CT-216
Prof. Yash Vasavada

Convolution Coding

Presentated by Group-2 (Sub Group-3)

Winter 2024

Honor code

We declare that,
→ The work that we are presenting is our own work.
→ We have not copied the work (the code, the results, etc.) that someone else has done.
→ Concepts, understanding and insights we will be describing are our own.
→ We make this pledge truthfully. We know that violation of this solemn pledge can carry grave consequences.

Signed by: All members

Table of Contents

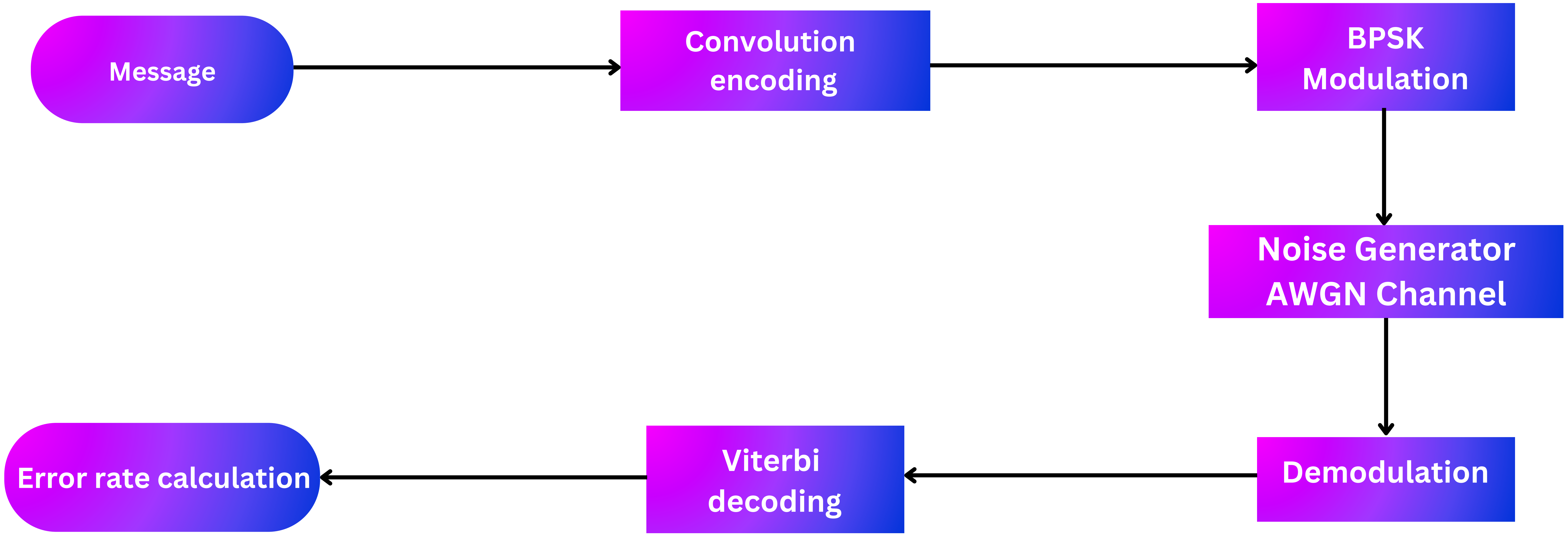
Introduction

- Convolutional codes are one of the type of error-correcting codes. They convolve input data with predefined generator sequence to generate a coded output sequence. This sequence is transmitted through a channel and decoded at the receiver in the presence of errors.

- Convolutional codes offer efficient error correction, making them crucial for applications like wireless communication.

- Convolutional codes were introduced by Peter Elias in 1955. Andrew Viterbi determined that convolutional codes could be maximum-likelihood decoded with reasonable complexity using time invariant trellis-based decoders — the Viterbi algorithm.

Block Diagram



Structure of a Convolution Code

Convolution codes are characterized as **(n, k, m).**
Where,

- n = No. of output bits produced for each k-bit sequence
- k = No. of input bits.
- m = No. of memory registers.

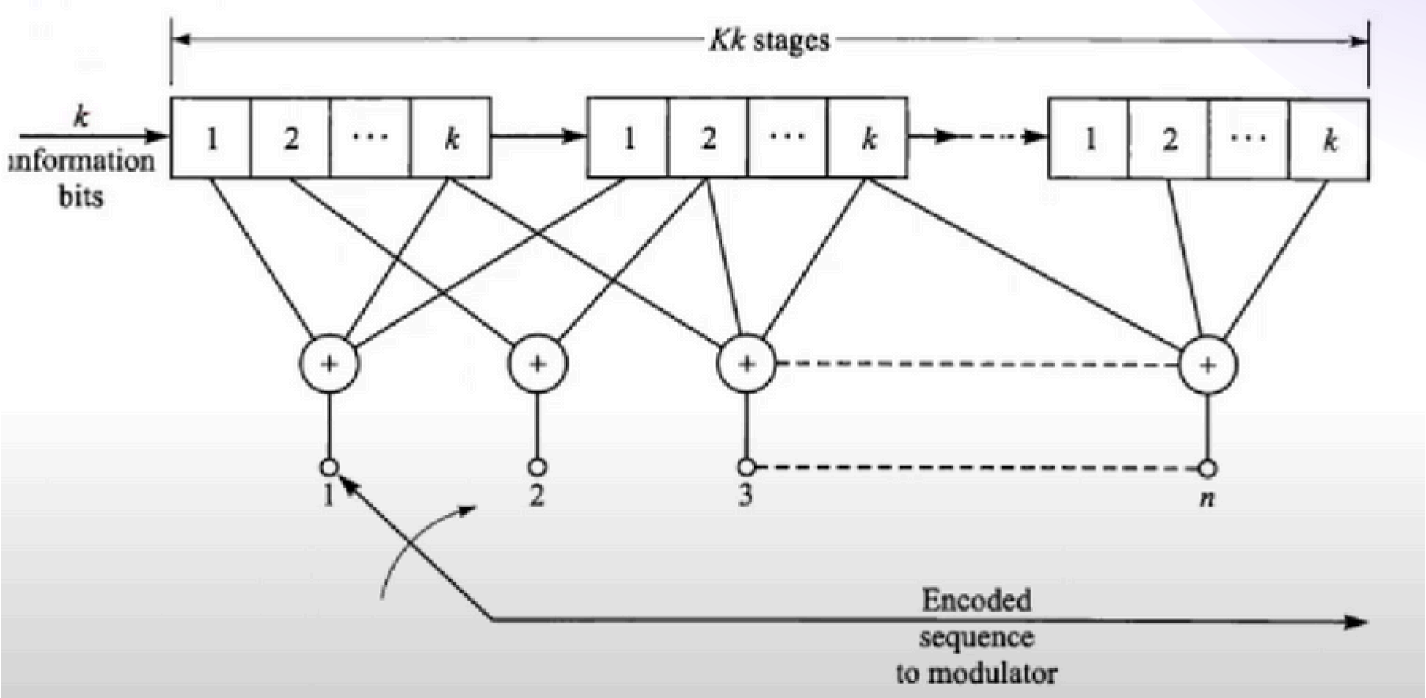


Fig 1: General structure

- Code rate R = k/n.
- Constraint length Kc = k(m - 1).
- So the message length will be n(k+Kc-1)

Encoding

Types of Convolution encoding representation:

- State diagram
- Tree diagram
- Trellis Diagram

Example : Rate = 1/2, Kc = 3

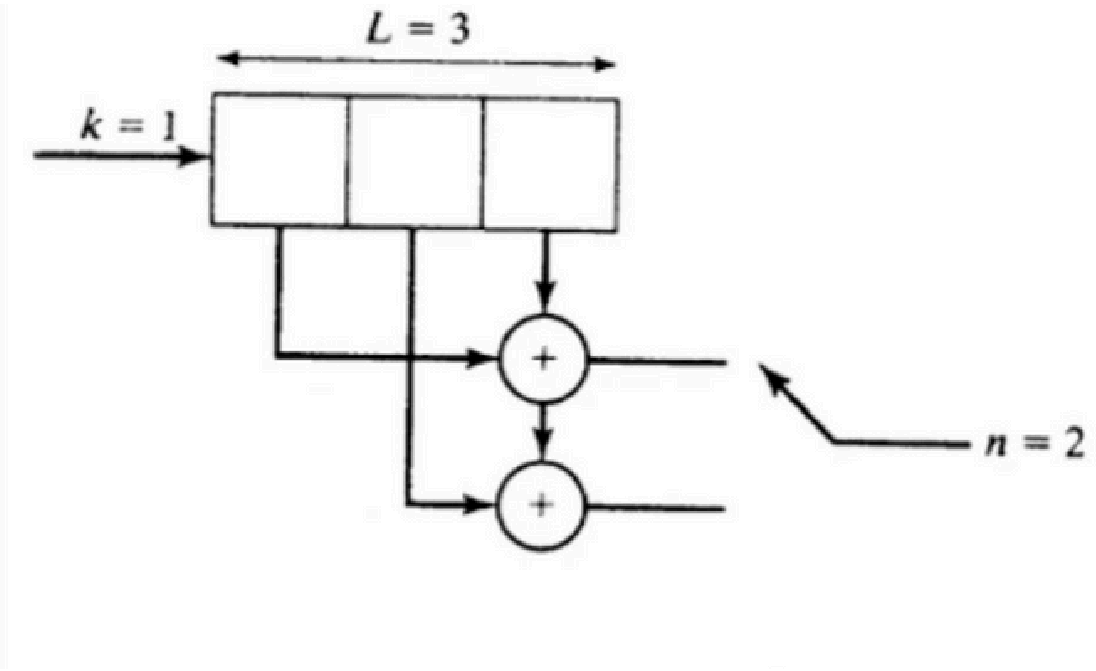


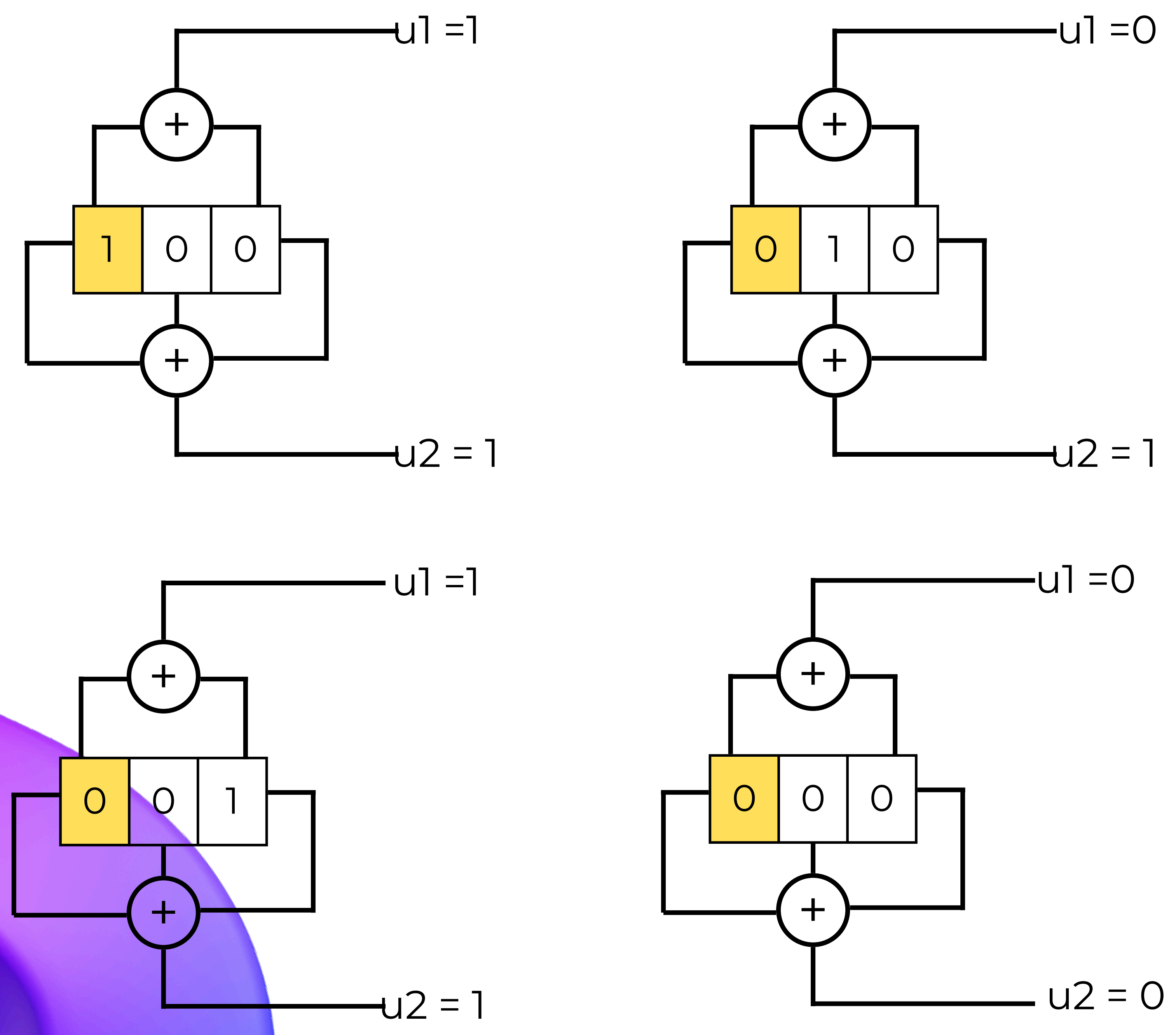
Fig 2: Rate 1/2, Kc = 3

- Rate R = 1/2;
- k = 1, n = 2, Kc = 3
- Let g(.) be a generator matrics.
  g(0) = [1 0 1]
  g(1) = [1 1 1]

- Let, m be the message sequence.
- m = [1 0 1 0 0 0] last three zero's are for termination.
- Output bits are u1 and u2.

Encoding Example

As input is coming from the left side this whole block of size Kc will shift in the right direction.



For message sequence **m[1 0 1 0 0 0],** the encoded sequence comes out to be **u[11 01 00 01 11 00]**

Various Representations

For rate = 1/2, Kc = 3:

State Table

I/P bit	Current state	Next state	Register content	O/P bit1	O/P bit2
0	00	00	000	0	0
1	00	10	100	1	1
0	01	00	001	1	1
1	01	10	101	0	0
0	10	01	010	0	1
1	10	11	110	1	0
0	11	01	011	1	0
1	11	11	111	0	1

Fig 3: State transition table

State Diagram

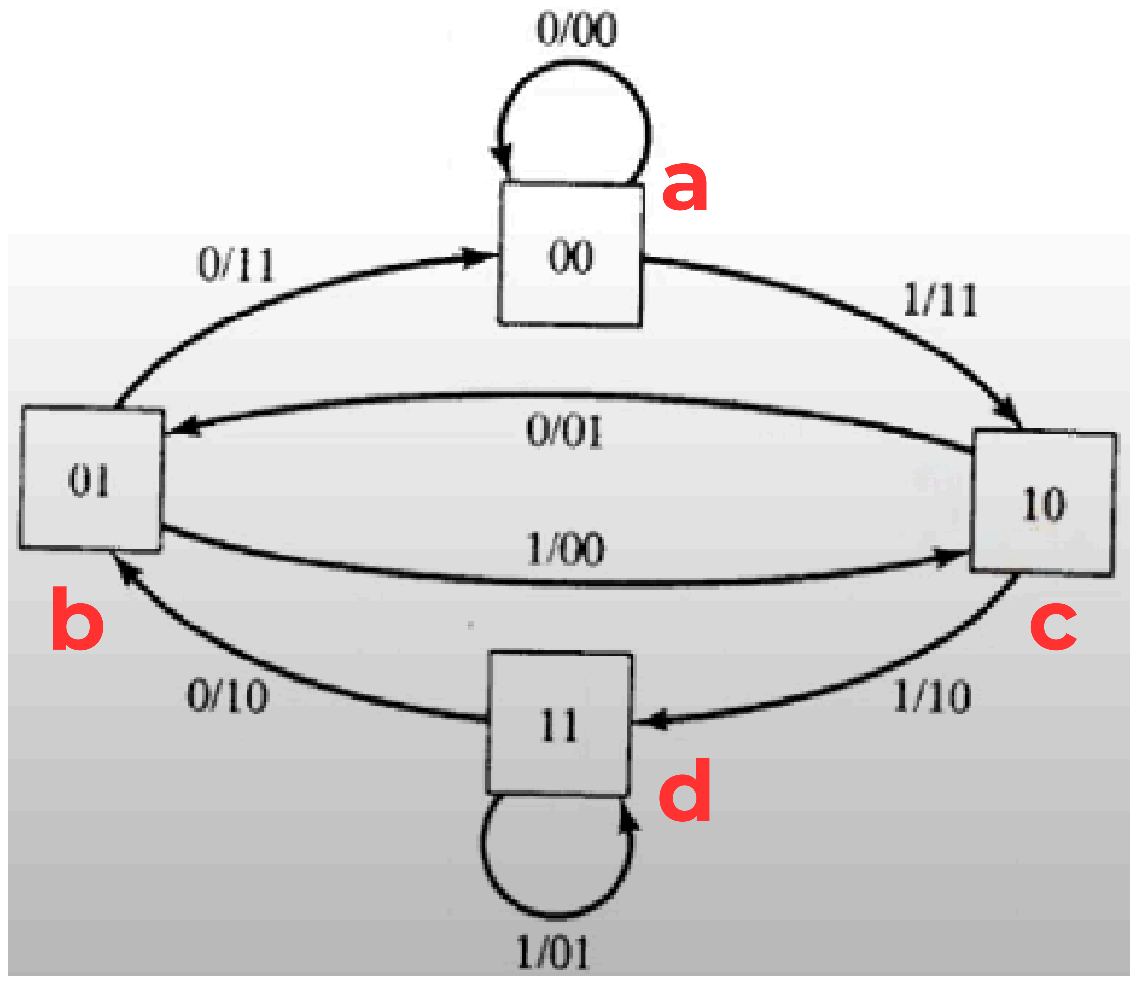


Fig 4: State Diagram

Various Representations

For rate = 1/2, Kc = 3:

Tree Diagram

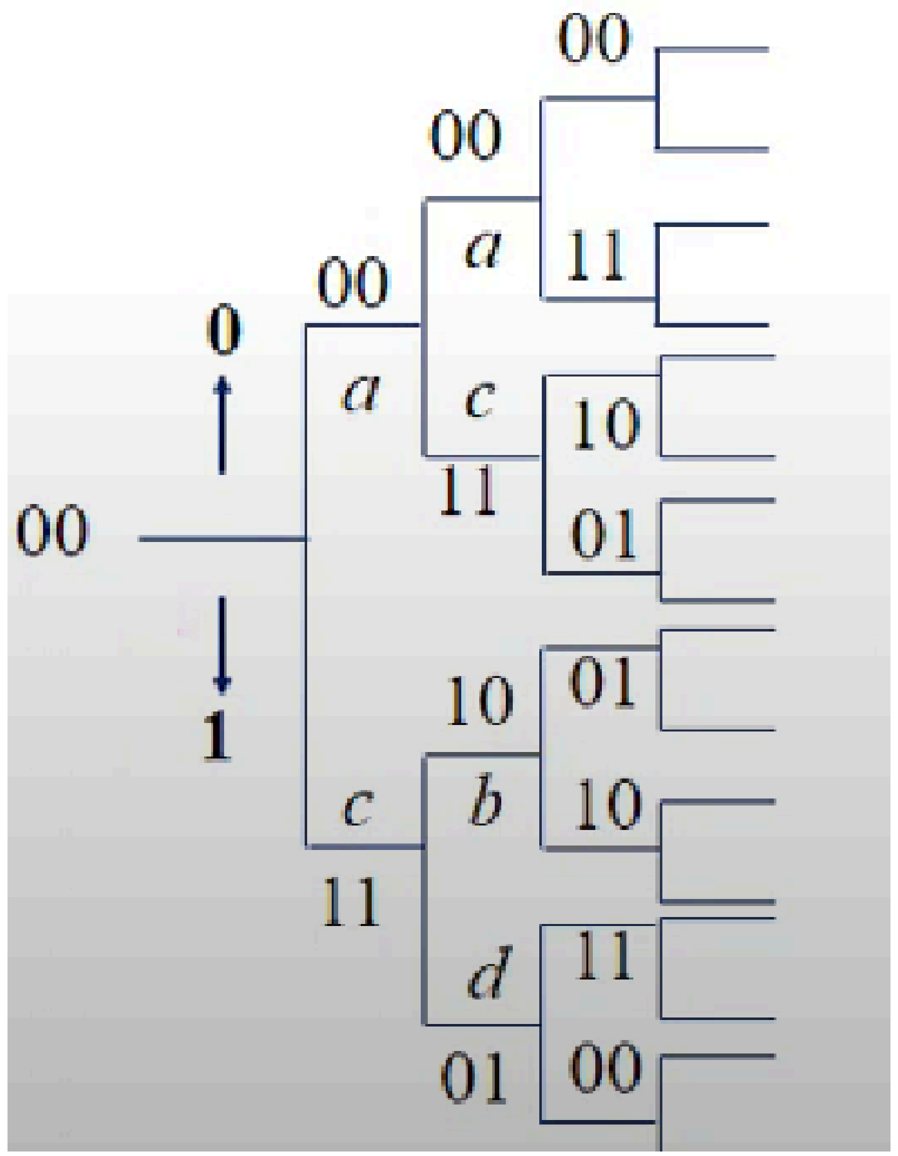


Fig 5: Tree Diagram

Trellis

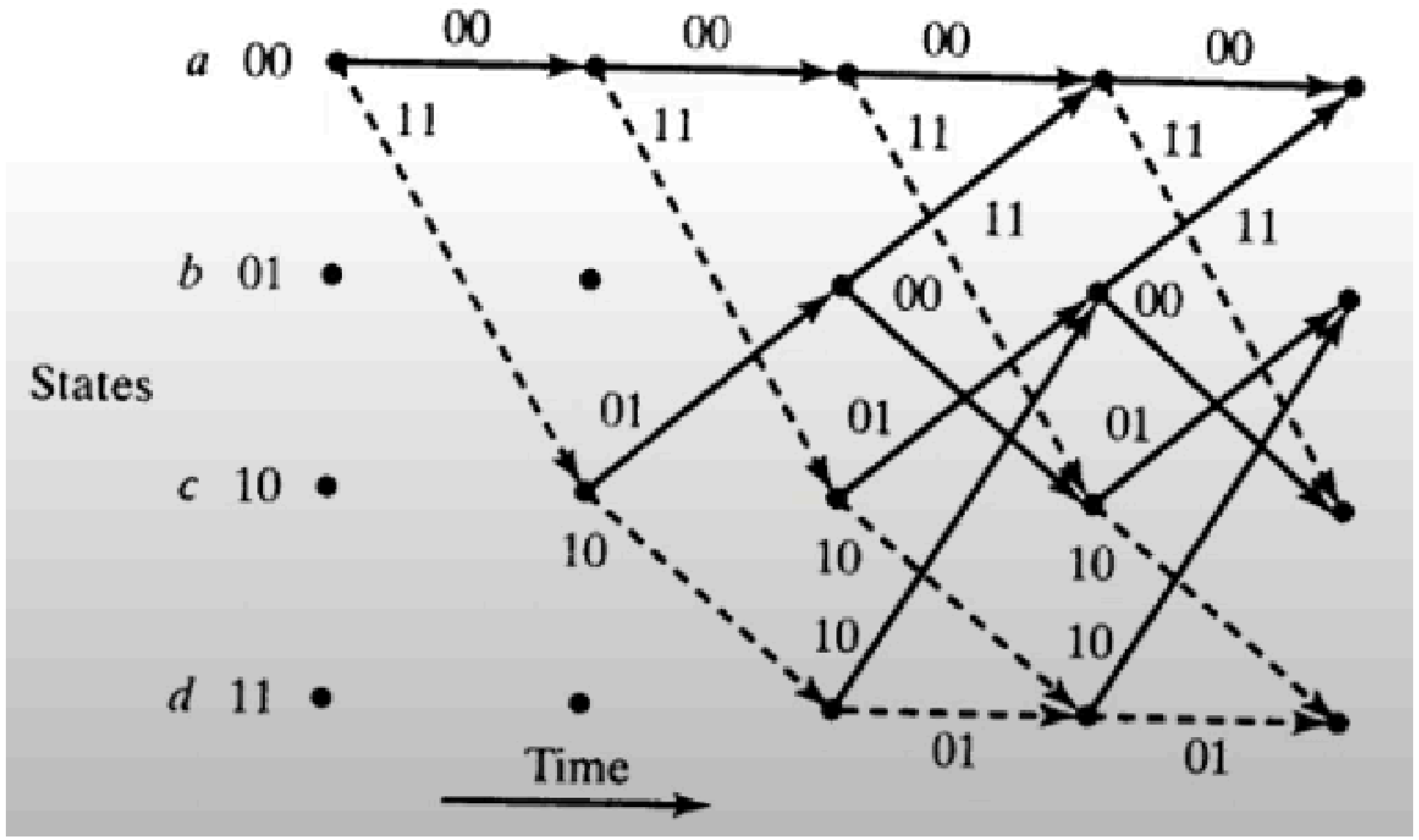


Fig 6: Trellis Representation

Transfer Function

- Transfer function is used to give the distance properties of a convolutional code.
- Furthermore, transfer function is an algebraic representation of all paths that start and end at all-zero state.

$$T(D,N,J) = \sum_{d=d_{free}}^{\infty} a_d D^d N^{f(d)} J^{g(d)}$$

- **D**: Exponent 'd' denotes the hamming distance of the sequence of output bits corresponding to each branch from the sequence of output bits corresponding to the all-zero branch.
  <u>OR</u>
  The number of ones in output codeword.
- **N**: This factor is introduced to traverse each branch.
  Exponent 'f(d)' denotes the number of ones in input block k-bits at a time.
- **J**: This factor serves as a counting variable to indicate the number of branches in any given path start node with all-zero state to end node with all-zero state.
  Exponent 'g(d)' denotes the number of branches spanned by the path.
- **dfree (OR dHmin)**: The minimum distance of the code also known as minimum free distance.

How to compute Transfer Function

1. Split all-zero state into two: one at the start another at the end
2. All other states are transition states
3. Transfer function is defined as Output at all zero end state/ Input at all zero start state

Example: Rate=1/2, Kc = 3:

From Fig 4, we can get the following diagram to compute the transfer function.

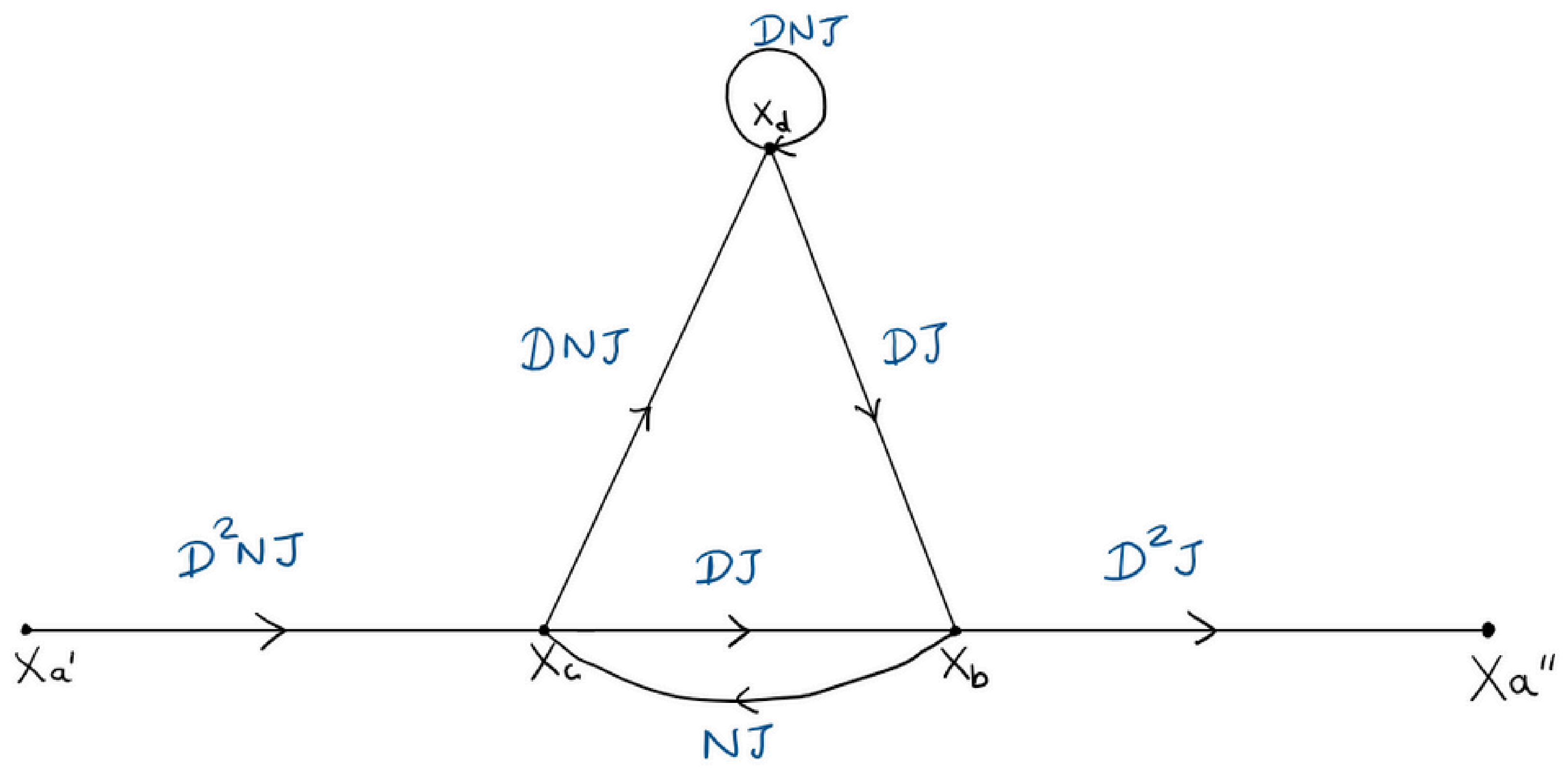


Fig 7: Flow Diagram

How to compute Transfer Function

$$X_C = D^2NJXa' + NJX_b \quad ....1$$

$$X_b = DJX_d + DJX_C \quad ....2$$

$$X_d = DNJX_C + DNJX_d \quad ....3$$

$$X_{a''} = D^2JX_b \quad ....4$$

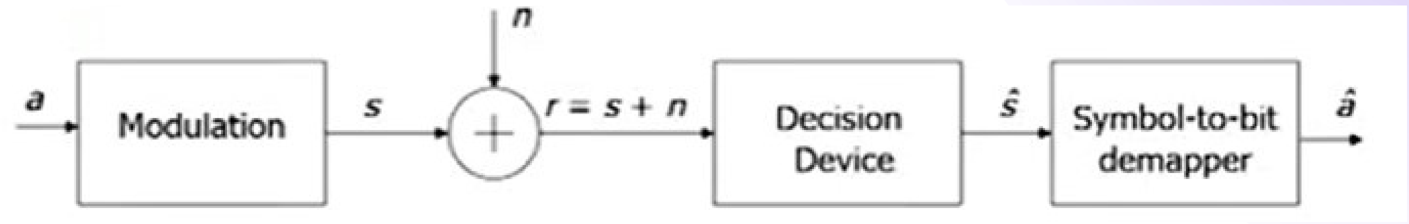
Transfer function = $\dfrac{X_{a''}}{X_{a'}} = \dfrac{D^2JX_b}{\frac{1}{D^2NJ}(X_C - NJX_b)} = \dfrac{D^4NJ^2X_b}{X_b\left(\frac{X_C}{X_b} - NJ\right)} = \dfrac{D^4NJ^2}{\left(\frac{X_C}{X_b} - NJ\right)} = \dfrac{D^4NJ^2}{\left(\frac{1-DNJ}{DJ} - NJ\right)} = \dfrac{D^5NJ^3}{1 - DNJ - DNJ^2}$

$$T(D,N,J) = D^5NJ^3 + D^6N^2J^4 + D^6N^2J^5 + D^7N^3J^5 + \cdots$$

The first term in the expansion indicates that the distance d = 5 path is of length 3 and of the three information bits one is a 1. The second and third term indicates that the distance d = 6 path is of length 4 and 5 and of the four information bits two bits are 1s and of the five information bits two bits are 1s.

BPSK and AWGN Modulation

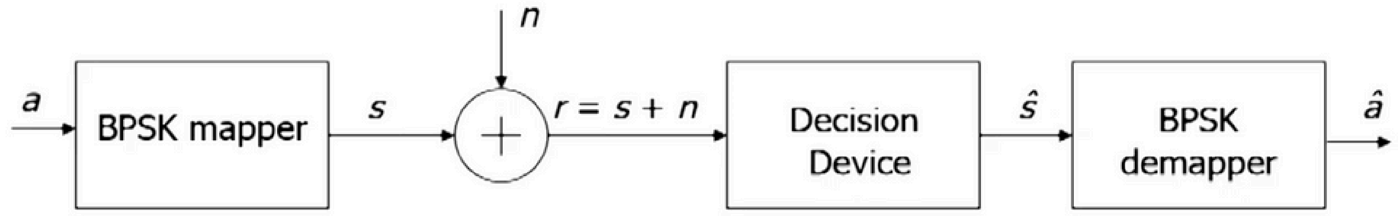
- AWGN(Adiitive White Gaussian Noise) is a commonly encountered type of noise in communication systems
- It directly gets added to original signal, and exhibits a Gaussian (normal) distribution with a constant power spectral density across all frequencies.



a = encoded data stream
s = symbol representaion s(t)
n = AWGN
s^ = symbol after noise addition a^ = binary data stream after demodulation

- BPSK (Binary Phase Shift Keying), is a digital modulation method used in communication systems to transmit binary data over a radio frequency carrier wave.

- In BPSK the phase of the carrier wave is shifted to represent '1' and '0'. For bit '1', the phase of the carrier wave is shifted by 180 degrees to represent 1 and for representing 0, the phase remains unchanged.



a:	0	0	1	0	1	1	0	1	1	0	1
s:	+1	+1	−1	+1	−1	−1	+1	−1	−1	+1	−1
r:	0.8	0.2	-0.8	1.9	-0.6	0.2	1.3	0.1	-1.2	0.3	-1.1
$\hat{s}$:	+1	+1	−1	+1	−1	+1	+1	+1	−1	+1	−1
$\hat{a}$:	0	0	1	0	1	0	0	0	1	0	1

Demodulation

- Demodulation is the process of reverting the process of modulation to extract the information from the modulated signal.

- The received information is then digitalized to their corresponding symbols (i.e. *s* are converted to binary bits b)

Decoding

- Andrew Viterbi proposed a Maximum Likelihood Sequence Based Estimation Procedure (MLSE) to optimally decode convolution codes, also known as **Viterbi Algorithm**.
- The decoder output is based on the largest value of log-likelihood function.

- There are two types of decoding techniques:
  Hard decision decoding (Hamming Distance)
  Soft decision decoding (Euclidian Distance)

- They require the use of two types of metrics - branch metric and path metric.

- **Branch Metric** is the measure of the Hamming distance OR Euclidian distance between the received code word and the expected parity bits.

- **Path Metric** is the sum of the branch metric of all its branches coming in its path.

Hard-Decision Decoding

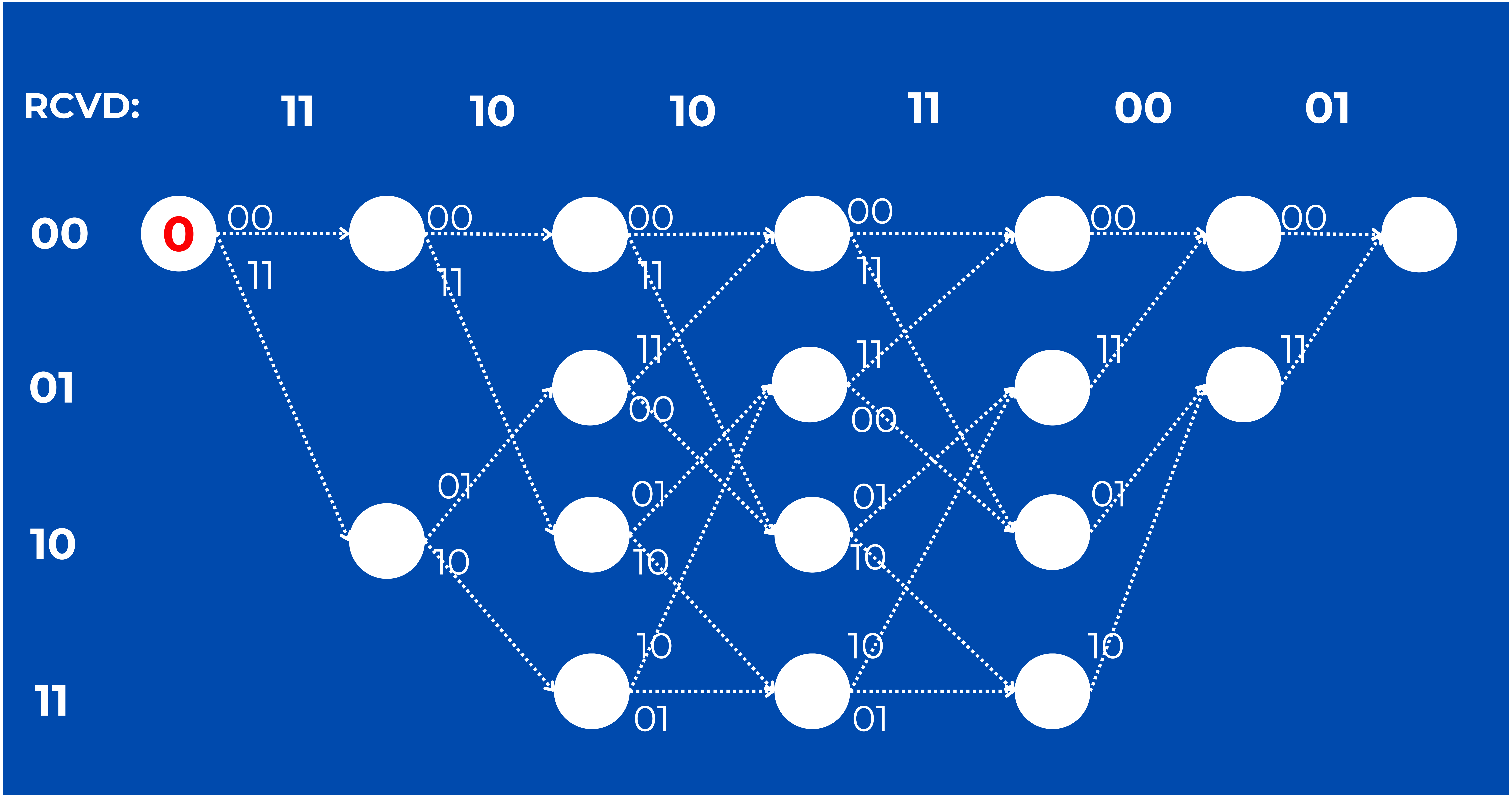
- In Hard Viterbi decoding, the received signal is compared directly with the possible codewords to find the closest match.
- Each received symbol is compared to all possible transmitted symbols to determine the likelihood of each symbol.
- The decoder selects the sequence that minimizes the distance between the received signal and the all possible sequences.
- This method is suitable for channels with low noise levels or when the received symbols are binary without any additional information about their reliability.

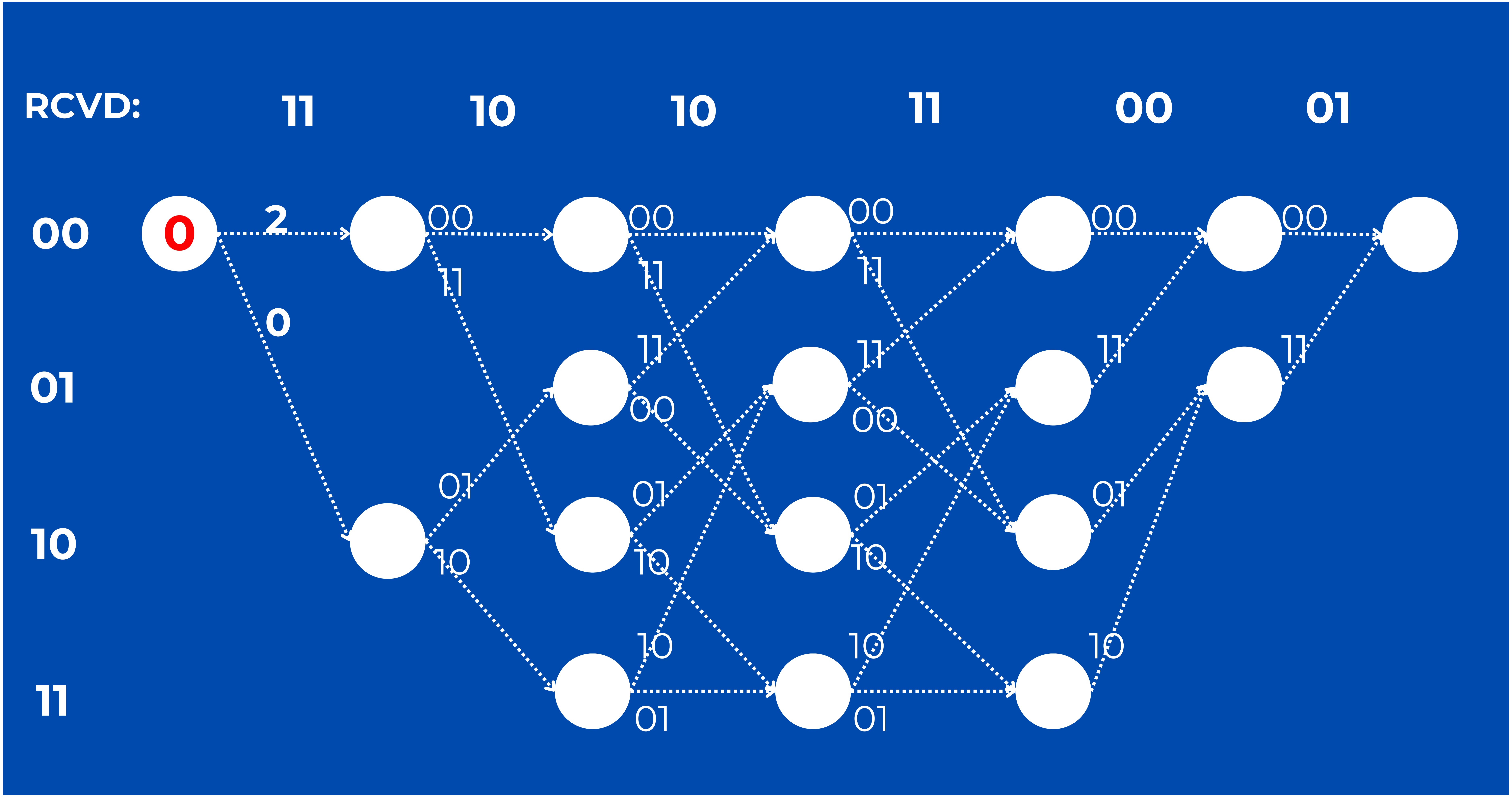
Steps to perform Hard-Decision Decoding

- Define states for trellis based on the constraint length(K). Total States : 2^(k-1)
- Make trellis diagram and set the all zero state of the trellis to zero.

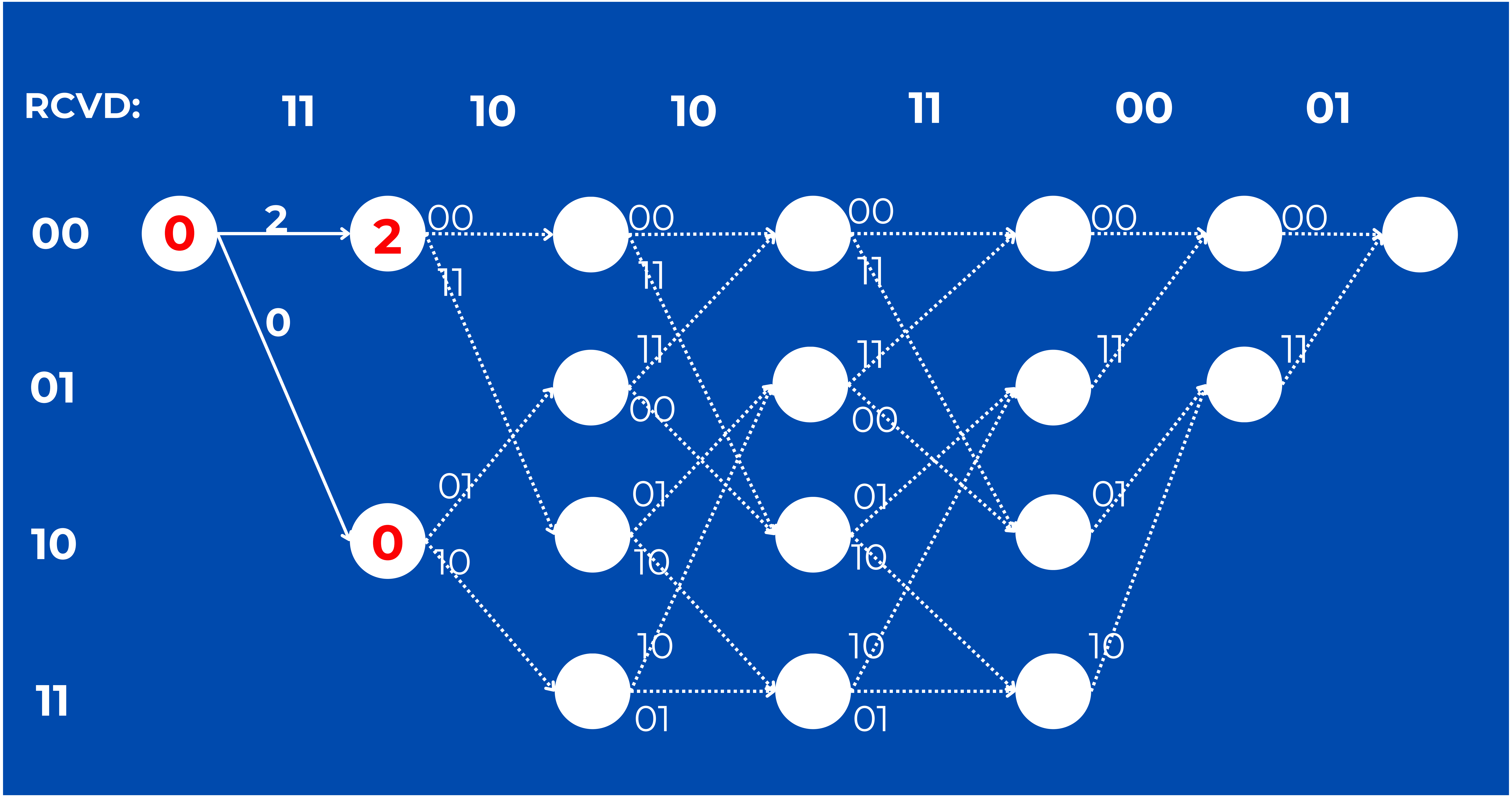
- Now, Compute the branch metric for each path that enters each state of the trellis and store the metric for each state and identify the survivor.
- Repeat the same process for all 2^(K - 1) paths.

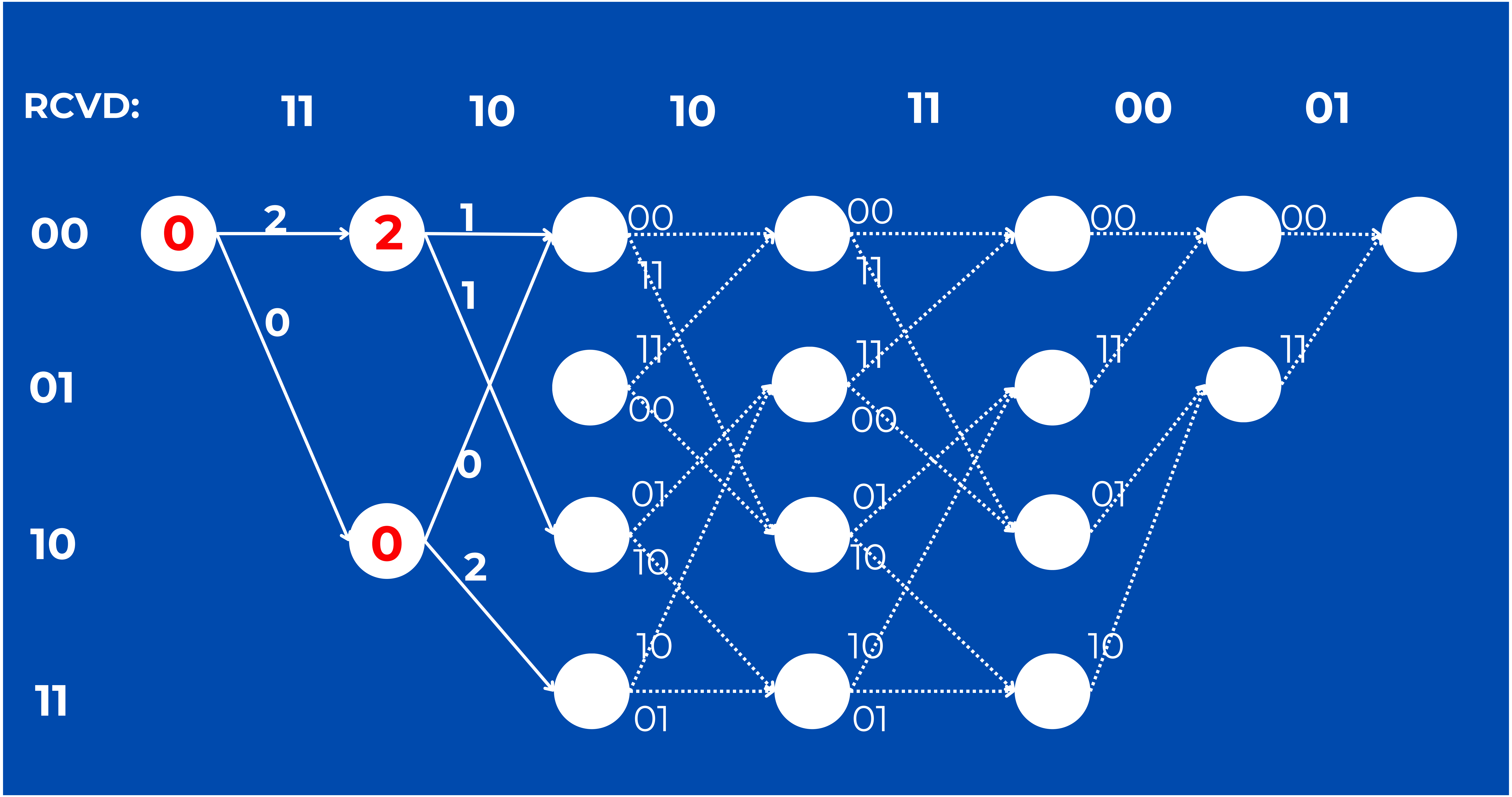
- Add the metrics entering the state to the metric of the survivor at previous state. compare the metrics of all 2^(K - 1) paths entering the state. Select the survivor with the largest metric, store it and discard all other trellis paths.
- Repeat this for < N + n’, where n is the length of the message sequence and n’ is the no. of tail zero’s

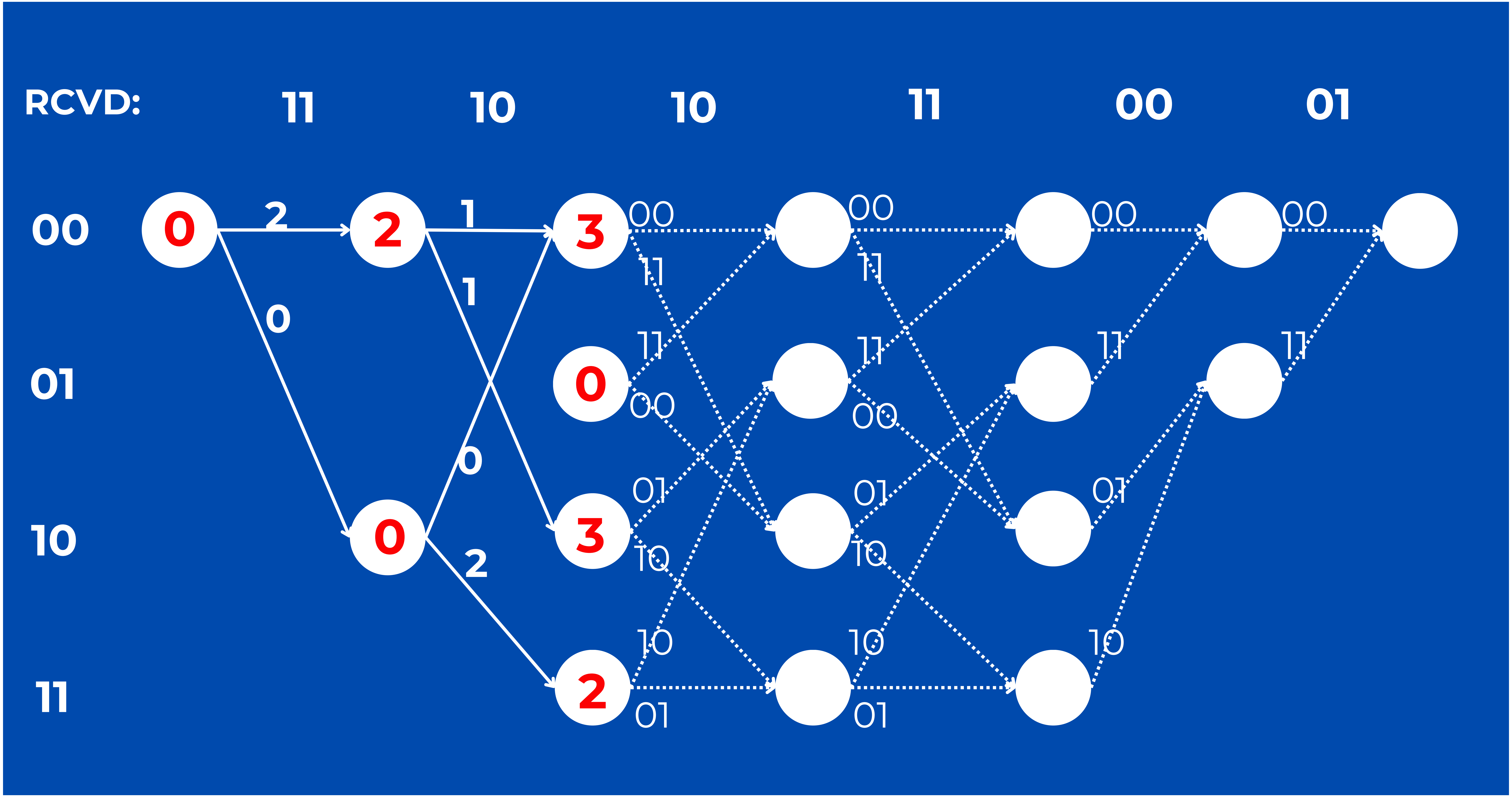

RCVD:
11
10
10
11
00
01
00
01
10
11
0

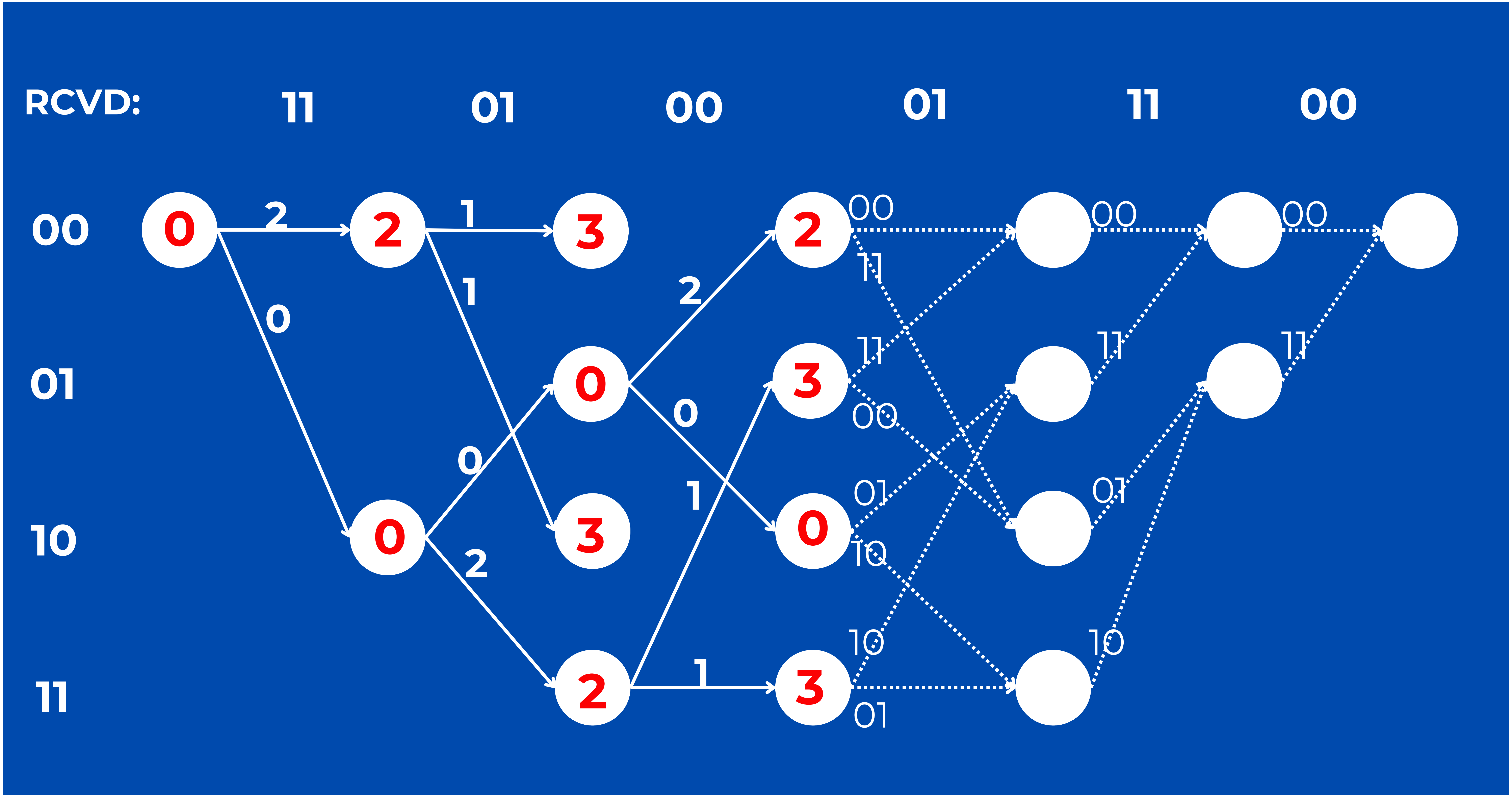


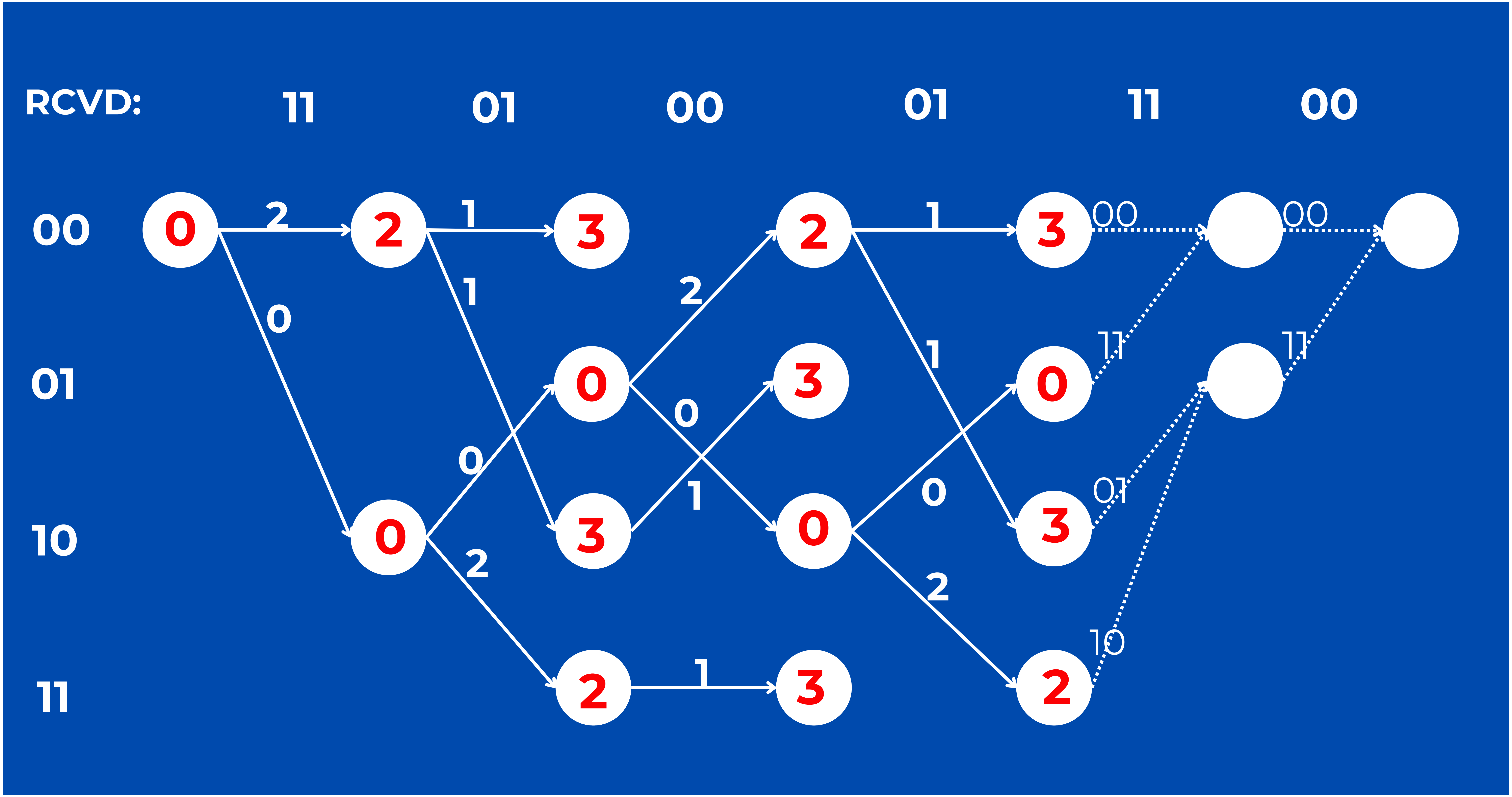
RCVD:
11
10
10
11
00
01
00
01
10
11
0
2
0
00
11
01
10
00
11
11
00
01
10
10
01
00
11
11
00
01
10
10
01
00
11
01
10
00
11

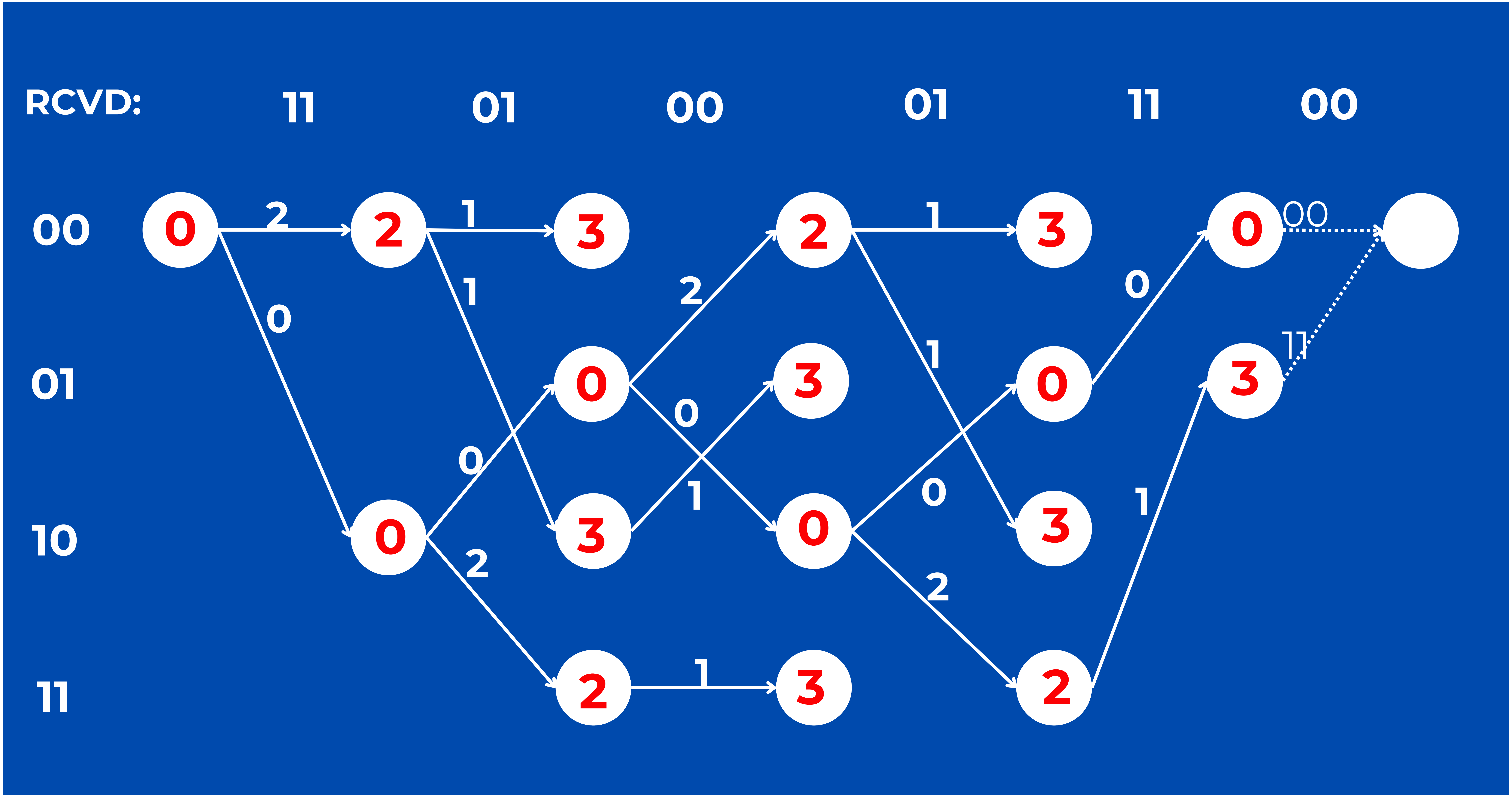

RCVD:
11
10
10
11
00
01
00
01
10
11
0
2
0
2
0
00
11
01
10
00
11
11
00
01
10
10
01
00
11
11
00
01
10
10
01
00
11
01
10
00
11


RCVD:
11
10
10
11
00
01
00
01
10
11
0
2
0
2
1
1
0
2
00
11
11
00
01
10
10
01
00
11
11
00
01
10
10
01
00
11
01
10
00
11

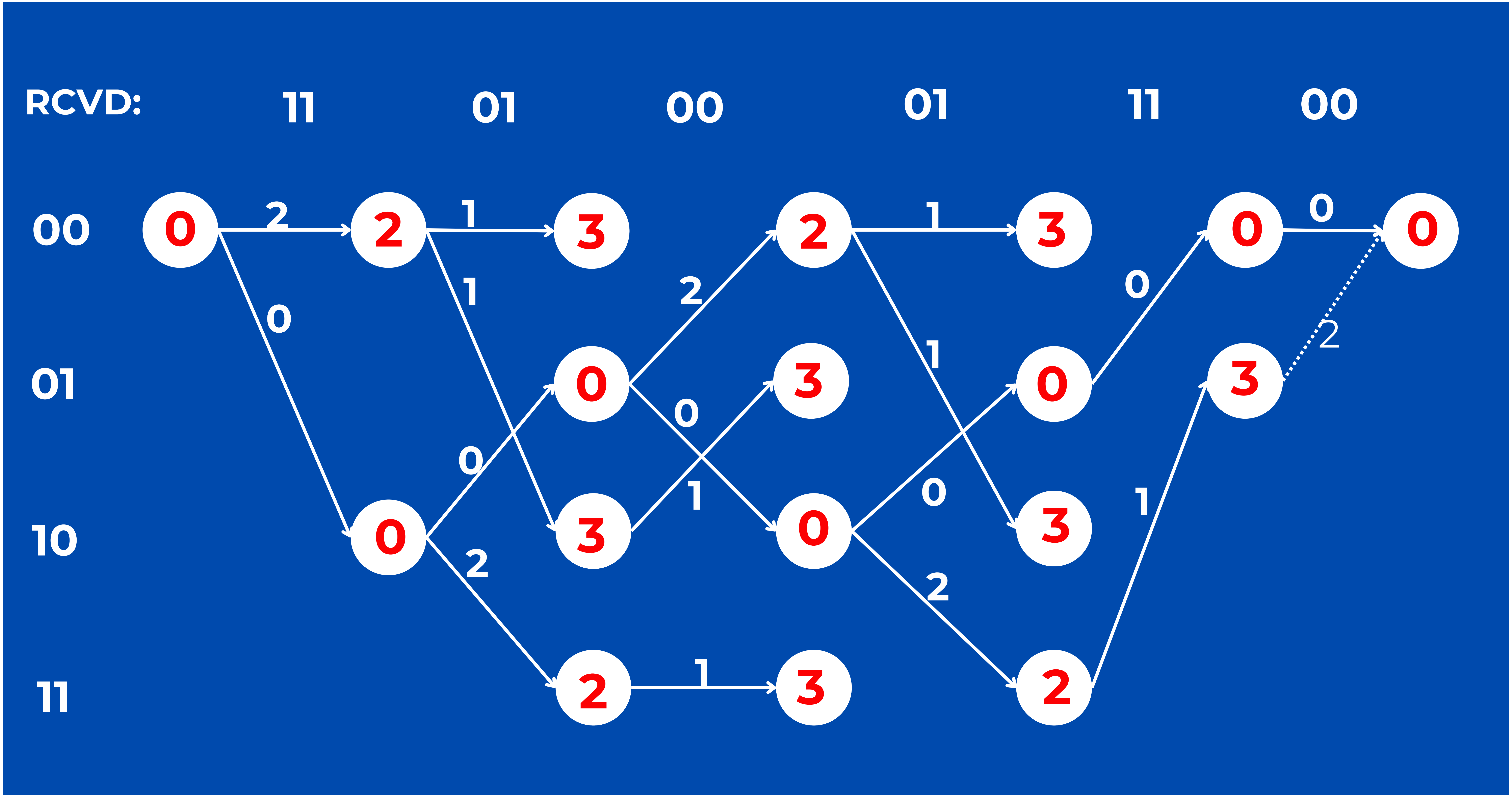

RCVD:
11
10
10
11
00
01
00
01
10
11
0
2
0
3
0
3
2
2
0
1
1
0
2
00
11
11
00
01
10
10
01

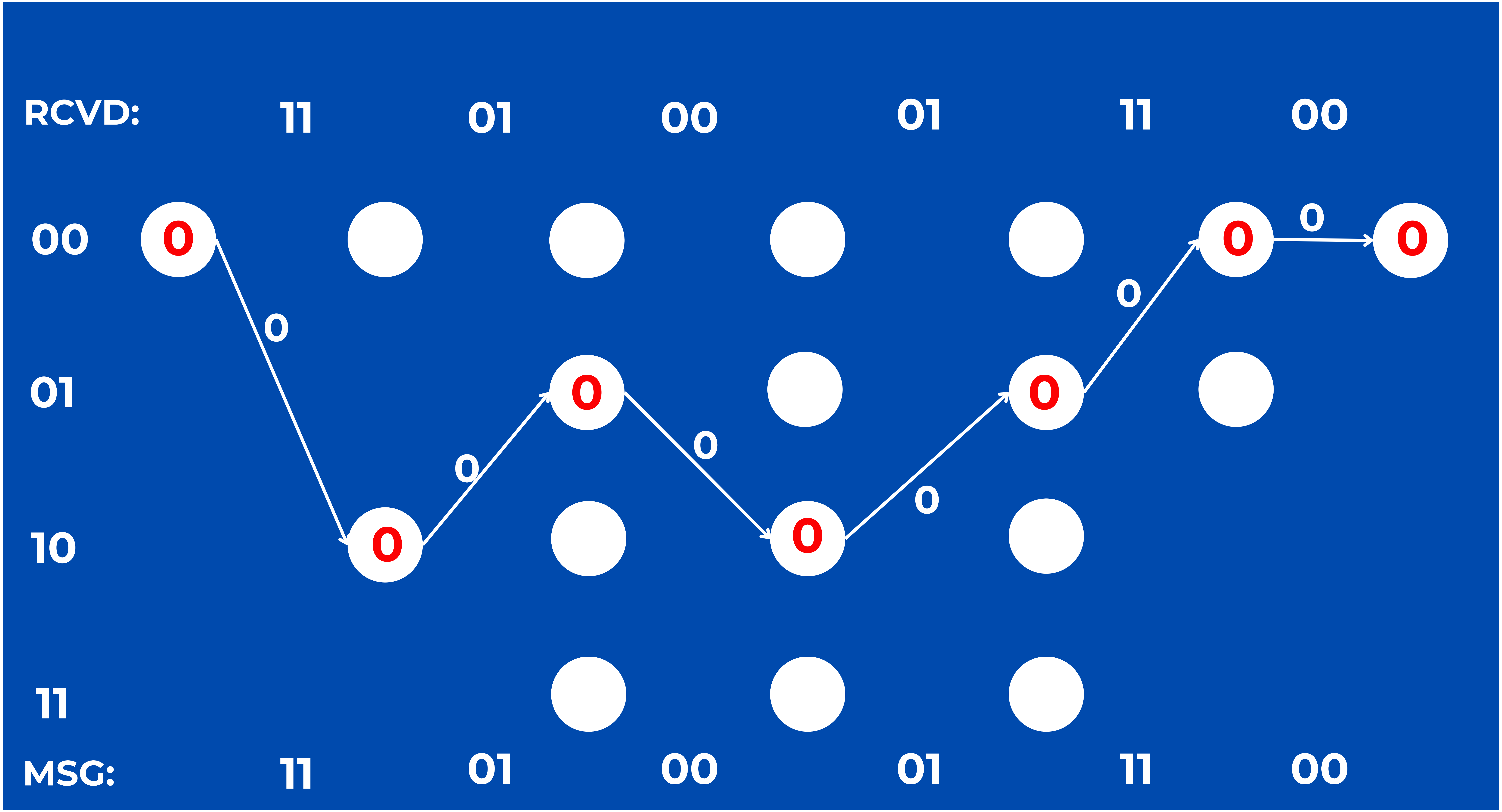

RCVD:
11
01
00
01
11
00
00
01
10
11
0
2
3
2
0
3
0
3
2
3
2
1
0
1
1
0
2
2
0
1
1
00
11
11
00
01
10
10
01
00
11
01
10
00
11


RCVD:
11
01
00
01
11
00
00
01
10
11
0
2
0
2
0
1
1
3
0
3
2
0
2
2
0
1
3
1
3
0
2
1
1
1
0
2
3
0
3
2
00
11
01
10
00
11



RCVD:
11
01
00
01
11
00
00
01
10
11
0
2
3
2
3
0
0
3
0
3
0
3
0
3
2
3
2
2
1
1
1
0
0
2
2
0
1
1
1
1
0
0
2
1
00
11


RCVD:
11
01
00
01
11
00
00
01
10
11
0
2
0
3
0
3
2
2
3
0
3
3
0
3
2
0
3
0



RCVD:
11
01
00
01
11
00
00
01
10
11
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
MSG:
11
01
00
01
11
00

Soft-Decision Decoding

- Soft Viterbi decoding is a method that factors in the reliability of received symbols, like signal-to-noise ratio or probabilities.

- Instead of making a strict decision based only on received symbols, it calculates the likelihood of each potential transmitted symbol based on signal quality.

- It uses metrics or probabilities associated with received symbols to find the most likely sequence, considering the channel's characteristics and noise levels.

- This approach is useful for channels with higher noise or when there's extra information about the symbols.

Steps to perform Soft-Decision Decoding

- Define states for trellis based on the constraint length(K). total states : 2^(k - 1).
- Make a Trellis diagram and set the all-zero state of the trellis to zero.

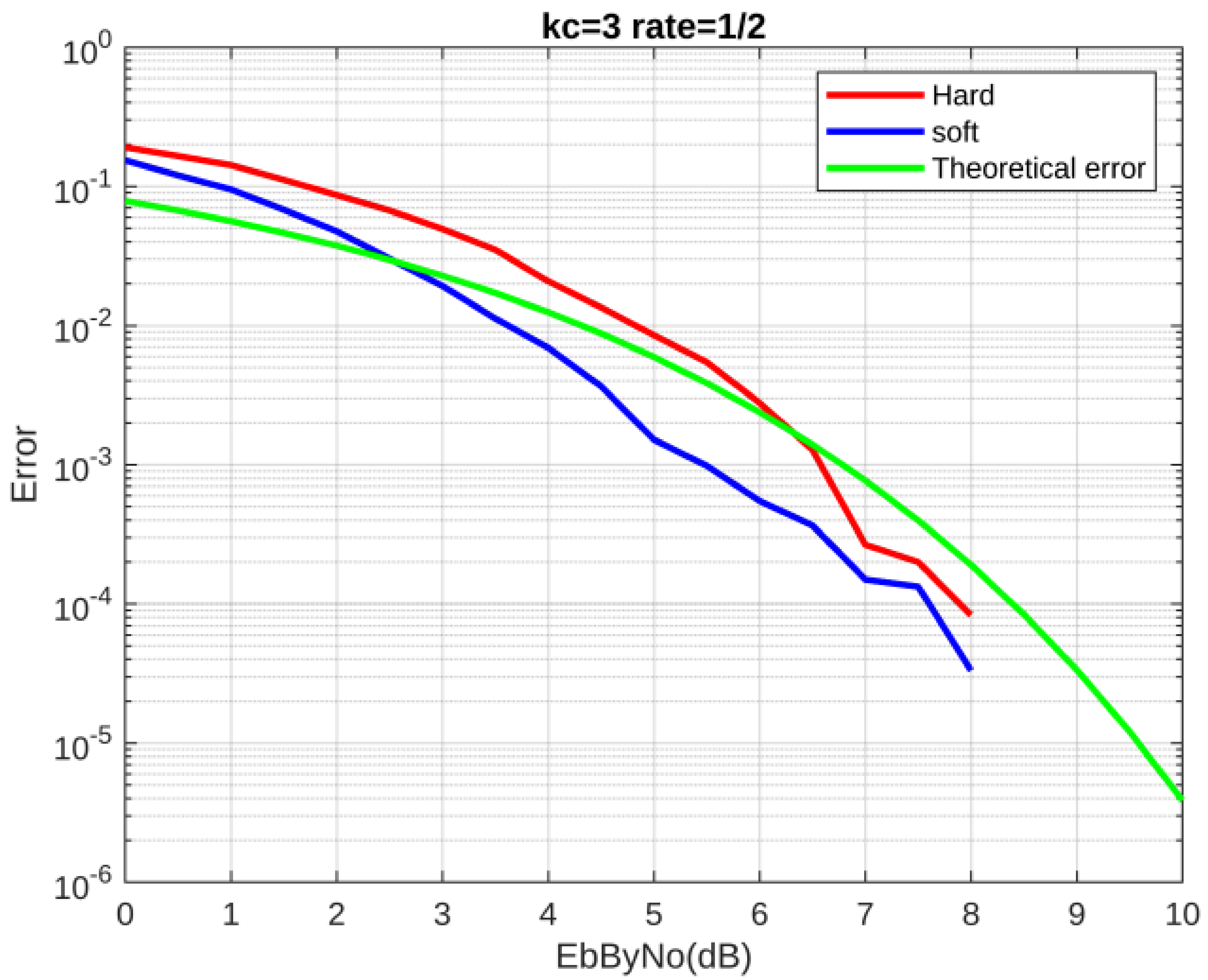
- Now, Compute the branch metric of a particular path metric by finding the Minimum Euclidean distance between the previous state output and all reachable nodes of that branch metric.
- The Euclidean distance is computed by taking the square root of the sum of the squares of the differences between corresponding received bits and that state's output bit .

- Revise the node metrices using the minimum branch metrics obtained.
- Once you reach the end of the trellis diagram, backtrack along the path that has the smallest Euclidean distance to obtain the decoded output.

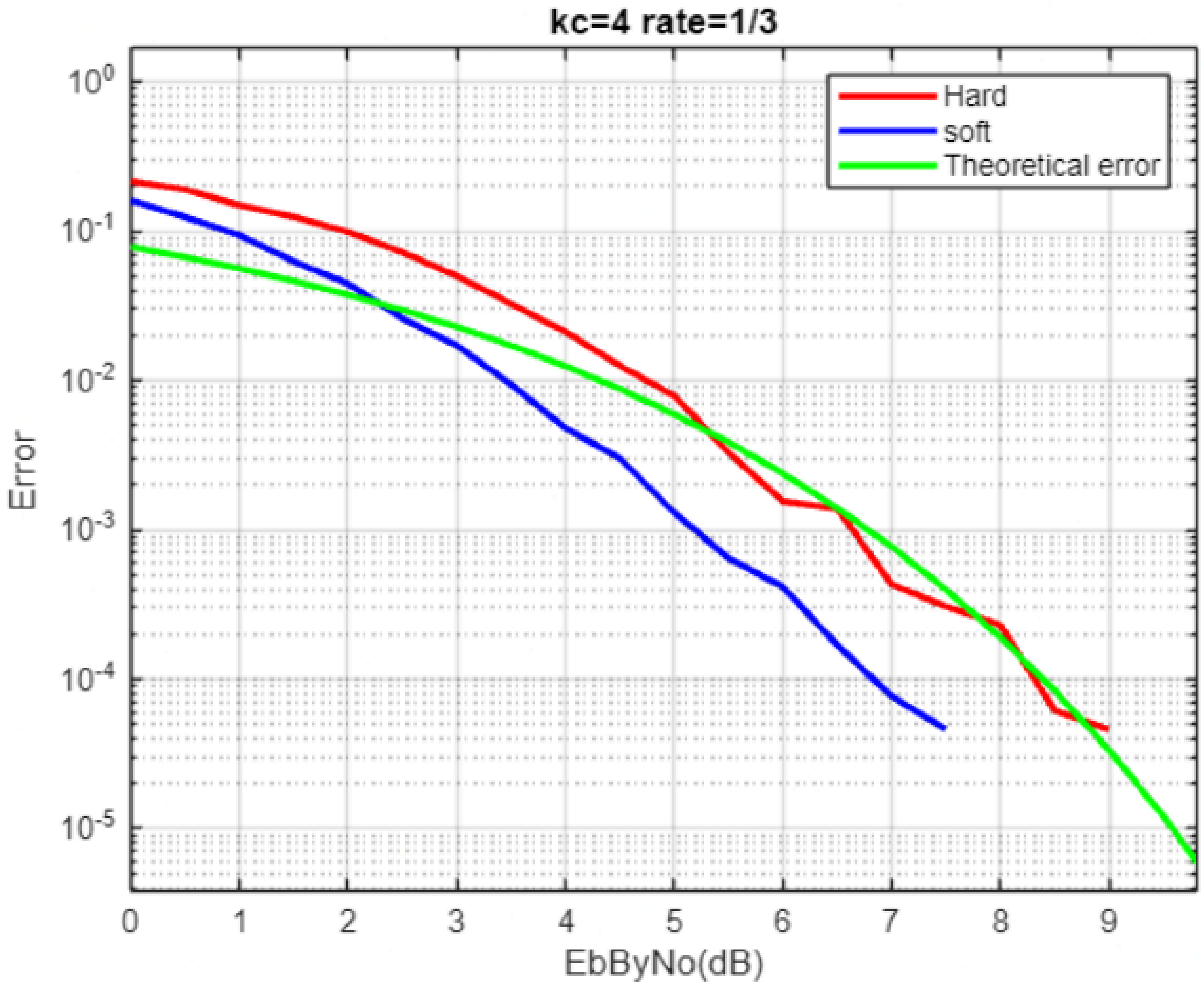
Analysis

Graphs

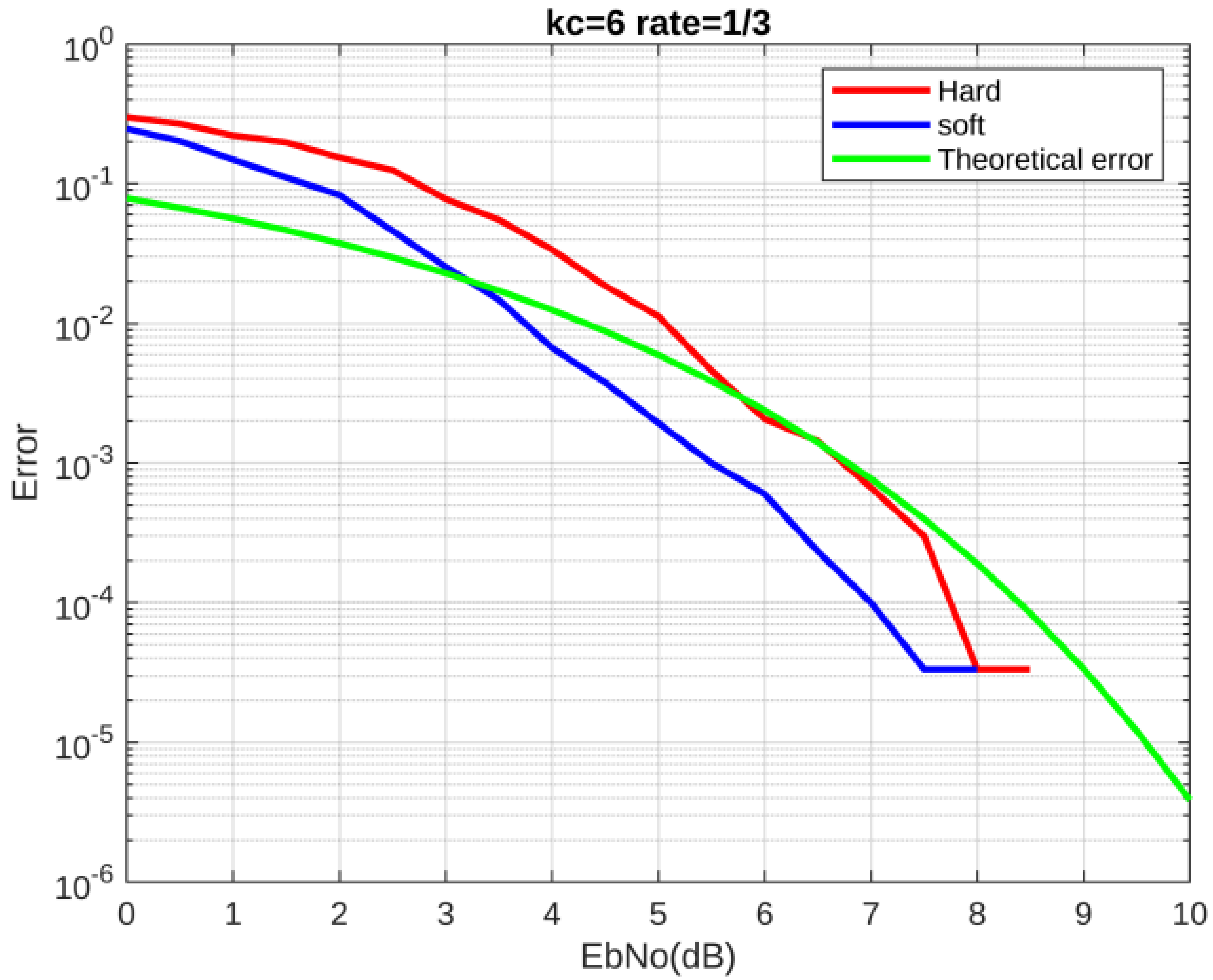
For Hard and Soft Decision Decoding Kc = 3, rate = 1/2



For Hard and Soft Decision Decoding Kc = 4, rate = 1/3



For Hard and Soft Decision Decoding Kc = 6, rate = 1/3



Application

Wireless Communication:

Convolutional codes are widely used in wireless communication systems like cellular networks (like 4G and 5G) and Wi-Fi to ensure reliable data transmission in noisy channels.

Digital Storage Systems

Convolutional codes are used in digital storage systems like hard disk drives (HDDs) and solid-state drives (SSD). By encoding data before storage and decoding it upon retrieval, convolutional codes help detect and correct errors.

Deep Space Communication

Convolutional codes are used to ensure the successful transmission of commands and data between spacecraft and ground stations. They help maintain communication reliability in the extreme conditions of deep space.

Digital Broadcasting:

Convolutional codes are utilized in digital broadcasting standards such as DVB (Digital Video Broadcasting) and ATSC (Advanced Television Systems Committee) for error correction in television and radio transmissions, they minimize the effect of signal interference.

Bibiliography

1. J. G. Proakis. Digital Communications. McGraw-Hill, 4th edition, 1995.

2. Matthew Valenti and Bibin Baby John. Introduction to Communication Systems: A Lecture on Convolution Coding and Viterbi Decoding. DA-IICT, Winter 2024.

3. Viterbi Decoding of Convolutional Codes. MIT 6.02 Lecture 9, Fall 2010.

4. An Introduction to Convolutional Codes (Youtube playlist). NPTEL. Electronics & Communication Engineering.

Group members

202201105 - Rishi Jigar Shah
202201110 - Dhruv Gaurav Suri
202201119 - Sahilkumar Sutariya
202201132 - David Lochan Baro
202201140 - Gajjar Harsh Dharmendrabhai
202201141 - Jani Ayushi Yogeshbhai
202201143 - Vegad Rutvik Rakeshbhai
202201144 - Dhangar Dhruv Kishorbhai
202201152 - Patel Ragan Dalpatbhai
202201153 - Dodiya Dev Manojbhai

Thank you!