Modulation, Demodulation and Coding Course

Spring - 2015

Jeffrey N. Denenberg

Lecture 7b: Trellis decoding

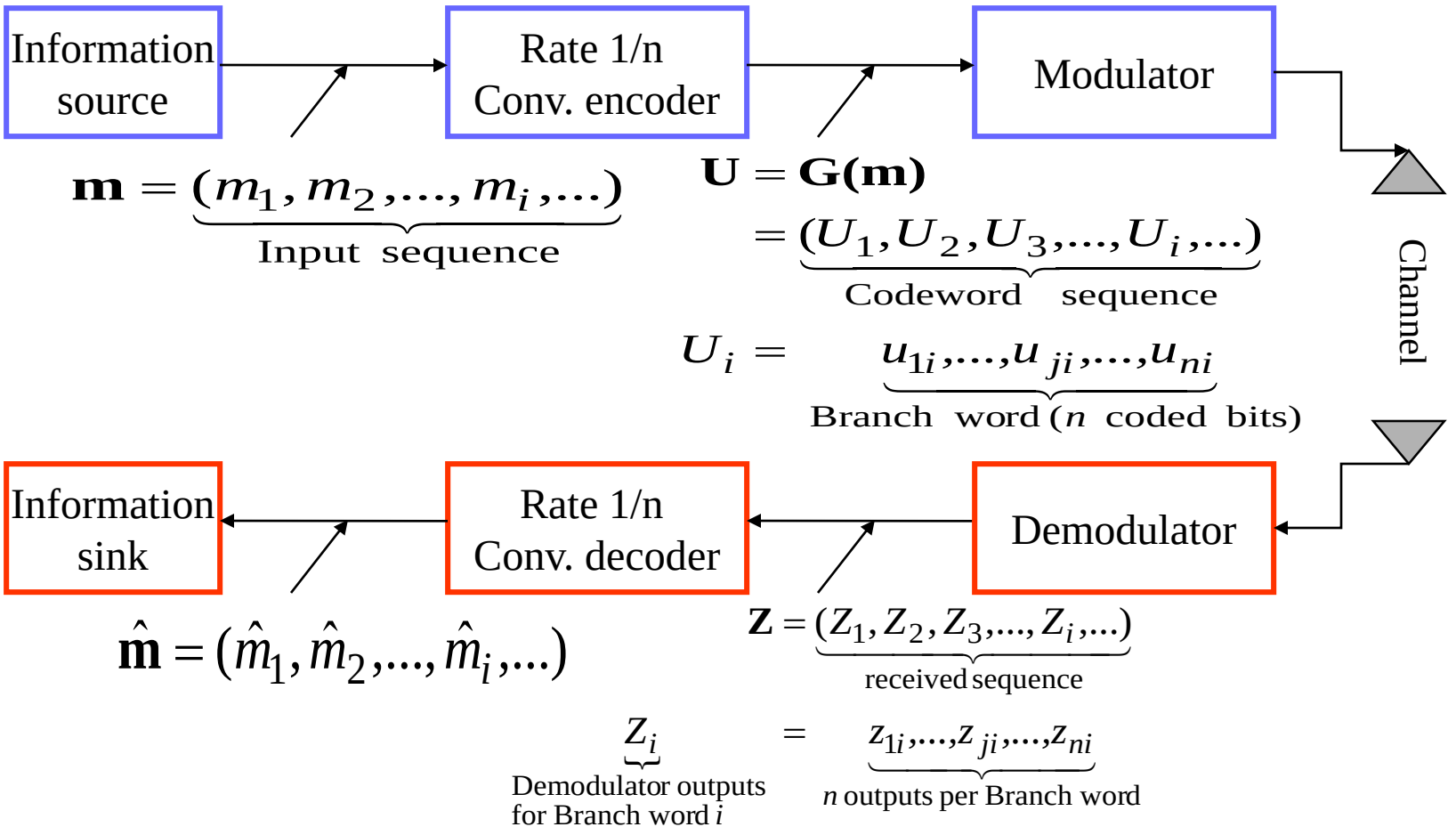
Last time, we talked about:

- Another class of linear codes, known as Convolutional codes.
- We studied the structure of the encoder and different ways for representing it.

Today, we are going to talk about:

- What are the state diagram and trellis representation of the code?
- How the decoding is performed for Convolutional codes?
- What is a Maximum likelihood decoder?
- What are the soft decisions and hard decisions?
- How does the Viterbi algorithm work?

Block diagram of the DCS



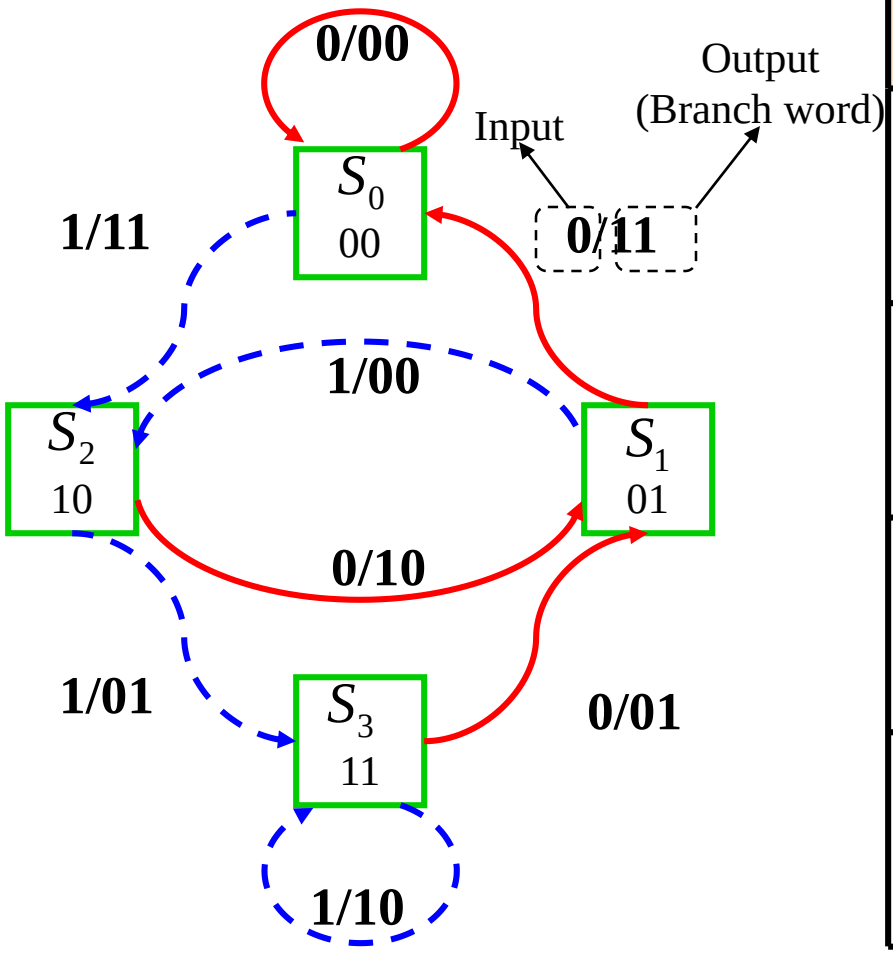
State diagram

- A finite-state machine only encounters a finite number of states.
- *State of a machine:* the smallest amount of information that, together with a current input to the machine, can predict the output of the machine.
- In a Convolutional encoder, the state is represented by the content of the memory.
- Hence, there are 2^{K-1} states.

State diagram – cont'd

- A state diagram is a way to represent the encoder.
- A state diagram contains all the states and all possible transitions between them.
- There can be only two transitions initiating from a state.
- There can be only two transitions ending up in a state.

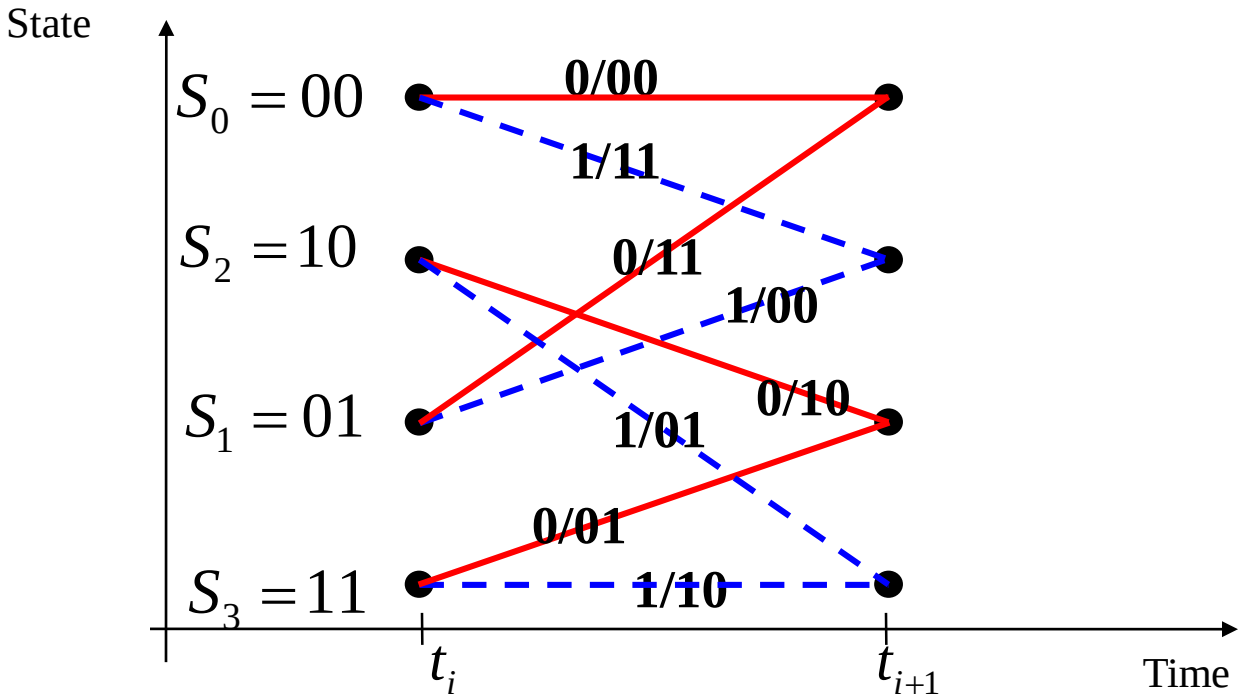
State diagram – cont'd



Current state	input	Next state	output
S_0 00	0	S_0	00
	1	S_2	11
S_1 01	0	S_0	11
	1	S_2	00
S_2 10	0	S_1	10
	1	S_3	01
S_3 11	0	S_1	01
	1	S_3	10

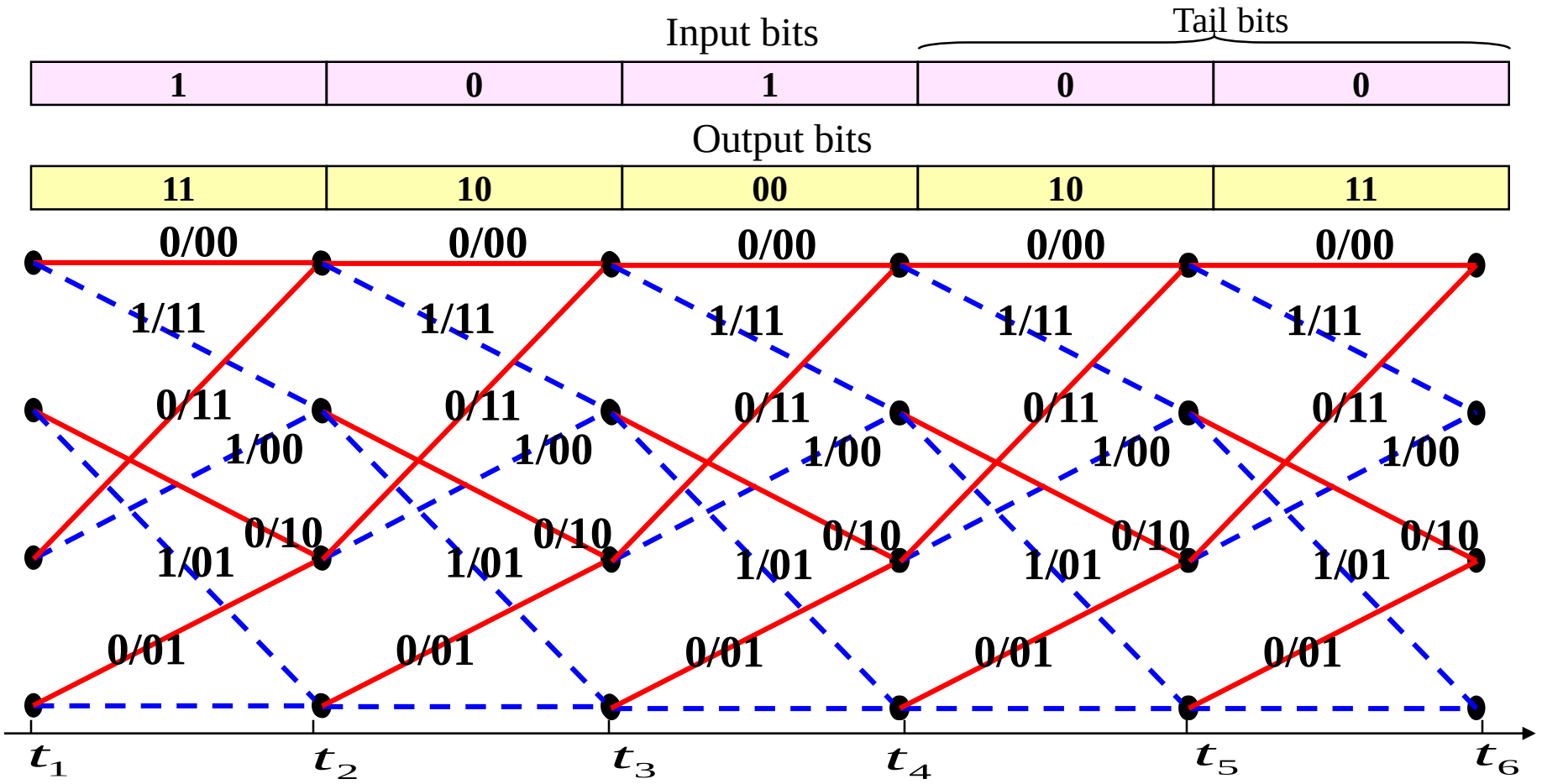
Trellis – cont'd

- The Trellis diagram is an extension of the state diagram that shows the passage of time.
  - Example of a section of trellis for the rate ½ code

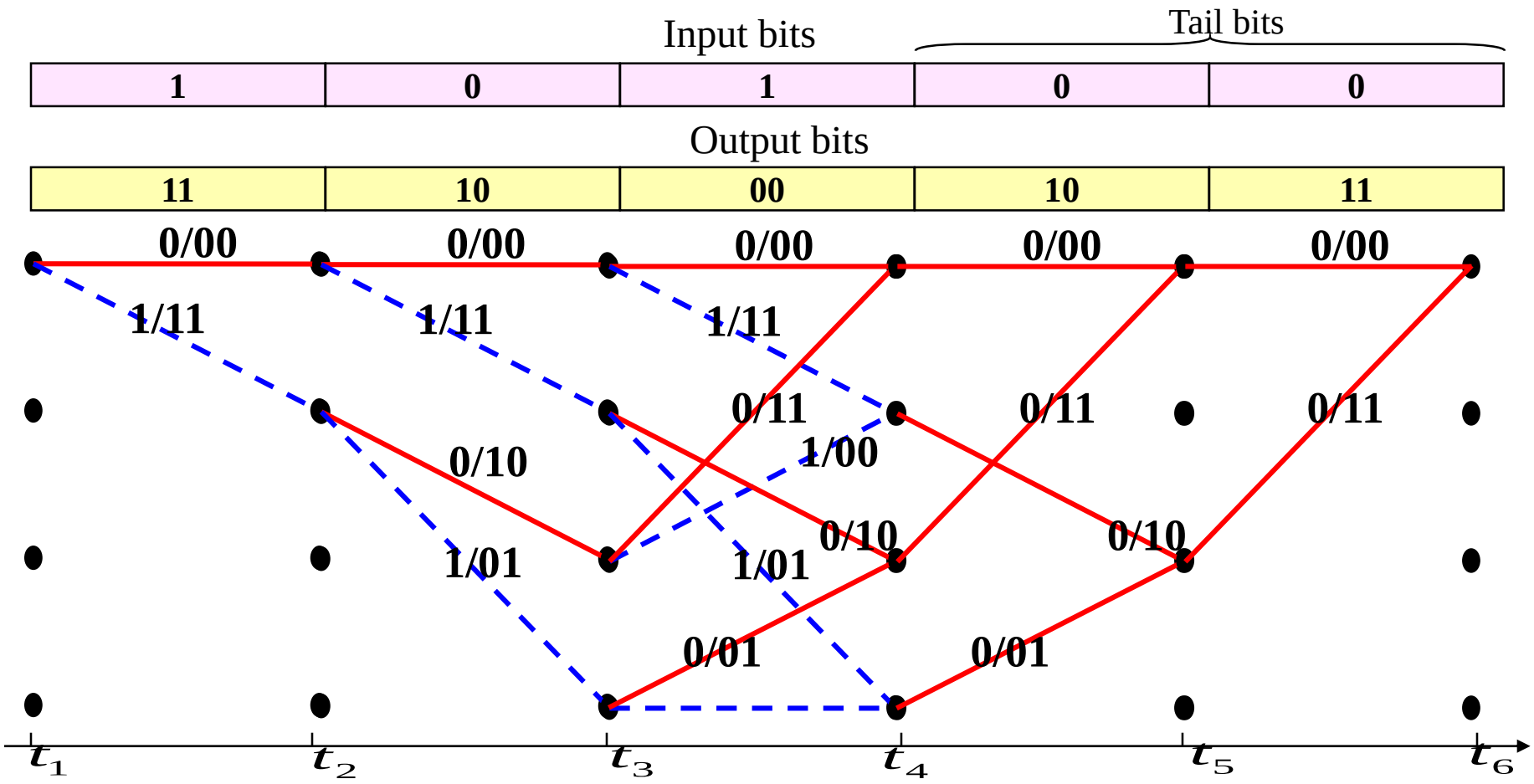


Trellis –cont'd

- A trellis diagram for the example code



Trellis – cont'd



Optimum decoding

- If the input sequence messages are equally likely, the optimum decoder which minimizes the probability of error is the *Maximum likelihood* decoder.

- The ML decoder, selects a codeword among all the possible codewords which maximizes the likelihood function $p(\mathbf{Z} \mid \mathbf{U}^{(m')})$ where $\mathbf{Z}$ is the received sequence and $\mathbf{U}^{(m')}$ is one of the possible codewords:

2^L codewords to search!!!

➢ML decoding rule:

Choose $\mathbf{U}^{(m')}$ if $p(\mathbf{Z} \mid \mathbf{U}^{(m')}) = \max_{\text{over all } \mathbf{U}^{(m)}} p(\mathbf{Z} \mid \mathbf{U}^{(m)})$

ML decoding for memory-less channels

- Due to the independent channel statistics for memoryless channels, the likelihood function becomes

$$p(\mathbf{Z}\,|\,\mathbf{U}^{(m)}) = p_{z_1,z_2,\ldots,z_i,\ldots}(Z_1,Z_2,\ldots,Z_i,\ldots|U^{(m)}) = \prod_{i=1}^{\infty} p(Z_i\,|\,U_i^{(m)}) = \prod_{i=1}^{\infty}\prod_{j=1}^{n} p(z_{ji}\,|\,u_{ji}^{(m)})$$

and equivalently, the log-likelihood function becomes

$$\gamma_{\mathbf{U}}(m) = \underbrace{\log p(\mathbf{Z}\,|\,\mathbf{U}^{(m)})}_{\text{Path metric}} = \underbrace{\sum_{i=1}^{\infty}\log p(Z_i\,|\,U_i^{(m)})}_{\text{Branch metric}} = \sum_{i=1}^{\infty}\underbrace{\sum_{j=1}^{n}\log p(z_{ji}\,|\,u_{ji}^{(m)})}_{\text{Bit metric}}$$

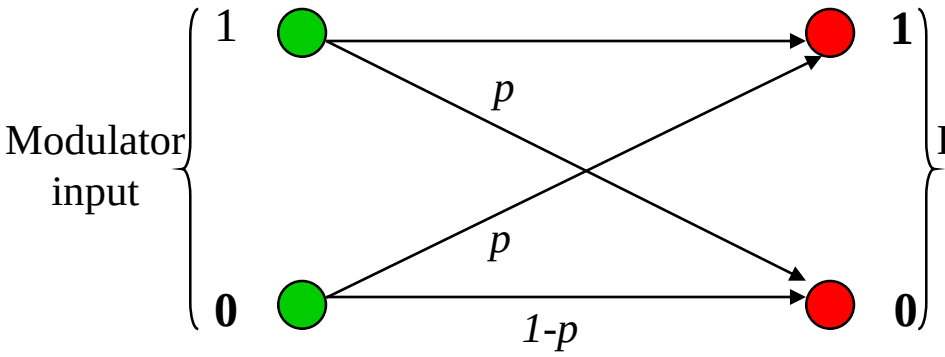
- The path metric up to time index "i", is called the partial path metric.

➢ML decoding rule:

Choose the path with maximum metric among all the paths in the trellis.

This path is the "closest" path to the transmitted sequence.

Binary symmetric channels (BSC)



$$p = p(1|0) = p(0|1)$$
$$1-p = p(1|1) = p(0|0)$$

- If $d_m = d(\mathbf{Z}, \mathbf{U}^{(m)})$ is the Hamming distance between $\mathbf{Z}$ and $\mathbf{U}$, then

$$p(\mathbf{Z}|\mathbf{U}^{(m)}) = p^{d_m}(1-p)^{L_n - d_m}$$

Size of coded sequence

$$\gamma_{\mathbf{U}}(m) = -d_m \log\left(\frac{1-p}{p}\right) + L_n \log(1-p)$$

➢ML decoding rule:

Choose the path with minimum Hamming distance from the received sequence.

AWGN channels

- For BPSK modulation the transmitted sequence corresponding to the codeword $\mathbf{U}^{(m)}$ is denoted by where $\mathbf{s}^{(m)} = (S_1^{(m)}, S_2^{(m)}, ..., S_i^{(m)}, ...)$ and $S_i^{(m)} = (s_{1i}^{(m)}, ..., s_{ji}^{(m)}, ..., s_{ni}^{(m)})$ and $s_{ij} = \pm\sqrt{E_c}$.
- The log-likelihood function becomes

$$\gamma_{\mathbf{U}}(m) = \sum_{i=1}^{\infty}\sum_{j=1}^{n} z_{ji} s_{ji}^{(m)} = < \mathbf{Z}, \mathbf{S}^{(m)} >$$

Inner product or correlation between **Z** and **S**

  - Maximizing the correlation is equivalent to minimizing the Euclidean distance.

> ➢ML decoding rule:
> Choose the path which with minimum Euclidean distance to the received sequence.

Soft and hard decisions

- In hard decision:
  - The demodulator makes a firm or hard decision whether a one or a zero was transmitted and provides no other information for the decoder such as how reliable the decision is.
  - Hence, its output is only zero or one (the output is quantized only to two level) that are called "hard-bits".
- Decoding based on hard-bits is called the "hard-decision decoding".

Soft and hard decision-cont'd

- In Soft decision:
  - The demodulator provides the decoder with some side information together with the decision.
  - The side information provides the decoder with a measure of confidence for the decision.
  - The demodulator outputs which are called soft-bits, are quantized to more than two levels.
- Decoding based on soft-bits, is called the "soft-decision decoding".
- On AWGN channels, a 2 dB and on fading channels a 6 dB gain are obtained by using soft-decoding instead of hard-decoding.

The Viterbi algorithm

- The Viterbi algorithm performs Maximum likelihood decoding.
- It finds a path through the trellis with the largest metric (maximum correlation or minimum distance).
  - It processes the demodulator outputs in an iterative manner.
  - At each step in the trellis, it compares the metric of all paths entering each state, and keeps only the path with the smallest metric, called the survivor, together with its metric.
  - It proceeds in the trellis by eliminating the least likely paths.
- It reduces the decoding complexity to $L2^{K-1}$!

The Viterbi algorithm - cont'd

- Viterbi algorithm:

A. Do the following set up:
   - For a data block of L bits, form the trellis. The trellis has $L+K-1$ sections or levels and starts at time t_1 and ends up at time t_{L+K} .
   - Label all the branches in the trellis with their corresponding branch metric.
   - For each state in the trellis at the time t_i which is denoted by $S(t_i) \in \{0,1,\ldots,2^{K-1}\}$, define a parameter $\Gamma(S(t_i), t_i)$

B. Then, do the following:

The Viterbi algorithm - cont'd

1. Set $\Gamma(0, t_1) = 0$ and $i = 2$.
2. At time t_i , compute the partial path metrics for all the paths entering each state.
3. Set $\Gamma(S(t_i), t_i)$ equal to the best partial path metric entering each state at time t_i

   Keep the survivor path and delete the dead paths from the trellis.

1. If $i < L + K$, increase i by 1 and return to step 2.

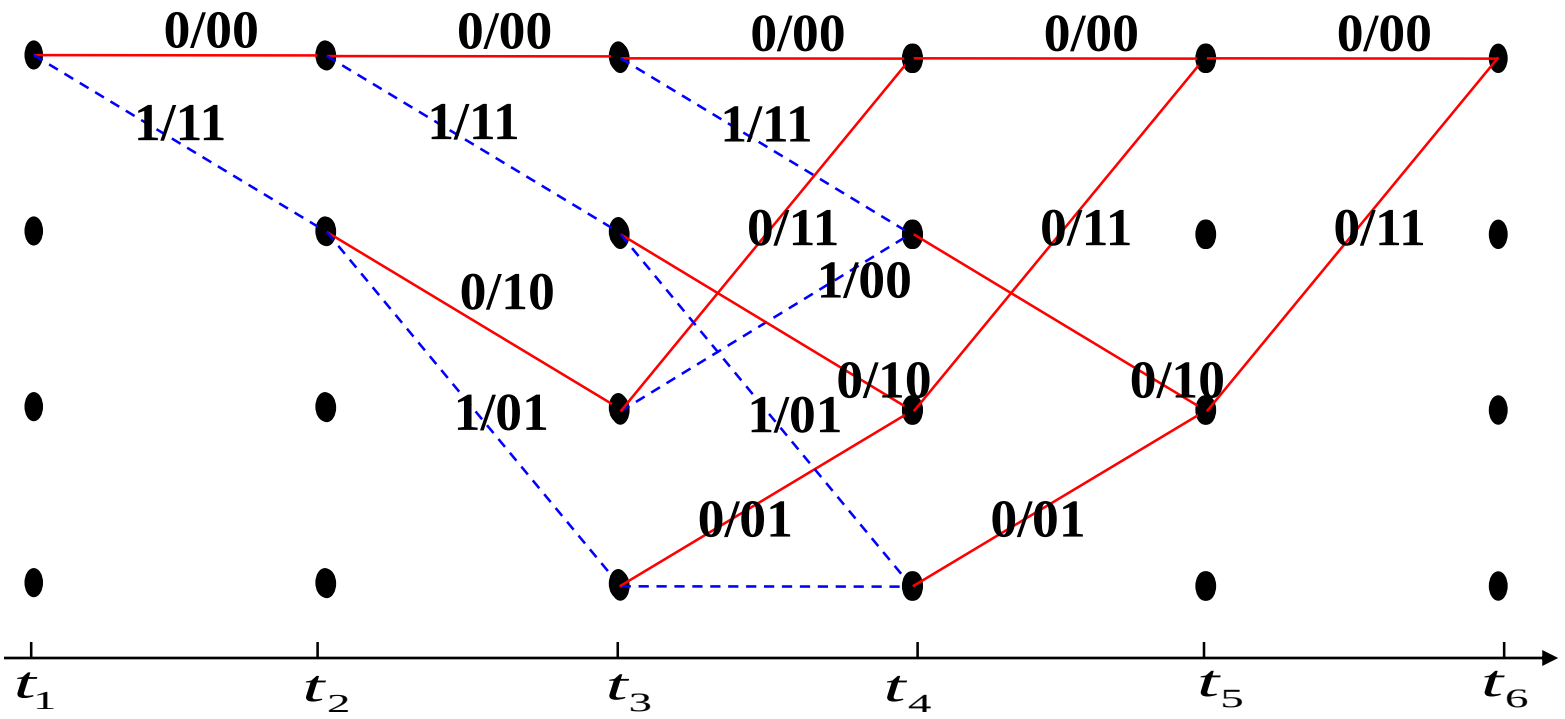
A. Start at state zero at time t_{L+K} . Follow the surviving branches backwards through the trellis. The path found is unique and corresponds to the ML codeword.

Example of Hard decision Viterbi decoding

$$\mathbf{m} = (101)$$

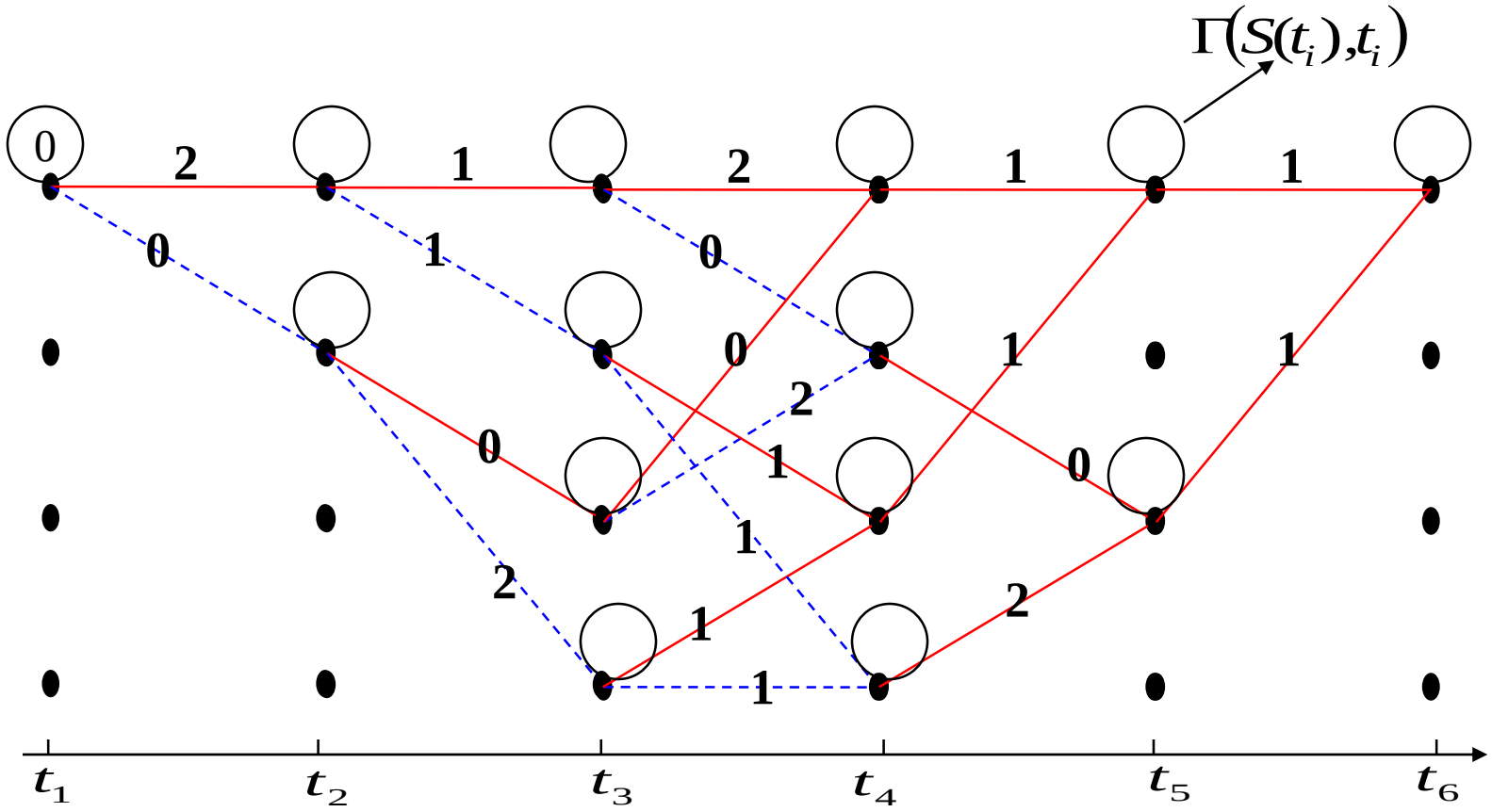
$$\mathbf{U} = (11 \quad 10 \quad 00 \quad 10 \quad 11)$$

$$\mathbf{Z} = (11 \quad 10 \quad 11 \quad 10 \quad 01)$$



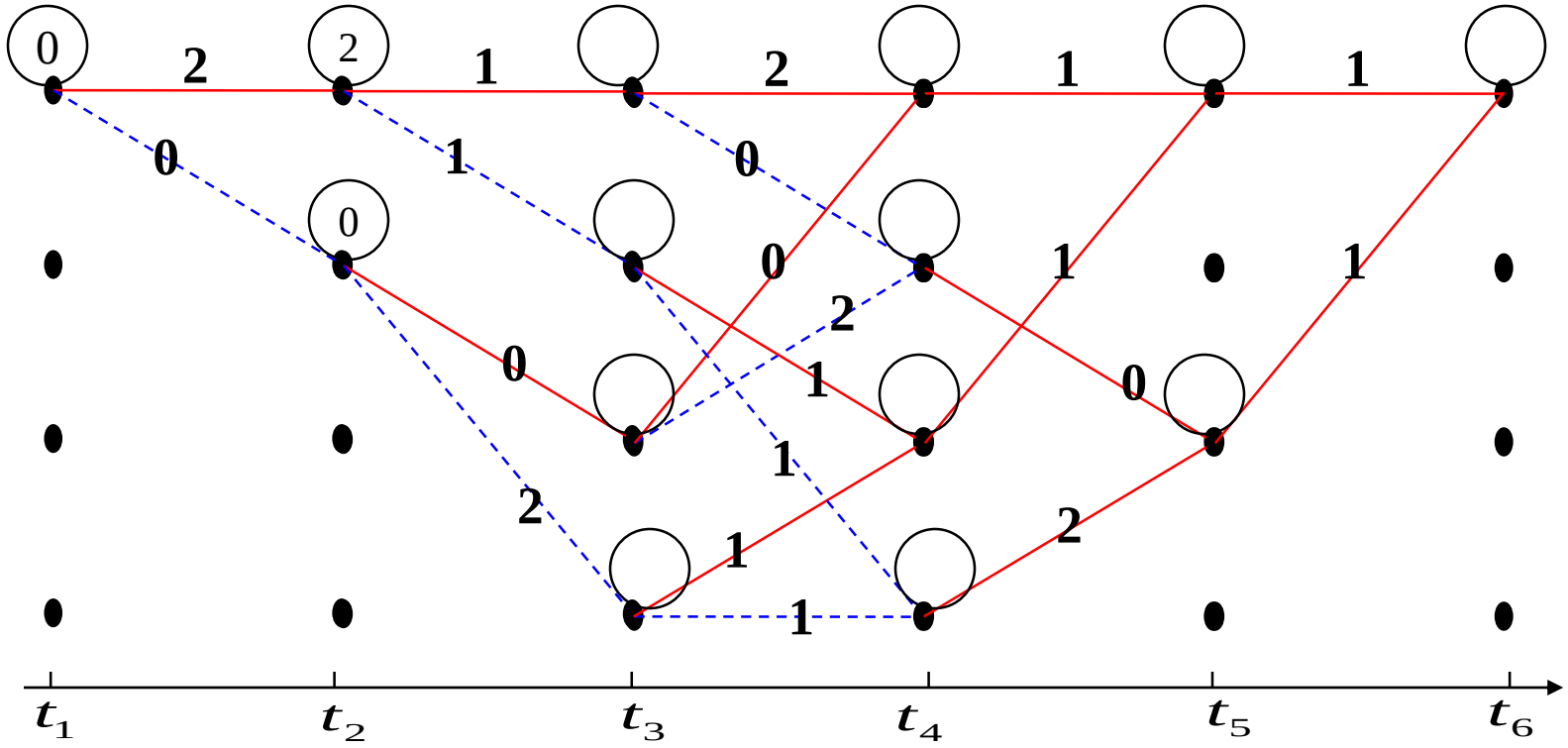
Example of Hard decision Viterbi decoding-cont'd

- Label all the branches with the branch metric (Hamming distance)



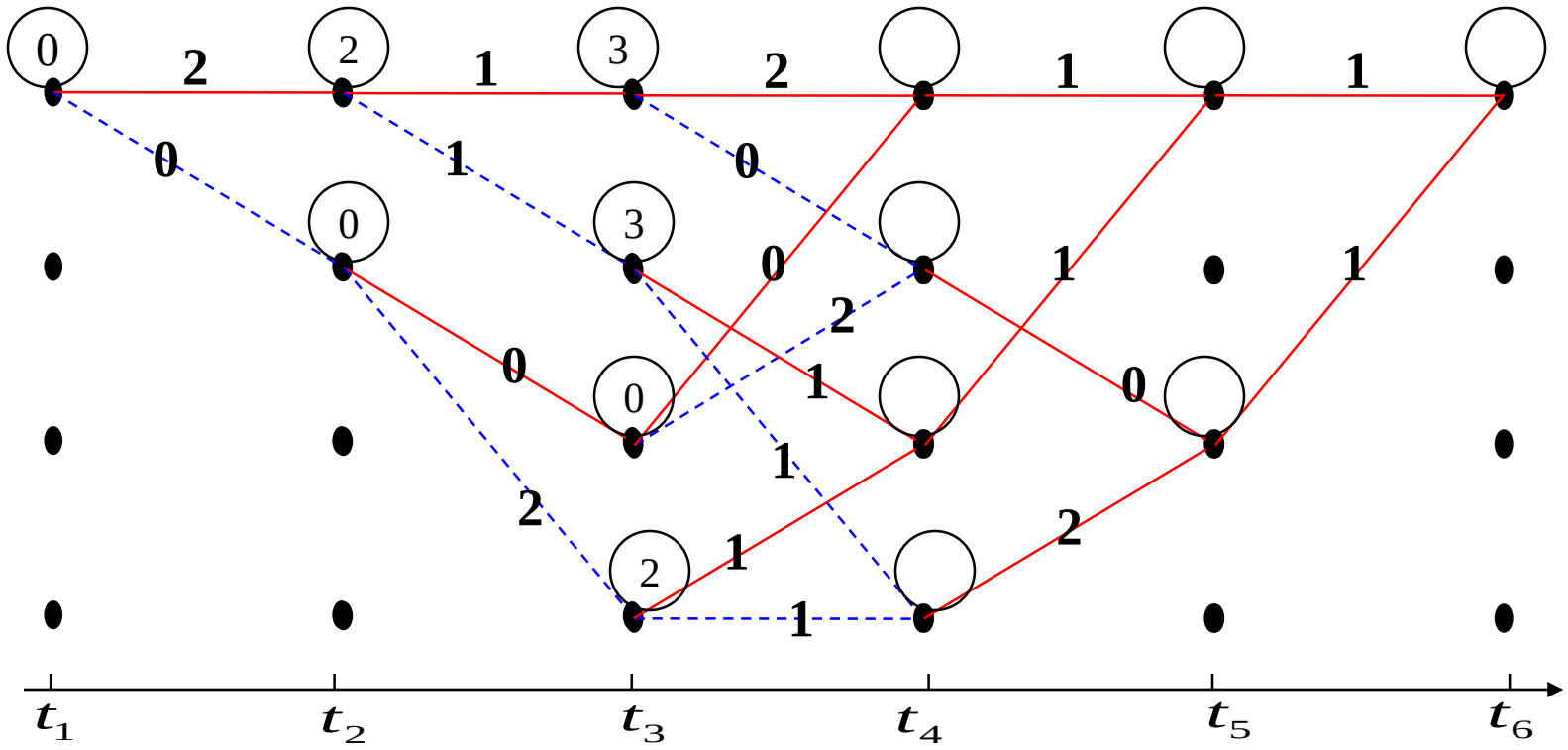
Example of Hard decision Viterbi decoding-cont'd

- i=2



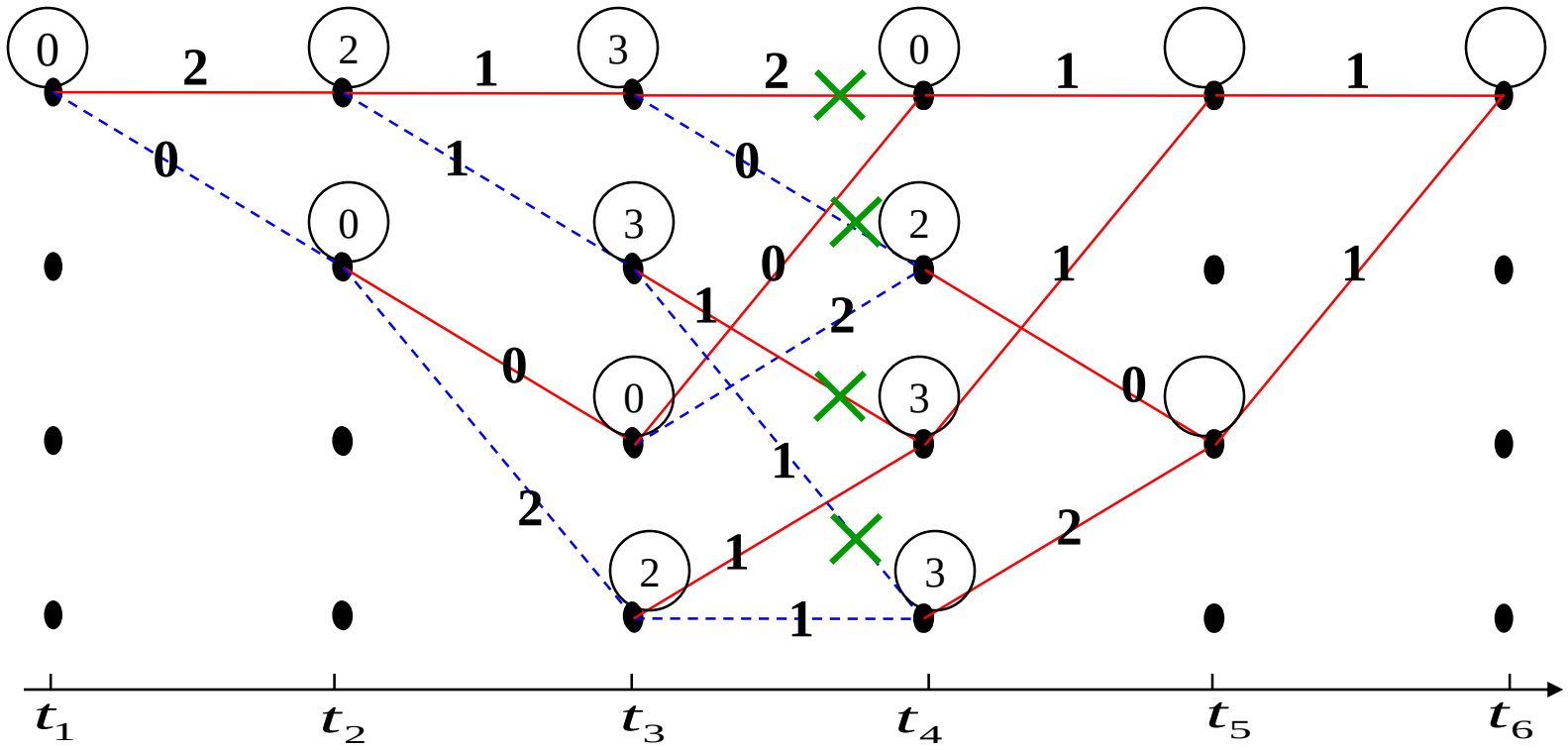
Example of Hard decision Viterbi decoding-cont'd

- i=3



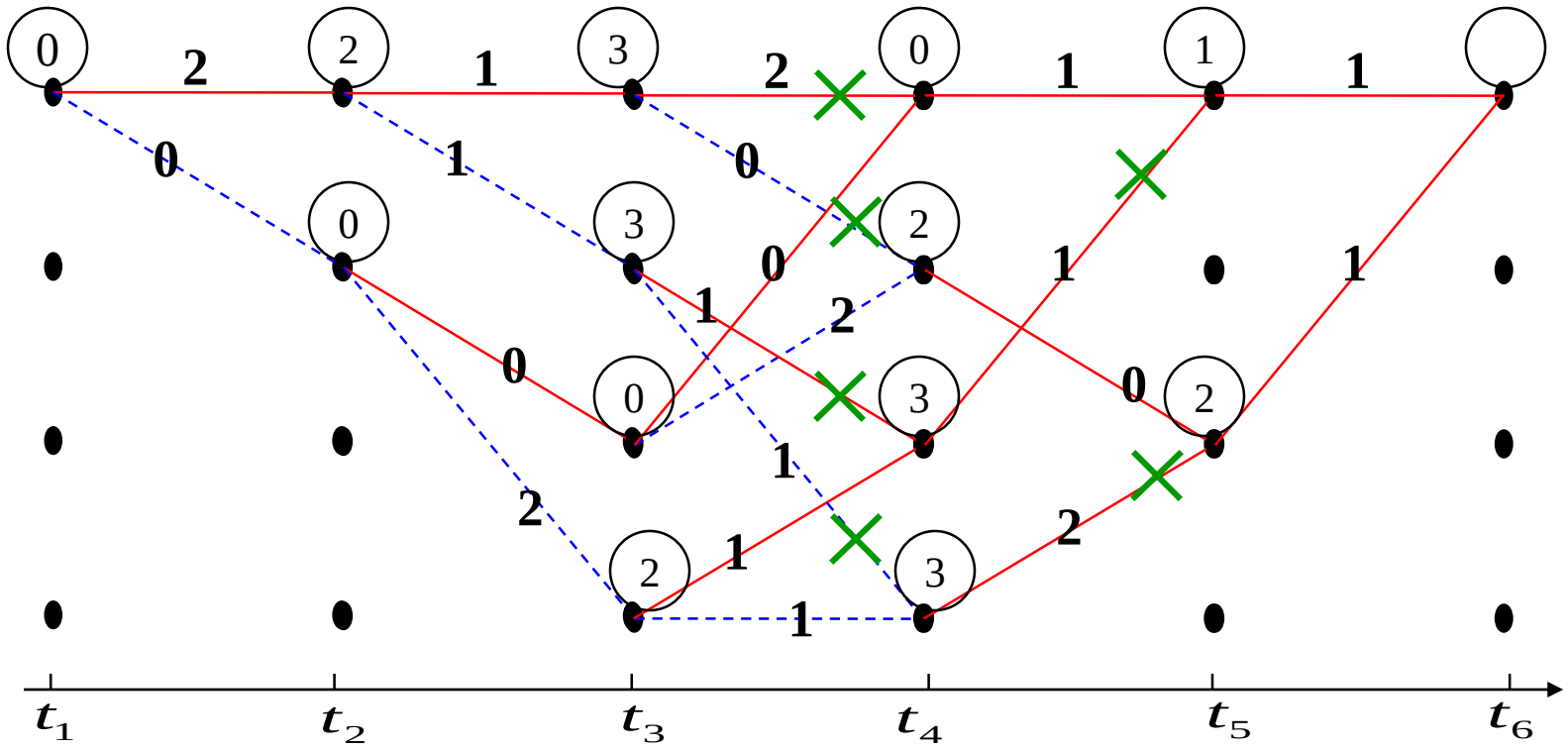
Example of Hard decision Viterbi decoding-cont'd

- i=4



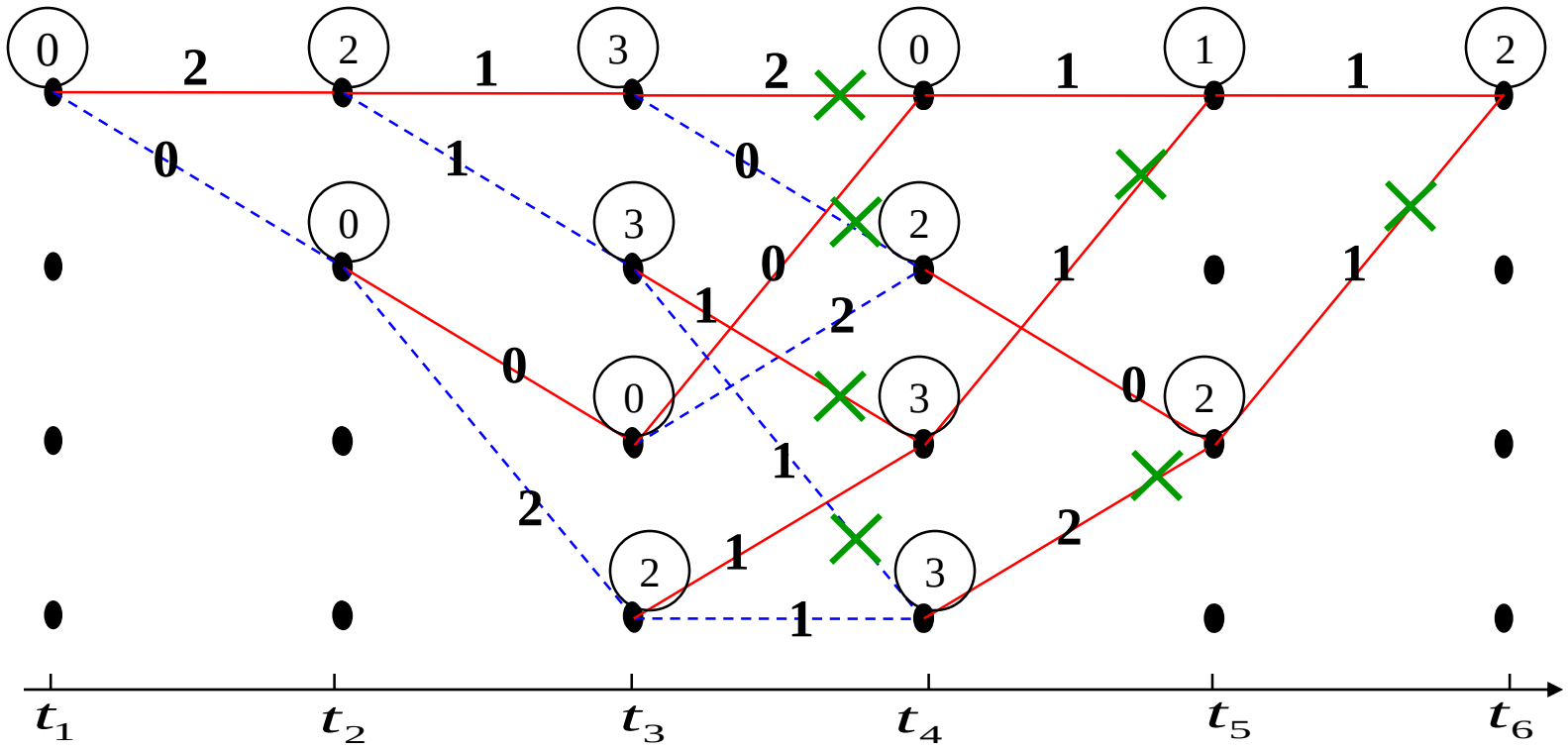
Example of Hard decision Viterbi decoding-cont'd

- i=5



Example of Hard decision Viterbi decoding-cont'd

- i=6

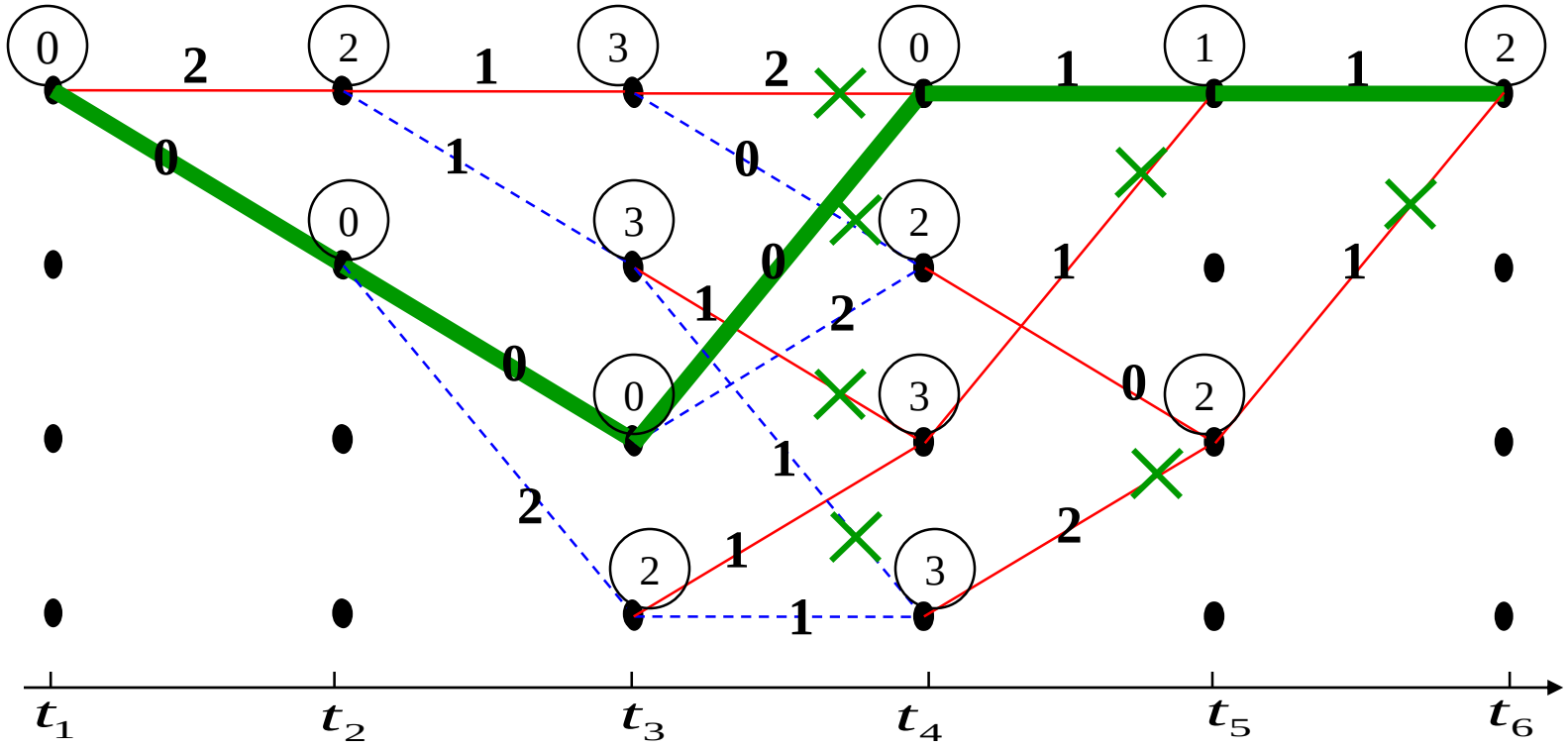


Example of Hard decision Viterbi decoding-cont'd

- Trace back and then:

$$\hat{\mathbf{m}} = (100)$$

$$\hat{\mathbf{U}} = (11 \quad 10 \quad 11 \quad 00 \quad 00)$$



Example of soft-decision Viterbi decoding

$$\mathbf{Z} = (1, \frac{2}{3}, \frac{2}{3}, \frac{-2}{3}, \frac{-2}{3}, 1, \frac{2}{3}, -1, \frac{-2}{3}, 1)$$

$$\mathbf{m} = (101)$$
$$\mathbf{U} = (11\ \ 10\ \ 00\ \ 10\ \ 11)$$

$$\hat{\mathbf{m}} = (101)$$
$$\hat{\mathbf{U}} = (11\ \ 10\ \ 00\ \ 10\ \ 11)$$