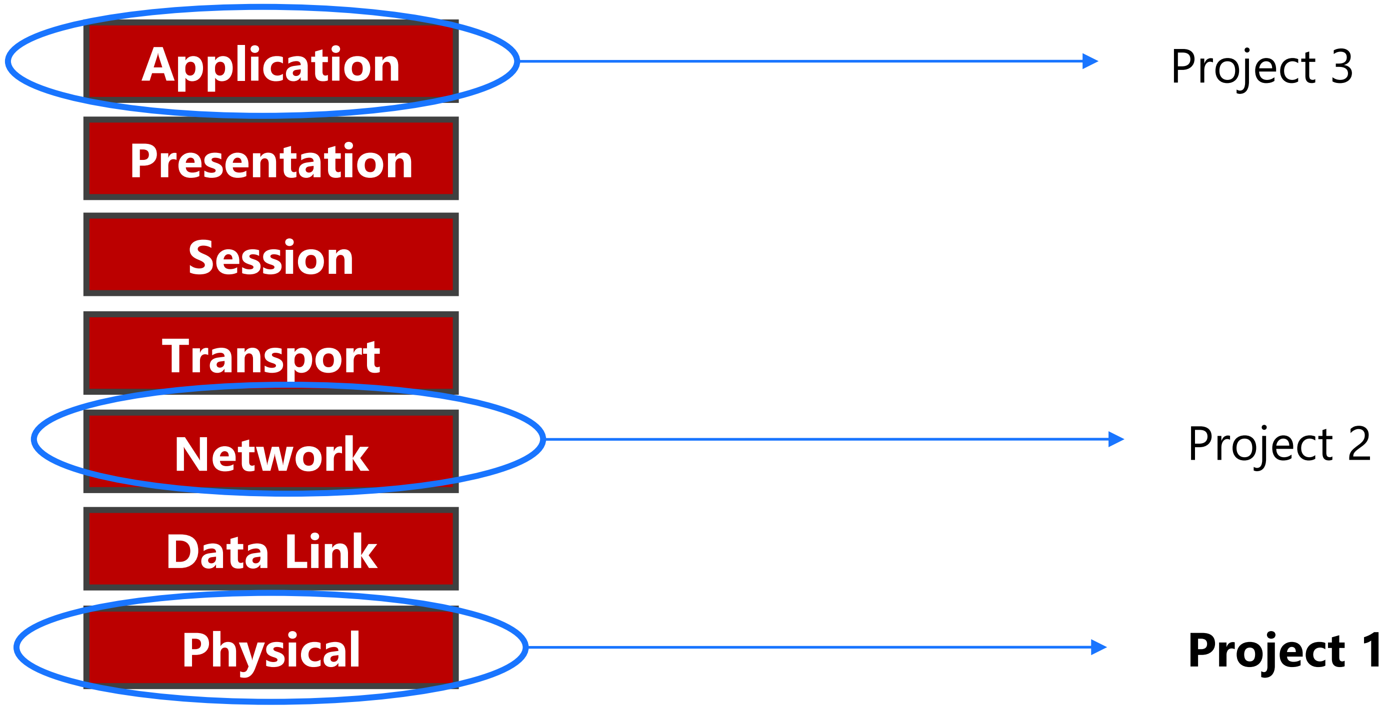
Carnegie Mellon University

Recitation 1: Exploring the WiFi Physical Layer

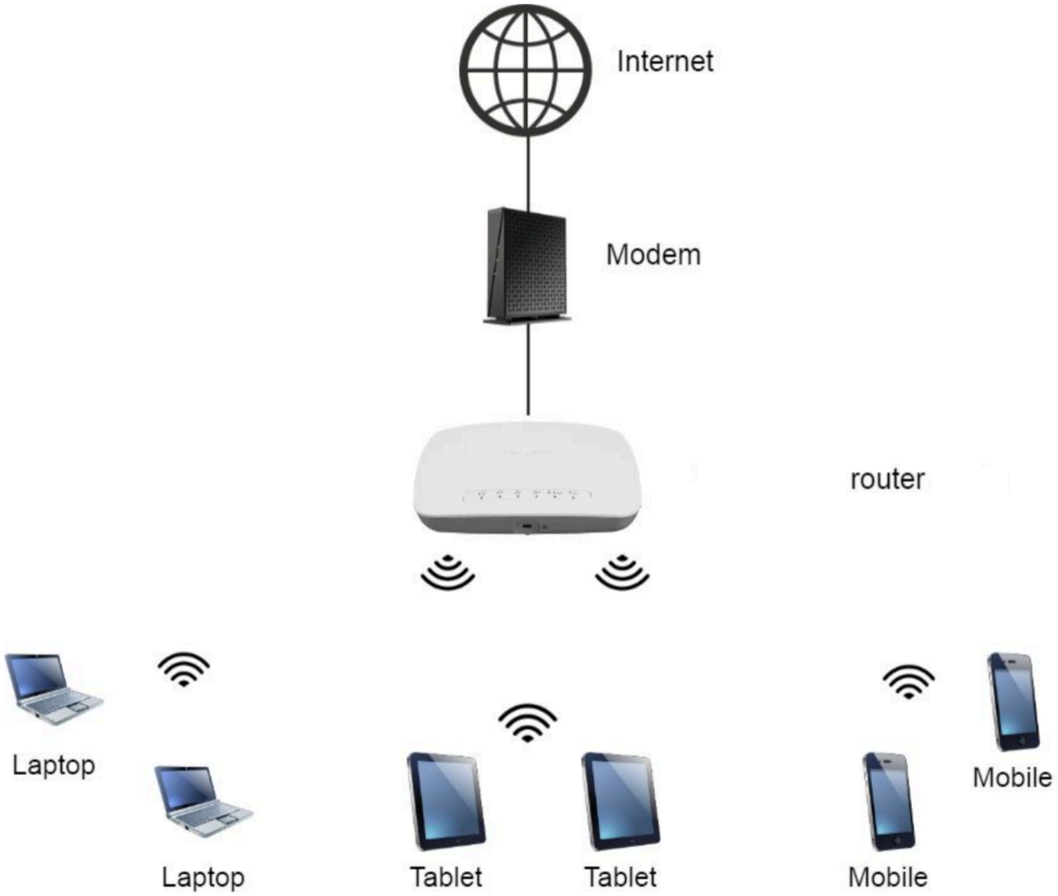
FEBRUARY 12, 2021

Atul Bansal

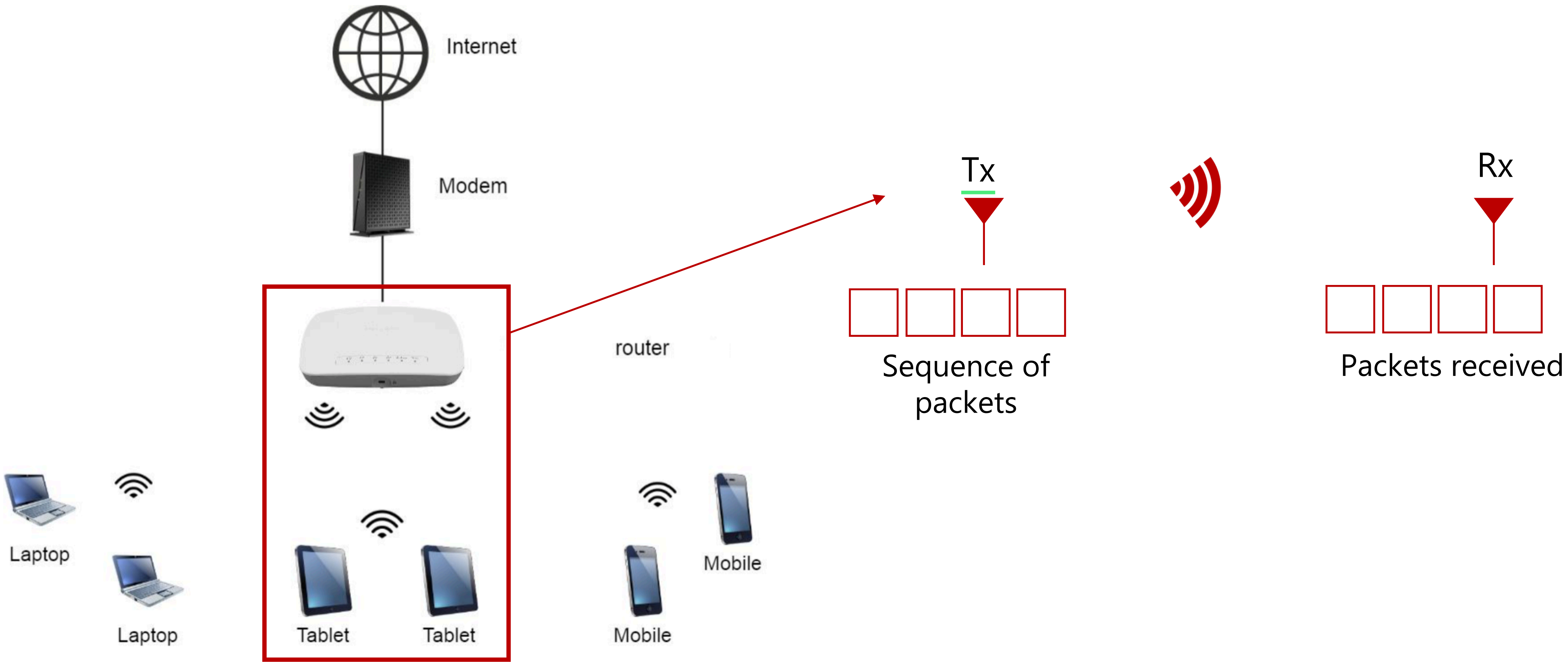
Open Systems Interconnection Model



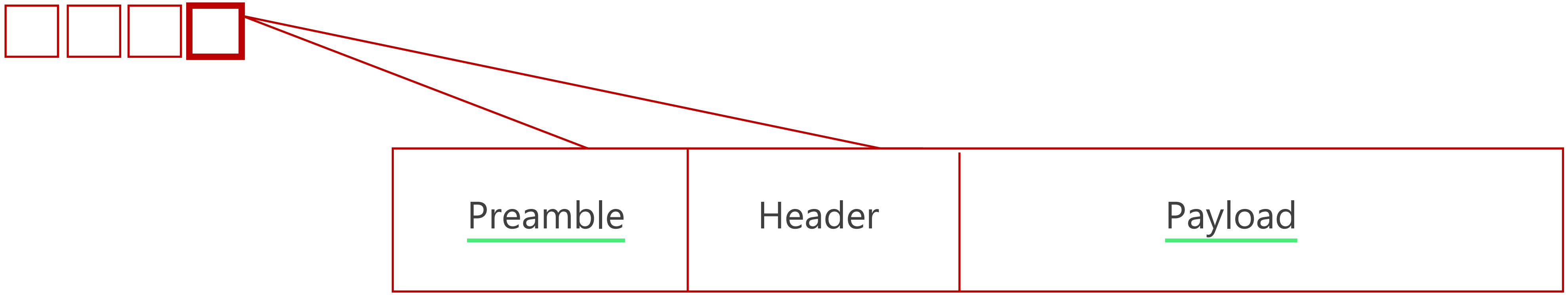
Project 1: Physical Layer(WiFi)



Project 1: Physical Layer(WiFi)



WiFi Packet

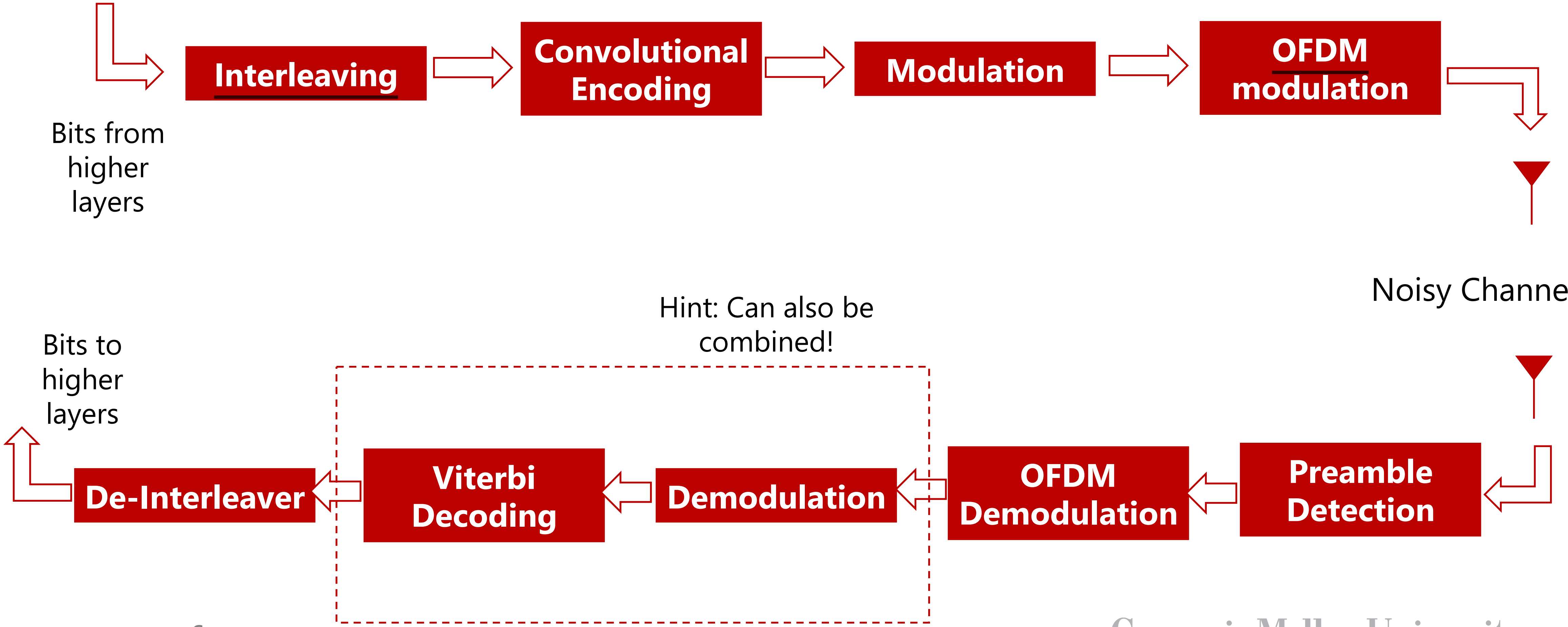


Known sequence used to detect the start of the packet

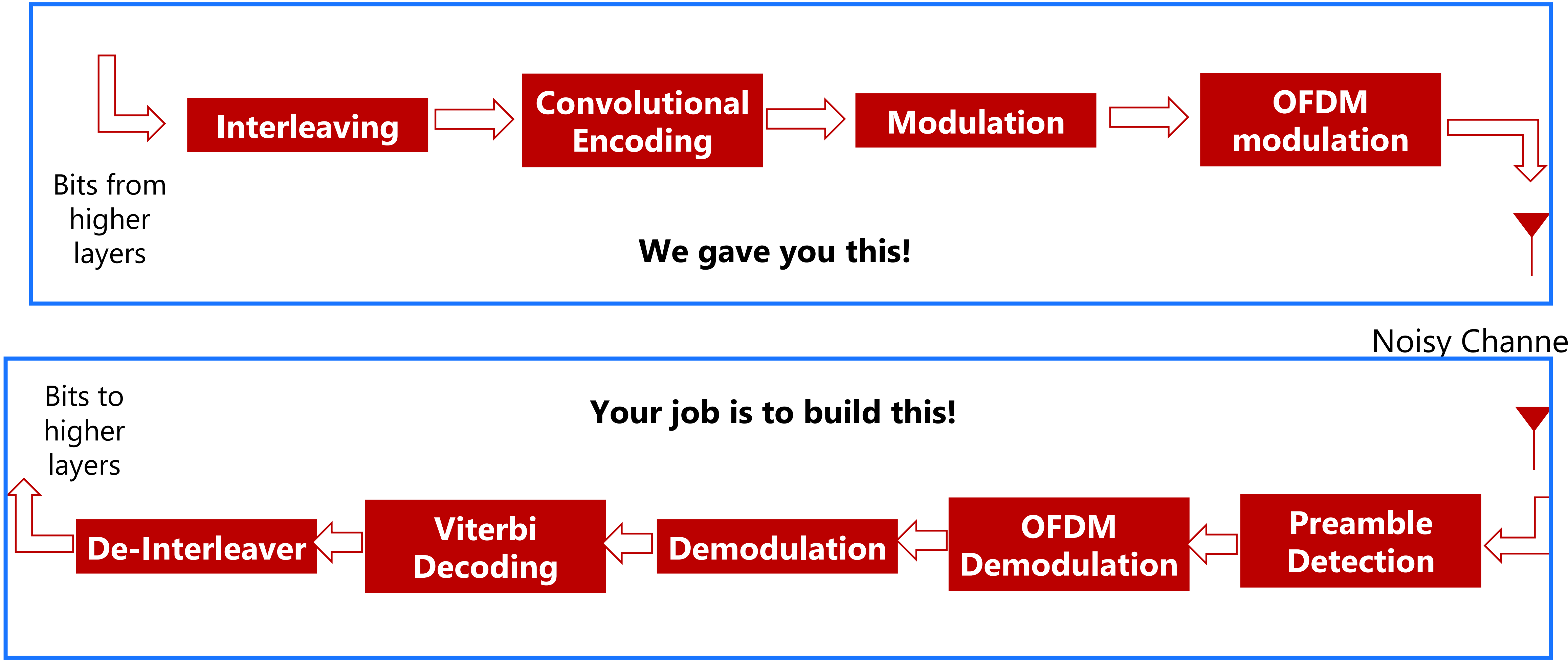
Extra information about the data

Actual Data

How WiFi packets are created?



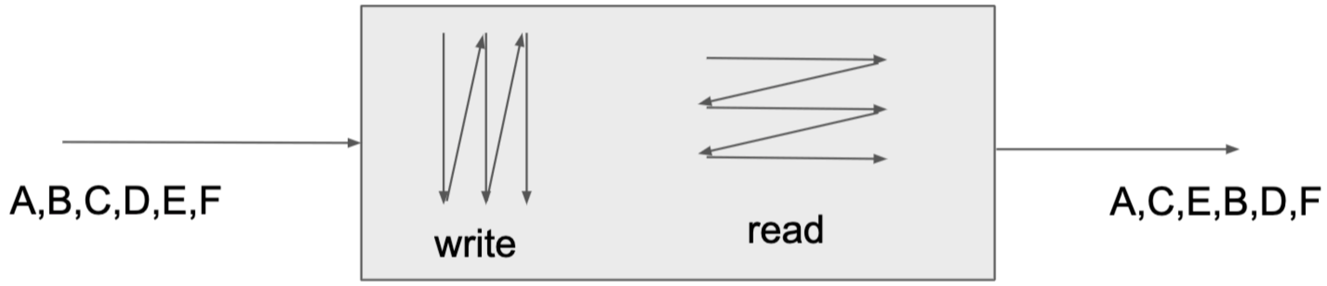
How WiFi packets are created?



Level 1: Interleaving

Input bits are rearranged by a well-known permutation pattern

Interleaver: Rows = 2, Columns = 3



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

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 ½ rate encoder with G(7,5) generator polynomial

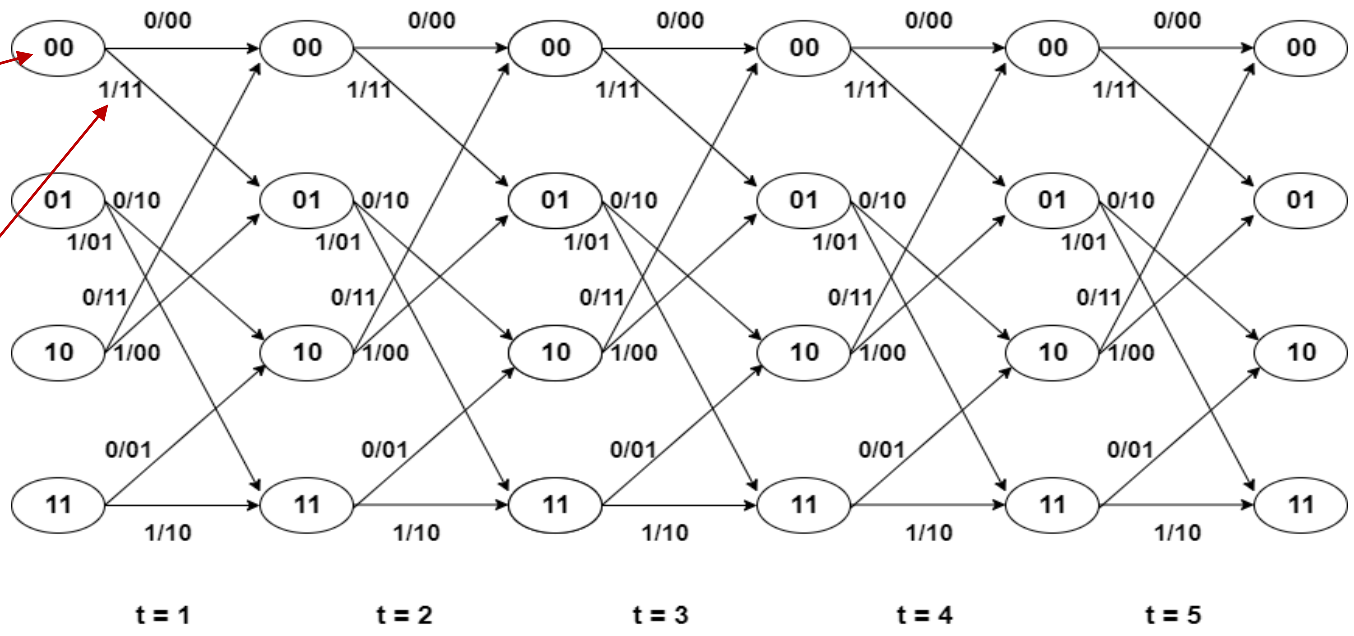
Input Bits: $i(n)$, $i(n-1)$, $i(n-2)$

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

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

State:

$i(n-2), i(n-1)$

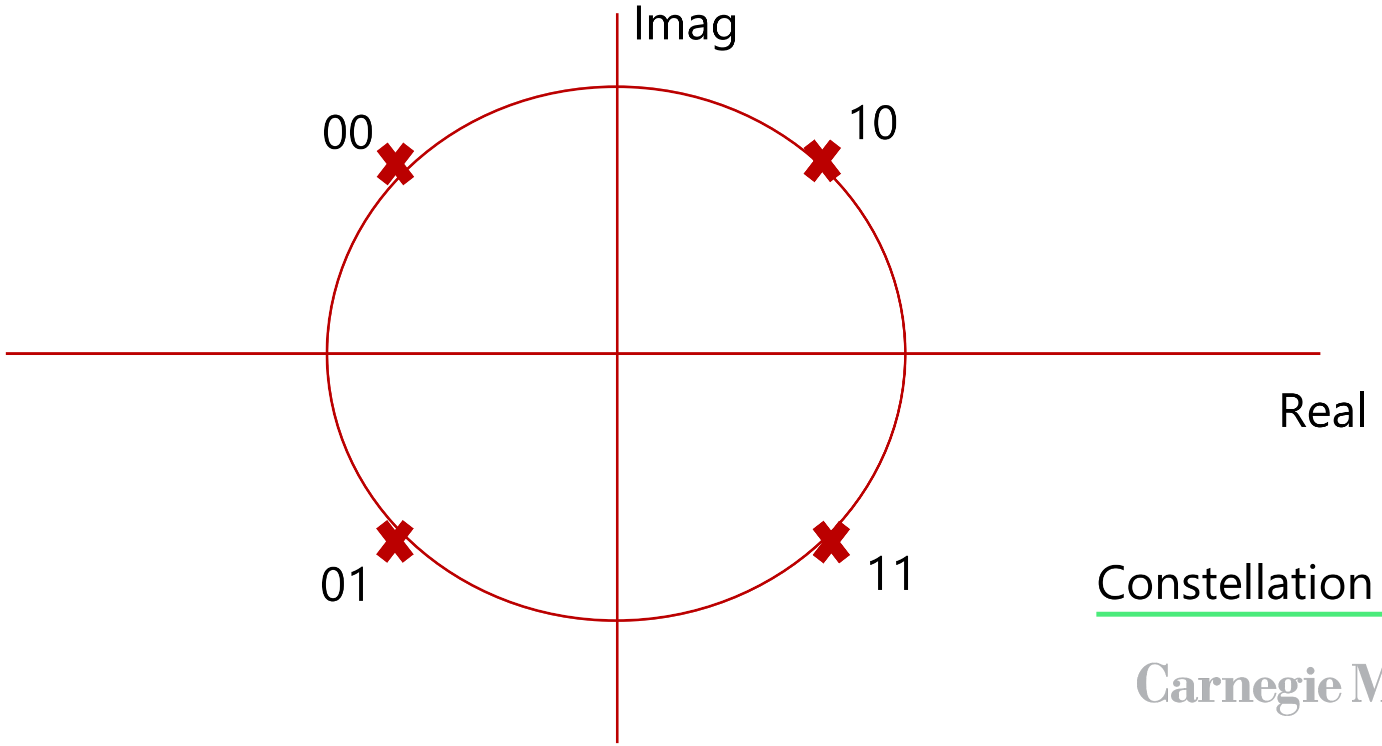


$i(n)/\ c_1(n),\ c_2(n)$

Level 3: Modulation

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

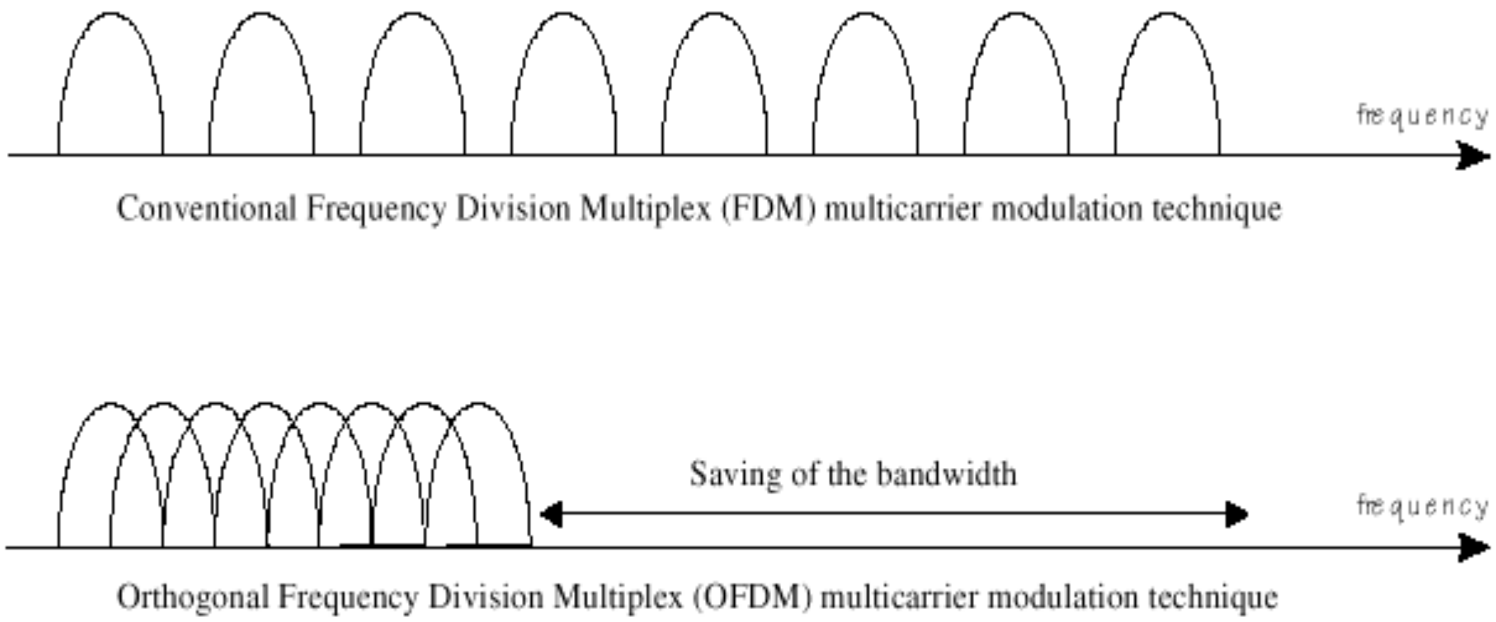
Conveying information by changing the signal and represent them in complex domain.
We use 4-QAM in this project.



Constellation Diagram

Level 4: Orthogonal Frequency Division Multiplexing (OFDM)

Divide the available bandwidth into multiple orthogonal subcarriers



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.

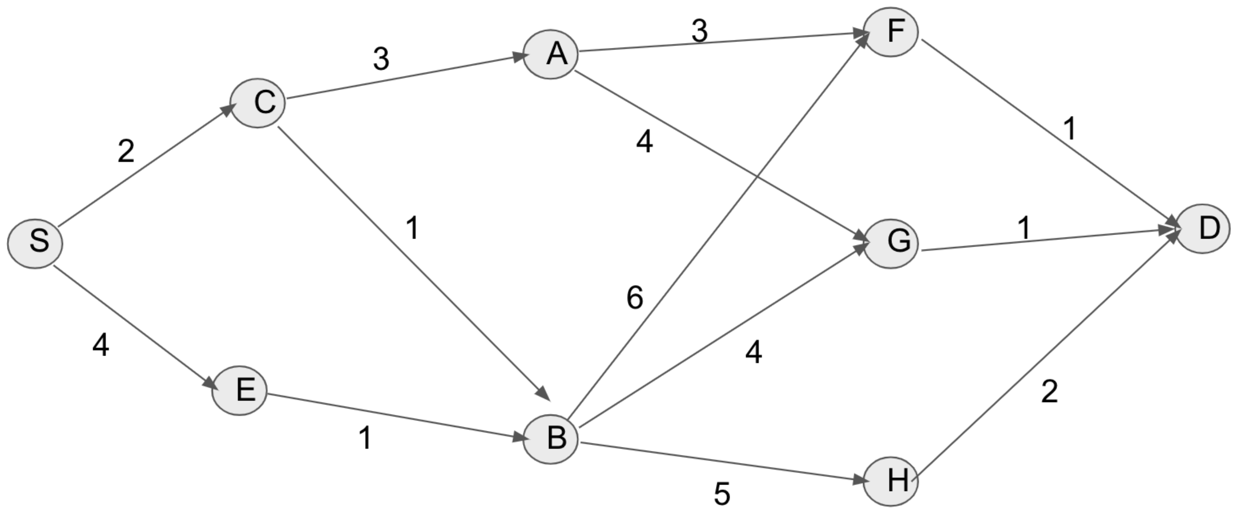
Detect beginning and end of info symbols (useful symbols)

Beginning: Preamble
End: ??

Viterbi Decoding

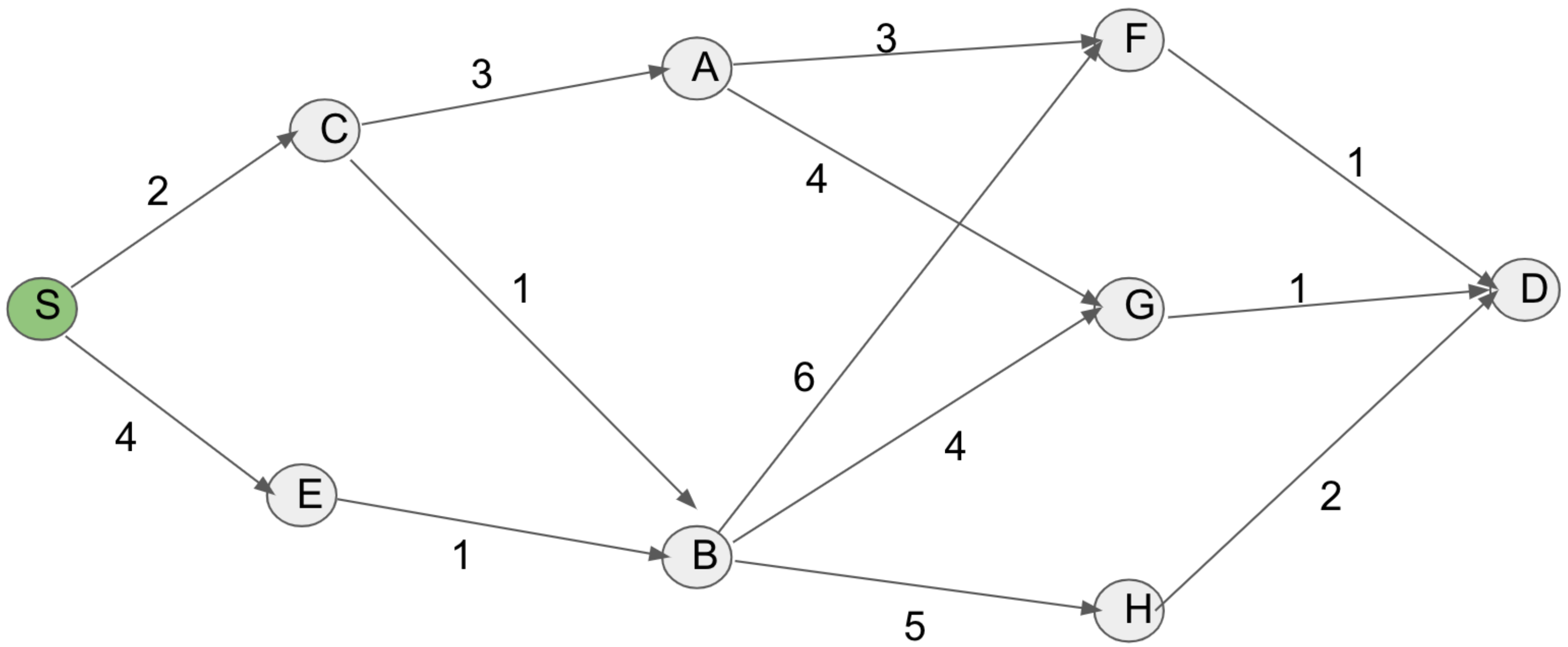
Used at WiFi receivers to inverse Convolutional coding

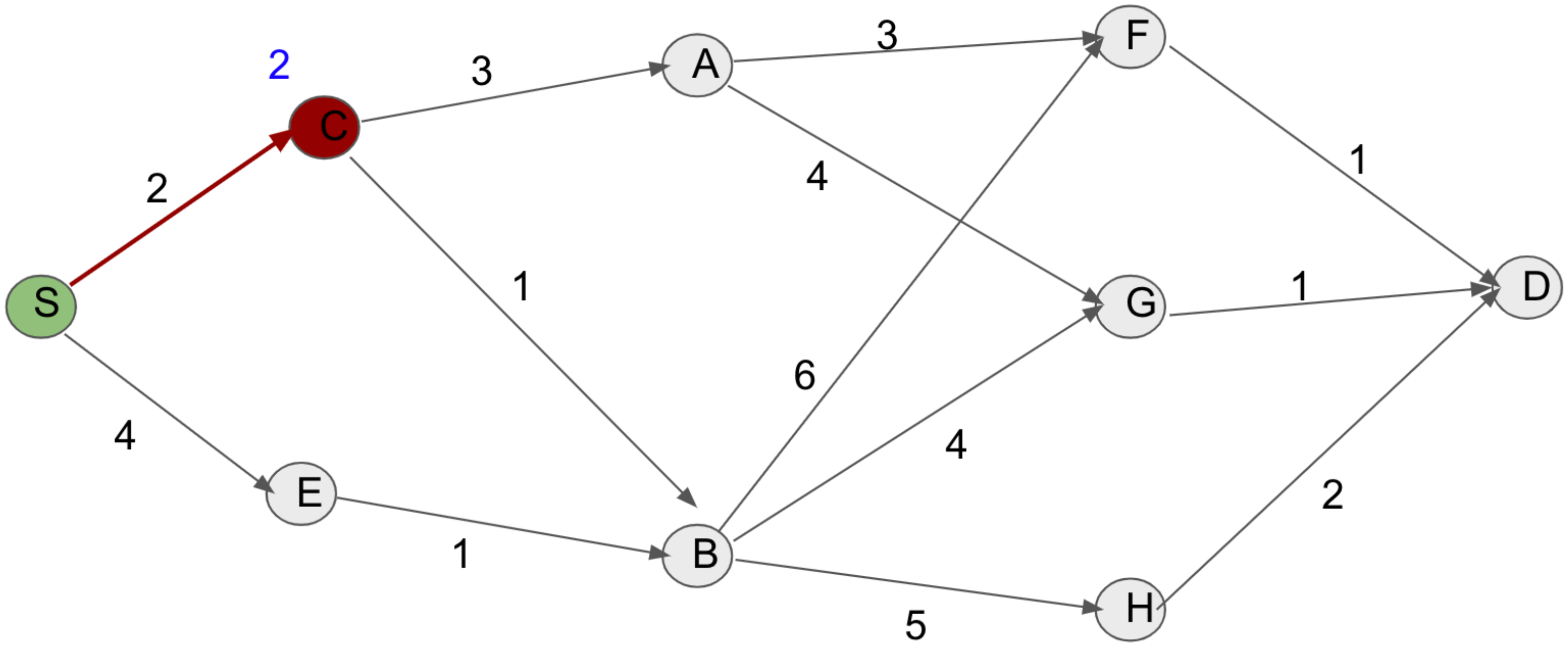
Before that, consider this problem

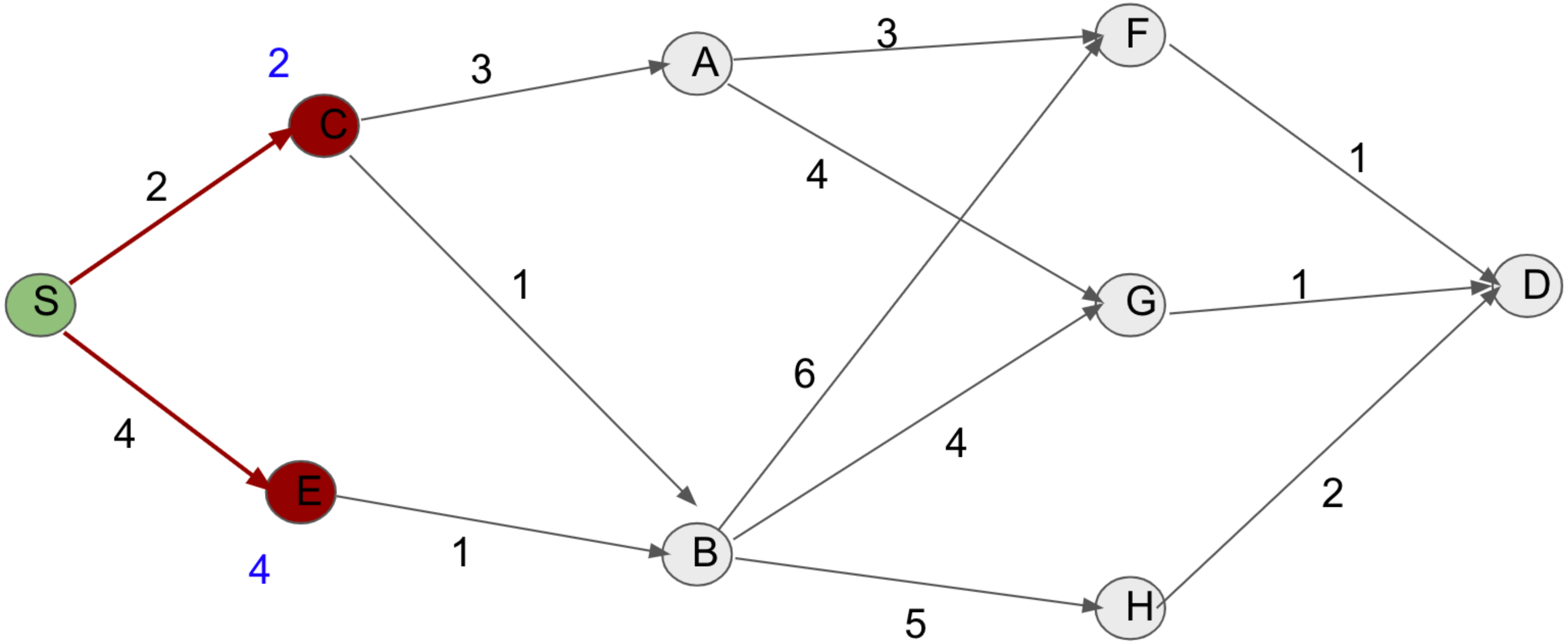


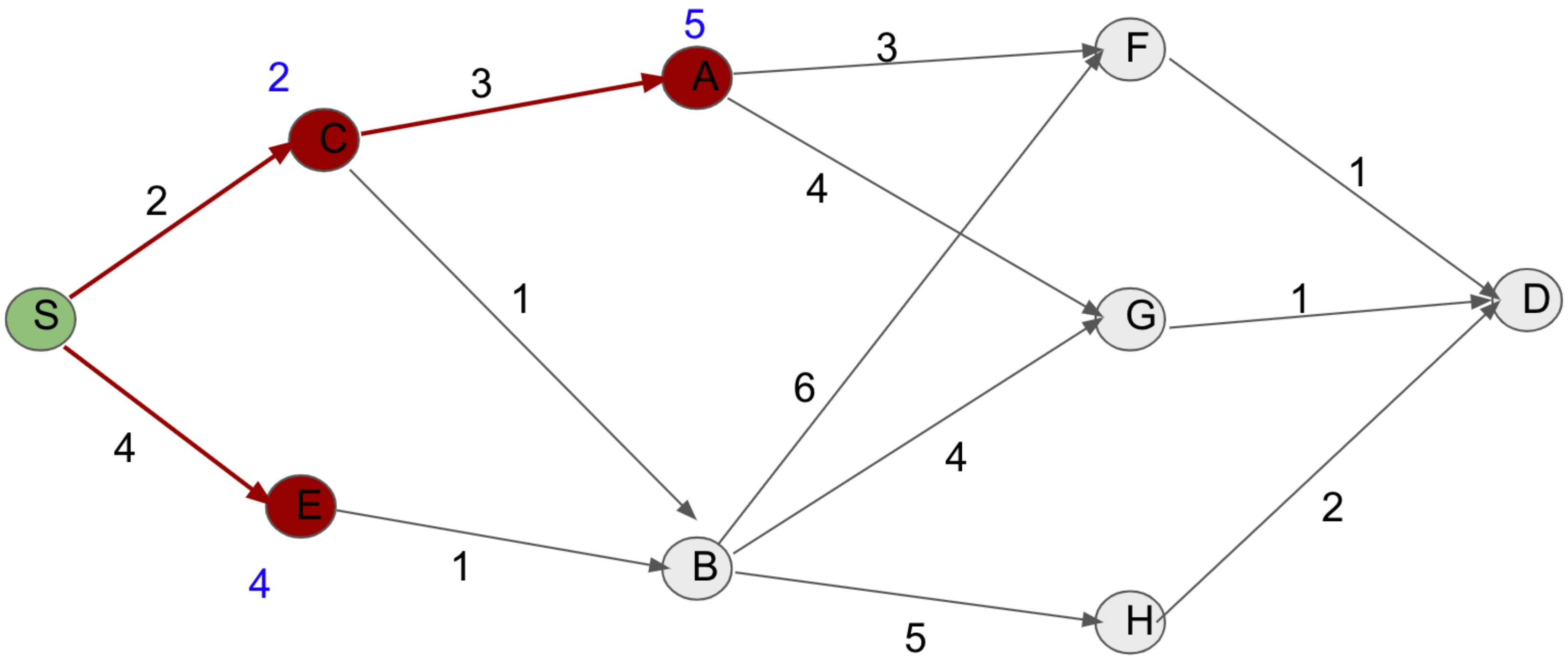
Find shortest distance from S to D?

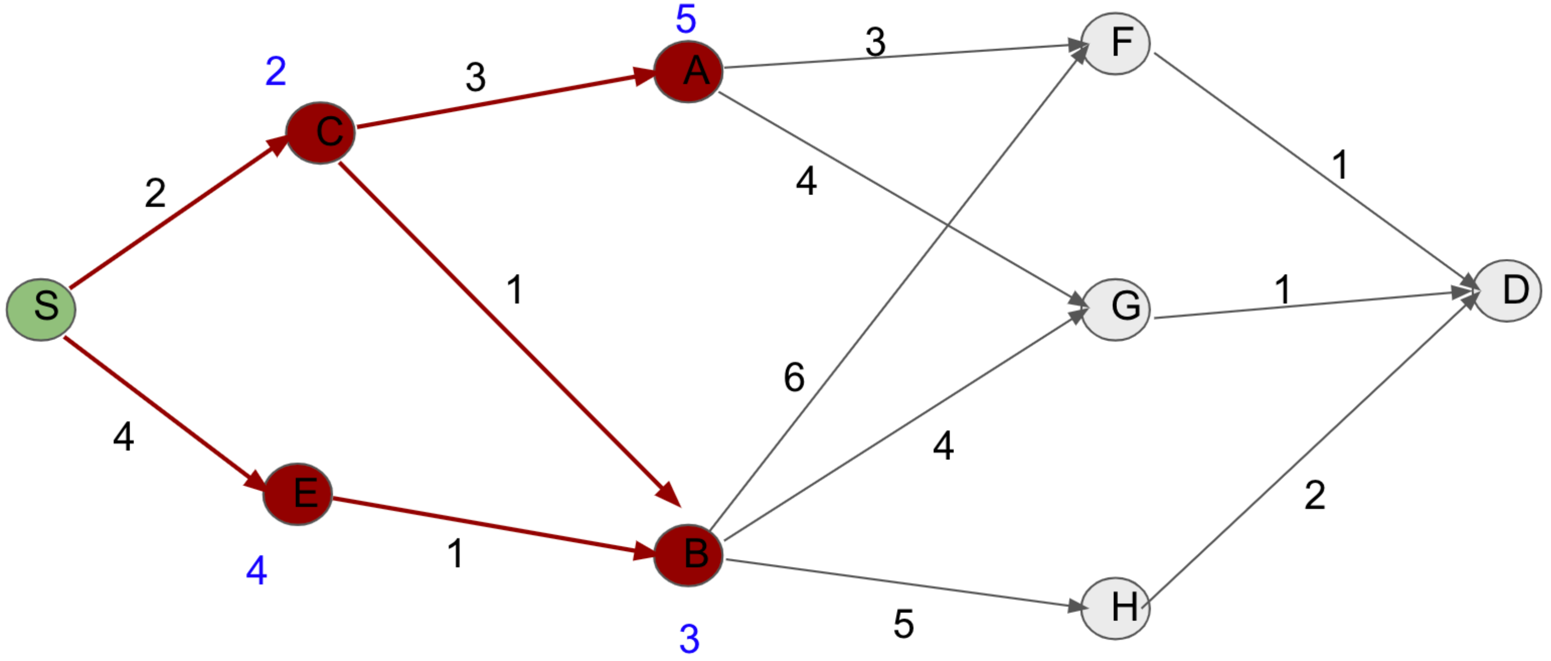
Use Dynamic Programming/Dijkstra's algorithm

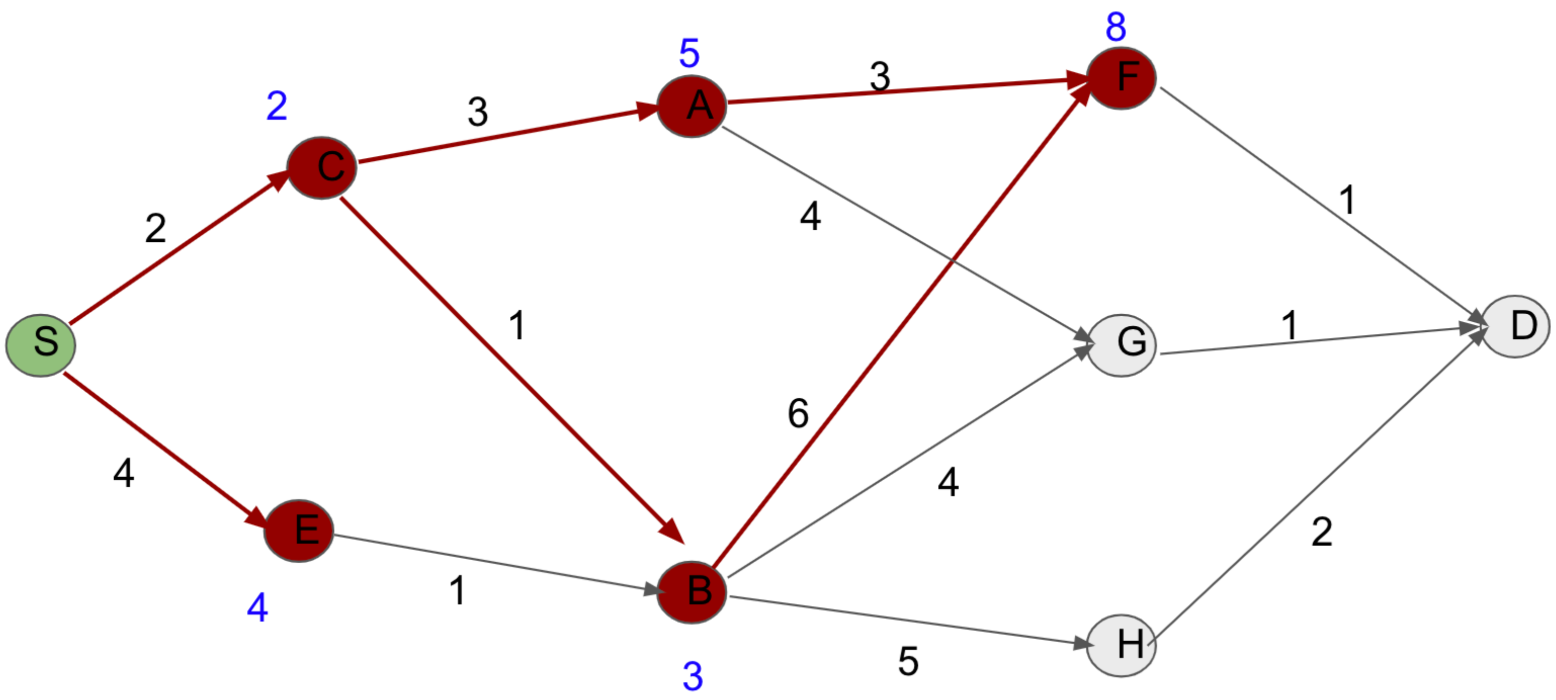

S
C
E
A
B
F
G
H
D
2
4
3
1
1
3
4
6
4
5
1
1
2

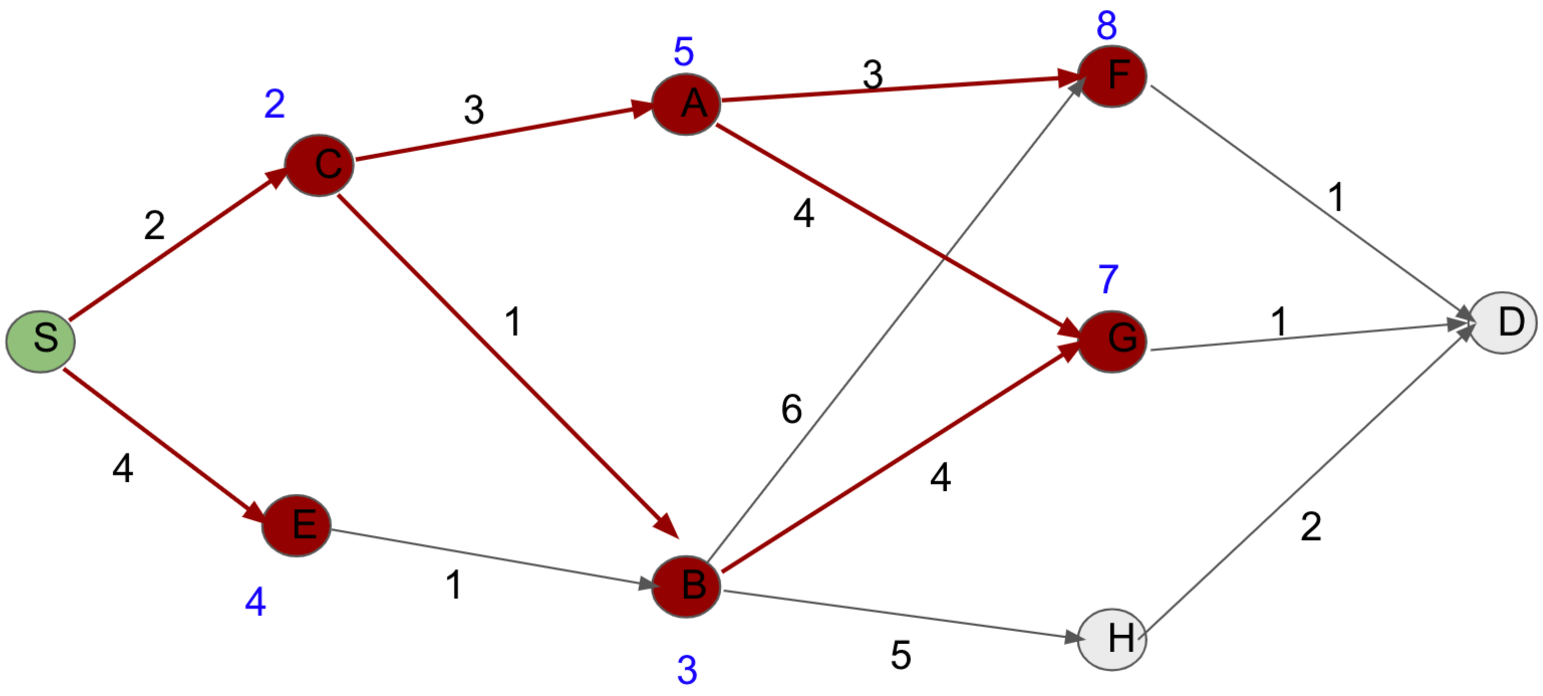

2
S
C
A
F
G
D
E
B
H
2
4
3
1
1
3
4
6
4
5
1
1
2

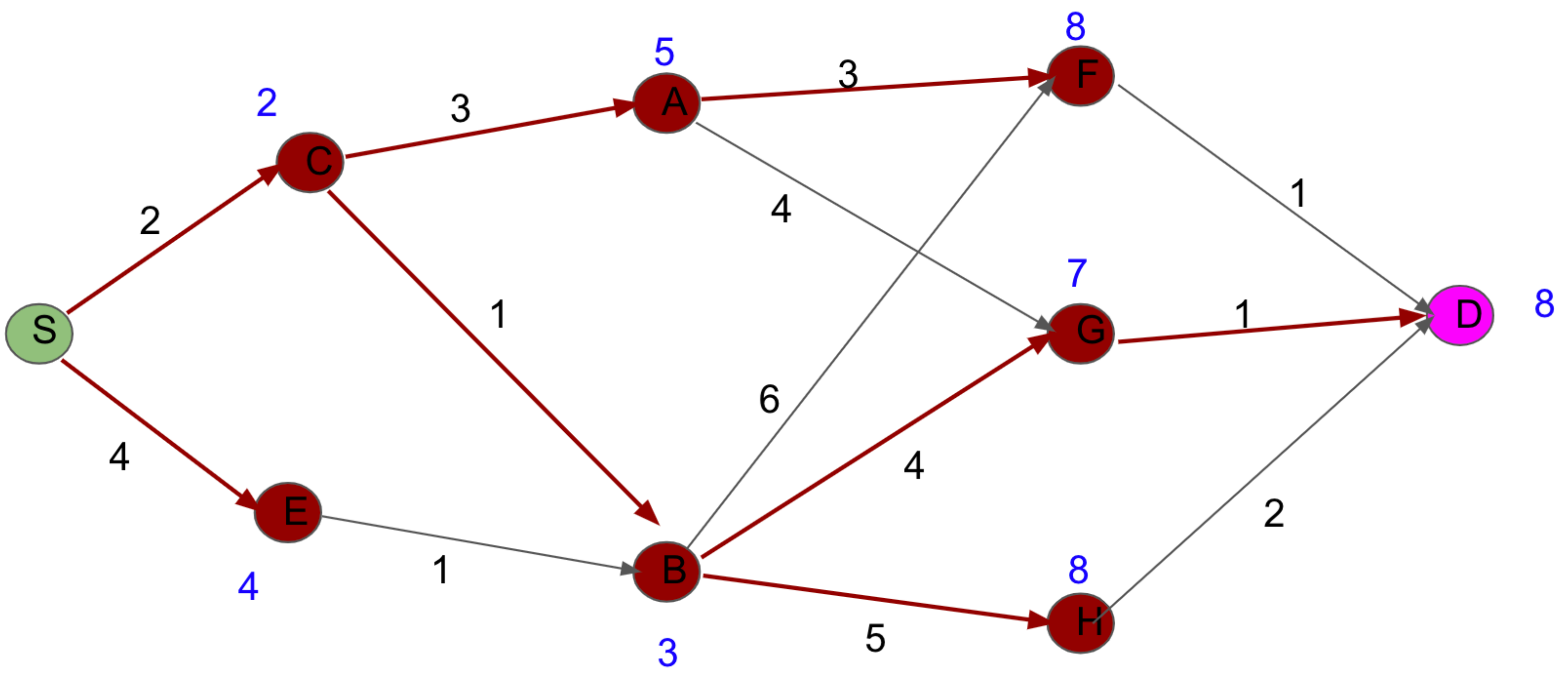

S
C
E
A
B
F
G
H
D
2
4
2
4
3
1
1
3
4
6
4
5
1
1
2


5
2
4
3
S
C
A
F
G
D
E
B
H
2
3
3
4
1
1
1
6
4
4
1
5
2


S
C
E
A
B
F
G
H
D
2
4
5
3
2
4
3
1
1
3
4
6
4
5
1
1
2

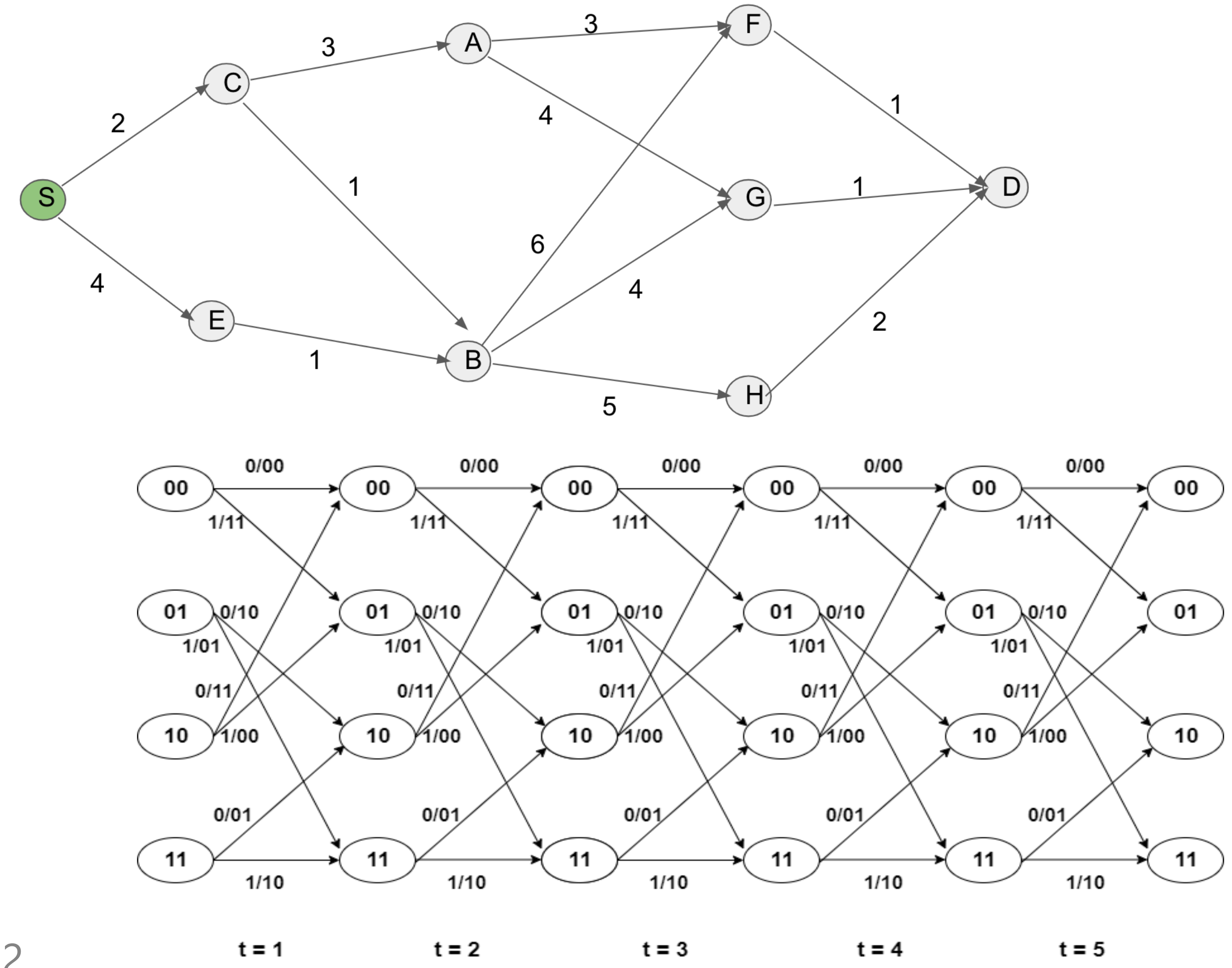

2
5
8
3
3
C
A
F
2
4
1
1
S
G
1
D
6
4
4
E
2
1
B
H
4
5
3


8
5
3
F
2
3
A
C
1
4
2
7
1
1
S
G
D
6
4
4
E
2
1
B
4
H
5
3



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

Looks similar



Starting Node = ‘00’ state

Destination node ‘D’ = any rightmost state

Adjacent node distance = branch metric

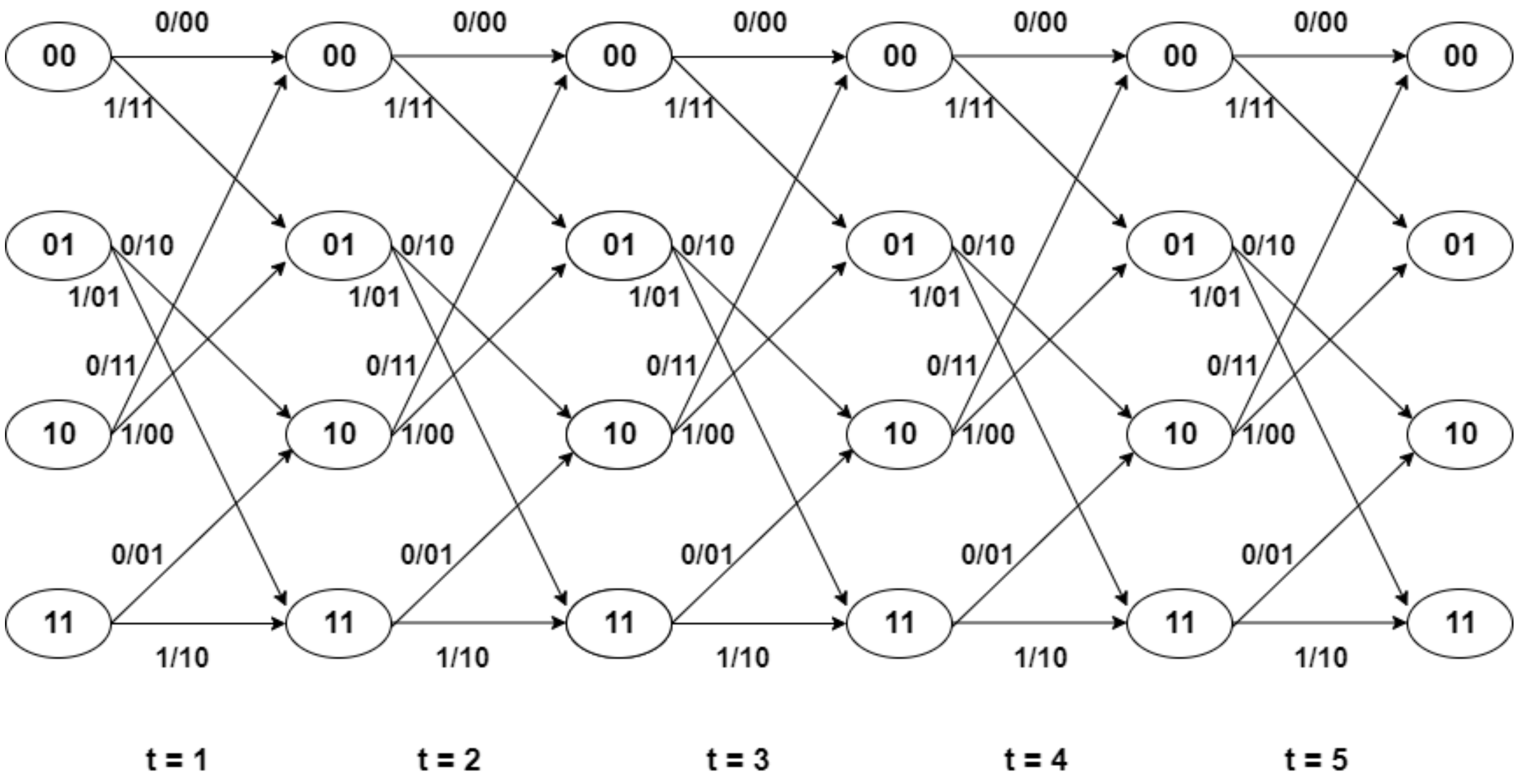
Received symbols: z_0,z_1,z_2,z_3,z_4

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…

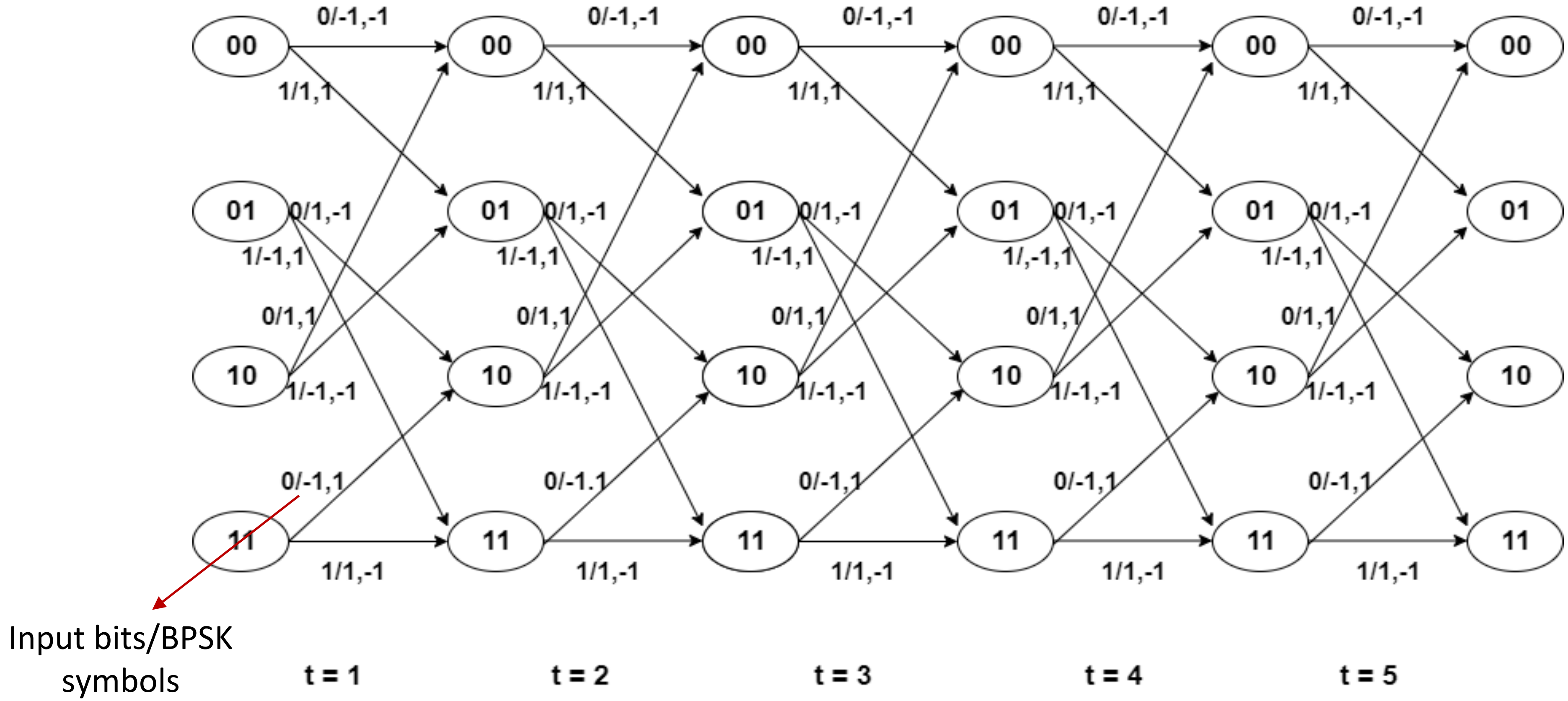
Viterbi Decoding (contd.)

Step 1: Prepare Trellis



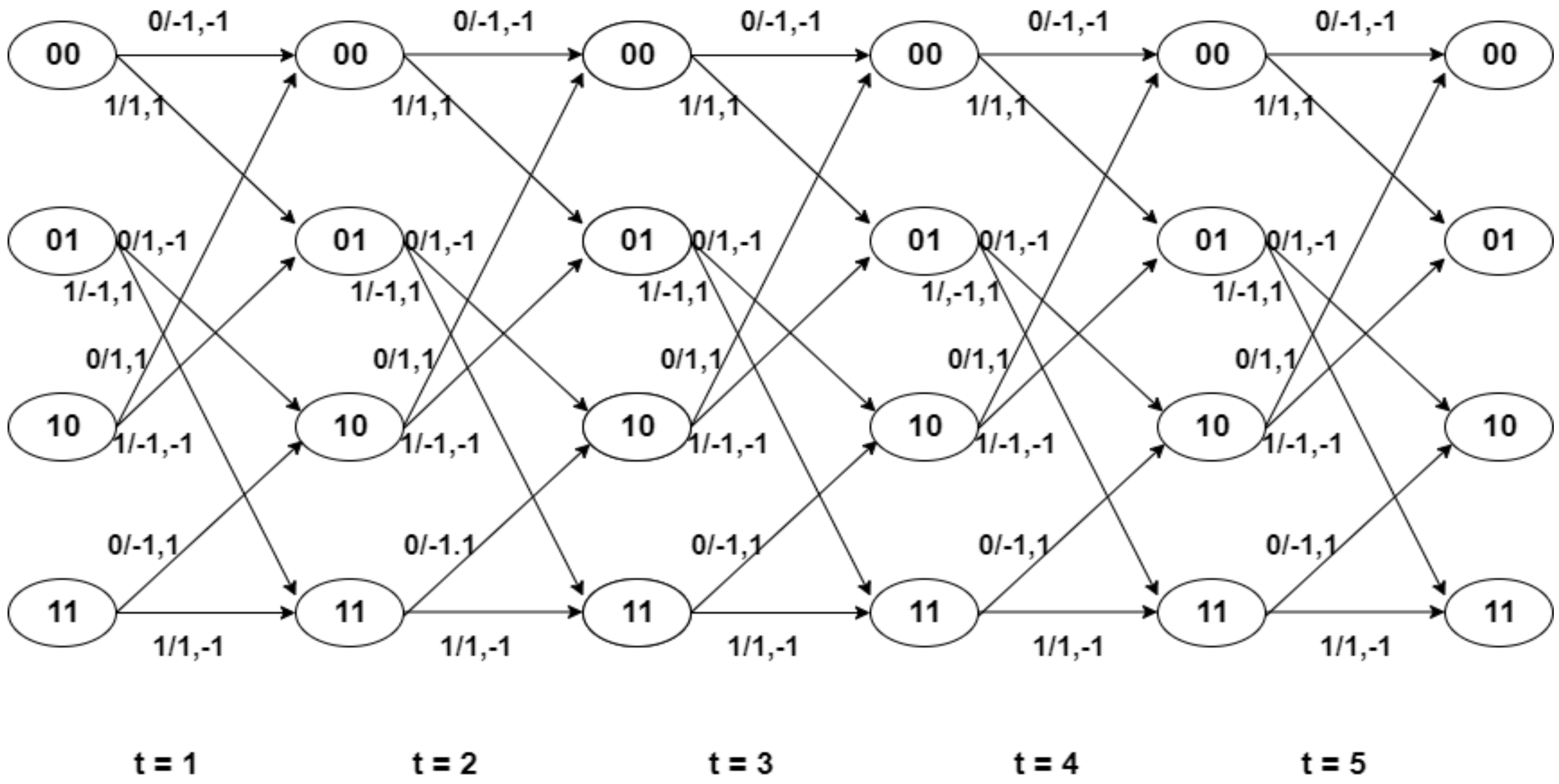
Viterbi Decoding (contd.)

Step 2: Label using symbols x_k(BPSK modulation)



Viterbi Decoding (contd.)

Step 3: Received sequency z_k

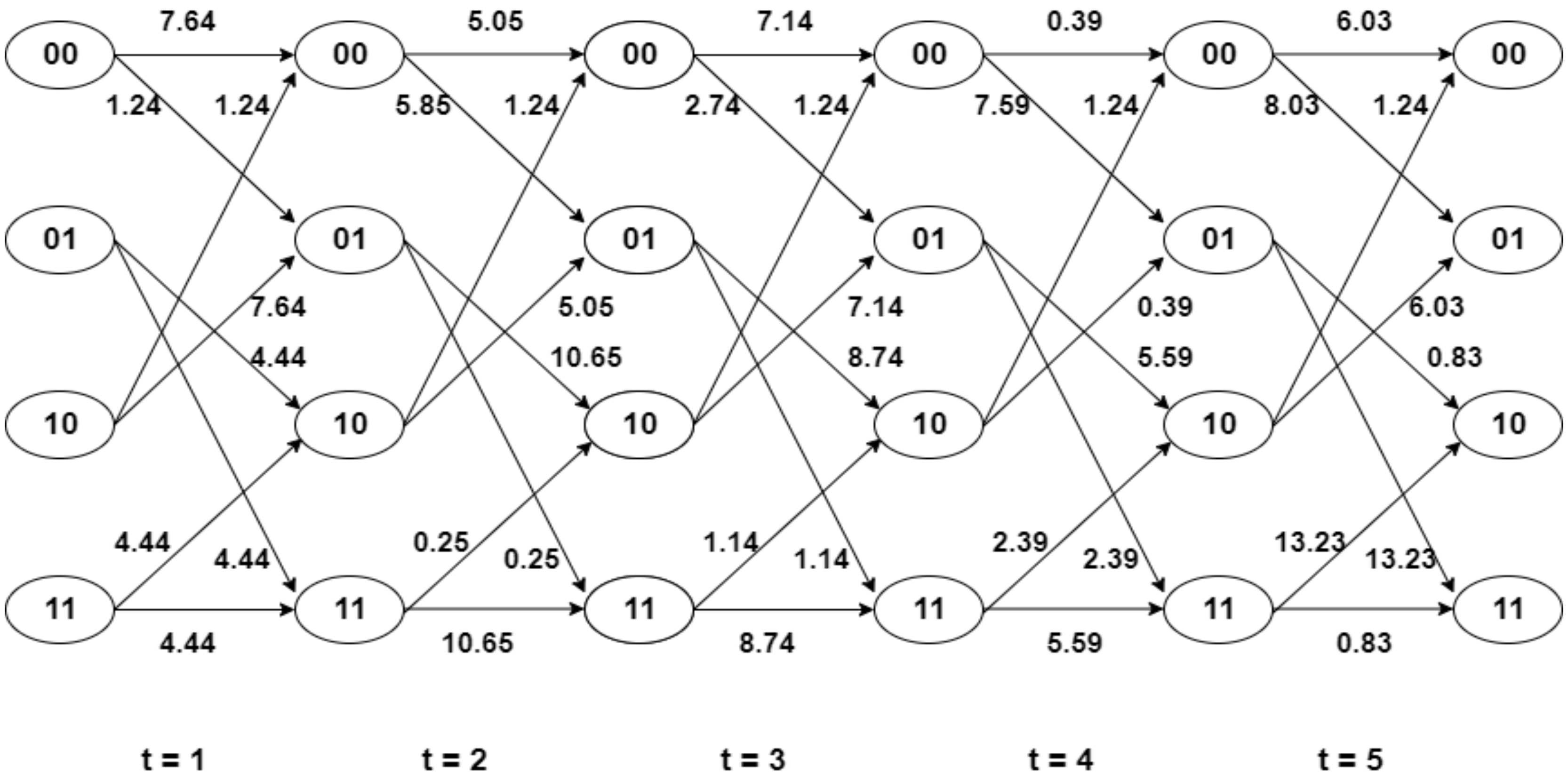


Received Sequence: $(z_0,z_1,z_2,z_3,z_4,z_5,z_6,z_7,z_8,z_9)$

= (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)

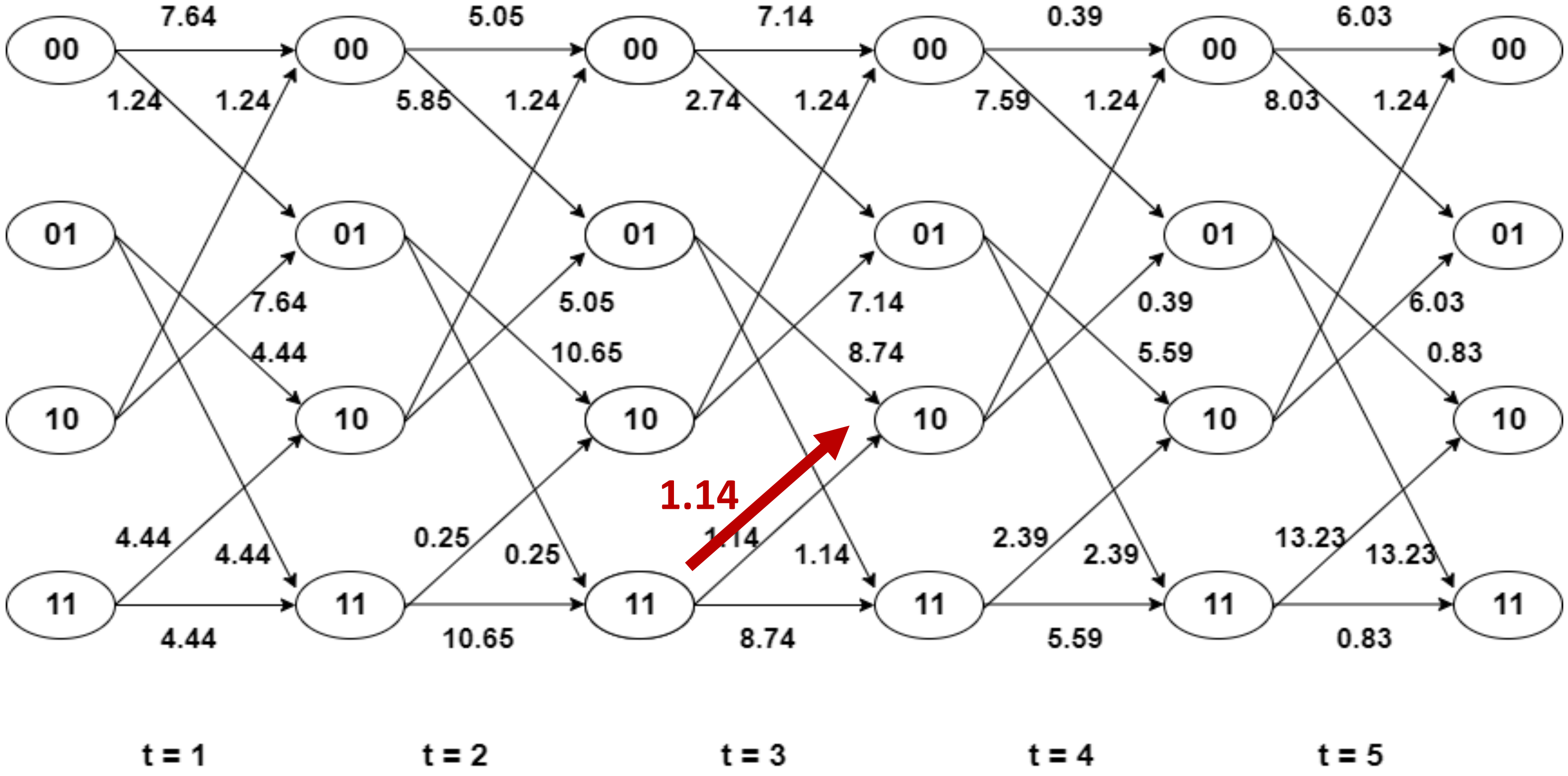
Viterbi Decoding (contd.)

Step 4: Calculate all branch metrics



Viterbi Decoding (contd.)

Branch metric calculation



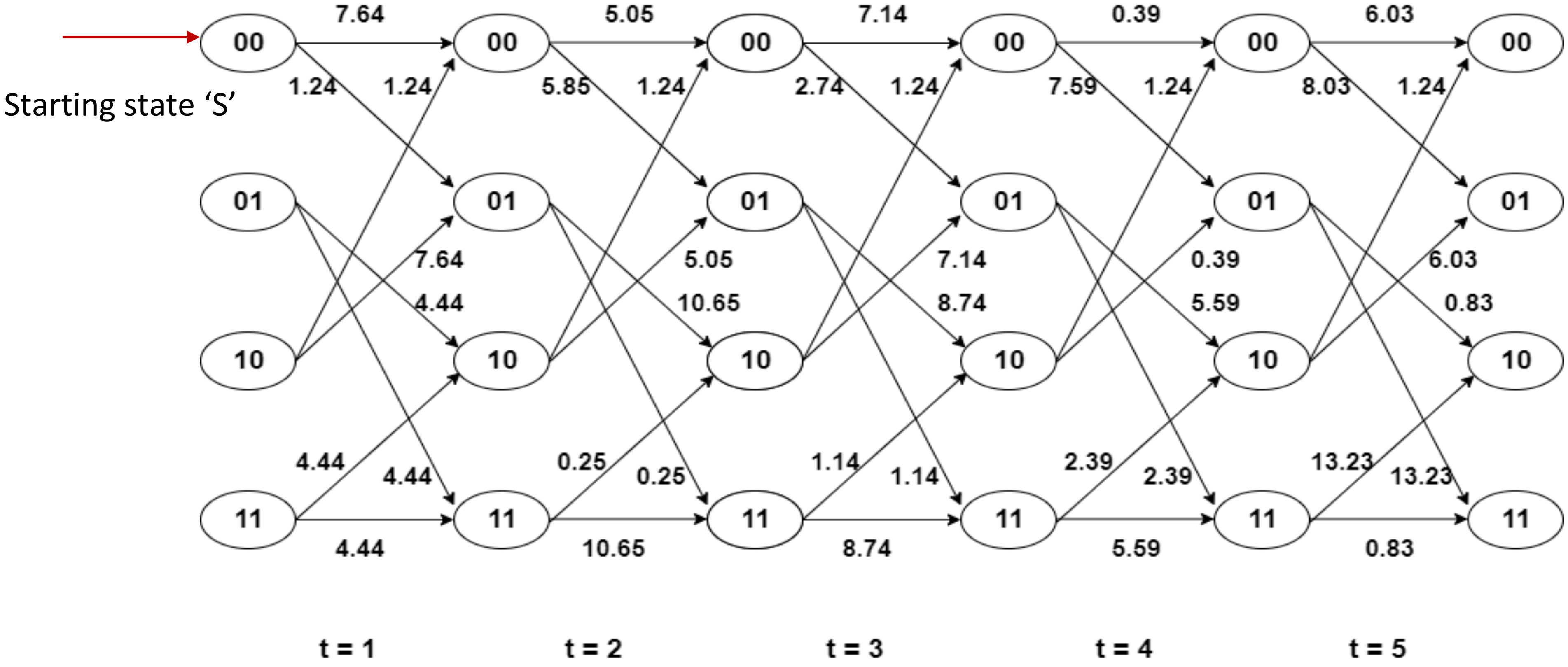
Branch metric = $|z_4 - x_4|^2 + |z_5 - x_5|^2 = 1.14$

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

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

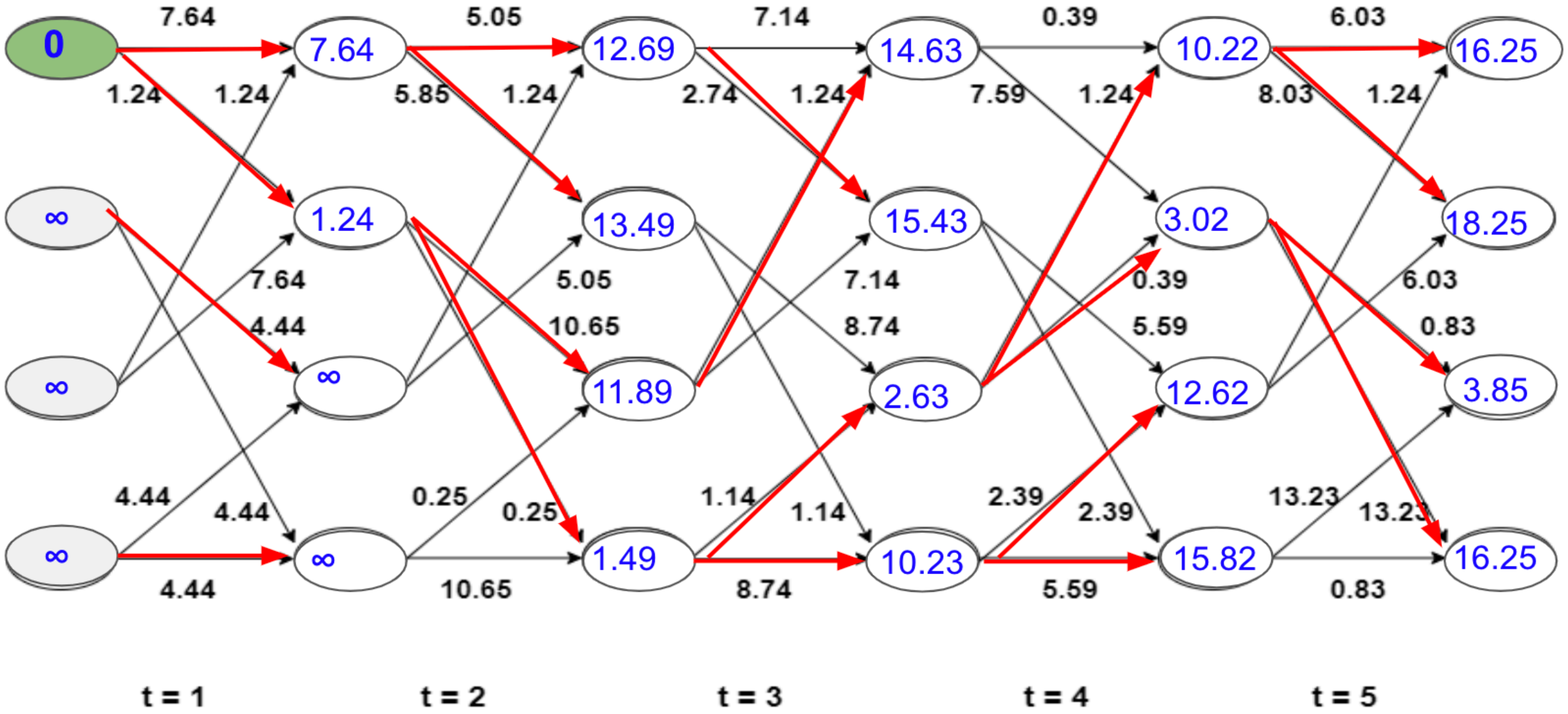
Viterbi Decoding (contd.)

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



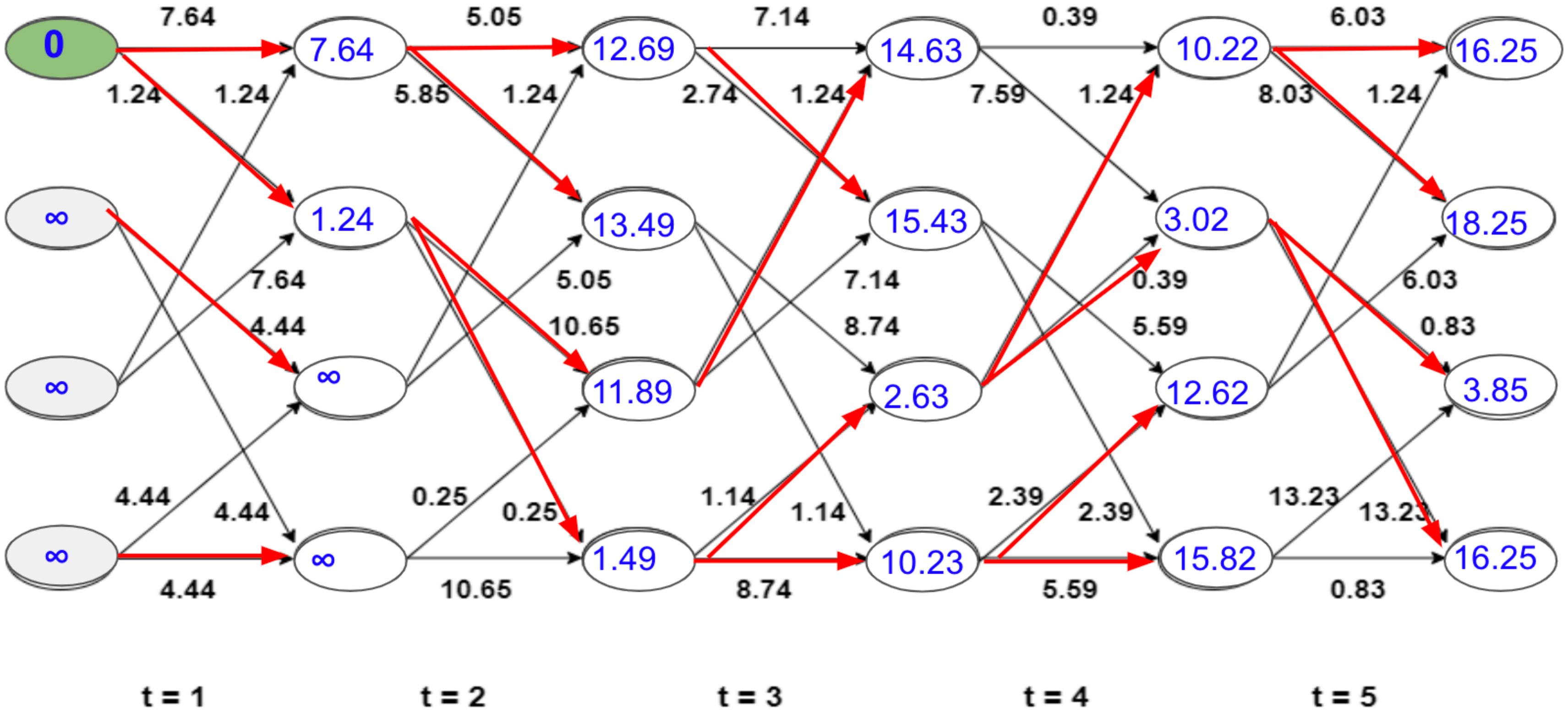
Viterbi Decoding (contd.)

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



Viterbi Decoding (contd.)

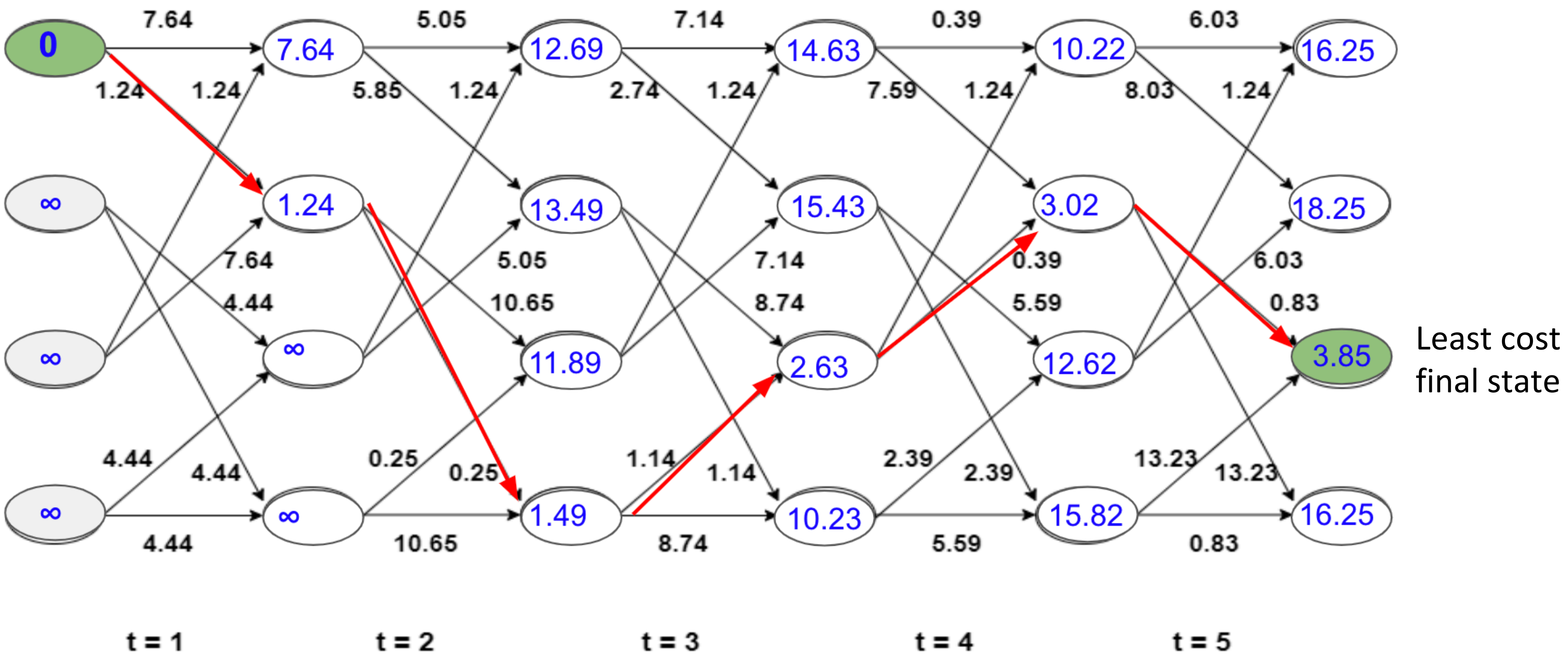
Step 5: Find cost of reaching t=5 for all states with 00 as starting state S



Least cost final state

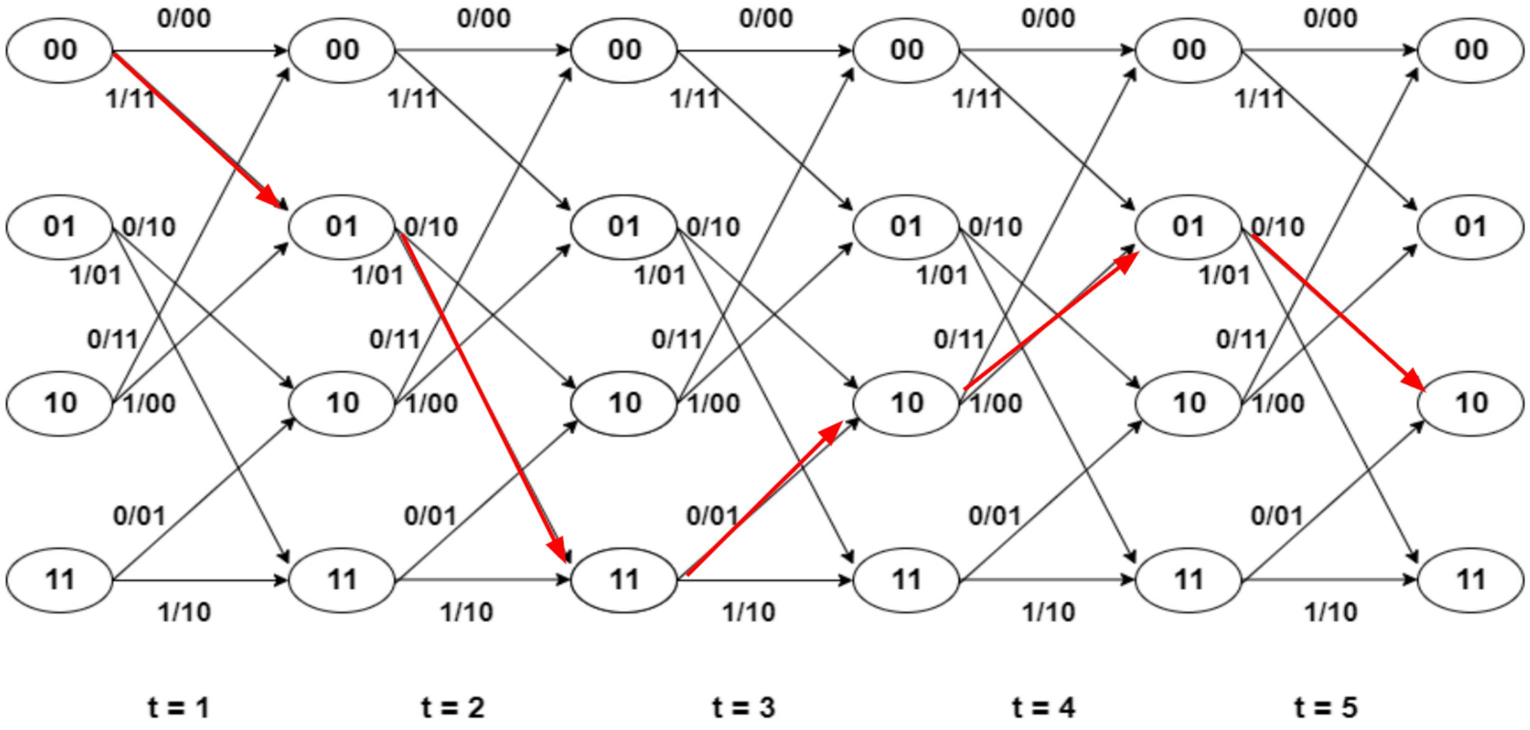
Viterbi Decoding (contd.)

Step 6: Traceback to find shortest path



Viterbi Decoding (contd.)

Step 7: Use Trellis with bits to get decoded bits



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

- Don’t submit executable files

- Due date: Feb 28, 11.59 pm EDT