COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

ADVANCED DIGITAL SIGNAL PROCESSING

Chapter 5: FEC Encoding/Decoding

18/11/2017

Content

- Channel coding
- Convolution Encoding
- Convolution Decoding – Viterbi Decoding

Channel coding

- Channel coding is used to is used to detect and often correct symbols that are received in error.
- Error detection can be used by receiver to generate ARQ to transmitter for a re-transmission of the frame in error, as in computer networks.
- When re-transmission is not an option: forward error correction coding, which introduces extra information (redundancy) into transmitted data for receiver to detect and correct errors.

Channel coding

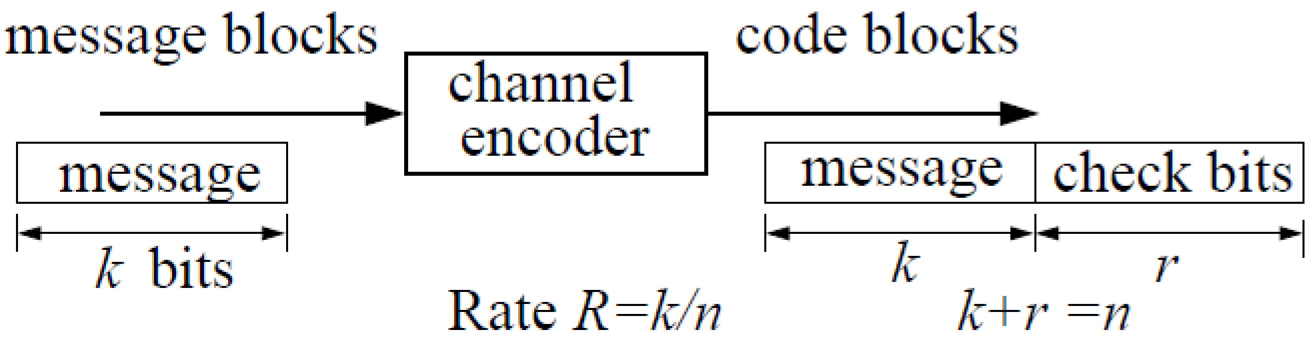
- Two types of channel coding:
  - Convolutional coding: processed on a bit-by-bit basis
    - Convolution coding – Viterbi decoding
  - Block coding: processed on a block-by-block basis
    - Hamming coding/decoding
    - Reed-Solomon coding/decoding
    - Turbo coding/decoding
    - Low-density parity check coding/decoding (LDPC)

Channel coding

- Error-correcting code encoder: generates redundant information according to the message information
- Advantage:
  - Allow receiver to detect a limited number of errors that may occur anywhere in the message, then correct it ➔ No need to retransmit ➔ Gain of channel bandwidth
- Disadvantage:
  - Cost of channel bandwidth

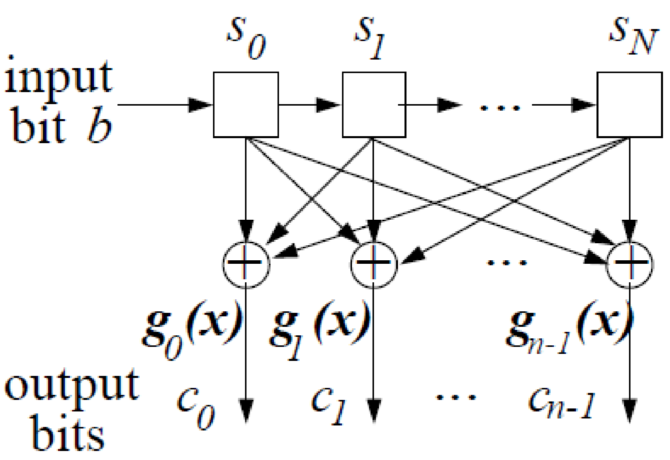
Convolution Encoding

- Convolutional Code: *CC*(n, k , N):
  - n = k + r
  - k information bits
  - r: check bits
  - R = k/n: Code rate
  - N: constraint length (or **memory** N+1 stages)
- Codebook: several codewords



Convolution Encoding

- CC(n, k=1, N) encoder
  - During each bit interval, the register is shifted one stage: s_N is shifted out, $s_i \rightarrow s_{i+1}$, and the data bit enters s_0
  - After this shift, the values of $s_1 \ldots s_N$ define the current state of the register, which is used to produce code bits
  - Modulo-2 adders produce code bits c_i, 0 <= i <= n-1, which are specified by the *n* polynomials.



$$g_i(x) = g_{i,0}\, x^0 + g_{i,1} x^1 + g_{i,2} x^2 + \ldots + g_{i,N} x^N \qquad 0 \le i \le n-1$$

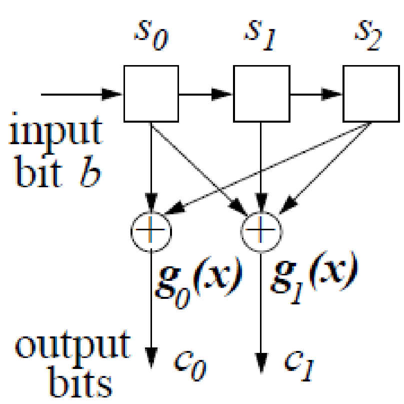
  - Output (code) bits depend on the state of $s_1 \ldots s_N$ after shift and the input bit

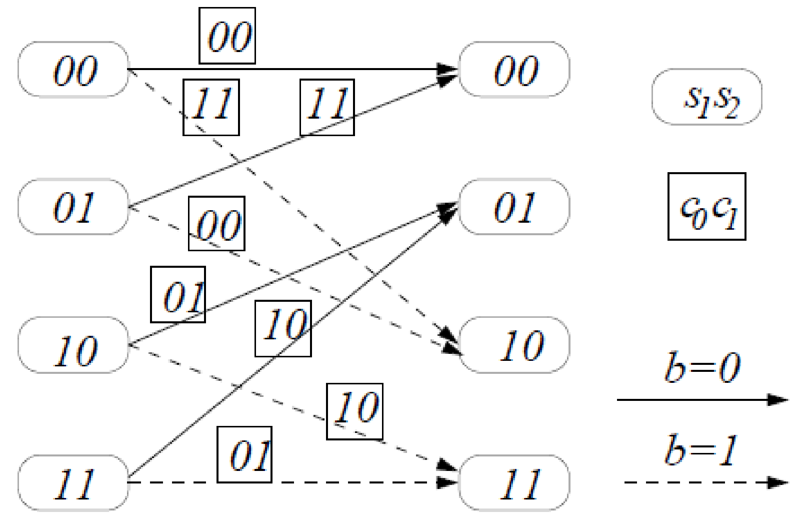
Convolution Encoding

State-Transition Diagram

- State-transition diagram describes the encoder of a CC code, showing all the 2N states and all the state transitions together with the output bits
- Example: half-rate constraint length N = 2, *CC*(2, 1, 2), defined by $g_0(x) = 1 + x^2$ and $g_1(x) = 1 + x + x^2$

input b	current $s_1 s_2$	next $s_1 s_2$	output $c_0 c_1$
0	0 0	0 0	0 0
0	0 1	0 0	1 1
0	1 0	0 1	0 1
0	1 1	0 1	1 0
1	0 0	1 0	1 1
1	0 1	1 0	0 0
1	1 0	1 1	1 0
1	1 1	1 1	0 1

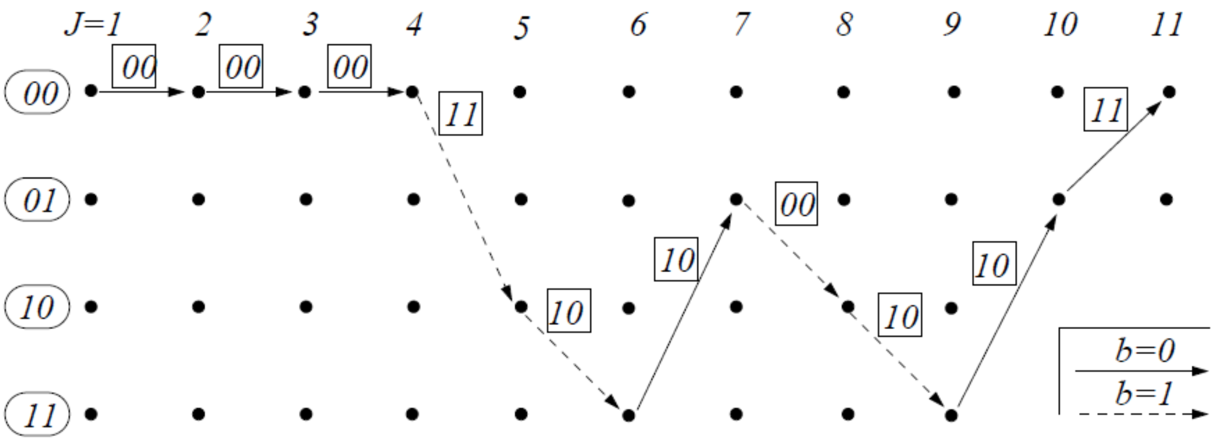




Convolution Encoding

Trellis Diagram

- An alternative way to describe a CC encoder is trellis diagram
- Same sample: $CC(2, 1, 2)$, defined by $g_0(x) = 1 + x^2$ and $g_1(x) = 1 + x + x^2$
- Information bit sequence $\cdots$ 0011011000 (assume rightmost enters encoder first)
- Recall the state-transition diagram of previous slide, we have trellis diagram:



Convolution Decoding – Viterbi Decoding

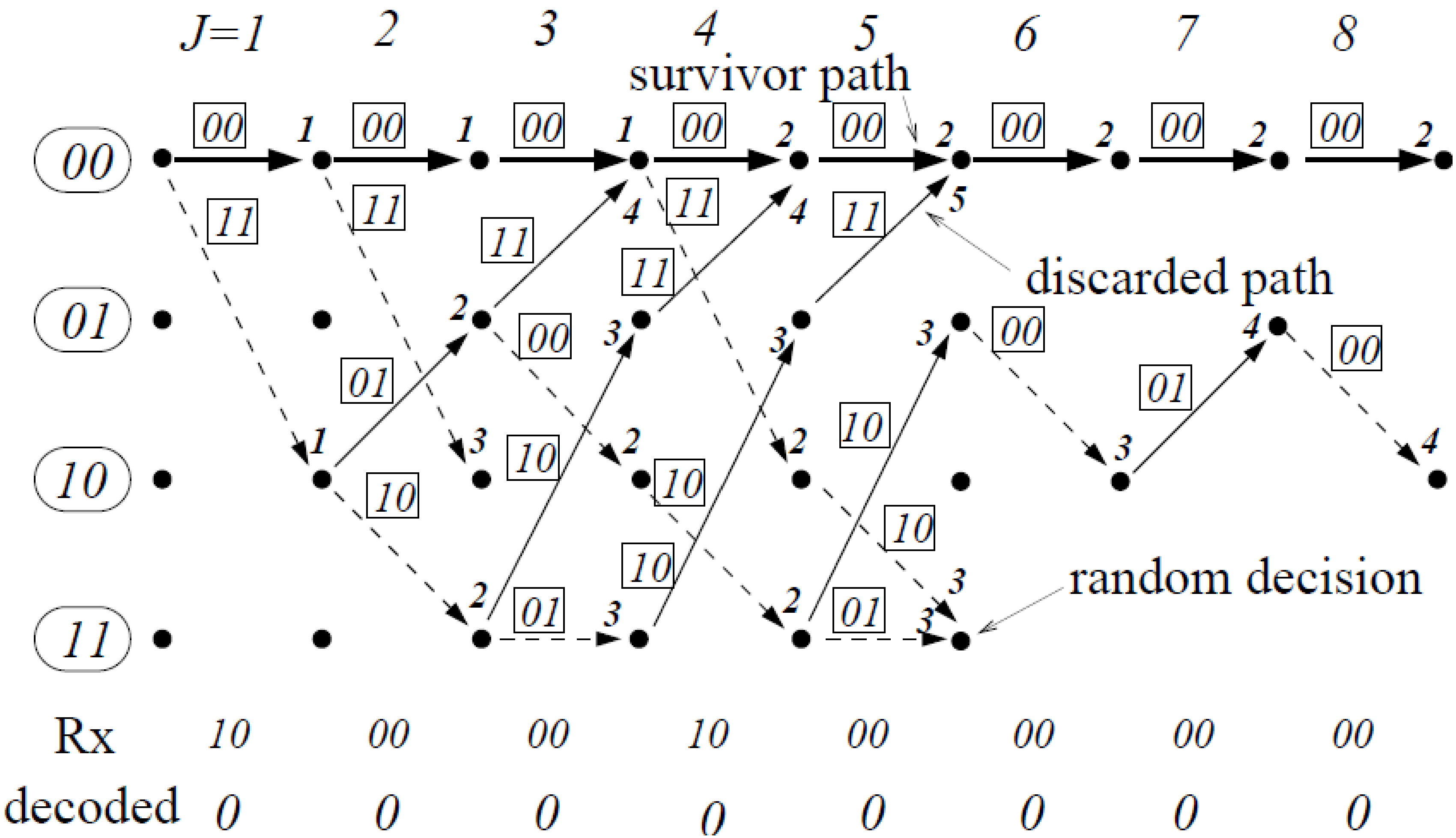
- Observe that:
  - The register sequence of state transitions is not arbitrary
  - Transmitted bit sequences are legitimate
- Therefore:
  - If a non-legitimate received sequence is encountered in the decoder, it must be due to channel errors, as such a sequence cannot be transmitted
- Decoder:
  - Choose a legitimate sequence that is most resembling to the received sequence: maximum likelihood sequence decoding
    - Efficiently implemented using the Viterbi algorithm

Hard-Input Hard-Output Viterbi Decoding

- Hard-Input Hard-Output Viterbi Algorithm
  - Hard-input decoding is for hard-decision demodulation where demodulator demapper (recall Lecture 3) has made binary decision concerning received bits
  - Hard-output decoding makes hard decision concerning decoded bits
    - Same example: $CC(2, 1, 2)$, defined by $g_0(x) = 1 + x^2$ and $g_1(x) = 1 + x + x^2$, all zero sequence is transmitted and received sequence is $10000010000000 \cdot \cdot \cdot$ (leftmost bit at leftmost position of trellis)

Hard-Input Hard-Output Viterbi Decoding

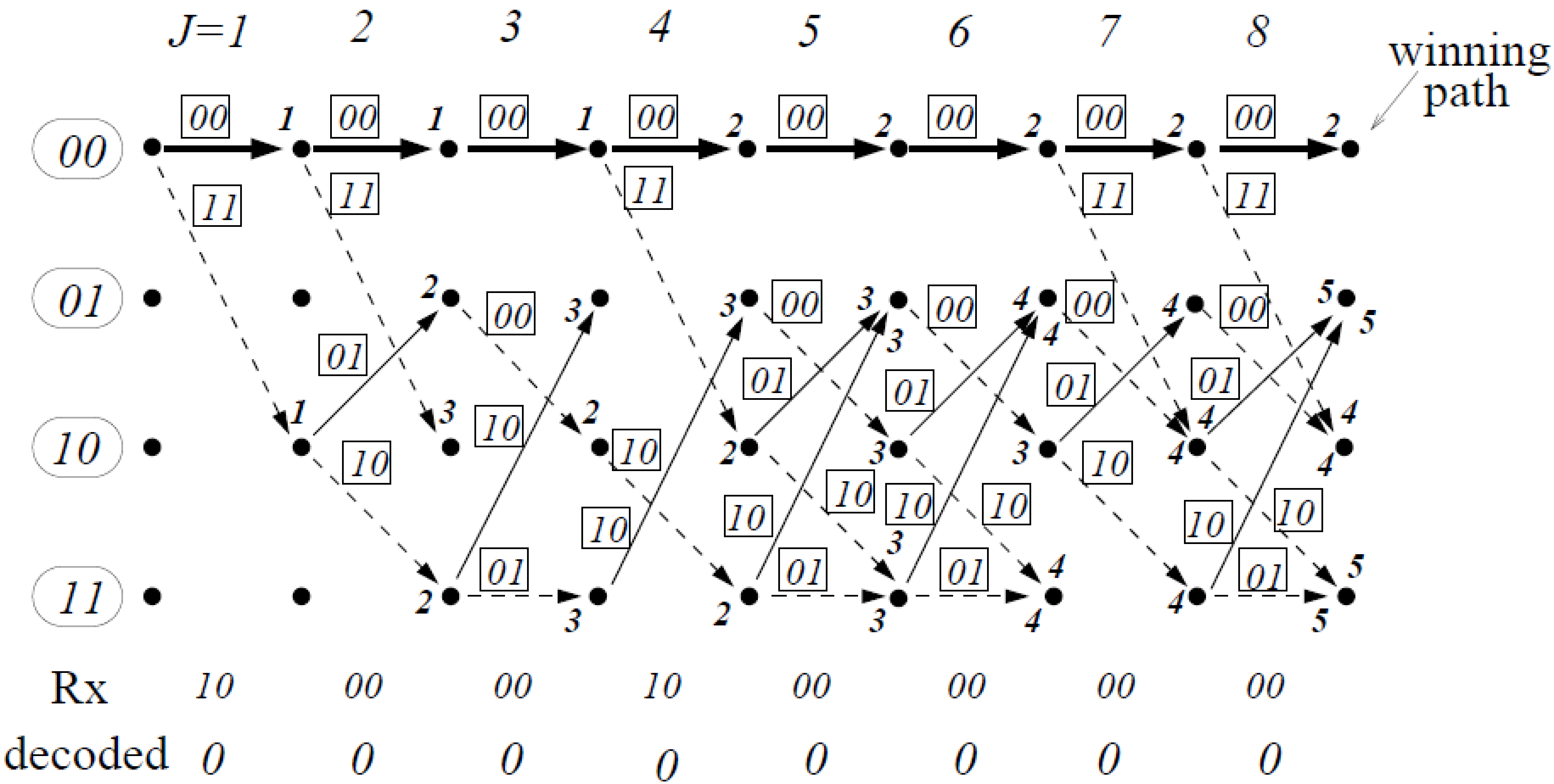
- Hard-Input Hard-Output Viterbi Algorithm



Hard-Input Hard-Output Viterbi Decoding

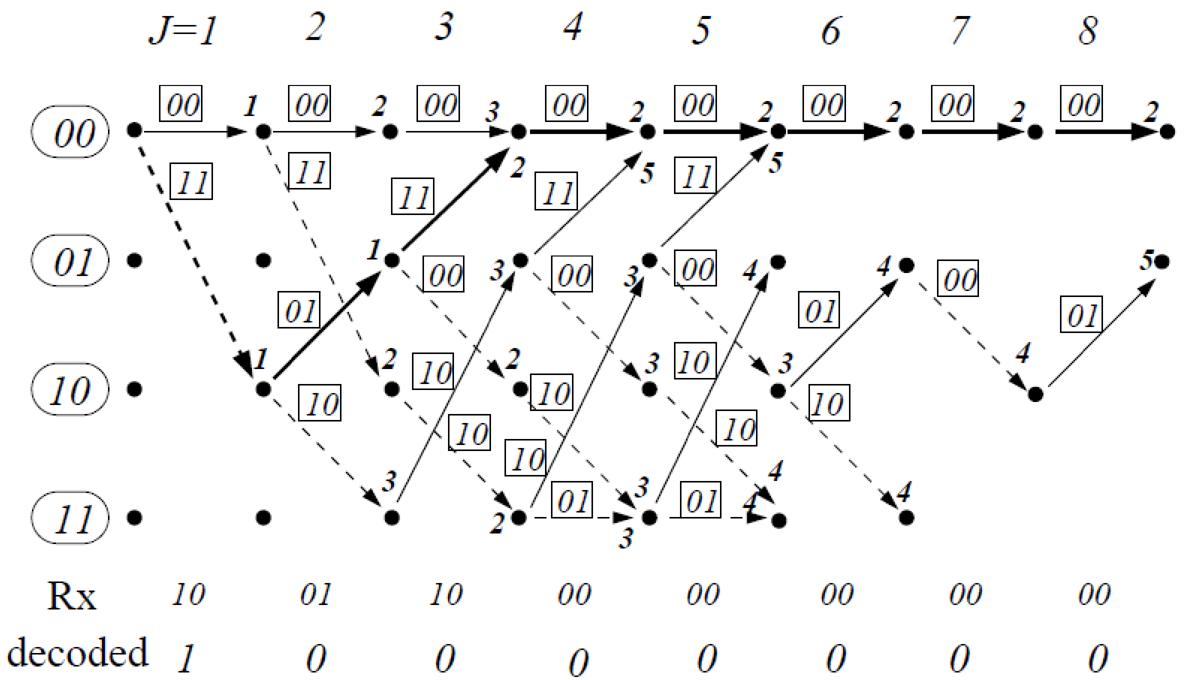
- Branch metric is the Hamming distance between the legitimate encoded (output) bits of a trellis branch at a stage and the received bits at the same stage
- Path metric is the accumulated branch metrics of a path (bold number shown on trellis diagram)
  - A path presents a legitimate sequence
- For two merging paths at a stage, the one with a larger path metric is discarded and the other, called survivor path, is kept
  - If two metrics are equal, a random decision is made to keep one path
- A final decision is made after a sufficiently large number of stages (8 for this example) to choose a winning path with the smallest path metric
  - This is MLSD, as winning path is most resembling to received sequence
- If decoding decision is correct, the winning path metric is the number of transmission errors inflicted by the channel

Hard-Input Hard-Output Viterbi Decoding



Hard-Input Hard-Output Viterbi Decoding

- Another example: $CC(2, 1, 2)$, defined by $g_0(x) = 1 + x^2$ and $g_1(x) = 1 + x + x^2$, all zero sequence is transmitted but received sequence is 10011000000000 $\cdots$ (leftmost bit at leftmost position of trellis)



- Winning path is most resembling to received sequence
- But this is erroneous decoding. If decoding decision is incorrect, the winning path metric is not the number of transmission errors caused by the channel

Soft-Input Hard-Output Viterbi Decoding

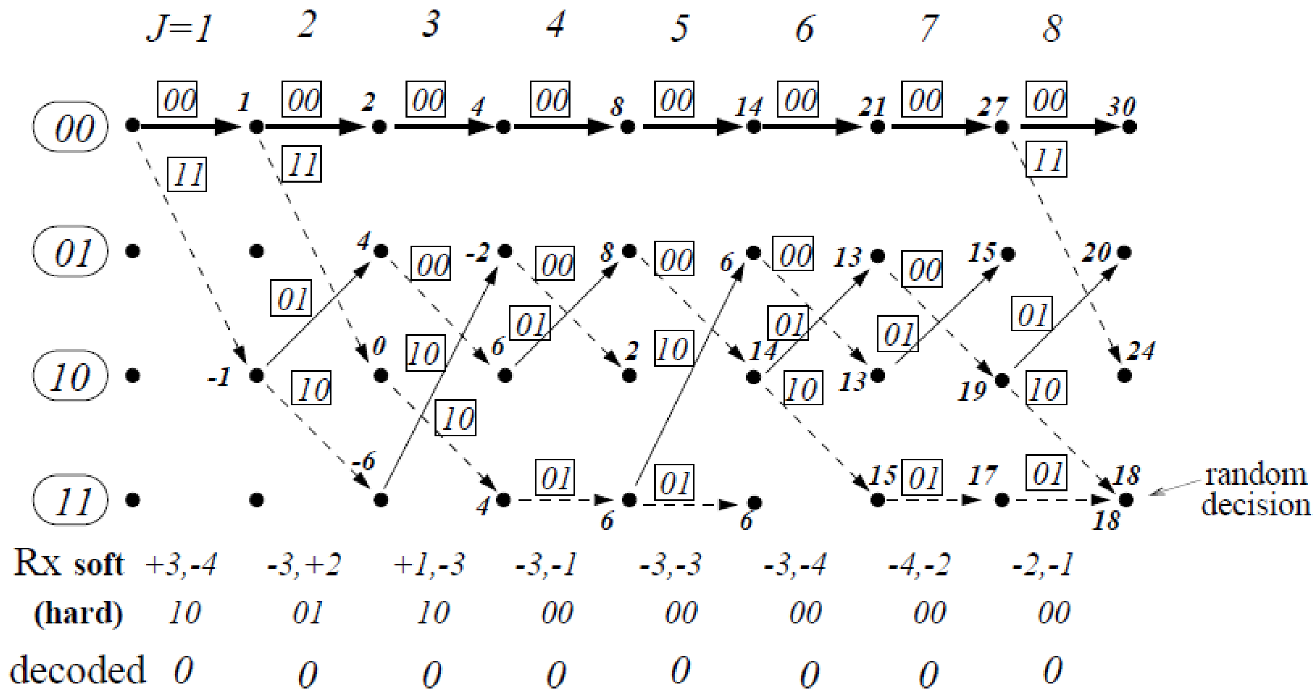
- Soft-input decoding is for soft-decision demodulation where demapper does not make hard binary decision but gives confidence measure concerning probability of a binary bit being 1 or 0
  - As the decoder has more information, it does better than hard-input decoding
  - Recall slide 112, this confidence measure or soft decision is given as log likelihood ratio
  - Here we consider a quantised confidence measure alternative
- Assume we use $\pm 4, \pm 3, \pm 2, \pm 1$ 8-level (or 3-bit quantisation) for confidence measure with interpretation:
  - +4: extremely like to be 1; +3: strongly like to be 1; +2: like to be 1; +1: weakly like to be 1
  - −4: extremely like to be 0; −3: strongly like to be 0; −2: like to be 0; −1: weakly like to be 0
- Hard-output decoder outputs hard decoded bits, but soft-output decoder outputs sequence of log likelihood ratios or confidence measures and it is for iterative decoding

Soft-Input Hard-Output Viterbi Decoding

- Previous example: $CC(2, 1, 2)$, defined by $g_0(x) = 1 + x^2$ and $g_1(x) = 1 + x + x^2$
  - An all zero sequence is transmitted but received soft decision sequence is:
  - +3,−4,−3,+2,+1,−3,−3,−1,−3,−3,−3,−4,−4,−2,−2,−1, · · ·
  - Hard demapper would produce the hard decisions 10011000000000 · · · (with the leftmost bit at leftmost position of trellis)
  - As previous slide shows, HIHO VA chooses a legitimate sequence that is most resembling to 10011000000000 · · · , but it is erroneous decoding

Soft-Input Hard-Output Viterbi Decoding

- Trellis of SIHO VA decoding:
- Previously, HIHO VA gave erroneous decoding but SIHO VA produces correct decoding



- In quantised confidence measure based soft-input decoding, the meaning of metric has changed
  - If trellis branch output bits are 01 and receive soft decisions are +3,+1, it has a penalty −3 for the 1st bit and a credit +1 for the 2nd bit, so that branch metric is (−3) + (+1) = −2
  - If trellis branch output bits are 00 and receive soft decisions are +3,−4, it has a penalty −3 for the 1st bit and a credit +4 for the 2nd bit, so that branch metric is (−3) + (+4) = +1
- The winning path is the survivor path with the largest path metric (credit)
- Straightforward to generalise to log likelihood ratio based soft-input decoding

Summary

- Convolutional code CC(n, k,N):
  - Code rate R = k/n, constraint length N (memory length N + 1)
  - Concepts of encoder states, state transitions, and generator polynomials
- CC(n, k = 1,N) encoding:
  - Encoder circuit, table of state transitions and output bits
  - State-transition diagram, and trellis diagram
- CC(n, k = 1,N) decoding:
  - Maximum likelihood sequence decoding, and trellis diagram based Viterbi decoding
  - Hard-input and hard-output decoding, soft-input and hard-output decoding

END