Diversity in wireless communication systems

- *Time diversity*: relates to the *coherence time*
  - Transmission over multiple time slots by channel coding plus interleaving
  - The amount of diversity is small over very slow fading channels
- *Frequency diversity*: relates to the *coherence bandwidth*
  - Transmission over multiple frequency bands
  - The amount of diversity is small over very flat fading channels
- *Spatial diversity*: relates to the *coherence distance*
  - Transmission and reception employing multiple antennas.

Time diversity: interleaving and coding

- The main idea was formalized by Claude Shannon in 1948.
- Channel coding introduces some redundancy in the transmitted bits to either
  - Detect errors at the receiver
  - Improve the bit error probability at the receiver.
- The redundancy is measured by the code rate $R = \frac{k}{n} < 1$, the ratio between k, the number of bits at the input of the encoder and n, the number of bits at the output of the encoder.
- Channel codes have been initially studied for AWGN transmissions but can be employed over fading channels.

Channel codes

- Linear algebraic codes are the most used type of codes:
  - Block codes
  - Convolutional codes
- All algebraic operations are performed in the GF(2), the Galois field of two elements $\{0,1\}$, where the '$+$' and '$\times$' operations are defined as

XOR

+	0	1
0	0	1
1	1	0

AND

x	0	1
0	0	0
1	0	1

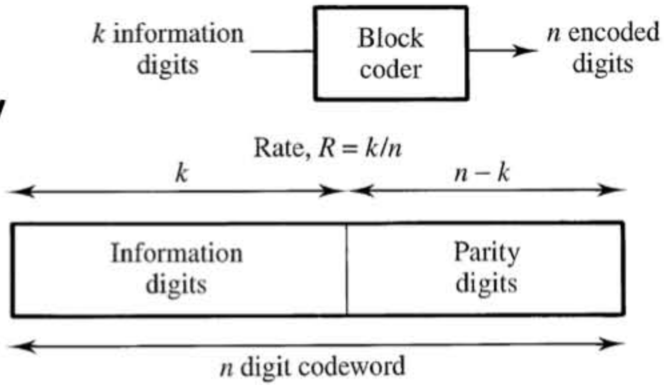
- While in general fields like the set of real number R have an infinite number of elements, Galois fields have a *finite number of elements*.

Block codes: encoder

- Block codes are most of the times in *systematic* form: the coded word is formed by k information bits and $n-k$ parity bits.
- The encoder can be represented as the code *generator matrix* G the encoder.
- The word u of k bits is encoded in the coded word d of n bits

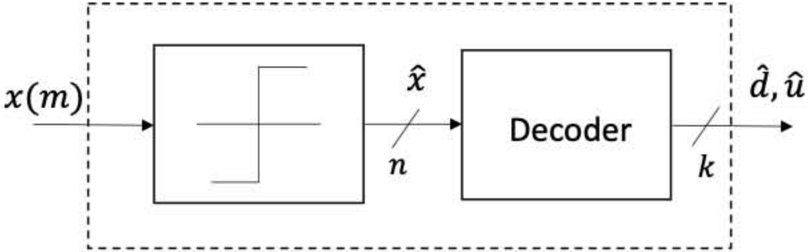
$$d = uG$$

- The encoder adds redundancy so that all coded words differ of as many bits as possible.

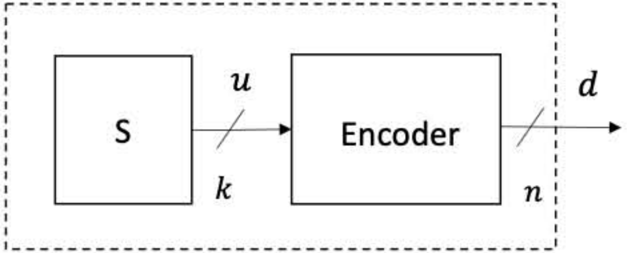


Error detection coding

- Very simple technique: one or more bits of parity are added at the end of a word.
- The receiver computes the parity check applying the same algorithm implemented at the receiver:
  - If the result computed at the receiver matches the parity check bits, the parity bits are discarded and the received bits are considered error-free.
  - If the result does not match the parity check bits, there is one or more errors in the string of received bits and the receiver requests a re-transmission.



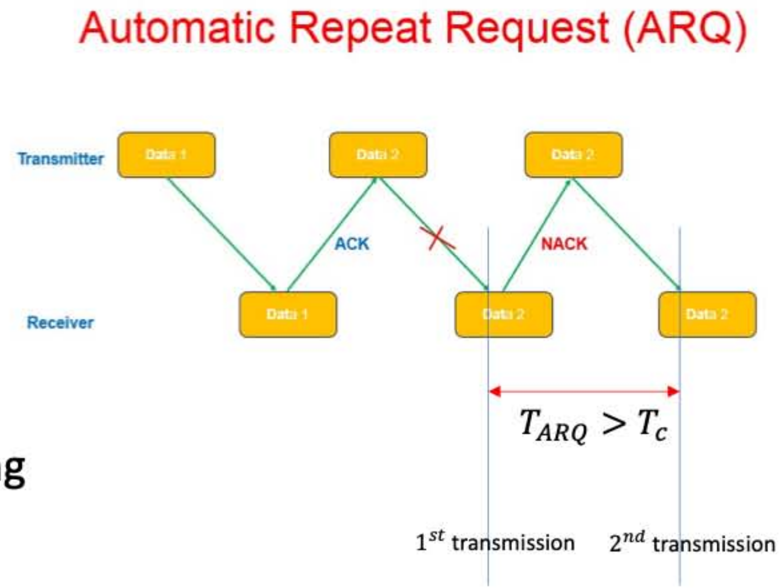
Parity check code



- Parity check code with rate $R = 7/8$.
  - Every word of 7 bit, the encoder adds a parity control bit: 1 if the word contains an odd number of '1', 0 if the word contains an even number of '1'.
  - The output of the encoder is an 8-bit word that always contains an even number of '1'.
  - The generator matrix of the code is the 7×8 matrix
  $$G = [I_7, 1_7]$$
  - The product $u1_7 = \sum_{i=1}^{7} u_i$ is a modulo-2 sum and it is '1' for an odd number of '1' and 0 otherwise.
- At the receiver, the decoder checks the number of bits in each 8-bit word an detects an error when the number of '1' is odd!
- Not all the errors are detected!

Data retransmission

- The receiver feeds back an ACK for a correct reception and a NACK for a faulty reception.
- After receiving a NACK, the transmitter resends the data packet
- ARQ exploits the *time diversity* of the channel by retransmitting the data after T_{ARQ}, a time interval, longer than the channel coherence time T_c.
- The new transmission will experience a different and hopefully better channel
- Advanced receivers are capable of combining the two received messages to further improve the chances of a successful reception.



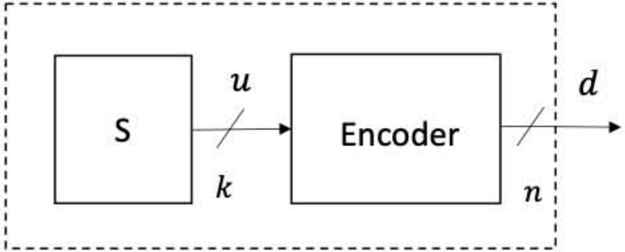
Error correction coding and channel capacity

- Channel codes can be employed to *correct* the errors introduced by the channel.
- Given a communication channel of bandwidth B, Shannon proved that the *channel capacity* can be computed as

$$C = B\log_2(1 + SNR)\ \mathrm{b/s}$$

  - For any transmission with rate $R < C$ and an arbitrarily small ϵ, it is possible to find an error correction code such that the error probability is $P_e < \epsilon$.
  - On the contrary, if $R > C$ it is not possible to find a code that can make the probability of error of the transmission over the channel arbitrarily small.

Block codes: repetition code



- The simplest block code is the $k\ =\ 1, n =\ 3$ *repetition code.*
- It is composed by $2^k = 2$ codewords:

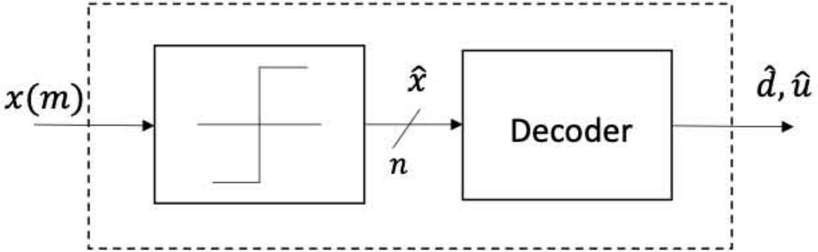
$$u = 0 \Longrightarrow \mathrm{d} = [0\ 0\ 0], \qquad u = 1 \Longrightarrow \mathrm{d} = [1\ 1\ 1]$$

- The generator matrix is

$$\boldsymbol{G} = [1\ 1\ 1]$$

- The decoder takes a majority decision
- $\hat{x}\ = \left.\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix}\right\} \rightarrow \hat{d} = [0\ 0\ 0] \rightarrow \hat{u} = 0, \hat{x}\ = \left.\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix}\right\} \rightarrow \hat{d} = [1\ 1\ 1] \rightarrow \hat{u} = 1$

Block codes: decoder

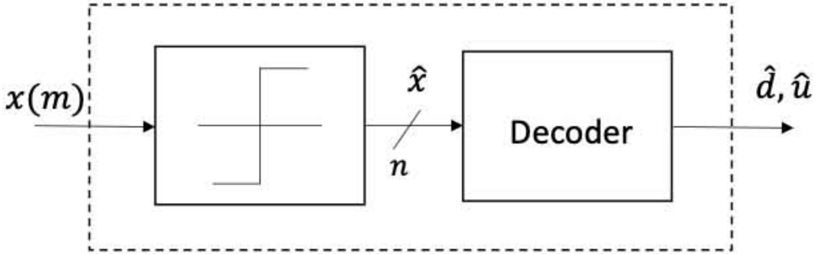


- After having received the string of n bits $\hat{x}$, the decoder selects the codeword $\hat{d}$ that has minimal distance from $\hat{x}$

$$\hat{d} = \arg\min_{d} d(d, \hat{x})$$

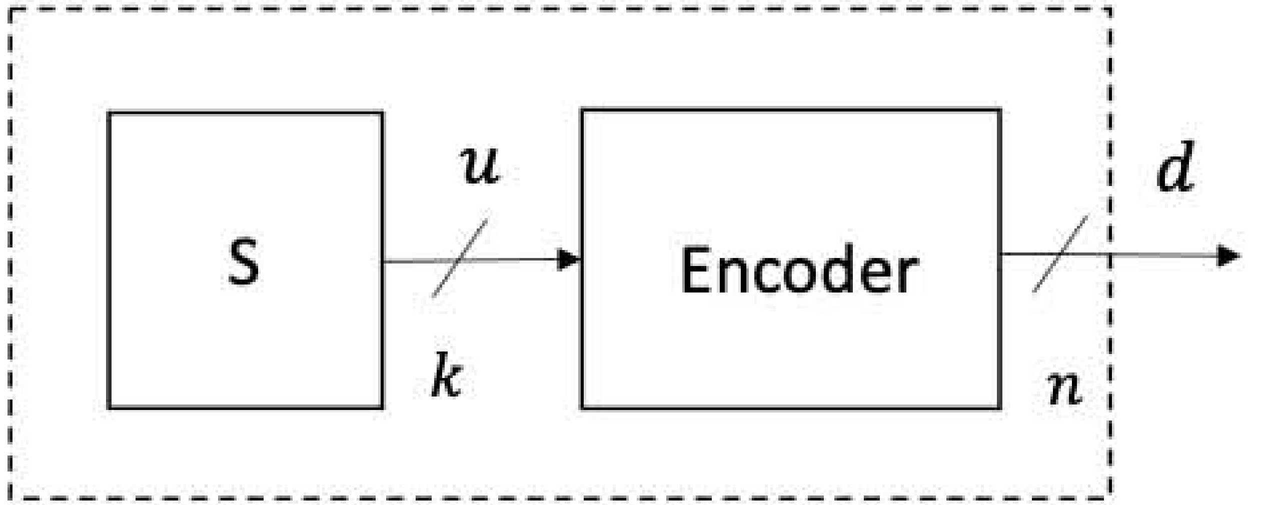
- In GF(2) the distance between codewords is computed as the number of bits that are different in the two string of bits (Hamming distance).

Block codes: decoder



- An error event occurs when, due to the noise, the received vector $\hat{x}$ is closer to a codeword different from the transmitted one.
- Codes where the distance between words is large are more robust against noise and fading than codes where the distance is small.
- Hamming distance is larger when k and n are large, unfortuntely also the receiver complexity grows with k.

Convolutional codes: encoder



- The encoder of a (n, k, L) convolutional code works as n linear filters in GF(2) in parallel.
- The new parameter L is the *constraint length* of the encoder and is the number of input words of k bits that affect the n output bits.
- Accordingly, the encoder output depends on both the current input and the previous $L - 1$ input words.
- The most important difference w.r.t. block codes is that we have now a system with *memory*: the codeword $d^{(i)}$ depends not only on $u^{(i)}$ but also on $u^{(i-1)}, \ldots, u^{(i-L+1)}$

Code generators

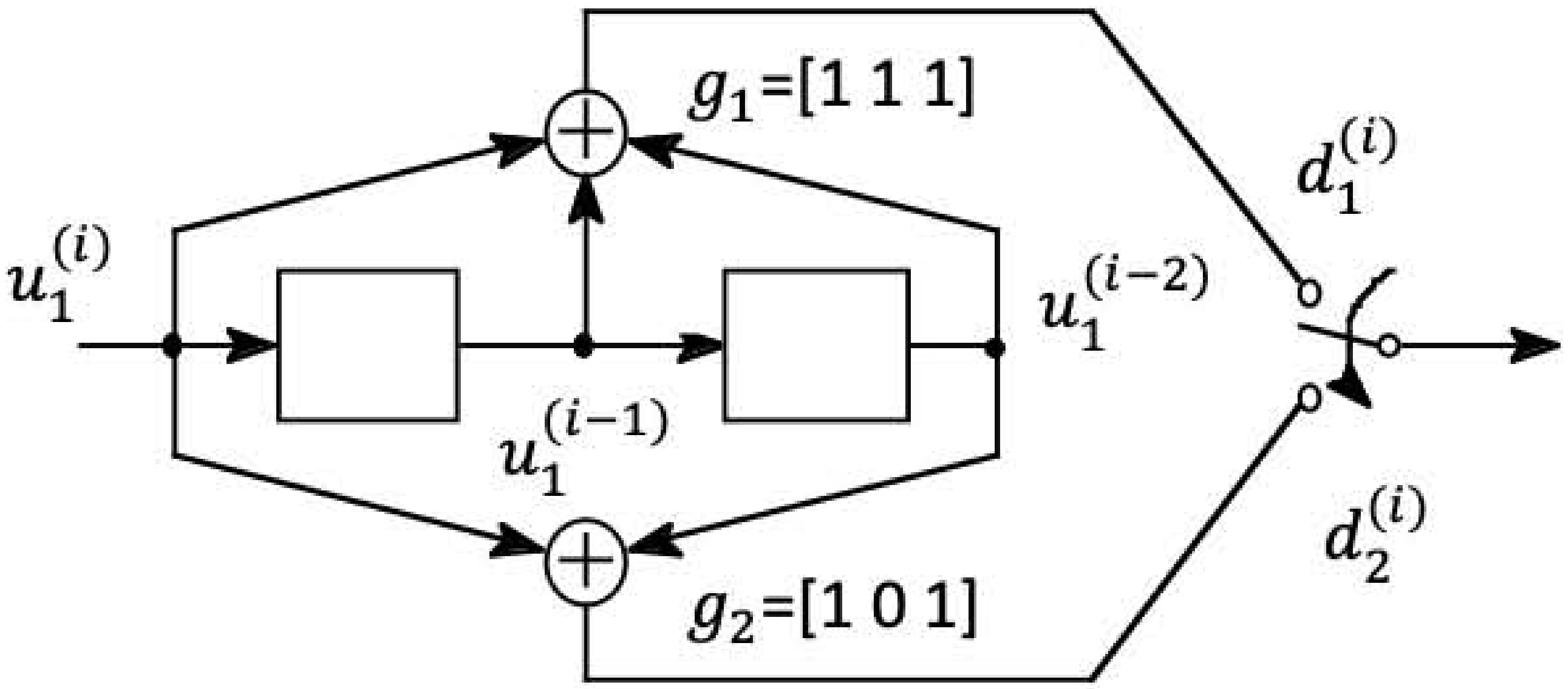
- Each of the n bit of a codeword is obtained by a convolution in $GF(2)$. The impulse response of each of the n convolutions is given by a different generator vector of length kL.
- In GF(2) the filter impulse response is a string of ‘0’ and ‘1’ .
- For example, the generators for a convolutional code with $k = 1, n = 2$ and $L = 3$ are two vectors of three elements each.
- The bits $d_j^{(i)}$ ($j = 1,2$) of the output codeword $d^{(i)}$ are computed as

$$d_j^{(i)} = \sum_{l=0}^{2} g_j(l)u^{(i-l)} = g_j(0)u^{(i)} + g_j(1)u^{(i-1)} + g_j(2)u^{(i-2)}$$

The convolutional code (2,1,3)

- Let's study the convolutional code (2,1,3) with generator $g_1 = [1\ 1\ 1]$ and $g_2 = [1\ 0\ 1]$. The code rate is $R = 1/2$ and the encoder has a memory of $3 - 1 = 2$ elements.
- The two bits composing the codeword $d^{(i)}$ are computed as

$$d_1^{(i)} = u^{(i)} + u^{(i-1)} + u^{(i-2)}, d_1^{(i)} = u^{(i)} + u^{(i-2)}$$

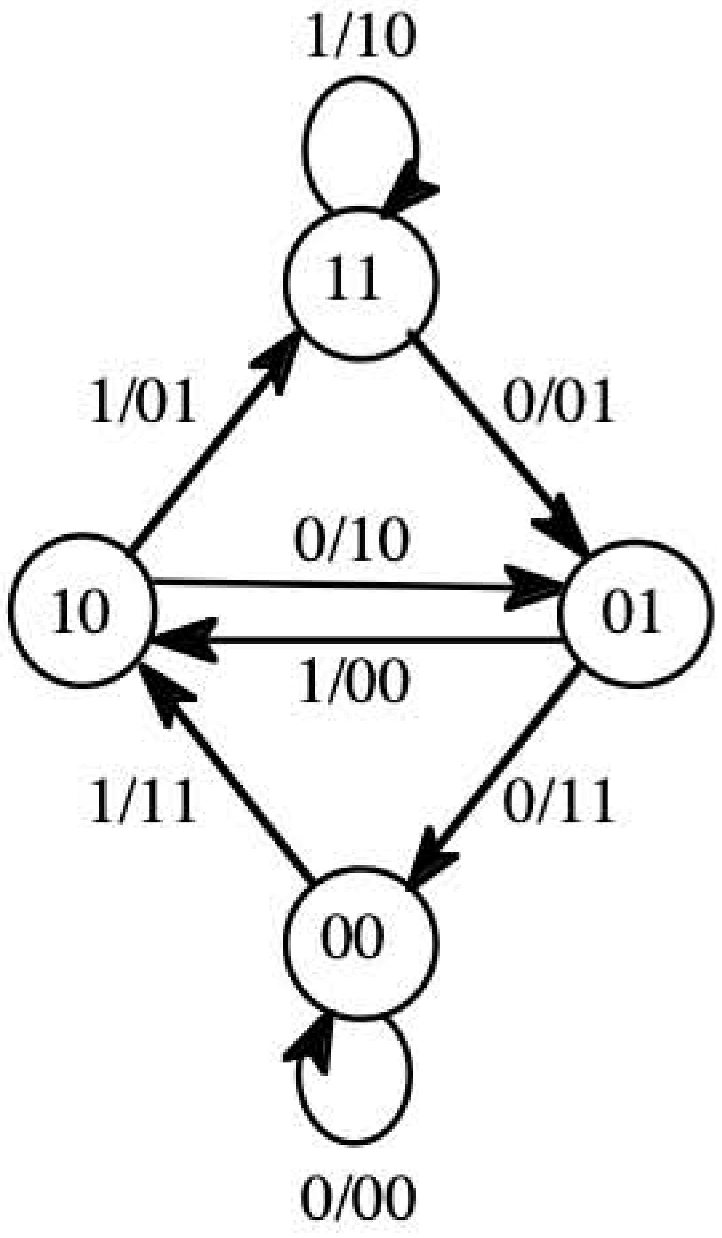


$$\begin{cases} d_1^{(i)} = u_1^{(i)} + u_1^{(i-1)} + u_1^{(i-2)} \\ d_2^{(i)} = u_1^{(i)} + u_1^{(i-2)}. \end{cases}$$

Convolutional codes: state diagram

- The encoder can be represented as a *finite-state machine*.
- The output of the encoder depends on two elements:
  - The input bit;
  - The *state* of the encoder: the content of the memory cells of the shift register.
- Each incoming bit determines
  - A new output sequence;
  - A new state
- The state diagram captures the transitions in the encoder.

State diagram for the (2,1,3) encoder



State transitions

$$00 \xrightarrow{0} 00,\ 00 \xrightarrow{1} 10$$

$$10 \xrightarrow{0} 01,\ 10 \xrightarrow{1} 11$$

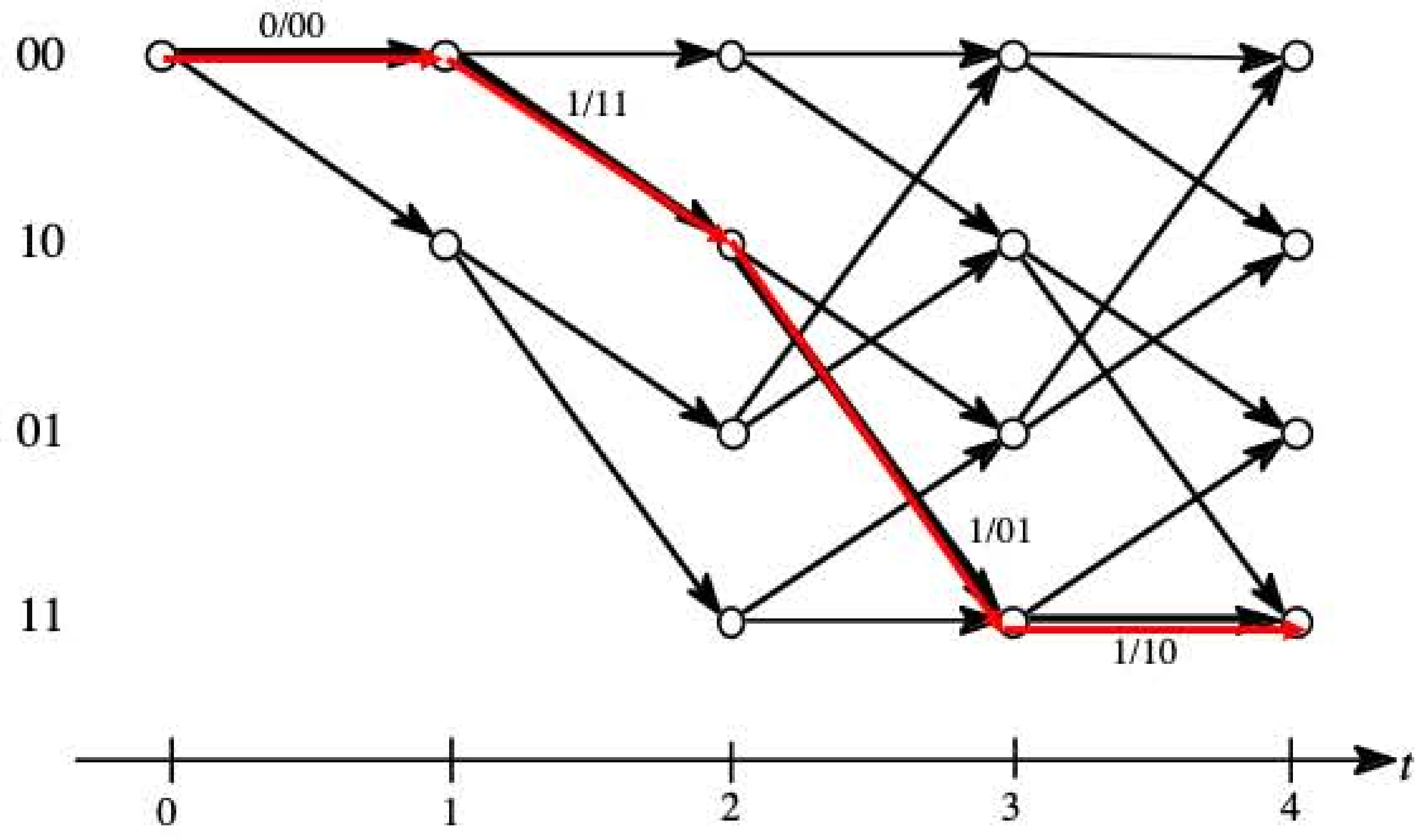
$$11 \xrightarrow{0} 01,\ 11 \xrightarrow{1} 11$$

$$01 \xrightarrow{0} 00,\ 01 \xrightarrow{1} 10$$

Convolutional codes: trellis diagram

- The time evolution of the encoder is captured by a *trellis* diagram, which represents the evolution of the states of the encoder as a function of time.
- Any given input sequence and the corresponding encoded word can be represented as a path on the trellis.

For the (2,1,3) encoder with generators $g_1 = [1\ 1\ 1]$ and $g_2 = [1\ 0\ 1]$, the path corresponding to the input sequence $u^{(0)} = 0$, $u^{(1)} = 1$, $u^{(2)} = 1$, $u^{(3)} = 1$ is highlighted in red.



Convolutional codes: decoder

- Unlike in block codes, the coded bits are not organized in blocks but they are a continuous flow of data.
- A transmission of N codewords implies that the sequence u of kN bits has been encoded into a sequence d of nN bits.
- Each input bit impacts on nL different coded bits so that the whole sequence needs to be *jointly* decoded.
- The decoder's task is to select among all the possible 2^{kN} convolutionally encoded sequences the one that minimizes the distance from the received sequence $\hat{x}$ of nN bits
$$\hat{d} = \arg\min_{d} d(d, \hat{x})$$

Convolutional codes: the Viterbi algorithm

- Until the discovery of the Viterbi algorithm in 1967, the use of powerful convolutional codes was limited by the exponential complexity of the decoder.
- Theoretically, to take a decision on nN bits the decoder needs to compare 2^{kN} different sequences.
- The Viterbi algorithm is an iterative algorithm that scales down the complexity from exponential to linear in N.
- The main idea is that of all the 2^{kN} possible paths on the trellis, a very large number of them can be discarded because non relevant.
- At each step of the algorithm, the algorithm selects a number of surviving paths on the trellis equal to the number of states of the encoder.

Andrew James Viterbi
(born Andrea Giacomo Viterbi in Bergamo, Italy)
is Professor of Electrical Engineering at the University of Southern California's Viterbi School of Engineering, which was named after him in recognition of his **$52 million** gift.

Convolutional codes: the Viterbi algorithm

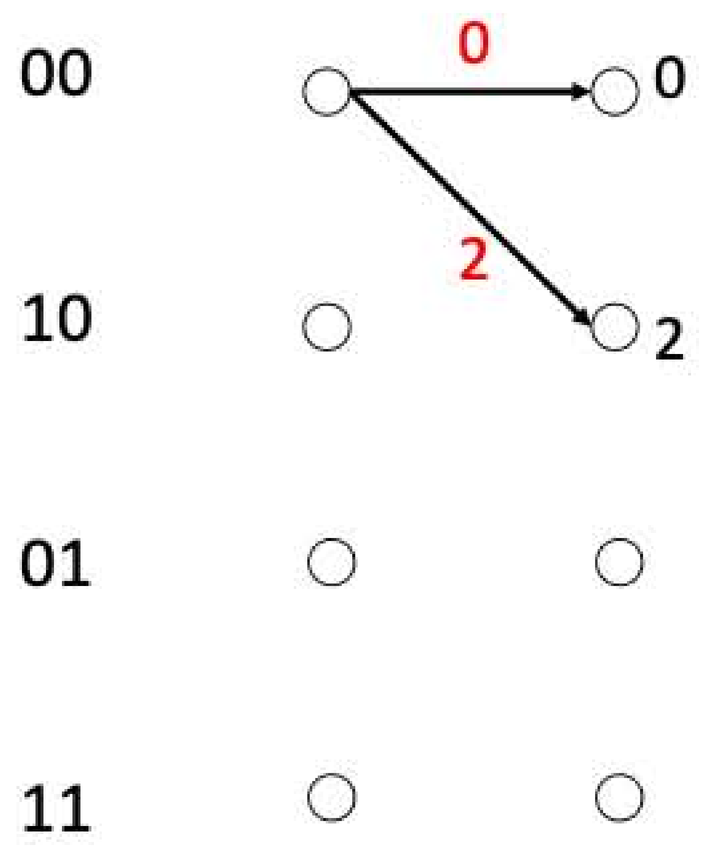
- The algorithm's objective is to find the path on the trellis that minimizes the distance from the received sequence of bits.
- Starting and finishing in a known state, the algorithm:
  1. Computes all the possible state transitions on the trellis and for each transition the distance from the corresponding received sequence of n bits (*branch metrics*);
  2. Assuming that there is a single path arriving at each state, computes the accumulated distance (*cumulated metric*) for each branch out of a given state. The distance is the sum of the cumulated metric of the path arriving at the state and the branch metric;
  3. Discard all the paths leading to one state except the one with minimum accumulated distance (*survivor path*).
- The survivor path leading to the last state is algorithm's output and the final cumulated metric is the number of corrected errors.

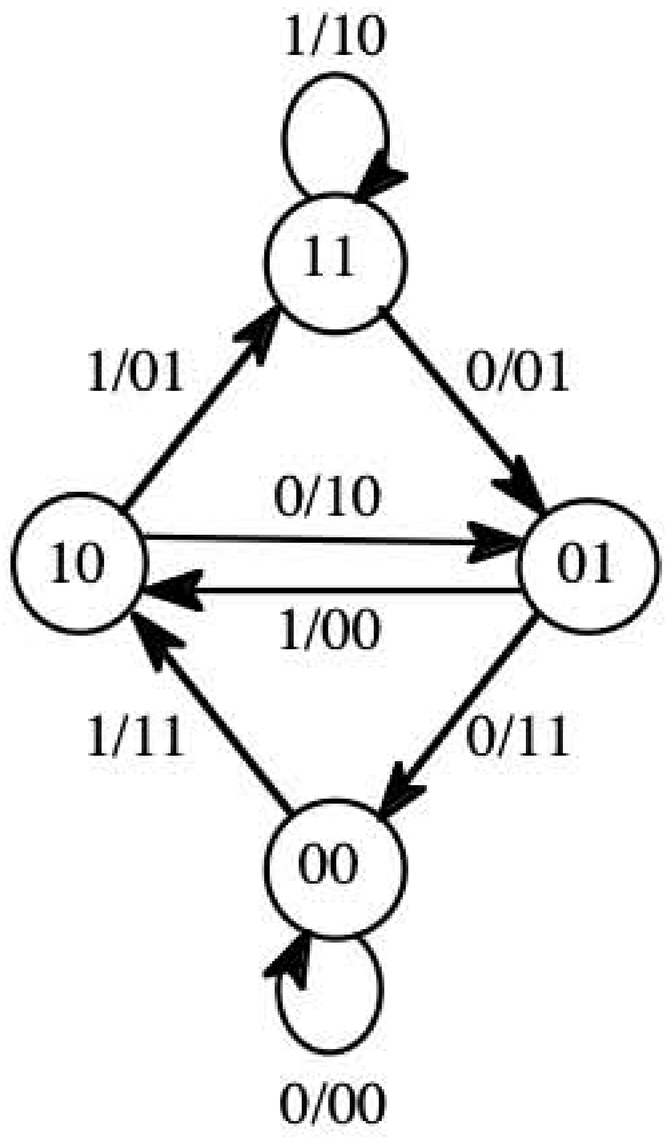
Toy example for the (2,1,3) code

- Let's assume that $N = 5$, the initial state of the encoder is 00, the sequence arriving at the encoder is
$$u^{(0)} = 0, u^{(1)} = 1, u^{(2)} = 0, u^{(3)} = 0, u^{(4)} = 0, u^{(5)} = 0,$$
so that the output of the decoder is
$$d^{(0)} = 00, d^{(1)} = 11, d^{(2)} = 10, d^{(3)} = 11, d^{(4)} = 00, d^{(5)} = 00.$$
- The sequence at the receiver is
$$\hat{x}^{(0)} = 00, \hat{x}^{(1)} = 10, \hat{x}^{(2)} = 10, \hat{x}^{(3)} = 11, \hat{x}^{(4)} = 00, \hat{x}^{(5)} = 00$$

Viterbi decoding

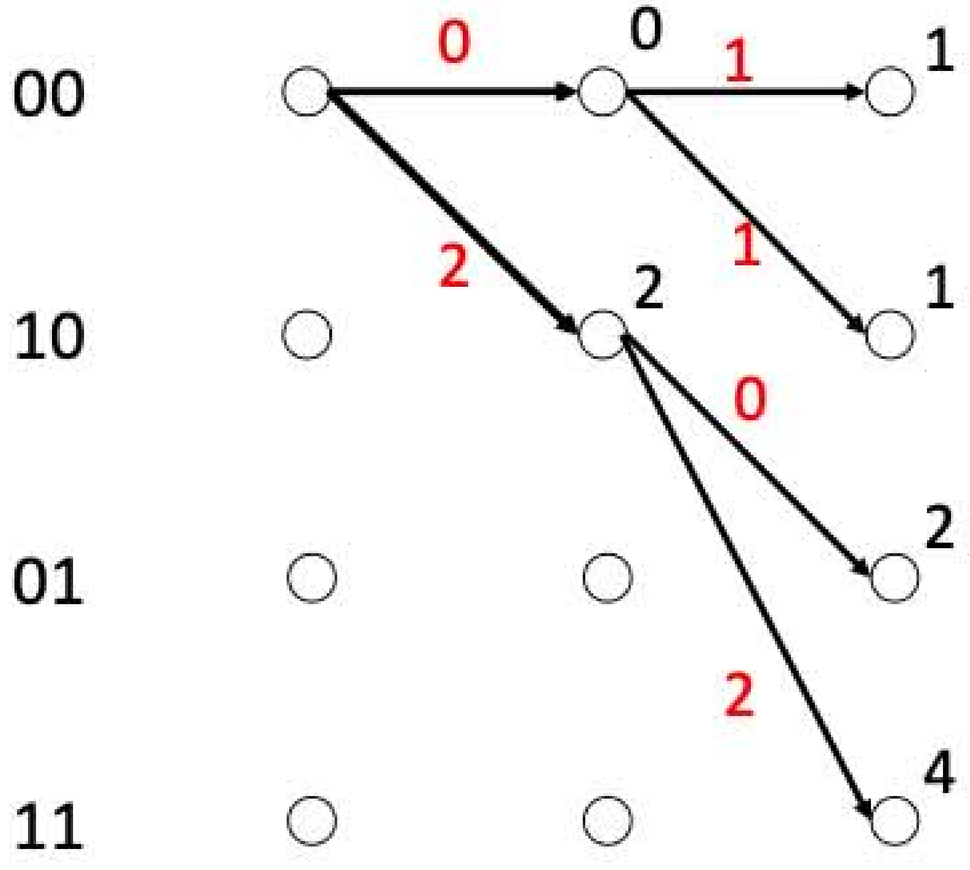
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

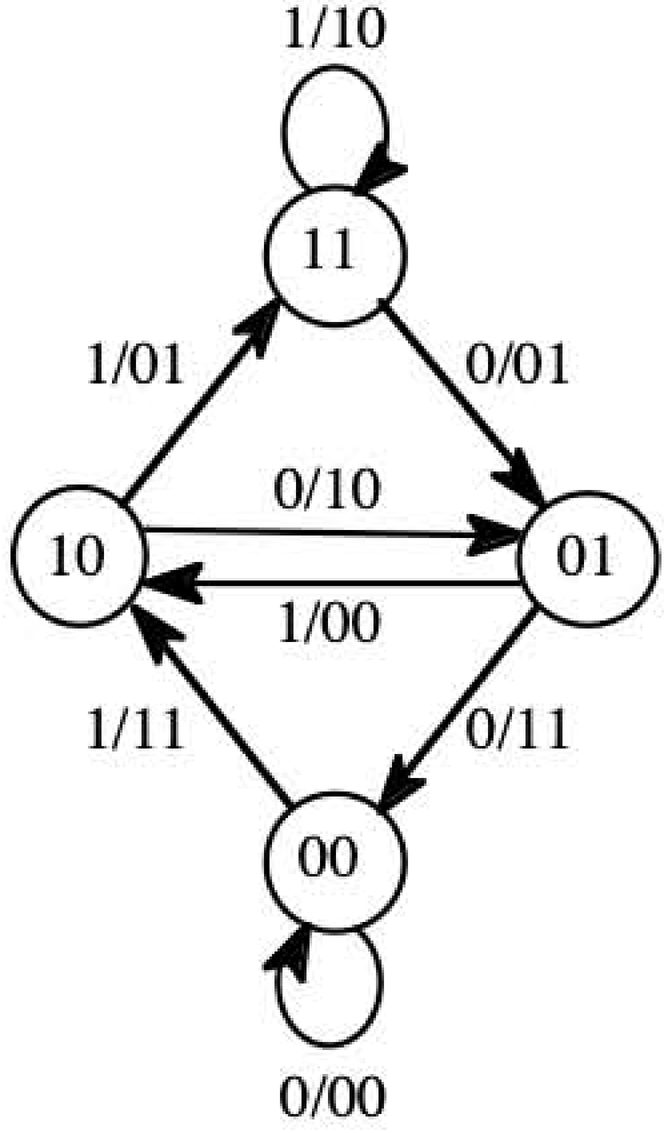




Viterbi decoding

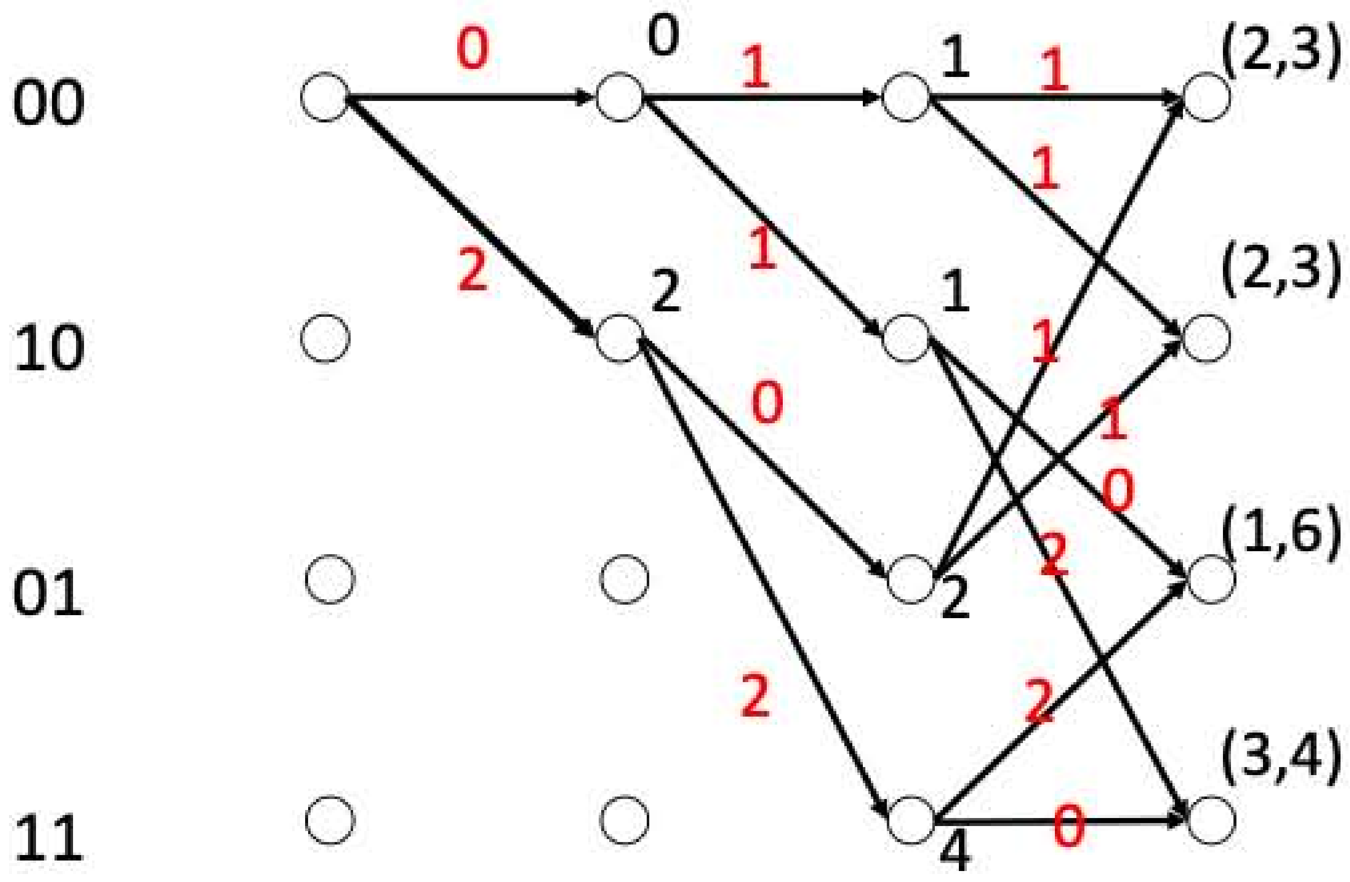
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

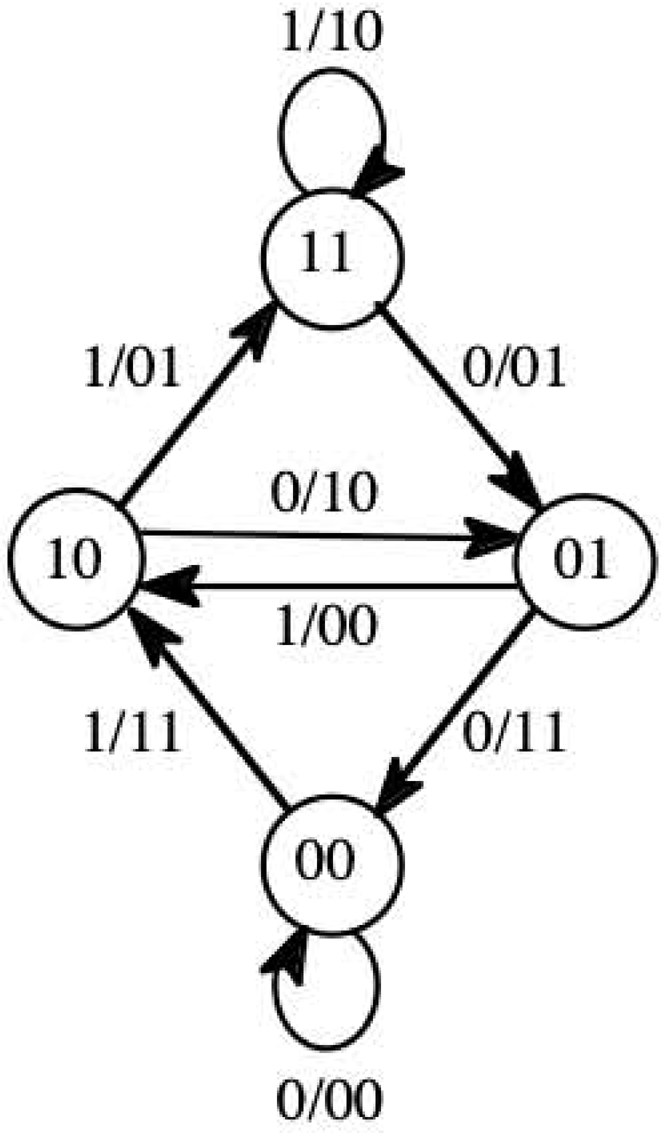




Viterbi decoding

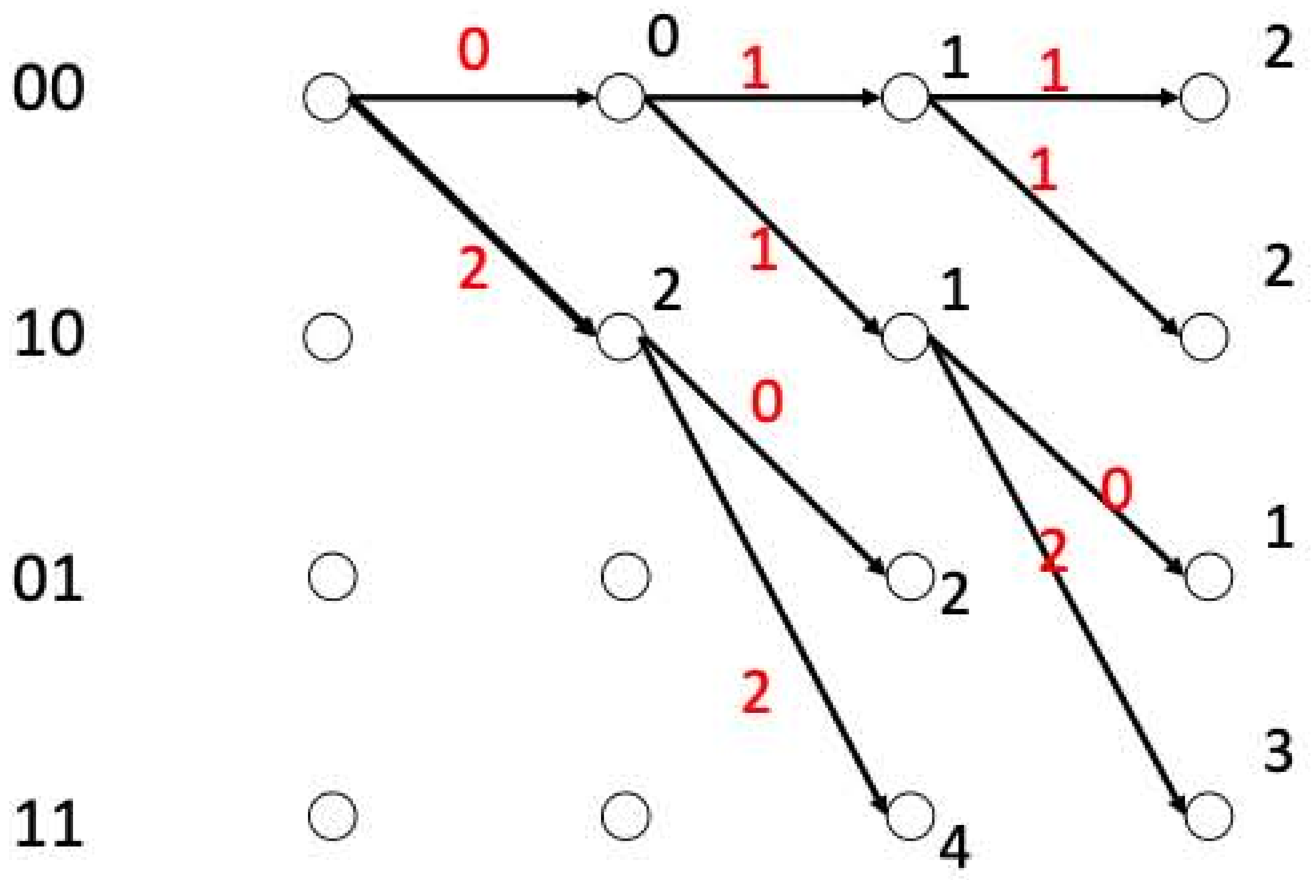
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

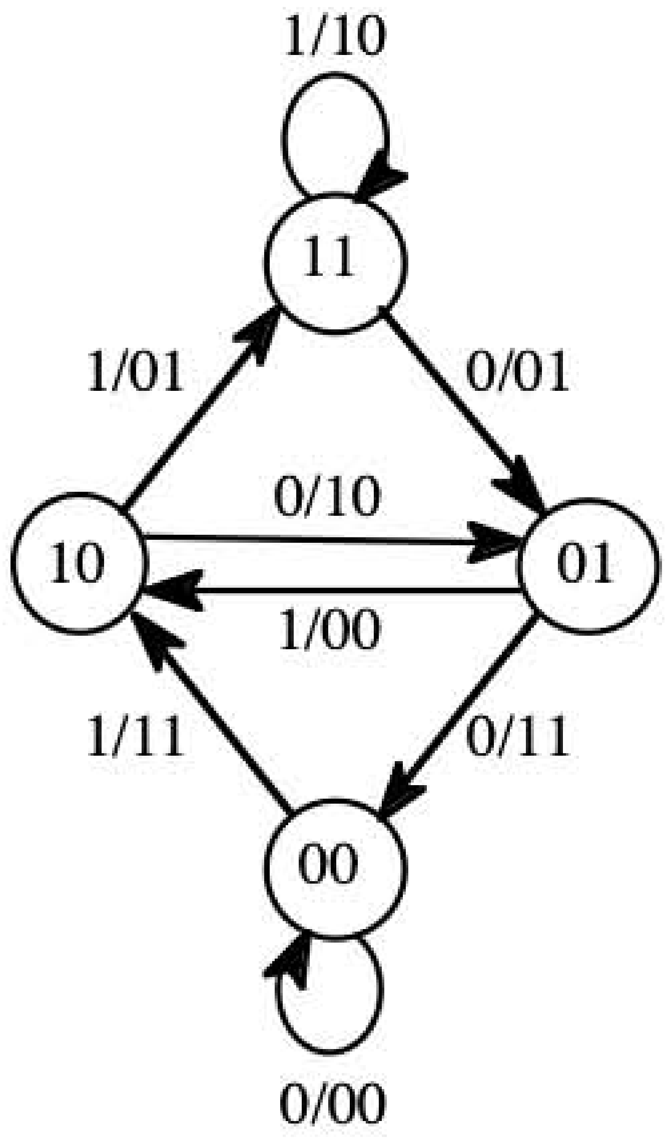




Viterbi decoding

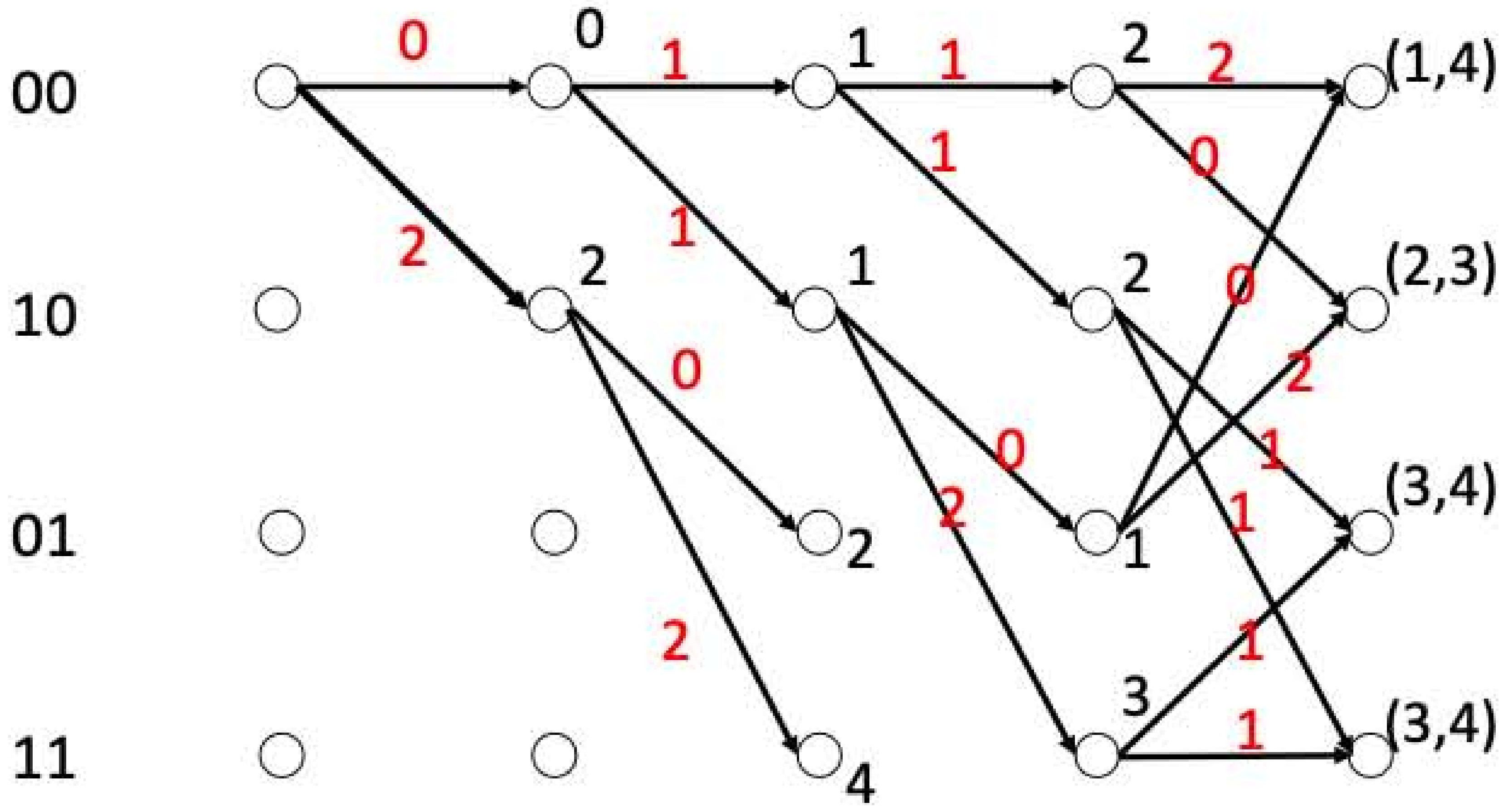
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

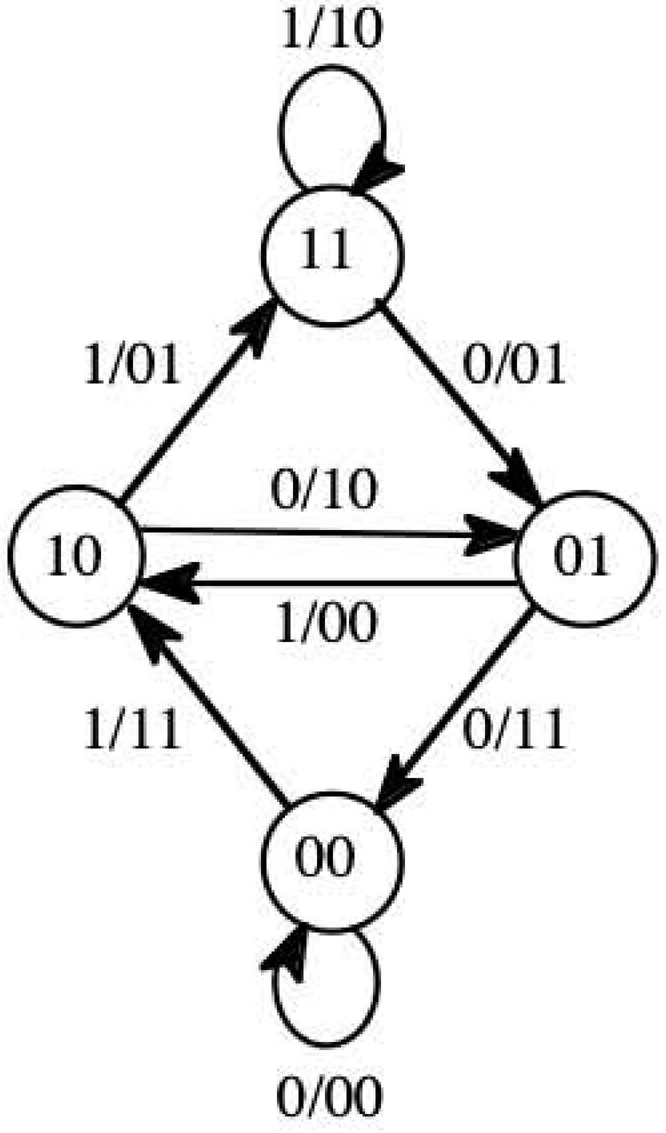




Viterbi decoding

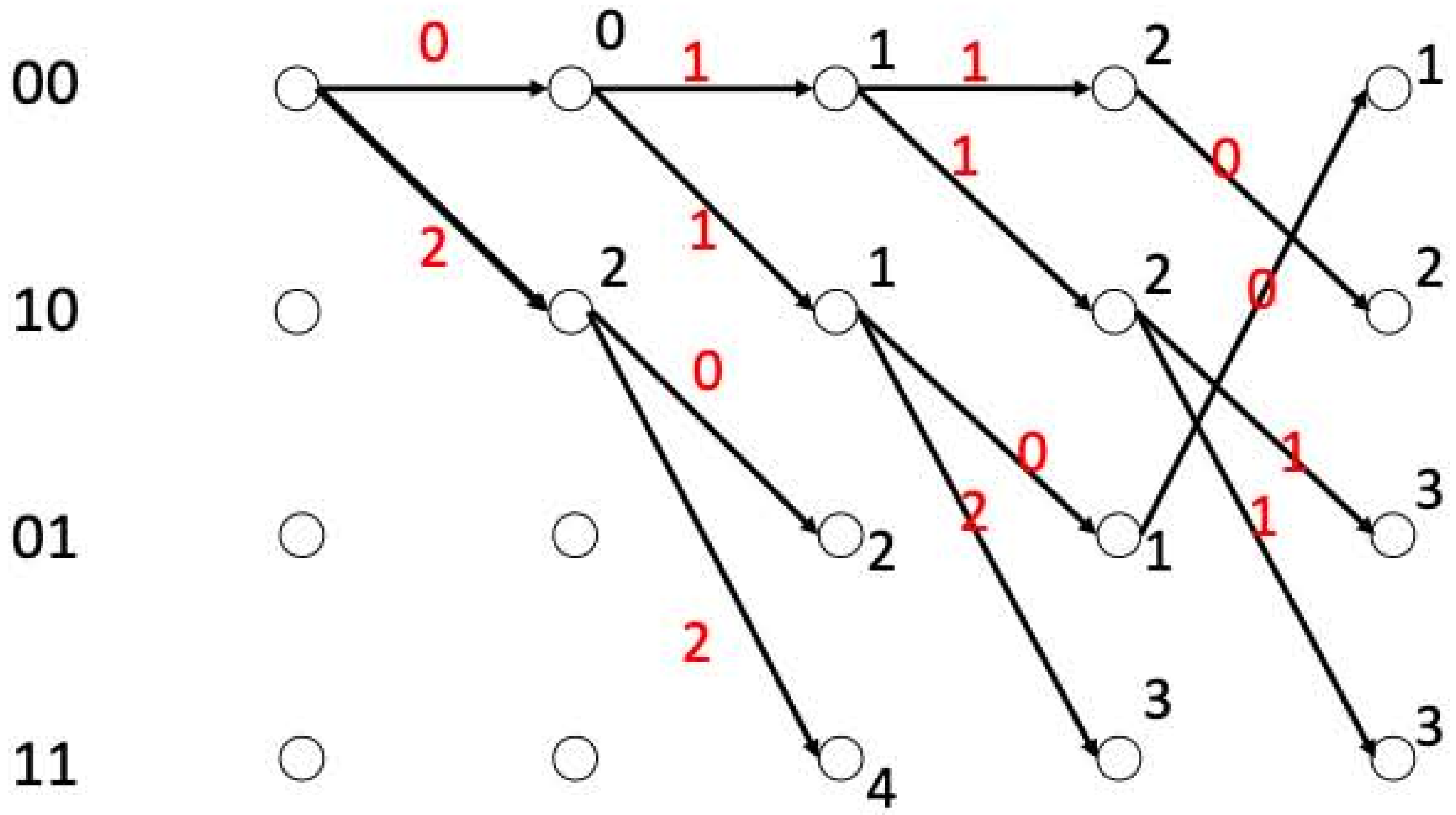
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

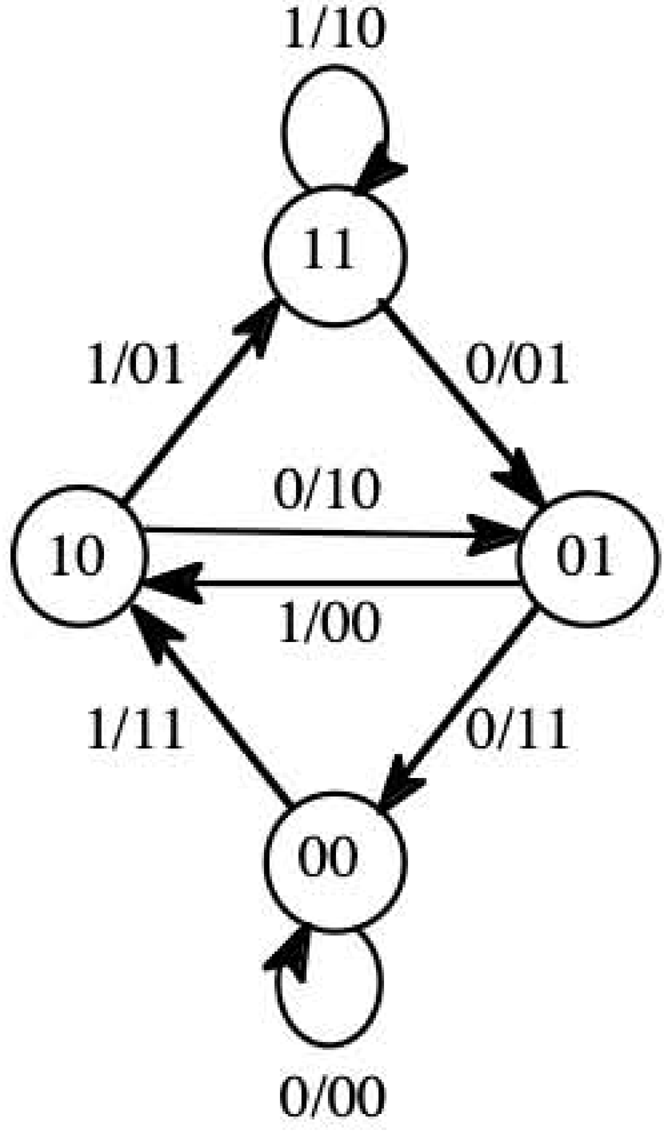




Viterbi decoding

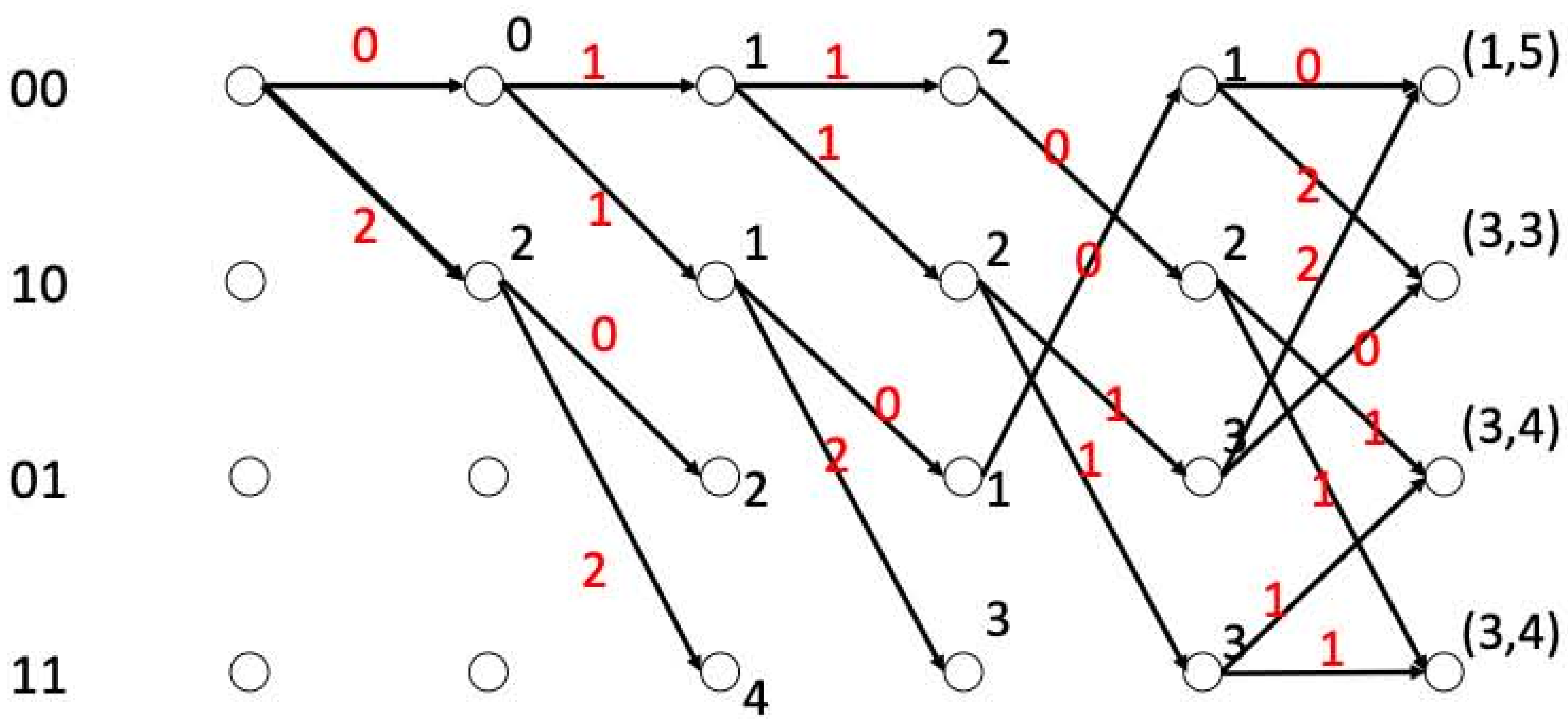
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

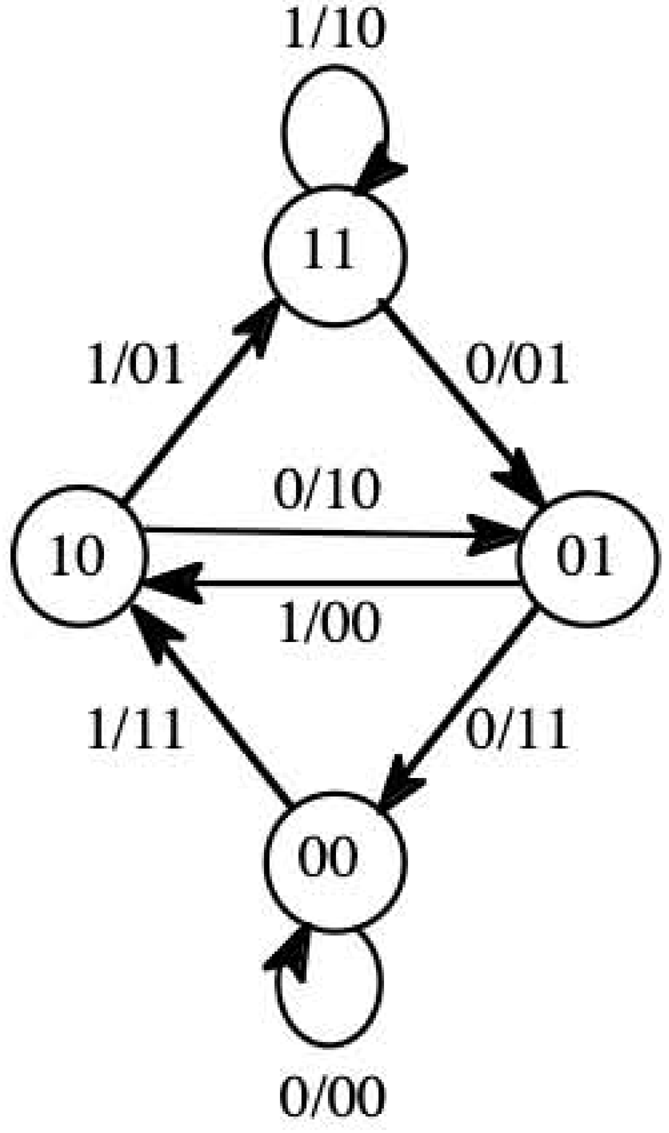




Viterbi decoding

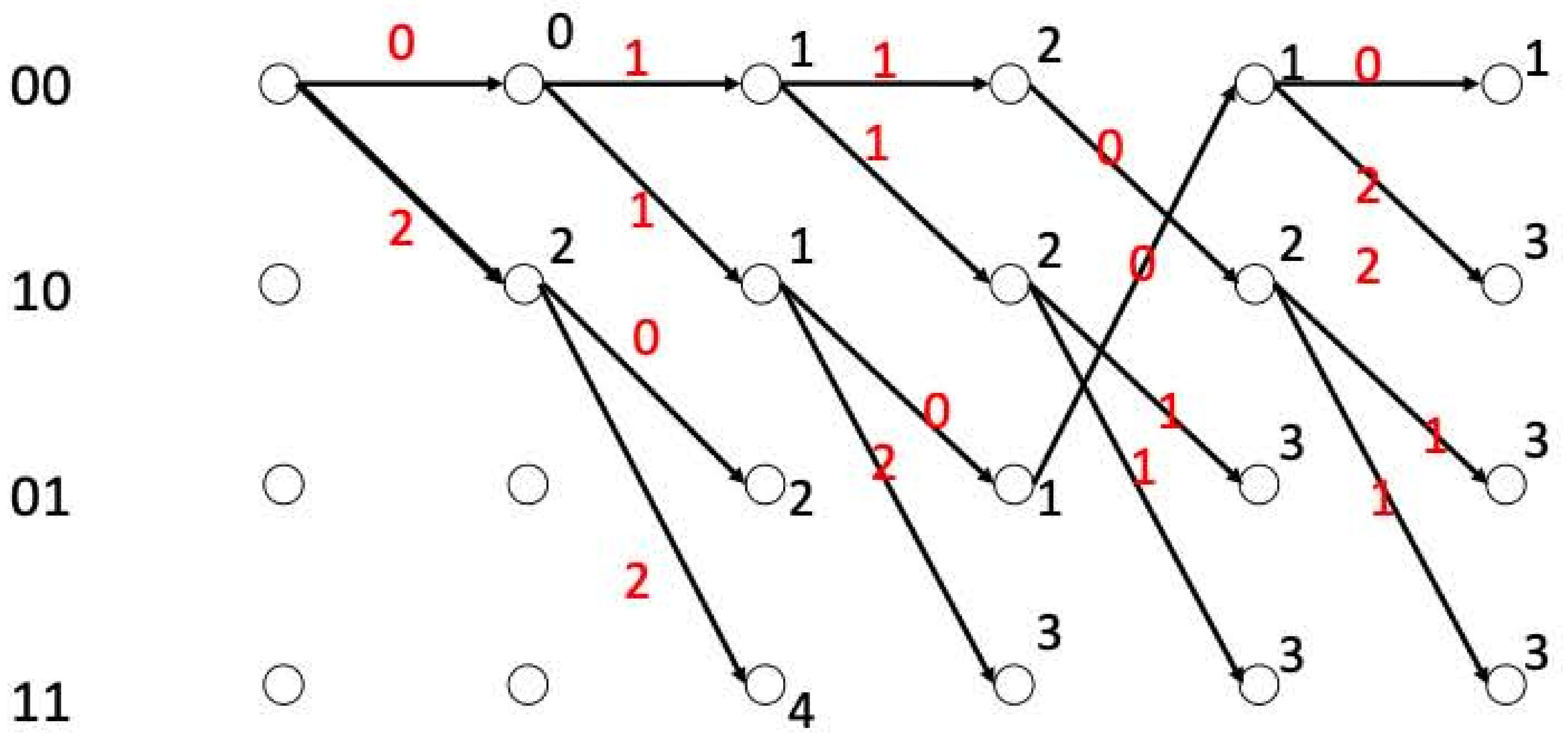
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

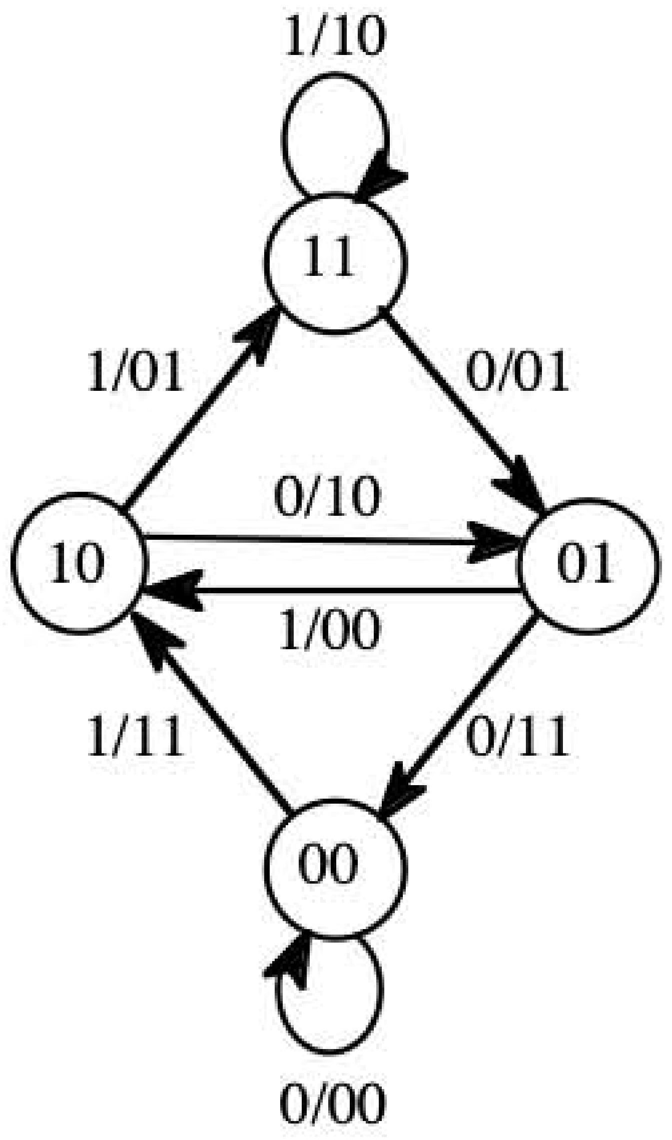




Viterbi decoding

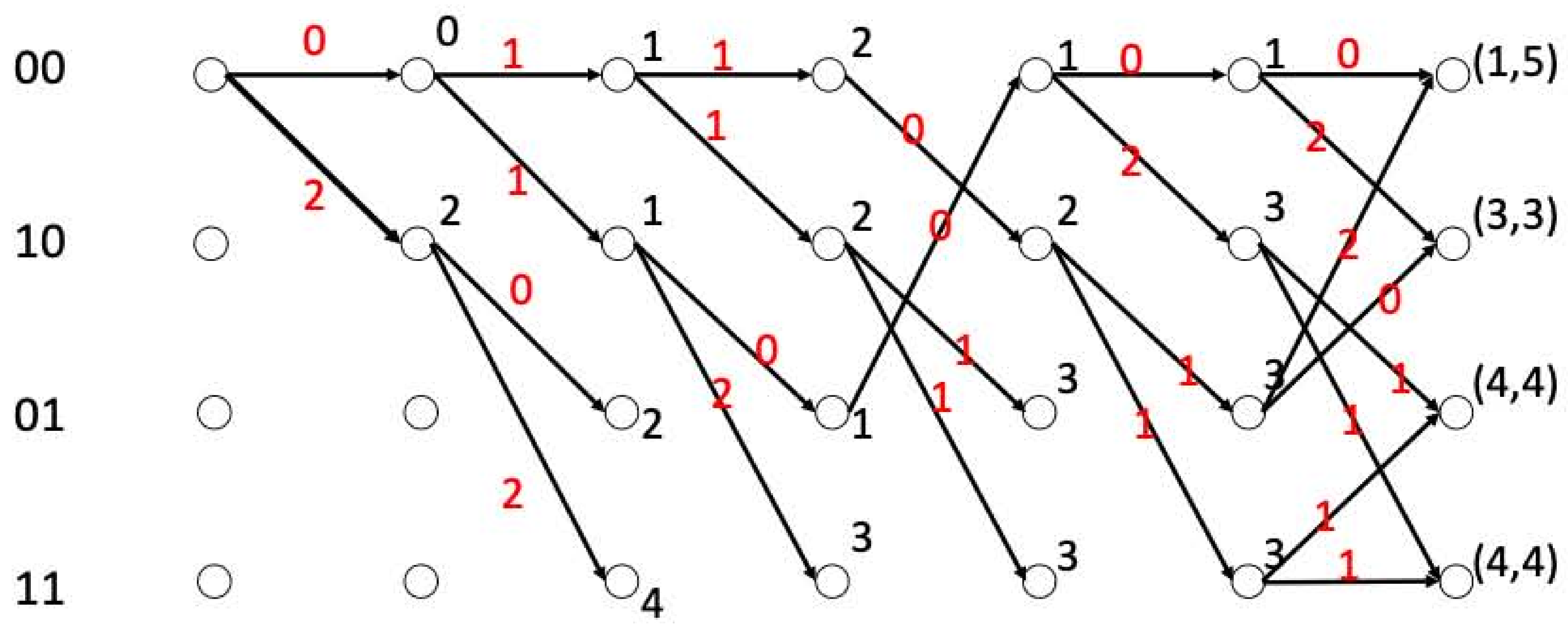
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

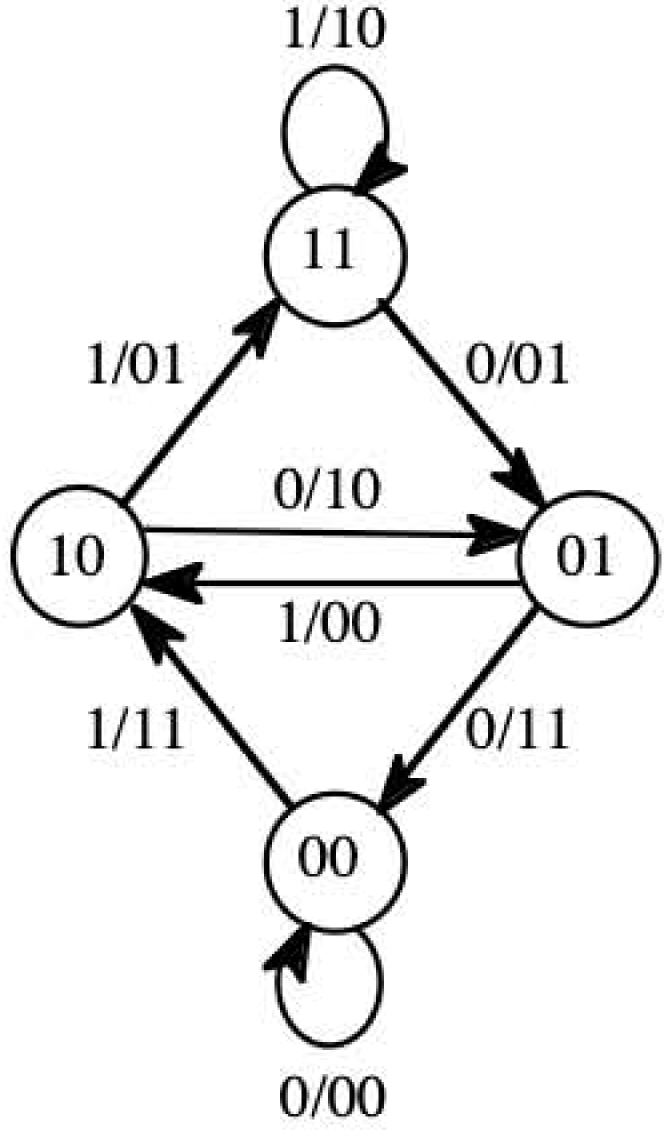




Viterbi decoding

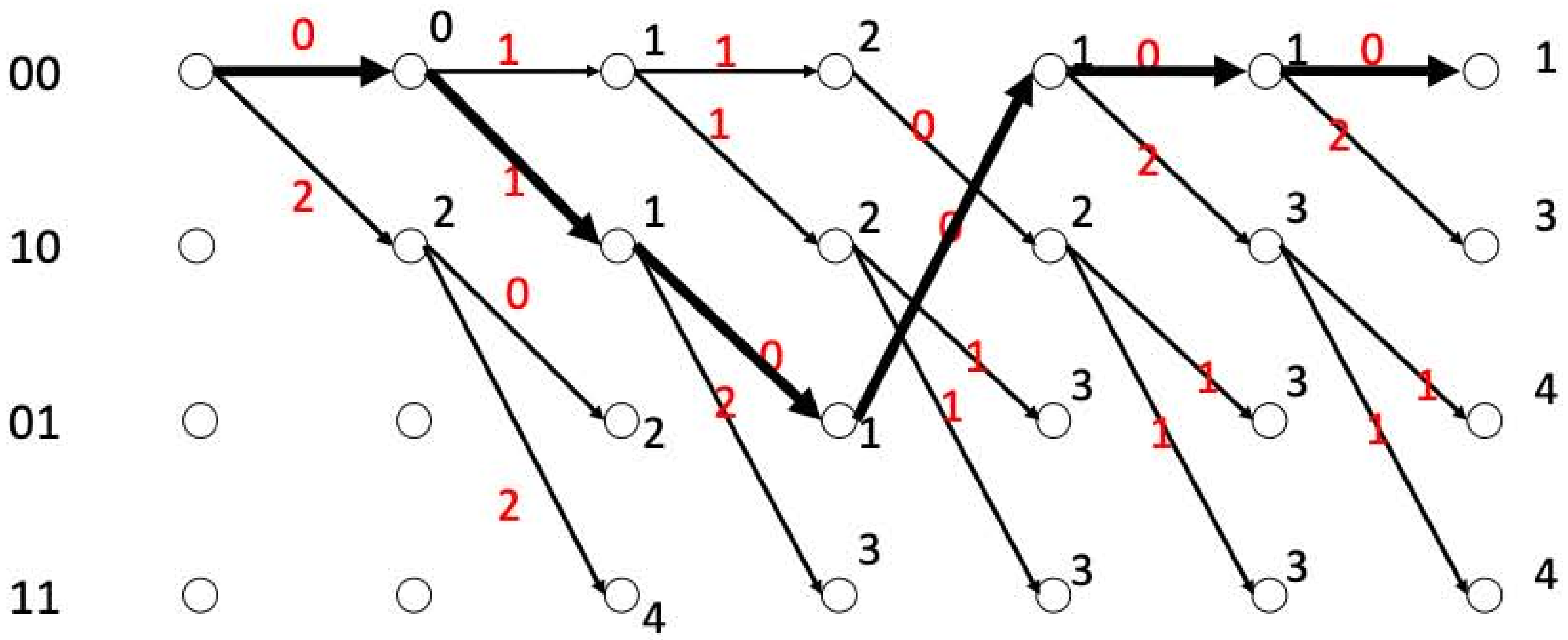
Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00





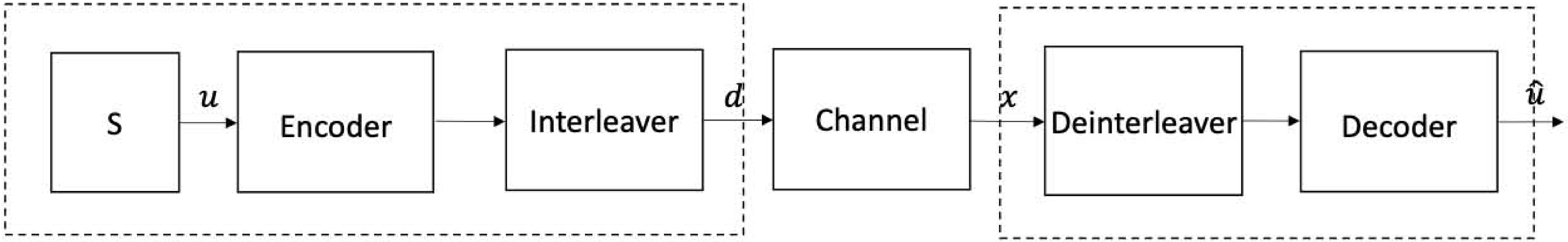
Viterbi decoding

Received sequence	00	10	10	11	00	00
Original sequence	00	11	10	11	00	00

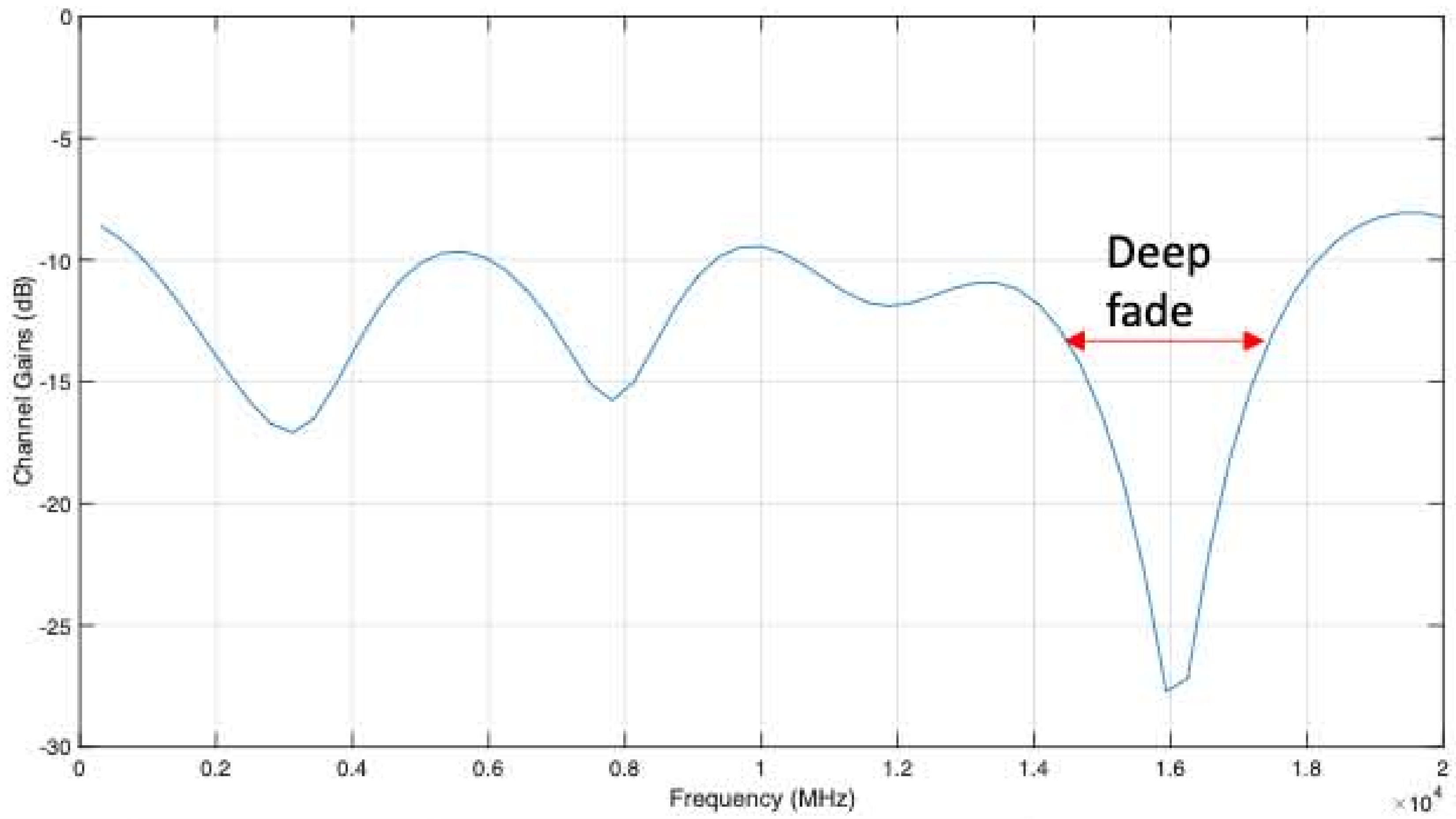


Interleaving

- Convolutional codes are mostly suitable for memoryless channels with random error events.
- Error correcting codes perform well when the errors are uniformly distributed and uncorrelated.
- Fading channels tend to cause *bursty* errors: when a channel is in a deep fade, there is a statistical dependence among successive error events.
- *Interleaving* makes the channel look like as a *memoryless channel* at the decoder and tends to *decorrelate* error events.



Block interleaver example



$\{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$

A	B	C	D
E	F	G	H
I	J	K	L
M	N	0	P

A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

$\{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$ → Interleaver → $\{A, E, I, M, B, F, J, N, C, G, K, O, D, H, L, P\}$

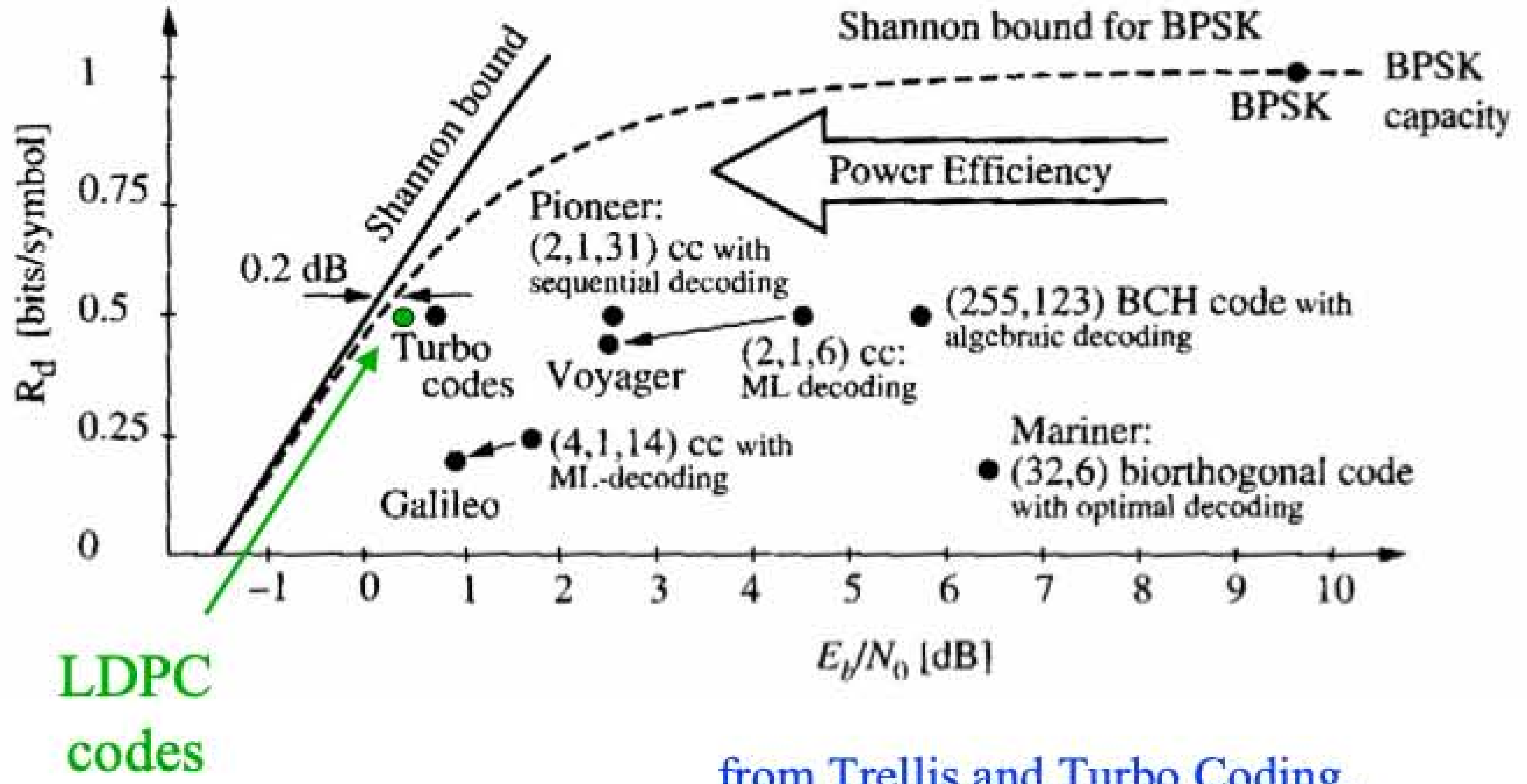
4 positions of distance between two consecutively encoded bits

Interleaving …

- Interleaving is achieved by spreading the coded symbols in time or frequency before transmission.
- The reverse is done at the receiver by deinterleaving the received sequence.
- Interleaving makes bursty errors look like random, so that convolutional codes perform best.
- The price to pay with interleaving is the large *latency*: both at transmitter and at the receiver it is necessary to have the entire block of data to start the encoding/decoding process.
- There is a trade-off: the larger the interleaver depth K, the more decorrelated are the errors but also the longer is the latency and delay.
- Types of interleaving:
  - Block interleaving
  - Convolutional or cross interleaving

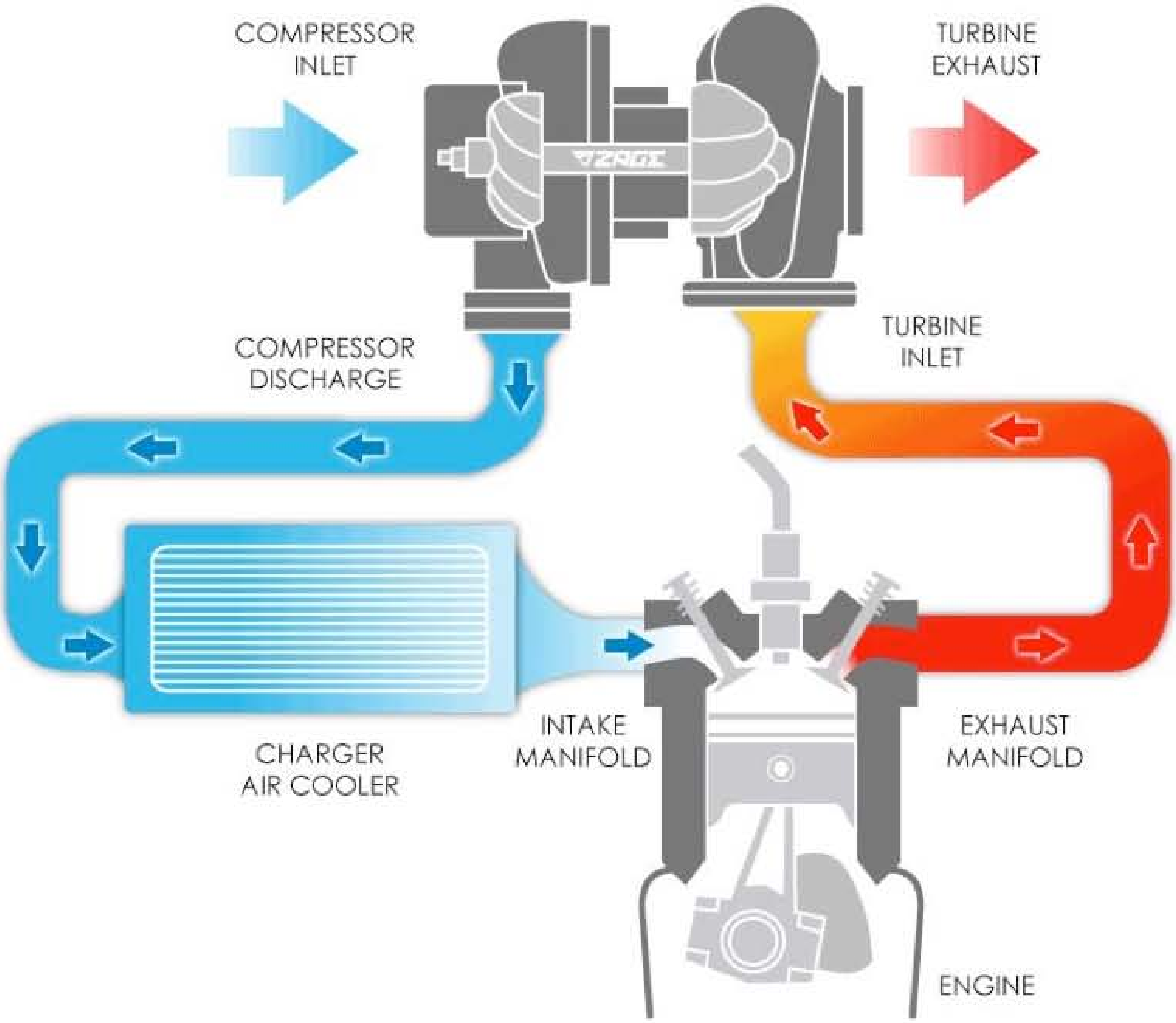
Turbo codes and LDPC

- In the channel capacity theorem, Shannon uses infinitely long codes, i.e. the code rate $R = k/n$ is fixed but $k \to \infty$ and $n \to \infty$.
- In practical systems, the length of the code is limited because of the complexity of decoding: the performance of physical systems are far from Shannon theoretical limits.
- Around the turning of the century there have been two major breakthroughs:
  - Turbo codes (1993)
  - Low-density parity check codes (LDPC, 1999).



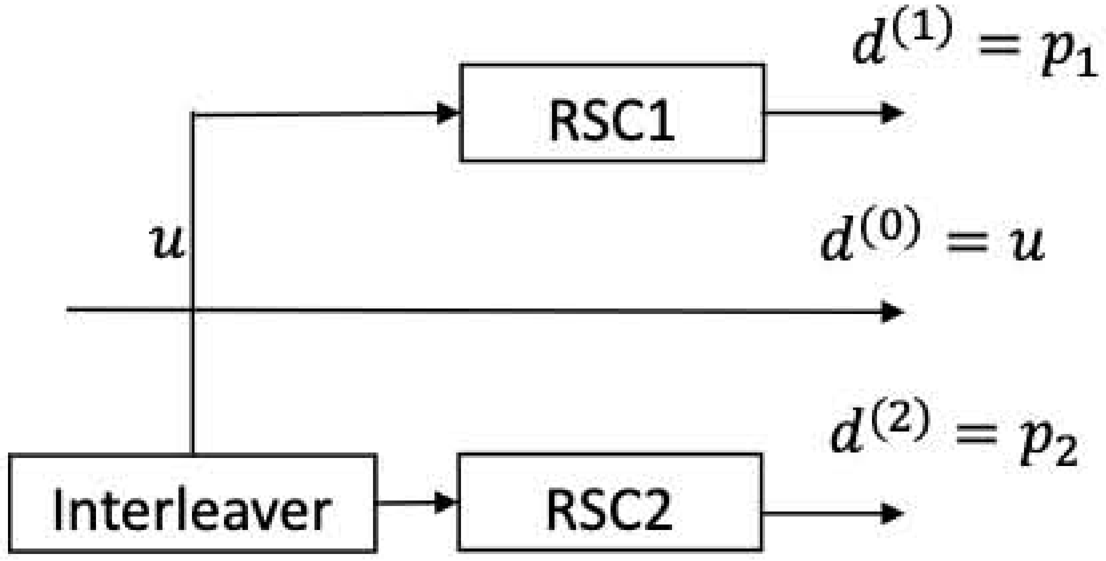
from Trellis and Turbo Coding, Schlegel and Perez, IEEE Press, 2004

Turbo.....

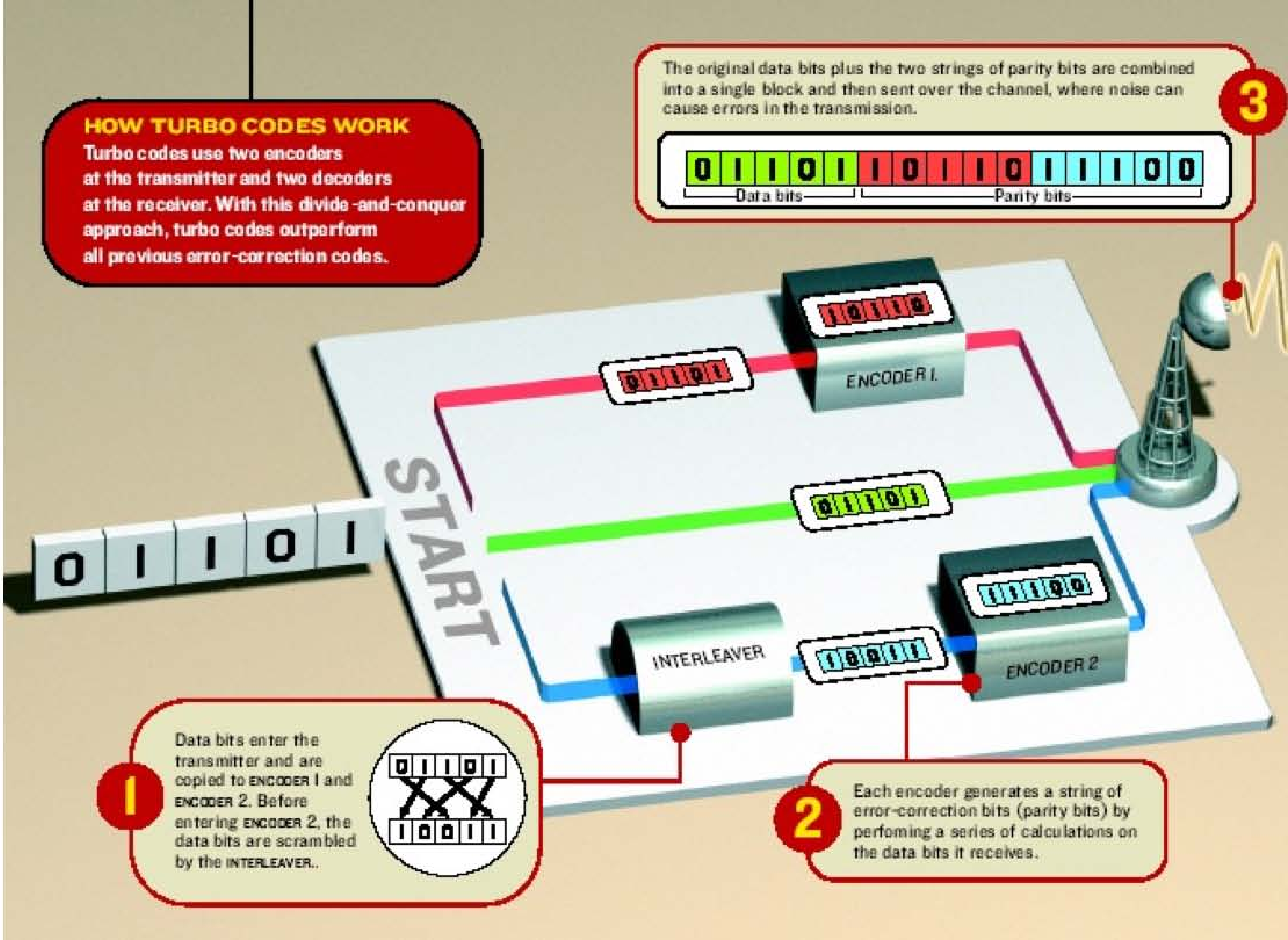

COMPRESSOR
INLET
TURBINE
EXHAUST
COMPRESSOR
DISCHARGE
TURBINE
INLET
INTAKE
MANIFOLD
EXHAUST
MANIFOLD
CHARGER
AIR COOLER
ENGINE

Turbo codes

- The combination of coding and interleaving can be exploited to boost the performance of convolutional codes in general.
- *Turbo codes* are a particular example of concatenated codes, where two encoders are in parallel and the input data of the second encoder are first interleaved.
- Thanks to the interleaver, the decoder will have two independent replicas of the same data and can use both streams to decode the information sequence.
- Increase the redundancy of the transmitted information but also the *diversity* of the system.

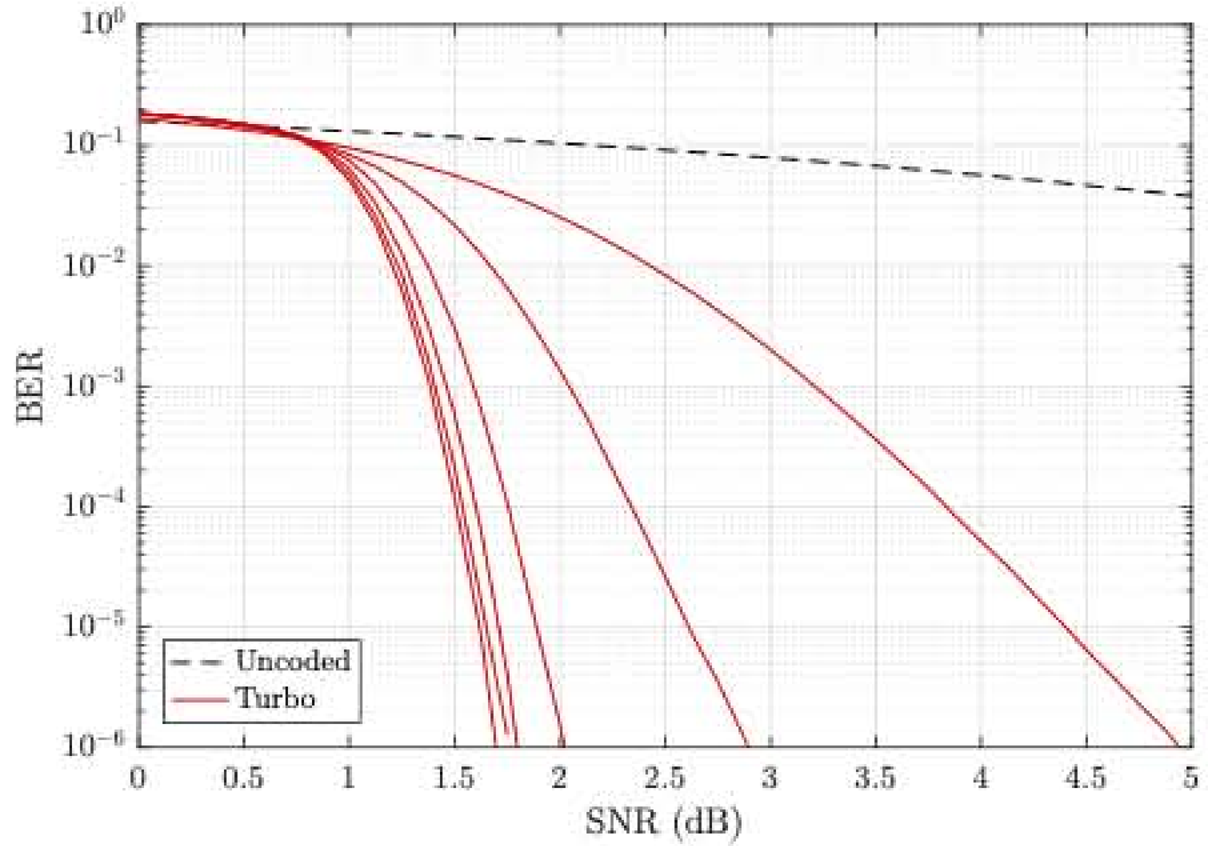


Turbo codes (© IEEE spectrum)

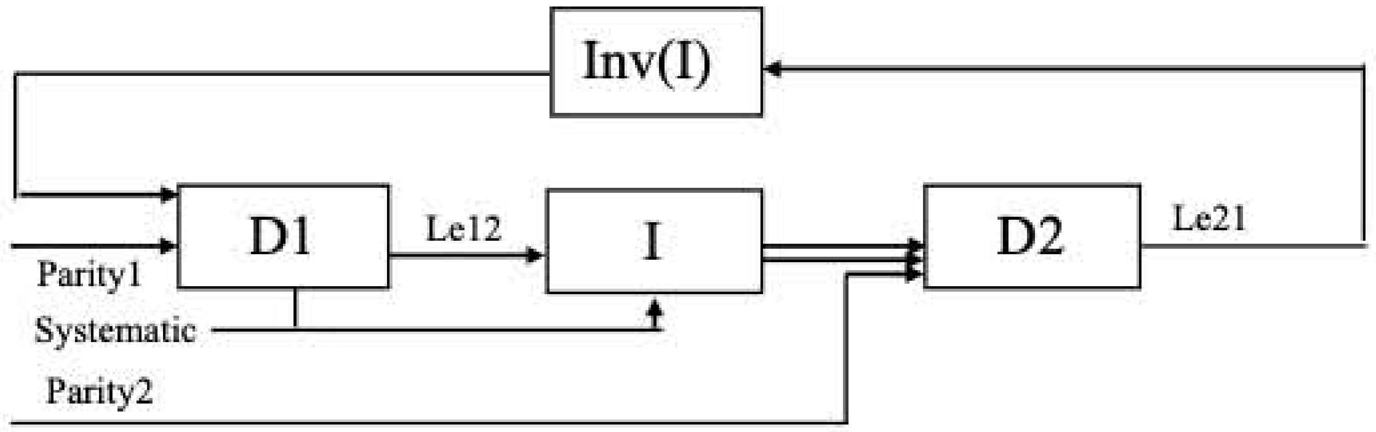


Turbo decoding

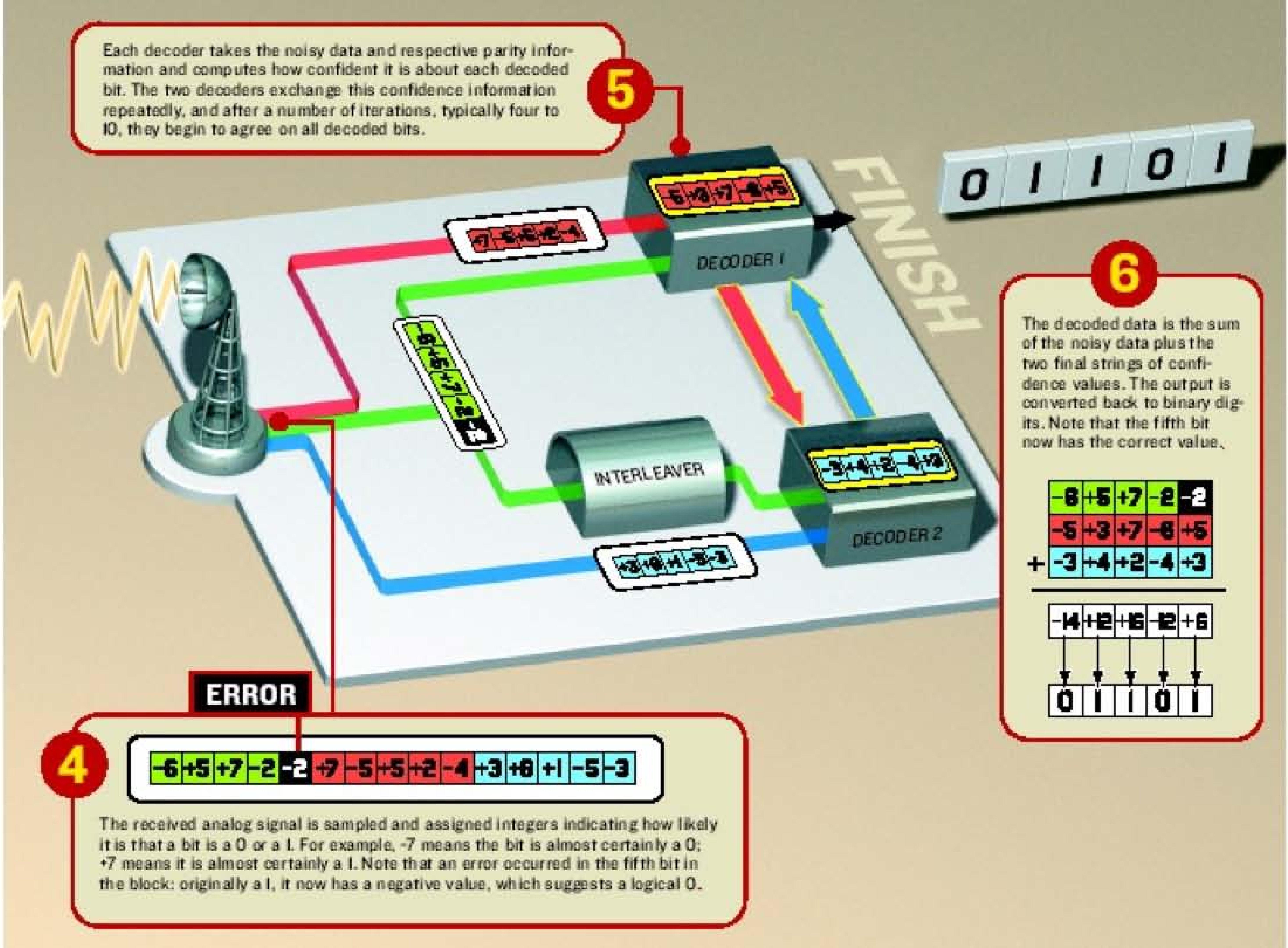
- The turbo decoder employs the cascade of two decoders, with the output of one decoder used as prior information to the next.
- Feedback in decoding circuit allows for multiple iterations and improves bit error performance.
- The process is iterative and at each step the combination of the two decoders corrects some of the errors.



Turbo code convergence, K = 2048, R = 1/2, Iterations = 32, 16, 8, 4, 2, 1 from left to right.



Turbo codes (© IEEE spectrum)



Turbo codes: latency

- Convolutional codes in general and turbo codes in particular have a problem with *latency* due to the presence of the interleaver and the iterative decoding process.
- At any given SNR there is a tradeoff between latency due to interleaver and QOS
    - Small block sizes ($K \sim 300$ bits) can be used for real time voice (medium-high BER can be tolerated).
    - Mid range block sizes ($K \sim 4000$ bits) used for video play back (low BER).
    - Large block sizes ($K \sim 16000$ bits), useful for file transfer (very low BER).

Spatial diversity

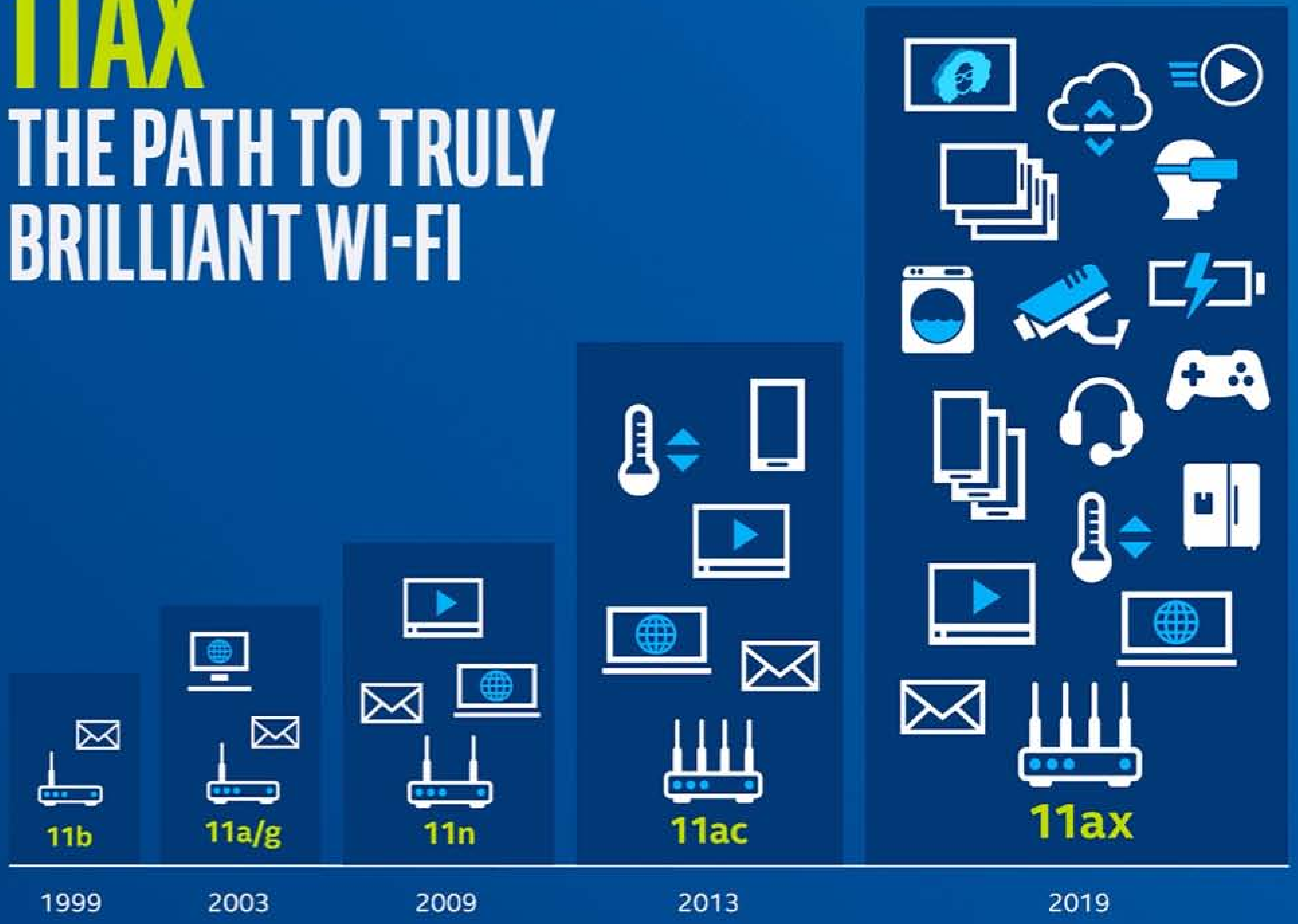
Receive diversity

- Among the many ways to obtain diversity:
  - *Frequency* and *time* diversity require expensive resources (bandwidth or time) and do not provide array gain.
  - *Space* diversity by means of multiple antennas does not sacrifice bandwidth or time and may provide array gain.
- *Array gain*: is the power *gain* achieved by using multiple antennas with respect to the single antenna case. The more correlated is the spatial channel the higher the potential array gain.
- *Diversity gain*: is the power gain due to the exploitation of the diversity of the spatial channel. It is maximum when the spatial channel is uncorrelated.

Antennas and carrier frequency

- In order to assume that the channel is uncorrelated the distance between antenna elements is
$$d_c = \lambda/2$$
- New telecom systems use increasingly large carrier frequencies and as a consequence shorter wavelengths.
- Wi-Fi, originally designed @2.4 GHz ($\lambda = 12.5$ cm) added a new band around 5GHz ($\lambda = 6$ cm).
- 5G is now working @3.8 GHz ($\lambda \approx 8$ cm) and it is already standardized for frequencies up to 52 Ghz ($\lambda \approx 6$ mm)!
    - The number of antennas at the transmitter and at the receiver grows of a factor up to $100\times$

11AX
THE PATH TO TRULY BRILLIANT WI-FI



4x **BETTER IN DENSE ENVIRONMENTS**

Improve average throughput per user by at least four times in dense or congested environments

FASTER THROUGHPUT

Deliver up to 40 percent higher peak data rates for a single client device

INCREASE NETWORK EFFICIENCY

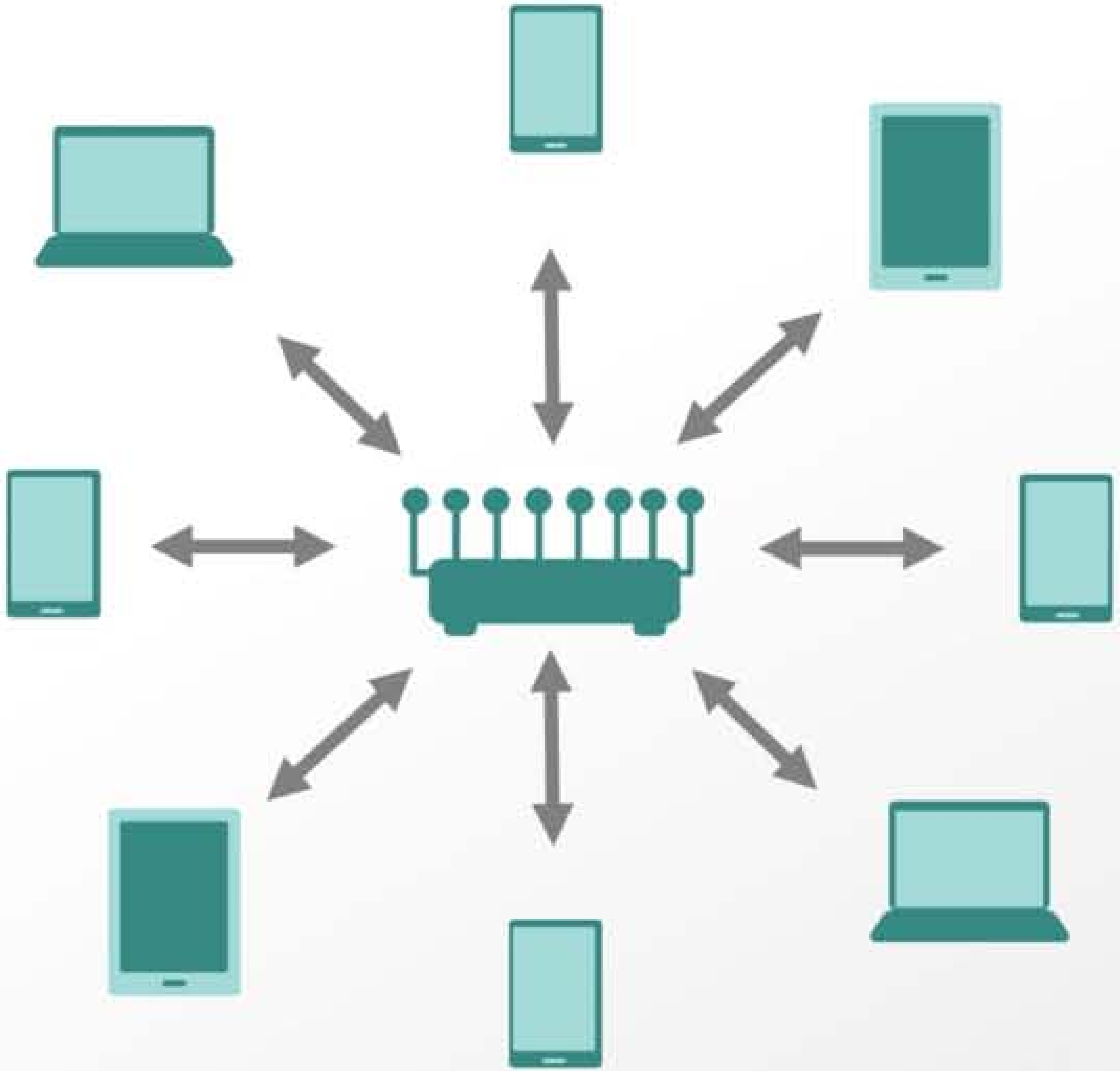
By more than four times

EXTEND BATTERY LIFE

Of client devices

Extending the benefits of proven 11ac MU-MIMO

11ac MU-MIMO is already mainstream

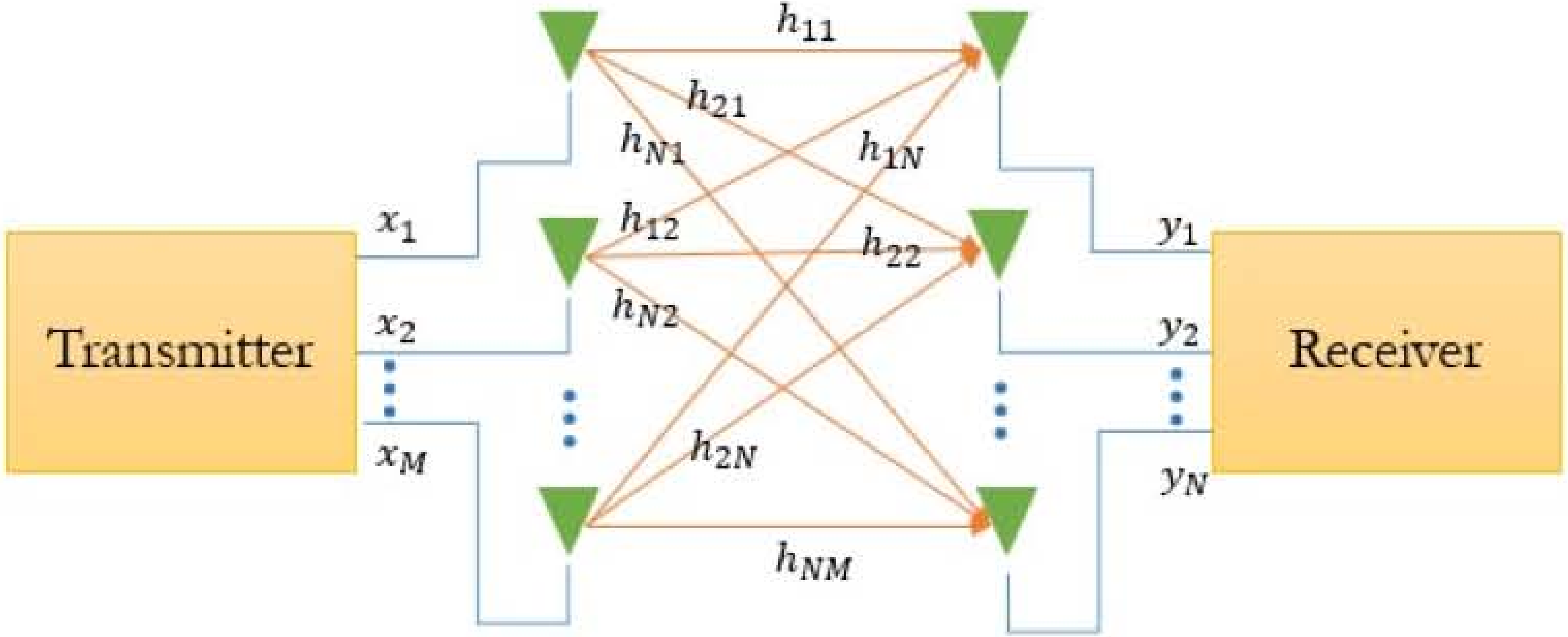


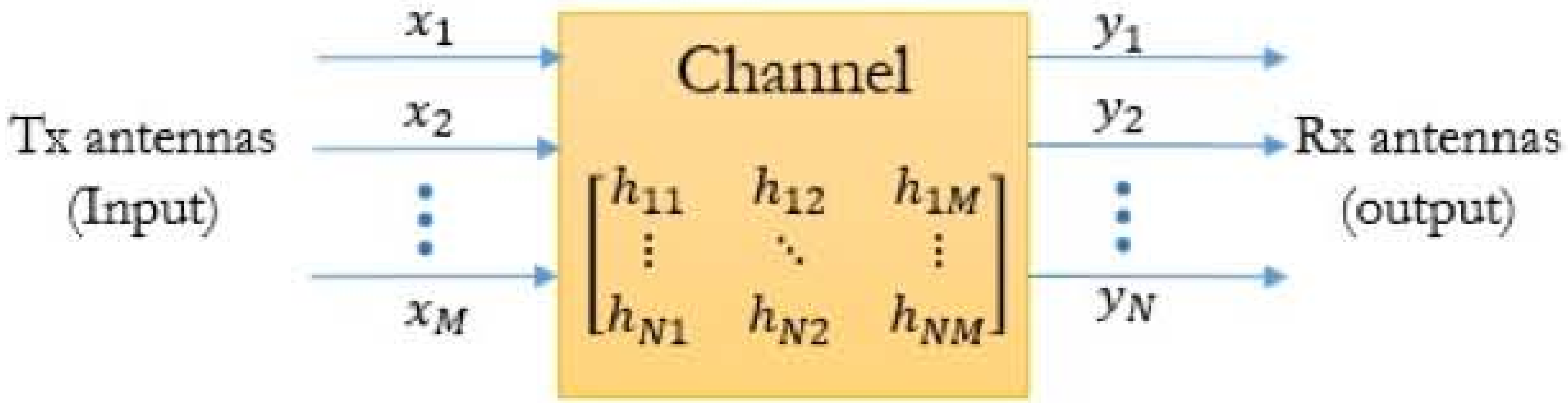
- Up to 8x8 MU-MIMO in the downlink
  - Serving up to 8 simultaneous users (in downlink)
  - Up to 2x increase in capacity vs. 4x4
- Up to 8x8 MU-MIMO in the uplink
  - Serving up to 8 simultaneous users (in uplink)
  - Up to 8x increase in capacity vs. 1x1
  - Extremely useful for uplink heavy apps such as social media, content sharing (video, picture uploads, Periscope, etc)
- Higher MU-MIMO gain with more client devices per AP

MIMO channel

- Multiple-input multiple-output systems are systems where where both the transmitter and the receiver are equipped with several antennas.
- *Narrowband* assumption: the channel linking the n-th receive antenna with the m-th transmit antenna is the complex scalar $h_{n,m}$

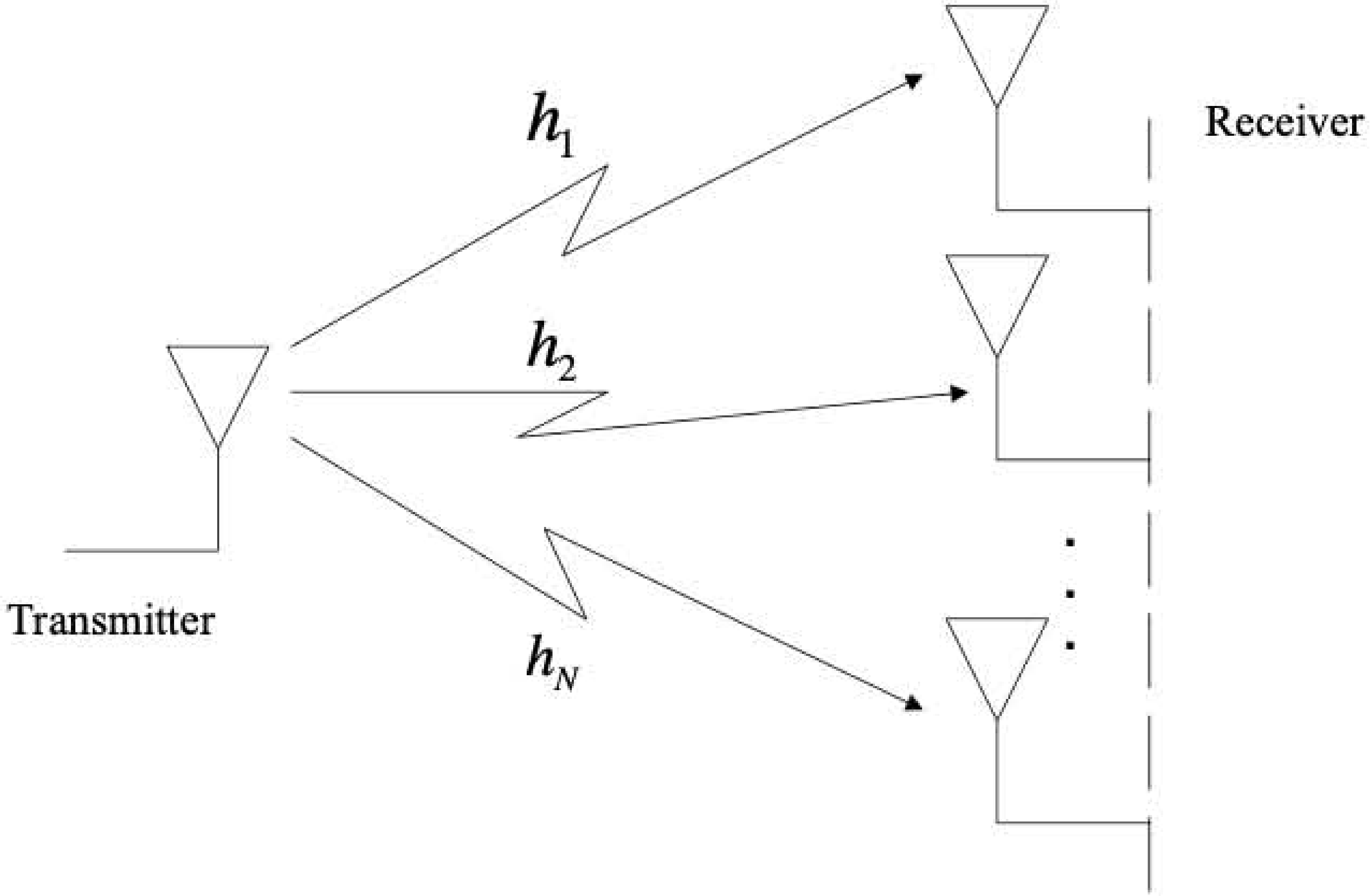
Multiple Input Multiple Output (MIMO) System





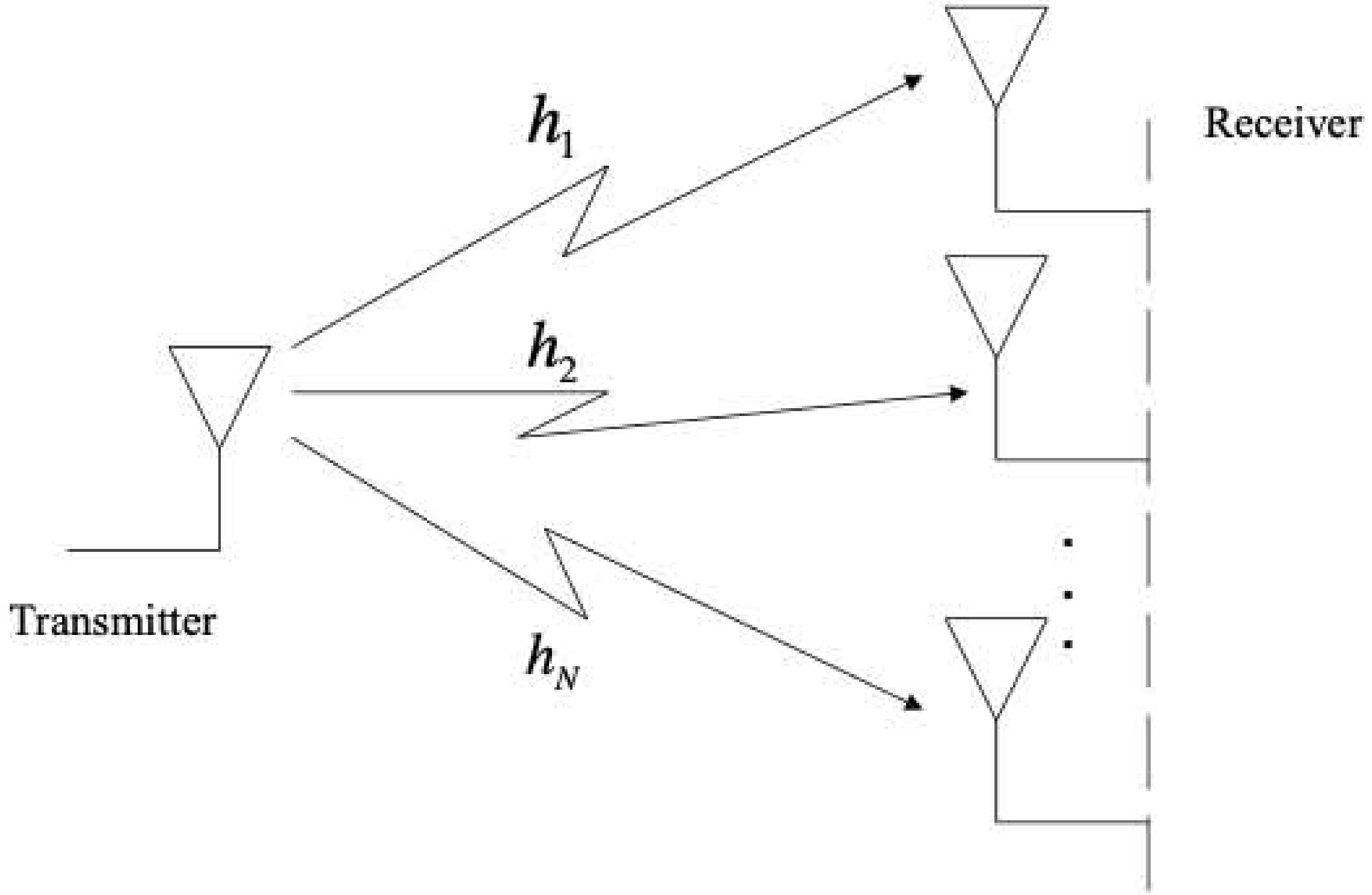
MIMO from channel perspective

SIMO channel: receive diversity



- $N > 1$ antennas at the receiver, $M = 1$ at the transmitter.
- The decision variable at the i-th receive antenna is
$$x_i(m) = h_i c_m + n_i(m)$$
- The signals received at the N antennas are combined together and the decision variable is
$$z(m) = w_1 x_1(m) + \cdots + w_N x_N(m)$$

SIMO channel: receive diversity



- The combining coefficients are chosen to maximize the SNR. Since it is

$$z(m) = \sum_{i=1}^{N} w_i\, h_i c_m + \sum_{i=1}^{N} w_i\, n_i(n)$$

The signal part is given by

$$\mathrm{E}\left\{\left|\sum_{i=1}^{N} w_i\, h_i c_m\right|^2\right\} = \left(\sum_{i=1}^{N} w_i\, h_i\right)^2 A$$

The noise part is

$$\mathrm{E}\left\{\left|\sum_{i=1}^{N} w_i\, n_i(m)\right|^2\right\} = \left(\sum_{i=1}^{N} w_i\right)^2 \sigma^2$$

The signal to noise is

$$\mathrm{SNR} = \frac{\left(\sum_{i=1}^{N} w_i\, h_i\right)^2}{\left(\sum_{i=1}^{N} w_i\right)^2}\,\frac{A}{\sigma^2}$$

Maximal ratio combining (MRC)

- By virtue of the Schwarz inequality, the ratio $\left(\sum_{i=1}^{N} w_i\, h_i\right)^2$ is upper bounded by

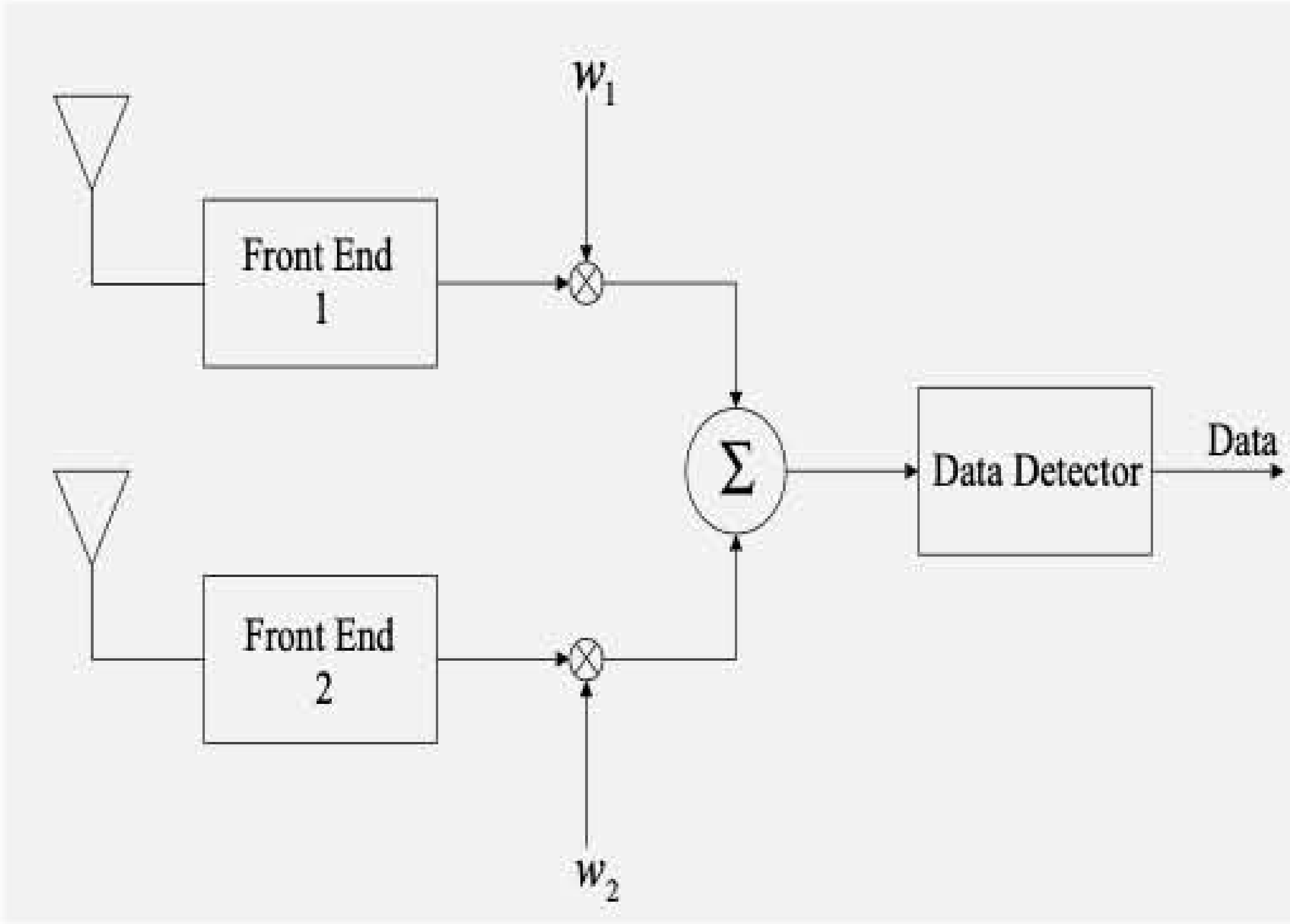
$$\left(\sum_{i=1}^{N} w_i\, h_i\right)^2 \leq \sum_{i=1}^{N} w_i^2 \sum_{i=1}^{N} h_i^2$$

- The equivalence is obtained only when $w_i = h_i$ for all i. In this case the optimal coefficients are

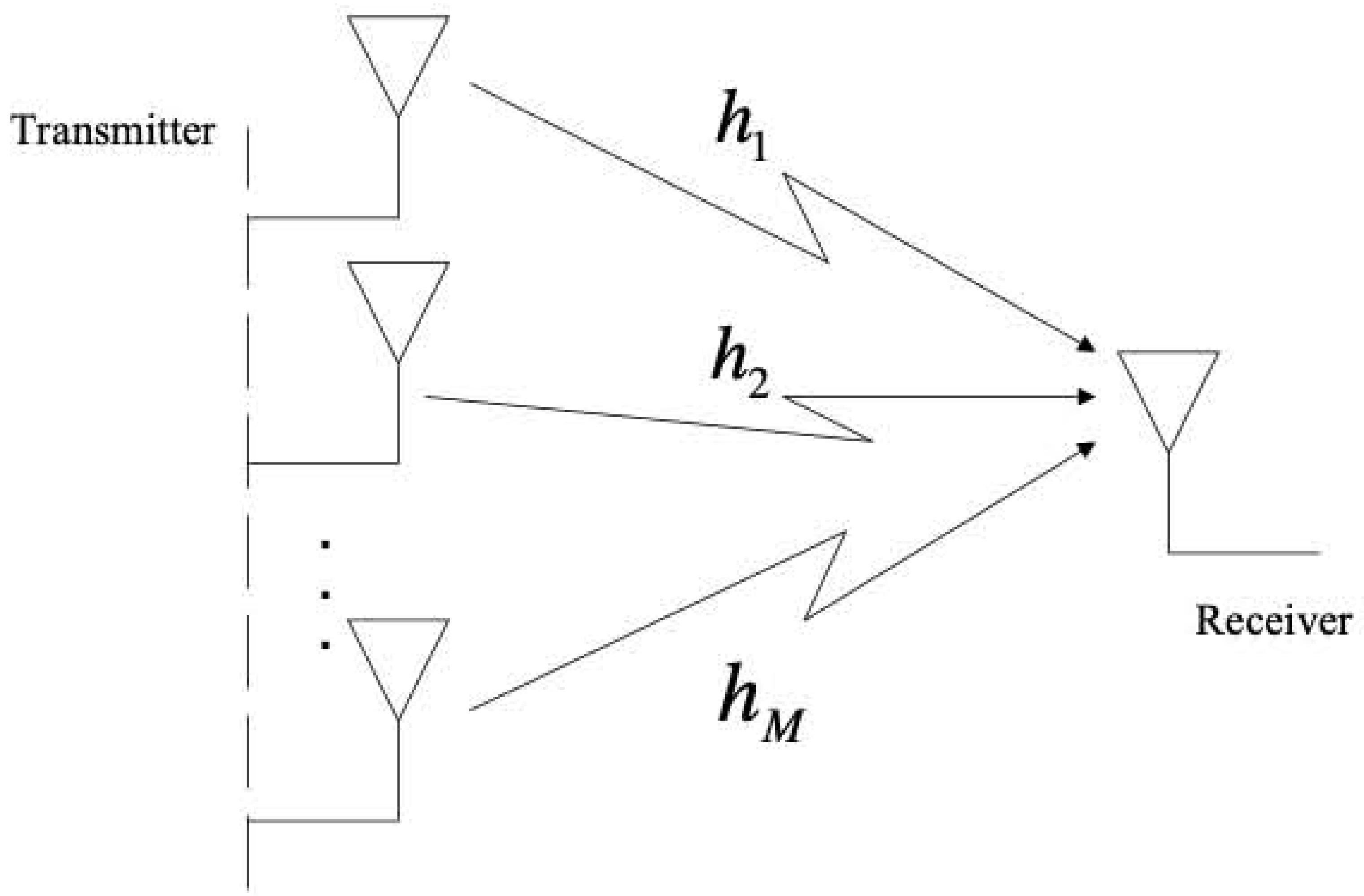
$$w_i = h_i$$

- The signal-to-noise ratio is

$$SNR = \frac{A}{\sigma^2} \sum_{i=1}^{N} h_i^2$$



MISO channel: transmit diversity



- $M > 1$ antennas at the transmitter and $N = 1$ at the receiver
- Spatial pre-coding: the signals are precoded at the transmitter.
- The signal at the j-th transmit antenna is $y_j(m) = b_j c_m$
- The transmitted energy depends also on the precoding weights.
- The received signal is

$$x(m) = h_1 y_1 + \cdots + h_M y_M$$

Maximal ratio transmit (MRT) combining

- The optimal precoding weight for the j-th transmit antenna is

$$b_j = h_j^* / \|h\|$$

so that the overall transmitted energy does not change.

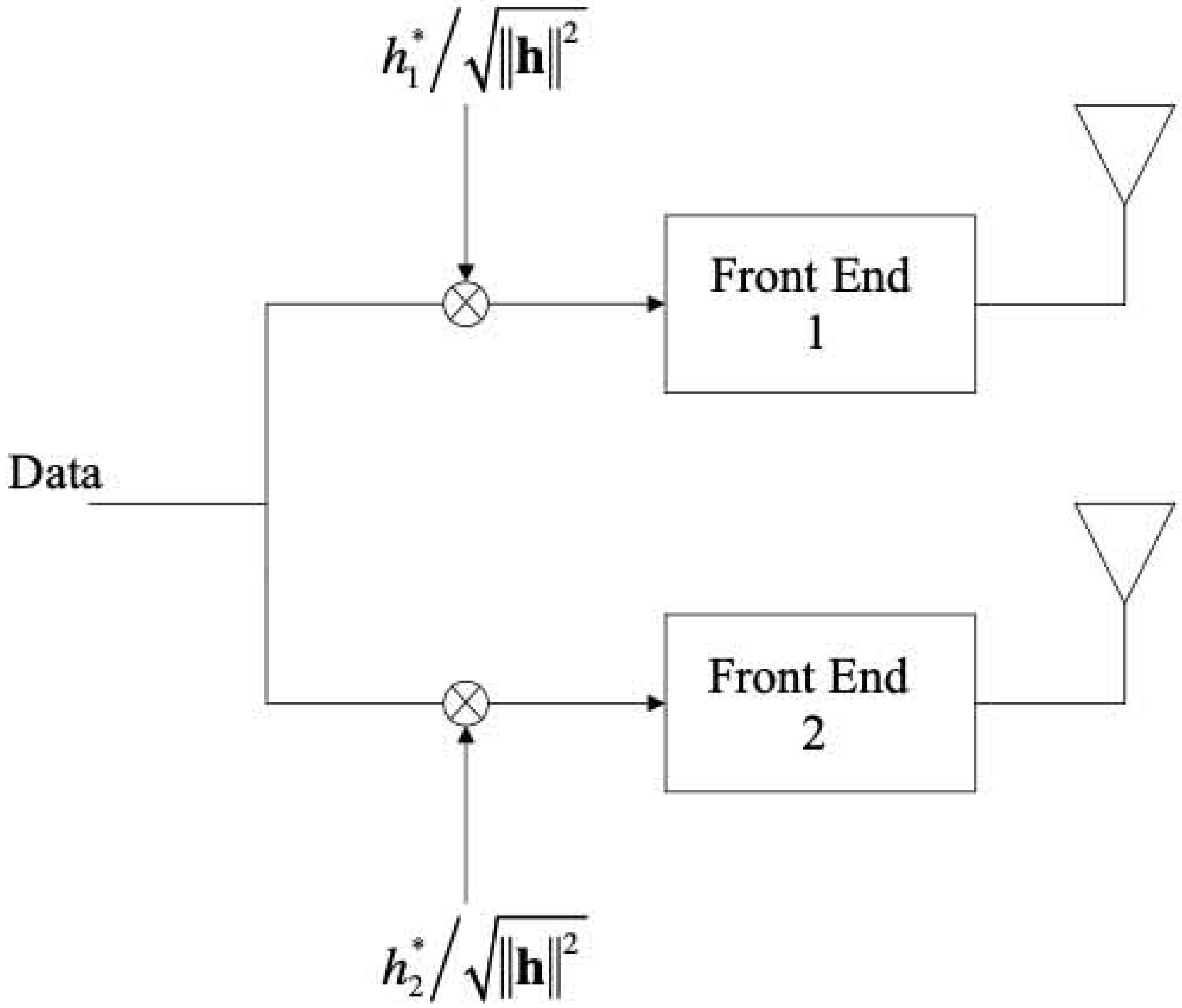
- At the receiver

$$x(m) = \sum_{j=1}^{N} h_j b_j\, c_m + n(m) = \sum_{j=1}^{N} \frac{h_j h_j^*}{|h_j|} c_m + n(m)$$

$$= \sum_{j=1}^{N} |h_j|\, c_m + n(m)$$

- The signal to noise is

$$SNR = \frac{A}{\sigma^2} \sum_{i=1}^{N} |h_j|^2$$

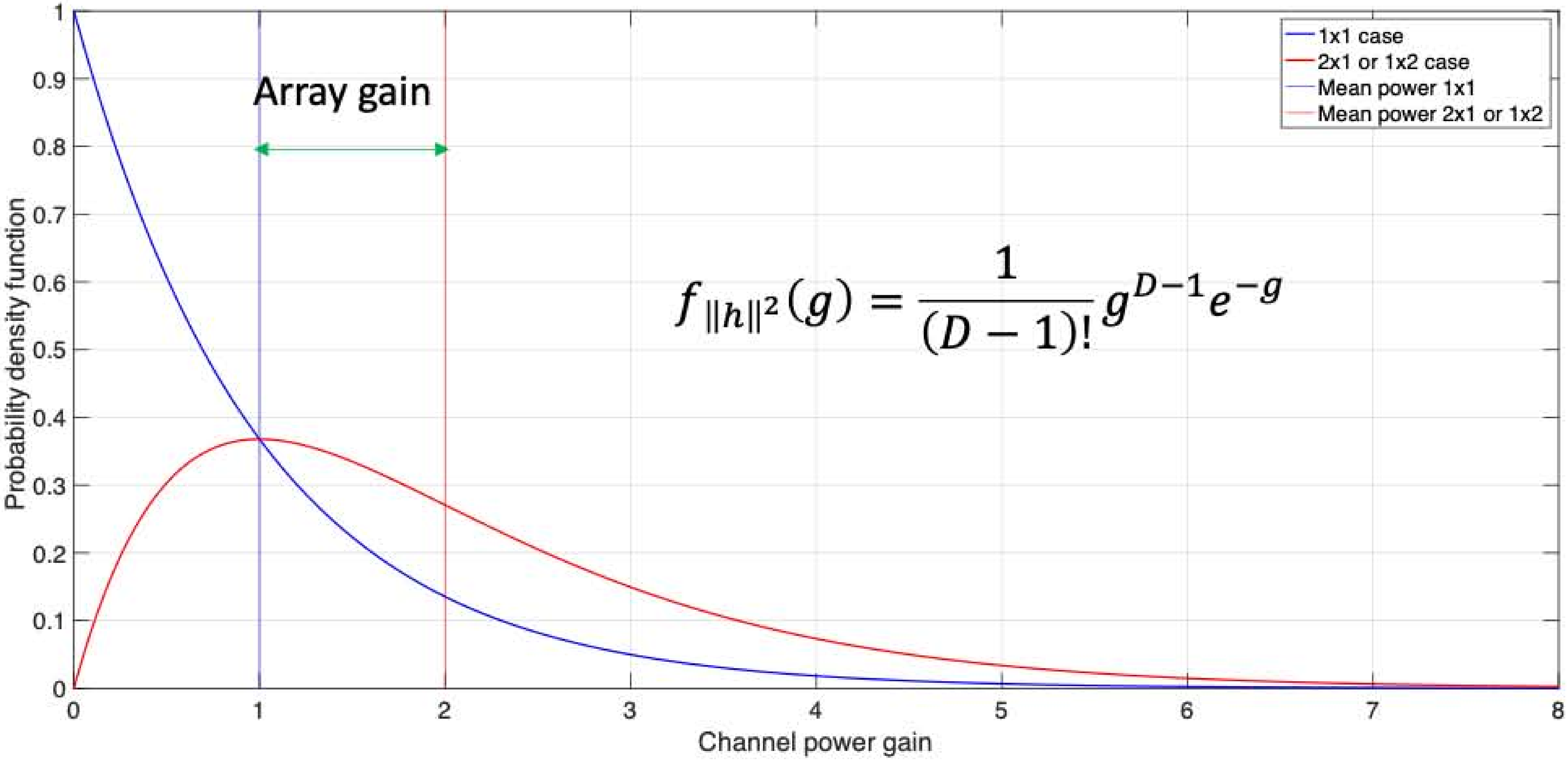


Channel gain $||h||^2$ distribution for D antennas

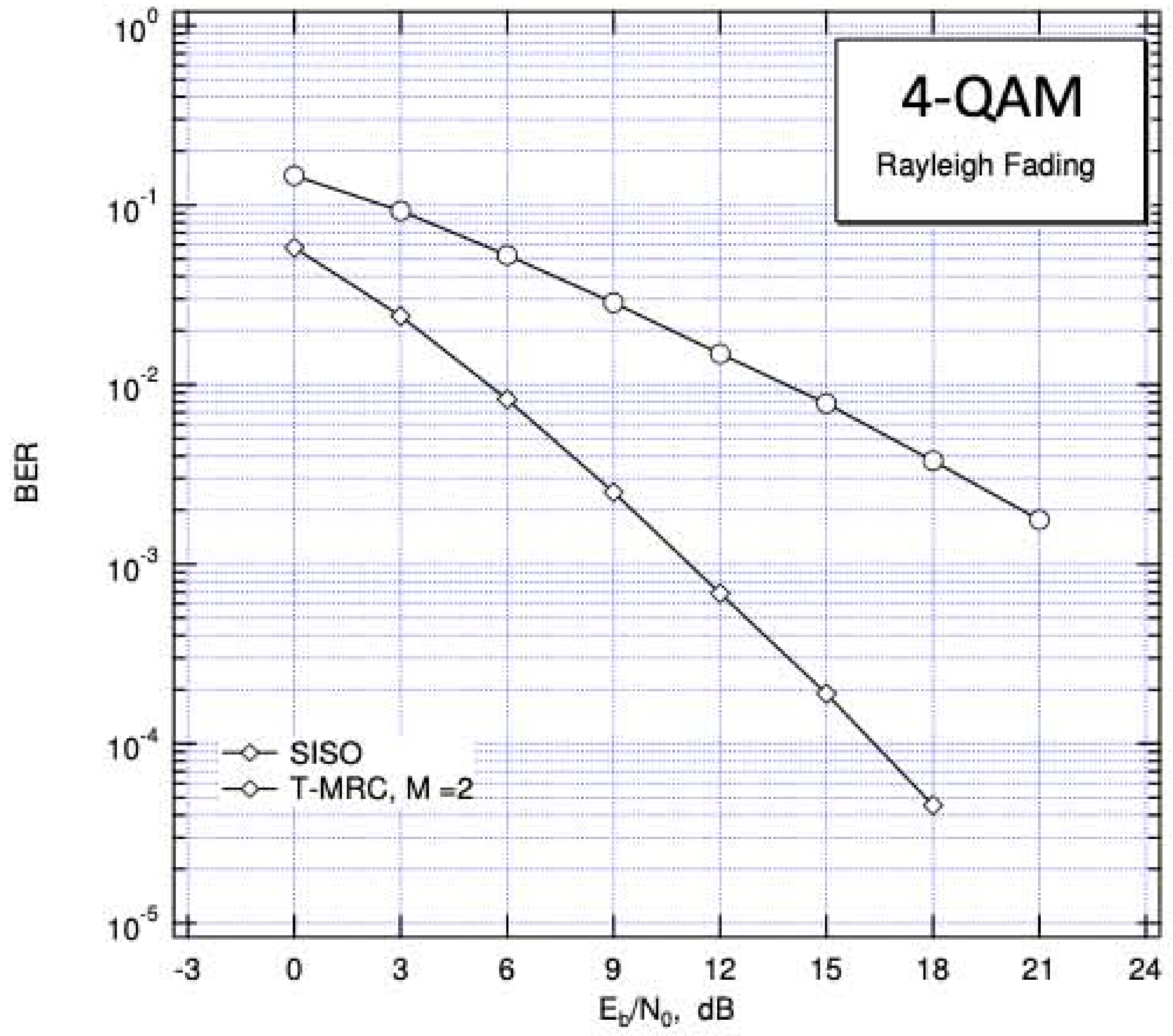
- With MRT or MRC, the SNR is proportional to the channel power gain

$$||h||^2 = \sum_{i=1}^{D} |h_j|^2$$

- The pdf of $||h||^2$ depends on D, the number of antennas!



Maximal ratio combining



- Pros
  - Diversity gain
  - Array gain
  - MRT: No additional processing at the receiver
- Cons
  - MRT: Requires channel knowledge at the transmitter
  - MRC: requires some extra processing at the receiver.

MIMO: spatial multiplexing

- The channel is a (N, M)-dimensional matrix
- The optimal technique is called *spatial multiplexing* based on the *singular value decomposition* (SVD) of the channel matrix H.
- By employing SVD and coordinating the precoding weights at the transmitter with the combiner weights at the receiver, it is possible to create a certain number of independent orthogonal channels.
- Assuming that $M = N$, spatial multiplexing creates N *independent spatial channels*.

Singular value decomposition

- Every matrix $A \in C^{m \times n}$ can be decomposed as
$$A = U\Sigma V^H$$
- where $U \in C^{m \times p}, V C^{n \times p}$ are unitary matrices and

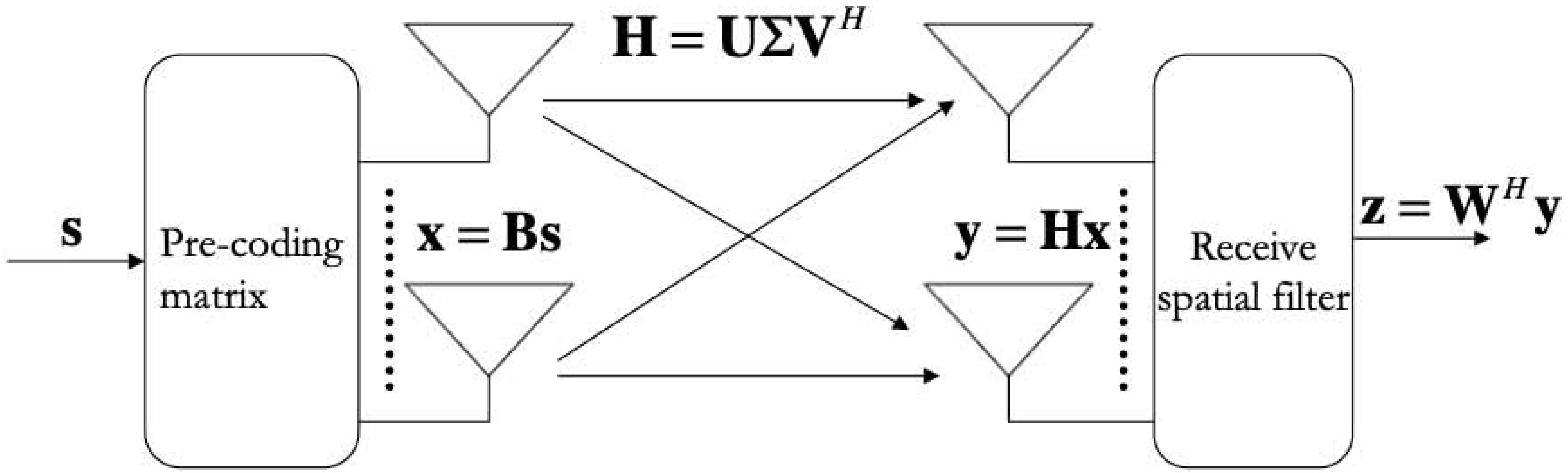
  $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p) \in C^{p \times p}$ is a diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_p$ and $p = \text{rank}(A)$.
- In a MIMO system if the channel is sufficiently multipath, the channel matrix is full rank and its rank is
$$p = \text{rank}(H) = \min(N_R, N_T)$$

Optimal MIMO scheme: spatial multiplexing

- By choosing $B = V$ and $W = U$, the MIMO channel is separated into a set of p parallel SISO channels

$$z = U^H(U\Sigma V^H V s + n) = \Sigma s + n$$

- The channel gains are the p singular values of the matrix H



MIMO: spatial multiplexing

- The channel is a (N, M)-dimensional matrix
- The optimal technique is called *spatial multiplexing* based on the *singular value decomposition* (SVD) of the channel matrix H.
- By employing SVD and coordinating the precoding weights at the transmitter with the combiner weights at the receiver, it is possible to create a certain number of independent orthogonal channels.
- Assuming that $M = N$, spatial multiplexing creates N *independent spatial channels*.

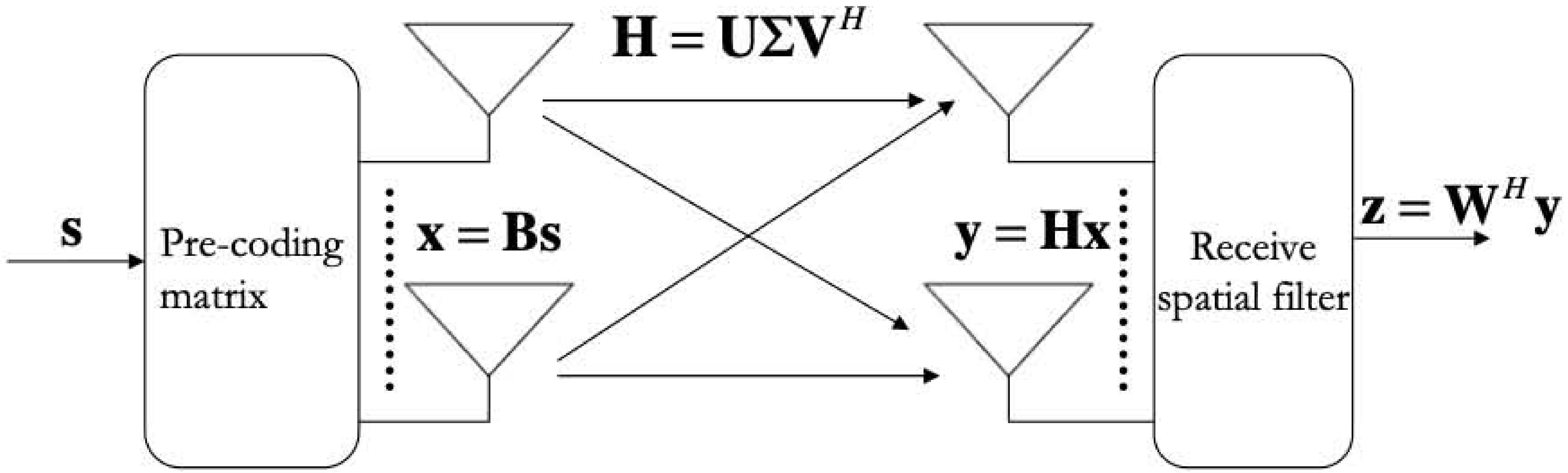
Singular value decomposition

- Every matrix $A \in C^{m\times n}$ can be decomposed as
$$A = U\Sigma V^H$$
- where $U \in C^{m\times p}, VC^{n\times p}$ are unitary matrices and

  $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_p) \in C^{p\times p}$ is a diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_p$ and $p = \mathrm{rank}(A)$.
- In a MIMO system if the channel is sufficiently multipath, the channel matrix is full rank and its rank is
$$p = \mathrm{rank}(H) = \min(N_R, N_T)$$

Optimal MIMO scheme: spatial multiplexing

- By choosing $B = V$ and $W = U$, the MIMO channel is separated into a set of p parallel SISO channels
$$z = U^H(U\Sigma V^H V s + n) = \Sigma s + n$$
- The channel gains are the p singular values of the matrix H



Frequency diversity

Adaptive modulation and coding

- The Shannon capacity formula gives the maximum rate which is sustainable over a given transmission channel
$$C = B\log_2(1 + SNR)$$
Where, given the channel gain H, the SNR is measured as
$$SNR = \frac{|H|^2 P}{\sigma^2}.$$
- The *spectral efficiency*, measured in terms of b/s/Hz is measured as $\log_2(1 + SNR)$.

Adaptive modulation and coding

- In a practical system, the spectral efficiency depends on the coding rate and the modulation order.
- Given the symbol timing T and the symbol rate $R_s = \frac{1}{T} \approx B$, a modulation of order M and a coding rate R, the bit rate R_b of a transmission is given by

$$R_b = \log_2 M \, R \frac{1}{T} \approx mRB$$

- Accordingly, since it is $R_b < C$, the modulation order and the coding rate are bounded by

$$mR < \log_2(1 + SNR)$$

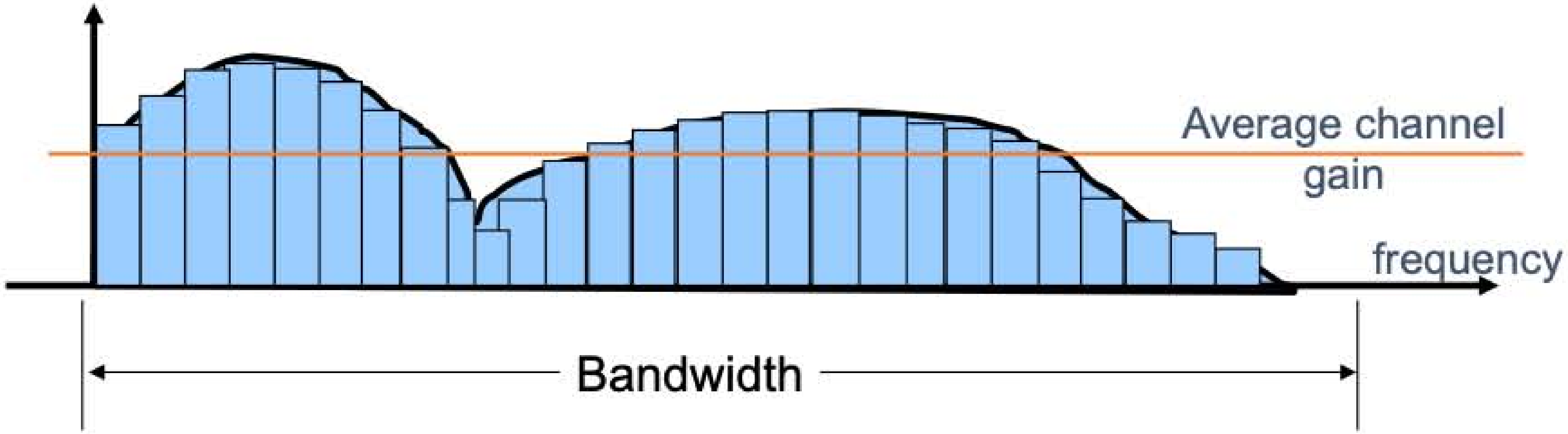
Adaptive modulation and coding

- In practice, there is a direct mapping between the measured SNR (channel quality indicator, CQI) and the specific modulation order and coding rate of a transmission

Radio Bearer Index	Name	Modulation	Channel Coding Rate	Bearer Efficiency (bits/symbol)
1	QPSK 1/12	QPSK	0.0761719	0.1523
2	QPSK 1/9	QPSK	0.117188	0.2344
3	QPSK 1/6	QPSK	0.188477	0.377
4	QPSK 1/3	QPSK	0.300781	0.6016
5	QPSK 1/2	QPSK	0.438477	0.877
6	QPSK 3/5	QPSK	0.587891	1.1758
7	16QAM 1/3	16QAM	0.369141	1.4766
8	16QAM 1/2	16QAM	0.478516	1.9141
9	16QAM 3/5	16QAM	0.601563	2.4063
10	64QAM 1/2	64QAM	0.455078	2.7305
11	64QAM 1/2	64QAM	0.553711	3.3223
12	64QAM 3/5	64QAM	0.650391	3.9023
13	64QAM 3/4	64QAM	0.753906	4.5234
14	64QAM 5/6	64QAM	0.852539	5.1152
15	64QAM 11/12	64QAM	0.925781	5.5547

Adaptive modulation and coding

- Multicarrier transmissions over selective channels experience a set of parallel channels with very diverse channel gains.



- The frequency diversity of the channel can be exploited by adapting the modulation format and the coding rate to the quality of the channel.

Optimal power distribution

- Optimal power allocation problem: distribute the available power P_0 over the N channels with the objective of maximizing the overall rate.
- By setting $\sigma_n^2 = \frac{\sigma^2}{|H(n)|^2}$, the *waterfilling* problem is formulated as

$$\max_P \sum_{n=1}^{N} \log_2\left(1 + \frac{P_n}{\sigma_n^2}\right)$$

$$s.t.$$

$$P_n \geq 0 \quad i = 1, \dots, n$$

$$\sum_{n=1}^{N} P_n = P_0$$

Optimal power distribution

- The solution of the power allocation problem can be found by employing Lagrange multipliers, which are commonly used to address constrained optimization problems.
- The optimal solution takes the form

$$P_n^\star = \max\{0, \mu - \sigma_n^2\} = (\mu - \sigma_n^2)^+ \quad n = 1, \ldots, N$$

  where the constant μ is chosen in such a way that the power constraint is met.

Waterfilling: how to compute μ

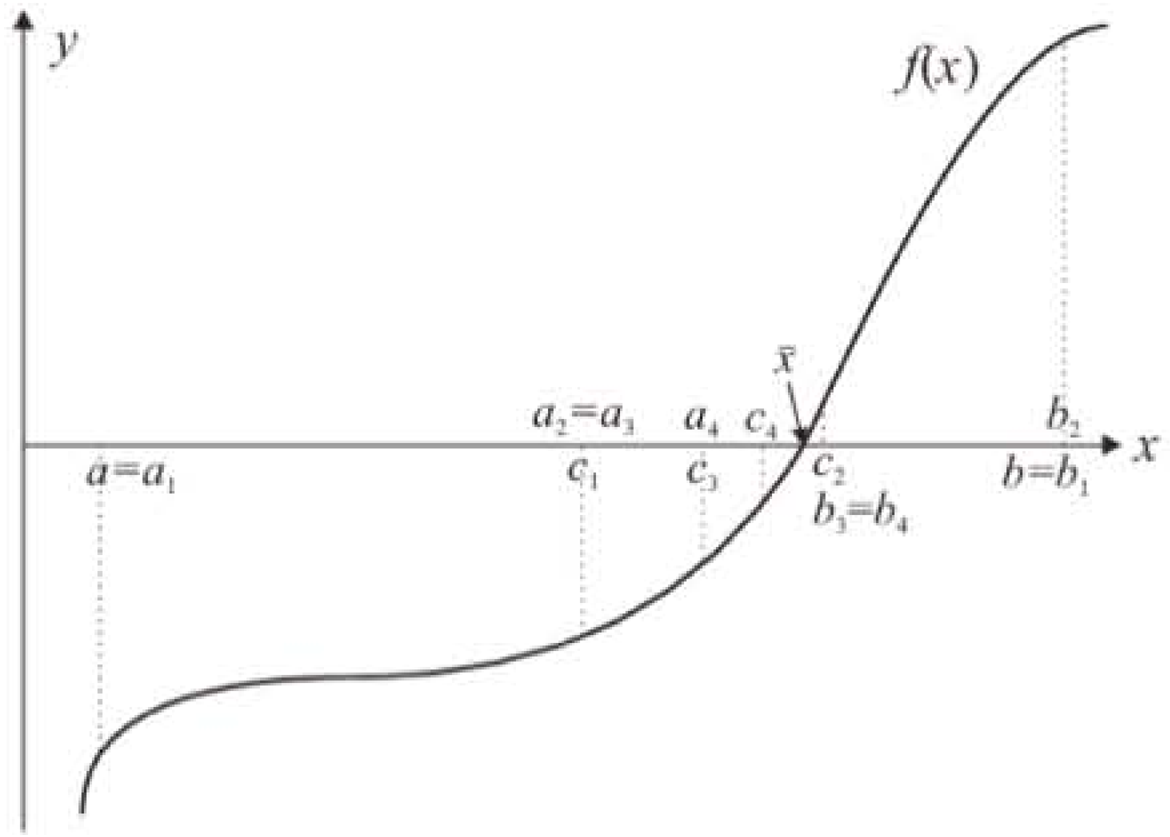
- The unknown value of μ can be found by plugging the expression of the power in equality constraint and it is the solution to the nonlinear equation

$$\sum_{n=1}^{N} (\mu - \sigma_n^2)^+ = P_0$$

- The algorithm takes its name by the characteristics of water to fill gradually a basin occupying firs the deeper holes. In our case the deeper quotes correspond to the best channels.

Waterfilling solution: bisection method

- The bisection method is an iterative algorithm used to find the roots of a continuous monotonic function $f: R^n \to R$.
- We can use the bisection method to find the root of
- $f(\mu) = \sum_{n=1}^{N}(\mu - \sigma_n^2)^+ - P_0$
- The initial points can be
  - $a_1 = 0 \Rightarrow f(0) = -P_0 < 0$
  - $b_1 = \frac{\sum_{n=1}^{N} \sigma_n^2 + P_0}{N} \Rightarrow f(b) \geq 0$

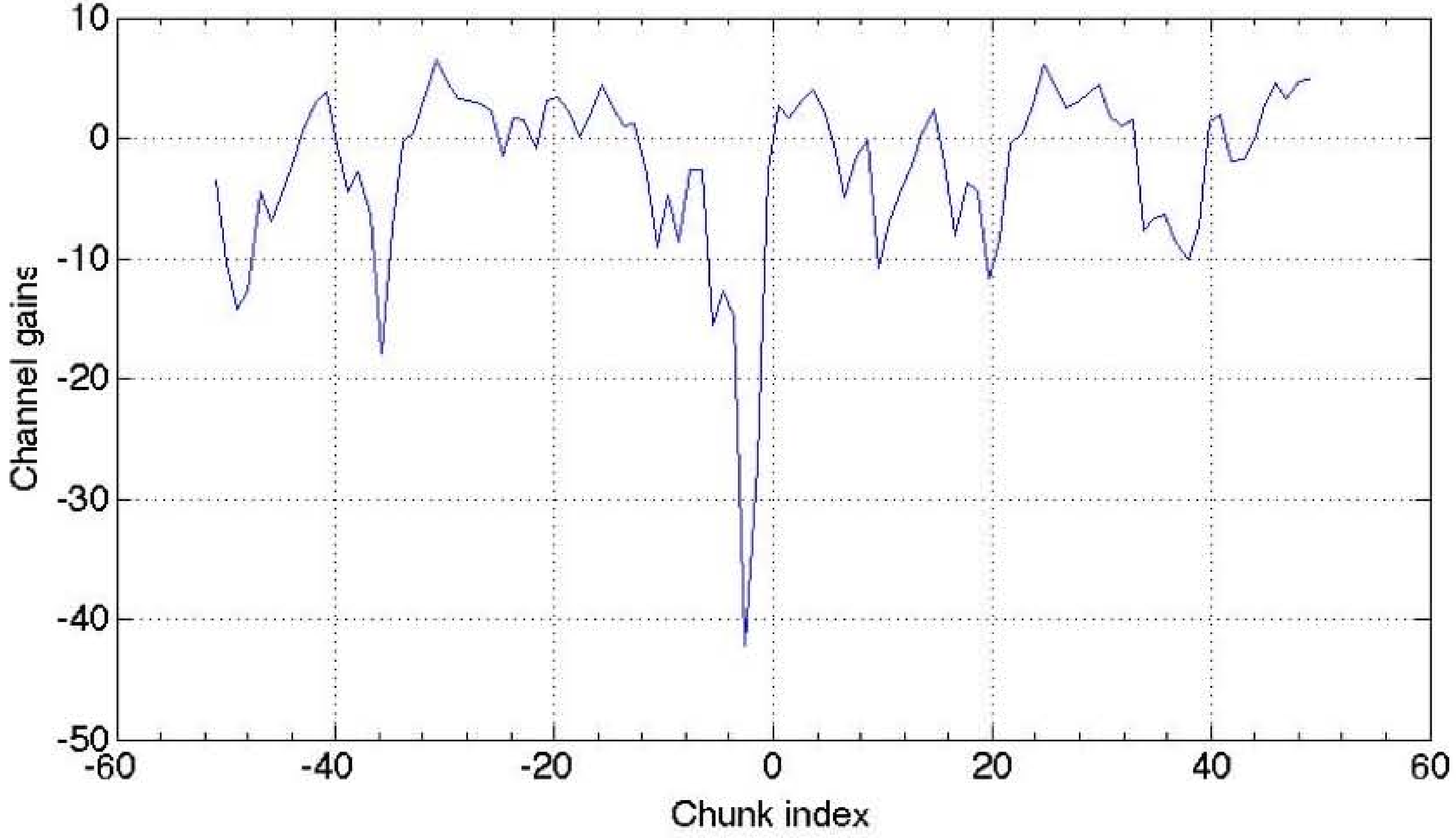


Waterfilling solution: the bisection method

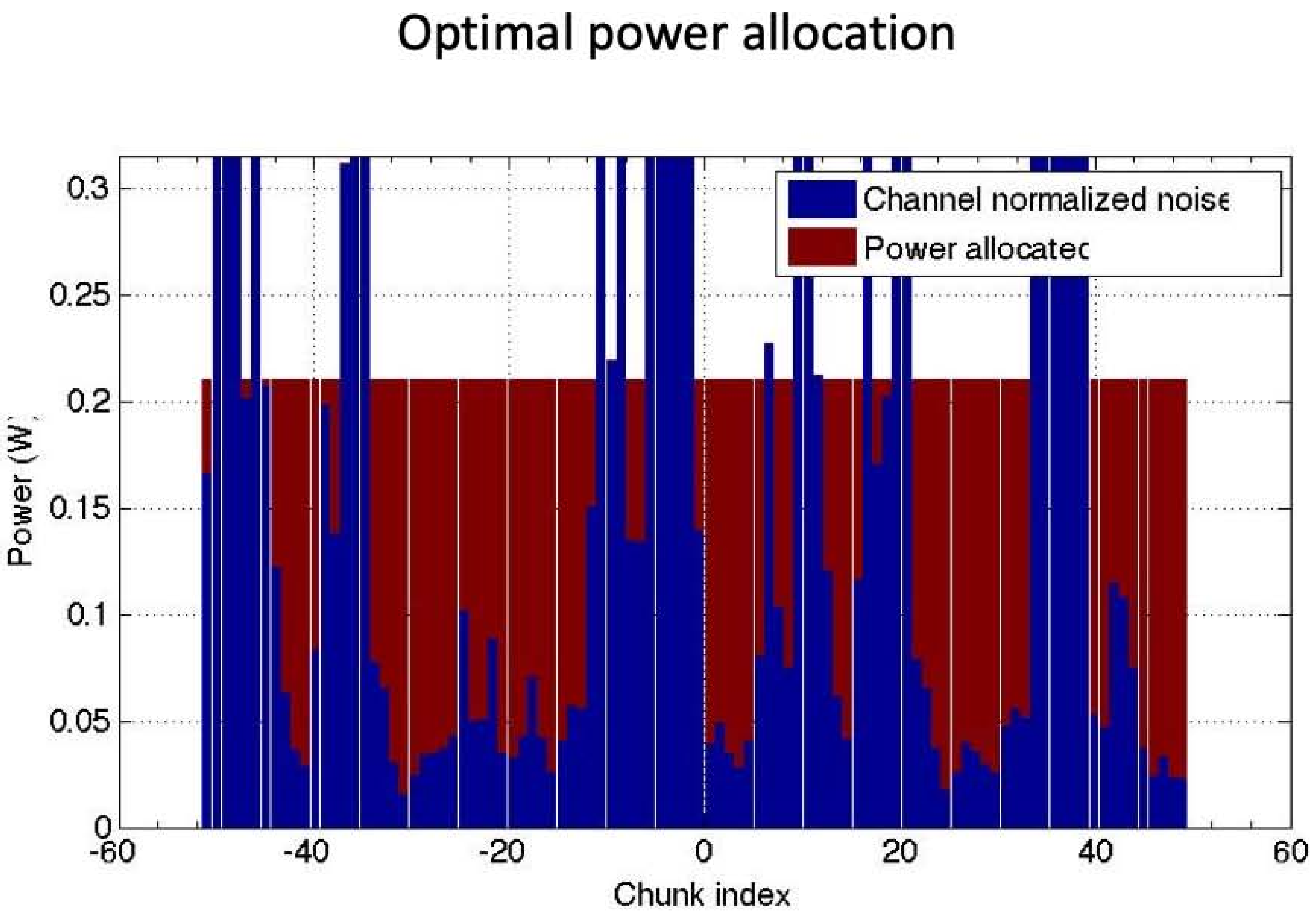
- The main idea is that at the end of each iteration the interval containing the solution is halved.
- At iteration k, the solution is inside the interval (a_k, b_k) with $f(a_k) < 0$ and $f(b_k) > 0$, and the following steps are executed
  - Take the center of the interval $c_k = \frac{a_k+b_k}{2}$;
  - Compute $f(c_k)$,
    - if $f(c_k) < 0 \Rightarrow a_{k+1} = c_k, b_{k+1} = b_k;$
    - if $f(c_k) > 0 \Rightarrow a_{k+1} = a_k, b_{k+1} = c_k.$
- When $|f(c)|$ is sufficiently small, the algorithm stops iterating and returns c_k.

Waterfilling example

OFDM channel

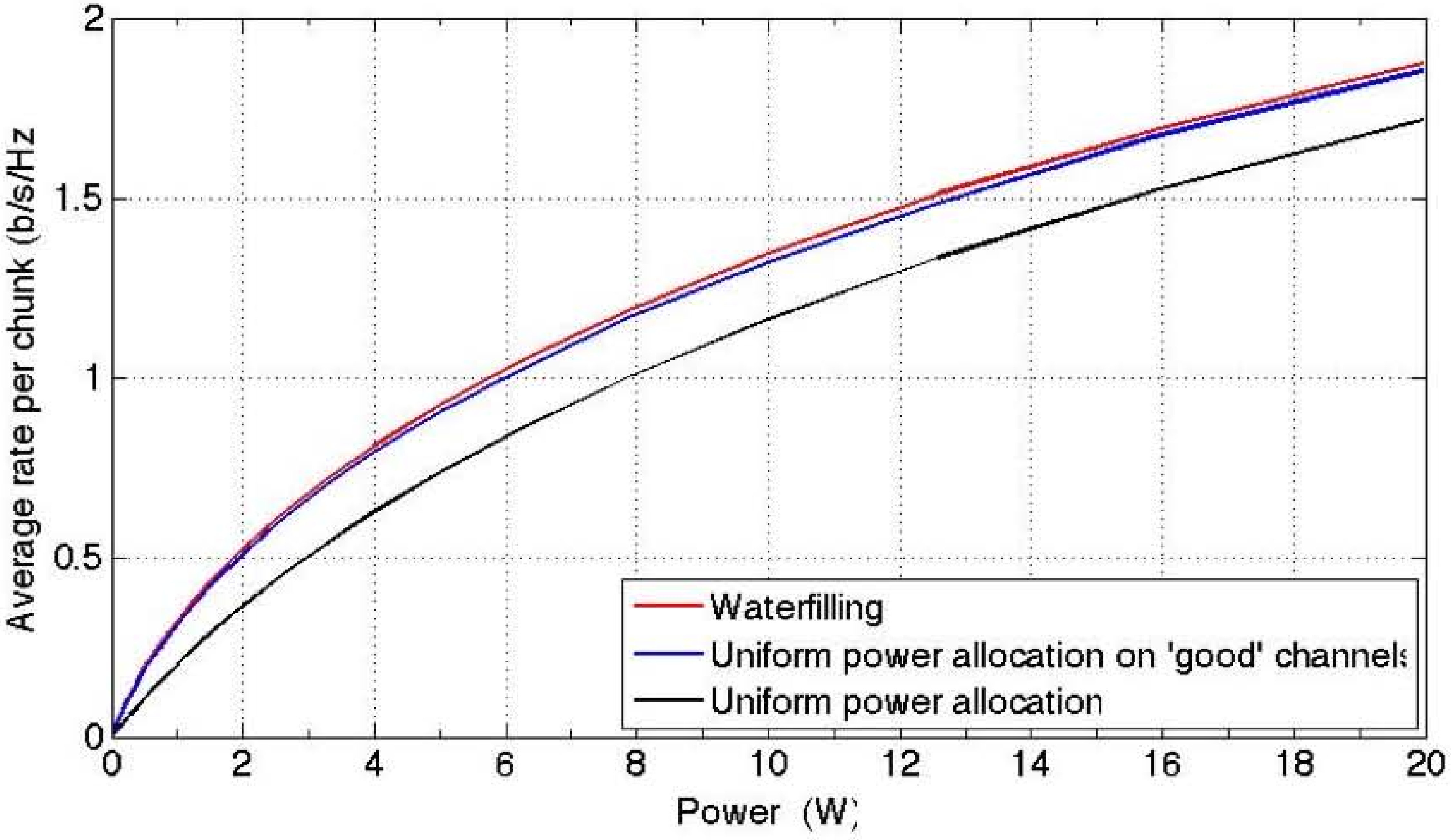


Waterfilling example



Waterfilling example

Achieved rate vs power



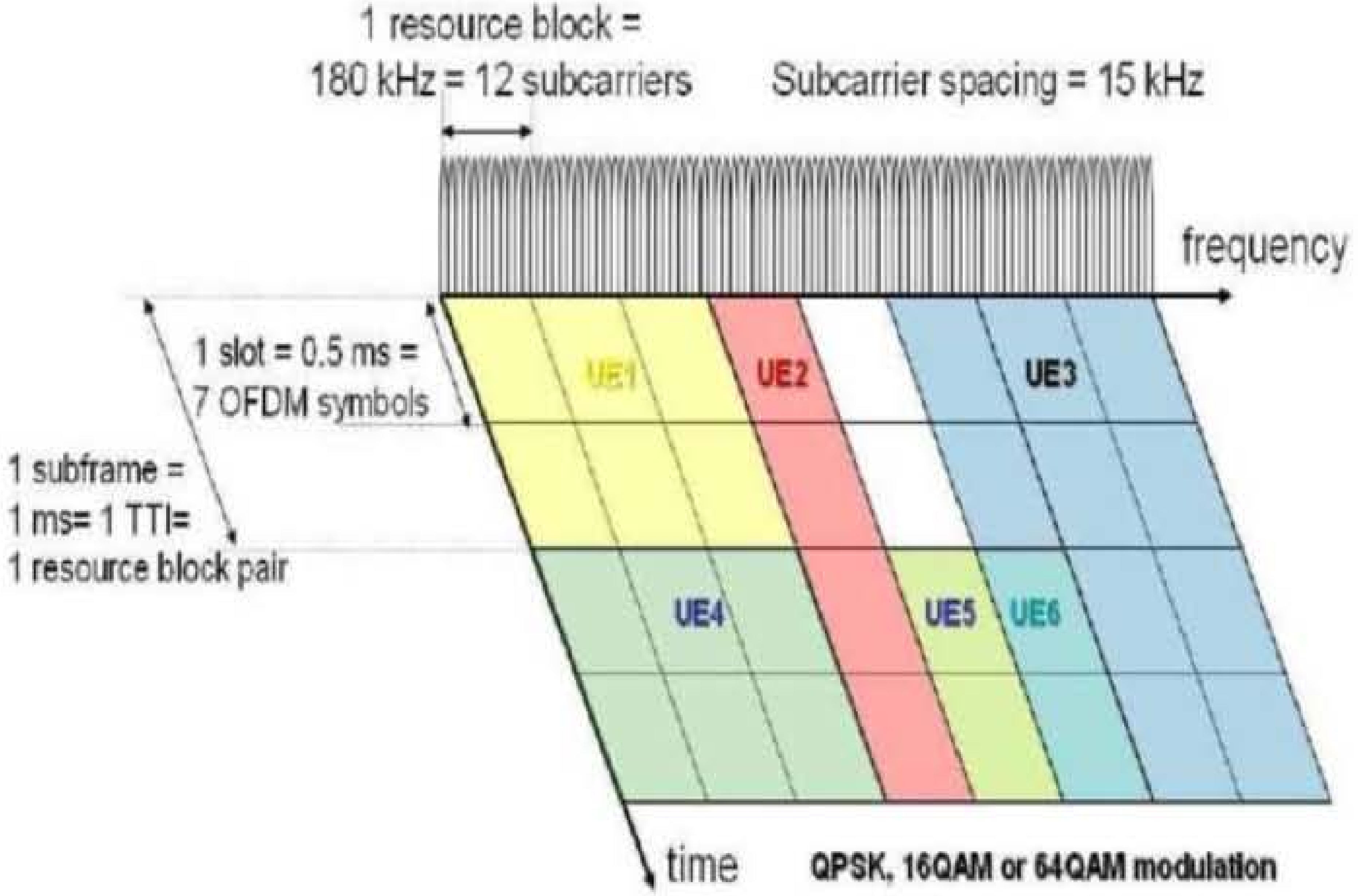
LTE physical layer

LTE

- Long Term Evolution (LTE) is the 4th wireless communication standard
  - 1G: various national analog FDMA-based systems
  - 2G: GSM. First digital standard. Maximum data rate supported $R = 9.6$ kb/s
  - 3G: UMTS (CDMA)
  - 4G: LTE (OFDM-based as WiFi)
  - 5G: NR (currently deployed now)
- LTE is a very flexible communication system mainly used for data transmissions

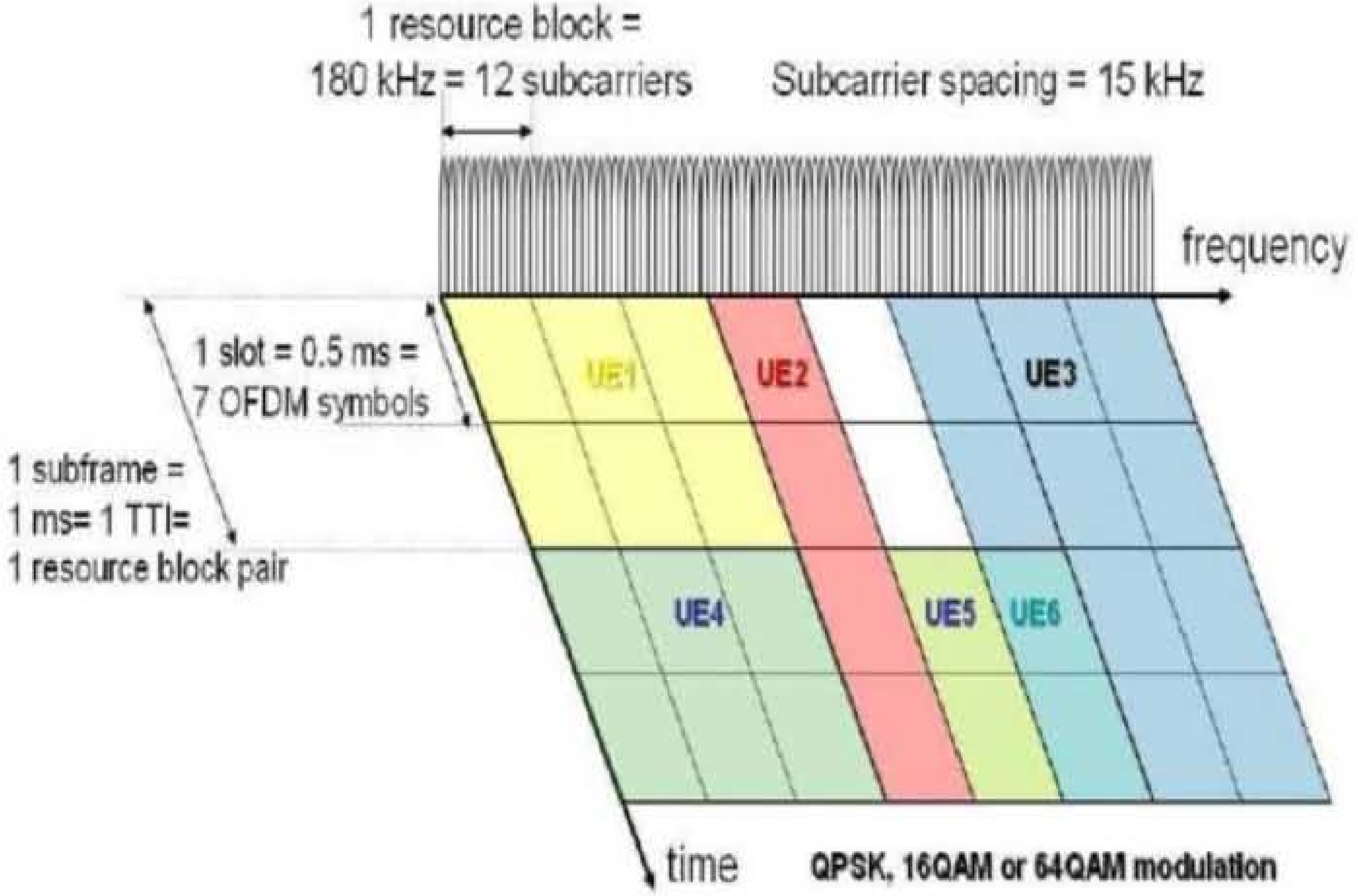
Pyhysical layer LTE numerology

- LTE is based on OFDM
- The sampling time is
$$f_s = 30.72 \text{ MHz}$$
- FFT size is $N = 2048$
- Subcarrier's bandwidth is
$$\Delta f = \frac{30.72}{2048}\text{MHz} = 15 \text{ kHz}$$
- A single *slot* groups 7 OFDM blocks and has a duration of 0.5 ms



Pyhysical layer LTE numerology

- The multiple access technique is OFDMA: groups of subcarriers are allocated to the user.
- The minimum allocation unit is a resource block (RB). A RB contains 12 subcarriers and spans 180 kHz for the duration of a slot.



Pyhysical layer LTE: peak data rate

- Recent high-end mobile phones fall in the 19-20 category list.
- Raw Bandwidth:

  Of the 2048 available subcarriers only 1200 (100 RB) are used the remaining 848 are 0-level guard subcarrier. The maximum available raw bandwidth for data transmission is

$$B_{raw} = 1200x15 \text{ kHz} = 18 \text{ MHz}$$

- Available bandwidth

  Approximately 4 MHz are spent for sending *control* and *synchronization information*

$$B_{av} = B_{raw} - 4 \text{ MHz} = 14 \text{ MHz}$$

Pyhysical layer LTE: peak data rate

- Modulation and coding

  In LTE *coding rate* ranges in the interval 0.0762 - 0.9258. The modulation used are 4-QAM to 256-QAM. Modulation order and coding rate are based on the radio link quality.

  Radio link quality is estimated based on CQI (Channel Quality Indicator), which is measured by the terminal and fed back to the base station. The highest coding rate (including CRC) is $R = 0.9258$ and the largest modulation order is $M = 256$ (8 bits per symbol).

  The maximum bit rate per SISO channel is is

$$R_{1x1} = 0.9258x8x14 \text{ MHz} \approx 100 \text{ Mb/s}$$

Pyhysical layer LTE: peak data rate

- MIMO

  In LTE various configurations are supported (1x1,2x1,1x2,2x2,4x4) depending on
    - user equipment capabilities (number of antennas);
    - eNode-B capabilities: cell traffic, hardware limits;
    - channel conditions.
  - Most performing scheme is spatial multiplexing 4x4 , which creates 4 parallel channels. With 4x4 MIMO spatial multiplexing the peak data rate is

$$R_{4x4} = 4xR_{1x1} = 400 \text{ Mb/s}$$

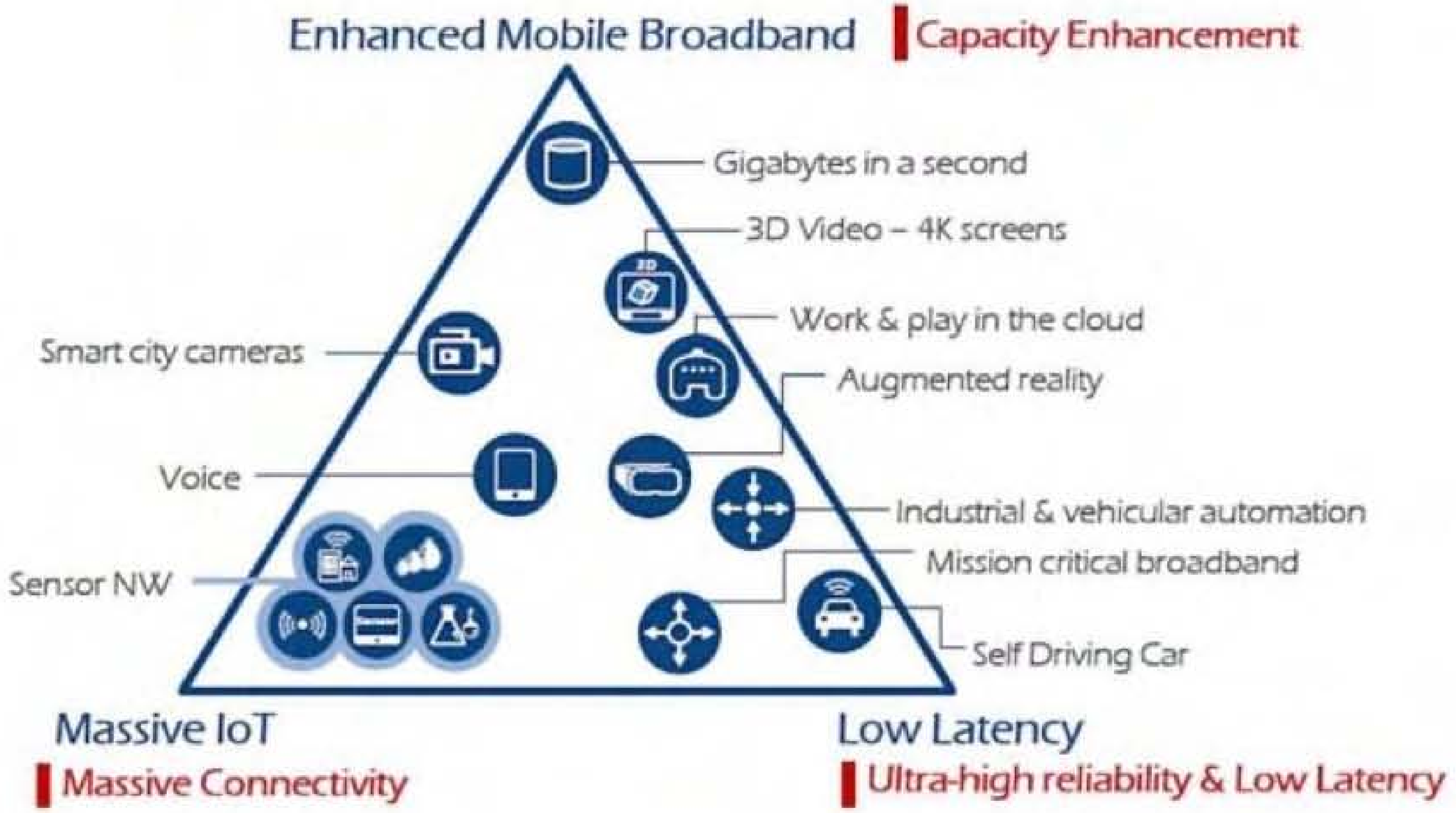
- Carrier aggregation
  - Most advanced terminals can aggregate several 20 MHz bandwidth together
  - Iphone 11 pro: 3 with $4x4$ MIMO and 2 with 2x2 MIMO

$$R_I = 3x400 + 2x200 = 1.6 \text{ Gb/s}$$

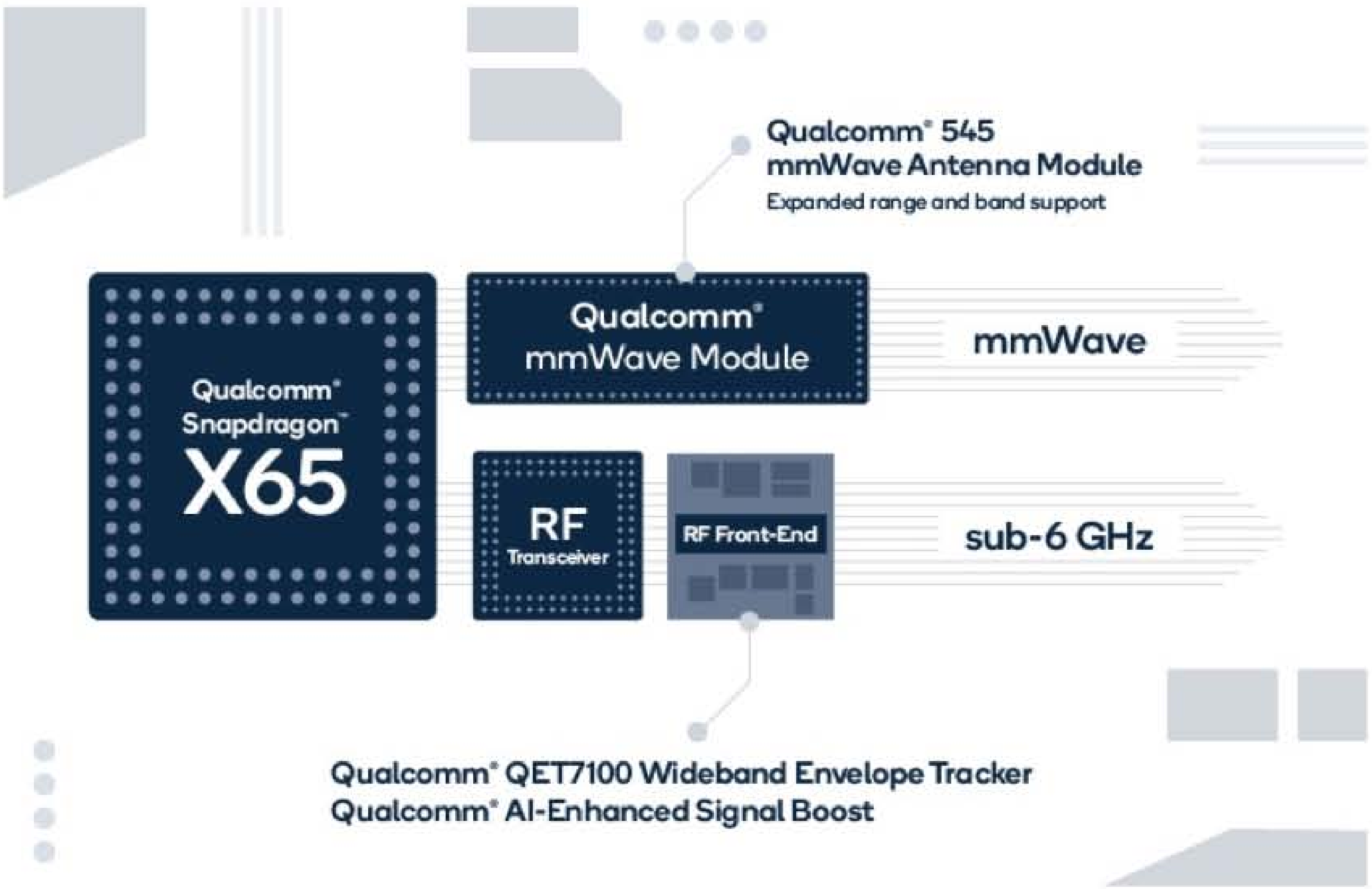
  - Samsung S20+

$$R_s = 5x400 = 2 \text{ Gb/s}$$

5G New Radio (NR)



Most recent 5G chip



Features

- 10 Gbps peak speeds in 5G standalone and non-standalone modes
- 3GPP Release 16 support
- Upgradable architecture for rapid feature rollout
- 5G mmWave-sub6 aggregation
- Global 5G band support including the new n259 (41 GHz), n70, n53
- Advanced power-saving tech
  - Qualcomm® 5G PowerSave 2.0
  - Qualcomm® Wideband Envelope Tracking (7th gen)
  - Qualcomm® AI-Enhanced Signal Boost

Specifications

- 5G Chipset: Qualcomm® Snapdragon™ X65 Modem-RF System
- 5G Spectrum: mmWave-sub6 aggregation, sub-6 carrier aggregation (FDD-TDD, FDD-FDD, TDD-TDD), FDD-TDD support for uplink-CA, Dynamic Spectrum Sharing (DSS)
- 5G Modes: FDD, TDD, SA (standalone), NSA (non-standalone)
- 5G mmWave specs: 1000 MHz bandwidth, 10 carriers, 2x2 MIMO
- 5G sub-6 GHz specs: 300 MHz bandwidth, 256-QAM, 4x4 MIMO
- 5G Peak Download Speed: 10 Gbps
- 5G Global Multi-SIM support
- Cellular Technology: 5G NR, LTE, LAA, WCDMA (DB-DC-HSDPA), TD-SCDMA, CDMA 1x, GSM/EDGE, CBRS