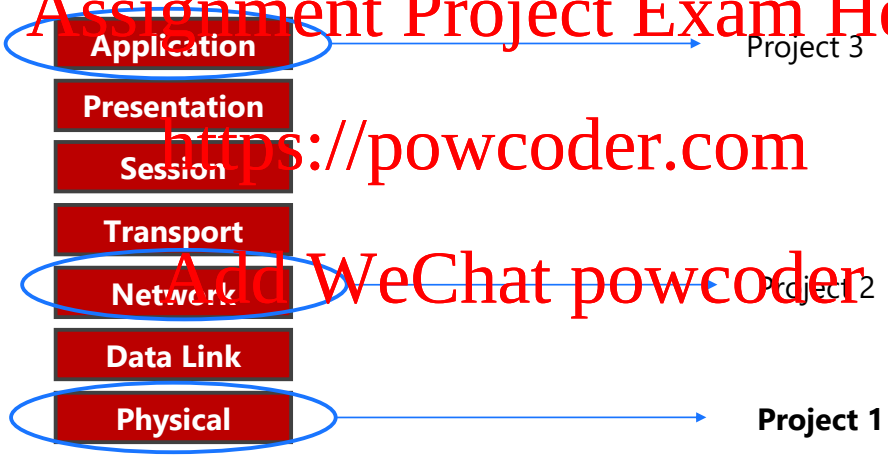
Carnegie Mellon University

Recitation 1: Exploring the WiFi Physical Layer

FEBRUARY 13, 2020

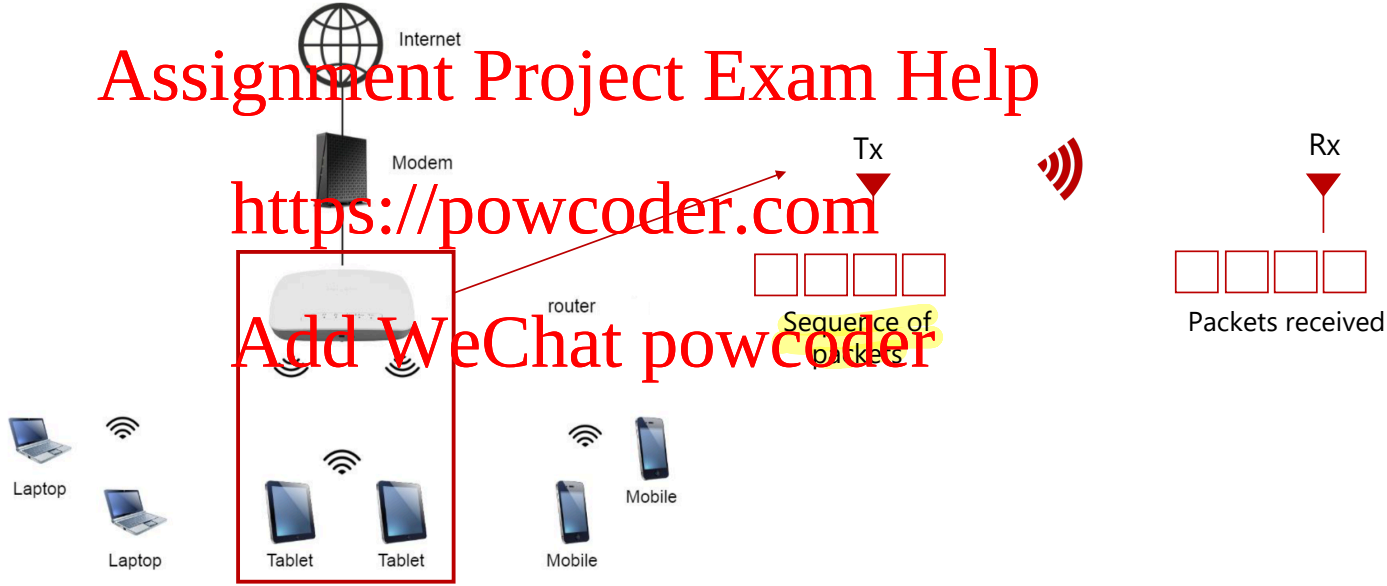
Atul Bansal

Open Systems Interconnection Model

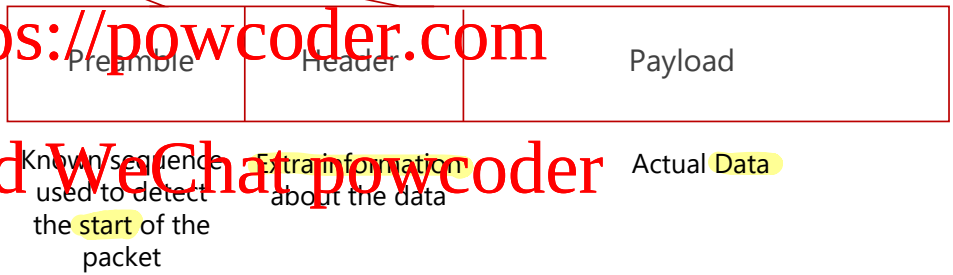


Project 1: Physical Layer(WiFi)

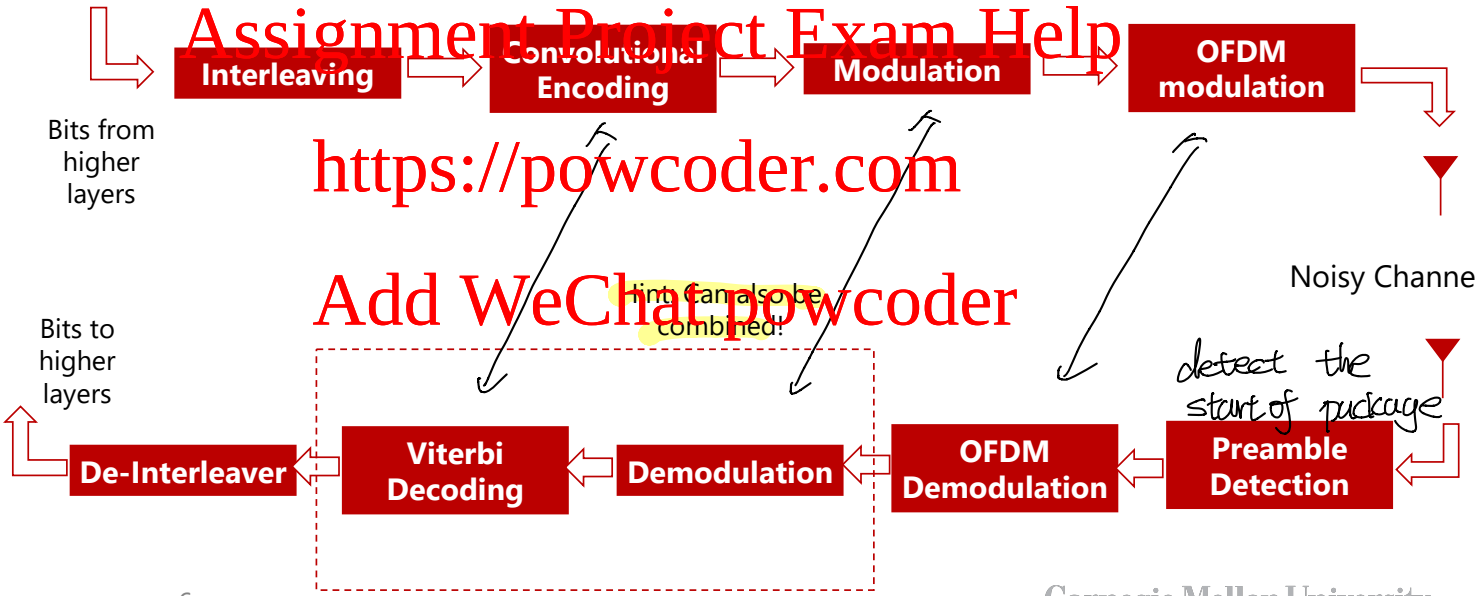
Project 1: Physical Layer(WiFi)



WiFi Packet



How WiFi packets are created?



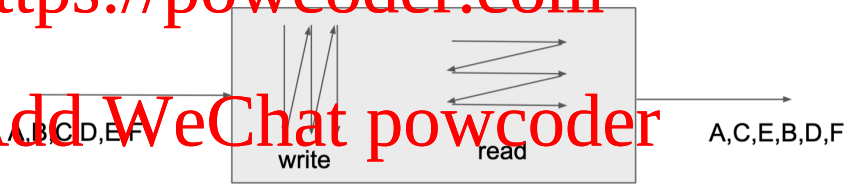
How WiFi packets are created?

Level 1: Interleaving

Input bits are rearranged by a well-known permutation pattern

Interleaver: Rows = 2, Columns = 3

example:



$$\begin{bmatrix} A & C & E \\ B & D & F \end{bmatrix}$$

Resilient to burst errors

Level 2: Convolutional Coding

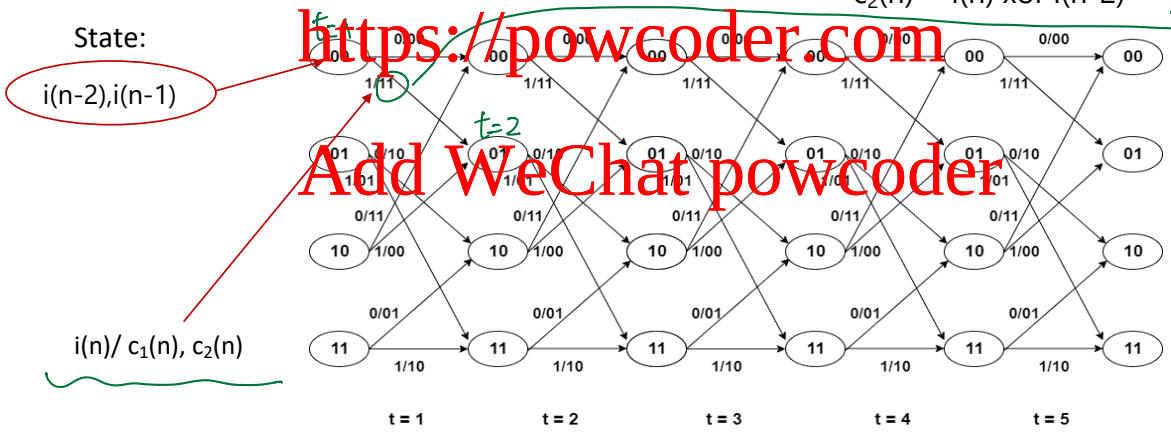
Encode the input bits using a 'special' polynomial to add redundancy.

For this project, we use a 1/2 rate encoder with G(7,5) generator polynomial

Input Bits: i(n), i(n-1), i(n-2)

Output Bits: $c_1(n) = i(n) \text{ xor } i(n-1) \text{ xor } i(n-2)$

$c_2(n) = i(n) \text{ xor } i(n-2)$



example when t=1 i(n-2), i(n-1) = 0,0

i(n) = 1

Then $c_1 = 1 \text{ xor } 0 \text{ xor } 0 = 1$

$c_2 = 1 \text{ xor } 0 = 1$

∴ output = 11

⇓

when t=2

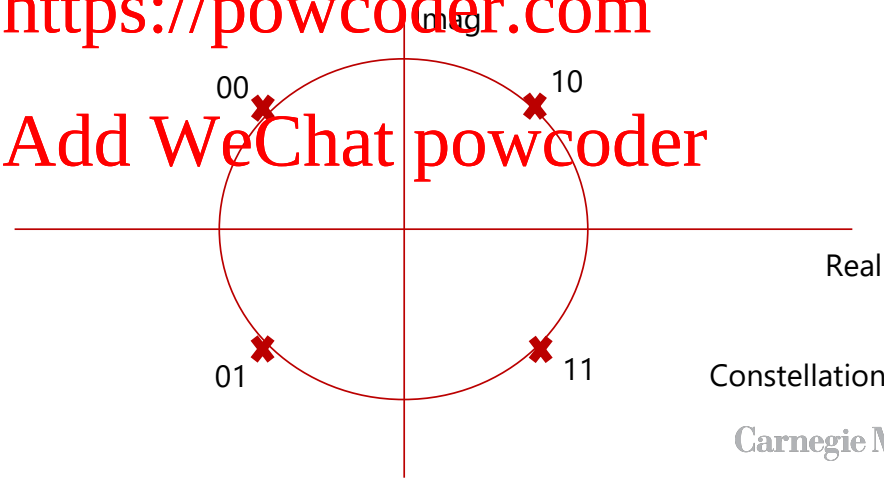
i(n-2) = 0

i(n-1) = 1

Level 3: Modulation

Coded Bits to Symbol Mapping: BPSK, QPSK, 4-QAM

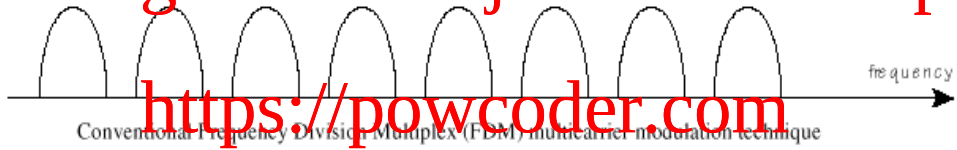
Conveying information by changing the signal and represent them in complex domain.
We use 4-QAM in this project.



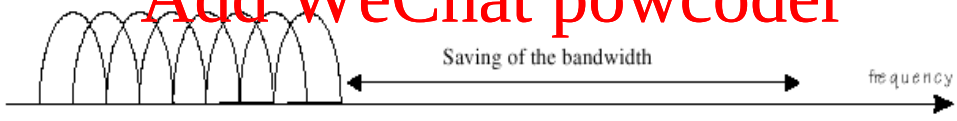
Constellation Diagram

Level 4: Orthogonal Frequency Division Multiplexing (OFDM)

Divide the available bandwidth into multiple orthogonal subcarriers



Conventional Frequency Division Multiplex (FDM) multicarrier modulation technique



Orthogonal Frequency Division Multiplex (OFDM) multicarrier modulation technique

sin + cos

Some excellent properties: spectral efficiency, robustness to fading and inter-symbol interference

Implemented using IFFT

Level 5: Padding and Noise

Random number of padding bits are added before and after the information symbols

Used to model wireless channel.

Detect beginning and end of info symbols (useful symbols)

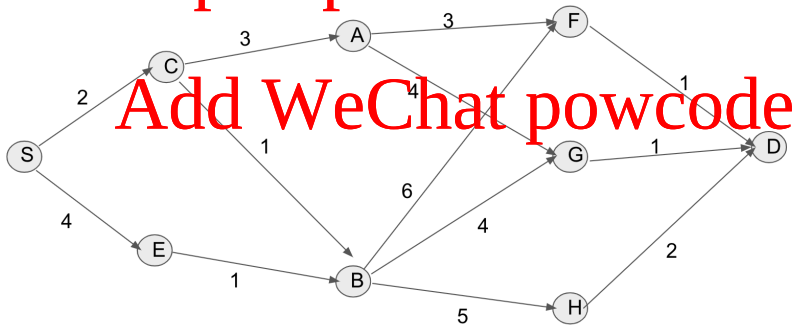
Beginning: Preamble
End: ??

Need to implement

Viterbi Decoding

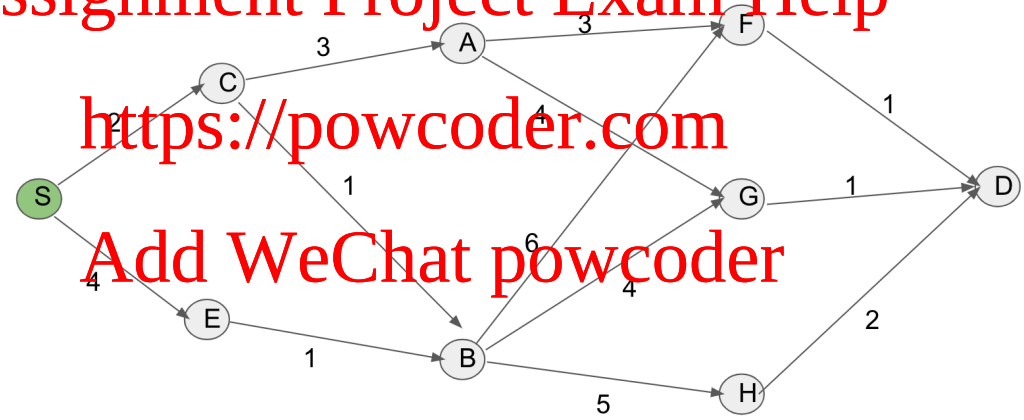
Used at WiFi receivers for inverse convolutional coding

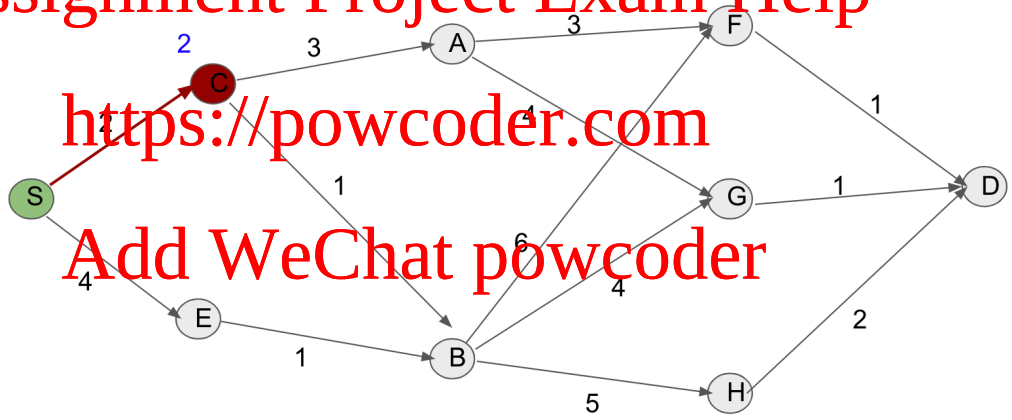
Before that, consider this problem

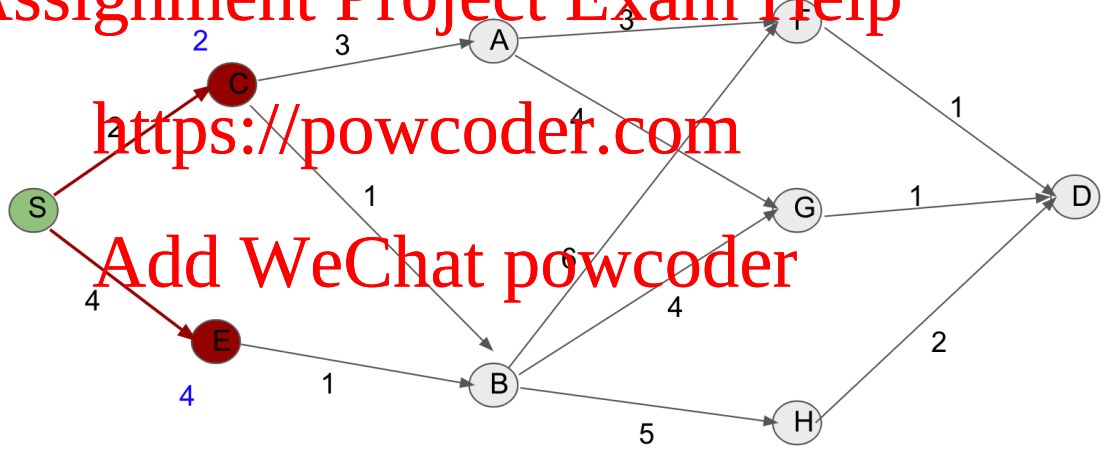


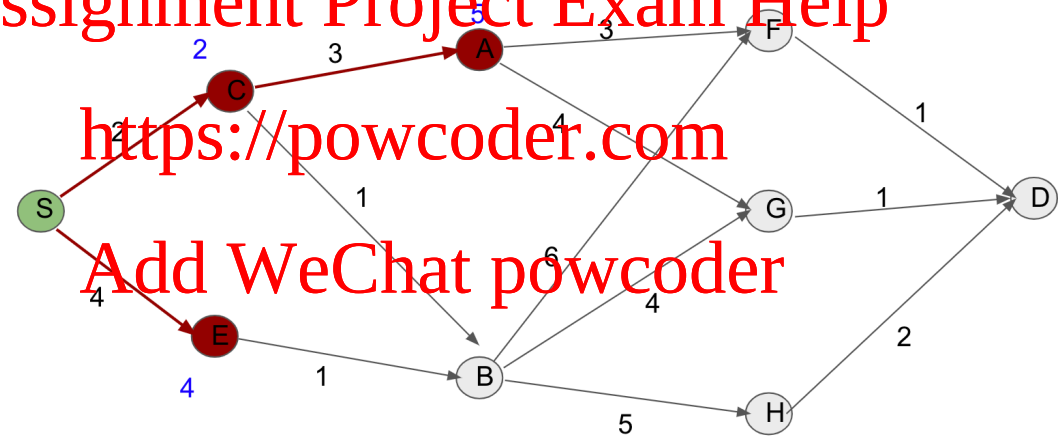
Find shortest distance from S to D?

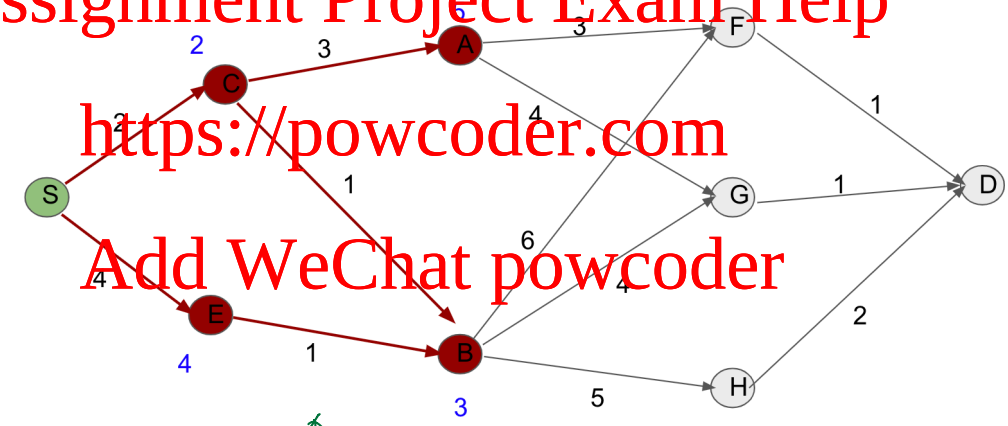
Use Dynamic Programming/Dijkstra's algorithm



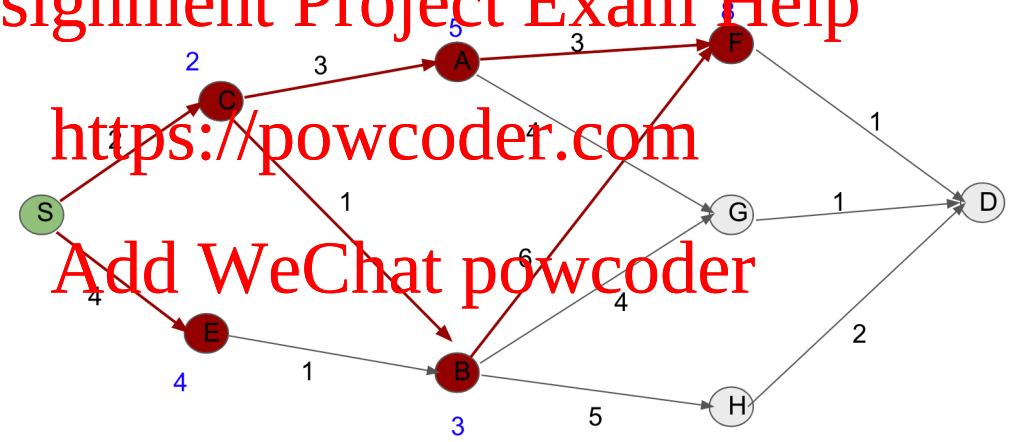


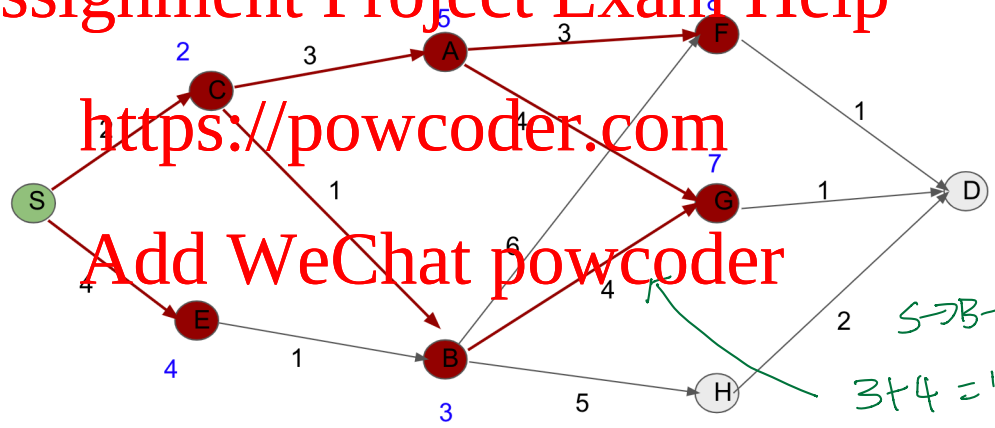


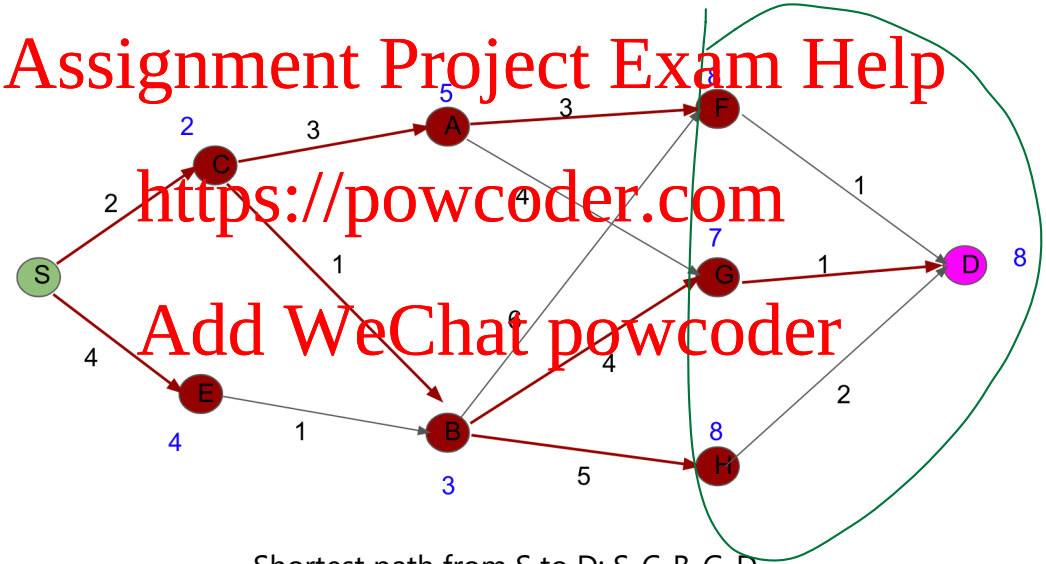




try to check distance, No update since 4+1=5>3

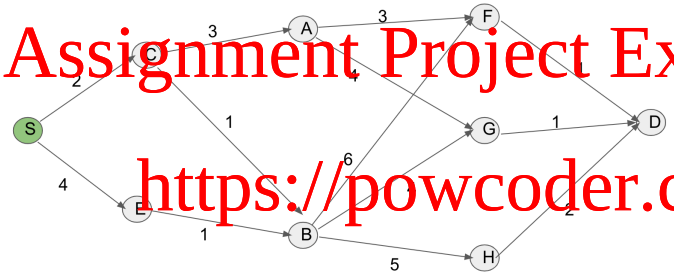






Shortest path from S to D: S-C-B-G-D

Looks similar



No signal at start

Starting Node = '00' state

Destination node 'D' = any rightmost state

Adjacent node distance = branch metric

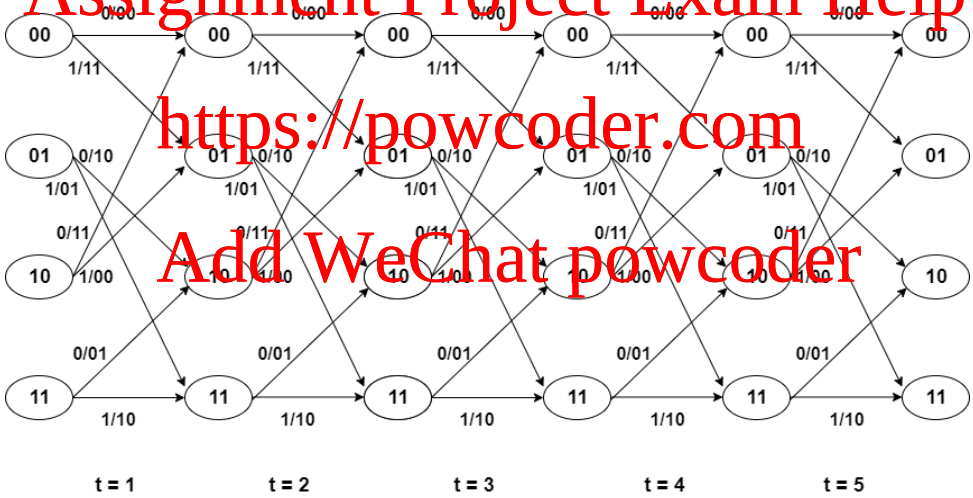
Received symbols: z_0,z_1,z_2,z_3,z_4

Code symbols: x_0,x_1,x_2,x_3,x_4

Branch metric = $|z_0 - x_0|^2 + |z_1 - x_1|^2$
and so on...

Viterbi Decoding (contd.)

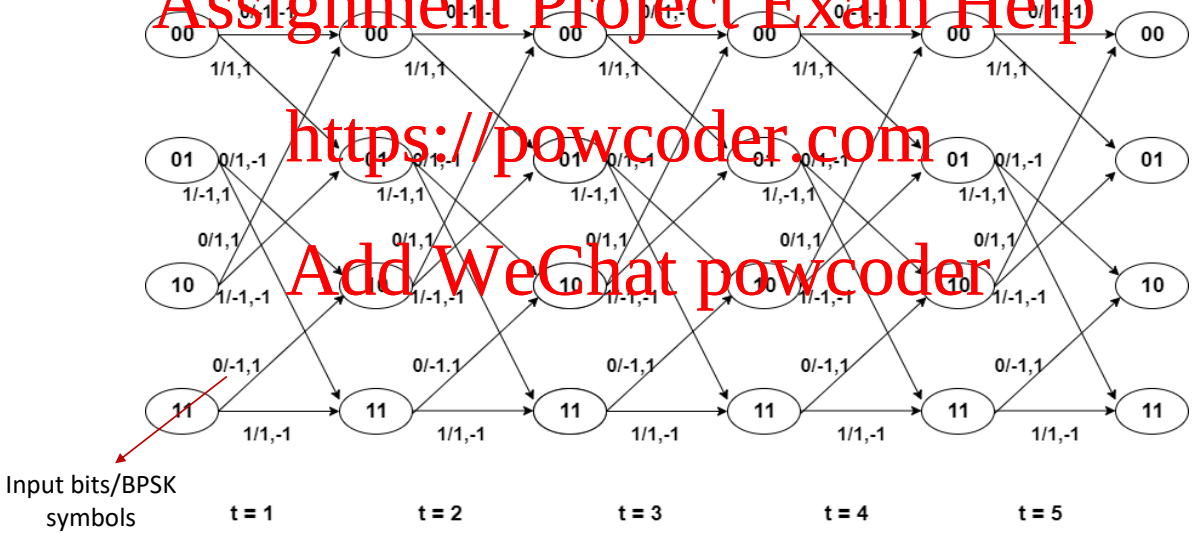
Step 1: Prepare Trellis



Viterbi Decoding (contd.)

This model look different for QAM Modulation!

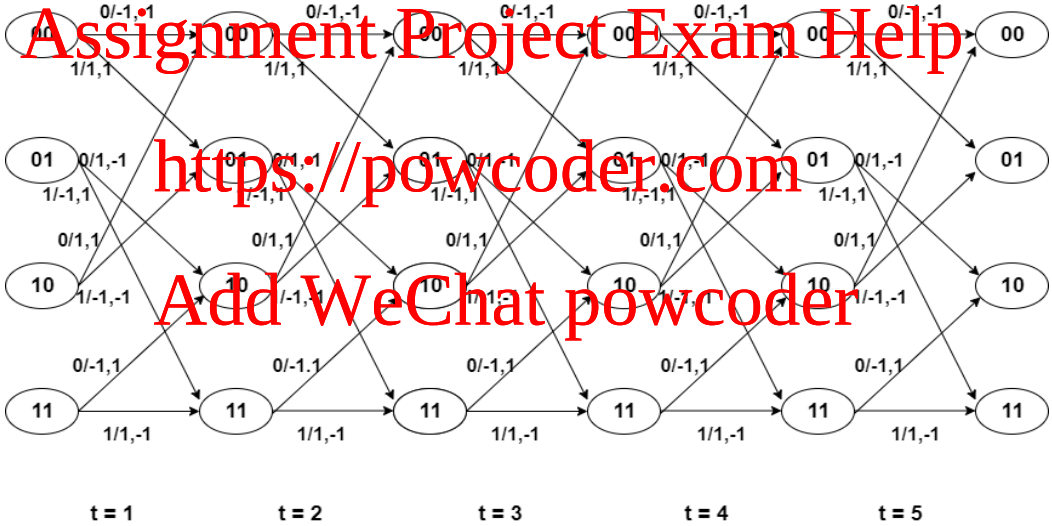
Step 2: Label using symbols x_k(BPSK modulation)



- For BPSK symbols are (1, -1)
- In project 1, values are different

Viterbi Decoding (contd.)

Step 3: Received sequency z_k

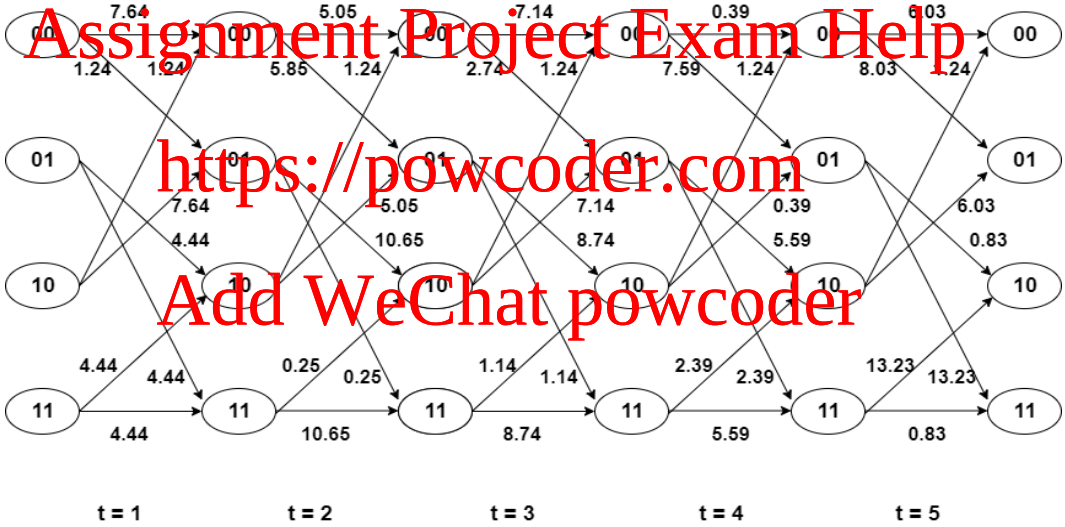


Received Sequence: $(z_0,z_1,z_2,z_3,z_4,z_5,z_6,z_7,z_8,z_9)$

= (0.8 + 0.4j , 0.8 - j, -1.4 + 0.1j , 1.2 - 0.2j, -0.4 + 0.2j, 1.5 + 0.7j , -1.3 + 0.2j , -0.5 -0.1j, 1.3 + 0.1j, -1.8-0.3j)

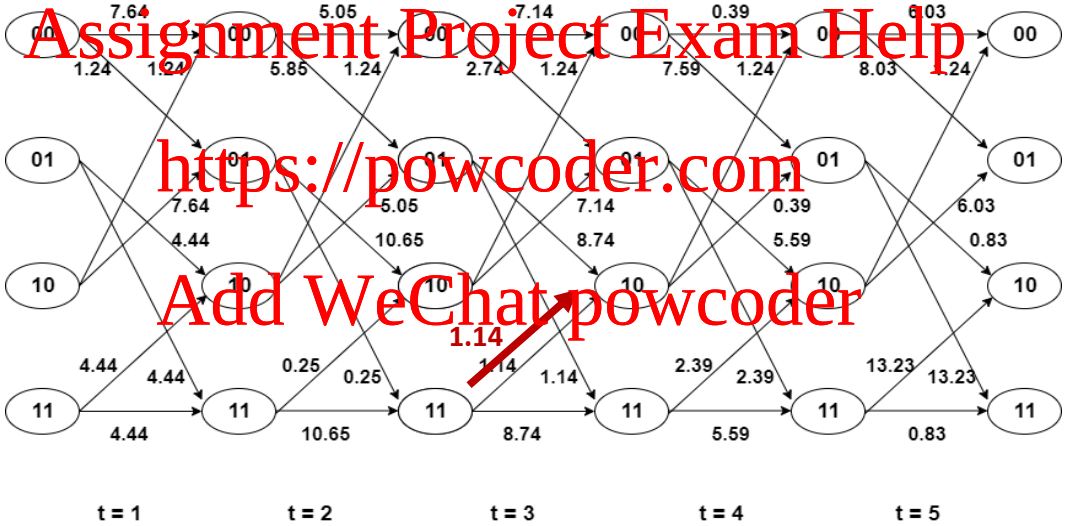
Viterbi Decoding (contd.)

Step 4: Calculate all branch metrics



Viterbi Decoding (contd.)

Branch metric calculation



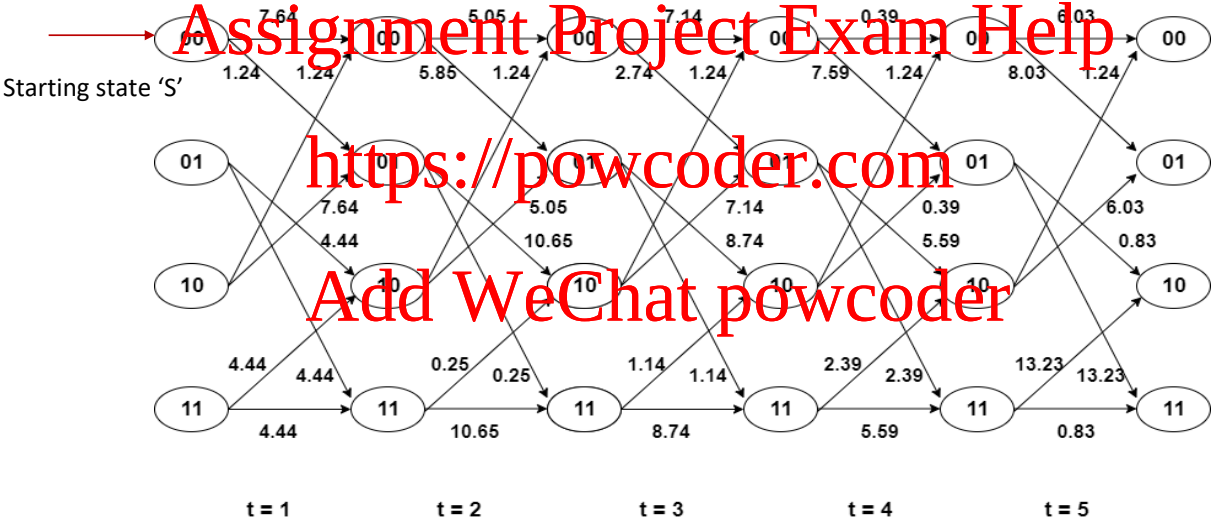
$(x_4 , x_5) = (-1 , 1)$

$(z_4 , z_5) = (-0.4 + 0.2j , 1.5 + 0.7j)$

Branch metric = $|z_4 - x_4|^2 + |z_5 - x_5|^2 = 1.14$

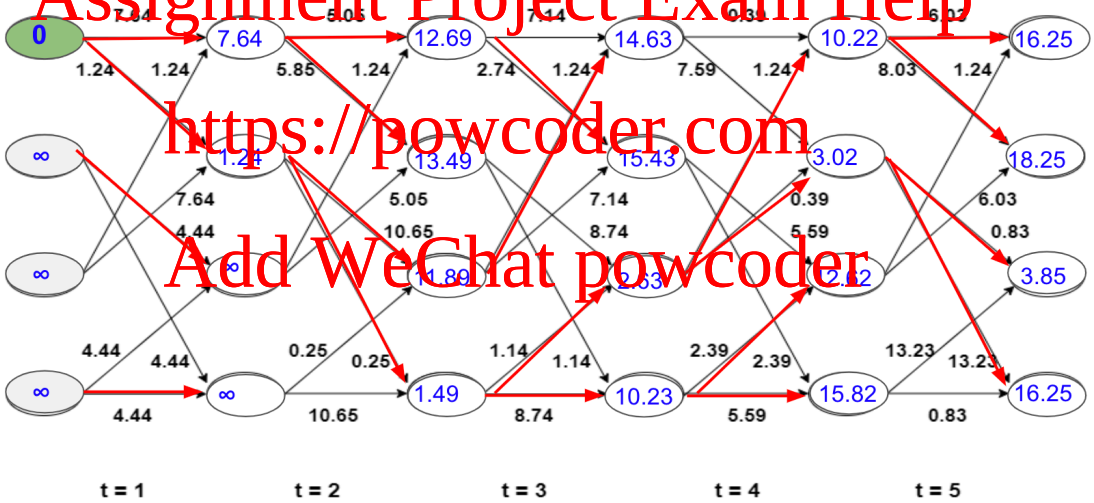
Viterbi Decoding (contd.)

Step 5: Find cost of reaching t=5 for all states with 00 as starting state S



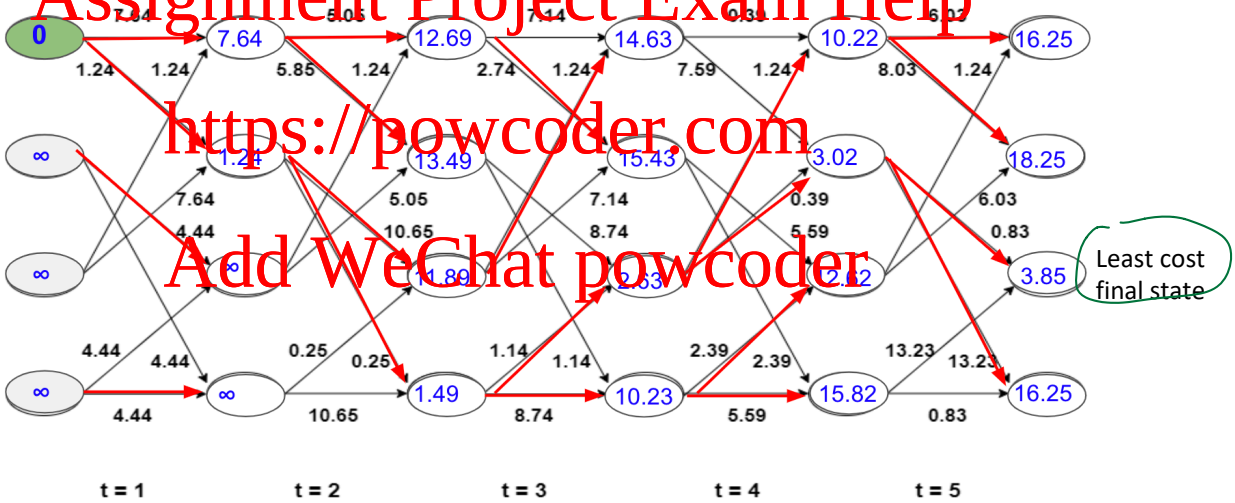
Viterbi Decoding (contd.)

Step 5: Find cost of reaching t=5 for all states with 00 as starting state S



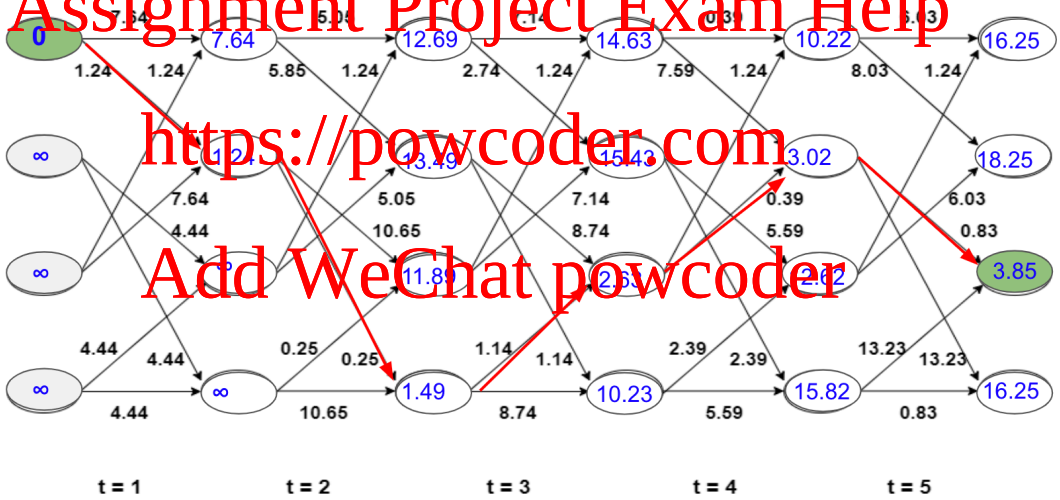
Viterbi Decoding (contd.)

Step 5: Find cost of reaching t=5 for all states with 00 as starting state S



Viterbi Decoding (contd.)

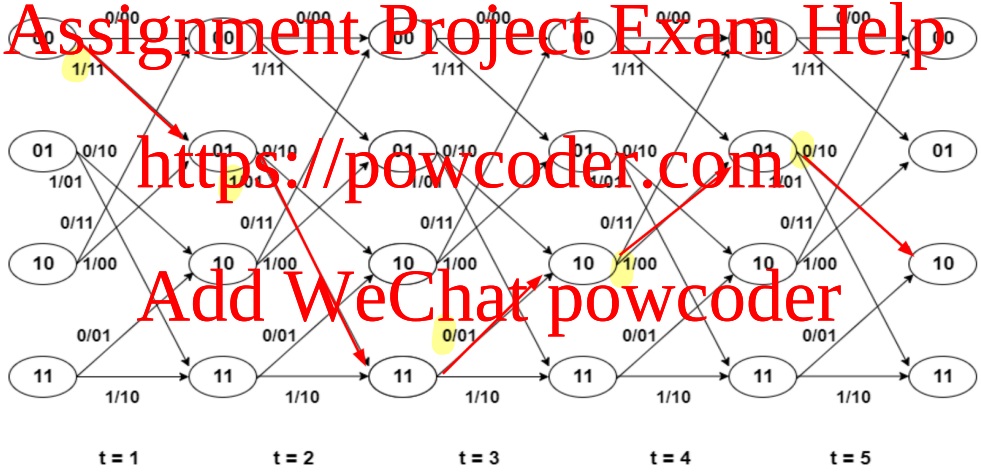
Step 6: Traceback to find shortest path



Least cost final state

Viterbi Decoding (contd.)

Step 7: Use Trellis with bits to get decoded bits



1 1 0 1 0

Input bit sequence estimate is 1 1 0 1 0

Closing Comments

- You are welcome to use any signal processing libraries except for turbo code decoding libraries
- Deliverables:
  - MATLAB source code: .m file
  - Design report 2-3 pages
- Don't submit executable files
- Due date: Feb 28, 11.59 pm EDT