MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.02 Introduction to EECS II
Spring 2011

Quiz 2

Name Michael Plasmeier	*Score* 91.5

- ☒ 10a Devavrat Shah 24-402
- ☐ 11a Devavrat Shah 24-402
- ☐ 12n Fabian Lim 38-166
- ☐ 1p John Sun 38-166
- ☐ 2p John Sun 38-166

Please write your answers legibly in the spaces provided. You can use the backs of the pages if you need extra room for your answer or scratch work. Make sure we can find your answer!

You can use a calculator and one 8.5" x 11" cribsheet.

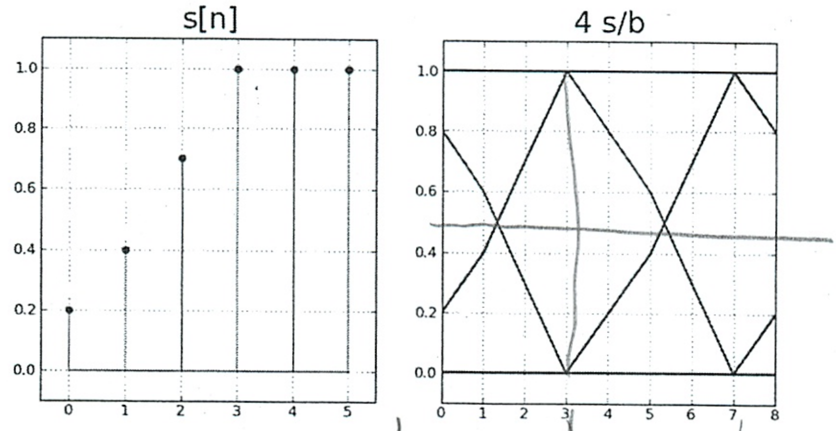
Partial credit will only be given in cases where you show your work and (very briefly) explain your approach.

Prob. #1 (20 pts)	Prob. #2 (20 pts)	Prob. #3 (20 pts)	Prob. #4 (20 pts)	Prob. #5 (20 pts)
20	17	19.5	20	15

Problem 1. Gaussian Noise and Bit Error Rates (20 points)

Consider the figure below, which shows the step response for a particular transmission channel along with the eye diagram for the channel response when transmitting 4 samples/bit.

x	$\Phi(x)$
-3.4	0.0003
-3.3	0.0005
-3.2	0.0007
-3.1	0.0010
-3.0	0.0013
-2.9	0.0019
-2.8	0.0026
-2.7	0.0035
-2.6	0.0047
-2.5	0.0062
-2.4	0.0082
-2.3	0.0107
-2.2	0.0139
-2.1	0.0179
-2.0	0.0228
-1.9	0.0287
-1.8	0.0359
-1.7	0.0446
-1.6	0.0548
-1.5	0.0668
-1.4	0.0808
-1.3	0.0968
-1.2	0.1151
-1.1	0.1357
-1.0	0.1587
-0.9	0.1841
-0.8	0.2119
-0.7	0.2420
-0.6	0.2743
-0.5	0.3085
-0.4	0.3446
-0.3	0.3821
-0.2	0.4207
-0.1	0.4602
-0.0	0.5000



Suppose there is additive Gaussian noise on this channel that sometimes causes a transmitted bit to be misidentified at the receiver. In answering the questions below, please assume that

- the receiver uses the optimal detection sample for each bit (corresponding to the "center" of the eye)
- the receiver uses a detection threshold of 0.5V
- 0 and 1 bits are transmitted with probability of 0.5
- the additive noise is independent of the bit being transmitted and has a Gaussian distribution with zero mean and standard deviation σ

(A) (8 points) When sending 4 samples/bit, if the bit error rate is measured to be .0062, what is the approximate value for σ, the standard deviation for the additive Gaussian noise? The table to the right shows $\Phi(x)$, the cumulative distribution function for the normal (Gaussian) distribution. Please show your work.

$BER = P(noise > .5 | 0)\, P(0) + P(noise < .5 | 1)\, P(1)$

$= \frac{1}{2} P(noise > .5 | 0) + \frac{1}{2} P(noise > .5 | 0)$

↑ same, since symetrical

$= P(noise > .5)$

$= P(\sigma N > .5)$

$= P(N > \frac{.5}{\sigma})$

$= P(N < -\frac{.5}{\sigma})$

$= \Phi(-\frac{.5}{\sigma}) = .0062$

$-2.5 = -\frac{.5}{\sigma}$

$-2.5\sigma = -.5$

$\cdot -2 \qquad \cdot -2$

$5\sigma = 1$

$\sigma = \frac{1}{5}$

Value for σ: $1/5$ ✓

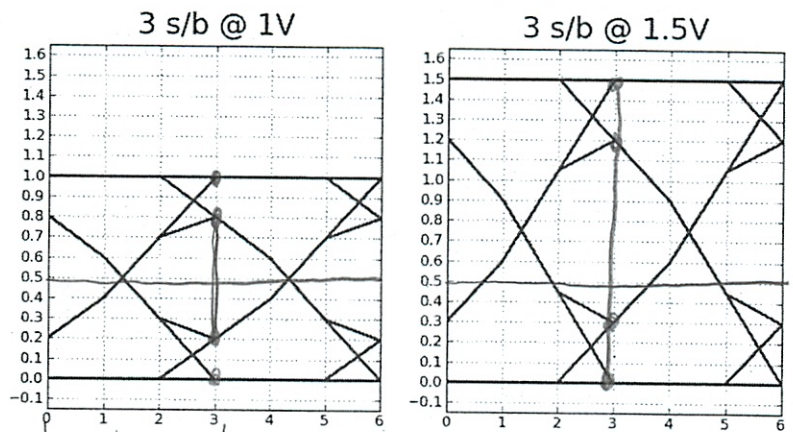
When transmitting at 3 bits/sample down the same channel, the designers are trying to decide between two alternatives for the transmitter design:

1. the transmitter sends 0V for a 0 bit and 1V for a 1 bit, resulting in the eye diagram shown on the left in the figure below. The possible sample voltages at the center of the eye are 0V, 0.2V, 0.8V and 1.0V.

2. the transmitter sends 0V for a 0 bit and 1.5V for a 1 bit, resulting in the eye diagram shown on the right in the figure below. The possible sample voltages at the center of the eye are 0V, 0.3V, 1.2V and 1.5V. The designers are hypothesizing that the increased transmission range will lower the bit error rate.

In either alternative *the receiver threshold is unchanged at 0.5V.*

0,1 = prob right



(B) (12 points) Assume that the Gaussian noise has $\sigma = 0.5$. For each alternative compute an estimate for the bit error rate using all four possible sample voltages. Indicate which alternative results in the fewest bit errors at the receiver. Please show your work.

$P(\text{error}|00)\,P(00) + P(\text{error}|01)\,P(01) + P(\text{error}|10)\,P(10) + P(\text{error}|11)\,P(11)$

$\frac{1}{4}\cdot P(\sigma N > .5) + \frac{1}{4} P(\sigma N > .3) + \frac{1}{4} P(\sigma N > .3) + \frac{1}{4} P(\sigma N > .5)$

at 1.0   at .8   at .2   at 0.0

$\frac{1}{2}\Phi\left(\frac{.5}{.5}\right) + \frac{1}{2}\Phi\left(\frac{.3}{.5}\right)$

$\frac{1}{2} \cdot .1587 + \frac{1}{2} \cdot .2743$

$= .2165$

BER for alternative 1: .2165 ✓

BER for alternative 2: .1517 ✓

Alternative with fewest bit errors: #2 ✓

- not what I expected

lower BER

$\frac{1}{4} P(\sigma N > 1) + \frac{1}{4} P(\sigma N > .7) + \frac{1}{4} P(\sigma N > .2) + \frac{1}{4} P(\sigma N > .5)$

at 1.5   at 1.2   at .3   at 0

$\frac{1}{4}\Phi\left(\frac{1}{.5}\right) + \frac{1}{4}\left(\frac{.7}{.5}\right) + \frac{1}{4}\left(\frac{.2}{.5}\right) + \frac{1}{4}\left(\frac{.5}{.5}\right)$

$\frac{1}{4} \cdot .0228 + \frac{1}{4} \cdot .0808 + \frac{1}{4} \cdot .3446 + \frac{1}{4} \cdot .1587$

$= .1517$

TA?

Problem 2. Error Detection and Correction (20 points) → < 1

Not ≤ 1

/ strictly

(A) (3 points) Suppose p is the probability of a bit error. What is the probability that a codeword of N bits experiences more than 1 bit error?

U(3)
- oh 6.042

$P(\text{no error}) = (1-p)^n$

$P(1 \text{ error}) = p(1-p)^{n-1}$

$P(>1 \text{ error}) = 1 - \text{O}p(1-p)^{n-1} - (1-p)^n$

Give expression

-2

(B) (3 points) Alice wants to transmit 4-bit messages along with any additional parity bits needed to perform single error correction at the receiver. What is the minimum number of bits needed in the transmitted codewords (message bits + parity bits) in order to ensure that single error correction is possible?

↓k

?n

$n \leq 2^{n-k} - 1$

Minimum number of bits in codeword: 7 ✓

Find n via G+V

$n=5$: $5 \leq 2^{5-4} - 1$, $5 \leq 2-1$

$n=6$: $6 \leq 2^2 - 1$

$n=7$: $7 \leq 2^3 - 1$ ✓

Bob is designing a channel coding scheme for a new spacecraft that will be sending back pictures from Io, one of the moons of Jupiter. He has chosen a (9,1,9) replication code, i.e., the two codewords are 000000000 and 111111111.

n k d

(C) (3 points) What is the largest number of bit errors in a single code word that can be corrected using the (9,1,9) code?

?, how find mathmatically?

at 5 would correct other way

Largest number of correctable bit errors: 4 ✓

(D) (3 points) Bob is particularly worried about burst errors, i.e., multiple bit errors that occur in successive bits. Suppose Bob would like correct error bursts of up to 10 bits by interleaving a block of (9,1,9) code words. Given your answer to part C, what is the minimum number of words that can be interleaved to handle burst errors up to 10 bits?

$\left\lceil \frac{10}{4} \right\rceil =$

Minimum size of interleave block: 3 ✓

An Internet Sudoku gaming site transmits messages containing nine data bits and seven parity bits, arranged in a rectangle as follows:

D_{00}	D_{01}	D_{02}	P_{0x}
D_{10}	D_{11}	D_{12}	P_{1x}
D_{20}	D_{21}	D_{22}	P_{2x}
P_{x0}	P_{x1}	P_{x2}	P_{xx}

Each D_{ij} in the above diagram indicates a data bit, equally likely to be a 0 or 1. Each P_{ix} and P_{xj} is an even parity bit chosen to make the total number of 1s in the i^{th} row or j^{th} column, respectively, even. P_{xx} is an even parity bit chosen to make the total number of 1s in the entire transmission even. Thus in an error-free transmission, the total number of 1s in 4-bit columns 0 thru 2 and 4-bit rows 0 thru 2, as well as in the entire 16-bit transmission, is even.

(E) (4 points) Suppose two nine-bit data words have a Hamming distance of 1. What is the Hamming distance between the 16-bit transmissions resulting from these two data words? ✓

So change 1 data bit

Hamming distance between transmissions: 4

- what changes - 3 parity bits + message bit

Each of the following represents a transmission received, with at most a single-bit error. For each message, circle the bit that was changed due to a transmission error, or write NO ERROR if no errors are detected (1 point each).

Circle bit in error or write NO ERROR

Dark circled is primary error source

(F)

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

No

(G)

1	0	1	0
1	1	0	0
0	1	1	0
0	1	0	$\boxed{0}$

Kn must be

(H)

SO

0	1	0	$\boxed{0}$
0	0	1	1
1	1	0	0
$\boxed{0}$	0	1	1

(I)

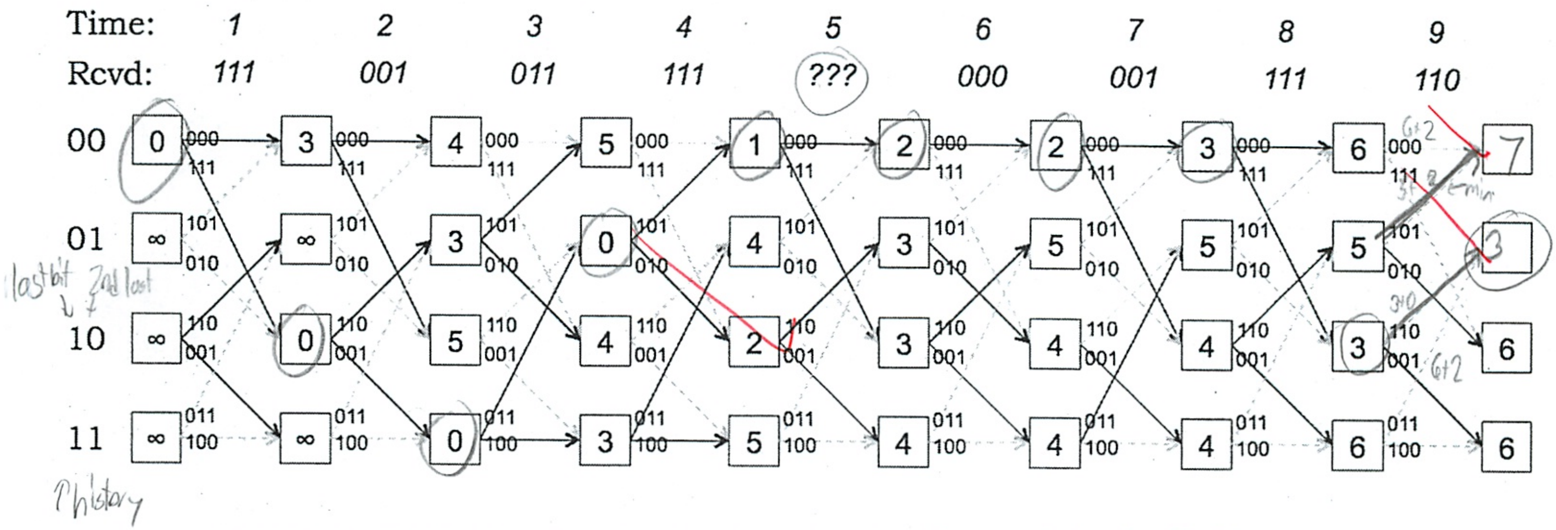
0	1	0	1
1	0	1	0
0	1	1	0
1	0	0	0

-1

No

Problem 3. Convolutional Codes (20 points)

Consider the following trellis showing the operation of the Viterbi algorithm using a hard branch metric at the receiver as it processes a message encoded with a convolutional code *C*. Most of the path metrics have been filled in for each state at each time and the predecessor states determined by the Viterbi algorithm are shown by a solid transition arrow.



(A) (1 point) What is the code rate and constraint length of the convolutional code *C*?

3 bits for each data bit = rate

1/2

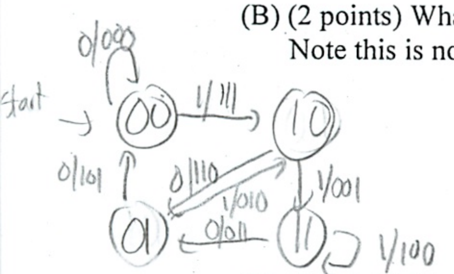
2 pieces of history

Code rate: 1/3 **Constraint length:** 2 x -.5

amt of history

means 3 generators each 2 bits long

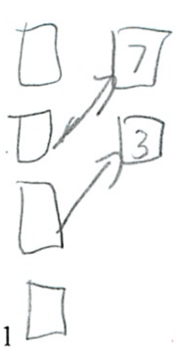
(B) (2 points) What bits would be transmitted if the message "1011" were encoded using *C*? Note this is not the message being decoding in the trellis above.



Transmitted bits: 111, 110, 010, 001

(C) (2 points) Please compute the missing path metrics in the top two boxes of rightmost column and enter their value in the appropriate boxes in the trellis diagram. Remember to draw the solid transition arrow showing the predecessor state for each metric you compute.

Compute missing path metrics, indicate predecessor state



(D) (4 points) The received parity bits for time 5 are missing from the trellis diagram. What values for the parity bits are consistent with the other information in the trellis? Note that there may be more than one set of such values.

Possible values for received parity bits at time 5: 010 or 100

Any thing hamming distance 1 from 000

So can be 001 010 100

but also! d=1 from 110 ← so can't be 001

d=2 from 111 ← does not further restrict

d=2 from 001 ← can't be 001

(E) (7 points) In the trellis diagram on the previous page, circle the states along the most-likely path through the trellis. Determine the decoded message that corresponds to that most-likely path.

States

Circle states on most-likely path

~~00~~ 10 11 01 00 00 00 00 10 01 **Decoded message:** ~~1~~ 1 1 0 0 0 0 0 1 0

take first digit

except first group (start state)

(F) (4 points) Based on your answer to part E, how many bit errors were detected in the received transmission and at what time(s) did those error(s) occur?

Number of bit errors detected: 3

- Whenever count goes up

Time(s) at which bit errors occurred: 4, 5, 7

Problem 4. MAC Protocols (20 points) throughput = utilization?

(A) (2 points each) Which of the following statements are always true for networks with $N > 1$ nodes using correctly implemented versions of unslotted Aloha, slotted Aloha, Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA)? Unless otherwise stated, assume that the slotted and unslotted versions of Aloha are stabilized and use the same stabilization method and parameters.

Circle TRUE or FALSE, then briefly explain or give counter example

TRUE FALSE — There exists some offered load pattern for which TDMA has lower throughput than slotted Aloha.

✓ Explain: When some TDMA nodes are not backlogged - are still assigned time slots where in slotted Aloha they don't transmit - and don't waste anything

Straight off tutor problems

TRUE FALSE — In a slotted Aloha MAC protocol using binary exponential backoff, the probability of transmission will always eventually converge to some value p, and all nodes will eventually transmit with probability p.

✓ Explain: It will not converge - will always bounce around.

?, How to prove

TRUE FALSE — Suppose nodes I, II and III use a fixed probability of $p = 1/3$ when transmitting on a 3-node slotted Aloha network. If all the nodes are backlogged then over time the utilization averages out to $1/e \approx 37\%$.

✗ Explain: $1/N$ leads to $U \approx \frac{1}{e}$ So that works here $N=3$

TRUE FALSE — When the number of nodes, N, is large in a stabilized slotted Aloha network, setting $p_{max} = p_{min} = 1/N$ will achieve the same utilization as a TDMA network if all the nodes are backlogged.

means no backoff adjustment

✓ Explain: So here $U = \frac{1}{e}$ for slotted Aloha where as TDMA: $U = 1$ for all bottlenecked

TRUE FALSE — Using contention windows with a CSMA implementation guarantees that a packet will be transmitted successfully (i.e., without collisions) within some bounded time.

✓ Explain: No guarantee - can still manage to pick the same value as another node from the CW - so collision

Suppose that there are three nodes – A, B and C – seeking access to a shared medium using slotted Aloha, each using some fixed probability of transmission, where each packet takes one slot to transmit. Assume that the nodes are always backlogged, and that node A has half the probability of transmission as the other two, i.e., $p_A = p$ and $p_B = p_C = 2p$.

(B) (4 points) If $p_A = .3$, compute the average utilization of the network. Please show your work.

$P_B = P_C = .6$

Average utilization of network: .364

$U_{slotted} = N p (1-p)^{N-1}$

p for if same

$U = .3 \cdot (1-.6)(1-.6) + (1-.3) .6 (1-.6) + (1-.3)(1-.6) .6$
$= .048 + .168 + .168$
$= .384$

(C) (6 points) What value of p_A maximizes the average utilization of the network and what is the corresponding maximum utilization? Please show your work.

value of p_A that maximizes utilization: .2022

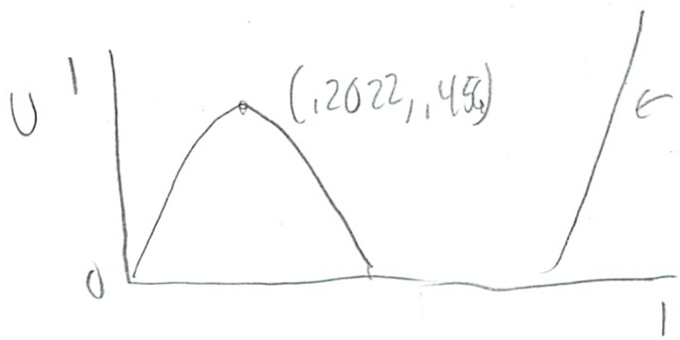
$U = P_a(1-2p_a)(1-2p_a) + (1-p_a)\, 2p_a(1-2p_a) + (1-p_a)(1-2p_a)\boxed{2p_a}$

corresponding maximum utilization: .456

makes sense

maximize, using TI-89

$= p(1-2p)^2 + 2(1-p)2p(1-2p)$



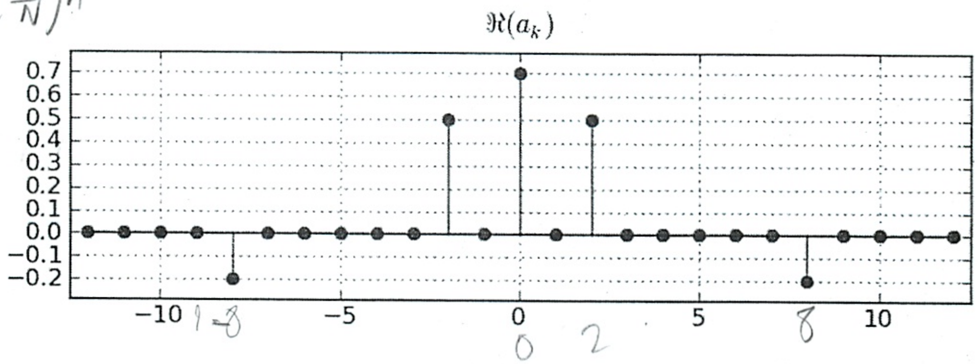
← I'm guessing this is wrong here
- going w/ local maximum

oh boy!

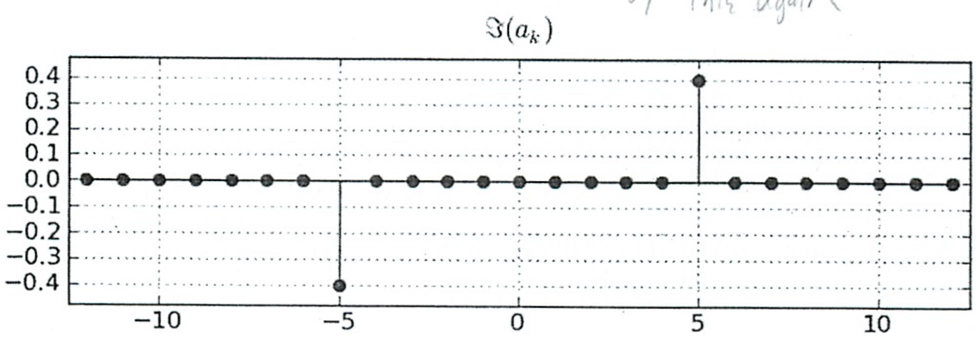
Problem 5. Fourier Series and Frequency Response (20 points)

The spectral coefficients, a_k, for a real-valued periodic signal $x[n]$ are plotted below using $N = 25$. Since the spectral coefficients are complex numbers, the plots show the real part and imaginary part for each index k, $-12 \le k \le 12$. The fundamental frequency of this periodic signal is $2\pi/N = 0.08\pi$ radians/timestep.

$x[n] = \sum_{k \in \langle N \rangle} a_k e^{jk\left(\frac{2\pi}{N}\right)n}$



What was the trick w/ this again?



(A) (5 points) Please give an equation for $x[n]$ in terms of constants and appropriately scaled sines and cosines that are harmonics of the fundamental frequency. Hint: your equation should be the sum of 4 terms, each involving a different frequency. does this mean anything?

Equation for x[n]: $2 \cdot .2\, e^{j\cdot 8 \cdot \frac{2\pi}{25} n} + -.4j\, e^{j \cdot -5 \cdot \frac{2\pi}{25} n} + 2 \cdot .5\, e^{j \cdot 2 \cdot \frac{2\pi}{25} \cdot n} + .7 e^0 + .4j\, e^{j 5 \cdot \frac{2\pi}{25} n}$

$x[0] = -.2 \cdot e^{j 8 \frac{2\pi}{25} \cdot 0} + -.4i \cdot e^0 + .5 + .7 + .5 + -.2 + .4i$

$= 1.3 + 0i$ since $e^0 = 1$

Oh want generic formula - easier!

$x[n] = -.2\, e^{j \cdot -8 \cdot \frac{2\pi}{25} \cdot n} + \text{etc...}$

and then repeats w/ same ± so 2x

Could prob combine one more but then i term would disappear and there would be 3 terms

Bitdiddle?

(B) (5 points) Ben designs a simple LTI system characterized by the following unit sample response:

$h[0] = 1$
$h[1] = -2$
$h[2] = 1$
$h[n] = 0$ otherwise

Please give the equation for the frequency response of the system, $H(e^{j\Omega})$, and evaluate the magnitude of the frequency response at frequencies 0, $\pi/2$ and π. If this system is used as filter, what frequency or frequencies are removed?

Equation for $H(e^{j\Omega})$: $e^{0} - 2e^{-j\Omega} + e^{-j\Omega 2}$ ✓

magnitude at frequency 0 = $|H(e^{j0})|$: 0 ✓

-1 **magnitude at frequency $\pi/2$ = $|H(e^{j\pi/2})|$:** $-2i$ ✗

magnitude at frequency π = $|H(e^{j\pi})|$: 4 ✓

where are the 0s?

frequency or frequencies removed by filter: $\cancel{2\pi}, 0$ ✓

(last pt

$$H(e^{j\Omega}) = e^{0} - 2e^{-j\Omega 1} + 1e^{-j\Omega 2}$$
$$= 1 - 2(\cos(-\Omega) + \sin(-\Omega)i) + \cos(-2\Omega) + \sin(-2\Omega)i$$

Now $\Omega = 0$

$$= 1 - 2(\cos(0) + \sin(0)i) + \cos(0) + \sin(0)i$$
$$= 1 - 2(1 + 0i) + 1 + 0i$$
$$= 1 - 2 + 1 = 0$$

$\Omega = \pi/2$

$$= 1 - 2(\cos(-\tfrac{\pi}{2}) + \sin(-\tfrac{\pi}{2})i) + \cos(-2\tfrac{\pi}{2}) + \sin(-2\tfrac{\pi}{2})i$$
$$= 1 - 2(0 - 1i) + -1 + 0i$$
$$= 0 - 2i$$

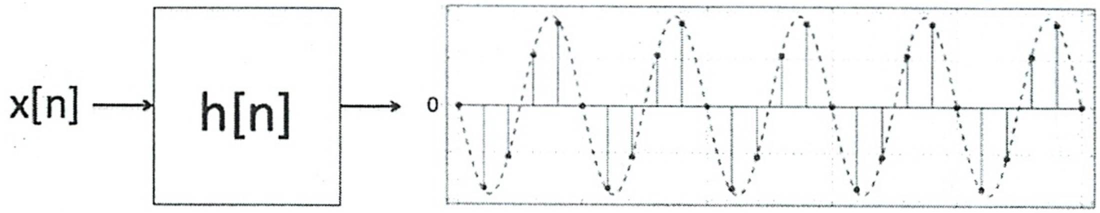
$\Omega = \pi$

$$= 1 - 2(\cos(-\pi) + \sin(-\pi)i + \cos(-2\pi) + \sin(-2\pi)i$$
$$= 1 - 2(-1 + 0i) + 1 + 0i = 4$$

Or $-2 = -2\cos(\varphi)$
Solve
$1 = \cos\varphi$
Where is $\cos = 1$
at $0, 2\pi$
So confirms

The graphs — The imag part

(C) (10 points) Design a LTI system that will filter the x[n] given at the start of this problem, removing all components except the sinusoid of frequency 5(2π/25) = 0.4π radians/sample. Note that the output sinusoid has some phase φ and amplitude A, with a zero average, i.e., it ranges in value between ±A about 0. The A associated with your system can have any non-zero value and φ can have any value between –π and π.



Please give the unit sample response, h[n], of your system. If your system is constructed by connecting simpler LTI systems in series, you can give the h[n] for each of the simpler systems and then give an expression for how their h[n] are combined to form the h[n] for the overall system.

In order to be eligible for partial credit, briefly explain what each component of your system is designed to do.

Give h[n] for system and explain your answer

Should be 0s everywhere else

So combine high and low pass filters

high pass w/ 0 at $f = 8 \cdot \frac{2\pi}{25}$

low pass w/ 0 at $f = 0 \cdot \frac{2\pi}{25}$, $f = 2 \cdot \frac{2\pi}{25}$

Convolve together to get, as a filter

To get 0s

$h[0]=1 \quad h[1] = -2\cos(\Omega) \quad h[2]=1$

$H(e^{j\Omega}) = (e^{j\Omega})^2 - 2\cos(\Omega)(e^{j\Omega}) + 1 = 0$

So for $\Omega \pm 8 \cdot \frac{2\pi}{25}$

-3

END OF QUIZ 2!

$0 = x^2 - 2\cos\left(\frac{16\pi}{25}\right)x + 1$ Solve for x

over

$X = \sqrt{2\cos\left(\frac{16\pi}{25}\right) - 1} = \cos\Omega + \sin\Omega\, i$

Solve for Ω

That is first Ω

Repeat for others

To get high pass from lowpass

high pass = 1 − low pass

?i

-never went over highpass

Then convolve (time) or multiply (freq) together to get what you want

So convolve +1

much easier in time freq

$h[0] = 0 \qquad h[1] = 1 - 2\cos\left(\frac{16\pi}{25}\right)^{+2} \qquad h[2] = 0 \qquad h[\text{otherwise}] = 1$

$h[0] = 1 \qquad h[1] = 2\cos(0)^{+2} \qquad h[2] = 1 \qquad h[\text{other}] = 0$

$h[1] = 2\cos\left(\frac{4\pi}{25}\right)^{+2}$

To get your filter

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.02 Introduction to EECS II
Spring 2011

Quiz 2

Name SOLUTIONS	*Score*

- ☐ 10a Devavrat Shah 24-402
- ☐ 11a Devavrat Shah 24-402
- ☐ 12n Fabian Lim 38-166
- ☐ 1p John Sun 38-166
- ☐ 2p John Sun 38-166

Avg = 82.1
σ = 14.2

Please write your answers legibly in the spaces provided. You can use the backs of the pages if you need extra room for your answer or scratch work. Make sure we can find your answer!

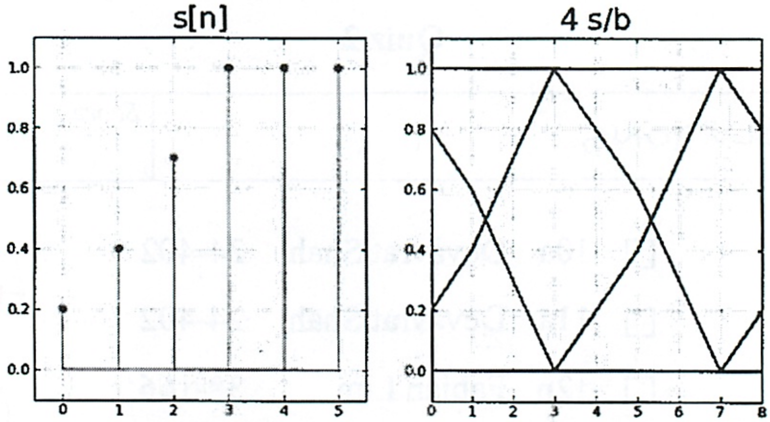
You can use a calculator and one 8.5" x 11" cribsheet.

Partial credit will only be given in cases where you show your work and (very briefly) explain your approach.

Prob. #1 (20 pts)	Prob. #2 (20 pts)	Prob. #3 (20 pts)	Prob. #4 (20 pts)	Prob. #5 (20 pts)

Problem 1. Gaussian Noise and Bit Error Rates (20 points)

Consider the figure below, which shows the step response for a particular transmission channel along with the eye diagram for channel response when transmitting 4 samples/bit.



x	Φ(x)
-3.4	0.0003
-3.3	0.0005
-3.2	0.0007
-3.1	0.0010
-3.0	0.0013
-2.9	0.0019
-2.8	0.0026
-2.7	0.0035
-2.6	0.0047
-2.5	0.0062
-2.4	0.0082
-2.3	0.0107
-2.2	0.0139
-2.1	0.0179
-2.0	0.0228
-1.9	0.0287
-1.8	0.0359
-1.7	0.0446
-1.6	0.0548
-1.5	0.0668
-1.4	0.0808
-1.3	0.0968
-1.2	0.1151
-1.1	0.1357
-1.0	0.1587
-0.9	0.1841
-0.8	0.2119
-0.7	0.2420
-0.6	0.2743
-0.5	0.3085
-0.4	0.3446
-0.3	0.3821
-0.2	0.4207
-0.1	0.4602
-0.0	0.5000

Suppose there is additive Gaussian noise on this channel that sometimes causes a transmitted bit to be misidentified at the receiver. In answering the questions below, please assume that

- the receiver uses the optimal detection sample for each bit (corresponding to the "center" of the eye)
- the receiver uses a detection threshold of 0.5V
- 0 and 1 bits are transmitted with probability of 0.5
- the additive noise is independent of the bit being transmitted and has a Gaussian distribution with zero mean and standard deviation σ

(A) (8 points) When sending 4 samples/bit, if the bit error rate is measured to be .0062, what is the approximate value for σ, the standard deviation for the additive Gaussian noise? The table to the right shows Φ(x), the cumulative distribution function for the normal (Gaussian) distribution. Please show your work.

Value for σ: 0.2

$$BER = .0062 = \frac{1}{2}\Phi\left(\frac{.5-1}{\sigma}\right) + \frac{1}{2}\left(1-\Phi\left(\frac{.5-0}{\sigma}\right)\right)$$

$$= \frac{1}{2}\Phi\left(\frac{-.5}{\sigma}\right) + \frac{1}{2}\Phi\left(\frac{-.5}{\sigma}\right)$$

$$= \Phi\left(\frac{-.5}{\sigma}\right)$$

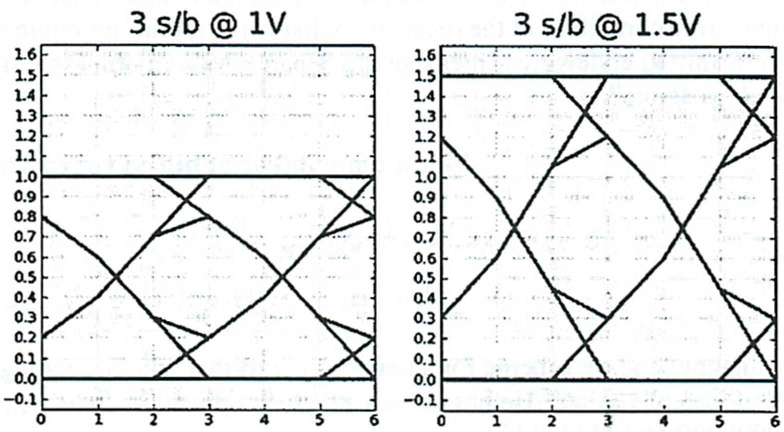
$$\Rightarrow \frac{-.5}{\sigma} = -2.5 \text{ so } \sigma = \frac{-.5}{-2.5} = .2$$

When transmitting at 3 bits/sample down the same channel, the designers are trying to decide between two alternatives for the transmitter design:

1. the transmitter sends 0V for a 0 bit and 1V for a 1 bit, resulting in the eye diagram shown on the left in the figure below. The possible sample voltages at the center of the eye are 0V, 0.2V, 0.8V and 1.0V.

2. the transmitter sends 0V for a 0 bit and 1.5V for a 1 bit, resulting in the eye diagram shown on the right in the figure below. The possible sample voltages at the center of the eye are 0V, 0.3V, 1.2V and 1.5V. The designers are hypothesizing that the increased transmission range will lower the bit error rate.

In either alternative *__the receiver threshold is unchanged at 0.5V.__*



(B) (12 points) Assume that the Gaussian noise has $\sigma = 0.5$. For each alternative compute an estimate for the bit error rate using all four possible sample voltages. Indicate which alternative results in the fewest bit errors at the receiver. Please show your work.

BER for alternative 1: 0.2165

BER for alternative 2: 0.1517

Alternative with fewest bit errors: ALT #2

ALT #1: $BER = \frac{1}{4}\left[\Phi\left(\frac{.5-1}{.5}\right) + \Phi\left(\frac{.5-.8}{.5}\right) + \left(1-\Phi\left(\frac{.5-.2}{.5}\right)\right) + \left(1-\Phi\left(\frac{.5-0}{.5}\right)\right)\right]$

$= \frac{1}{4}\left[\Phi(-1) + \Phi(-.6) + \Phi(-.6) + \Phi(-1)\right]$

$= \frac{1}{4}\left[.1587 + .2743 + .2743 + .1587\right] = .2165$

ALT #2: $BER = \frac{1}{4}\left[\Phi\left(\frac{.5-1.5}{.5}\right) + \Phi\left(\frac{.5-1.2}{.5}\right) + \left(1-\Phi\left(\frac{.5-.3}{.5}\right)\right) + \left(1-\Phi\left(\frac{.5-0}{.5}\right)\right)\right]$

$= \frac{1}{4}\left[\Phi(-2) + \Phi(-1.4) + \Phi(-.4) + \Phi(-1)\right] = .1517$

.0228 .0808 .3446 .1587

Problem 2. Error Detection and Correction (20 points)

(A) (3 points) Suppose p is the probability of a bit error. What is the probability that a codeword of N bits experiences more than 1 bit error?

Give expression

$$p(>1 \text{ error}) = 1 - p(\text{no error}) - p(\text{exactly 1 error})$$

$$= 1-(1-p)^N - Np(1-p)^{N-1}$$

(B) (3 points) Alice wants to transmit 4-bit messages along with any additional parity bits needed to perform single error correction at the receiver. What is the minimum number of bits needed in the transmitted codewords (message bits + parity bits) in order to ensure that single error correction is possible?

Minimum number of bits in codeword: 7

$n \le 2^{n-4} - 1 \Rightarrow$ smallest n is 7.

i.e. the Hamming (7,4,3) code.

Bob is designing a channel coding scheme for a new spacecraft that will be sending back pictures from Io, one of the moons of Jupiter. He has chosen a (9,1,9) replication code, i.e., the two codewords are 000000000 and 111111111.

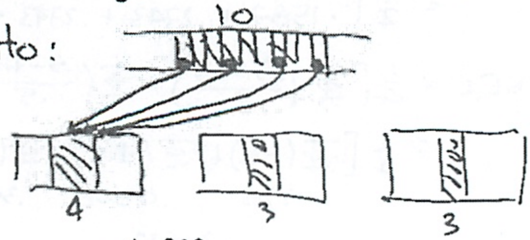
(C) (3 points) What is the largest number of bit errors in a single code word that can be corrected using the (9,1,9) code?

corrected $= \left\lfloor \frac{HD-1}{2} \right\rfloor = \left\lfloor \frac{9-1}{2} \right\rfloor$ **Largest number of correctable bit errors:** 4

(D) (3 points) Bob is particularly worried about burst errors, i.e., multiple bit errors that occur in successive bits. Suppose Bob would like correct error bursts of up to 10 bits by interleaving a block of (9,1,9) code words. Given your answer to part C, what is the minimum number of words that can be interleaved to handle burst errors up to 10 bits?

Minimum value for I: 3

with 3-way interleaving, a 10-bit burst error will be deinterleaved into:



An Internet Sudoku gaming site transmits messages containing nine data bits and seven parity bits, arranged in a rectangle as follows:

D_{00}	D_{01}	D_{02}	P_{0x}
D_{10}	D_{11}	D_{12}	P_{1x}
D_{20}	D_{21}	D_{22}	P_{2x}
P_{x0}	P_{x1}	P_{x2}	P_{xx}

Each D_{ij} in the above diagram indicates a data bit, equally likely to be a 0 or 1. Each P_{ix} and P_{xj} is an even parity bit chosen to make the total number of 1s in the i^{th} row or j^{th} column, respectively, even. P_{xx} is an even parity bit chosen to make the total number of 1s in the entire transmission even. Thus in an error-free transmission, the total number of 1s in 4-bit columns 0 thru 2 and 4-bit rows 0 thru 2, as well as in the entire 16-bit transmission, is even.

(E) (4 points) Suppose two nine-bit data words have a Hamming distance of 1. What is the Hamming distance between the 16-bit transmissions resulting from these two data words?

changing D → change row & col P bits

Hamming distance between transmissions: 4

∴ 3 bits have changed → change P_{xx}

Each of the following represents a transmission received, with at most a single-bit error. For each message, circle the bit that was changed due to a transmission error, or write NO ERROR if no errors are detected (1 point each).

Circle bit in error or write NO ERROR

(F)

NO ERROR

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

(G)

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

(H)

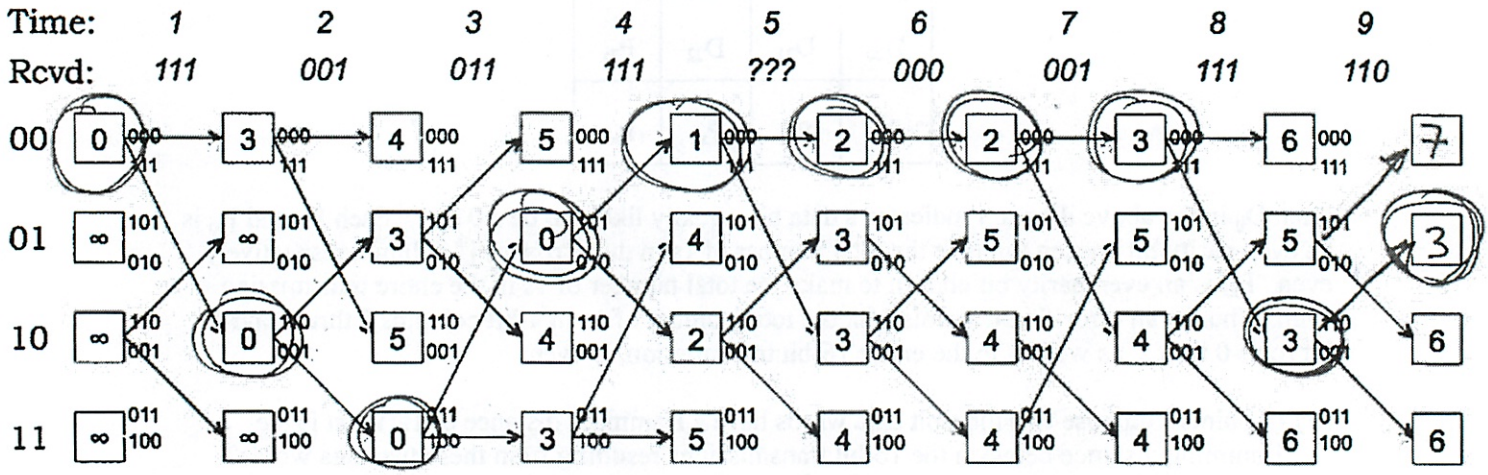
0	1	0	0
0	0	1	1
1	1	0	0
0	0	1	1

(I)

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

Problem 3. Convolutional Codes (20 points)

Consider the following trellis showing the operation of the Viterbi algorithm using a hard branch metric at the receiver as it processes a message encoded with a convolutional code *C*. Most of the path metrics have been filled in for each state at each time and the predecessor states determined by the Viterbi algorithm are shown by a solid transition arrow.



(A) (1 point) What is the code rate and constraint length of the convolutional code *C*?

3 parity bits sent for each message bit.

Code rate: 1/3 **Constraint length:** 3

(B) (2 points) What bits would be transmitted if the message "1011" were encoded using *C*? Note this is not the message being decoding in the trellis above.

Transmitted bits: 111 110 010 001

1 0 1 1

(C) (2 points) Please compute the missing path metrics in the top two boxes of rightmost column and enter their value in the appropriate boxes in the trellis diagram. Remember to draw the solid transition arrow showing the predecessor state for each metric you compute.

Compute missing path metrics, indicate predecessor state

for state 00: PM = min(6+2 (from 00), 5+2 (01)) = 7

for state 01: PM = min(3+0 (from 10), 6+2 (11)) = 3

(D) (4 points) The received parity bits for time 5 are missing from the trellis diagram. What values for the parity bits are consistent with the other information in the trellis? Note that there may be more than one set of such values.

Possible values for received parity bits at time 5: 010, 100

using PMs!

(A) HD(???,000) = 1

(B) HD(???,111) = 2

(C) HD(???,110) = 1

(D) HD(???,001) = 2

???	
000	x violates (A)
001	x violates (C)
010	✓
011	x violates (A)
100	✓
101	x violates (A)
110	x violates (A)
111	x violates (A)

(E) (7 points) In the trellis diagram on the previous page, circle the states along the most-likely path through the trellis. Determine the decoded message that corresponds to that most-likely path.

Circle states on most-likely path

Decoded message: 110 000 010

(6.02 in octal!)

(F) (4 points) Based on your answer to part E, how many bit errors were detected in the received transmission and at what time(s) did those error(s) occur?

PM of final state ⇒ **Number of bit errors detected:** 3

Time(s) at which bit errors occurred: 4, 5, 7

where PM increments along most-likely path

Problem 4. MAC Protocols (20 points)

(A) (2 points each) Which of the following statements are always true for networks with $N > 1$ nodes using correctly implemented versions of unslotted Aloha, slotted Aloha, Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA)? Unless otherwise stated, assume that the slotted and unslotted versions of Aloha are stabilized and use the same stabilization method and parameters.

Circle TRUE or FALSE, then briefly explain or give counter example

TRUE FALSE — There exists some offered load pattern for which TDMA has lower throughput than slotted Aloha.

Explain: suppose 1 node is backlogged, others aren't.
TDMA throughput = $\frac{1}{N}$, slotted Aloha thruput $\approx$ 1.

TRUE FALSE — In a slotted Aloha MAC protocol using binary exponential backoff, the probability of transmission will always eventually converge to some value p, and all nodes will eventually transmit with probability p.

Explain: in a stabilized Aloha protocol, p is always changing with successes & failures. ⇒ does not converge to a particular value

TRUE FALSE — Suppose nodes I, II and III use a fixed probability of $p = 1/3$ when transmitting on a 3-node slotted Aloha network. If all the nodes are backlogged then over time the utilization averages out to $1/e \approx 37\%$.

Explain: $U = 3(\frac{1}{3})(1-\frac{1}{3})^2 = \frac{4}{9} \neq \frac{1}{e}$

TRUE FALSE — When the number of nodes, N, is large in a stabilized slotted Aloha network, setting $p_{max} = p_{min} = 1/N$ will achieve the same utilization as a TDMA network if all the nodes are backlogged.

Explain: TDMA can achieve full utilization but Aloha achieves $\sim \frac{1}{e}$

TRUE FALSE — Using contention windows with a CSMA implementation guarantees that a packet will be transmitted successfully (i.e., without collisions) within some bounded time.

Explain: contention windows guarantee a transmission attempt within bounded time, but there's no guarantee of success

Suppose that there are three nodes – A, B and C – seeking access to a shared medium using slotted Aloha, each using some fixed probability of transmission, where each packet takes one slot to transmit. Assume that the nodes are always backlogged, and that node A has half the probability of transmission as the other two, i.e., $p_A = p$ and $p_B = p_C = 2p$.

(B) (4 points) If $p_A = .3$, compute the average utilization of the network. Please show your work.

Average utilization of network: 0.384

$$U = (p)(1-2p)^2 + 2(2p)(1-p)(1-2p)$$

$$= (.3)(.4)^2 + 2(.6)(.7)(.4)$$

$$= 0.048 + .336 = 0.384$$

(C) (6 points) What value of p_A maximizes the average utilization of the network and what is the corresponding maximum utilization? Please show your work.

value of p_A that maximizes utilization: 0.202

corresponding maximum utilization: 0.456

Note: $p < .5$ since p_B & p_C must be < 1.

$$U = p(1-2p)^2 + 2(2p)(1-p)(1-2p)$$

$$= p(1-4p+4p^2) + 4p(1-3p+2p^2)$$

$$= p - 4p^2 + 4p^3 + 4p - 12p^2 + 8p^3$$

$$= 5p - 16p^2 + 12p^3$$

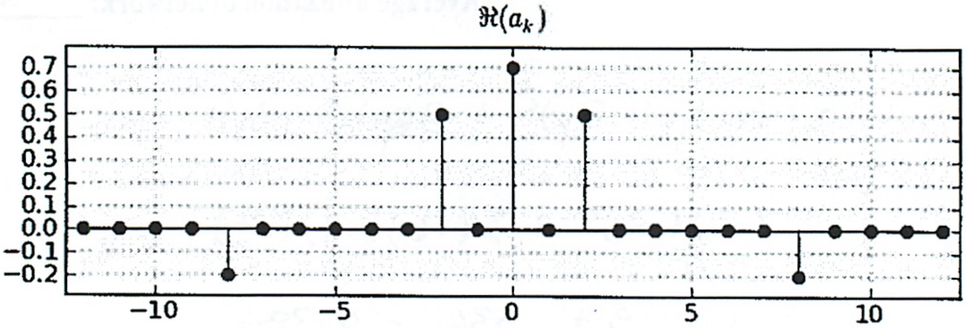
max when $\frac{dU}{dp} = 0 = 5 - 32p + 36p^2$

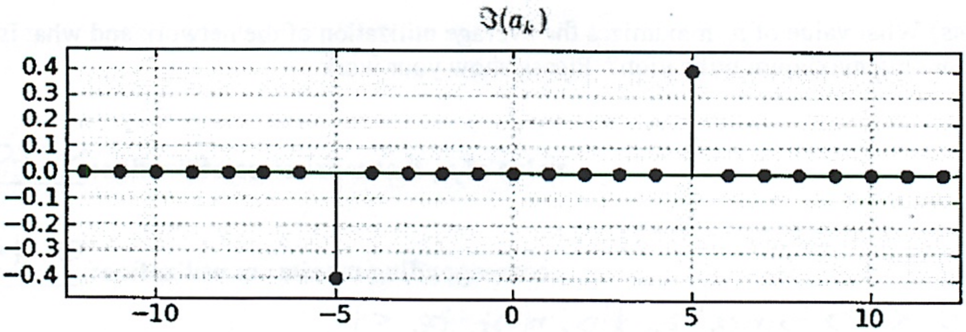
roots @ $\frac{32 \pm \sqrt{1024-720}}{72} = .687, .202$ $< .5$

$\Rightarrow U_{max} = 5(.202) - 16(.202)^2 + 12(.202)^3 = .456$

Problem 5. Fourier Series and Frequency Response (20 points)

The spectral coefficients, a_k, for a real-valued periodic signal $x[n]$ are plotted below using $N = 25$. Since the spectral coefficients are complex numbers, the plots show the real part and imaginary part for each index k, $-12 \le k \le 12$. The fundamental frequency of this periodic signal is $2\pi/N = 0.08\pi$ radians/timestep.





(A) (5 points) Please give an equation for $x[n]$ in terms of constants and appropriately scaled sines and cosines that are harmonics of the fundamental frequency. Hint: your equation should be the sum of 4 terms, each involving a different frequency.

Equation for x[n]: $x[n] = .7 + \cos(0.16\pi n) - .8\sin(0.4\pi n) - .4\cos(.64\pi$

- $a_0 = .7 \Rightarrow$ constant term $= .7$
- $a_2 = a_{-2} = 0.5 \Rightarrow$ term of $\cos(2 \cdot \frac{2\pi}{25} n)$
- $a_5 = .4j \quad a_{-5} = -.4j \Rightarrow$ term of. $-.8\sin(5 \cdot \frac{2\pi}{25} n)$
- $a_8 = a_{-8} = -.2 \Rightarrow$ term of $-.4\cos(8 \cdot \frac{2\pi}{25} n)$

(B) (5 points) Ben designs a simple LTI system characterized by the following unit sample response:

h[0] = 1
h[1] = -2
h[2] = 1
h[n] = 0 otherwise

Please give the equation for the frequency response of the system, $H(e^{j\Omega})$, and evaluate the magnitude of the frequency response at frequencies 0, $\pi/2$ and π. If this system is used as filter, what frequency or frequencies are removed?

Equation for $H(e^{j\Omega})$: $1 - 2e^{-j\Omega} + e^{-j2\Omega}$

magnitude at frequency 0 = $|H(e^{j0})|$: 0

magnitude at frequency $\pi/2$ = $|H(e^{j\pi/2})|$: 2

magnitude at frequency π = $|H(e^{j\pi})|$: 4

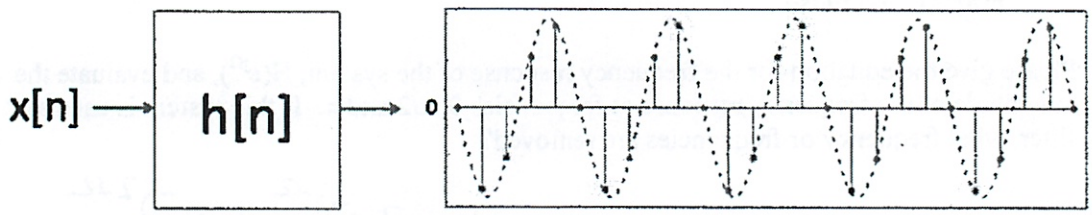
frequency or frequencies removed by filter: $\Omega = 0$

$$H(e^{j0}) = 1 - 2 + 1 = 0$$

$$H(e^{j\pi/2}) = 1 - 2(\cos\tfrac{-\pi}{2} + j\sin\tfrac{-\pi}{2}) + (\cos -\pi + j\sin -\pi)$$
$$= 1 - 2(0 + j(-1)) + (-1 + j\cdot 0)$$
$$= 1 - 2j - 1 = -2j \qquad |H(e^{j\pi/2})| = |-2j| = 2$$

$$H(e^{j\pi}) = 1 + 2 + 1 = 4$$

(C) (10 points) Design a LTI system that will filter the x[n] given at the start of this problem, removing all components except the sinusoid of frequency $5(2\pi/25) = 0.4\pi$ radians/sample. Note that the output sinusoid has some phase φ and amplitude A, with a zero average, i.e., it ranges in value between ±A about 0. The A associated with your system can have any non-zero value and φ can have any value between $-\pi$ and π.



Please give the unit sample response, h[n], of your system. If your system is constructed by connecting simpler LTI systems in series, you can give the h[n] for each of the simpler systems and then give an expression for how their h[n] are combined to form the h[n] for the overall system.

In order to be eligible for partial credit, briefly explain what each component of your system is designed to do.

Give h[n] for system and explain your answer

STRAIGHT FORWARD:

h_1: remove $k=0 \Rightarrow h_1 = [1,-2,1]$ (from part B)

h_2: remove $k=\pm 2 \Rightarrow h_2 = [1, -2\cos(.16\pi), 1] = [1, -1.753, 1]$

h_3: remove $k=\pm 8 \Rightarrow h_3 = [1, -2\cos(.64\pi), 1] = [1, +.852, 1]$

$x[n] \rightarrow \boxed{h_1} \rightarrow \boxed{h_2} \rightarrow \boxed{h_3} \rightarrow$ only $k=\pm 5$ remains

$h[n] = h_1 * h_2 * h_3 = [1, -2.9, 3.31, -2.82, 3.31, -2.9, 1]$

CLEVER:

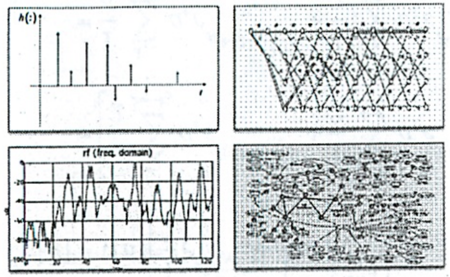
WANT: $x[n] \rightarrow \boxed{H(e^{j\Omega})} \rightarrow$

but this is 'just a cosine!

$h[n] = 2\cos(5 \frac{2\pi}{25} n)$

END OF QUIZ 2!

Take signals + move around in spectrum



INTRODUCTION TO EECS II
DIGITAL COMMUNICATION SYSTEMS

6.02 Spring 2011
Lecture #16

Quiz 2
4/12
26-100
- see website and old tutorial problems

- sharing the frequency spectrum
- modulation
- demodulation

freq-division multiplexing

f_s, frequency, Ω and k

Various frequency specifications we'll use
- f_s, the sample frequency in samples/sec – radio tells you 102.1 FM
- f, the signal frequency in Hz = cycles/sec
  - $-f_s/2 \le f \le f_s/2$ limit
- Ω, the angular frequency in radians/sample
  - $-\pi \le \Omega \le \pi$ convert sample # to radians
- k, the spectral coefficient index
  - $-N/2 \le k \le N/2$

7π was ⊖

$$\boxed{\Omega = 2\pi\frac{f}{f_s} = 2\pi\frac{k_\Omega}{N}}$$

$\frac{3\pi}{2} = -\frac{\pi}{2}$

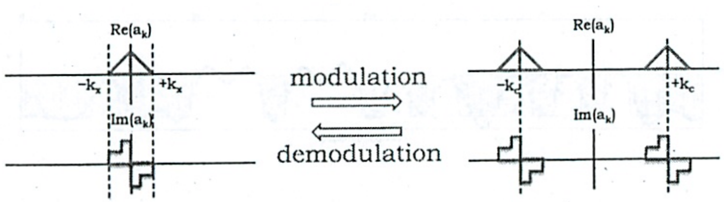
Examples: f_s=1e6 samples/sec, f = 10 kHz, N = 1000
so Ω = .02π and k_Ω = 10

k = 15, N = 100, f_s = 1e6
so Ω = .3π and f = 150 kHz

f, fs don't need to be multiples

Using Some Piece of the Spectrum

- You have: a band-limited signal x[n] at *baseband* (i.e., centered around 0 frequency).
- You want: the same signal, but centered around some specific frequency $k_c(2\pi/N)$.
- Modulation: convert from baseband up to $k_c(2\pi/N)$
  Demodulation: convert from $k_c(2\pi/N)$ down to baseband



Signal centered at 0   Signal centered at k_c

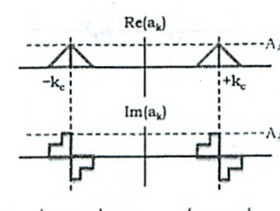
Modulation

x[n] → ⊗ → y[n]
↑
$\cos[k_c(2\pi/N)n]$

For band-limited signal a_k are nonzero only for small range of ±k

$$y[n] = \left[\sum_{k=-k_x}^{k_x} a_k e^{jk\frac{2\pi}{N}n}\right]\left[\frac{1}{2}e^{jk_c\frac{2\pi}{N}n} + \frac{1}{2}e^{-jk_c\frac{2\pi}{N}n}\right]$$

$$= \frac{1}{2}\sum_{k=-k_x}^{k_x} a_k e^{j(k+k_c)\frac{2\pi}{N}n} + \frac{1}{2}\sum_{k=-k_x}^{k_x} a_k e^{j(k-k_c)\frac{2\pi}{N}n}$$



$y[n] = x[n] \cdot \cos\left(k_c \cdot \frac{2\pi}{N} n\right)$

$= \sum_{k=-k_x}$ express as $a_k\, e^{j\frac{k2\pi n}{N}}$

→ flip

$$= \left[\sum_{k=-k_x}^{k_x} a_k e^{j\frac{k2\pi}{N}n}\right]\left[\frac{1}{2}e^{jk_c\frac{2\pi}{N}n} + \frac{1}{2}e^{-jk_c\frac{2\pi}{N}n}\right]$$

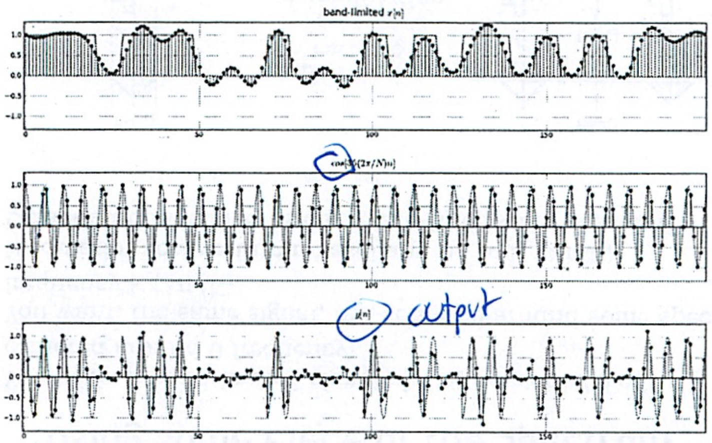
Continued sep sheet

Example: Modulation (time)

baseband

modulating signal

passband



= original

35 cycles of wave points in N

output

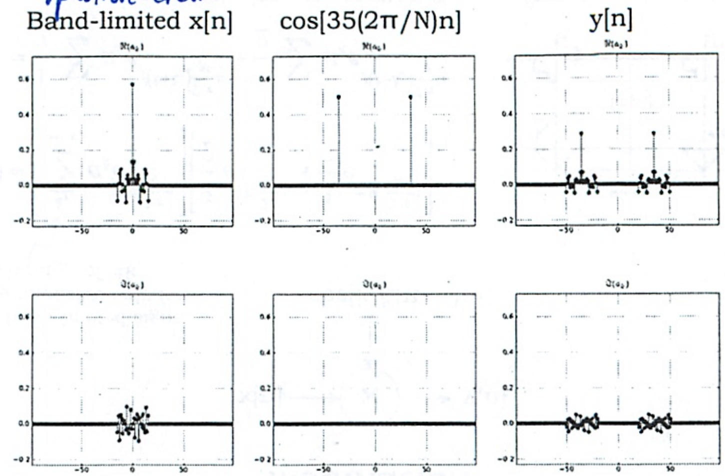
everything in = 1

get a burst out of carrier freq

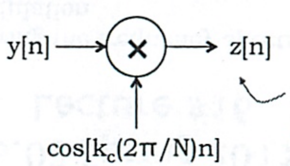
turn waveform into spectral coeffs.

Example: Modulation (freq)

Spectral coeffs



Demodulation



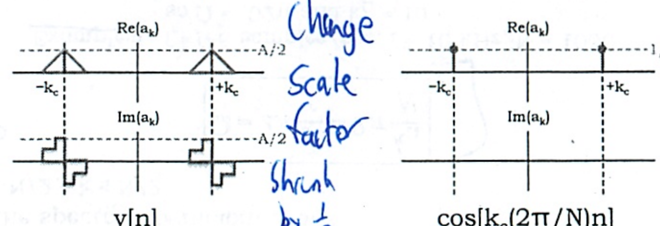
Hmm. So z[n] has what we want at baseband, but has signal we don't want at $\pm 2f_c$

$$z[n] = y[n]\left[\frac{1}{2}e^{jk_c\frac{2\pi}{N}n} + \frac{1}{2}e^{-jk_c\frac{2\pi}{N}n}\right]$$

$$= \left[\frac{1}{2}\sum_{k=-k_x}^{k_x} a_k e^{j(k+k_c)\frac{2\pi}{N}n} + \frac{1}{2}\sum_{k=-k_x}^{k_x} a_k e^{j(k-k_c)\frac{2\pi}{N}n}\right]\left[\frac{1}{2}e^{jk_c\frac{2\pi}{N}n} + \frac{1}{2}e^{-jk_c\frac{2\pi}{N}n}\right]$$

$$= \frac{1}{4}\sum_{k=-k_x}^{k_x} a_k e^{j(k+2k_c)\frac{2\pi}{N}n} + \underbrace{\frac{1}{2}\sum_{k=-k_x}^{k_x} a_k e^{jk\frac{2\pi}{N}n}}_{\text{What we want}} + \frac{1}{4}\sum_{k=-k_x}^{k_x} a_k e^{j(k-2k_c)\frac{2\pi}{N}n}$$

Demodulation Frequency Diagram

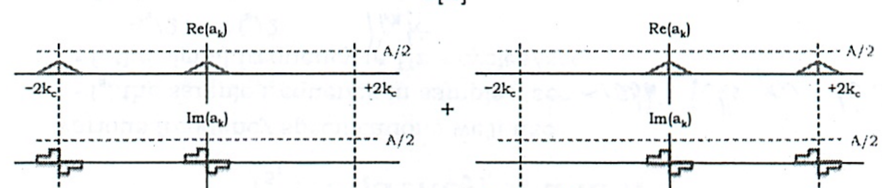


Change scale factor

Shrink by $\frac{1}{2}$

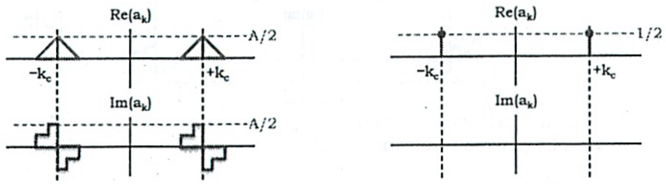
z[n]

Center around kc



Add together for final spectrum

Demodulation Frequency Diagram

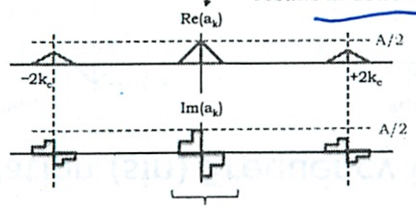


y[n] $\cos[k_c(2\pi/N)n]$

Note combining of signals around 0 results in doubling of amplitude

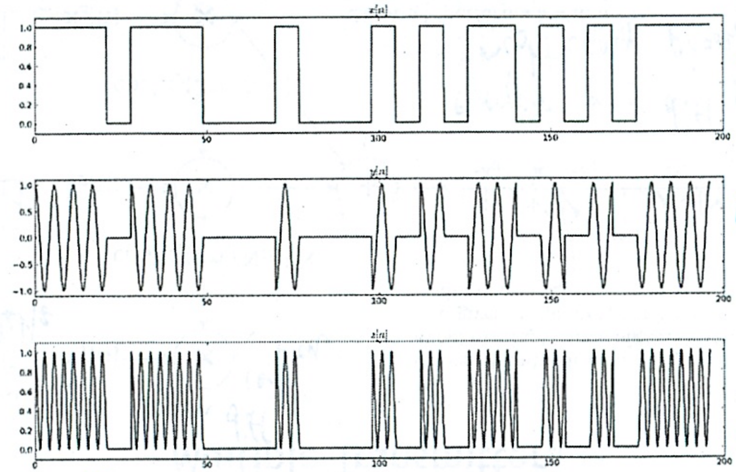
back to original height

z[n]



What we want

Example: Demodulation (time)



original

modulated up to 35

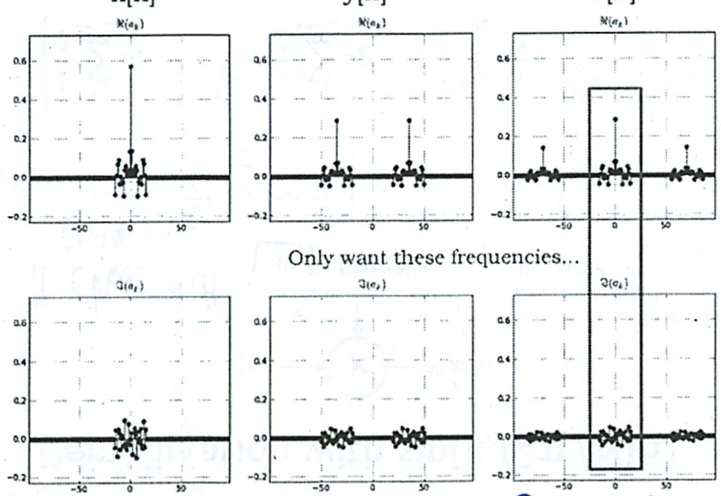
demod using 35

Showing idealized signals

? filled in waveform!

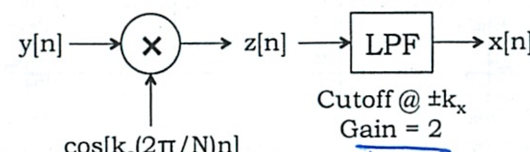
Example: Demodulation (freq)

Spectrum involved



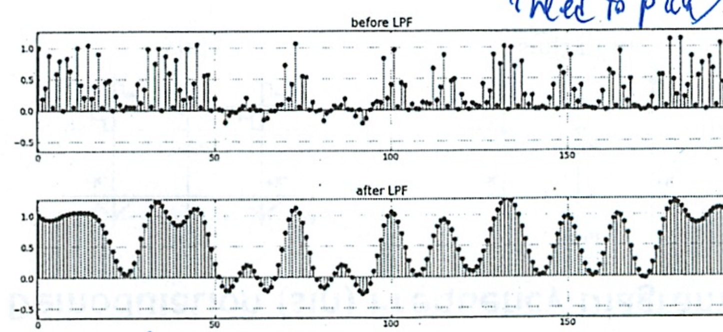
? get rid of this — w/ low pass filter

Demodulation + LPF



Cutoff @ $\pm k_x$
Gain = 2

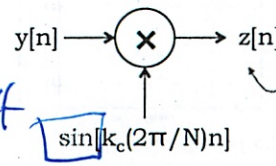
Need to pick in both low freq want and high freq



pick freq want to "tune in"
demodulate to base band - filter to only signal you want

decoded w/ same phase

Demodulation with sin[$k_c(2\pi/N)n$]



Hmm. So z[n] no longer has the signal we want at baseband!

what if chose diff phase?

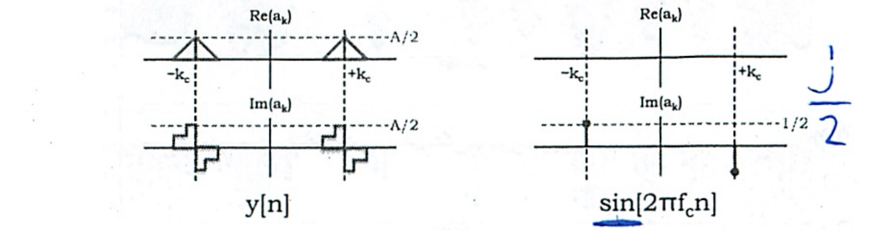
$$z[n]=y[n]\left[-\frac{j}{2}e^{jk_c\frac{2\pi}{N}n}+\frac{j}{2}e^{-jk_c\frac{2\pi}{N}n}\right]$$

$$=\left[\frac{1}{2}\sum_{k=-k_x}^{k_x}a_k e^{j(k+k_c)\frac{2\pi}{N}n}+\frac{1}{2}\sum_{k=-k_x}^{k_x}a_k e^{j(k-k_c)\frac{2\pi}{N}n}\right]\left[-\frac{j}{2}e^{jk_c\frac{2\pi}{N}n}+\frac{j}{2}e^{-jk_c\frac{2\pi}{N}n}\right]$$

$$=-\frac{j}{4}\sum_{k=-k_x}^{k_x}a_k e^{j(k+2k_c)\frac{2\pi}{N}n}+\frac{j}{4}\sum_{k=-k_x}^{k_x}a_k e^{j(k-2k_c)\frac{2\pi}{N}n}$$

Oops, no baseband signal!

Demodulation (sin) Frequency Diagram

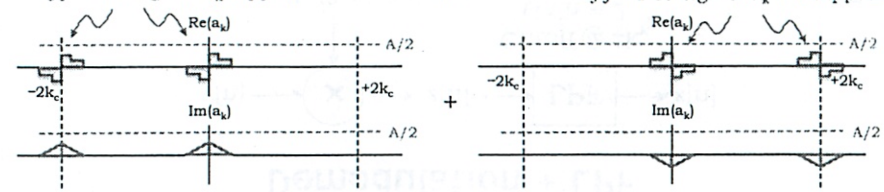


j/2

z[n]

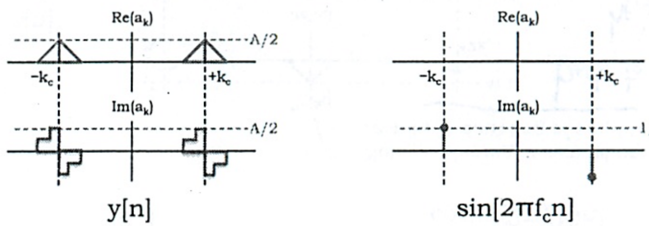
j·j=−1 so sign of a_k flipped

−j·j=+1 so sign of a_k *not* flipped



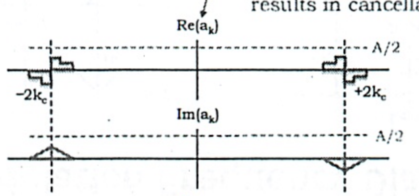
All real → img
img → real flipped (−1)

Demodulation (sin) Frequency Diagram



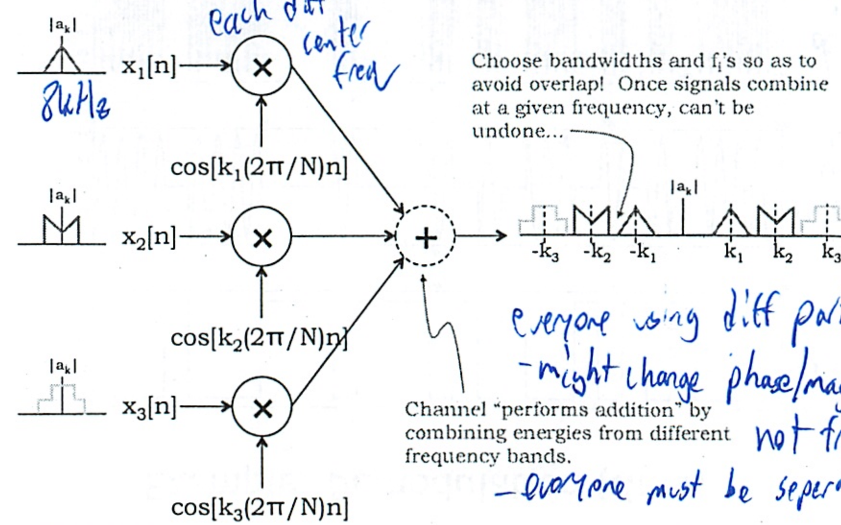
Note combining of signals around 0 results in cancellation!

z[n]



signal cancles!

Multiple Transmitters



8kHz

each diff center freq

Choose bandwidths and f_i's so as to avoid overlap! Once signals combine at a given frequency, can't be undone...

Channel "performs addition" by combining energies from different frequency bands.

everyone using diff part of sys
– might change phase/mag – but not freq
– everyone must be seperate

↓ solved wed

Everyone would need to agree when t=0! But this is hard to be in phase

Lecture 16 Addendum

$$= \frac{1}{2}\sum_k a_k e^{j(k+k_c)\frac{2\pi}{N}n} + \frac{1}{2}\sum_k a_k e^{j(k-k_c)\frac{2\pi}{N}n}$$

Look at some examples

 is original

---

Demo

Peaks are diff radio stations

"Tune" to one

- pull out base band signal

Need to be exactly in center

Sounds weird if b/w stations

All of this is happening in real time

6.02 Recitation

4/5

Quiz - 1 week from today
Lectures 6-15
& last wed

Modulation + Filter Design

Last time LTI channel
- in terms $h[\,]$ - time domain

$$x[\,] \rightarrow h[\,] \Rightarrow y[\,]$$

h convolved w/ x

but we are writing
- for each freq

$$x[\,] = \sum a_k e^{j\Omega_k}$$

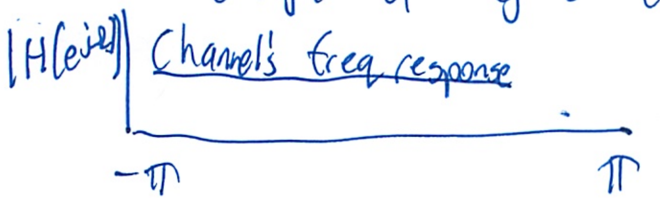
$$\downarrow$$

$$y[\,] = \sum_k a_k H(e^{j\Omega_k}) \cdot e^{j\Omega_k}$$

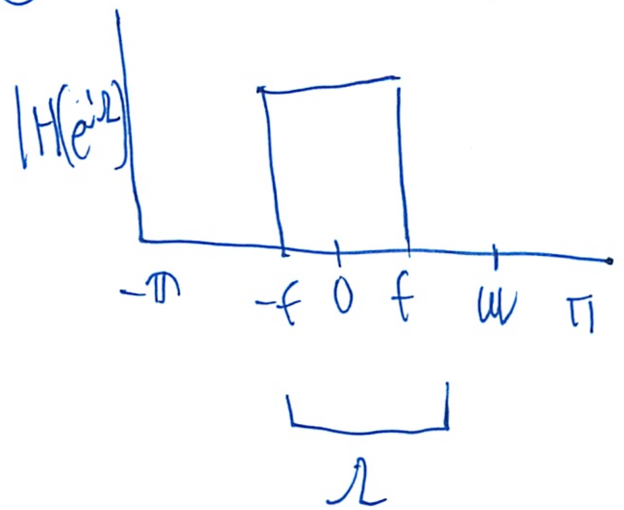
channel freq response

$$\sum_{m=-\infty}^{\infty} h(m)\, e^{-j\Omega m}$$

Can we design a good channel? Filter design



②



$$x = \sum_{\Omega \in [-f, f]} a_\Omega e^{j\Omega n}$$

But some noise gets added
- high freq

$$z(n) = x(n) + N(n)$$

what receive   transmitted   noise $b_\omega \cdot e^{j\omega n}$

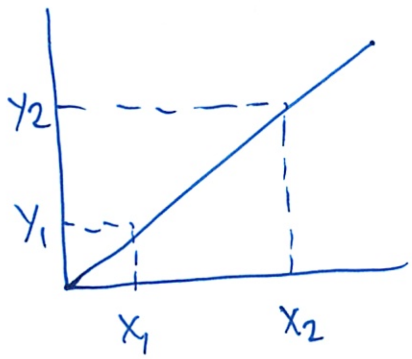
$$= \sum_{\Omega \in [-f, f]} a_\Omega e^{j\Omega n} + b_\omega e^{j\omega n}$$

throw this out w/ a filter

Can design a filter as an LTI channel
- Choose $h()$ so that its freq response has a certain behavior

(3)

In HW5, we did curve fitting of a line



$y = Ax + B$

$y_1 = Ax_1 + B$

$y_2 = Ax_2 + B$) Solve A, B

Will do pattern matching for h( )

$$H(e^{j\Omega_k}) = \sum_{m=-\infty}^{\infty} h(m)\, e^{-j\Omega_k m}$$

$$H(e^{j\Omega}) = \sum h(m)(e^{-j\Omega})^m$$

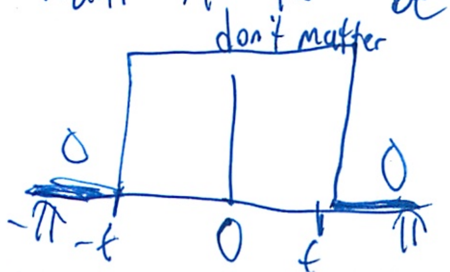
$$H(x) = \sum h(m)\, X^n$$

$\llcorner\, x = e^{-j\Omega}$) its a polynomial!

Want to fit a polinomial to get a certain behavior

Can use numpy fitting Function

- want it to be 0 at certain points



want it to have roots here

@ many points

4

Could also do a more constuctive, simple manner
- like in lecture

If had a quadratic, Want 0 at A, B. Real

$$g(x) = \text{quadratic} = \alpha x^2 + \beta x + \gamma$$

Want $g(a) = 0$ $\quad g(b) = 0$

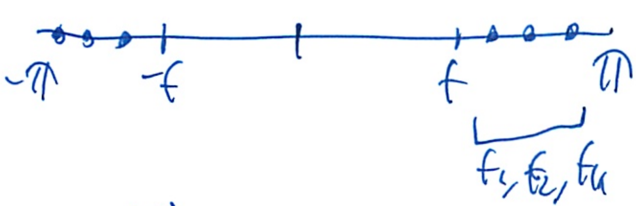
One such one is

$$(x-a)(x-b)$$

Now suppose want $g(a) = g(b) = g(c) = 0$

$$(x-a)(x-b)(x-c)$$

Pick at a few values



$$\underbrace{(x - e^{-jf})(x - e^{jf})}_{\text{1st filter}} \underbrace{(x - e^{-jf_1})(x - e^{jf_2})}_{\text{2nd filter}} \cdots (x - e^{-jf_k})(x - e^{jf_k})$$

$e^{-j\Omega}$

The more points you have, the better the approx.

⑤

When put filter in series → multiply

$$\boxed{(x-e^{-jf})(x-e^{jf})} \to \boxed{(x-e^{-jf_1})(x-e^{jf_1})} \to \ldots \to \boxed{(x-e^{-jf_k})(x-e^{jf_k})}$$

So this is same as previous filter

What should its small hs be?

quadratic expansion

$$X^2-(e^{jf}-e^{-jf})x+1$$

$h(2)=1$

$h(1)=-2\cos(f)$

$h(0)=1$

Only powers that are there read off coefficents

$H(x)=\sum h(m)x^m$

$x=e^{-j\Omega}$

Often this seems so technical – need to step back + see bigger pic

$$e^{jx}=\cos x+j\sin x$$

$$e^{-jx}=\cos(-x)+j\sin(-x)$$

$$e^{jx}+e^{-jx}=2\cos(x)$$

Could do in time domain

- Convolve 1 by 1
- too much convolution!

(6)

Theme: using polynomials to find a fit algebraically for a particular response

Add 0 padding to overcome periodicity

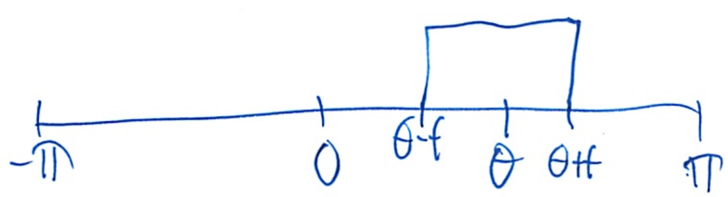
Modulation

If all you care about a certain freq band $-f \to f$

- Use a filter

But what if want to tune into 2 things?

Related to modulation



First design normal filter, centered around 0

Then modulate over $\rightarrow$ to be centered around θ

$$x[n] = a_{\Omega} e^{j\Omega}$$
$$\Omega \in [-f, f]$$

Multiply by $\cos(\theta n) = \frac{1}{2}(e^{j\theta n} + e^{-j\theta n})$

7

Resulting output

$$y[n] = x(n) \cdot \cos(\theta n)$$

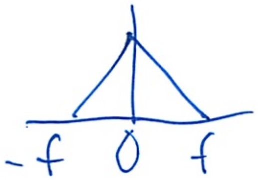
$$= \left(\sum_{\Omega} a_{\Omega} e^{j\Omega n}\right)\left(e^{j\theta n} + e^{-j\theta n}\right) \cdot \frac{1}{2}$$

Ω shifted by $\pm\theta$

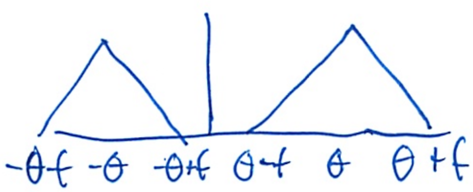
$$= \frac{1}{2}\sum_{\Omega} a_{\Omega} e^{j(\Omega+\theta)n} + \frac{1}{2}\sum_{\Omega} a_{\Omega} e^{j(\Omega-\theta)n}$$

Shifted in 2 parts
(split into 2 pieces)

started w/



$\cdot \cos()$



both showed up

- real part and imag part
  ↑ focusing on real part

So shifting

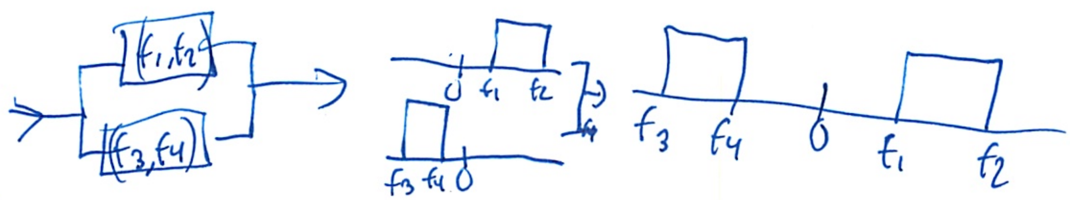
$\cdot$ by $e^{-j\theta n}$

then use a normal filter

8

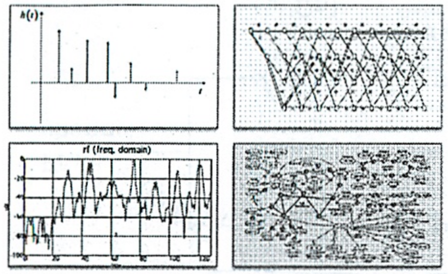
Modulation / filter design kinda the same thing

Filters in parallel



They are like gatekeepers

If don't get through one, try the other!



INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Spring 2011
Lecture #17

- mismatch of rcvr's freq & phase
- quadrature demod
- BPSK, QPSK, DQPSK

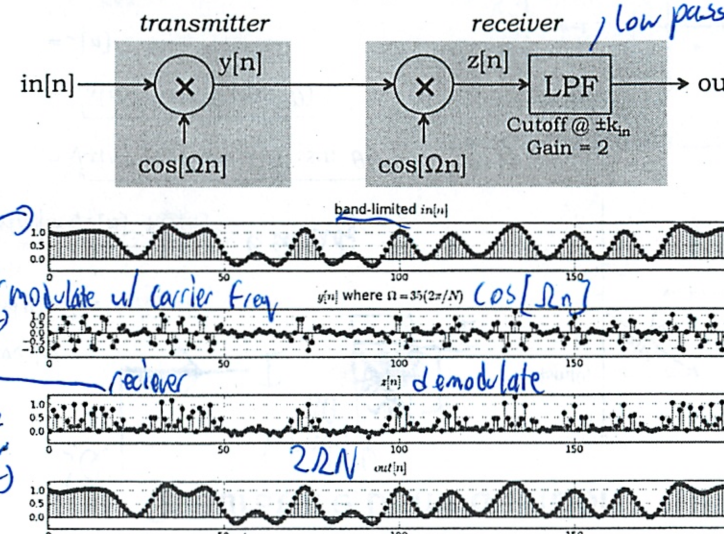
Quiz Tue 7:30-9:30 26-100

No lecture mon

Mon's lecture

Ideal Modulation/Demodulation

low pass filter



cut everything else

modulate w/ carrier freq cos[Ωn]

reciever demodulate

LPF

2ΩN

discretized

just like x[n]

x[n] = in[x]

modulation

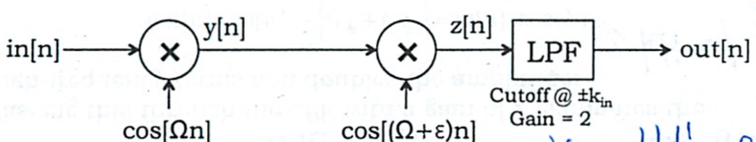
reciever

LPF

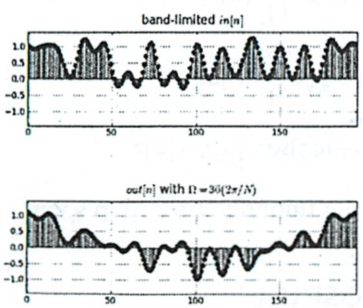
2Ω

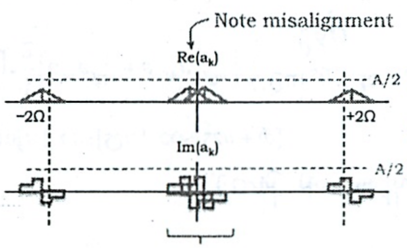
Frequency Error in Demodulator

baseband - very low freq - only 1 can use - huge antenna



if modulation error





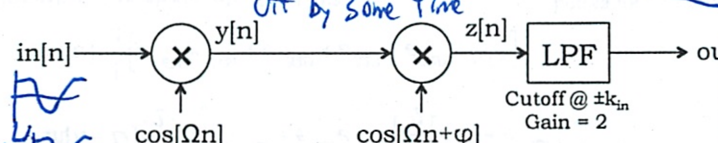
Baseband k's not correct
Combine amplitudes from different k's!

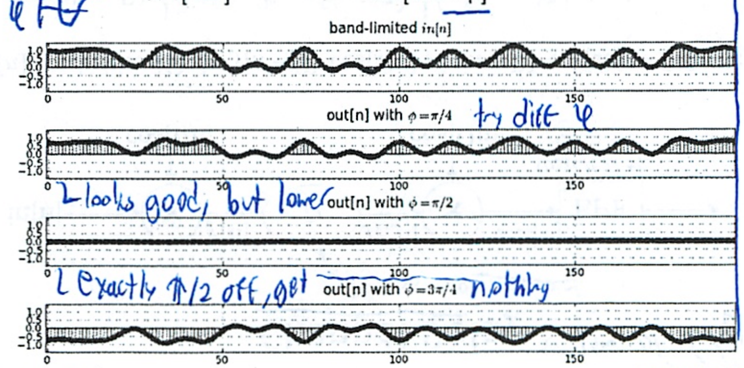
demodulate not where modulate was

Made copies in weird places

Phase Error in Demodulator

off by some time





try diff φ

looks good, but lower

exactly π/2 off, get nothing

Phase Error Math

Let's derive an equation for z[n]: Could do math in cos

$$z[n] = y[n]\cdot\cos[\Omega n+\varphi] = \text{in}[n]\cdot\cos[\Omega n]\cdot\cos[\Omega n+\varphi]$$

$$= \text{in}[n]\cdot\frac{1}{2}\left[e^{j\Omega n}+e^{-j\Omega n}\right]\cdot\frac{1}{2}\left[e^{j\varphi}e^{j\Omega n}+e^{-j\varphi}e^{-j\Omega n}\right]$$

easier to use complex exp

$$= \text{in}[n]\cdot\frac{1}{4}\left[e^{j\varphi}e^{j2\Omega n}+e^{j\varphi}+e^{-j\varphi}+e^{-j\varphi}e^{-j2\Omega n}\right]$$

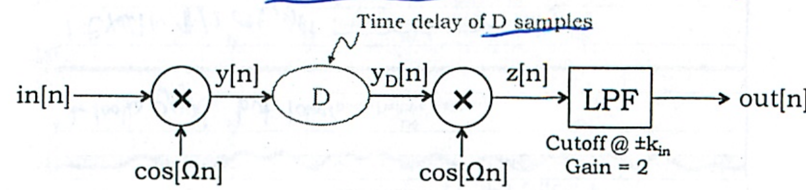
scaling factor 2Ω disappear

Passing this through the LPF with a gain of 2 eliminates the high-frequency terms and doubles the amplitude: ← left w/ just

$$\text{out}[n] = \text{in}[n]\cdot\frac{2}{4}\left[e^{j\varphi}+e^{-j\varphi}\right] = \text{in}[n]\cdot\cos(\varphi)$$

So a phase error of φ results in amplitude scaling of cos(φ).

Channel Delay = time of flight



$$z[n] = y_D[n]\cdot\cos[\Omega n] = y[n-D]\cdot\cos[\Omega n] = \text{in}[n-D]\cdot\cos[\Omega(n-D)]\cdot\cos[\Omega n]$$

$$= \text{in}[n-D]\cdot\frac{1}{2}\left[e^{-j\Omega D}e^{j\Omega n}+e^{j\Omega D}e^{-j\Omega n}\right]\cdot\frac{1}{2}\left[e^{j\Omega n}+e^{-j\Omega n}\right]$$

$$= \text{in}[n]\cdot\frac{1}{4}\left[e^{-j\Omega D}e^{j2\Omega n}+e^{-j\Omega D}+e^{j\Omega D}+e^{j\Omega D}e^{-j2\Omega n}\right]$$

Passing this through the LPF: Looks like a phase error

$$\text{out}[n] = \text{in}[n]\cdot\frac{2}{4}\left[e^{-j\Omega D}+e^{j\Omega D}\right] = \text{in}[n]\cdot\cos(\Omega D)$$

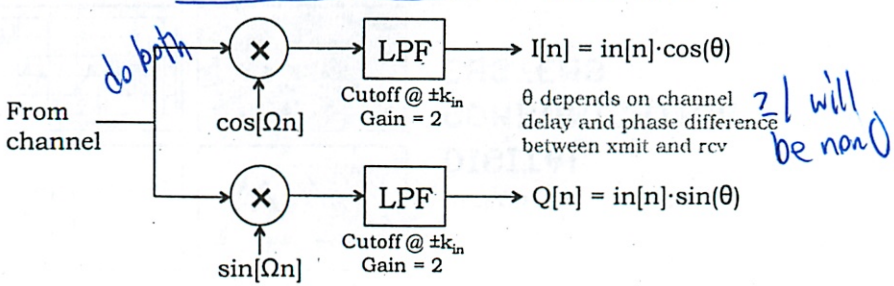
Channel delay + phase error = same issue
just don't want $\frac{\pi}{2}$ – take a giant step
had to do on old radios

Fixing Phase Problems in the Receiver

So phase errors and channel delay both result in a scaling of the output amplitude, where the magnitude of the scaling can't necessarily be determined at system design time:

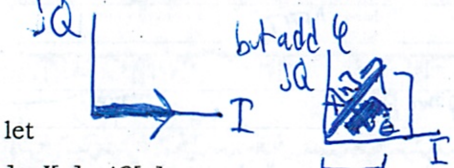
- channel delay varies on mobile devices
- phase difference between transmitter and receiver is arbitrary

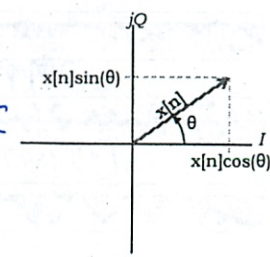
One solution: *quadrature demodulation* ← modern radios use



↑ could compensate for delays

Quadrature Demodulation





If we let

$$w[n] = I[n] + jQ[n]$$

then ← so add

$$|w[n]| = \sqrt{I[n]^2+Q[n]^2}$$

$$= \sqrt{x[n]^2\cdot\cos^2\theta + x[n]^2\cdot\sin^2\theta}$$

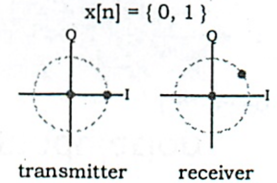
$$= \sqrt{x[n]^2\cdot(\cos^2\theta+\sin^2\theta)}$$

$$= x[n]$$

belt + suspenders approach

Constellation diagrams:

x[n] = { 0, 1 }



Other types of modulation systems

Phase Modulation

A sinusoid is characterized by its frequency, amplitude and phase – one can modulate anyone of these using x[n], which represents the message to be transmitted.

- Amplitude modulation (AM) – what we've done so far – times when signal goes away
- Frequency modulation (FM)
- Phase modulation – our next topic

detect changes in phase

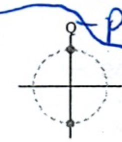
Using AM the signal can have zero amplitude, indistinguishable from no signal at all, which can confound the circuitry that "fine tunes" the amplitude and frequency at the receiver.

Using phase modulation, aka phase-shift keying (PSK) (aka), the transmitted signal has constant amplitude; information is encoded in the phase of the carrier sinusoid.

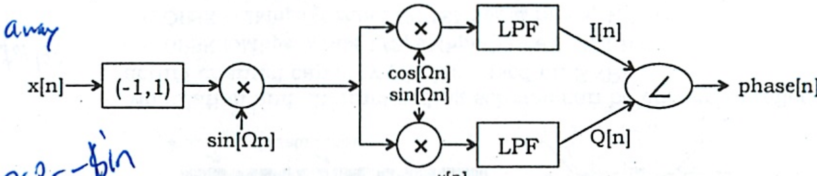
always getting cos in

Need to tell 3 things about waves
- mag
- freq
- phase

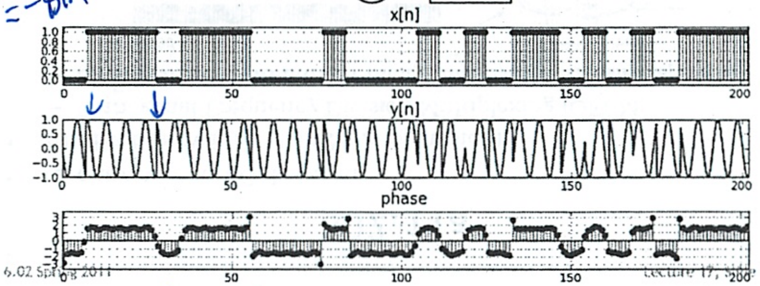
BPSK



In binary phase-shift keying (BPSK), the message bit selects one of two phases for the carrier, e.g., $\pi/2$ and $-\pi/2$.



sin $-180° = -$sin



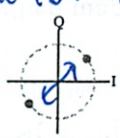
every time it changes it changes phase 180°

Calc phase angle – could sample, get x[n]

apparent phase change at reciever

Dealing With Phase Ambiguity

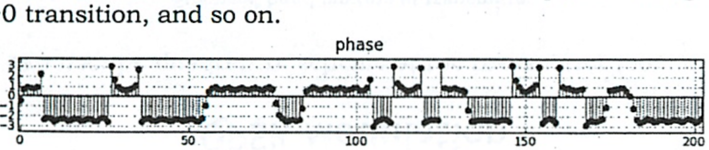
reciever going back and forth b/w



BPSK is also subject phase changes introduced by channel delays or phase difference between xmit and rcv: the received constellation will be rotated with respect to the transmitter's constellation. Which phase corresponds to which bit?

was it $\to$ or $\leftarrow$ or ... can't tell

The fix? Think of the phase encoding as *differential*, not absolute: a change in phase corresponds to a change in bit value. Assume that, by convention, messages start with a single 0 bit, i.e., prepend a 0 to each to message. Then the first phase change represents a 0→1 transition, the second phase change a 1→0 transition, and so on.



don't know which was 1 or 0
- put know, Change in Phase

Differential PSK

many variations on differential means

- Approach 1: encode bits
  - DBPSK – process one message bit at a time
    - "0": phase change is 0
    - "1": phase change is π
  - DQPSK – process two message bits at a time
    - "00": phase change is 0
    - "01": phase change is $\pi/2$
    - "10": phase change is π
    - "11": phase change is $3\pi/2$
- Approach 2: encode transitions
  - DBPSK
    - No transition: phase change is 0
    - Transition: phase change is π
  - DQPSK: assume cyclic order is 00, 01, 11, 10
    - No advance in order: phase change is 0
    - Advance one position in order: phase change is $\pi/2$
    - And so on…

What about 3 bits at time, i.e., 8 constellation points?

Could encode in 2 phases

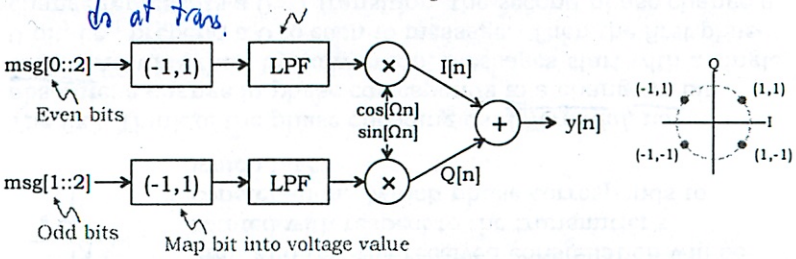
Noise gives cloud of dots – cuts into noise margins

be flexible!

QPSK Modulation

We can use the quadrature scheme at the transmitter too:

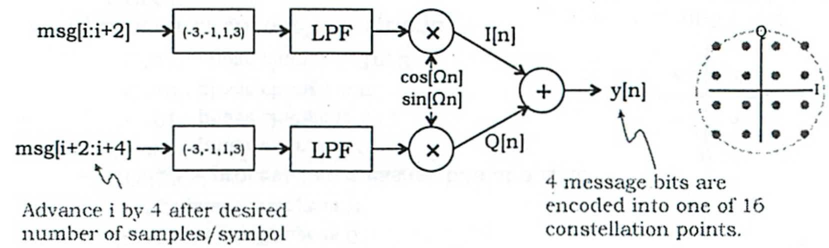
do at trans.



When mapping bits to voltage values, we should choose the values so that the maximum amplitude of y[n] is 1. For QPSK (also referred to as QAM-4) that would mean $\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)=(.707,.707)$

Quadrature Amplitude Modulation (QAM)

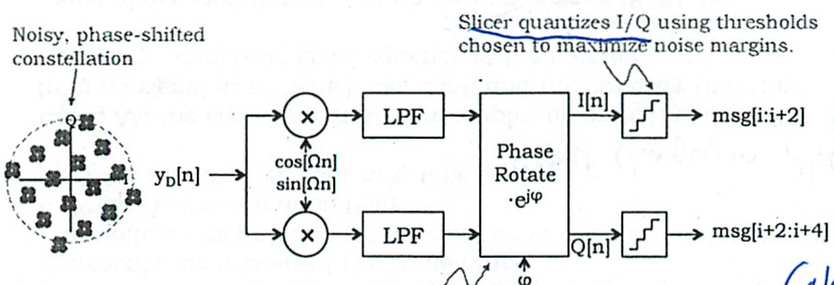
Using more message bits at a time, we can generate larger constellations using quadrature amplitude modulation. Here's a diagram of a QAM-16 system with 16 constellation points:



Larger constellations mean points are closer together, so there's more sensitivity to noise. Some systems adapt the size of the constellation to the noise level (QAM-4, QAM-16, QAM-64, ...), i.e., use 2, 4, 6, ... bits/symbol.

QAM Receiver

Here's a simplified diagram of a QAM-16 receiver:



did salelight chips when it first started

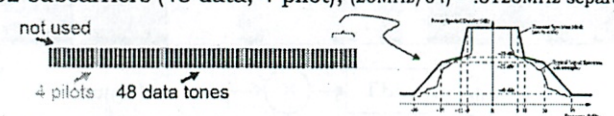
Calculated φ

Complex multiply needed to rotate constellation to correct for phase change from channel delay and phase offset. φ determined by looking at msg bits.

— so constillations come out right

802.11a

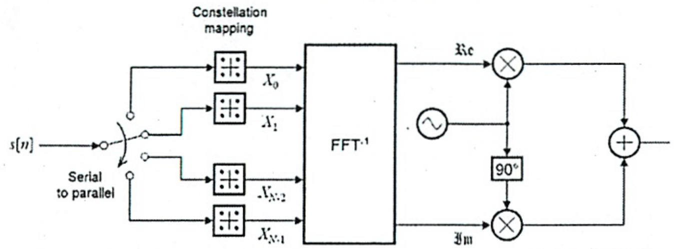
- 12 channels in 5GHz band
- 20MHz bandwidth (16.6MHz occupied)
  - Orthogonal Frequency Division Multiplexing (OFDM)
  - 52 subcarriers (48 data, 4 pilot), (20MHz/64) = .3125MHz separation



  - Modulation and channel coding scheme can be chosen to reflect actual channel capacity (choice based on SNR):
    - BPSK (6Mbps @ rate 1/2, 9Mbps @ rate 3/4, 1 bit)
    - QPSK (12Mbps @ rate 1/2, 18Mbps @ rate 3/4, 2 bits)
    - 16-QAM (24Mbps @ rate 1/2, 36Mbps @ rate 3/4, 4 bits)
    - 64-QAM (48Mbps @ rate 2/3, 54Mbps @ rate 3/4, 6 bits)
  - Symbol duration 4us (includes guard interval of 0.8us)
    - ~250k combined symbols per second, 48 data channels
    - ~Data rate = (48)(250k)(# bits/symbol)(code rate)
      = (12M)(# bits/symbol)(code rate)

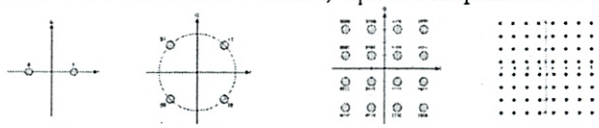
http://stellar.mit.edu/S/course/6/sp06/6.973/ -- Lecture 4

OFDM Transmitter



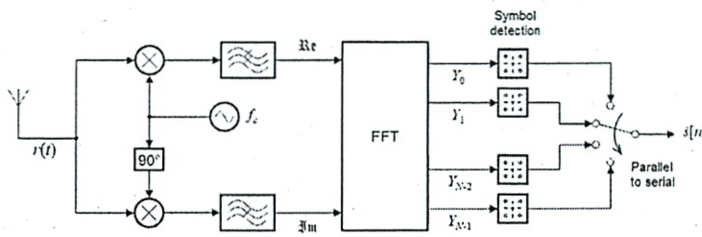
http://en.wikipedia.org/wiki/OFDM

- Signal is interleaved across multiple subchannels
  - For bidirectional links, can use some subchannels for each direction
- Constellation mapping can be chosen separately for each subchannel: for other than BPSK, X_i are complex values



BPSK QPSK 16-QAM 64-QAM

OFDM Receiver



- Need good frequency synchronization at receiver to keep subchannels orthogonal, need good gain control to keep amplitudes correct for slicing.
- Low-pass filter selects demodulated baseband signal
- Symbol detection for each subchannel is matched to modulation scheme selected by transmitter

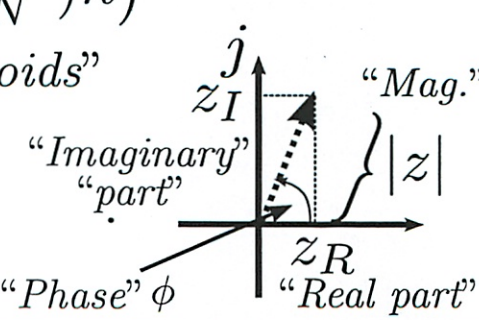
	Time	Frequency
Must know	*"Decomposition"*	*"Get Freq. coefs."*
	$x[n]=\sum_{<k>} a_k e^{j(\frac{2\pi k}{N})n}$	$a_k = \sum_{<n>} x[n] e^{-j(\frac{2\pi k}{N})n}$
	$x[n]$ has N samples /periodic in N	$a_k = a_{k+N}$ "periodic in N"
	$x_1[n] + x_2[n]$	$a_k = b_k + c_k$ *"Linear"* $x_1[n] \leftrightarrow b_k,\ x_2[n] \leftrightarrow c_k$
	$\alpha \cdot x[n]$	$\alpha \cdot a_k$ *"Linear"*
Properties	$x[n]$ is real	$a_k = a^*_{-k}$ *"Conjugate Sym."*
	$x[n]$ real and $x[n]=x[-n]$	a_k's are real
	$x[n-\alpha]$	$a_k \cdot e^{-j(\frac{2\pi k}{N})\alpha}$ *"Time Shift"*
	$x[n] \cdot e^{j(\frac{2\pi k}{N})\alpha}$	$a_{k-\alpha}$ *"Freq. Shift"*
Examples	$x[n] = 1$	$a_k = \delta[k]$ *"DC-constant"*
	$x[n] = \delta[n]$	$a_k = \frac{1}{N}$ *"unit sample"*
	$x[n] = \cos\left((\frac{2\pi \ell}{N})n\right)$	$a_k = \frac{1}{2}\delta[\ell] + \frac{1}{2}\delta[-\ell]$
	$x[n] = \sin\left((\frac{2\pi \ell}{N})n\right)$	$a_k = \frac{1}{2j}\delta[\ell] - \frac{1}{2j}\delta[-\ell]$

$$e^{j\underbrace{(\frac{2\pi k}{N})}_{\text{"Freq." (fraction of } 2\pi)}n} = \cos\left((\frac{2\pi k}{N})n\right) + j\sin\left((\frac{2\pi k}{N})n\right)$$ *"Euler's"*

"discrete-time sinusoids"

"A complex number has 4 parts"

$$z = \underbrace{z_R}_{|z|\cos(\phi)} + j\underbrace{z_I}_{|z|\sin(\phi)} = |z|e^{\phi}$$



$\lvert z\rvert^2 = z_R^2 + z_I^2$	$\phi = \tan^{-1}(\frac{z_I}{z_R})$

"Conjugation"

$z^* = z_R - jz_I = |z|e^{-\phi}$

Massachusetts Institute of Technology
Dept. of Electrical Engineering and Computer Science
Fall Semester, 2006
6.082 Introduction to EECS 2

Modulation and Demodulation

Introduction

A communication system that sends information between two locations consists of a *transmitter*, *channel*, and *receiver* as illustrated in Figure 1. The channel refers to the physical medium carrying the *information signal* (voice, video, data etc.) from one location to another. The physical medium can be free space or a variety of waveguides (wires, optical fibers, etc.) that direct the energy across the channel to the receiver. The *transmitted signal* carrying the information through the channel can be electromagnetic, optical, acoustic, or other forms of energy radiation. Cell phones and wireless networks send information across free space using electromagnetic waves.

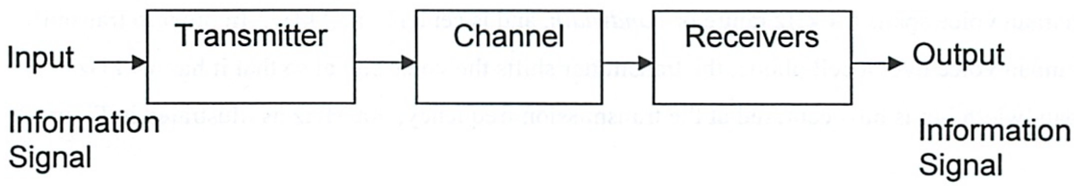


Figure 1: Communication System Block Diagram

In order to send these electromagnetic waves across free space the frequency of the transmitted signal must be quite high compared to the frequency of the information signal. For example the information signal in a cell phone is a voice signal with a bandwidth on the order of 4kHz. The typical frequency of the transmitted and received signal is on the order of 900MHz. Figure 2 illustrates how the spectrum of voice signals falls outside the frequency range of the transmission channel; the figure is not drawn to scale.

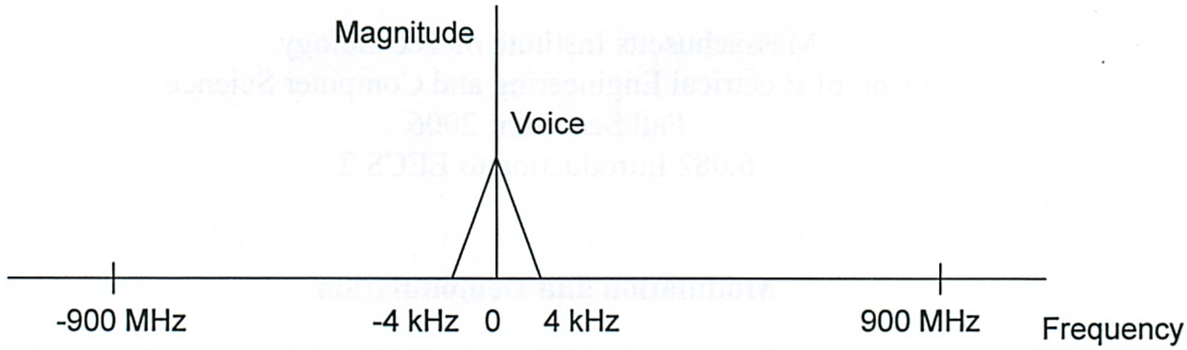


Figure 2

The main reason the transmission frequency is so high is that the wavelength of the electromagnetic wave is proportional to the reciprocal of the frequency. For example the wavelength of a 1 GHz electromagnetic wave in free space is 30cm whereas a 1kHz electromagnetic wave is one million times larger or 30km. As you will see when we study antennas, the size of the antenna and other components are related to the wavelength. For small portable devices, higher frequency transmission is a requirement.

To transmit signals with frequencies required by the communication channel, *the transmitter centers the spectrum of the information signal at the transmission frequency.* This process of shifting the frequency spectrum of a signal is called *modulation.* As an example human voice spans a 4 kHz range or *bandwidth*, and is centered at 0 kHz. In order to transmit human voice over a cell phone, the transmitter shifts the voice signal so that it has a 4 kHz bandwidth but is now centered at the transmission frequency, 900MHz as illustrated in Figure 3.

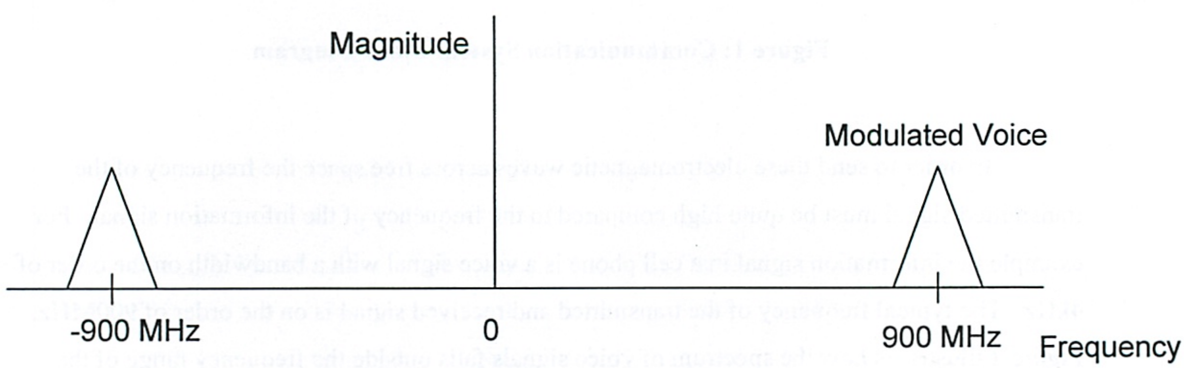


Figure 3

In the receiver the reverse process takes place. The receiver *centers the spectrum of the received signal at the original center frequency of the information signal*; we refer to this process as *demodulation.* In the case of human voice transmission, the receiver shifts the spectrum of the received signal (the received signal had a spectrum centered at 900 MHz) so that it is centered at 0 kHz as shown in Figure 4.

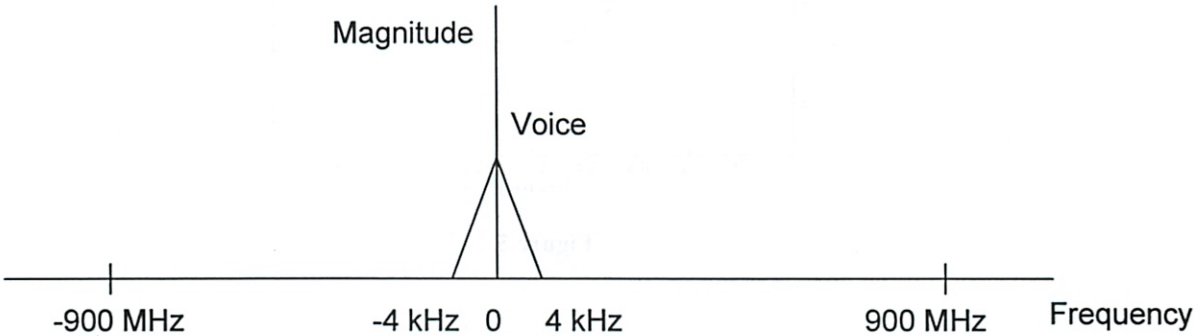


Figure 4

Modulation

We take advantage of the trigonometric identity shown below in the implementation of signal modulation. The identity shows that the product of cosines with frequencies f_1 and f_2 results in cosines with frequencies f_1+f_2 and f_1-f_2. *In other words, multiplication by the f_2 cosine shifts or modulates the f_1 cosine to the new frequencies f_1+f_2 and f_1-f_2.* modulation

baseband

$$\cos(2\pi f_1 t) * \cos(2\pi f_2 t) = \frac{\cos\{2\pi(f_1 + f_2)t\} + \cos\{2\pi(f_1 - f_2)t\}}{2}$$

Let's study the effect of modulation in the time and frequency domain; assume $f_1 = 1$ Hz and $f_2 = 10$ Hz. Figure 5 and 6 show the time-domain plots of the 1 Hz and 10 Hz cosines, and Figure 7 shows the time-domain plot of the product of these two cosines. Notice how the 1 Hz cosine appears as the *envelope* that shapes the 10 Hz cosine.

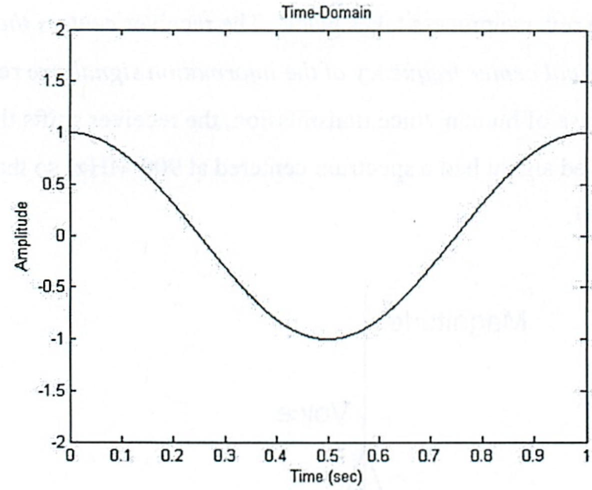


Figure 5

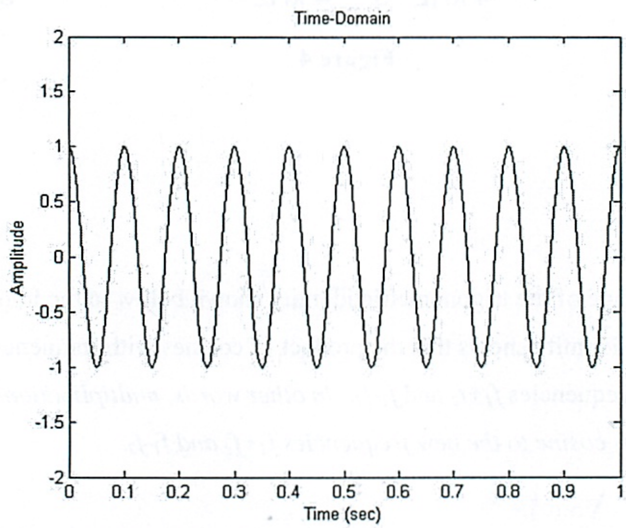


Figure 6

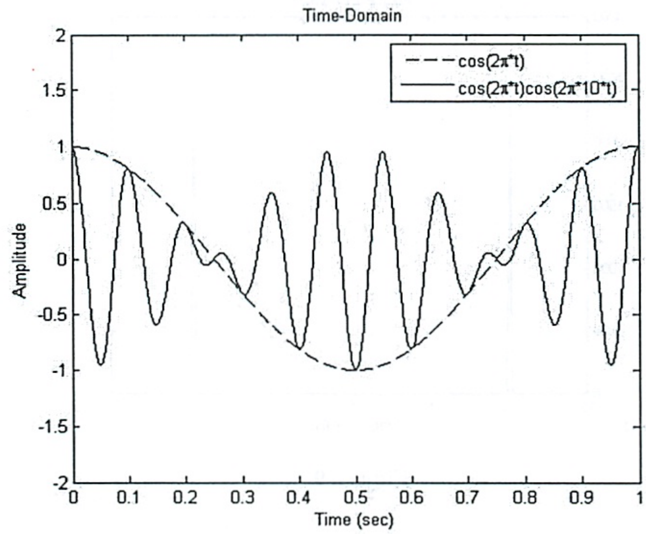


Figure 7

In the frequency domain the 1 Hz and 10 Hz cosines appear as illustrated in Figures 8 and 9 respectively; recall that spectrum of a real signal has even magnitude, which is why you see spectral peaks at +/- 1 Hz and +/- 10 Hz.

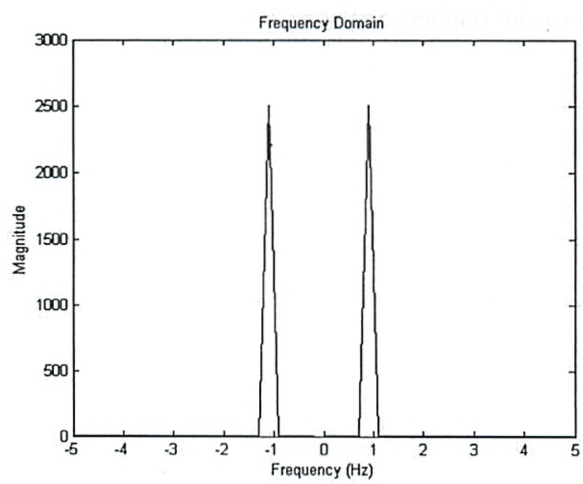


Figure 8

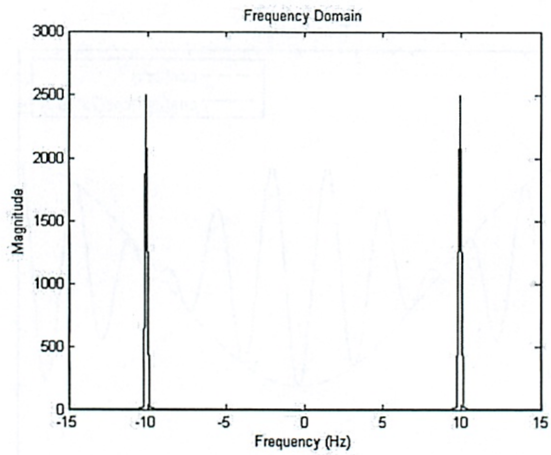


Figure 9

Figure 10 illustrates the spectrum of the product of the 1 and 10 Hz cosines; note how the spectrum has four peaks. The two spectral peaks at +/- 11 Hz correspond to the f_1+f_2 cosine while the two peaks at +/- 9Hz correspond to the f_1-f_2 cosine. At this point, we have modulated a 1 Hz cosine up to 9 and 11 Hz. Of course we could have also said that we modulated the 10 Hz signal to 9 and 11 Hz, but it is customary to think of the lower frequency signal as the information signal that we modulate up to the carrier (higher) frequency signal.

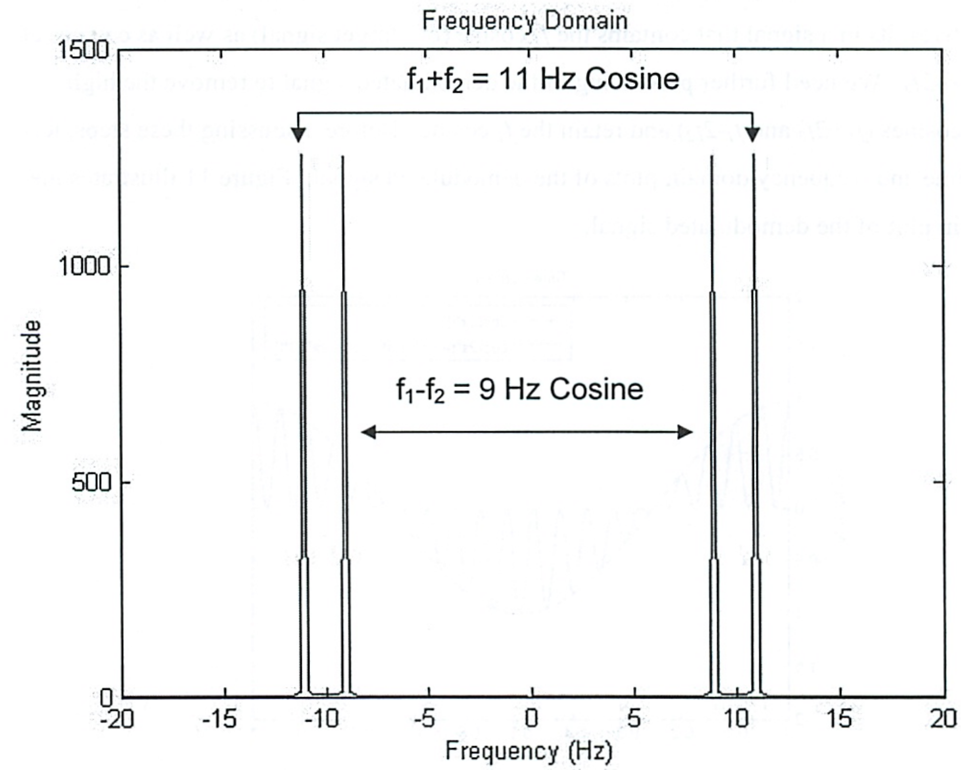


Figure 10

Demodulation

We know that demodulation involves shifting spectra, and that shifting spectra involves multiplication by cosines. Let's multiply the modulated signal from the previous section (the signal containing the f_1+f_2 and f_1-f_2 cosines) by another cosine with a frequency of f_2. The hope is that this operation will allow us to recover the f_1 cosine.

$$\frac{1}{2}\{\cos(2\pi(f_1+f_2)t)+\cos(2\pi(f_1-f_2)t)\}*\cos(2\pi f_2 t)$$

$$=\frac{1}{2}\{\cos(2\pi(f_1+f_2)t)*\cos(2\pi f_2 t)+\cos(2\pi(f_1-f_2)t)*\cos(2\pi f_2 t)\}$$

$$=\frac{1}{2}\{\frac{\cos(2\pi(f_1+2f_2)t)+\cos(2\pi f_1 t)}{2}+\frac{\cos(2\pi f_1 t)+\cos(2\pi(f_1-2f_2)t)}{2}\}$$

$$=\frac{1}{4}\cos(2\pi(f_1-2f_2)t)+\frac{1}{2}\cos(2\pi f_1 t)+\frac{1}{4}\cos(2\pi(f_1+2f_2)t)$$

From the above analysis, we see that demodulating by multipyling by a cosine with frequency f_2 results in a signal that contains the f_1 cosine (our target signal) as well as cosines at f_1+2f_2 and f_1-2f_2. We need further processing of the demodulated signal to remove the high frequency cosines (f_1+2f_2 and f_1-2f_2) and retain the f_1 cosine. Before discussing these steps, lets study the time and frequency domain plots of the demodulated signal. Figure 11 illustrates the time-domain plot of the demodulated signal.

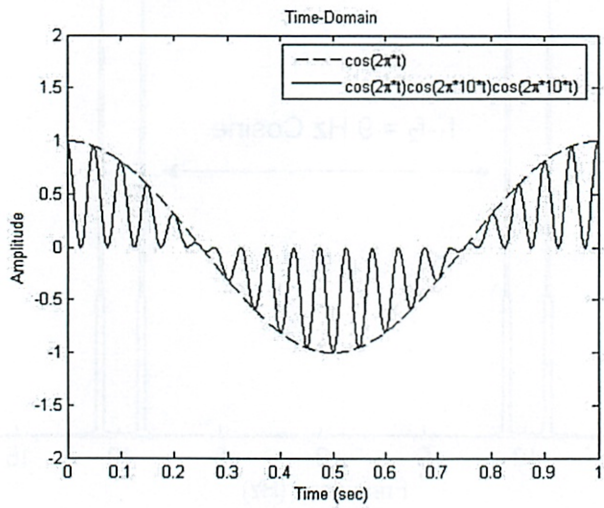


Figure 11

Figure 12 illustrates the spectrum of the demodulated signal; note how the spectrum has six peaks. The two spectral peaks at +/- 21 Hz correspond to the f_1+2f_2 cosine; the two peaks at +/- 19 Hz correspond to the f_1-2f_2 cosine; and the two spectral peaks at +/- 1 Hz correspond to the f_1 cosine.

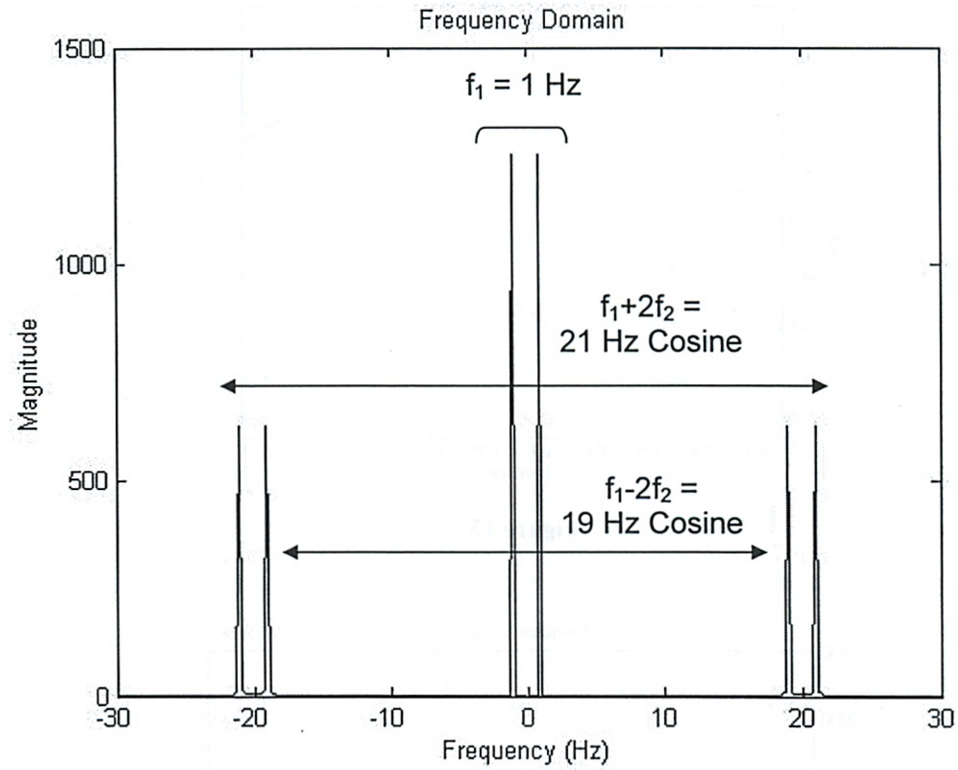


Figure 12

To remove the higher frequency cosines (f_1+2f_2 and f_1-2f_2) and only retain the f_1 cosine we will apply a low frequency filter with cutoff frequency $f_c = 10$ Hz. We will learn about filtering in the next lecture. Figure 13 and 14 illustrate the time and frequency plots of the demodulated signal after the low pass filter; note all that is left is the f_1 cosine (target signal).

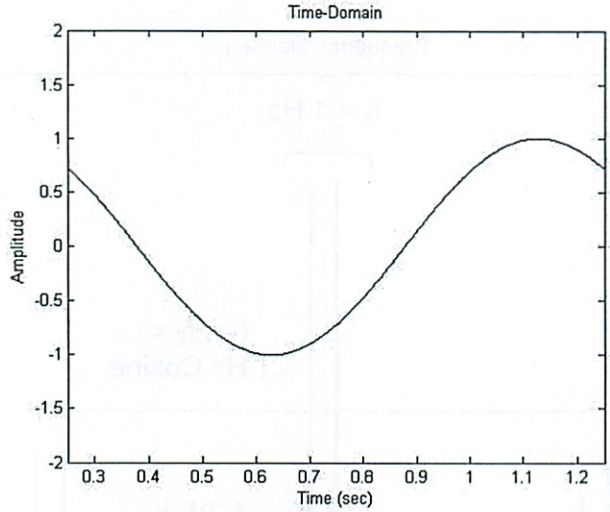


Figure 13

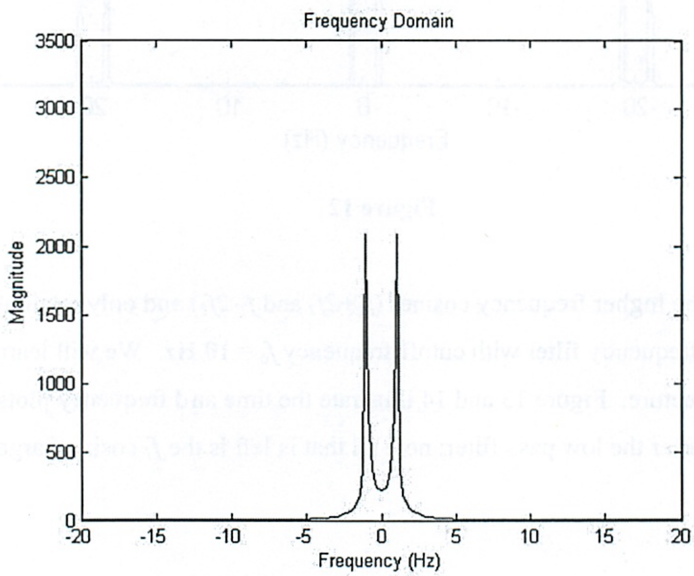


Figure 14

Modulating Signals With Nonzero Bandwidth

So far we have been dealing with the modulation of sinusoids; these signals have a single frequency component and zero bandwidth. Nonetheless signals with non-zero bandwidths, such as the voice signal whose time and frequency domain representations are shown in Figures 15 and 16, are also modulated by multiplying the signal by a cosine. The reason this works is that these signals are made up of sine and cosines (recall Fourier series and transform), and modulation simply moves each of those sine/cosine components (and therefore the entire signal) up/down the spectrum.

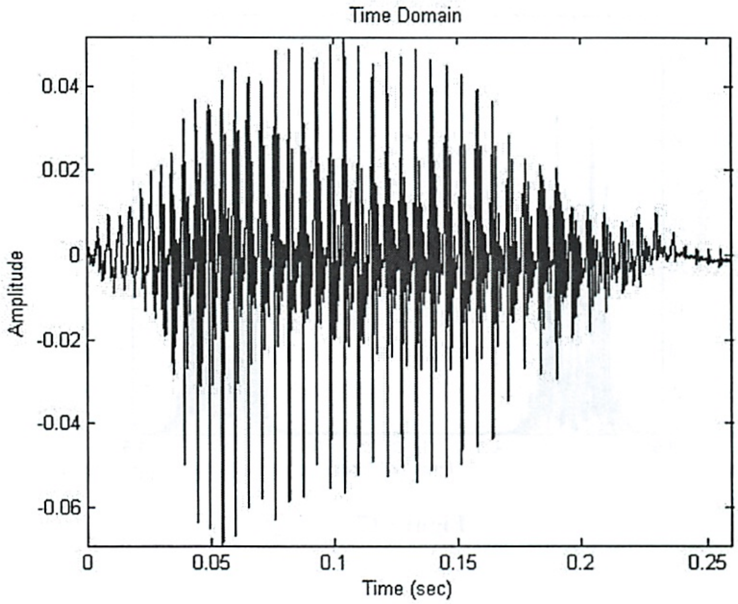


Figure 15

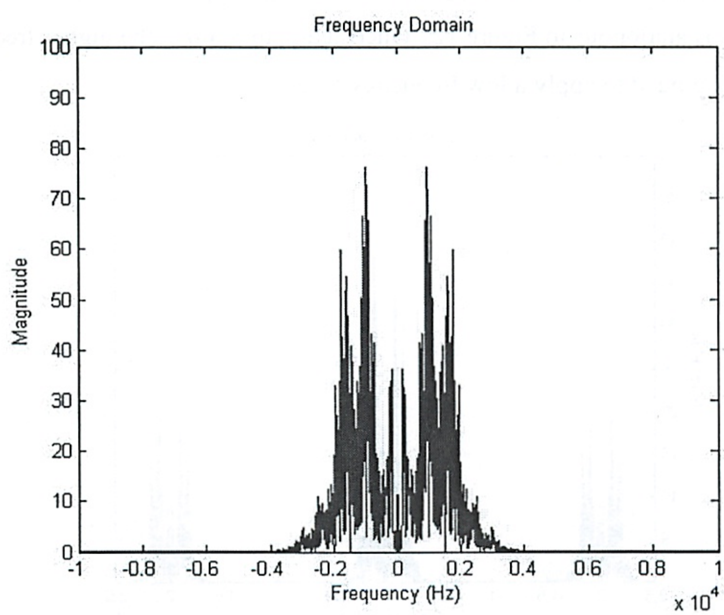


Figure 16

As an example, Figure 17 illustrates the spectrum of the voice signal in Figure 15 after it is modulated by a 10 kHz cosine. Note how the spectrum in Figure 16 is centered at +/- 10kHz; the plot is analogous to Figure 10.

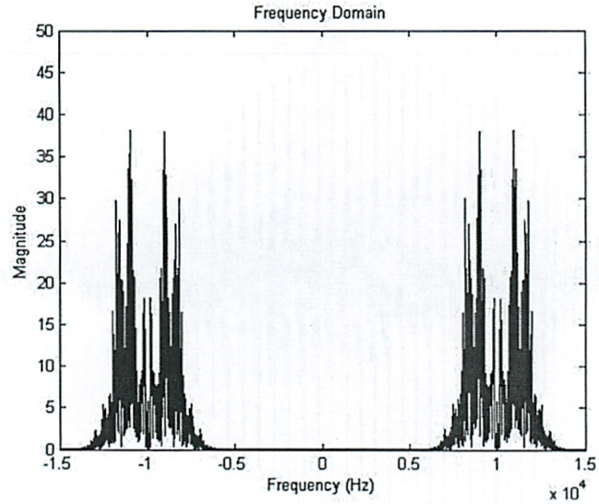


Figure 17

Now we will demodulate the voice signal back to DC (0 Hz). Figure 18 illustrates the spectrum of the voice signal after demodulation; multiplying the modulated signal (Figure 17) by a 10kHz cosine. Note that we have one copy of the voice signal spectrum at 0Hz and two copies at +/- 20kHz, which is analogous to Figure 12. Once again, to remove the higher frequency copies at +/-20kHz we need to apply a low frequency filter.

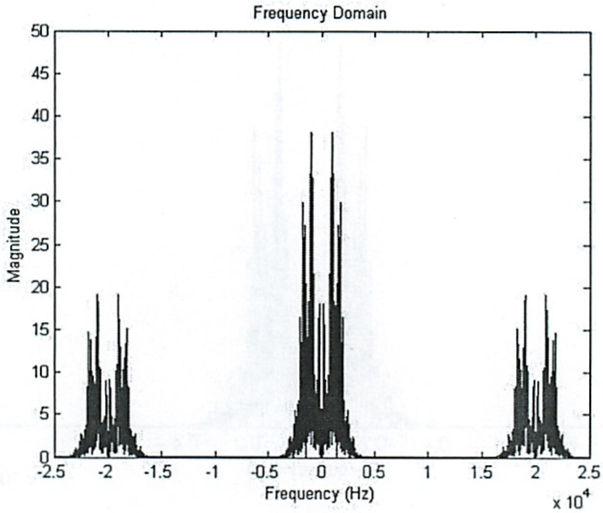


Figure 18

Modulation and Demodulation Summary

Figure 19 is a block diagram that summarizes the steps involved in modulating and demodulating a signal by a frequency f_2. Modulation involves multiplying the input signal x(t) by a cosine with a frequency f_2. Demodulation involves multiplying the modulated signal again by a cosine with a frequency f_2, and then applying a low pass filter with cutoff frequency f_2. The low pass filter removes the high frequency components centered about $2f_2$.

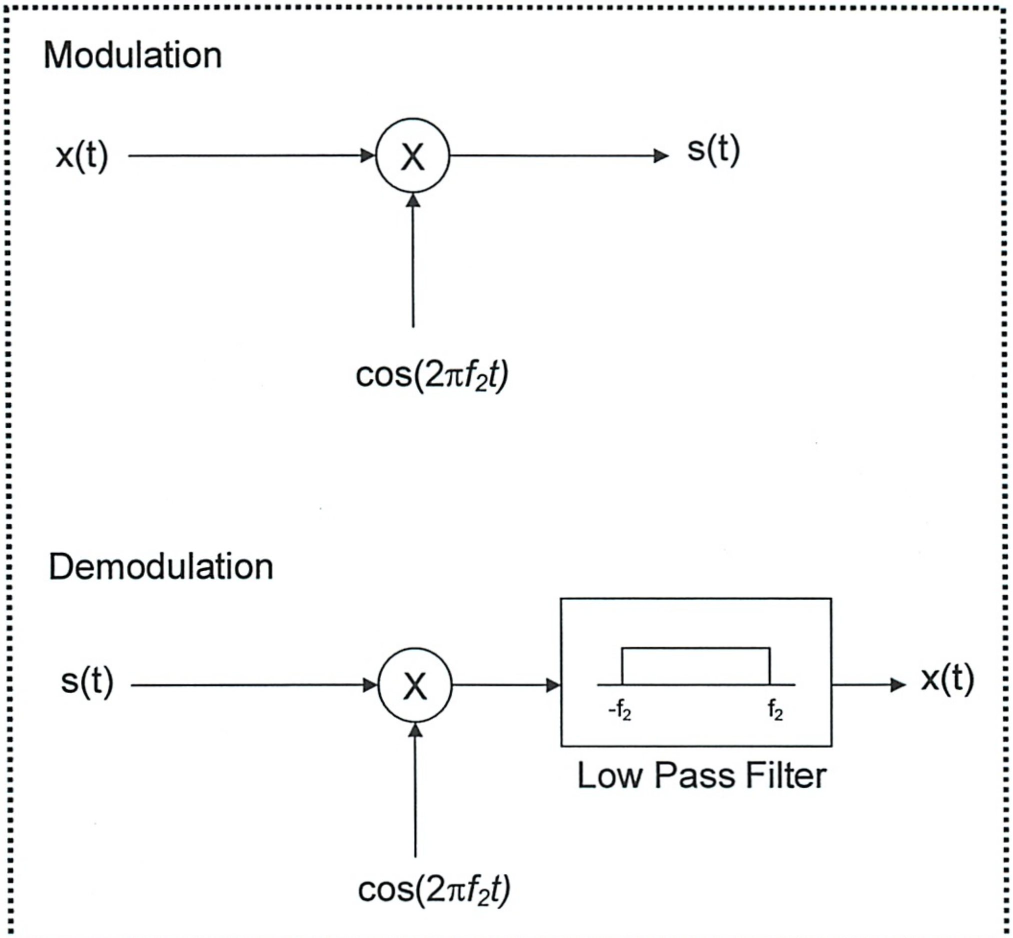
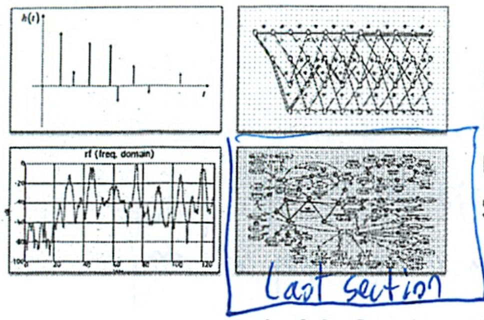


Figure 19

INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS



6.02 Spring 2011
Lecture #18

- multi-hop networks: design criteria
- network topologies
- circuit vs. packet switching
- queues, Little's Law

7 more lectures

Next Steps…

Spent all our time on →

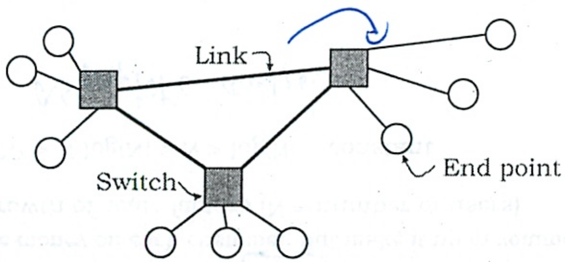
- Have: digital point-to-point

- Want: many interconnected points

build out global network

– multi-hop

Multi-hop Networks



How do switches communicate among themselves?

- Switches orchestrate flow of information through the network, often multiplexing many logically-independent flows over a single physical link.
  - Appropriate sharing of network resources
  - Circuit switching vs. packet switching

    set up a circuit for entire comm
- Design criteria
  - Reliability, scalability, performance, cost, security, …

Appropriate sharing of network

Network Reliability

- Design engineers: ↓ buy reliable components
  - Low MTBF components, failure prediction ← will know if it is about to fail
  - Easy to identify and fix problems
    - Remote observability and controllability
  - Replace/expand/evolve network incrementally
  - Defend against malicious users
- Redundancy – very hard on this
  - No single point of failure
  - Fail-safe, Fail-safe, Fail-hard – not the same ever
  - Automated adaptation to component failures – think about common modes of failure
- Degradation, not failure
- Users: error feedback
  - High availability
  - Meaningful feedback on failure (e.g., busy signal)

4/13

Communications network only valuable if it can expand

Network Scalability

- Enable incremental build-out
  - increase in usage involves incremental costs (both at edges and in interior of network)
  - Address bottlenecks without fundamental changes
- Economies of scale
  - Larger number of users → less cost/user
  - "lose money on each customer, but make it up in volume"
- Slow growth of scale factors (N = number of users)

$2^N > N^2 > N \log(N) > N > \log(N) >$ constant

asymptotic analysis

Network Performance

- Design Engineers
  - Utilization
  - Minimize protocol overhead
  - Quality of Service (performance tiers)
- User
  - Throughput (guaranteed minimum)
    - Opportunistic improvements
  - Latency (guaranteed maximum)
    - One-way, round-trip
  - Isochrony?

getting guaranteed speed

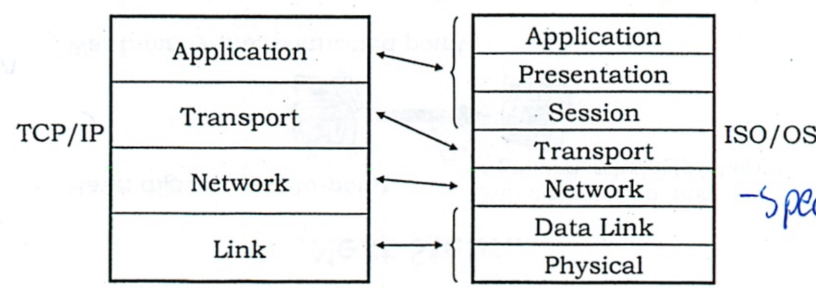
Network Costs

- NRE (non-recurring expenses, ie, one-time costs)
- Basic infrastructure
- Per connection
- Per message transported
- Economies of scale, amortization

has to last for a while
must do in cost-effective way

Dealing with System Complexity

- Manage complexity with abstraction layers
  - Easier to design if details have been abstracted away
  - Desired behavior implemented using only component behaviors (not component implementation details), but check that correct behavior actually happened!
  - Change implementation without changing behavior
- Communication Network Layers Series of compartments



-Specs/APIs

Packet vs network switching

Abstraction Layers

Application Layer:
HTTP, Mail, FTP, Telnet, RPC, DNS, NFS

Transport Layer (eg, TCP or UDP):
Using packets to create illusion of continuous stream of data. In-order, connection-based vs. connectionless

Best-effort

Network Layer (eg, IP):
Addressing and routing fixed-length datagrams through a multi-hop network, breaking up larger packets

← Won't spend time on this

Link Layer:
Channel access, how bits are transmitted on the channel, framing of data, channel-specific addressing, channel coding.

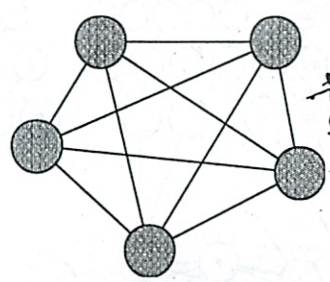
Spent a lot of time on this

Network Topologies

Networks built from two types of channels:

- Point-to-point channels ← easiest to think about
  - Simplex: one-way communication
  - Half-duplex: bidirectional, but one-way at a time
  - Full-duplex: simultaneous bidirectional communications
    - Symmetric and asymmetric bandwidths
- Multiple-access channels (eg, wireless, ethernet)
  - Sharing mechanism:
    - Regimented: Fixed allocations in time or frequency TDMA
    - Ad-hoc: Contend for use, eg, collision detect/back-off
      Contention
- Local-area (LANs) vs. wide-area (WANs)
  - Appropriate technologies? topologies?

Fully-Connected Graph



Asymptotic analysis: Each point-to-point link requires 1 unit of hardware, each communication, 1 unit of time

Real life is ~~a hybrid~~ a hybrid b/w the two

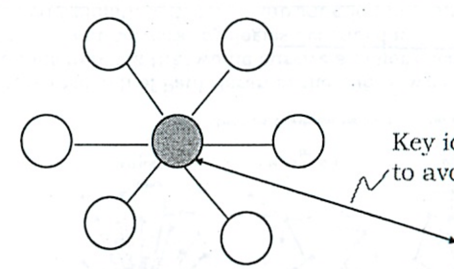
**But in the real world, hardware lives in 3-space, so N nodes take at least $O(N^{1/3})$ space which means that comm. time grows as $O(N^{1/3})$*

- Throughput: $O(N^2)$
- Latency: $O(1)$
- Cost: $O(N^2)$ overall, $O(N)$ per node – very expensive
- no single points of failure, many alternate paths
- no blocking/congestion
- Expensive, doesn't easily scale to large regions

each link has cost 1

So much connectivity so lots of redundancy

Star

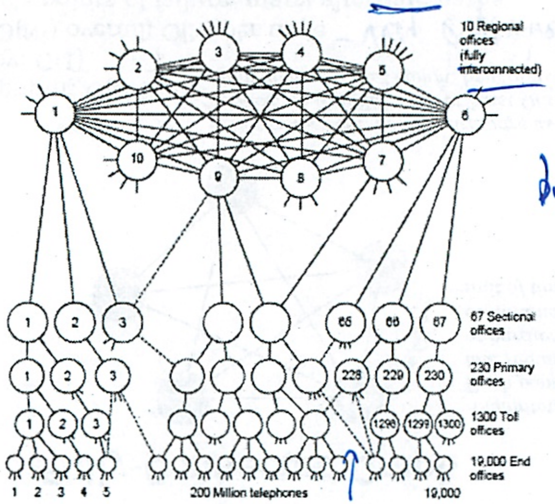


- Throughput: $O(N)$
- Latency: $O(1)$* ← almost as fast
- Cost: $O(N)$ for center node, but only $O(1)$ for leaf node much cheaper
- single point of failure!
- congested link affects all communications to that node
- Inexpensive, long links possible

but failure in central node will be big deal

Example: Telephone Network

ATT North America, c. mid-1990's



$$$

but very reliable

here more star

cheating wires
-not a strict tree

Wide Area Networks

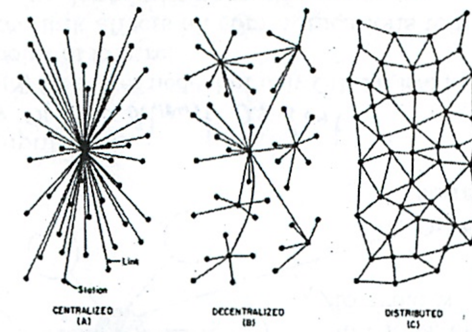
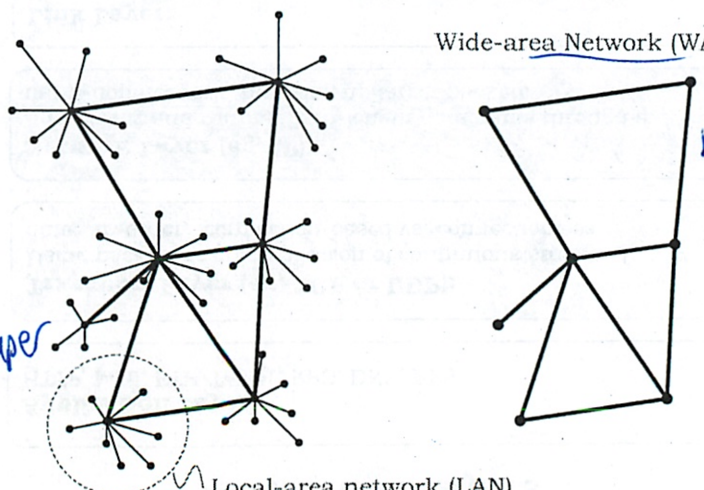


FIG. 1 — Centralized, Decentralized and Distributed Networks

← much better for attacks P2P

The pioneering research of Paul Baran in the 1960s, who envisioned a communications network that would survive a major enemy attack. The sketch shows three different network topologies described in his RAND Memorandum, "On Distributed Communications: 1. Introduction to Distributed Communications Network" (August 1964). The distributed network structure was judged to offer the best survivability.

Modern Networks: LANs + WANs



Cheaper

more redundant

Works at lots of diff scales

Sharing the internetwork links

Consider a single node in our wide-area network:

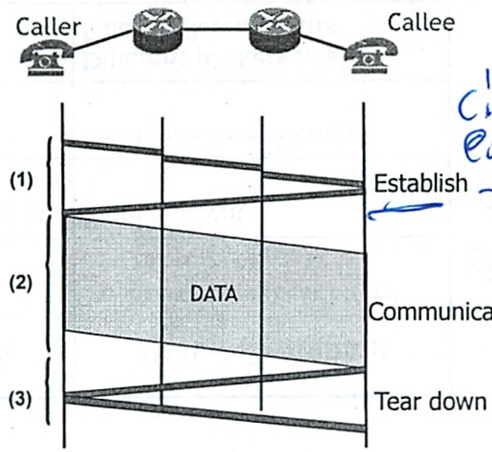
We have many (low-bandwidth) application-level connections that need to mapped onto a small number of (high-bandwidth) channels. Some originate/terminate at the current node, some are just passing through on their way to another node.

How should we share the link between all the connections?

Circuit switching (isochronous) - phone network
Packet switching (asynchronous) - no guarentees
- early engineers never got this

Circuit Switching

- First establish a *circuit* between end points
  - E.g., done when you dial a phone number
  - Message propagates from caller toward callee, establishing some state in each switch
- Then, ends send data ("talk") to each other
- After call, *tear down* (close) circuit
  - Remove state

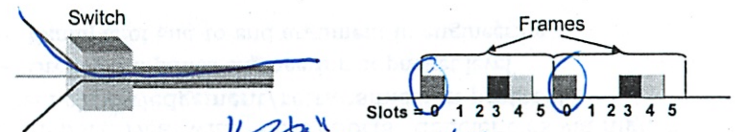


circuit been established -TDMA

to do

at some points all circuit are busy

Multiplexing/Demultiplexing



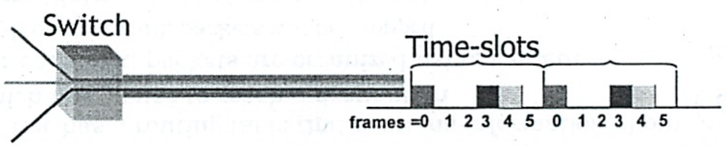
~~"epoch"~~ "epoch"

One sharing technique: time-division multiplexing (TDM)

- Time divided into frames and frames divided into slots
  - Number of slots = number of concurrent conversations
- Relative slot position inside a frame determines which conversation the data belongs to
  - E.g., slot 0 belongs to the red conversation
  - Mapping established during setup, removed at tear down
- Forwarding step at switch: consult table

have a slot dedicated to this call

TDM Shares Link Equally, But Has Limitations



- Suppose link capacity is C bits/sec
- Each communication requires R bits/sec
- #frames in one "epoch" (one frame per communication) = C/R
- Maximum number of concurrent communications is C/R
- What happens if we have more than C/R communications?
- What happens if the communication sends less/more than R bits/sec?

→ Design is unsuitable when traffic arrives in *bursts*

can't have > C/R

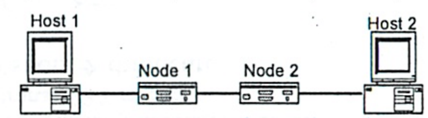
-slot has been reserved

or under utilization

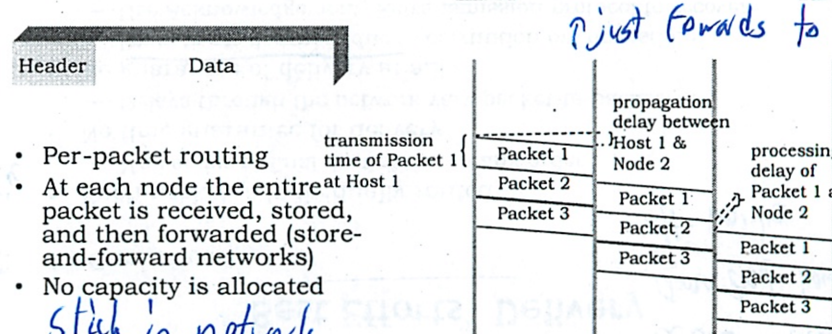
circuit Esp not good w/ data → so

Packet Switching

- Used in the Internet
- Data is sent in packets (header contains control info, e.g., source and destination addresses)



just forwards to its best guess



- Per-packet routing
- At each node the entire packet is received, stored, and then forwarded (store-and-forward networks)
- No capacity is allocated

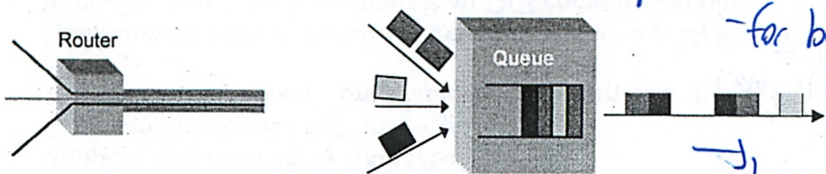
stick in network

equivilant of Post box

Packet Switching: Multiplexing/Demultiplexing

↓ have room to accept +queue packet

-for bursty traffic



→ have enough capacity to empty queue

- Router has a routing table that contains information about which link to use to reach a destination
- For each link, packets are organized using a queue
  - If queue is full, packets will be dropped
- Demultiplex using information in packet header
  - Header has destination

"Best Efforts" Delivery

Seems flaky amazed how well it works

No Guarantees!

- Each packet is individually routed
  - May arrive at final destination in any order
- No time guarantee for delivery
  - Delays through the network vary packet-to-packet
- No guarantee of delivery at all!
  - Packets get dropped (due to corruption or congestion)
  - Use Acknowledgement/Retransmission protocol to recover
    - How to determine when to retransmit? Timeout?
- If packet is re-transmitted too soon → duplicate

Sounds like the US Mail!

Comparison of Two Techniques

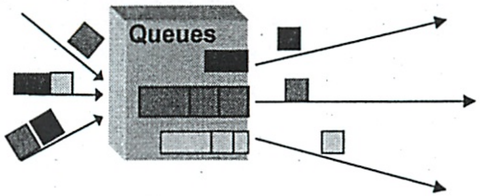
Circuit switching	Packet Switching
Guaranteed capacity	No guarantees (best effort)
Capacity is wasted if data is bursty	More efficient
Before sending data establishes a path	Send data immediately
All data in a single flow follow one path	Different packets might follow different paths
No reordering; constant delay; no dropped packets	Packets may be reordered, delayed, or dropped

even comm networks are much more packet

We'll Explore Packet Switching

- Commonly used for data networks
  - Data traffic is "bursty", fixed BW allocation isn't best
  - makes most efficient use of communications link
- Routing: choose paths through network
  - "best" route changes dynamically
  - To preserve scalability prefer distributed, decentralized management of routing choices
- Transport: Deal with "best efforts" deficiencies via higher level acknowledgement/retransmission protocol
  - Greatly simplifies engineering at packet level
  - Example of end-to-end argument in engineering

Queues are Essential

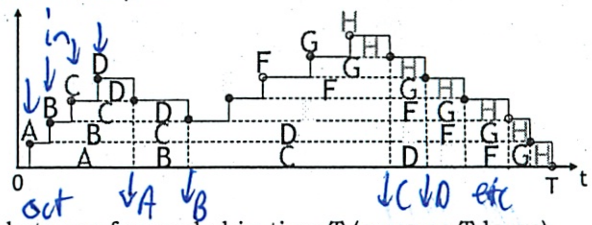


- Queues manage packets between arrival and departure
- They are a "necessary evil"
  - Needed to absorb bursts
  - But they add delay by making packets wait until link is available
- So they shouldn't be too big

Little's Law

FIFO

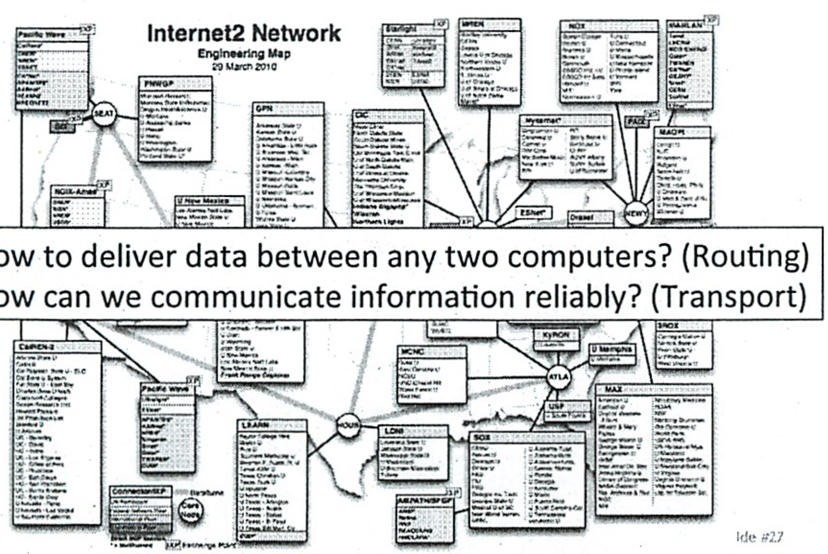
n(t) = # pkts at time t in queue



- P packets are forwarded in time T (assume T large)
- Rate = λ = P/T
- Let A = area under the n(t) curve from 0 to T
- Mean number of packets in queue = N = A/T over time
- A is aggregate delay weighted by each packet's time in queue. So, mean delay D per packet = A/P
- Therefore, **N = λ D** ← Little's Law # = rate · delay
- For a given link rate, increasing queue size increases delay

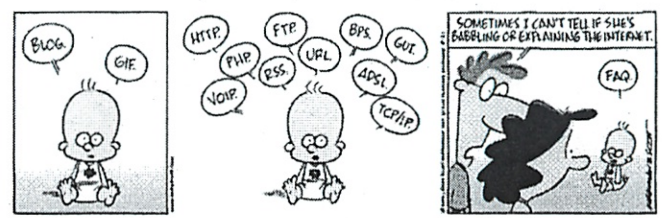
N = queue size

Multi-Hop Packet Networks



How to deliver data between any two computers? (Routing)
How can we communicate information reliably? (Transport)

don't make your queue too big!
- stuff taking so long
- so add more!
- instead drop quickly



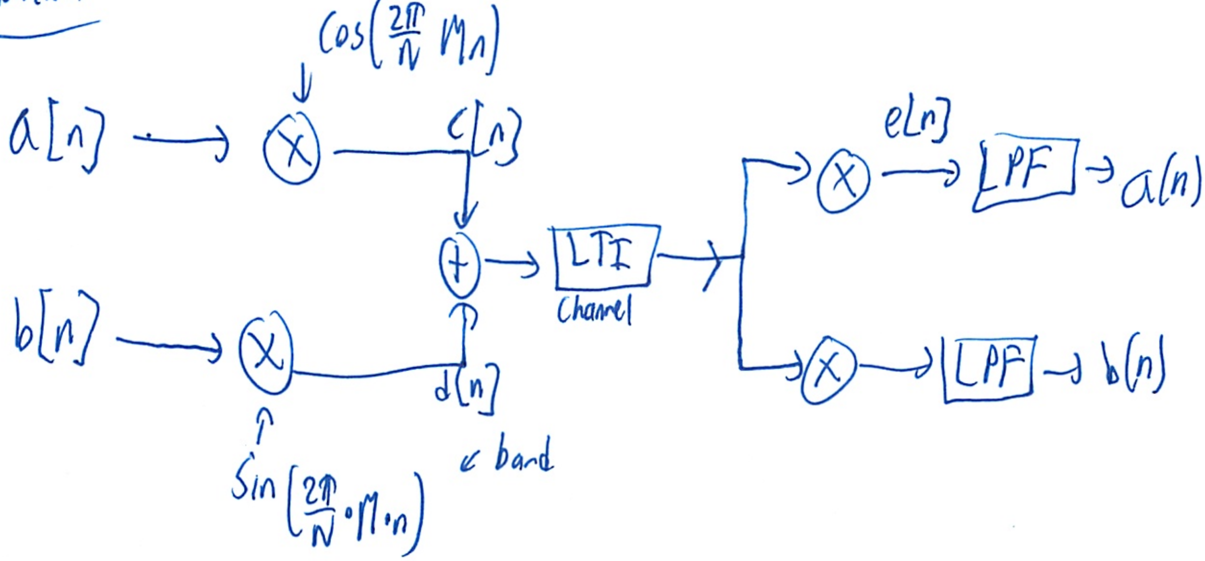
Baby Blues (Kirkman/Scott)

6.02 Recitation

4/14

Today: Modulation and Little's Law

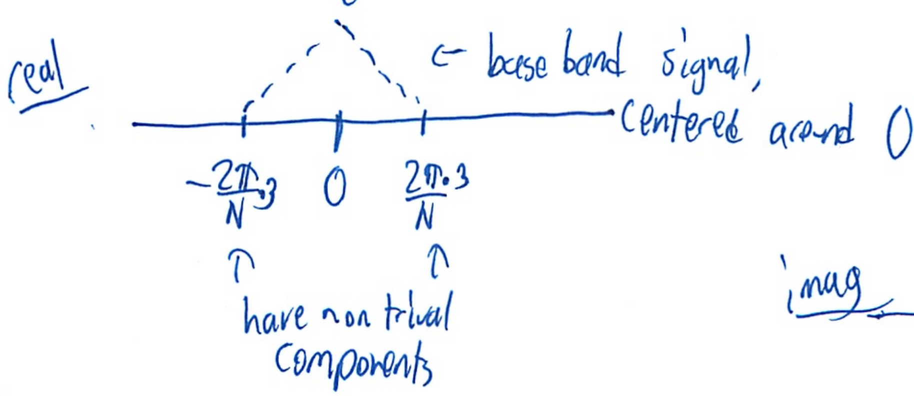
Modulation

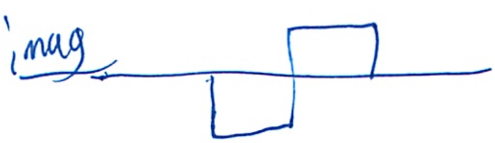


* about transmitting multiple sequences on same channel at same time at different freq *

Say $a[n] = \cos\left(\frac{2\pi}{N} \cdot 3 \cdot n\right)$

$$= \frac{1}{2} \cdot \exp\left(j \cdot \frac{2\pi}{N} \cdot 3n\right) + \frac{1}{2} \exp\left(-j \cdot \frac{2\pi}{N} \cdot 3n\right)$$





Multiply by carrier
- have 2, recenter 0

②

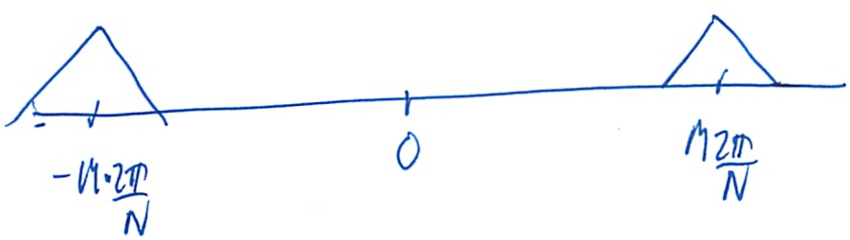
So now $c[n]$ is

$$c[n] = a[n] \cdot \cos\left(\frac{2\pi}{N} \cdot M \cdot n\right)$$

$$= \frac{1}{4}\left[\left(\exp\left(j \cdot \frac{2\pi}{N} \cdot 3 \cdot n\right) + \exp\left(-j \cdot \frac{2\pi}{N} \cdot 3 \cdot n\right)\right)\left(\exp\left(j \cdot \frac{2\pi}{N} \cdot M \cdot n\right) + \exp\left(-j \cdot \frac{2\pi}{N} \cdot M \cdot n\right)\right)\right]$$

$$= \frac{1}{4}\left[\exp\left(j \cdot \frac{2\pi}{N}(M+3)\right) + \exp\left(j \cdot \frac{2\pi}{N}(M-3)\, n\right) + \exp\left(j \cdot \frac{2\pi}{N}(-M+3)\, n\right) + \exp\left(j \cdot \frac{2\pi}{N}(-M-3)\, n\right)\right]$$

So again, we now have



Now have 4 freq components

Transmit over LTI channel

Now time to demodulate at reciever

$$e[n] = c[n] \cdot \cos\left(\frac{2\pi}{N} \cdot M \cdot n\right)$$

$$= c[n] \cdot \left(\frac{1}{2} \exp\left(j \cdot \frac{2\pi}{N} \cdot M \cdot n\right) + \exp\left(-j \cdot \frac{2\pi}{N} \cdot M \cdot n\right)\right)$$

(3)

Just write what freq come out

$(2m+3), (2m-3), 3, -3$

$3, -3, (-2m+3), (-2m-3)$

We started w/ $-3, 3$ each w/ half, half

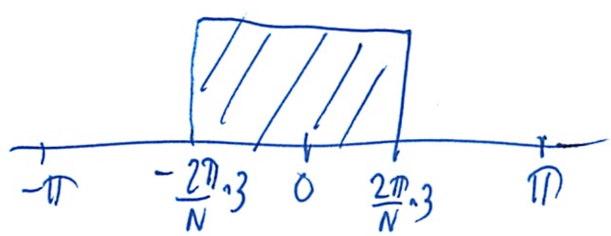
Now we are back w/ $-3, 3$ and a bunch of other stuff - each are $\frac{1}{8}$ weight

So 3 totals to $1/4$ weight

-3 $1/4$

Get rid of the other ones to recover the $3, -3$

Need a Low Pass Filter (LPF) that keeps



You are given a band

Use cos to transmit in 2 places

- multiply in 2 places
- doing amplitude modulation
  - can't do complex modulation
  - could rotate in optics - could do

(4)

Phase shift?

Then $a[n]$ is another $a[n+\sigma]$

↑ 1 period

So $a[n] = a[n+\sigma]$

$= \cos(f(n-\sigma))$

$= \cos(fn - f\sigma)$

↑ phase shift

$\frac{1}{2}\Big(\exp(\sigma - f) \cdot \exp(-f\sigma j) + \exp(-j f) \cdot \exp(+\sigma f j)\Big)$

So make signal slightly weaker

- less intense

In trouble if $\sigma = \pi/2$

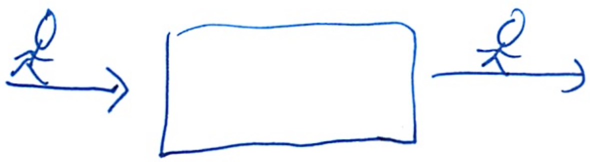
Don't want top term to $= 0$

Chris covered quickly in class

(5)

Little's Law

- Uses integration
  - 2 ways to integrate, both should be =
  - this is what it should be



→ each time someone left, ask how long how long wait tent

λ - rate at which people served sucessfully

D - delay

Q - avg # people in restarant

- look every minute or so

$$\boxed{Q = \lambda \cdot D}$$

Can be operationally useful

Say ammusement park

$Q = \lambda D$

Q: 80 ~~[illegible]~~ people in line, 20 on ride

λ: ride does 20 people every 2 min

D: So what is wait time? $= \dfrac{100 \text{ people}}{10 \text{ people/min}} = 10$ min

6

But this is wait + ride time

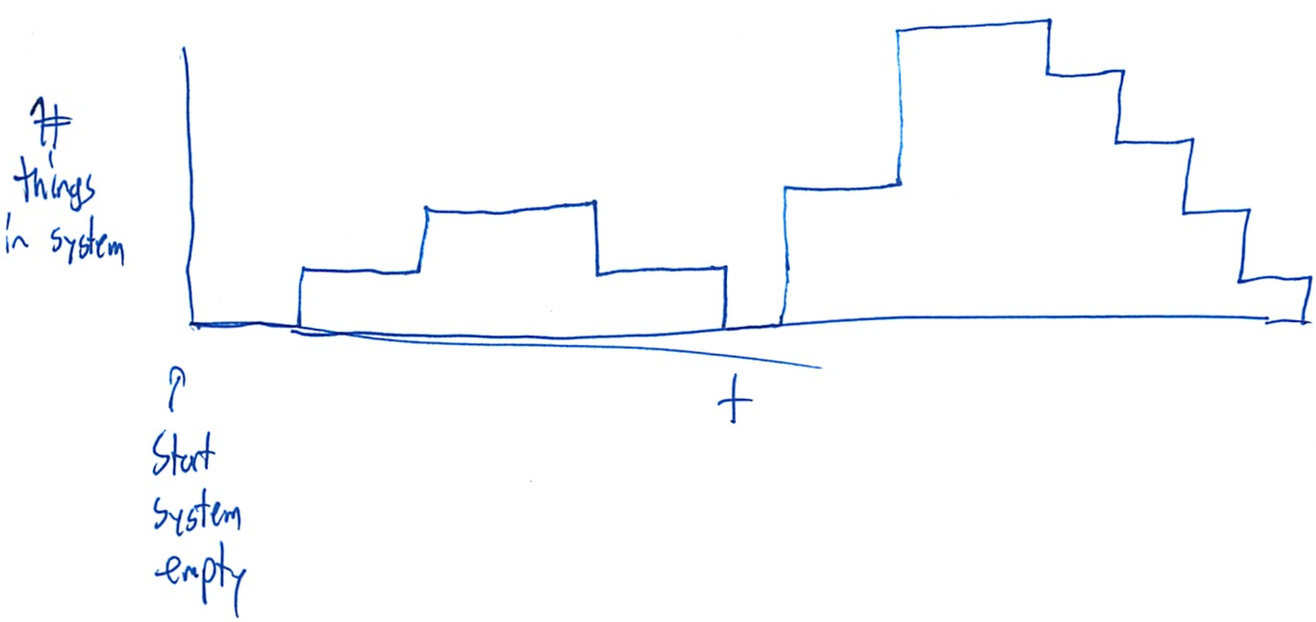
Just wait time

$$Q = \lambda \cdot D$$

$Q = 80$, $\lambda = 10$ p/m, so $D = \frac{80}{10} = 8$ min

which fits previous result

<u>Proving</u>



So compute area $A(T)$

- could add horizontally or vertically

Want $\frac{A(T)}{T}$ to find Q

$$= \frac{1}{T}\left[\int_0^T A(t)\right] = Q \qquad \text{vertical way}$$

7

But could also caculate horizontally

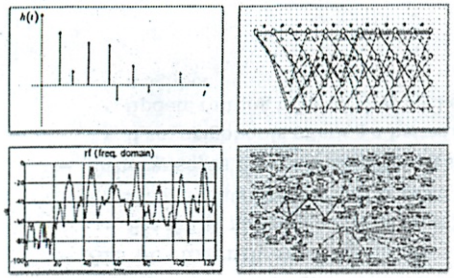
- 1 person waited for certain time
- Cumulative waiting

$\sum$ everones delay

$$\frac{A(T)}{T} = \frac{N(T)}{T} \frac{(\text{total delay})}{N(T)}$$

$$Q = \hat{\lambda} \cdot \hat{D}$$

4/20



INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Spring 2011
Lecture #19

- addressing, forwarding, routing
- liveness, advertisements, integration
- distance-vector routing
- routing loops, counting to infinity

Quiz 2 back today

Class Grade

A 40%

B 40%

C ~20%

Q1+Q2 = 150

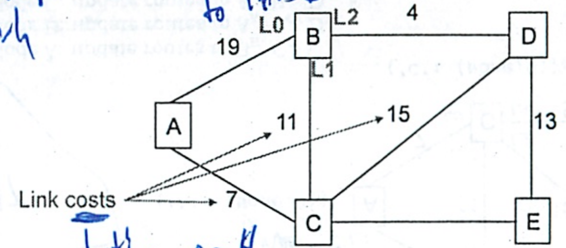
How do packets find their way through network?

Multi-hop network

- Switching

The Problem: Finding Paths

local name to links



time or $

Each packet takes own path

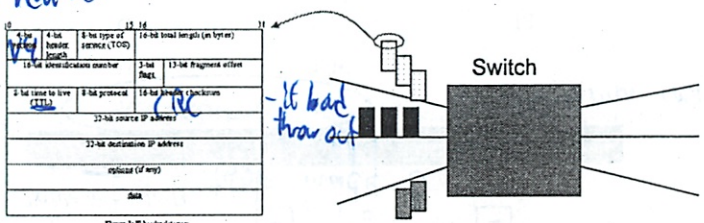
no guarantee will finish

- Addressing (how to name nodes?)
  - Unique identifier for global addressing IP addresses
  - Link name for neighbors
- Forwarding (how does a switch process a packet?)
- Routing (building and updating data structures to ensure that forwarding works) who is connected to whom
- Functions of the *network layer*

What do I do w/ packet?

Forwarding

IP header



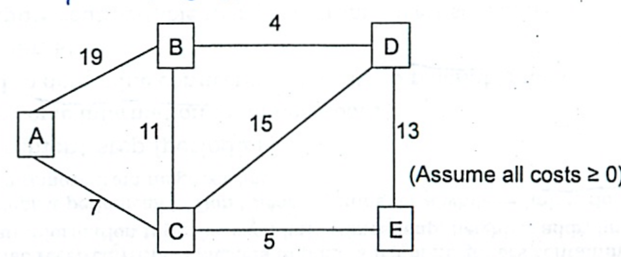
UDP packet → "datagram"

- Core function is conceptually simple
  - `lookup(dst_addr)` in routing table returns *route* (i.e., *outgoing link*) for packet
  - `enqueue(packet, link_queue)` put in queue
  - `send(packet)` along outgoing link
- And do some bookkeeping before enqueue
  - Decrement hop limit (TTL); if 0, discard packet - something wrong
  - Recalculate checksum (in IP, header checksum)

look up in routing table

Shortest Path Routing

really Min Cost



- Each node wants to find the path with *minimum total cost* to other nodes
  - We use the term "shortest path" even though we're interested in min cost (and not min #hops)
- Several possible distributed approaches
  - Vector protocols, esp. *distance vector* (DV) from neighbors today
  - *Link-state* protocols (LS)

need a way that is not n^2

Mon from each node

no single pts of failure

not scalable

6.042 Combinatorics

Routing Table Structure

At B
The best place to send each packet

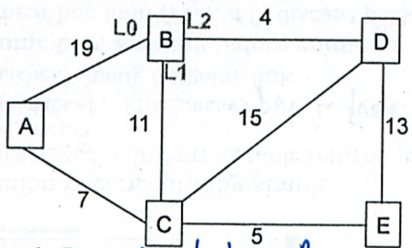


Table @ node B table inside B

Destination			Cost
A	ROUTE	L1	18
B		'Self'	0
C		L1	11
D		L2	4
E		L1	16

← min cost - not shortest visually here

just to next hop

"travel 1 hop + ask for directions"

then at self: "directions: you're here!"

Distributed Routing: A Common Plan

- Determining live neighbors
  - Common to both DV and LS protocols
  - HELLO protocol (periodic) 1 hop packet
    - Send HELLO packet to each neighbor to let them know who's at the end of their outgoing links who are we connected to
    - Use received HELLO packets to build a list of neighbors containing an information tuple for each link: (timestamp, neighbor addr, link)
    - Repeat periodically. Don't hear anything for a while → link is down, so remove from neighbor list.
- Advertisement step (periodic)
  - Send some information to all neighbors
  - Used to determine connectivity & costs to reachable nodes
- Integration step
  - Compute routing table using info from advertisements
  - Dealing with stale data

1. Who is alive?
2. Send out my routing info
3. Recieve other's advertisements

Distance-Vector Routing

- DV advertisement all the nodes that node can reach w/ cost
  - Send info from routing table entries: (dest, cost)
  - Initially just (self,0)
- DV integration step [Bellman-Ford]
  - For each (dest,cost) entry in neighbor's advertisement
    - Account for cost to reach neighbor: (dest,my_cost)
    - my_cost = cost_in_advertisement + link_cost
  - Are we currently sending packets for dest to this neighbor?
    - See if link matches what we have in routing table
    - If so, update cost in routing table to be my_cost
  - Otherwise, is my_cost smaller than existing route?
    - If so, neighbor is offering a better deal! Use it...
    - update routing table so that packets for dest are sent to this neighbor

You advertise
MIT 17
Tufts 44

Me → You = 10

So for me I can do MIT 27
Tufts 54
via you

Look in table - is this a better deal for each dest.

DV Example: round 1

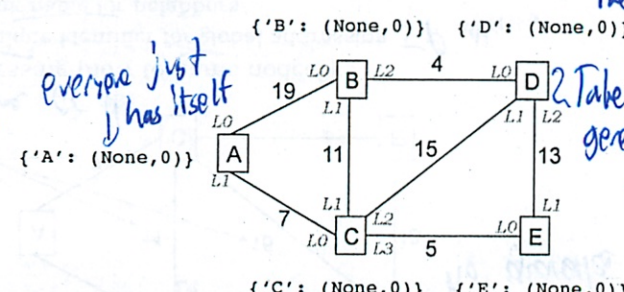
Get routes for A

1 A sends to everyone connected to it the cost of links

2 Takes multiple generations of ads

...

everyone just has itself



Node A: update routes to B_B, C_C

Node B: update routes to A_A, C_C, D_D

Node C: update routes to A_A, B_B, D_D, E_E

Node D: update routes to B_B, C_C, E_E

Node E: update routes to C_C, D_D

Subscript indicates node that gave better route

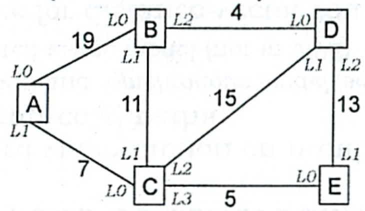
if even pick 1

Costs don't have to be symetrical

DV Example: round 2

{'A': (L0,19),
'B': (None,0),
'C': (L1,11),
'D': (L2,4)
}

{'B': (L0,4),
'C': (L1,15),
'D': (None,0),
'E': (L2,13)
}



{'A': (None,0),
'B': (L0,19),
'C': (L1,7)
}

{'A': (L0,7),
'B': (L1,11),
'C': (None,0),
'D': (L2,15),
'E': (L3,5)
}

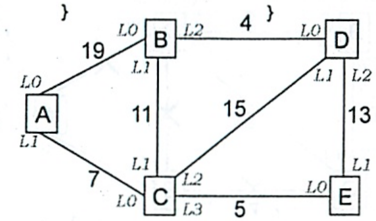
{'C': (L0,5),
'D': (L1,13),
'E': (None,0)
}

Node A: update routes to B_C, D_C, E_C
Node B: update routes to A_C, E_C
Node C: no updates
Node D: update routes to A_C
Node E: update routes to A_C, B_C

DV Example: round 3

{'A': (L1,18),
'B': (None,0),
'C': (L1,11),
'D': (L2,4),
'E': (L1,16)
}

{'A': (L1,22),
'B': (L0,4),
'C': (L1,15),
'D': (None,0),
'E': (L2,13)
}



{'A': (None,0),
'B': (L1,18),
'C': (L1,7),
'D': (L1,22),
'E': (L1,12)
}

Node A: no updates
Node B: no updates
Node C: no updates
Node D: no updates
Node E: no updates

{'A': (L0,7),
'B': (L1,11),
'C': (None,0),
'D': (L2,15),
'E': (L3,5)
}

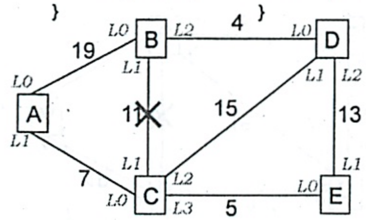
{'A': (L0,12),
'B': (L0,16),
'C': (L0,5),
'D': (L1,13),
'E': (None,0)
}

DV Example: Break a Link

Re think the link - now (i, ∞)
- have to re-hello
- in next ads incoming
- replaces (i, ∞)
- repeat

~~{'A': (L1,18),~~
'B': (None,0),
~~'C': (L1,11),~~
'D': (L2,4),
~~'E': (L1,16)~~
}

{'A': (L1,22),
'B': (L0,4),
'C': (L1,15),
'D': (None,0),
'E': (L2,13)
}



{'A': (None,0),
'B': (L1,18),
'C': (L1,7),
'D': (L1,22),
'E': (L1,12)
}

{'A': (L0,7),
~~'B': (L1,11),~~
'C': (None,0),
'D': (L2,15),
'E': (L3,5)
}

{'A': (L0,12),
'B': (L0,16),
'C': (L0,5),
'D': (L1,13),
'E': (None,0)
}

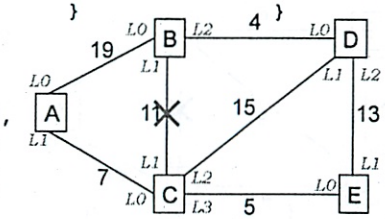
When link breaks: eliminate routes that use that link.

DV Example: round 4

∞ costs propagate

{'A': (None,∞),
'B': (None,0),
'C': (None,∞),
'D': (L2,4),
'E': (None,∞)
}

{'A': (L1,22),
'B': (L0,4),
'C': (L1,15),
'D': (None,0),
'E': (L2,13)
}



{'A': (None,0),
'B': (L1,18),
'C': (L1,7),
'D': (L1,22),
'E': (L1,12)
}

Node A: update cost to B_C
Node B: update routes to A_A, C_D, E_D
Node C: update routes to B_D
Node D: no updates
Node E: update routes to B_D

{'A': (L0,7),
'B': (None,∞),
'C': (None,0),
'D': (L2,15),
'E': (L3,5)
}

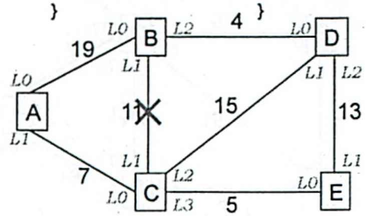
{'A': (L0,12),
'B': (L0,16),
'C': (L0,5),
'D': (L1,13),
'E': (None,0)
}

DV Example: round 5

```
{'A': (L0,19),     {'A': (L1,22),
 'B': (None,0),     'B': (L0,4),
 'C': (L2,19),      'C': (L1,15),
 'D': (L2,4),       'D': (None,0),
 'E': (L2,17)       'E': (L2,13)
}                  }
```

Update cost

```
{'A': (None,0),
 'B': (L1, ∞),
 'C': (L1,7),
 'D': (L1,22),
 'E': (L1,12)
}
```



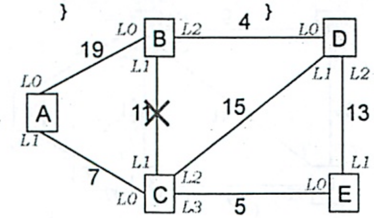

```
{'A': (L0,7),      {'A': (L0,12),
 'B': (L2,19),      'B': (L1,17),
 'C': (None,0),     'C': (L0,5),
 'D': (L2,15),      'D': (L1,13),
 'E': (L3,5)        'E': (None,0)
}                  }
```

Node A: update route to B_B
Node B: no updates
Node C: no updates
Node D: no updates
Node E: no updates

DV Example: final state

```
{'A': (L0,19),     {'A': (L1,22),
 'B': (None,0),     'B': (L0,4),
 'C': (L2,19),      'C': (L1,15),
 'D': (L2,4),       'D': (None,0),
 'E': (L2,17)       'E': (L2,13)
}                  }
```

```
{'A': (None,0),
 'B': (L0,19),
 'C': (L1,7),
 'D': (L1,22),
 'E': (L1,12)
}
```




```
{'A': (L0,7),      {'A': (L0,12),
 'B': (L2,19),      'B': (L1,17),
 'C': (None,0),     'C': (L0,5),
 'D': (L2,15),      'D': (L1,13),
 'E': (L3,5)        'E': (None,0)
}                  }
```

Node A: no updates
Node B: no updates
Node C: no updates
Node D: no updates
Node E: no updates

Correctness & Performance

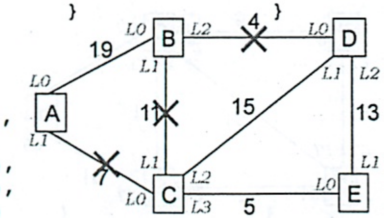
- Optimal substructure property fundamental to correctness of both Bellman-Ford and Dijkstra's shortest path algorithms
  - ***Suppose shortest path from X to Y goes through Z. Then, the sub-path from X to Z must be a shortest path.***
- Proof of Bellman-Ford via induction on number of walks on shortest (min-cost) paths
  - Easy when all costs > 0 and *synchronous model* (see notes)
  - Harder with distributed async model (not in 6.02)
- How long does it take for distance-vector routing protocol to *converge*?
  - Time proportional to largest number of hops considering all the min-cost paths

Partitioning the Network

if network breaks in 2

```
{'A': (L0,19),     {'A': (L1,22),
 'B': (None,0),     'B': (None,∞),
 'C': (None,∞),     'C': (L1,15),
 'D': (None,∞),     'D': (None,0),
 'E': (None,∞)      'E': (L2,13)
}                  }
```

```
{'A': (None,0),
 'B': (L0,19),
 'C': (None,∞),
 'D': (None,∞),
 'E': (None,∞)
}
```




```
{'A': (None,∞),    {'A': (L0,12),
 'B': (L2,19),      'B': (L1,17),
 'C': (None,0),     'C': (L0,5),
 'D': (L2,15),      'D': (L1,13),
 'E': (L3,5)        'E': (None,0)
}                  }
```

Node A: delete routes to C, D, E
Node B: delete routes to C, D, E
Node C: update routes to A_E, B_E
Node D: update routes to A_E, B_E
Node E: update route to A_D, cost to B_D

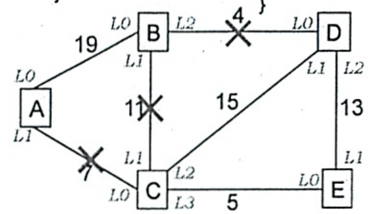
C gets an initial offer from E for A
very simple!

DV Example: round 6

```
{'A': (L0,19),
 'B': (None,0)
}
```

```
{'A': (L2,25),
 'B': (L2,30),
 'C': (L1,15),
 'D': (None,0),
 'E': (L2,13)
}
```

```
{'A': (None,0),
 'B': (L0,19)
}
```




```
{'A': (L3,17),
 'B': (L3,22),
 'C': (None,0),
 'D': (L2,15),
 'E': (L3,5)
}
```

```
{'A': (L1,35),
 'B': (None,∞),
 'C': (L0,5),
 'D': (L1,13),
 'E': (None,0)
}
```

Node A: no updates
Node B: no updates
Node C: update costs to A_E, B_E
Node D: update route to A_C, cost to B_E
Node E: update routes to A_C, B_C

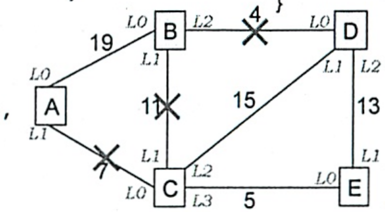
get a routing loop to ∞

Counting to Infinity

```
{'A': (L0,19),
 'B': (None,0)
}
```

```
{'A': (L2,48),
 'B': (None,∞),
 'C': (L1,15),
 'D': (None,0),
 'E': (L2,13)
}
```

```
{'A': (None,0),
 'B': (L0,19)
}
```



Nodes C, D, and E each update their costs in response to earlier updates by neighbors. Costs spiral upwards towards ∞!

remove route when cost reaches self.INFINITY

```
{'A': (L3,40),
 'B': (None,∞),
 'C': (None,0),
 'D': (L2,15),
 'E': (L3,5)
}
```

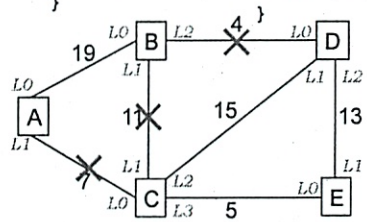
```
{'A': (L0,22),
 'B': (L0,27),
 'C': (L0,5),
 'D': (L1,13),
 'E': (None,0)
}
```


Routing Loop!

```
{'A': (L0,19),
 'B': (None,0)
}
```

```
{'A': (L2,48),
 'B': (None,∞),
 'C': (L1,15),
 'D': (None,0),
 'E': (L2,13)
}
```

```
{'A': (None,0),
 'B': (L0,19)
}
```



Suppose E sends a packet to A:

- E forwards to D
- D forwards to E
- ... repeat ...
- Drop packet when TTL is decremented to 0

```
{'A': (L3,40),
 'B': (None,∞),
 'C': (None,0),
 'D': (L2,15),
 'E': (L3,5)
}
```

```
{'A': (L0,22),
 'B': (L0,27),
 'C': (L0,5),
 'D': (L1,13),
 'E': (None,0)
}
```

So have a TTL so drop eventually

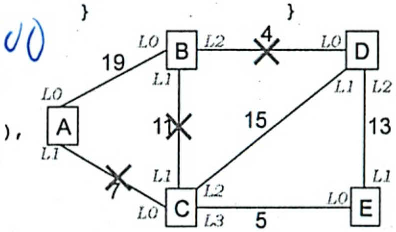
Eventual Final State

eventually hellos get to 100,000 and that is ∞ so forget it

```
{'A': (L0,19),
 'B': (None,0)
}
```

```
{'C': (L1,15),
 'D': (None,0),
 'E': (L2,13)
}
```

```
{'A': (None,0),
 'B': (L0,19)
}
```



Eventually all the unreachable nodes are removed from routing table and all routing loops are resolved.

```
{'C': (None,0),
 'D': (L2,15),
 'E': (L3,5)
}
```

```
{'C': (L0,5),
 'D': (L1,13),
 'E': (None,0)
}
```

Review

4/20

Modulation

- vary waveform (aka carrier signal)
  - I think also baseband
- w/ respect to modulating signal
- like varying music's volume, timing, pitch
  - amp, phase, freq
- final = passband

- can do amplitude modulation AM
  - freq " FM

Never went over in recitation