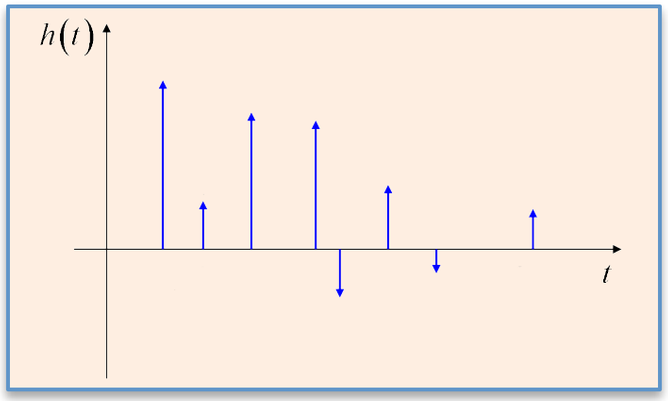
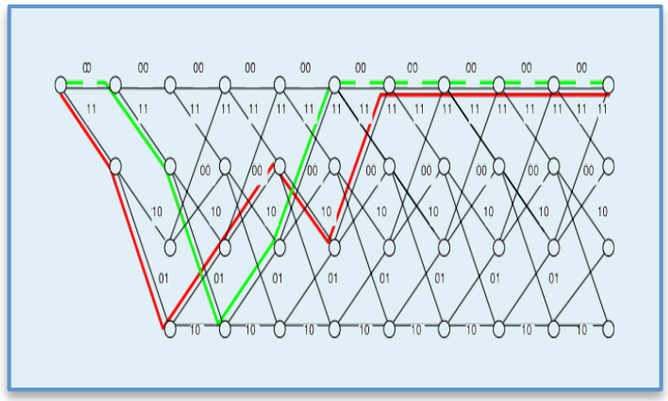
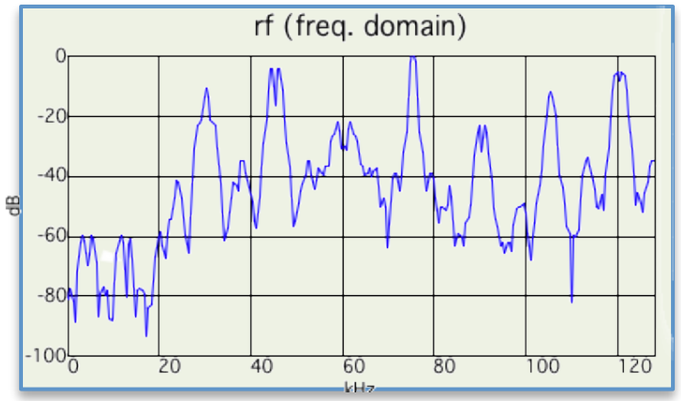
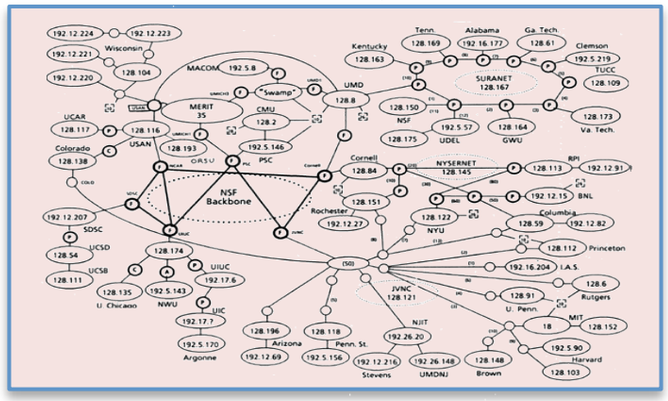








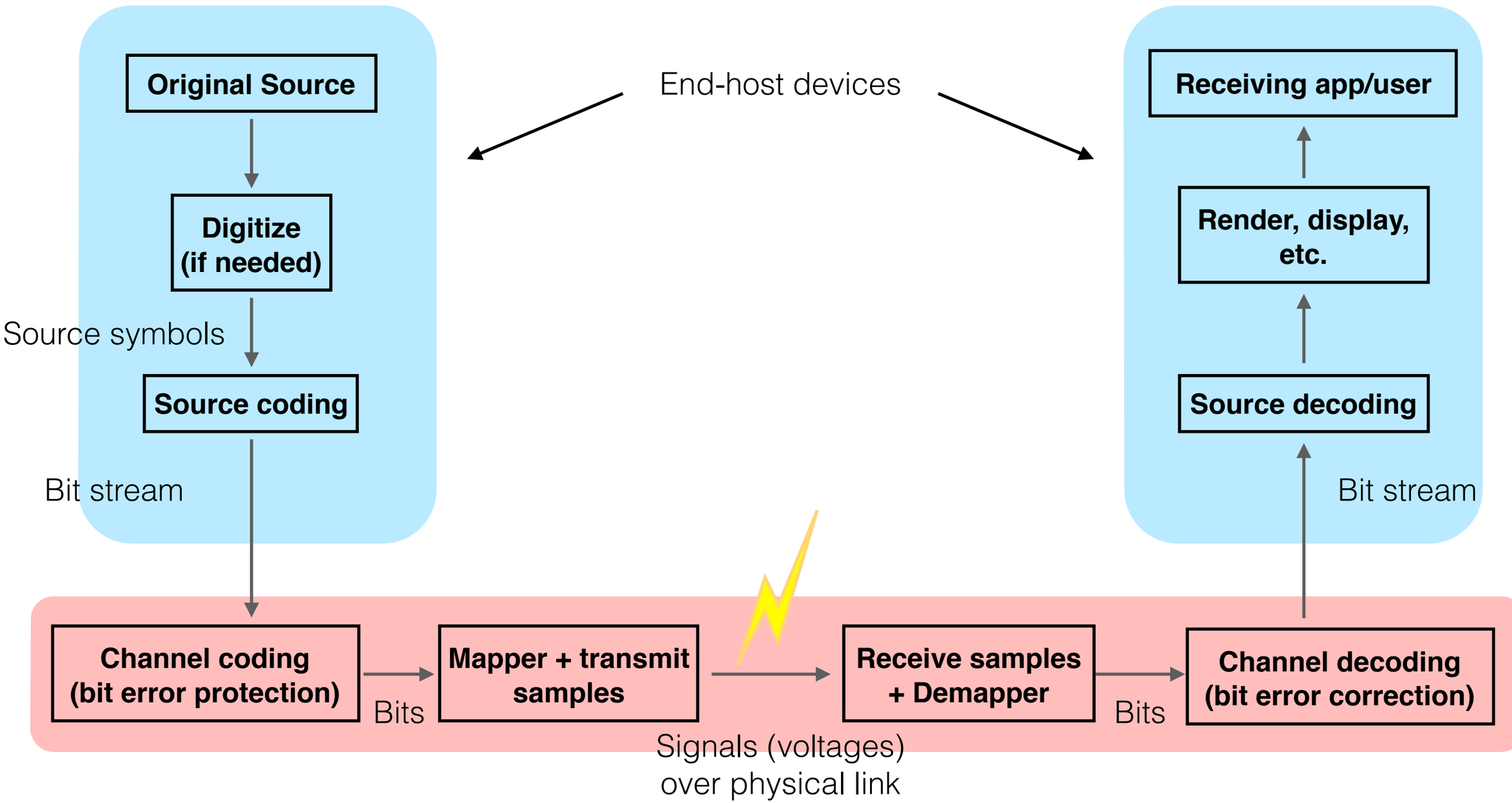
INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Fall 2014
Lecture #7

- Decoding convolutional codes with the Viterbi algorithm

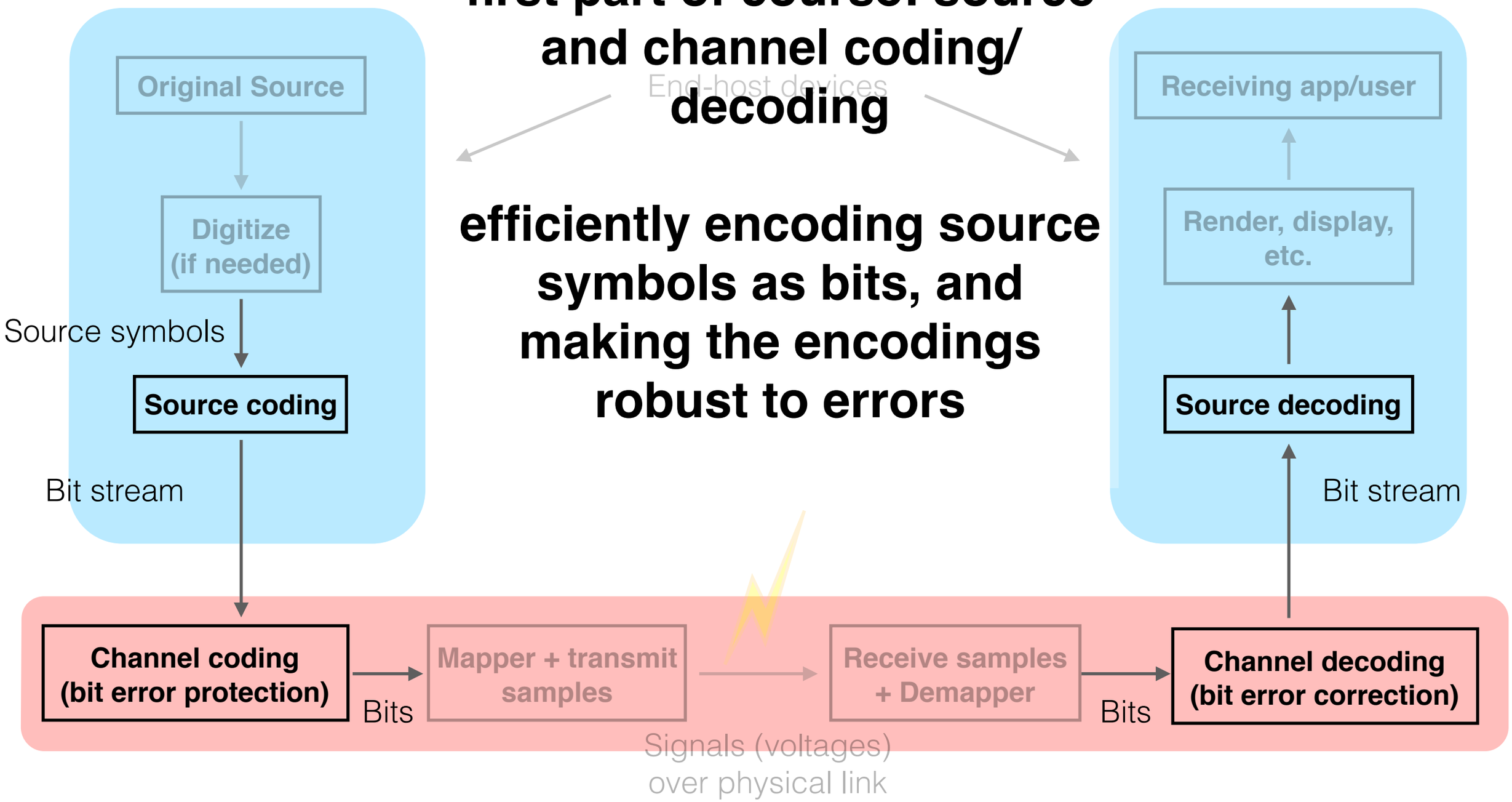
Single Link Communication Model



Single Link Communication Model

first part of course: source and channel coding/ decoding

efficiently encoding source symbols as bits, and making the encodings robust to errors

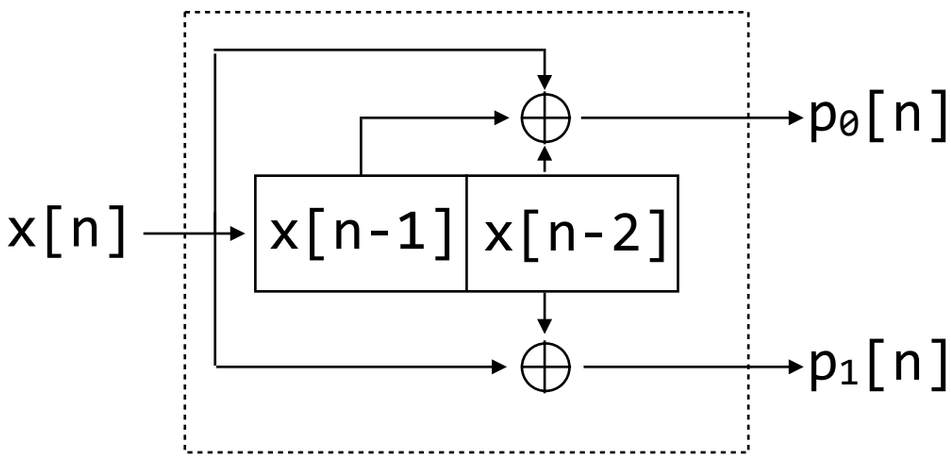


today's goal: efficiently decode convolutional codes

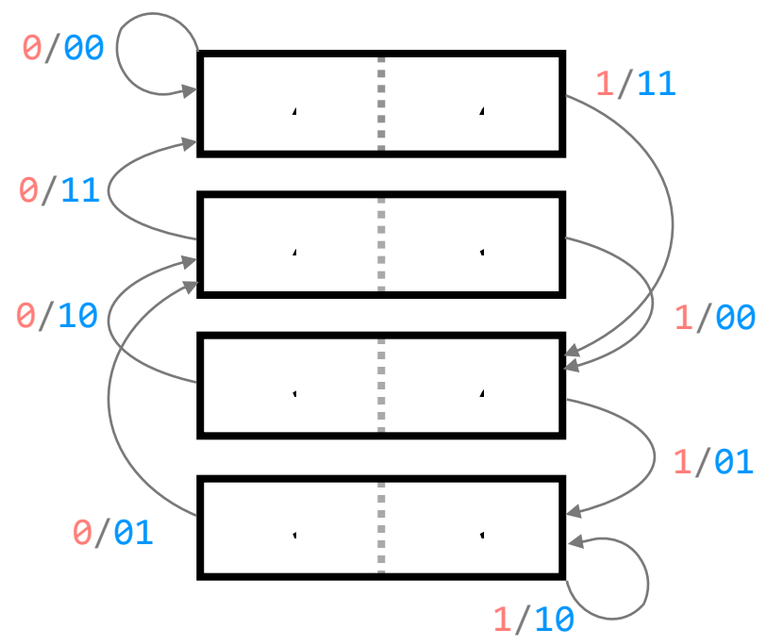
Convolutional Codes

1. Act on a **stream of message bits** into the encoder, rather than on discrete blocks of data

2. We **transmit only the parity bits**, not the data and parity bits.



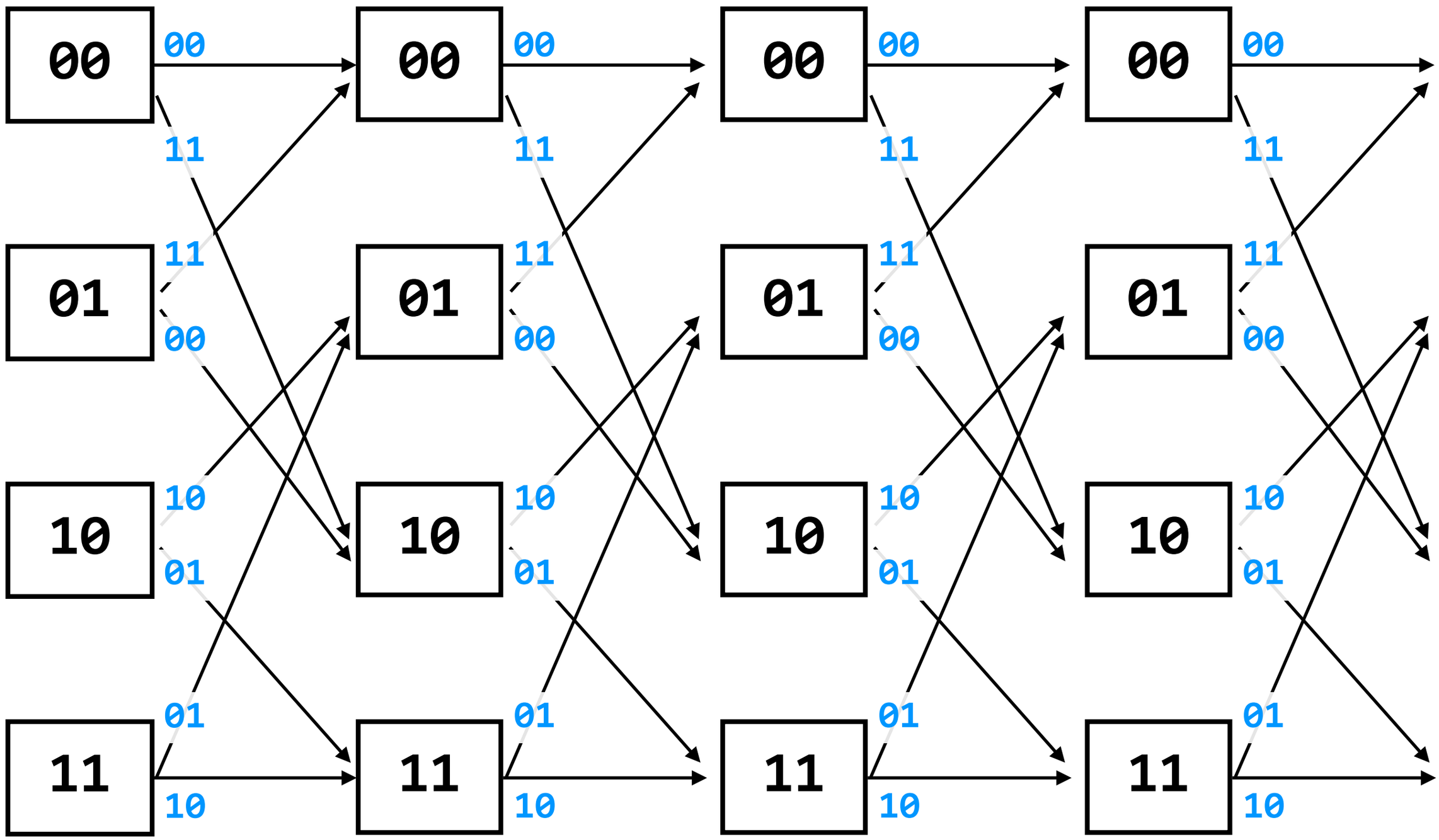
shift-register view



state-machine view

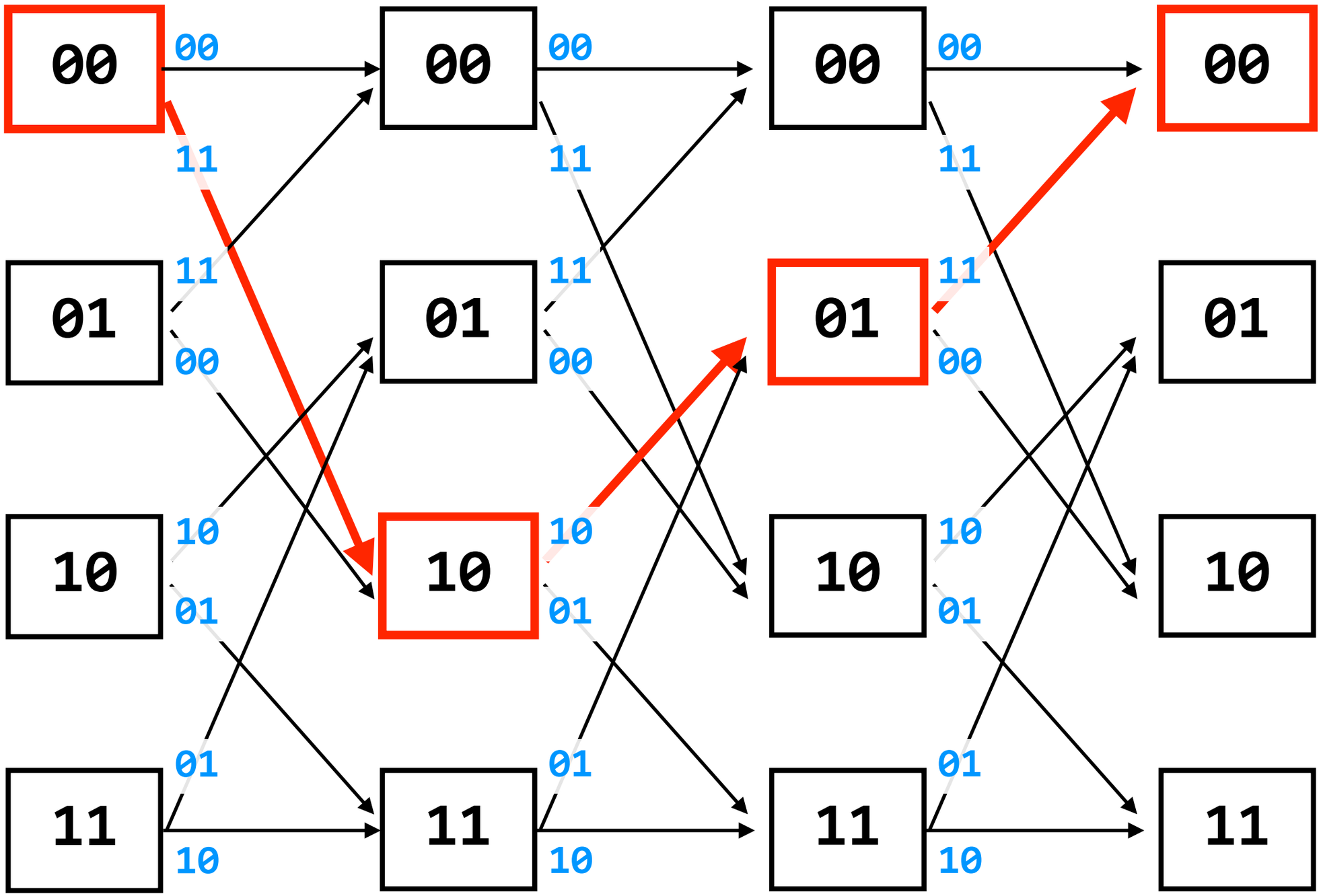
$$p_0[n] = x[n] + x[n-1] + x[n-2]$$
$$p_1[n] = x[n] + x[n-2]$$

Trellis View



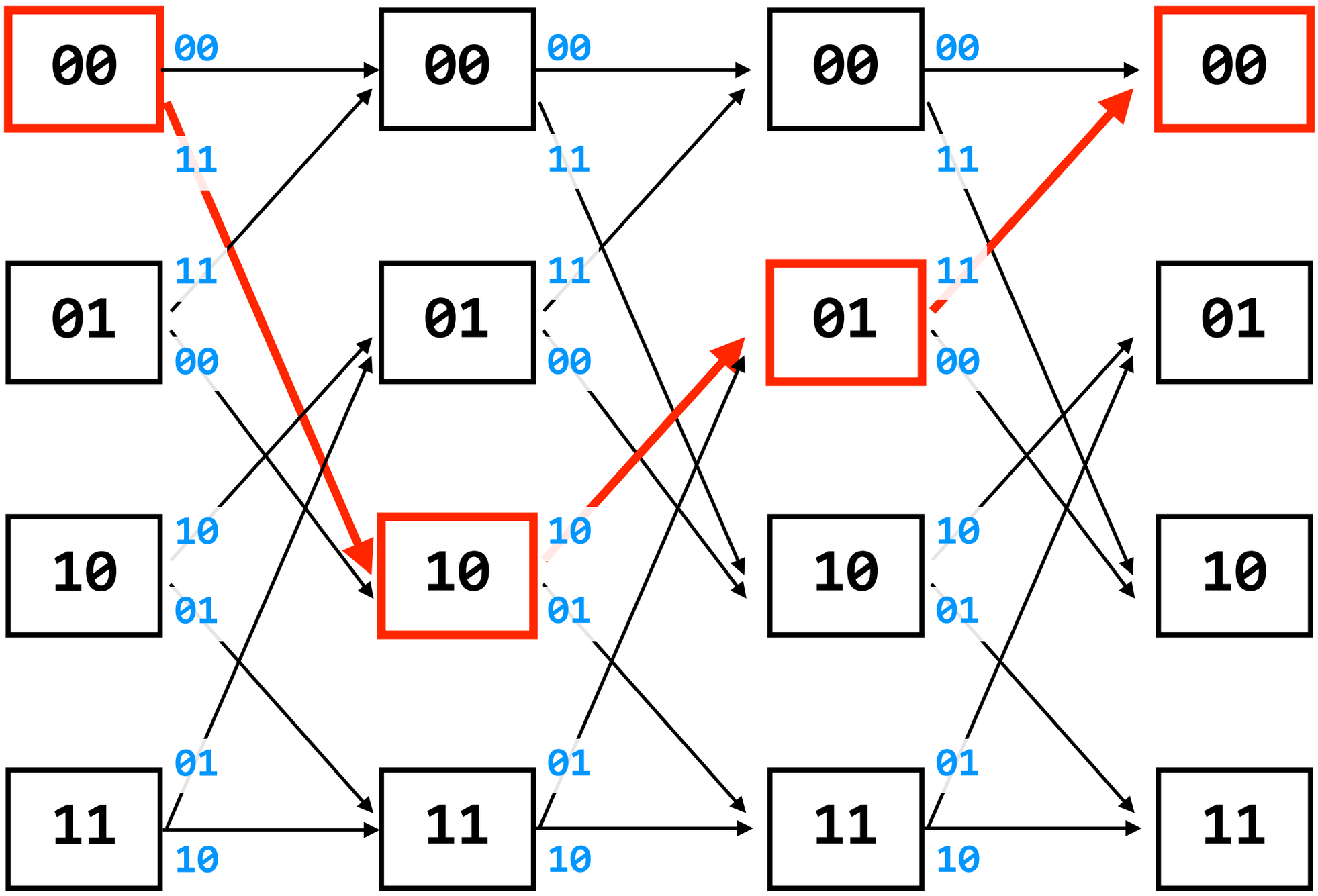
codeword: 11 10 11

message: 100



received codeword: 11 00 01
path's codeword: 11 10 11

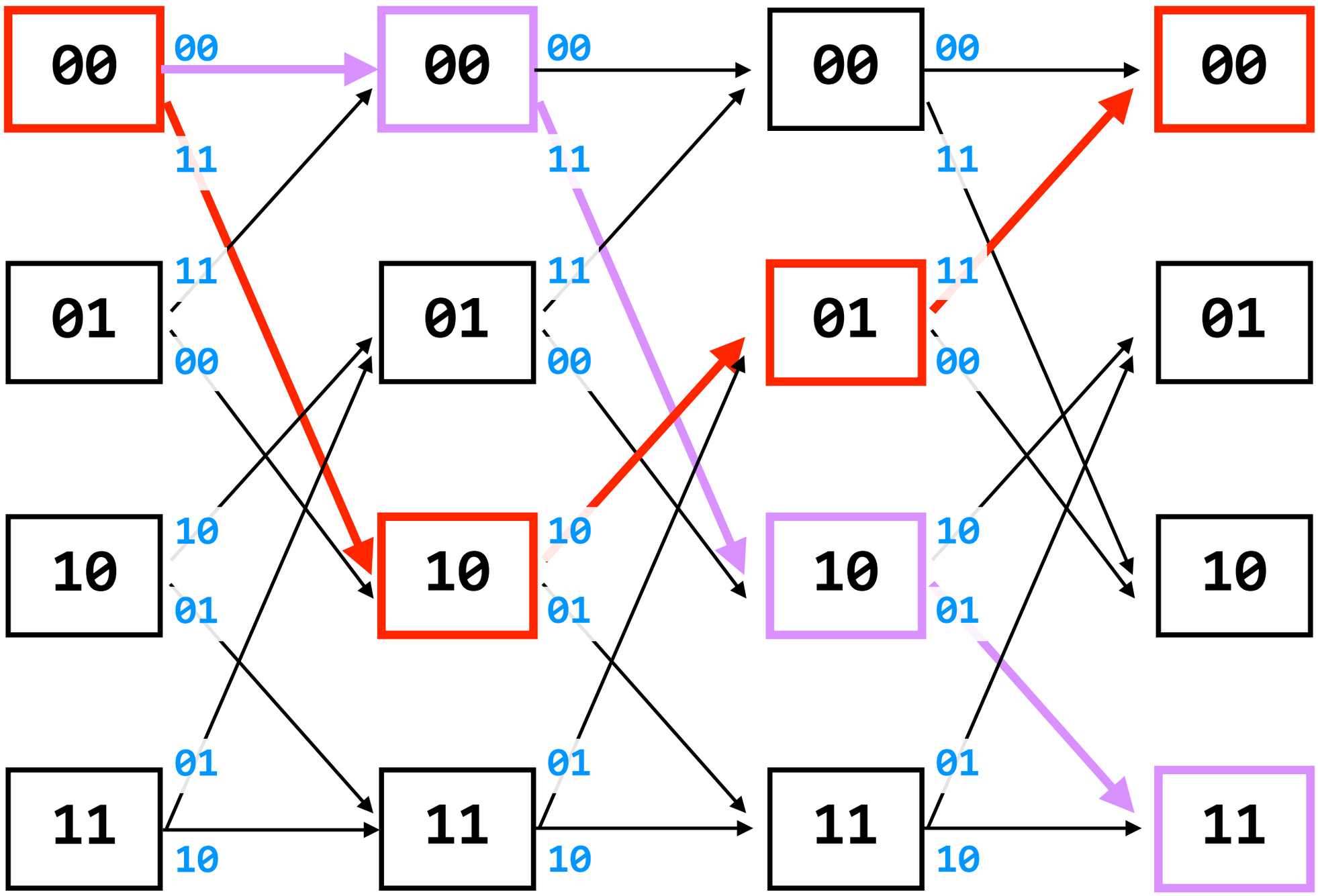
Hamming distance (110001, 111011) = 2



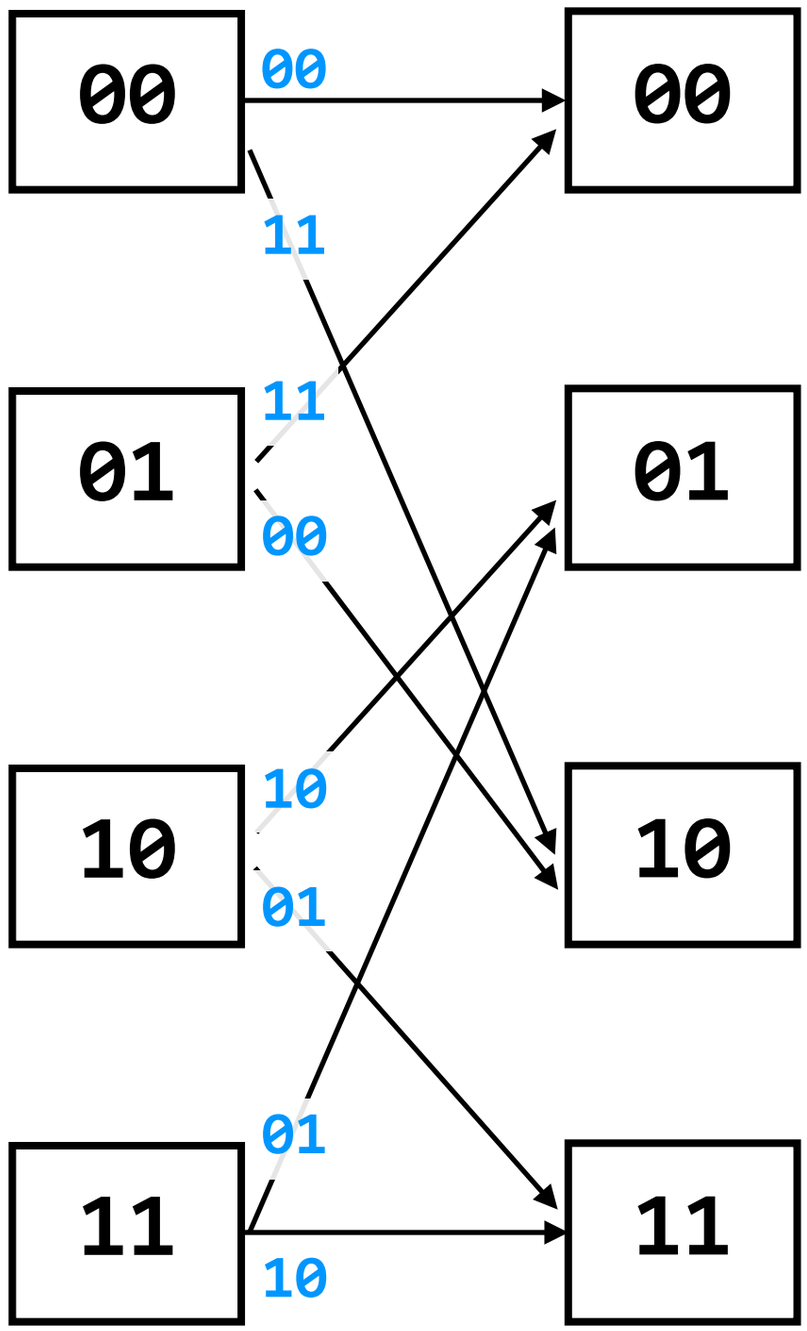
received codeword: 11 00 01

Hamming distance (110001, 111011) = 2
Hamming distance (110001, 001101) = 4

it's more likely that the correct codeword corresponds to the **red** path, not the **purple** path.



$$p_0[n] = x[n] + x[n-1] + x[n-2]$$
$$p_1[n] = x[n] + x[n-2]$$



branch metric: the distance between the received parity bits and the expected parity bits on each transition

path metric [s,i]: the smallest sum of branch metrics, minimized over all message sequences that place the transmitter at state **s** at time **i**

Viterbi Algorithm

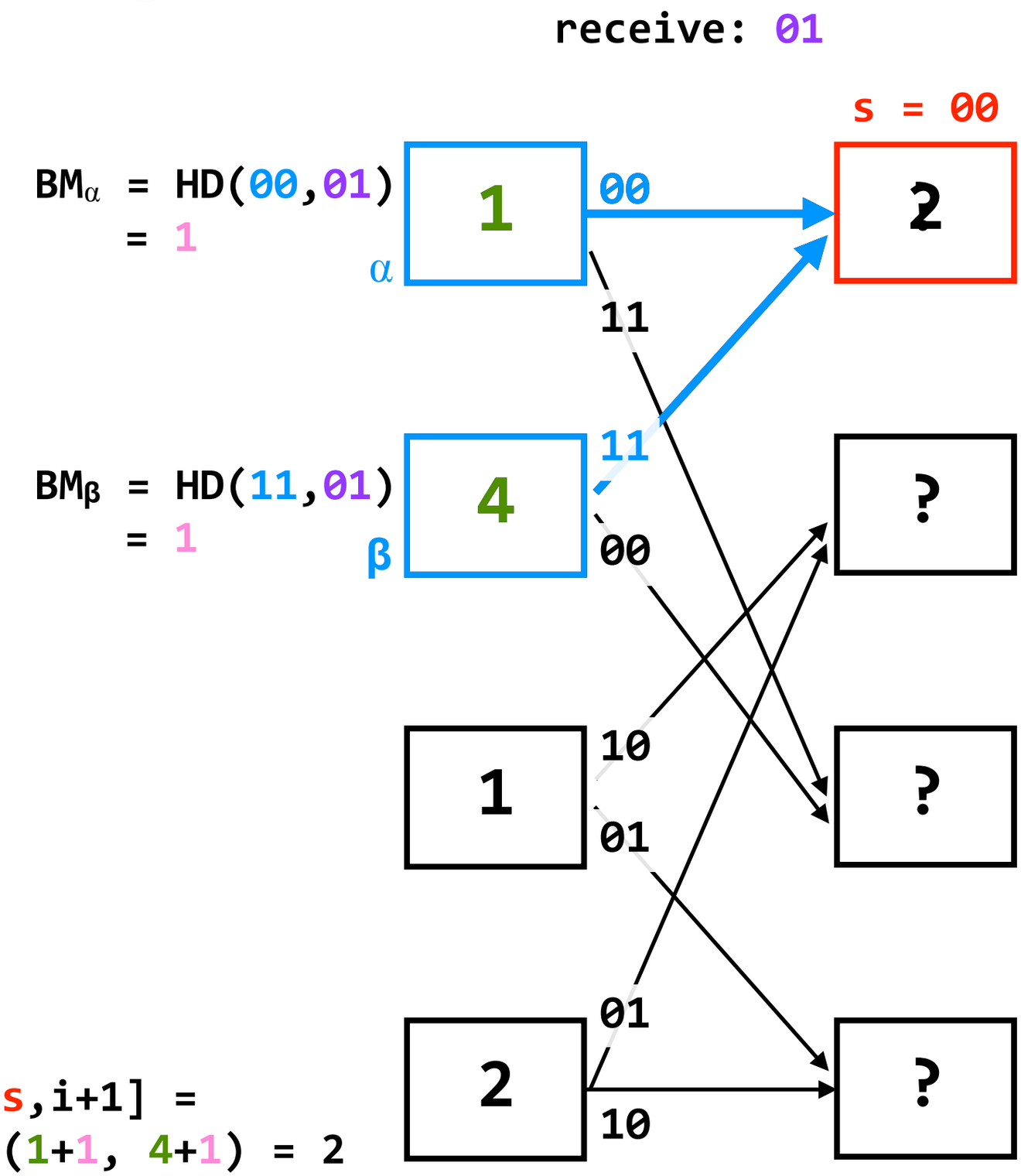
For each state s

1. Determine the predecessor states α and β

2. Calculate the branch metrics for the transitions α → s and β → s using the received parity bits

3. Determine PM[s,i+1] = min(PM[α,i] + BM_α, PM[β,i] + BM_β)

4. Keep track of which predecessor state was used



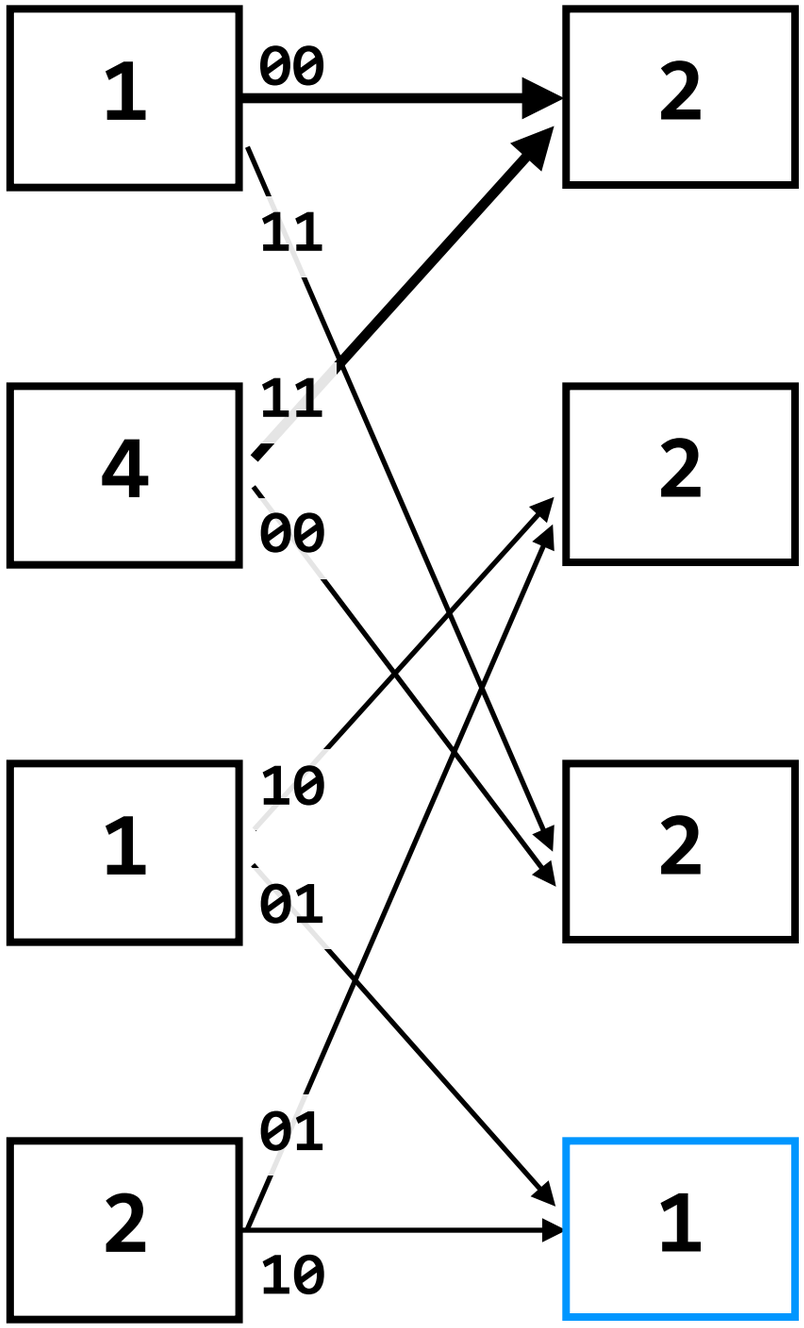
PM[s,i+1] =
min(1+1, 4+1) = 2

Viterbi Algorithm

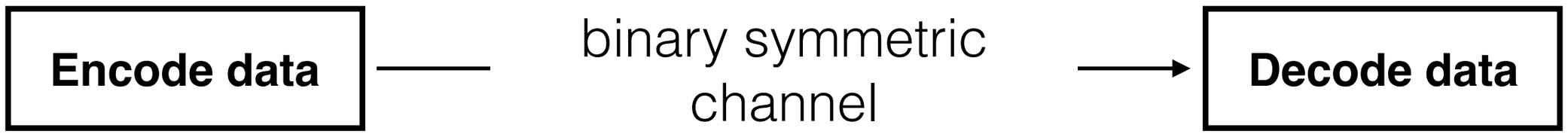
For each state s

1. Determine the predecessor states α and β
2. Calculate the branch metrics for the transitions $\alpha \rightarrow s$ and $\beta \rightarrow s$ using the received parity bits
3. Determine PM[s,i+1] = min(PM[α,i] + BM_α, PM[β,i] + BM_β)
4. Keep track of which predecessor state was used

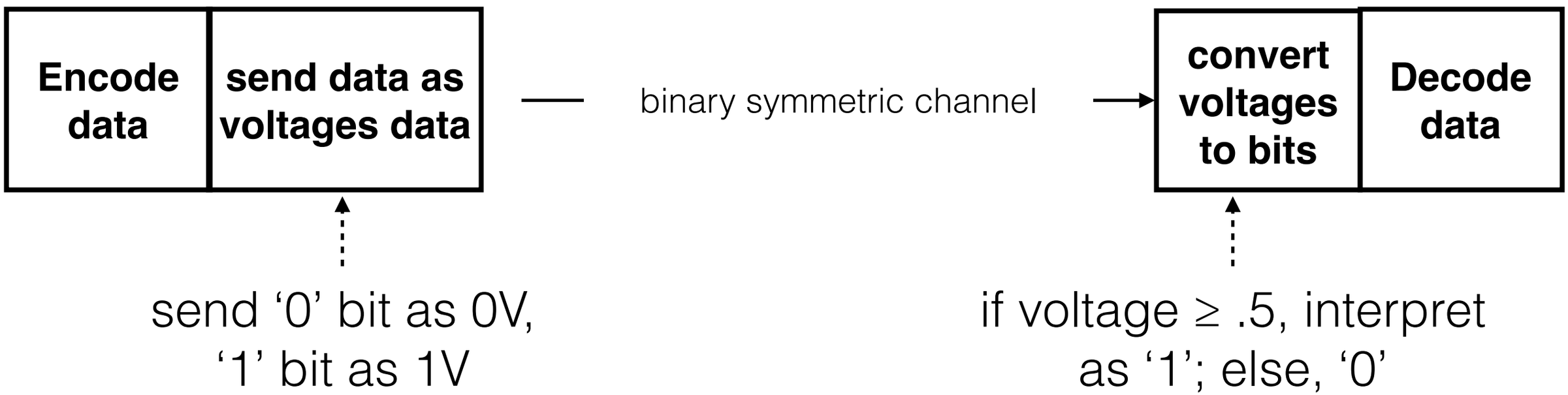
The ending state **s** with the smallest path metric is the most-likely ending state; to recover the message, we trace back the path from state **s**



Binary Symmetric Channel

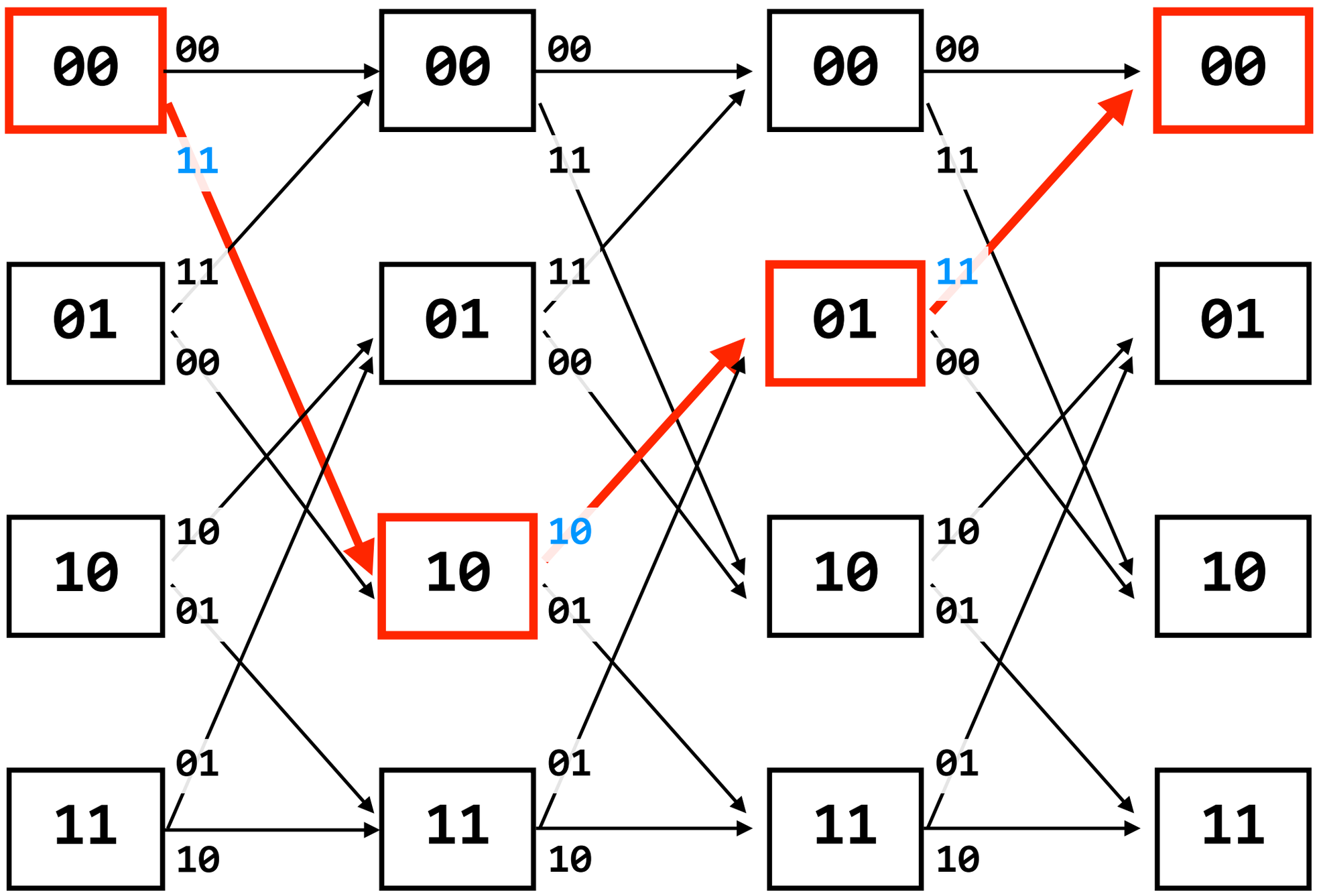


Binary Symmetric Channel



Free Distance

difference in path metrics between the all-zero output and the path with the smallest non-zero path metric going from 00 to 00



$d_{free} = 5$

Summary of 6.02 Part 1

- **Encoding data efficiently** via Huffman codes and LZW compression
- Adding **error-correcting capabilities** to our codes via linear block codes (e.g., Hamming codes) and convolutional codes
- **Decoding efficiently** with syndrome decoding (for linear block codes) and Viterbi decoding (for convolutional codes)