$$\text{Average throughput for A} = p_A(1-p_B) = 3 \cdot (\text{Average throughput for B})$$
$$\text{Average throughput for B} = p_B(1-p_A)$$

Next, we can use these two equations to solve for p_A in terms of p_B:

$$\begin{aligned} p_A(1-p_B) &= 3p_B(1-p_A) \\ p_A(1-p_B) + p_A(3p_B) &= 3p_B \\ p_A(1-p_B+3p_B) &= 3p_B \\ p_A(1+2p_B) &= 3p_B \\ p_A &= \frac{3p_B}{1+2p_B} \end{aligned}$$

$$3p(1-p)^{N-1} + [N-1]p(1-p)^{N-2}(1-3p)$$

The maximum rate is 12 Mbps, which occurs when the base station sends data exclusively (i.e., in every time slot) to node A. This solution, however, would not be "fair" because none of the other nodes ever receive any data.

To start with, we can observe that all four nodes are going to receive some fraction of the 1 second interval, and the time spent by all four nodes will fill each interval. Therefore, we can say that: $t_A + t_B + t_C + t_D = 1$.

Next, we can write out the relationship between these times, based on the relative differences between each node's downstream rate (for example, node A only needs 1/12 as much time as node D to receive the same amount of data): $12t_A = 6t_B = 3t_C = t_D$. At this point, we can solve for t_D, which is node D's fraction of the interval:

$$\begin{aligned} \frac{t_D}{12} + \frac{t_D}{6} + \frac{t_D}{3} + t_D &= 1 \\ \frac{t_D + 2t_D + 4t_D + 12t_D}{12} &= 1 \\ t_D &= \frac{12}{19} \end{aligned}$$

Now that we've solved for t_D, we can see that $t_A = \frac{1}{19}$, $t_B = \frac{2}{19}$, and $t_C = \frac{4}{19}$. Finally, the average transmission rate can be calculated as follows:

$$\frac{1}{19}(12\text{ Mbps}) + \frac{2}{19}(6\text{ Mbps}) + \frac{4}{19}(3\text{ Mbps}) + \frac{12}{19}(1\text{ Mbps}) = 2.53\text{ Mbps}$$

In this case, it turns out that we can maximize the total average transmission rate by having nodes A and B receive twice as much data as nodes C and D. As before, all four nodes will share some portion of the 1 second interval, such that $t_A + t_B + t_C + t_D = 1$. Let's solve for t_D again, followed by t_A, t_B, and t_C:

$$\begin{aligned} 6t_A = 3t_B &= 3t_C = t_D \\ \frac{t_D}{6} + \frac{t_D}{3} + \frac{t_D}{3} + t_D &= 1 \\ \frac{t_D + 2t_D + 2t_D + 6t_D}{6} &= 1 \end{aligned}$$
$$t_D = \frac{6}{11}, \text{ and therefore } t_A = \frac{1}{11}, t_B = \frac{2}{11}, \text{ and } t_C = \frac{2}{11}$$

Finally, this maximum average transmission rate can be calculated as follows:

$$\frac{1}{11}(12\text{ Mbps}) + \frac{2}{11}(6\text{ Mbps}) + \frac{2}{11}(3\text{ Mbps}) + \frac{6}{11}(1\text{ Mbps}) = 3.27\text{ Mbps}$$

[15 pts] Suppose an 802.11b station (with a transmission rate of 11 Mbps) is configured to always reserve the channel using the RTS/CTS sequence. Suppose this station suddenly wants to transmit 1000 bytes of data, and all other stations are idle. As a function of SIFS and DIFS, and ignoring propagation delay and assuming there are no bit errors, calculate the total time required to transmit the frame and receive the acknowledgment.

When the station wants to transmit, it will listen to the idle channel for a fixed amount of time (DIFS), and then send out a 32 byte RTS control frame (just an 802.11 frame with a 0 byte payload). When the station's destination receives the RTS frame, it will also wait for a fixed amount of time (SIFS), and then send out a 32 byte CTS control frame.

Once the original station receives the CTS frame, it will wait for SIFS, and then begin sending its entire 1000 + 32 byte frame of data. When the station's destination receives the data frame, it will again wait for SIFS, and then send a 32 byte ACK control frame back to the station.

Therefore, the total amount of time required for the station to send its 1000 bytes of data will be: DIFS + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK. Although DIFS and SIFS are not given for this problem, we can calculate the rest of the terms as follows:

DATA = time required to send 1000 bytes of data and 32 bytes of 802.11 MAC frame header
$$= \frac{1032\text{ bytes} \times 8\text{ bits/byte}}{11\text{ Mbps}} = 750.5\ \mu s$$
RTS, CTS, ACK = time required to send a 32 byte control frame with no payload (802.11 MAC header only)
$$= \frac{32\text{ bytes} \times 8\text{ bits/byte}}{11\text{ Mbps}} = 23.3\ \mu s$$

The final result is:

DIFS + 23.3 μs + SIFS + 23.3 μs + SIFS + 750.5 μs + SIFS + 23.3 μs = DIFS + 3*SIFS + 820.4 μs

encoded signal = (original data) X (chipping sequence)

decoding: inner-product of encoded signal and chipping sequence

- 802.11b: 2.4GHz-2.485GHz spectrum divided into 11 channels at different frequencies
  - AP admin chooses frequency for AP
  - interference possible: channel can be same as that chosen by neighboring AP!
- host: must *associate* with an AP
  - scans channels, listening for *beacon frames* containing AP's name (SSID) and MAC address
  - selects AP to associate with
  - may perform authentication [Chapter 8]
  - will typically run DHCP to get IP address in AP's subnet

passive scanning:

(1) beacon frames sent from APs
(2) association Request frame sent: H1 to selected AP
(3) association Response frame sent from selected AP to H1

active scanning:

(1) Probe Request frame broadcast from H1
(2) Probe Response frames sent from APs
(3) Association Request frame sent: H1 to selected AP
(4) Association Response frame sent from selected AP to H1

- *datagram* network provides network-layer *connectionless* service
- *virtual-circuit* network provides network-layer *connection* service
- analogous to TCP/UDP connecton-oriented / connectionless transport-layer services, but:
  - *service:* host-to-host
  - *no choice:* network provides one or the other
  - *implementation:* in network core

efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

- *suppose:* N nodes with many frames to send, each transmits in slot with probability p
- prob that given node has success in a slot = $p(1-p)^{N-1}$
- prob that *any* node has a success = $Np(1-p)^{N-1}$
- max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:
  max efficiency = 1/e = .37

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

CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

To find the value of p that maximizes slotted ALOHA's efficiency $E(p)$, we begin by taking the derivative of the efficiency with respect to p, and then setting it equal to 0 to find the value of p that maximizes the efficiency:

$$\begin{aligned} \frac{d(E(p))}{dp} &= \frac{d(Np(1-p)^{N-1})}{dp} \\ &= (Np)(N-1)(1-p)^{N-2}(-1) + (N)(1-p)^{N-1} \\ 0 &= N(1-p)^{N-2} \cdot [(1-p) - p(N-1)] \end{aligned}$$

At this point, we can see by inspection that setting $p = 1$ is one way to make the entire expression equal to 0 (via the $N(1-p)^{N-2}$ term). If $p = 1$, however, then all N nodes will always transmit in every slot; this must mean that $p = 1$ actually minimizes the efficiency. The other value of p that can make the entire derivative equal to 0 can be found by examining the $[(1-p) - p(N-1)]$ term:

$$\begin{aligned} 0 &= (1-p) - p(N-1) \\ p(N-1) &= 1-p \\ pN - p &= 1-p \\ p &= 1/N \end{aligned}$$

The value of p that maximizes slotted ALOHA's efficiency is therefore $1/N$.

Next, we can find the long-run efficiency by taking the limit as described in the problem:

$$\begin{aligned} \text{Long-run efficiency} &= \lim_{N\to+\infty} E(1/N) \\ &= \lim_{N\to+\infty} N \cdot (1/N) \cdot (1-1/N)^{N-1} \\ &= \lim_{N\to+\infty} (1-1/N)^{N-1} \\ &= \lim_{N\to+\infty} (1-1/N)^{N} \cdot (1-1/N)^{-1} \\ &= \lim_{N\to+\infty} \frac{(1-1/N)^N}{(1-1/N)} \\ \text{Long-run efficiency} &= \frac{(1/e)}{1} = 1/e \end{aligned}$$

a. What is Sender 2's output to the channel, before it is added to the signal from Sender 1? In other words, wha is $Z^2_{i,m}$?

- The first part of Sender 2's output, $Z^2_{0,m}$, is defined as: $d^2_0 \cdot c^2_m$. This is simply $(1)\cdot(1,-1,1,1,1,-1,1,1)$ $(1,-1,1,1,1,-1,1,1)$
- The second part of Sender 2's output, $Z^2_{1,m}$, is defined as: $d^2_1 \cdot c^2_m$. This is simply $(1)\cdot(1,-1,1,1,1,-1,1,1)$ $(1,-1,1,1,1,-1,1,1)$

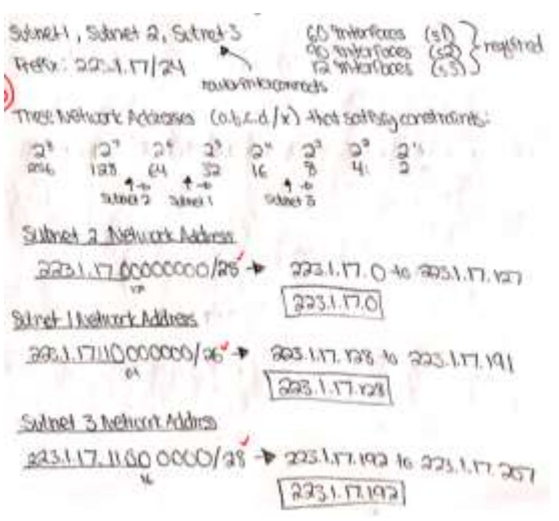
b. Suppose that a second receiver (Receiver 2) wants to receive the data sent by Sender 2. Show (by calculatio that the receiver is able to recover Sender 2's data from the combined signal ($Z^*_{i,m}$) by using Sender 2's cod

The first part of the combined signal, $Z^*_{0,m}$, is $(2,0,2,0,2,-2,0,0)$. We can recover Sender 2's data from th combined signal by using the following equation:

$$d^2_0 = \frac{\sum_{m=1}^{8} Z^*_{0,m} \cdot c^2_m}{8} = \frac{(1\cdot2)+(-1\cdot0)+(1\cdot2)+(1\cdot0)+(1\cdot2)+(-1\cdot-2)+(1\cdot0)+(1\cdot0)}{8} = 1$$

The second part of the combined signal, $Z^*_{1,m}$, is $(0,-2,0,2,0,0,2,2)$. We can recover Sender 2's data fro the combined signal by using the following equation:

$$d^2_1 = \frac{\sum_{m=1}^{8} Z^*_{1,m} \cdot c^2_m}{8} = \frac{(1\cdot0)+(-1\cdot-2)+(1\cdot0)+(1\cdot2)+(1\cdot0)+(-1\cdot0)+(1\cdot2)+(1\cdot2)}{8} = 1$$



Subnet 1, Subnet 2, Subnet 3 — 60 interfaces (s1), 90 interfaces (s2), 12 interfaces (s3) } required

Prefix: 223.1.17/24

routers interconnects

Three Network Addresses (a.b.c.d/x) that satisfy constraints:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1
256	128	64	32	16	8	4	2
	Subnet 2		Subnet 1		Subnet 3		

Subnet 2 Network Address

223.1.17.0 0000000/25 → 223.1.17.0 to 223.1.17.127

128

$\boxed{223.1.17.0}$

Subnet 1 Network Address

223.1.17.10 000000/26 → 223.1.17.128 to 223.1.17.191

64

$\boxed{223.1.17.128}$

Subnet 3 Network Address

223.1.17.1100 0000/28 → 223.1.17.192 to 223.1.17.207

16

$\boxed{223.1.17.192}$

graph: G = (N,E)

N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

algorithm complexity: n nodes

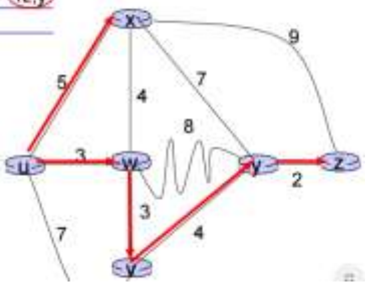
- each iteration: need to check all nodes, w, not in N
- n(n+1)/2 comparisons: $O(n^2)$
- more efficient implementations possible: O(nlogn)

Dijkstra's algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)


x
u
w
y
z
v
5
3
4
7
8
9
2
3
4
7