes

- Amplitude Modulation (AM)
- Frequency Modulation (FM)
- Phase Modulation (PM)

Spread spectrum technology

Spread spectrum technology

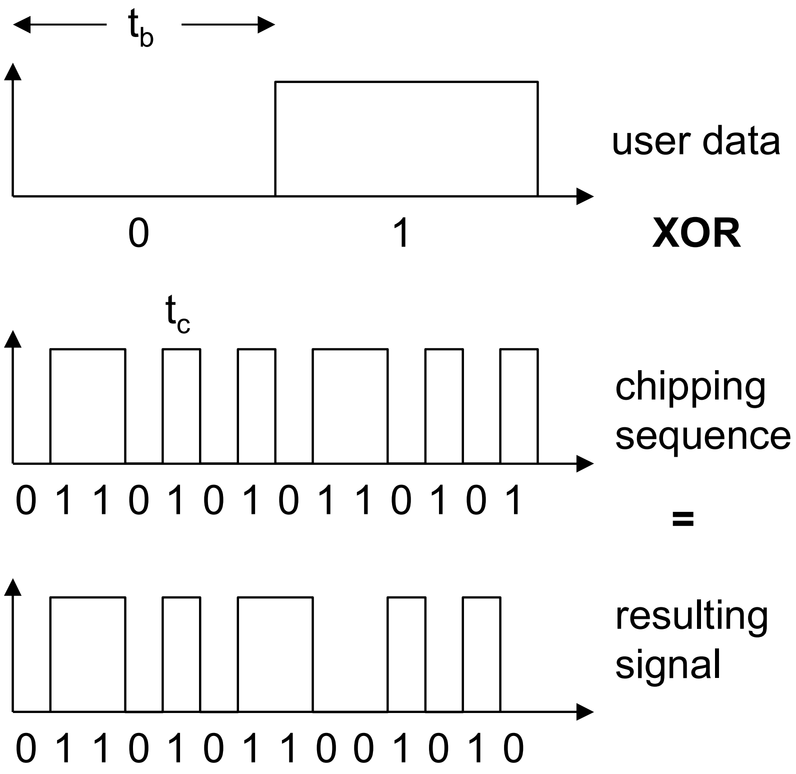
- Problem of radio transmission: frequency dependent fading can wipe out narrow band signals for duration of the interference
- Solution: spread the narrow band signal into a broad band signal using a special code



- Side effects:
  - coexistence of several signals without dynamic coordination
  - tap-proof
- Alternatives: Direct Sequence, Frequency Hopping

DSSS
(Direct Sequence Spread Spectrum)

- XOR of the signal with pseudo-random number (chipping sequence)
  - generate a signal with a wider range of frequency: spread spectrum



t_b: bit period
t_c: chip period

FHSS
(Frequency Hopping Spread Spectrum)

- Discrete changes of carrier frequency
  - sequence of frequency changes determined via pseudo random number sequence
- Two versions
  - Fast Hopping: several frequencies per user bit
  - Slow Hopping: several user bits per frequency
- Advantages
  - frequency selective fading and interference limited to short period
  - simple implementation
  - uses only small portion of spectrum at any time

Sample Question

- Path loss at reference distance $d_0 = 1$m is 40 dB. Path loss exponent $n = 3.5$. Calculate the path loss at 100 m (ignore shadowing).

Sample Question

- If transmit power increases by 6 dB, by what factor does coverage distance increase assuming path loss exponent $n = 4$?

Sample Question

- A link has:
  - Bandwidth = 1 MHz
  - SNR = 10 dB
- Compute the Shannon capacity.

Link Layer

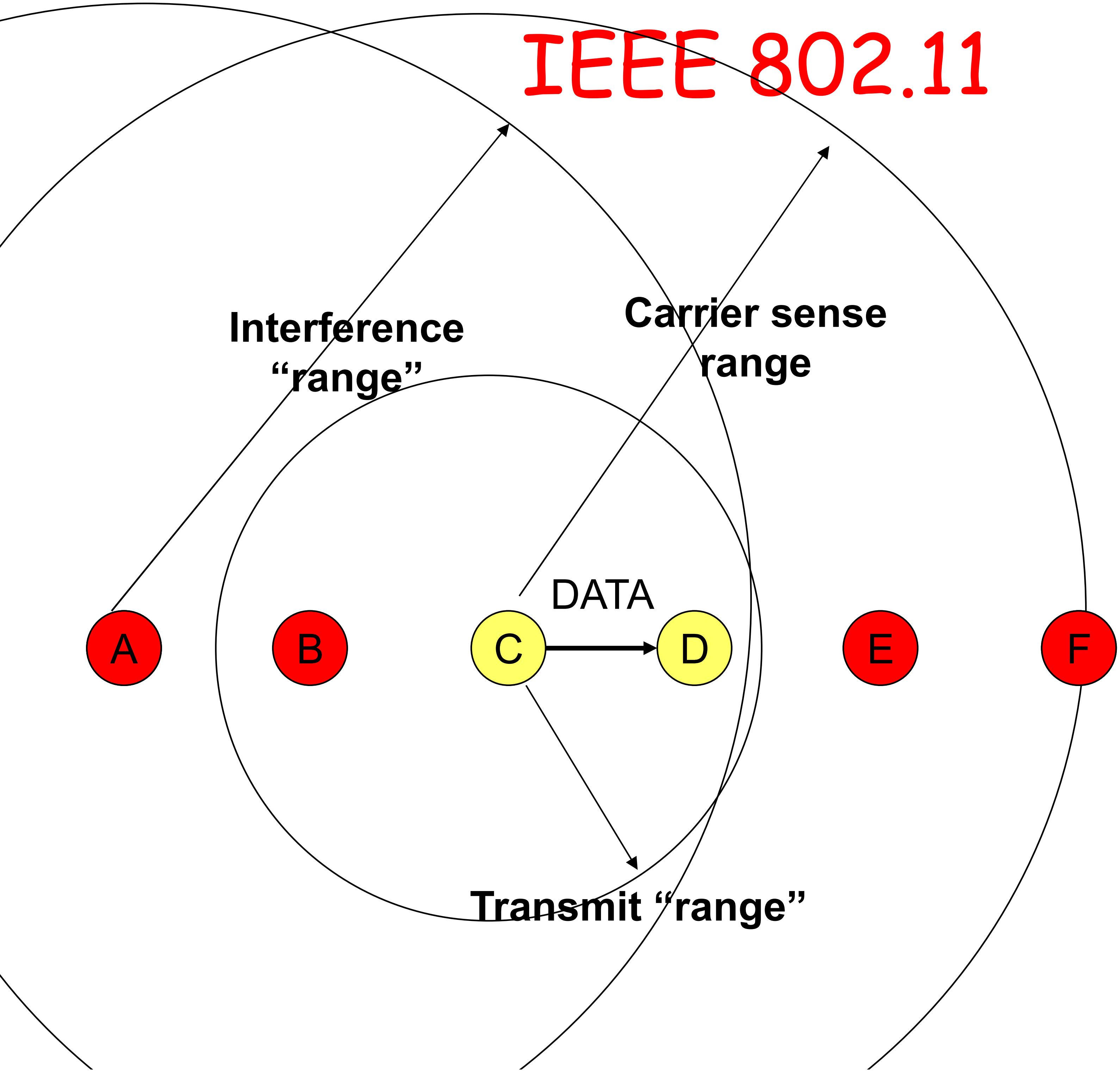
IEEE 802.11 DCF

- DCF is CSMA/CA protocol
  - Why?
- DCF suitable for multi-hop ad hoc networking
- Optionally uses RTS-CTS exchange to avoid hidden terminal problem
  - Any node overhearing a CTS cannot transmit for the duration of the transfer
- Uses ACK to provide reliability

CSMA/CA

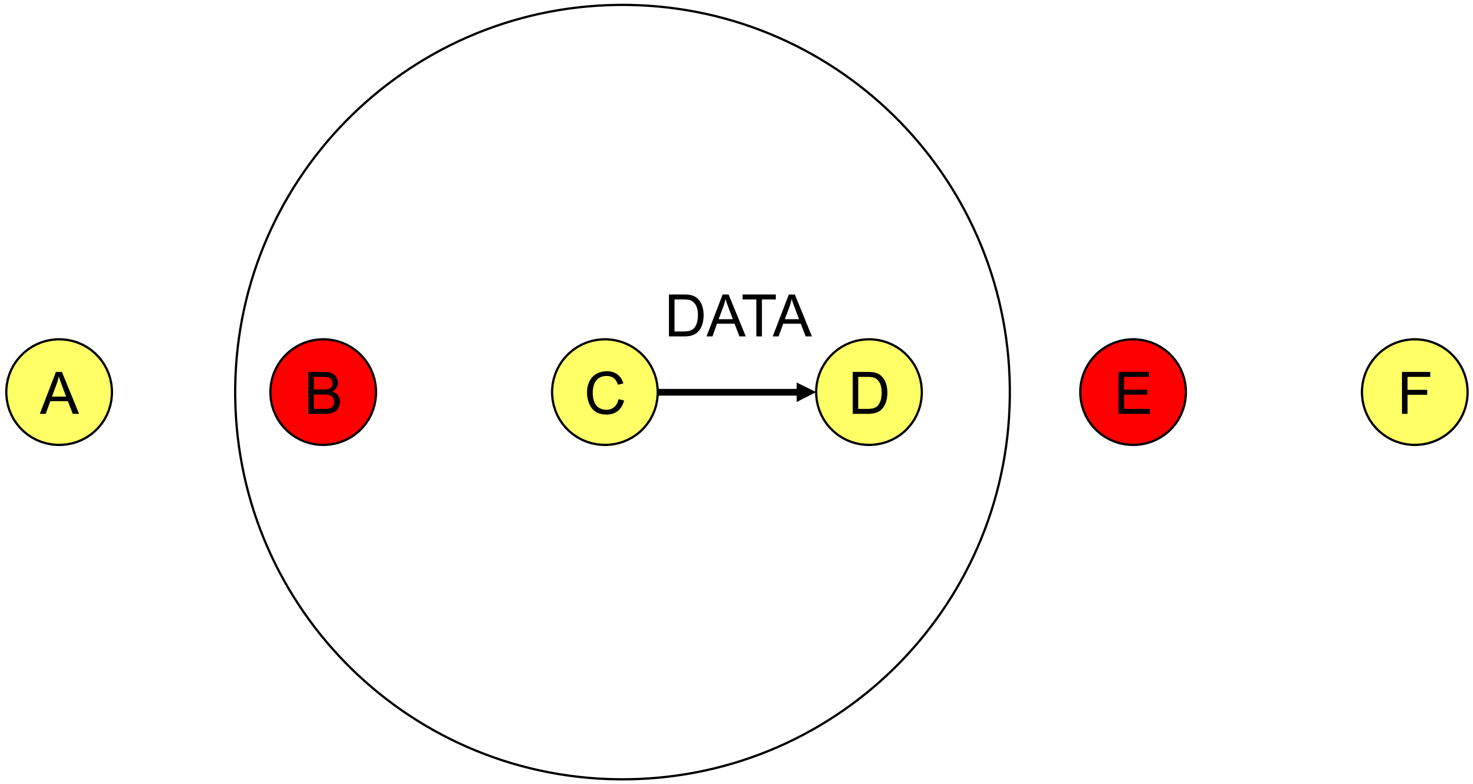
- Why not CSMA/CD?
  - Impossible to detect collision using half-duplex radios
  - Hidden terminal
- Solution: CSMA/CA
- Carrier sense
  - Nodes hearing RTS or CTS stay silent for the duration of the corresponding transmission.
  - Physical carrier sense
    - Carrier sense threshold
  - Virtual carrier sense using Network Allocation Vector (NAV)
    - NAV is updated based on overheard RTS/CTS/DATA/ACK packets
- Collision avoidance
  - Once channel becomes idle, the node waits for a randomly chosen duration before attempting to transmit.

IEEE 802.11

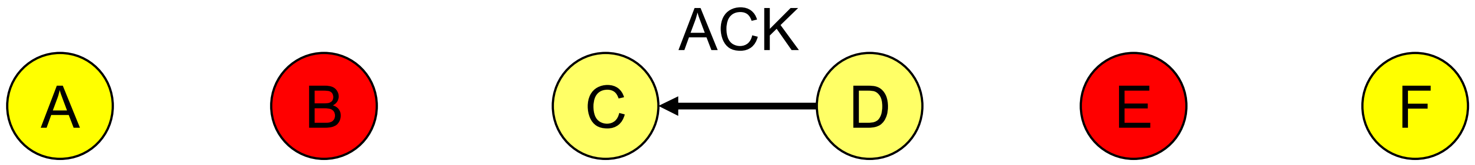

Interference
“range”
Carrier sense
range
DATA
A
B
C
D
E
F
Transmit “range”

IEEE 802.11: CSMA/CA

- DATA packet, followed by ACK.

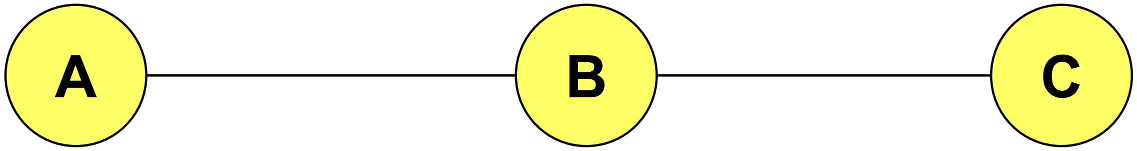


IEEE 802.11



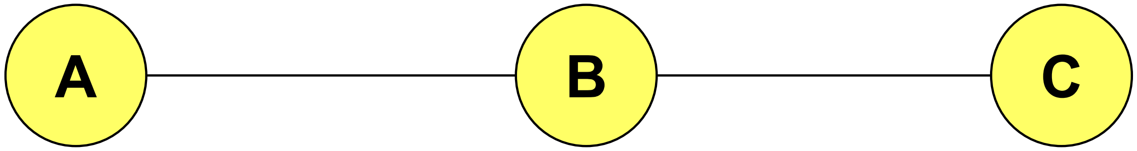
Hidden Terminal Problem

Hidden Terminal Problem



- B can communicate with both A and C
- A and C cannot hear each other
- Problem
  - When A transmits to B, C cannot detect the transmission using the carrier sense mechanism
  - If C transmits, collision will occur at node B
- Solution
  - Hidden sender C needs to defer
- How does 802.11 solve hidden terminal problem?

Solution for Hidden Terminal Problem: MACA



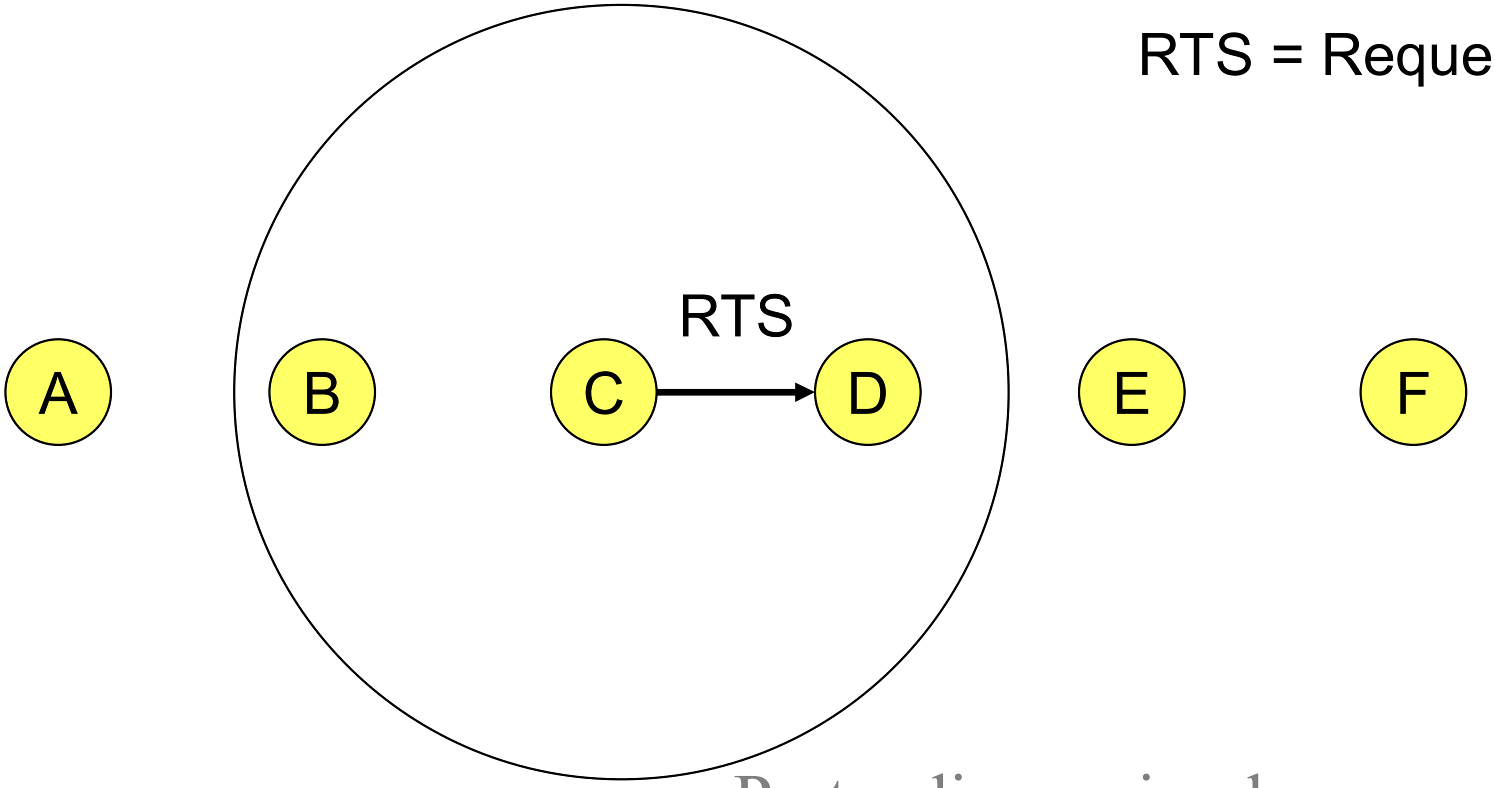
- When A wants to send a packet to B, A first sends a Request-to-Send (RTS) to B

- On receiving RTS, B responds by sending Clear-to-Send (CTS), provided that A is able to receive the packet

- When C overhears a CTS, it keeps quiet for the duration of the transfer
    - Transfer duration is included in both RTS and CTS

IEEE 802.11

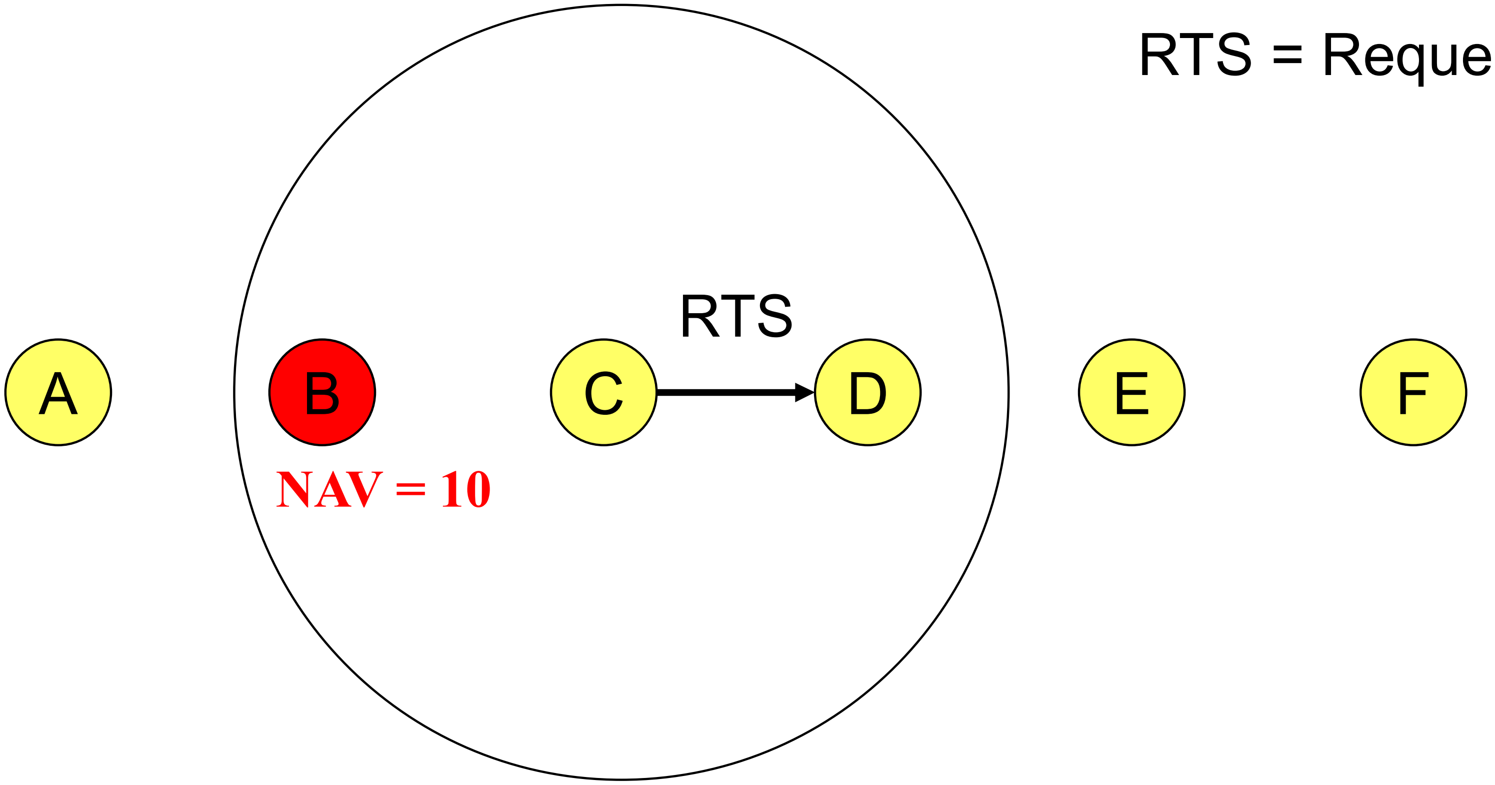
RTS = Request-to-Send



Pretending a circular range

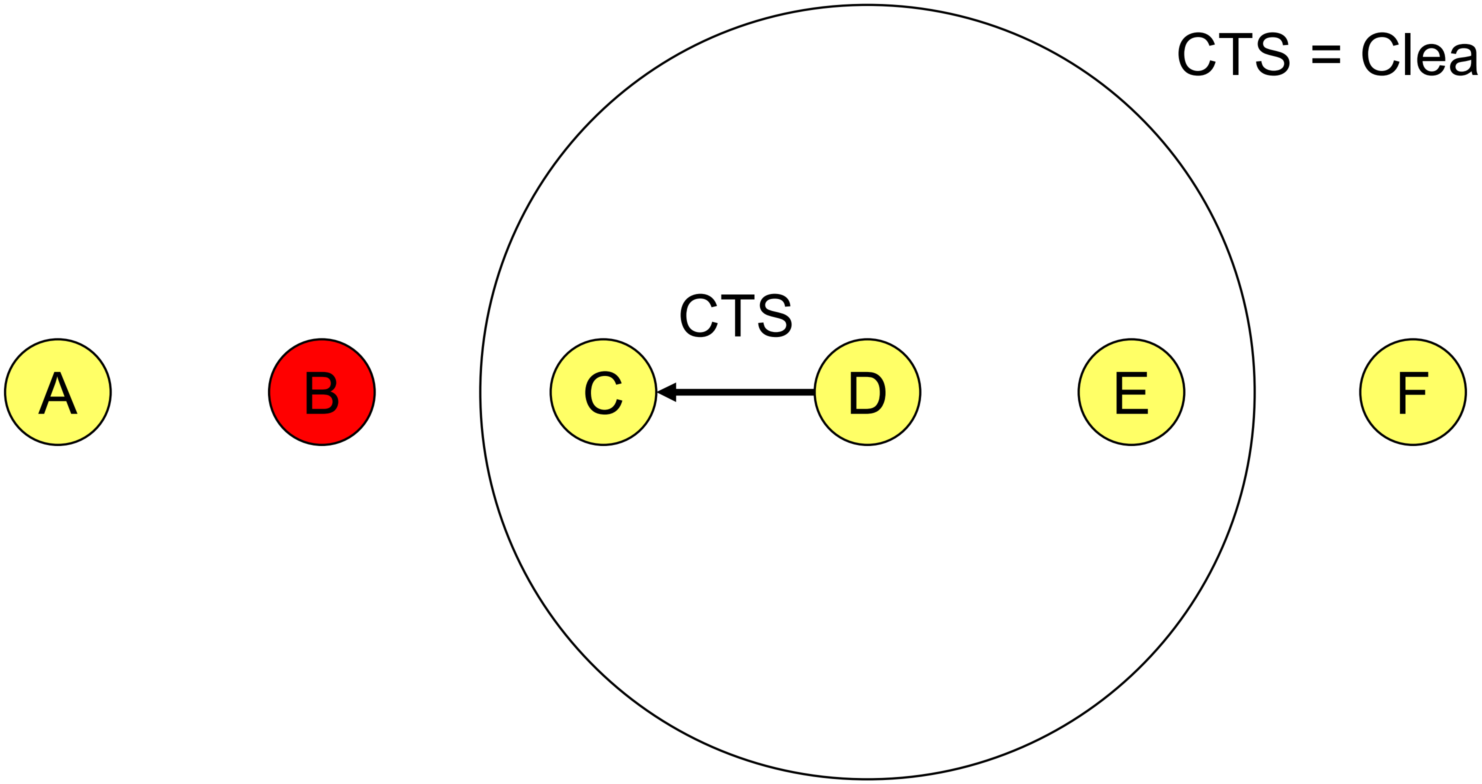
IEEE 802.11

RTS = Request-to-Send



NAV = remaining duration to keep quiet

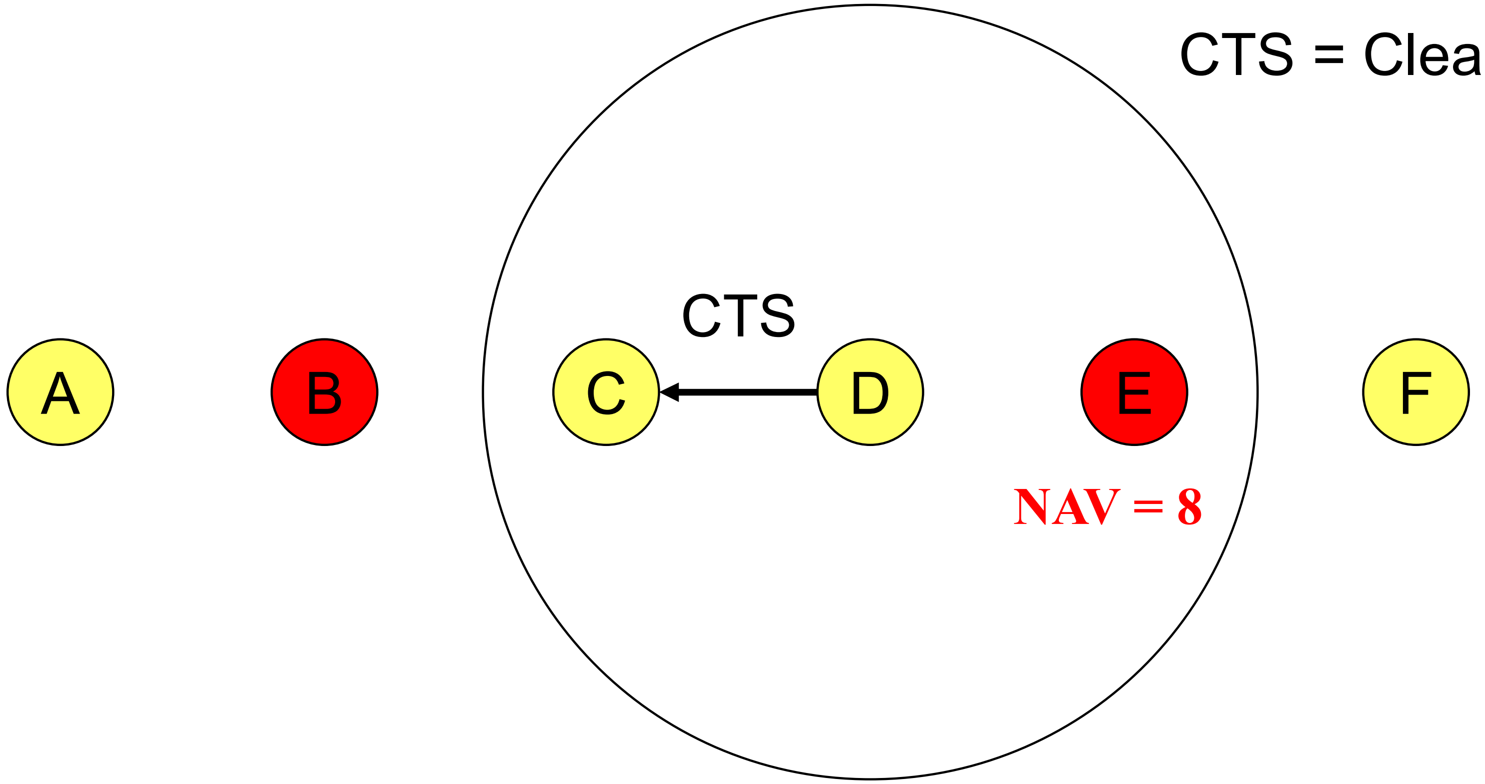
IEEE 802.11



CTS = Clear-to-Send

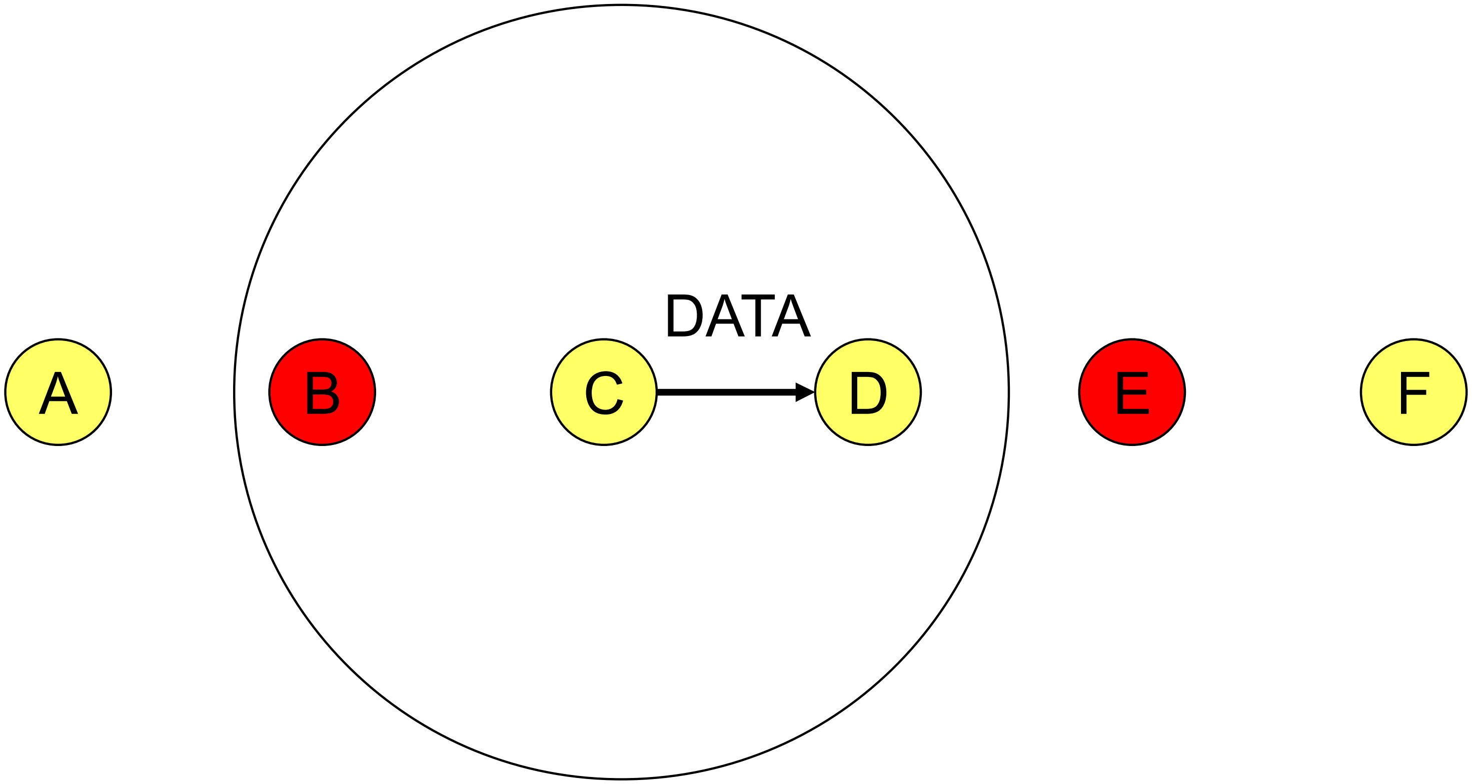
IEEE 802.11

CTS = Clear-to-Send

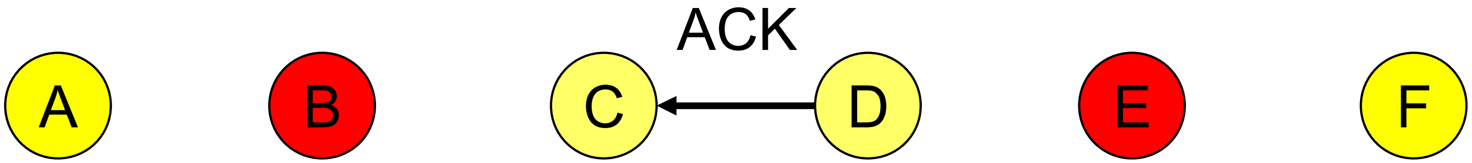


IEEE 802.11

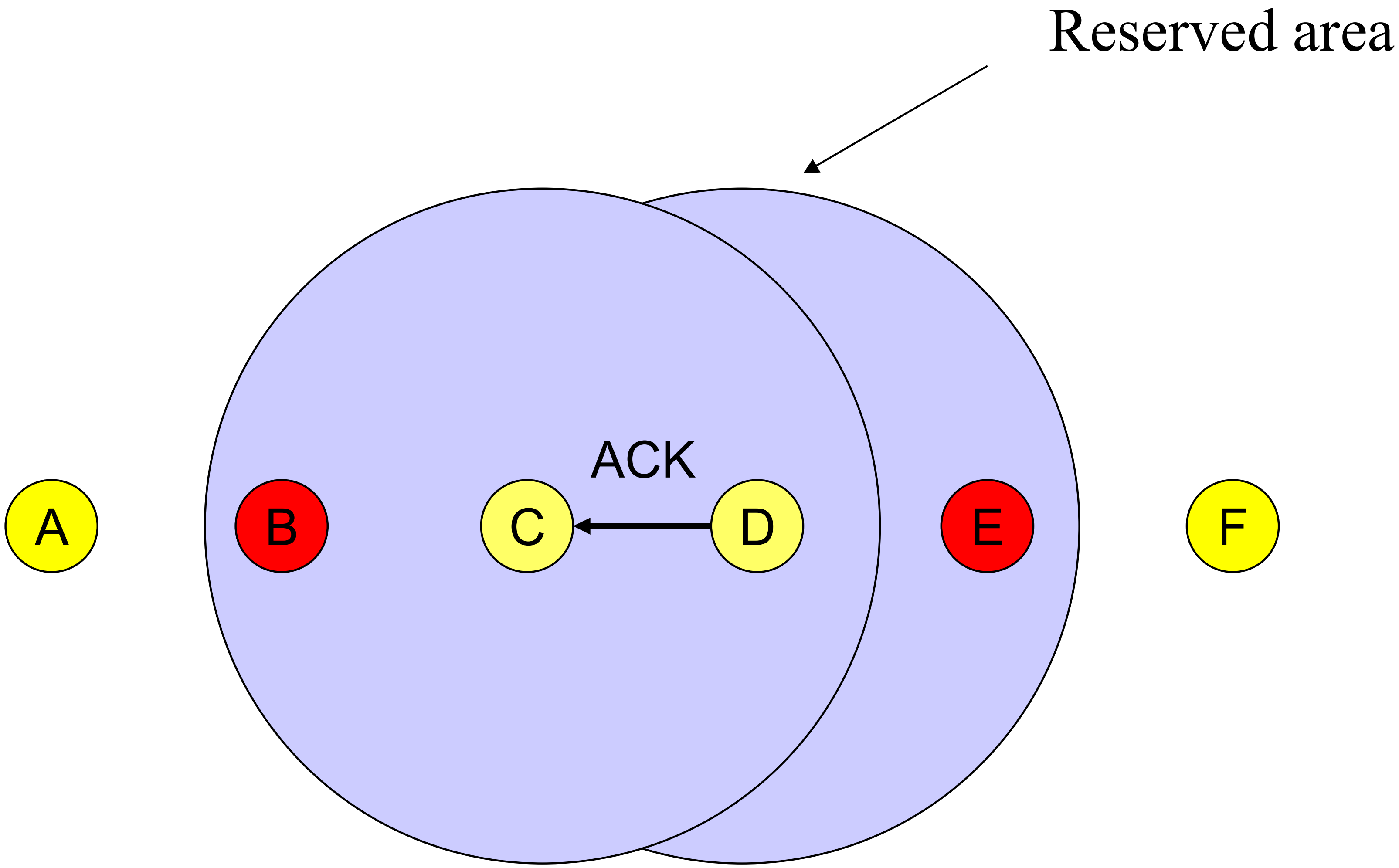
- DATA packet follows CTS. Successful data reception acknowledged using ACK.



IEEE 802.11



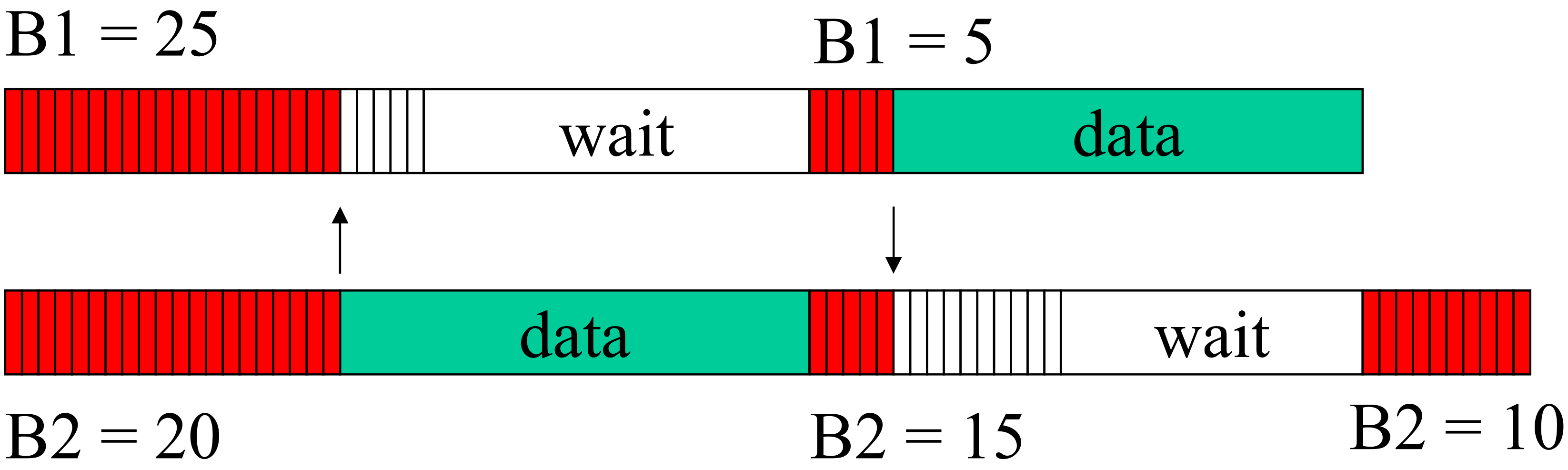
IEEE 802.11



Backoff Interval

- When transmitting a packet, choose a backoff interval in the range [0, CW]
  - CW is contention window
- Count down the backoff interval when medium is idle
  - Count-down is suspended if medium becomes busy
- Transmit when backoff interval reaches 0

DCF Example



cw = 31

B1 and B2 are backoff intervals at nodes 1 and 2

Backoff Interval

- The time spent counting down backoff intervals is a part of MAC overhead

- Important to choose CW appropriately
  - large CW → large overhead
  - small CW → may lead to many collisions (when two nodes count down to 0 simultaneously)

Backoff Interval (Cont.)

- Since the number of nodes attempting to transmit simultaneously may change with time, some mechanism to manage contention is needed

- IEEE 802.11 DCF: contention window CW is chosen dynamically depending on collision occurrence

Binary Exponential Backoff in DCF

- When a node fails to receive CTS in response to its RTS, it increases the contention window
  - CW is doubled (up to an upper bound)
  - More collisions ➔ longer waiting time to reduce collision
- When a node successfully completes a data transfer, it restores CW to CW_{min}

Sample Question

- Why we don't use CSMA/CD in wireless networks?

Sample Question (Cont.)

- 802.11 physical layer transmission rate = 6Mbps, MAC layer data payload = 1000 bytes, MAC header = 28 bytes, ACK Frame Size = 14 bytes, RTS length = 20 bytes, CTS length = 14 bytes, Propagation Delay = 1μs, Slot Time = 9 μs, SIFS Time = 16μs, DIFS Time = 34μs, Physical layer overhead = 20μs, Cwin = 31, RTS, CTS, and ACK are always sent at 6 Mbps.
- Suppose an 802.11 station wants to transmit 1,000 bytes of data, and all other stations are idle at this time. Ignore the propagation delay and assume no bit errors, what is the total time required to transmit the frame and receive the ACK when RTS/CTS is turned off?
- What is the MAC layer data rate?
- What is the efficiency?
- Repeat the above when RTS/CTS is turned on.
- Repeat the above when transmission rate = 54 Mbps.

Answer

- Assuming DCF basic access with one station, and using the given CW = 31 → average backoff of $\mathbb{E}[BO] = \frac{31}{2} \cdot 9\mu s = 15.5 \cdot 9 = 139.5\mu s$.

  Also assume every transmitted frame (DATA/ACK/RTS/CTS) includes the PHY overhead = 20 µs.
- Given:
- Payload = 1000 B → 8000bits
- DATA frame size = payload + MAC hdr = $1000 + 28 = 1028\text{B} \rightarrow 8224\text{bits}$
- ACK = 14 B, RTS = 20 B, CTS = 14 B
- DIFS = 34 µs, SIFS = 16 µs
- Useful per-frame transmit times:
- At 6 Mbps:
- $T_{DATA,6} = 20 + \frac{8224}{6e6} \cdot 10^6 = 20 + 1370.667 = 1390.667\ \mu s$
- $T_{ACK} = 20 + \frac{112}{6e6} \cdot 10^6 = 20 + 18.667 = 38.667\ \mu s$
- $T_{RTS} = 20 + \frac{160}{6e6} \cdot 10^6 = 46.667\ \mu s$
- $T_{CTS} = T_{ACK} = 38.667\ \mu s$
- At 54 Mbps (DATA only):
- $T_{DATA,54} = 20 + \frac{8224}{54e6} \cdot 10^6 = 20 + 152.296 = 172.296\ \mu s$

Answer (Cont.)

- A) RTS/CTS OFF, PHY rate = 6 Mbps
- Sequence: DIFS + Backoff + DATA + SIFS + ACK
- $T_{tot} = 34 + 139.5 + 1390.667 + 16 + 38.667 = 1618.833\ \mu s$
- MAC layer data rate (goodput):
- $R_{MAC} = \frac{8000\ bits}{1618.833\ \mu s} = 4.942\ Mbps$
- Efficiency (goodput / PHY rate):
- $\eta = \frac{4.942}{6} = 0.824 \approx 82.4\%$

Answer (Cont.)

- B) RTS/CTS ON, PHY rate = 6 Mbps
- Sequence: DIFS + Backoff + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK
- $$T_{tot} = 34 + 139.5 + 46.667 + 16 + 38.667 + 16 + 1390.667 + 16 + 38.66$$ $$= 1736.167\ \mu s$$
- MAC layer data rate (goodput):
- $R_{MAC} = \frac{8000}{1736.167\ \mu s} = 4.608\ Mbps$
- Efficiency:
- $\eta = \frac{4.608}{6} = 0.768 \approx 76.8\%$

Answer (Cont.)

- C) RTS/CTS OFF, DATA rate = 54 Mbps (RTS/CTS/ACK still 6 Mbps)
- Sequence: DIFS + Backoff + DATA(54) + SIFS + ACK(6)
- $T_{tot} = 34 + 139.5 + 172.296 + 16 + 38.667 = 400.463\ \mu s$
- MAC layer data rate (goodput):
- $R_{MAC} = \frac{8000}{400.463\ \mu s} = 19.977\ Mbps$
- Efficiency (relative to 54 Mbps):
- $\eta = \frac{19.977}{54} = 0.370 \approx 37.0\%$

Answer (Cont.)

- D) RTS/CTS ON, DATA rate = 54 Mbps (RTS/CTS/ACK still 6 Mbps)
- Sequence: DIFS + Backoff + RTS + SIFS + CTS + SIFS + DATA(54) + SIFS + ACK
- $T_{tot} = 34 + 139.5 + 46.667 + 16 + 38.667 + 16 + 172.296 + 16 + 38.667 = 517.796\ \mu s$
- MAC layer data rate (goodput):
- $R_{MAC} = \frac{8000}{517.796\ \mu s} = 15.450\ Mbps$
- Efficiency (relative to 54 Mbps):
- $\eta = \frac{15.450}{54} = 0.286 \approx 28.6\%$

Network Layer

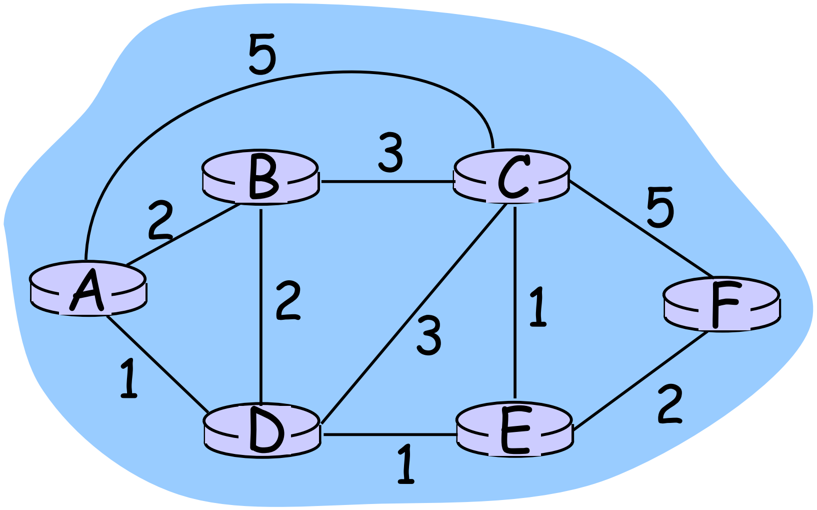
Routing

Routing protocol

Goal: determine "good" path (sequence of routers) thru network from source to dest.

Graph abstraction for routing algorithms:

- graph nodes are routers
- graph edges are physical links
  - link cost: delay, $ cost, or congestion level



- "good" path:
  - typically means minimum cost path
  - other def's possible

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ('source") to all other nodes
  - gives routing table for that node
- iterative: after k iterations, know least cost path to k dest.'s

Notation:

- c(i,j): link cost from node i to j. cost infinite if not direct neighbors
- D(v): current value of cost of path from source to dest. V
- p(v): predecessor node along path from source to v, that is next v
- N: set of nodes whose least cost path definitively known

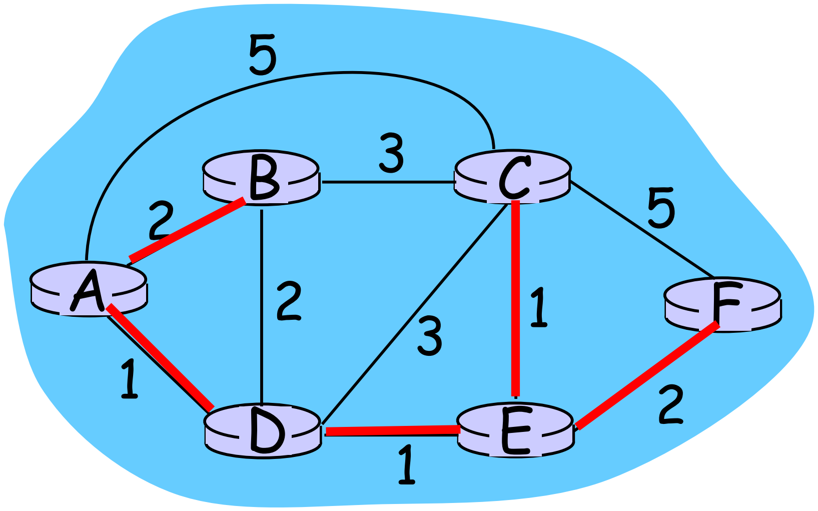
Dijsktra's Algorithm

```
Initialization:
  N = {A}
  for all nodes v
    if v adjacent to A
      then D(v) = c(A,v)
      else D(v) = infinity

Loop
  find w not in N such that D(w) is a minimum
  add w to N
  update D(v) for all v adjacent to w and not in N:
      D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N
```

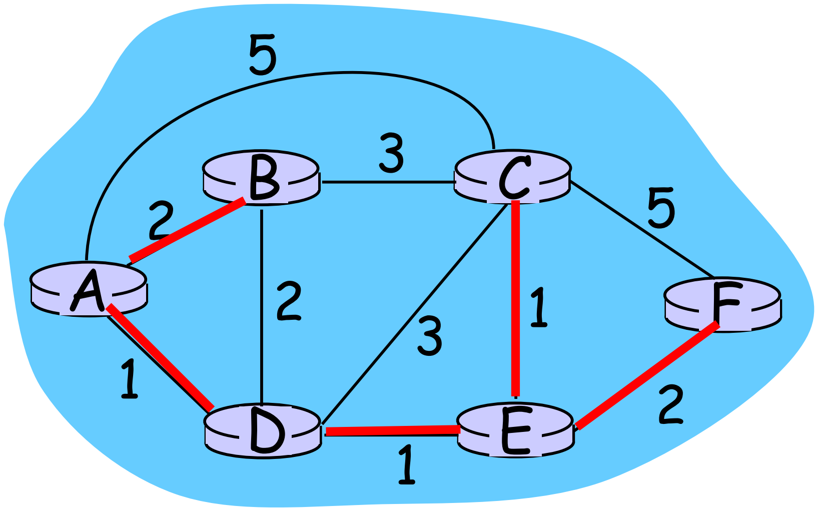
Dijkstra's algorithm: example

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A			1,A		
1						
2						
3						
4						
5						



Dijkstra's algorithm: example

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinity	infinity
1	AD	2,A	4,D		2,D	infinity
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



Distance Vector Routing Algorithm

iterative:

- continues until no nodes exchange info.
- *self-terminating*: no "signal" to stop

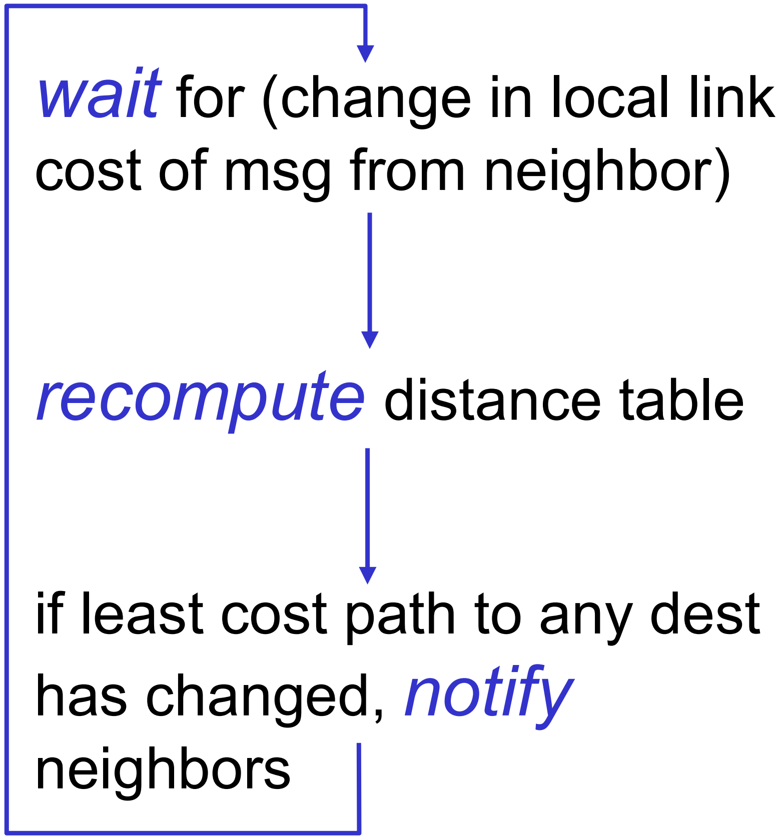
asynchronous:

- nodes need *not* exchange info/iterate in lock step!

distributed:

- each node communicates *only* with directly-attached neighbors

Each node:



wait for (change in local link cost of msg from neighbor)

recompute distance table

if least cost path to any dest has changed, *notify* neighbors

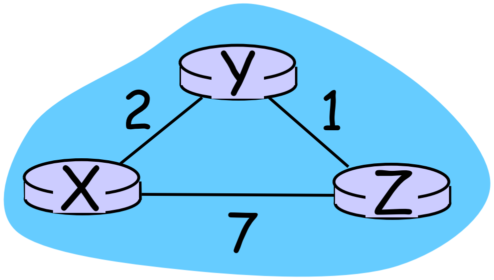
Distance Vector Algorithm: Data Structures

- Each node *x* maintains:
  - For each neighbor *v*, cost *c(x,v)*
  - Node *x*'s distance vector: $D_x = [D_x(y): y \in N]$ containing x's estimate of cost to all destinations
  - Distance vectors for each neighbor v: $D_v = [D_v(y): y \in N]$

- Basic operation: Bellman-Ford algorithm

$$D_x(y) = \min_v \{c(x,v) + D_v(y)\} \quad y \in N$$

Distance Vector Algorithm:



At all nodes, X:

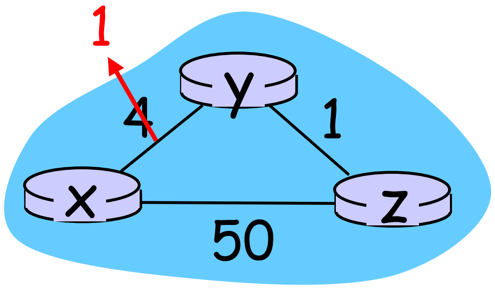
```
Initialization:
For all destinations $y \in N$:
    $D_x(y) = c(x,y)$ /* if y is not a neighbor, then $c(x,y) = \infty$ */:
For each neighbor w
    $D_w(y) = \infty$ for all destinations $y \in N$
For each neighbor w
    Send distance vector $D_x = [D_x(y): y \in N]$ to w

Loop:
Wait (until communication from neighbor w)
For each $y \in N$:
    $D_x(y) = \min_v \{c(x,v) + D_v(y)\}$
If $D_x(y)$ changes for any destination y
    Send distance vector $D_x = [D_x(y): y \in N]$ to all neighbors
```

Distance Vector: link cost changes

Link cost changes:

- node detects local link cost change
- updates routing info, recalculates distance vector
- if DV changes, notify neighbors



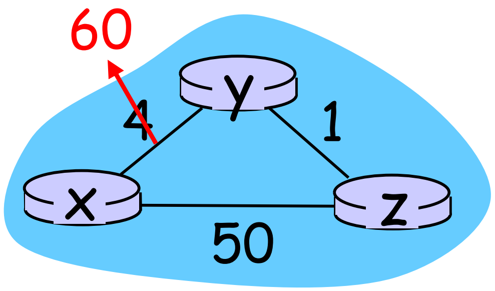
"good news travels fast"

At time t_0, *y* detects the link-cost change, updates its DV, and informs its neighbors.

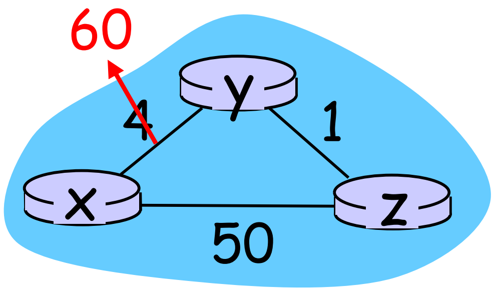
At time t_1, *z* receives the update from *y* and updates its table. It computes a new least cost to *x* and sends its neighbors its DV.

At time t_2, *y* receives *z*'s update and updates its distance table. *y*'s least costs do not change and hence *y* does *not* send any message to *z*.

Distance Vector: link cost changes

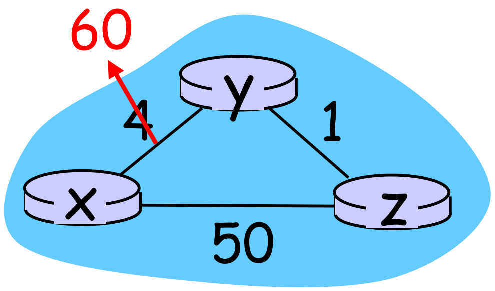


Distance Vector: link cost changes



$$Dy(x) = \min\{c(y,x)+Dx(x),\ c(y,z)+Dz(x)\}$$
$$= \min\{60+0,\ 1+5\} = 6$$

Distance Vector: link cost changes



Link cost changes:

- good news travels fast
- bad news travels slow - "count to infinity" problem!
- 44 iterations before algorithm stabilizes

Poissoned reverse:

- If Z routes through Y to get to X :
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- will this completely solve count to infinity problem?

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent each
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: O(n^2) algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

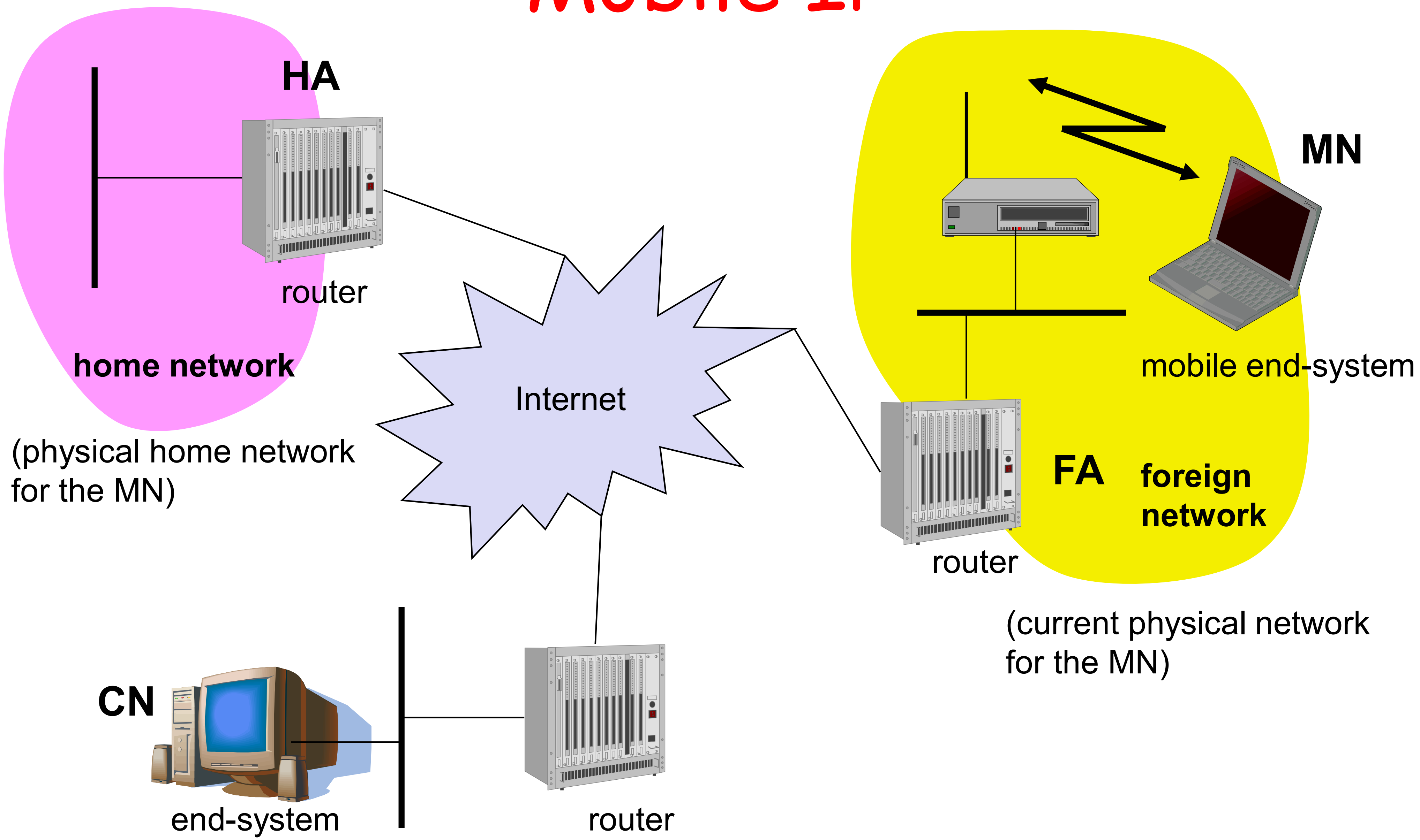
LS:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

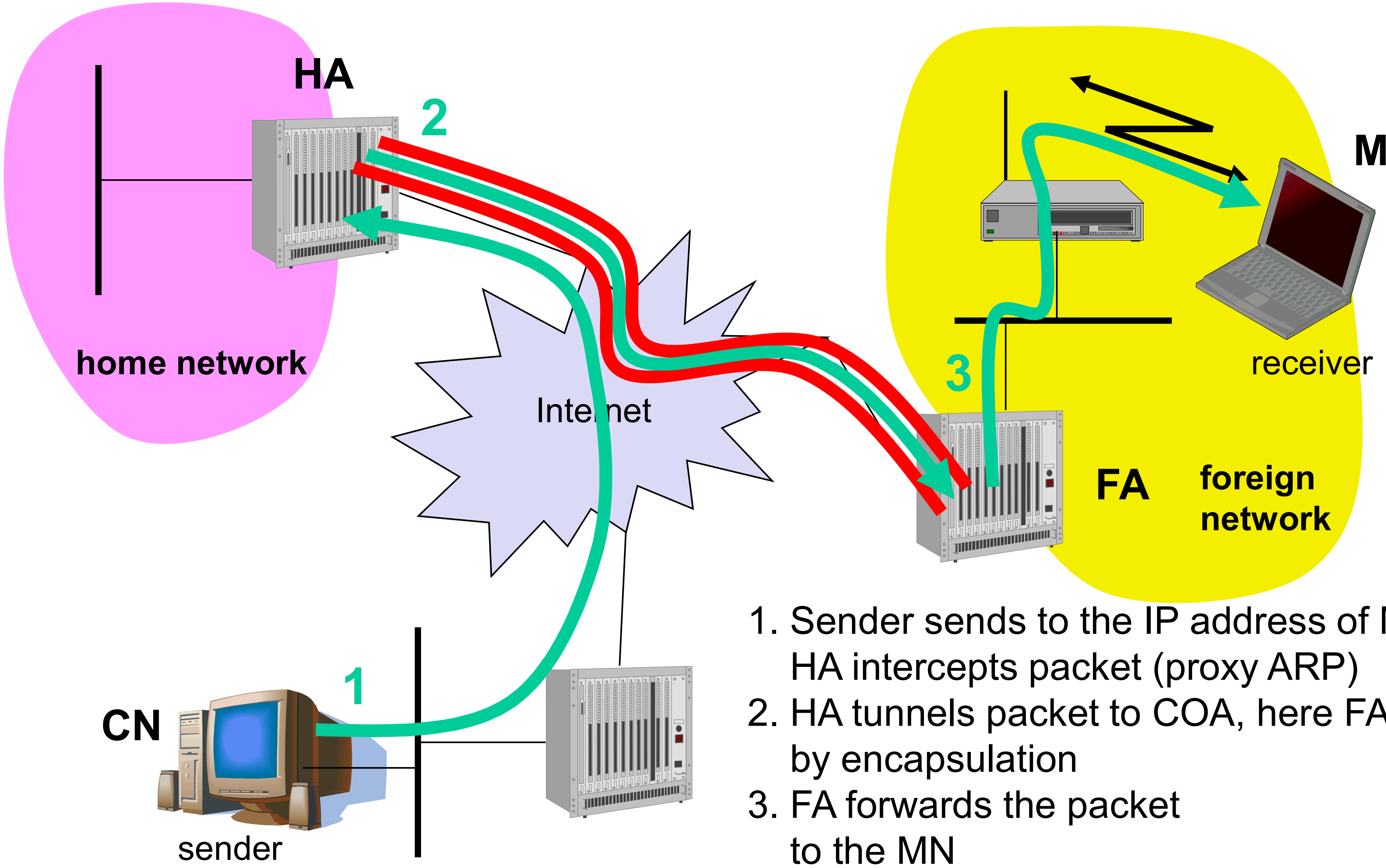
DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
  - error propagate thru network

Mobile IP

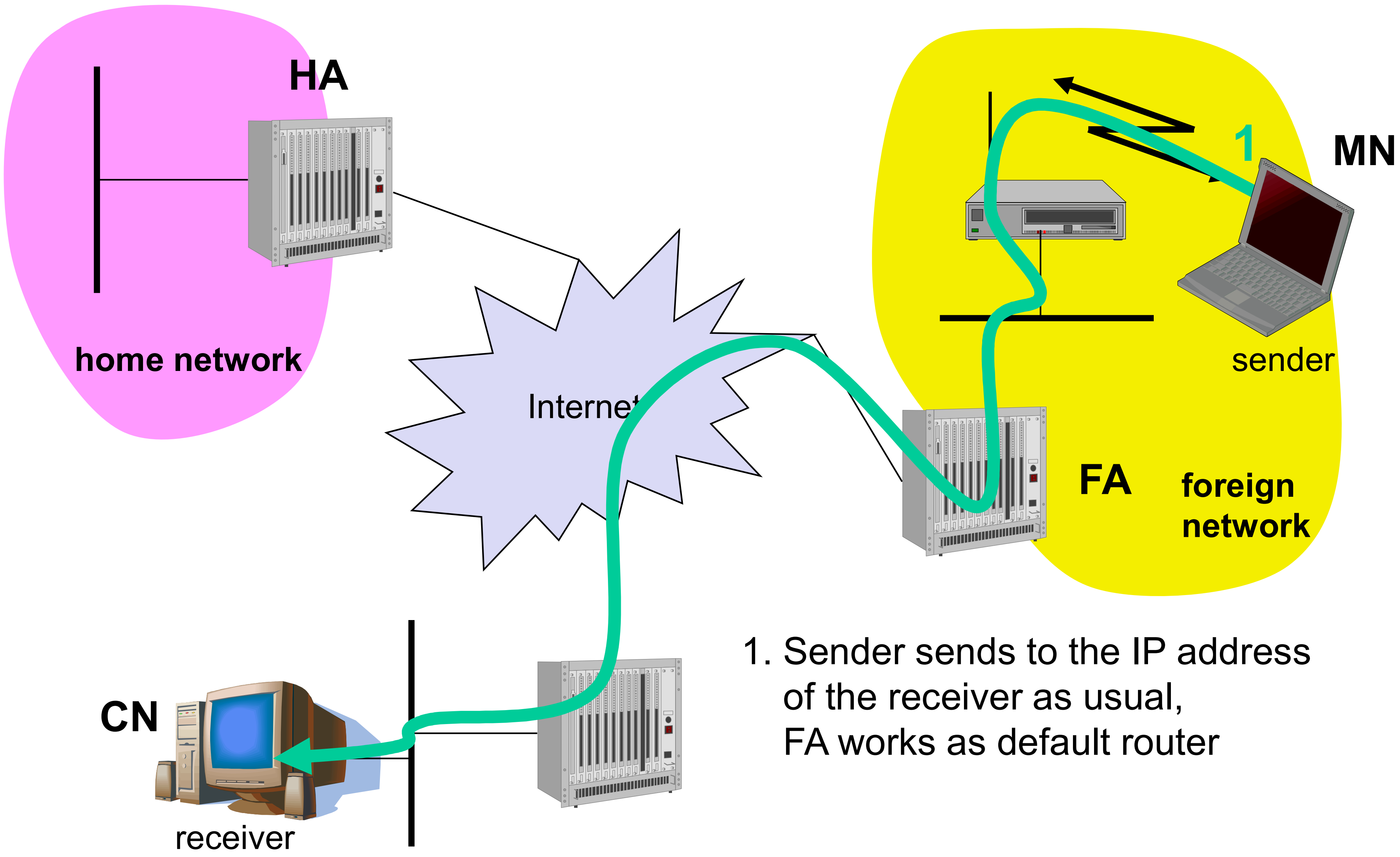

HA
router
home network
(physical home network
for the MN)
Internet
MN
mobile end-system
FA
foreign
network
router
(current physical network
for the MN)
CN
end-system
router

Data transfer to the mobile system



1. Sender sends to the IP address of MN, HA intercepts packet (proxy ARP)
2. HA tunnels packet to COA, here FA, by encapsulation
3. FA forwards the packet to the MN

Data transfer from the mobile system



1. Sender sends to the IP address
 of the receiver as usual,
 FA works as default router

Name one difference and one commonality between DSR and ADOV

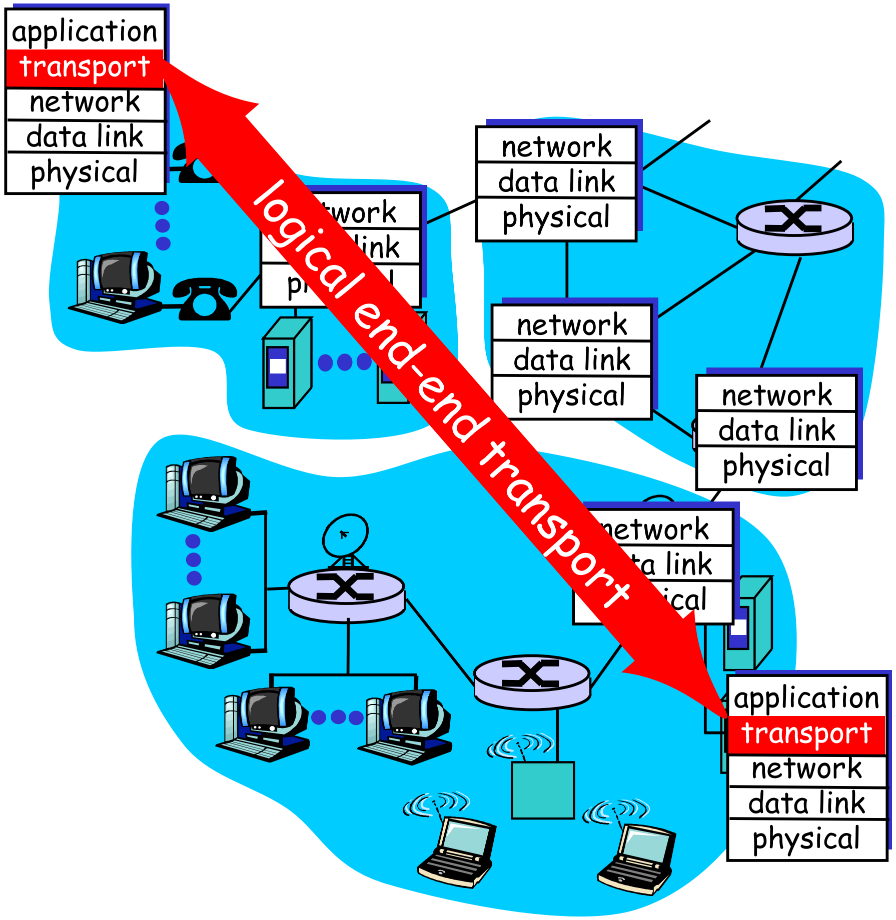
Comparison between DSR and AODV

	DSR	ADOV
Commonality	-Reactive routing -Discover routes via flooding -Obtain routes from route reply -Send route error upon failure detection	
Differences	Source routing	Distance vector routing
	Aggressive cache routes	Proactively delete expired route cache

Transport Layer

Transport services and protocols

- provide *logical communication* between app processes running on different hosts
- transport protocols run in end systems
  - send side: breaks app messages into segments, passes to network layer
  - rcv side: reassembles segments into messages, passes to app layer



Transport vs. network layer

- *network layer:* logical communication between hosts
- *transport layer:* logical communication between processes
  - relies on and enhances, network layer services

Household analogy:

12 kids sending letters to 12 kids

- processes = kids
- app messages = letters in envelopes
- hosts = houses
- transport protocol = Ann and Bill
- network-layer protocol = postal service

UDP: User Datagram Protocol [RFC 768]

- "no frills," "bare bones" Internet transport protocol
- "best effort" service, UDP segments may be:
  - lost
  - delivered out of order
- *connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small segment header
- no congestion control: UDP can blast away as fast as desired

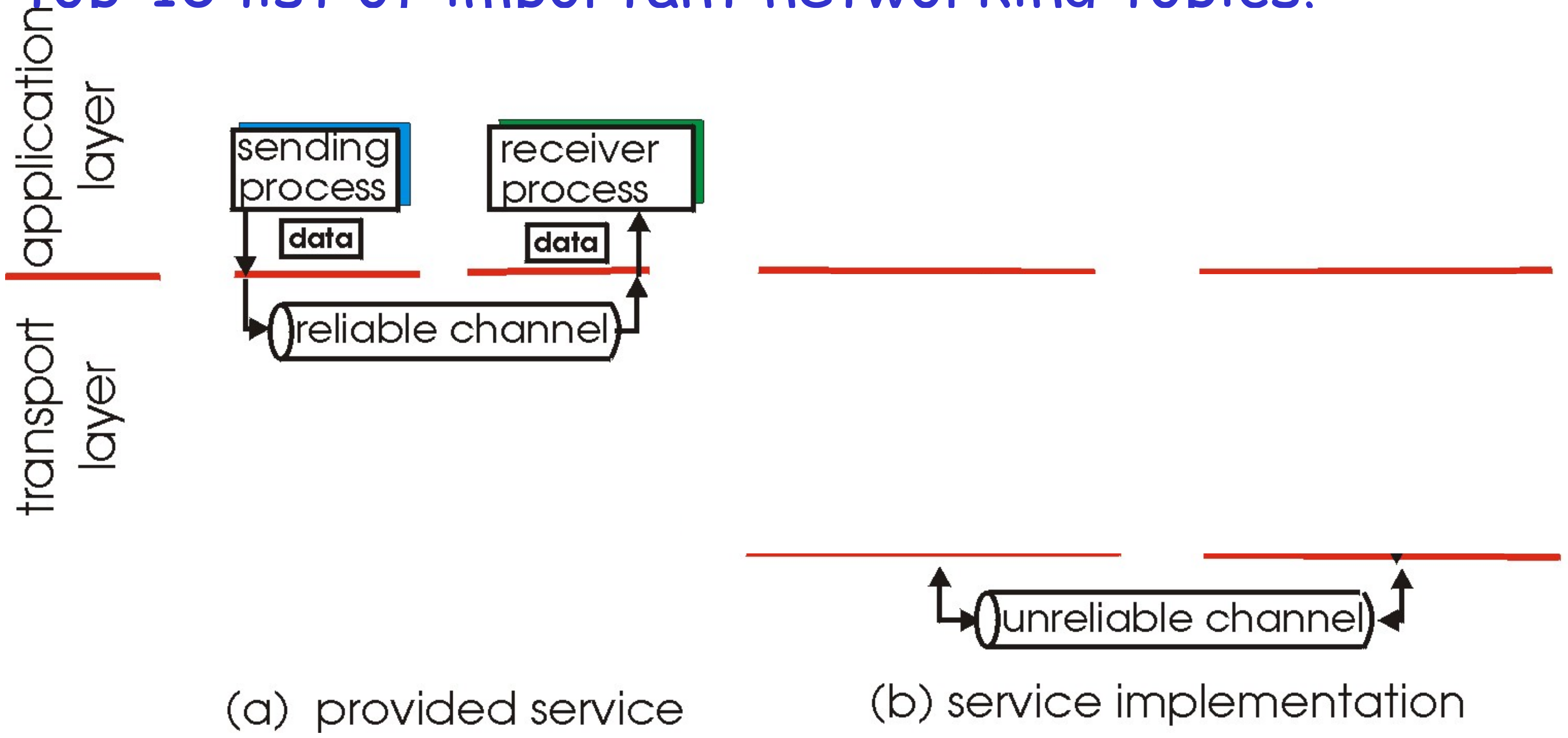
TCP: Overview

RFCs: 793, 1122, 1323, 2018, 2581

- point-to-point:
  - one sender, one receiver
- reliable, in-order *byte steam:*
  - no "message boundaries"
- pipelined:
  - TCP congestion and flow control set window size
- *send & receive buffers*
- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- connection-oriented:
  - handshaking (exchange of control msgs) init's sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver

Principles of Reliable data transfer

- important in app., transport, link layers
- top-10 list of important networking topics!



(a) provided service

(b) service implementation

- characteristics of unreliable channel will determine complexity of reliable data transfer protocol (rdt)

Reliable Data Transfer

Reliable data transfer over a reliable channel

- over a reliable channel
- over a channel with error
  - Checksum, NACK + ACK
- over a channel with error and loss
  - Checksum, ACK, timeout, seqno

Transport Layer Protocols

- Stop-and-wait
  - Achieve reliability
  - Suffer low throughput
- Increase throughput using pipelining
  - Go-back-N
  - Selective repeat

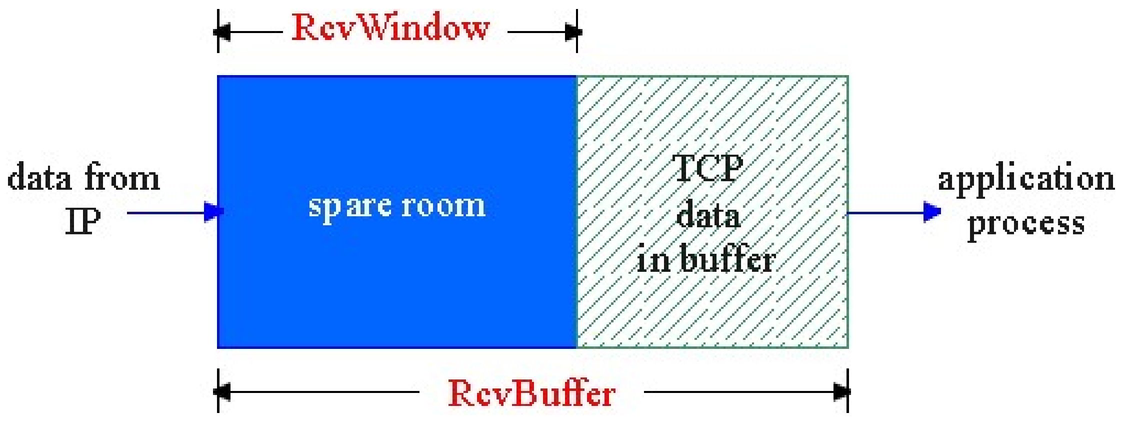
Reliable Data Transfer Mechanisms

Mechanism	Details
Checksum	Detect bit errors
Timer	Detect packet loss at sender
Sequence number	Detect packet loss and duplicates at receiver
ACK	Inform sender that pkt has been received
NACK	Inform sender that pkt has not been received correctly
Window, pipelining	Increase throughput, and adapt to receiver buffer size and network congestion

Flow control vs. congestion control

TCP Flow Control

- receive side of TCP connection has a receive buffer:



- app process may be slow at reading from buffer

flow control

sender won't overflow receiver's buffer by transmitting too much, too fast

- speed-matching service: matching the sending rate to the receiving app's drain rate
- Rcvr advertises spare room by including value of `RcvWindow` in segments
- Sender limits unACKed data to `RcvWindow`
  - guarantees receive buffer doesn't overflow

Principles of Congestion Control

Congestion:

- informally: “too many sources sending too much data too fast for *network* to handle”
- different from flow control
- manifestations:
  - lost packets (buffer overflow at routers)
  - long delays (queueing in router buffers)
- a top-10 problem!

Approaches towards congestion control

Two broad approaches towards congestion control:

End-end congestion control:

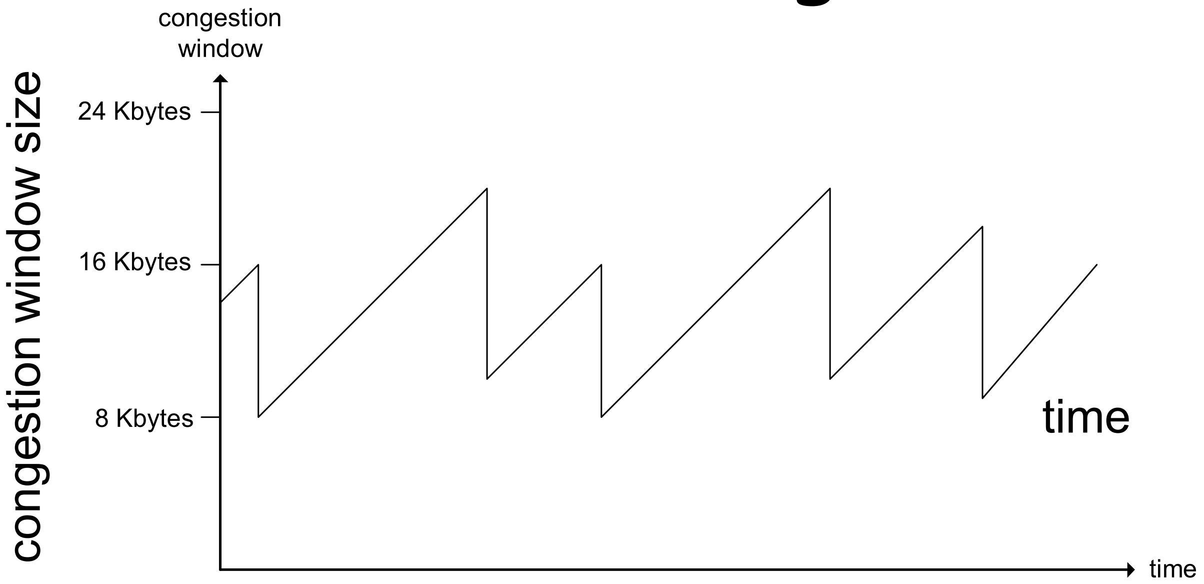
- no explicit feedback from network
- congestion inferred from end-system observed loss, delay
- approach taken by TCP

Network-assisted congestion control:

- routers provide feedback to end systems
  - single bit indicating congestion (SNA, DECbit, TCP/IP ECN, ATM)
  - explicit rate sender should send at (XCP)

TCP congestion control: additive increase, multiplicative decrease

- *Approach:* increase transmission rate (window size), probing for usable bandwidth, until loss occurs
  - *additive increase:* increase **CongWin** by 1 MSS every RTT until loss detected
  - *multiplicative aecrease*: cut **CongWin** in half after loss



Saw tooth behavior: probing for bandwidth

TCP Congestion Control: details

- sender limits transmission:

  LastByteSent-LastByteAcked $\leq$ CongWin

- Roughly,

$$\boxed{\text{rate} = \frac{\text{CongWin}}{\text{RTT}} \text{ Bytes/sec}}$$

- **Both CongWin** and RTT are time-varying

How does sender perceive congestion?

- loss event = timeout *or* 3 duplicate acks
- TCP sender reduces rate (**CongWin**) after loss event

three mechanisms:

- AIMD
- slow start
- conservative after timeout events

Summary: TCP Congestion Control

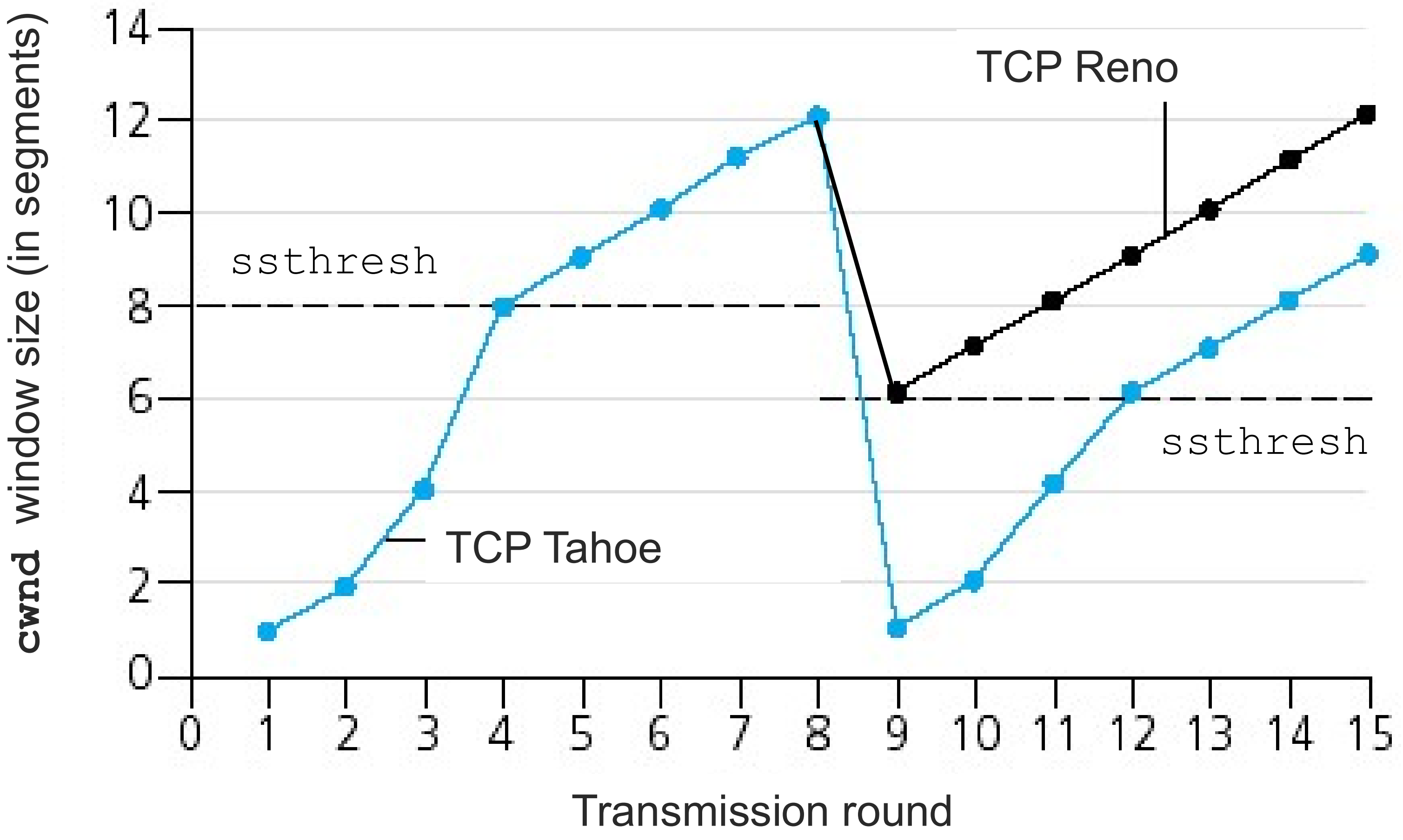
- When `CongWin` is below `Threshold`, sender in slow-start phase, window grows exponentially.

- When `CongWin` is above `Threshold`, sender is in congestion-avoidance phase, window grows linearly.

- When a triple duplicate ACK occurs, `Threshold` set to `CongWin/2` and `CongWin` set to `Threshold`.

- When timeout occurs, `Threshold` set to `CongWin/2` and `CongWin` is set to 1 MSS.

Popular “flavors” of TCP



TCP in Wireless Networks

- Transmission errors
  - Random errors
  - Burst errors
- Mobility
  - Infrastructure wireless networks
  - Wireless ad hoc networks

Various Schemes

- Link-layer retransmissions
- Split connection approach
- TCP-Aware link layer
- TCP-Unaware approximation of TCP-aware link layer

Hide losses

- Inferring reasons for packet losses
  - Explicit notification
  - Receiver-based discrimination
  - Sender-based discrimination

Avoid congestion Control under non-congestion losses

Summary

Schemes	Idea	Who	Characteristics
Link layer	Link layer retx	Wireless end points	Hide wireless error
Split connection	Local retx + Independent optimization of wireless conn.	Base station	Hide wireless error
Snoop	Link layer retx + drop dup ACK at base station	Base station	Hide wireless error + avoid unnecessary cwnd reduction
Delayed dup ACK	Link layer retx + delay dup ACK at wireless host	Base station Wireless host	Hide wireless error + avoid unnecessary cwnd reduction

Comparison (Cont.)

Schemes	Idea	Who	Characteristics
ELN	Tag dup ack with ELN if loss occurs at wireless link	Base station	Avoid unnecessary cwnd reduction
Receiver	When receiver believes that packet loss is due to errors, it sends a notification to the TCP sender	Receiver Sender	Avoid unnecessary cwnd reduction
Sender	When sender believes that packet loss is due to errors, it does not reduce cwnd.	Sender	Avoid unnecessary cwnd reduction

Techniques to Improve TCP Performance in Presence of Mobility

Classification

- Hide mobility from the TCP sender
- Make TCP adaptive to mobility

Sample Question

- What is the utilization of stop-and-wait protocol for a 10Mbps path and 500 byte packets, where the distance between source and destination is 1000 km?

- What if the distance decreases to 200 m?

Quiz 1

- A communication link has:
- Bandwidth B = 10 MHz
- Signal-to-noise ratio SNR = 20 dB
- Convert the SNR to non-dB form. [4 points]
- Compute the maximum achievable data rate (Mbps) according to Shannon's capacity formula. [6 points]

Quiz 2

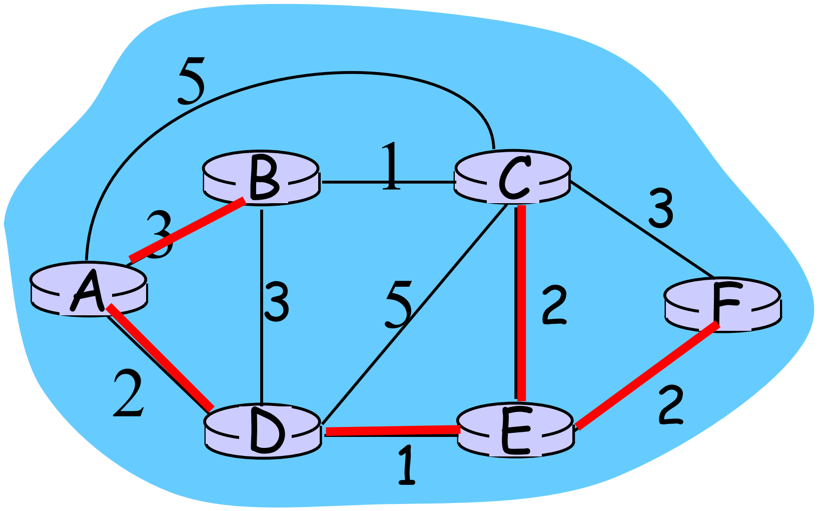
- Nodes A and B are within carrier-sense range of each other and both have saturated traffic destined for node C. The network uses IEEE 802.11 DCF with RTS/CTS (RTS-CTS-DATA-ACK). At time t_0, node D finishes a transmission and the channel becomes idle. After the channel has been idle for DIFS, nodes A and B start (or resume) their backoff countdown. For this problem, assume A and B both initialize their first backoff counters at $t_0$and begin counting down after DIFS according to standard DCF rules.
  Backoff sequences (in slots) for successive transmission attempts are predetermined as follows (use in order; if another attempt is needed beyond the list, stop):

  Node A: 5, 7, 2, 4. 6
  Node B: 5, 3, 5, 2, 7

Quiz 3

- Dijkstra’s algorithm

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A					
1						
2						
3						
4						
5						



2) Name two differences between link-state routing vs. distance vector routing. (2 points)

3) What is encapsulation? How is encapsulation used in mobile IP? (3 points)

HW 1

- 1. Convert 10 W (i) into dBm, and (ii) into dBW. Convert 0.01 mW to (i) dBm, and (ii) into dBW (5 points each).

HW 1 (Cont.)

- 2. Suppose a transmitter produces 1 mW power, what's the received power (in dBm) at distance 100 meters away from the transmitter under free-space propagation model? What's the received power under two-ray ground reflection model? (Both transmitter and receiver antenna gain is 1, the transmitter's height is 10m and the receiver's height is 1m. Carrier frequency is 2.4 GHz.) (30 points)

HW 1 (Cont.)

- 3. What's the minimum signal-to-noise ratio in order to reliably support 20 Mbps using a channel spanning from 5 GHz to 5.05 GHz? What is the minimum received signal strength if the background noise is -100 dBm? (26 points)

- 4. You are designing two wireless systems: 1. A high-throughput Wi-Fi link 2. A satellite communication system (a) Choose one digital modulation scheme for each system. (b) Justify your choice. (c) Give one advantage and one disadvantage of each scheme.

HW 2

- Nodes A and B are within carrier-sense range of each other and both have saturated traffic destined for node C. The network uses IEEE 802.11 DCF with RTS/CTS (RTS–CTS–DATA–ACK). At time t, node D finishes a transmission and the channel becomes idle. After the channel has been idle for DIFS, nodes A and B start (or resume) their backoff countdown. For this problem, assume A and B both initialize their first backoff counters at t□and begin counting down after DIFS according to standard DCF rules. Backoff sequences (in slots) for successive transmission attempts are predetermined as follows (use in order; if another attempt is needed beyond the list, stop):
- Node A: 6, 3, 5, 6, 4
- Node B: 6, 8, 6, 2, 7

HW 2

- Hidden terminal w/o RTS/CTS

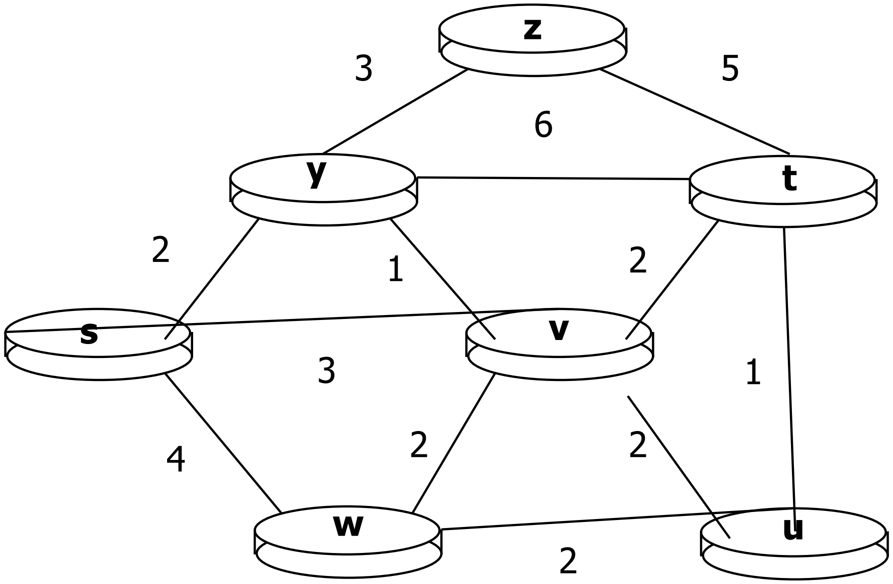
- Hidden terminal w/ RTS/CTS

HW 2

- If the NAV mechanism were removed but physical carrier sensing remained, what problems would arise? Explain with at least one concrete scenario (e.g., hidden terminals, exposed terminals, or virtual carrier sense use cases).

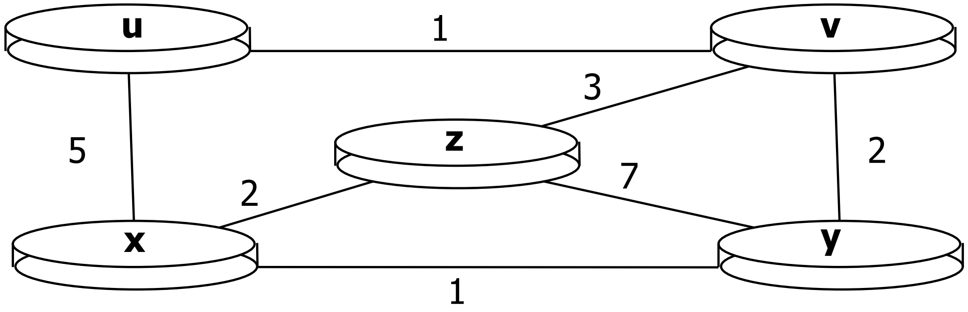
HW 3

- Consider the following network. With the indicated link costs, use Dijkstra’s shortest-path algorithm to compute the shortest path from *s* to all network nodes. Show how the algorithm works by computing a table similar to our lecture slide.



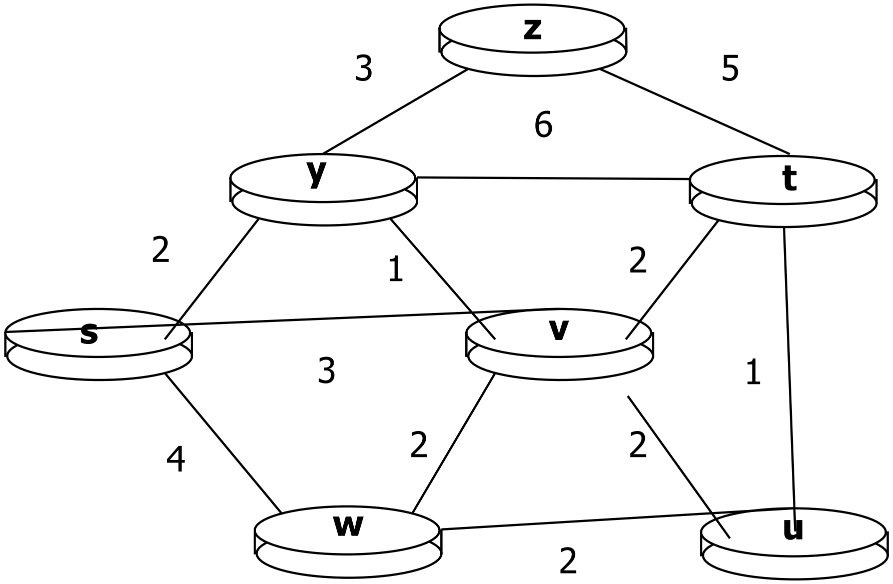
HW 3

- Consider the network shown below, and assume that each node initially only knows the costs to each of its neighbors. Consider the distance-vector algorithm and show how the distance table entries at node z evolve.



HW 3

- Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from *s* to all network nodes. Show how the algorithm works by computing a table similar to our lecture slide.



HW 3

- Consider that a TCP sender, Host A, wants to send a 10,000,000-bit MP3 file to Host B. The slowest link between Host A and Host B is 2 Mbps, the RTT (including transmission delays) is 100 ms, and the TCP MSS is 11,680 bits. The operating system of Host B sets a default receiver window of 16 segments and ssthresh to 16 segments. What is the network utilization in the first round trip time? What is the TCP/Reno throughput in the 5th round-trip time when the sender detects a packet loss using 3 duplicate ACKs during the 4th RTT?

Quiz 4

- When a mobile host is handing off to a new non-overlapping base station, what performance problem will it experience if it uses TCP? How to address this problem?

- What is the minimum congestion window size for the following network to achieve 100% utilization: 100 Mbps path, 1 KB packet, 50 kilometer distance between the sender and receiver, and the signal travels at the speed of light.

Quiz 1

- A communication link has:
- Bandwidth B = 10 MHz
- Signal-to-noise ratio SNR = 20 dB
- Convert the SNR to non-dB form. [4 points]
- Compute the maximum achievable data rate (Mbps) according to Shannon's capacity formula. [6 points]

Quiz 2

- Nodes A and B are within carrier-sense range of each other and both have saturated traffic destined for node C. The network uses IEEE 802.11 DCF with RTS/CTS (RTS-CTS-DATA-ACK). At time t_0, node D finishes a transmission and the channel becomes idle. After the channel has been idle for DIFS, nodes A and B start (or resume) their backoff countdown. For this problem, assume A and B both initialize their first backoff counters at $t_0$and begin counting down after DIFS according to standard DCF rules.
  Backoff sequences (in slots) for successive transmission attempts are predetermined as follows (use in order; if another attempt is needed beyond the list, stop):

  Node A: 5, 7, 2, 4. 6
  Node B: 5, 3, 5, 2, 7

Sample Question 1

- Why do we have protocol layers?

Sample Question 1

- Why do we have protocol layers?
  - Pros
    - Abstraction
    - Easy to reuse
    - Easy to maintain
  - Cons
    - Overhead with layering
    - Less transparent
    - Hard to optimize and troubleshoot

Sample Question 2

- P=1000 mW. What is P in dBm?
- P2 = 0.01 P1. What P2/P1 in dB?

Sample Question 4

- DSR is designed for mobile ad hoc networks. If we use it in wireless mesh networks, how well does it perform?

Sample Question 5

- What protocols we use to route in sensor networks? Why?

Good luck!