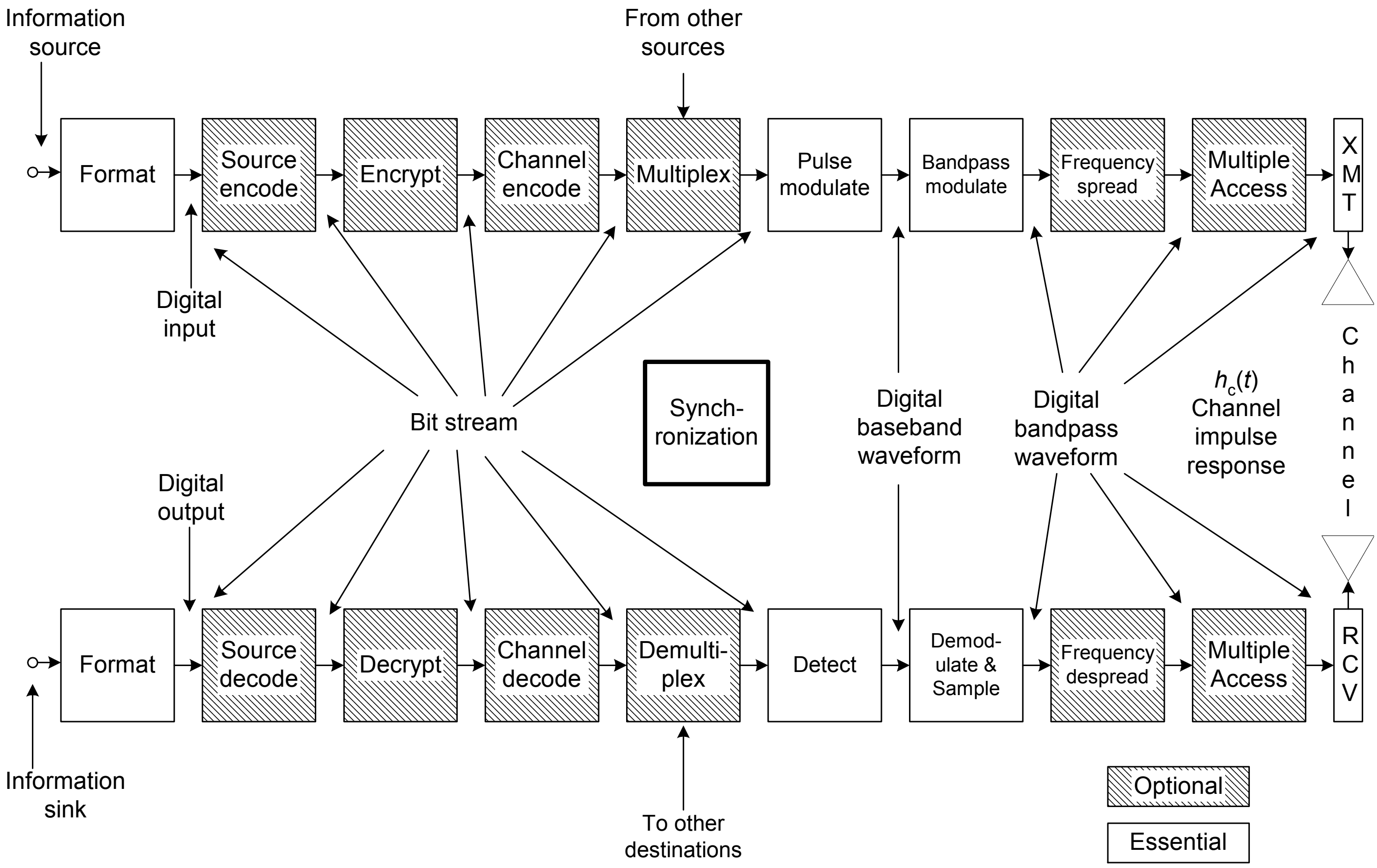
Introdução à Teoria da Informação: Codificação de Fonte

PÓS-GRADUAÇÃO EM ENGENHARIA DE REDES E SISTEMAS DE TELECOMUNICAÇÕES

Prof. Rausley Adriano Amaral de Souza
rausley@inatel.br

Conteúdo

- A *Entropia* como uma medida básica de informação.
- O Teorema da *Codificação de Fonte* e algoritmos de compactação de dados.
- *Informação Mútua* e sua relação com a *capacidade do canal de informação* para a transmissão de informação.
- Teorema da *Codificação de Canal* como base para comunicações confiáveis.
- Teorema da *Capacidade de Informação* como a base para a solução de compromisso entre a largura de faixa do canal e a relação sinal-ruído.



Block diagram of a typical digital communication system

Referências Bibliográficas

- Este material está baseado nas seguinte referências:
  - Notas de Aulas do prof. Geraldo Gil. **Limites Fundamentais na Teoria da Informação.**
  - Sistemas de Comunicação. Analógicos e Digitais. Simon Haykin. 4ª Edição. Bookman
  - Digital Communications – Fundamentals and Applications. Bernard Sklar. Second Edition. Prentice Hall.
- A Universal Algorithm for Sequential Data Compression. Lempel-Ziv Paper: IEEE Transactions on Information Theory, 23(3), pp. 337–343, May 1977. http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv_lempel_1977_universal_algorithm.pdf
- Shannon Paper: http://www.stanford.edu/class/ee104/shannonpaper.pdf
- A Mathematical Theory of Communication. C. E. SHANNON: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf
- Huffman's original article: D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., September 1952, pp 1098–1102 http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf
- Proof of the Kraft-McMillan Inequality: http://www.cl.cam.ac.uk/teaching/2004/InfoTheory/proofKraftMcMillan.pdf

Duas questões fundamentais

- Qual é a complexidade irredutível abaixo da qual um sinal não pode ser comprimido?
- Qual é a máxima taxa de transmissão para uma comunicação confiável em um canal ruidoso?
- Resposta: codificação de fonte (comunicação eficiente) e codificação de canal (comunicação confiável).

Incerteza, Informação e Entropia

- Seja um *experimento probabilístico* que envolva a observação da saída de uma fonte de eventos discretos em unidades de intervalo de tempo (intervalo de sinalização).
- Estes eventos podem ser modelados como variáveis discretas aleatórias (s_k) que fazem parte de um conjunto ou *alfabeto* ($\mathbf{S}$).
  - $\mathbf{S} = \{s_0, s_1, \ldots , s_{K-1}\}$
  - probabilidades $P(\mathbf{S} = s_k) = p_k$, para $k = 0, 1, \ldots , K\text{-}1$.

$$\sum_{k=0}^{K-1} p_k = 1$$

Incerteza, Informação e Entropia

- Símbolos emitidos pela fonte sejam estatisticamente independentes
- *Fonte discreta sem memória* (tem o sentido de que o símbolo emitido a qualquer tempo é independente de uma escolha prévia).
- *Um sistema com realimentação não é uma fonte discreta sem memória*

Incerteza, Informação e Entropia

- Antes de o evento $S = s_k$ ocorrer, há incerteza. Quando o evento $S = s_k$ ocorre, há surpresa. Depois da ocorrência do evento $S = s_k$, há ganho na quantidade de informação, cuja essência pode ser vista como a resolução da incerteza.
- *Incerteza ou surpresa*: Se não há surpresa não há informação.
- A quantidade de informação, $I(s_k)$, obtida por um evento, s_k

$$I(s_k) = \log_2\left(\frac{1}{p_k}\right) \text{bits}$$

Propriedades da Quantidade de Informação

$$1. \quad I(s_k) = 0 \qquad \text{para} \quad p_k = 1$$

– Evidentemente, se estivermos absolutamente certos do resultado de um evento, mesmo antes de ele ocorrer, nenhuma informação será ganha.

$$2. \quad I(s_k) \geq 0 \qquad \text{para} \quad 0 \leq p_k \leq 1$$

– Isto equivale a dizer que a ocorrência de um evento $S = s_k$ fornece alguma informação ou não fornece qualquer informação, jamais provoca perda de informação

Propriedades da Quantidade de Informação

$$3. \quad I(s_k) > I(s_l) \qquad \text{então} \quad p_k < p_l$$

– Ou seja, quanto menos provável um evento for, mais informação ganharemos quando ele ocorrer.

$$4. \quad I(s_k s_l) = I(s_k) + I(s_l)$$

– Se s_k e s_l forem estatisticamente independentes.

Exemplo

- Admitamos que p indique a probabilidade de algum evento. Represente graficamente a quantidade de informação obtida pela ocorrência deste evento para $0 \leq p \leq 1$.

Entropia

- A média da quantidade de informação $I(s_k)$ de uma fonte de *alfabeto* ($\mathbf{S}$) é dada por

$$H(\mathbf{S}) = E[I(s_k)]$$

$$H(\mathbf{S}) = \sum_{k=0}^{K-1} p_k \log_2\left(\frac{1}{p_k}\right)$$

- $H(\mathbf{S})$ é definida como *entropia*.
- A entropia determina a quantidade média de informação por símbolo (evento) da fonte.

Propriedades da Entropia

$$0 \leq H(\mathbf{S}) \leq \log_2 K$$

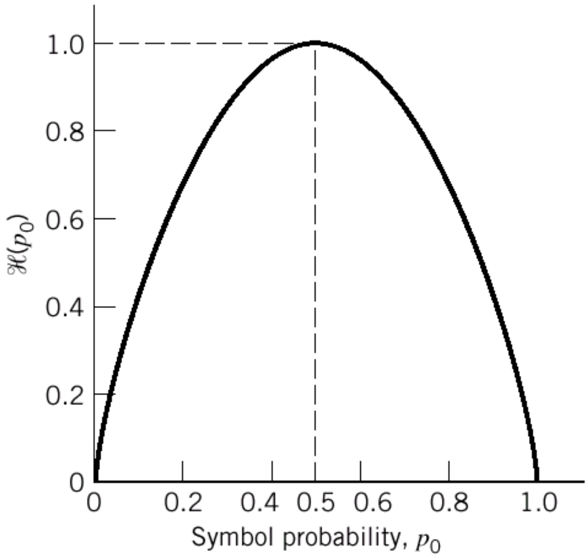
– onde K é o número de símbolos do alfabeto da fonte

- $H(\mathbf{S}) = 0$, se e somente se a probabilidade p_k = 1 para um dado valor de k e todas as outras probabilidades são iguais a zero. Neste caso não há incerteza.
- $H(\mathbf{S}) = \log_2 K$, se e somente se $p_k = 1/K$, ou seja, todos os símbolos são equiprováveis. Neste caso a incerteza é máxima.

Exemplo

- Considere uma fonte discreta sem memória que emite os símbolos s_0 = 0 e s_1 = 1, com probabilidades p_0 e p_1, respectivamente.
  – Dica:

$$x \log x \to 0 \text{ quando } x \to 0$$



$$H(S) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0) \text{ bits}$$

Exemplo

- Uma fonte emite um de quatro símbolos possíveis durante cada intervalo de sinalização. Os símbolos ocorrem com probabilidades $p_0=0.4$, $p_1=0.3$, $p_2=0.2$ e $p_3=0.1$. Encontre a quantidade de informação obtida observando-se a emissão desses símbolos pela fonte.

Exemplo

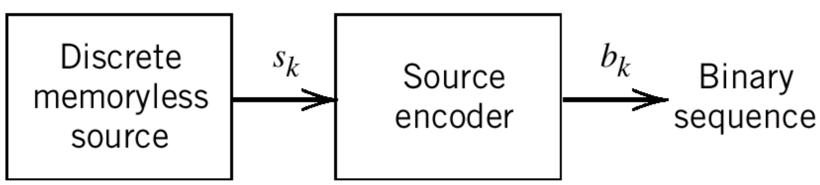
- Uma fonte emite um de quatro símbolos s_0, s_1, s_2 e s_3 com probabilidades 1/3, 1/6, 1/4 e 1/4, respectivamente. Os símbolos sucessivos emitidos pela fonte são estatisticamente independentes. Calcule a entropia da fonte.

Exemplo

- Calcule a média de informação, em bits/caractere, para o idioma inglês, assumindo que cada um dos 26 caracteres no alfabeto ocorrem com igual probabilidade. Não considere os espaços e pontuações.
- Como é sabido, os caracteres no alfabeto não aparecem com igual frequência no idioma inglês (ou em qualquer outro idioma), a resposta anterior representará um limitante superior na quantidade média de informação por caractere. Refaça os cálculos assumindo que os caracteres ocorrem com as seguintes probabilidades:
  - p = 0.10: para as letras a, e , o, t
  - p = 0.07: para as letras h, i, n, r, s
  - p = 0.02: para as letras c, d, f, l, m, p, u, y
  - p = 0.01: para as letras b, g, j, k, q, v, w, x,z

Teorema da Codificação de Fonte

- Transmissão *eficiente*



- Necessidade de conhecimento da estatística da fonte
- Código Morse (código de tamanho variável)
  - *e* representada por um "·"
  - *q* representada pela sequência "**-** · **- -**".

Codificador de fonte

- Objetivo: retirar redundância
- Eficiência: De maneira geral, quanto menor o número de dígitos para a representação da mensagem melhor o sistema
- Classificação dos códigos de fonte
  - Comprimento fixo
  - Comprimento variável
- Se alguns símbolos são mais prováveis que outros, os códigos de comprimento variável são mais eficientes

Código ASCII

USASCII code chart

b_7 b_6 b_5 →					000	001	010	011	100	101	110	111
b_4	b_3	b_2	b_1	Row ↓ / Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	–	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	—	o	DEL

- $b_1, b_2, b_3, b_4, b_5, b_6, b_7$

Souce: Wikipedia: http://en.wikipedia.org/wiki/ASCII

Comprimento médio

$$\overline{L} = \sum_{k=0}^{K-1} p_k l_k$$

– onde l_k é o número de bits médio da palavra código correspondente ao símbolo k.

– Admitamos que $L_{\min}$ seja o valor mínimo possível de $\overline{L}$. Definimos, então, a eficiência de codificação como

$$\eta = \frac{L_{\min}}{\overline{L}}$$

Exemplo

- Um arquivo com 10k símbolos é transmitido. Quantos bits, em média, são transmitidos com os códigos I e II.

Símbolo	Probabilidade	Código I	Código II
S_0	0,500	00	0
S_1	0,250	01	10
S_2	0,125	10	110
S_3	0,125	11	111

Primeiro Teorema de Shannon

- Teorema da Codificação de Fonte: Dada uma fonte discreta sem memória de entropia $H(\mathbf{S})$, o comprimento médio das palavras códigos para um esquema de codificação livre de distorção é limitado como

$$\bar{L} \geq H(\mathbf{S})$$

  - A entropia $H(\mathbf{S})$ representa um limite fundamental ao número médio de bits por símbolo da fonte necessário para representar uma fonte discreta sem memória no sentido de que ele pode ser tão pequeno quanto a entropia $H(\mathbf{S})$, mas não menor do que ela.

Eficiência de codificação

$$\eta = \frac{H(\mathbf{S})}{\bar{L}}$$

- Uma vez que o Teorema da Codificação de Fonte admite que $\bar{L} = H(\mathbf{S})$, então o maior valor possível para a eficiência de codificação de fonte é igual a unidade.

Compactação de Dados

- Transmissão eficiente: informações redundantes devem ser removidas do sinal que será transmitido.
- *Compactação:* não há perda de informação (compressão sem perdas),
- *Compressão:* admite-se perda de informação.
- Na compactação de dados, o processo de codificação de fonte é limitado, necessariamente, pela entropia da fonte.
- A compactação de dados é obtida atribuindo-se descrições breves aos resultados mais frequentes da saída de fonte e descrições mais longas aos menos frequentes.

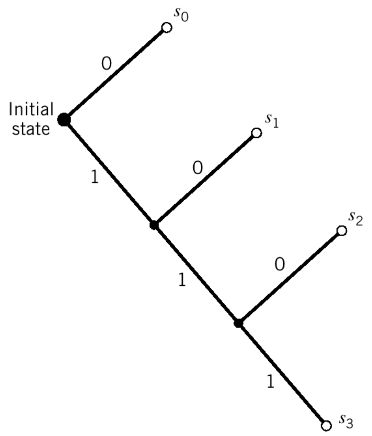
Codificação de Prefixo

- Um código prefixo é aquele que nenhuma palavra código é prefixo para qualquer outra palavra código (singularmente decodificável).
- Exemplos de códigos de fonte

Símbolo	Probabilidade	Código I	Código II	Código III
S_0	0,5	0	0	0
S_1	0,25	1	10	01
S_2	0,125	00	110	011
S_3	0,125	11	111	0111

Decodificação de um Código de Prefixo

- Árvore de decodificação para o Código II.



- Decodificação é inequívoca e *instantânea*.
- Decodifique 1011111000

Desigualdade de *Kraft-McMillan*

- Uma condição necessária, mas não suficiente, para que um código possa ser um código prefixo

$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1$$

- Exemplo para os códigos I, II e III

Propriedades adicionais

- Se a probabilidade de um evento é igual ao inverso da base numérica da codificação elevado ao comprimento da palavra código correspondente a este evento, ou seja

$$p_k = 2^{-l_k}$$

então

$$\overline{L} = H(S)$$

Entretanto, isso é uma casualidade uma vez que é necessário que a correspondência estabelecida ocorra para todas as probabilidades e suas respectivas palavras códigos. Para esta condição particular, diz-se que o código prefixo está *casado* com a fonte.

Exemplo

- Analise os quatro códigos relacionados abaixo.
  a) Dois desses quatro códigos são códigos-prefixo. Identifique-os e construa suas árvores de decisão individuais.
  b) Aplique a desigualdade de Kraft-McMillan aos códigos. Discuta os resultados.

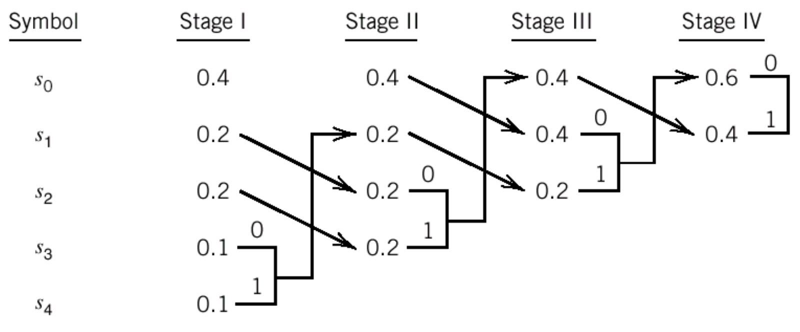
Símbolo	Código I	Código II	Código III	Código IV
S_0	0	0	0	00
S_1	10	01	01	01
S_2	110	001	011	10
S_3	1110	0010	110	110
S_4	1111	0011	111	111

Algoritmo de Huffman

- Exemplo de Código de Prefixo
- Os símbolos da fonte são listados em ordem decrescente de probabilidade. Aos dois símbolos de menor probabilidade, localizados no pé da coluna, são atribuídos os 0 e 1.
- Os dois símbolos de menor probabilidade são agrupados em um único símbolo, cuja probabilidade é a soma das probabilidades. As probabilidades são novamente listadas em ordem decrescente de probabilidade em uma segunda coluna. Na reordenação dos símbolos por ordem decrescente de probabilidade o novo símbolo, resultado do agrupamento, é colocado na posição mais alta possível, sem violar a ordem decrescente de probabilidade.
- O procedimento é repetido até que restem apenas dois símbolos, aos quais são atribuídos novamente os bits 0 e 1.
- As palavras códigos binárias são obtidas fazendo o caminho inverso, a partir da última coluna, em direção a primeira coluna, anotando-se os bits encontrados em cada percurso, até cada símbolo listado na primeira coluna

Exemplo

- Suponha uma fonte discreta sem memória $\mathbf{S} = \{s_0, s_1, s_2, s_3, s_4\}$ com probabilidades $P(\mathbf{S} = s_k)$ = {0,4; 0,2; 0,2; 0,1; 0,1}.
  - Palavras códigos, Comprimento médio das palavras códigos, Entropia, Eficiência, Árvore de decisão



(a)

Symbol	Probability	Code word
s_0	0.4	00
s_1	0.2	10
s_2	0.2	11
s_3	0.1	010
s_4	0.1	011

(b)

Algoritmo de Huffman

- O processo de codificação de Huffman não é único
  - No agrupamento dos símbolos, o rótulo 0 e 1 para cada símbolo é arbitrário, poderia ser 1 e 0. Entretanto, a mesma convenção deve ser mantida do início ao fim.
  - A colocação de símbolos combinados com mesma probabilidade que outro(s), na posição mais alta da coluna, também é arbitrário, poderia ser na posição mais baixa.

Exemplo

- Refaça o exemplo anterior, adotando como padrão a colocação de símbolos combinados com mesma probabilidade que outro(s) na posição mais baixa da coluna.

Variância do comprimento das palavras códigos

$$\sigma^2 = \sum_{k=0}^{K-1} p_k \left(l_k - \overline{L}\right)^2$$

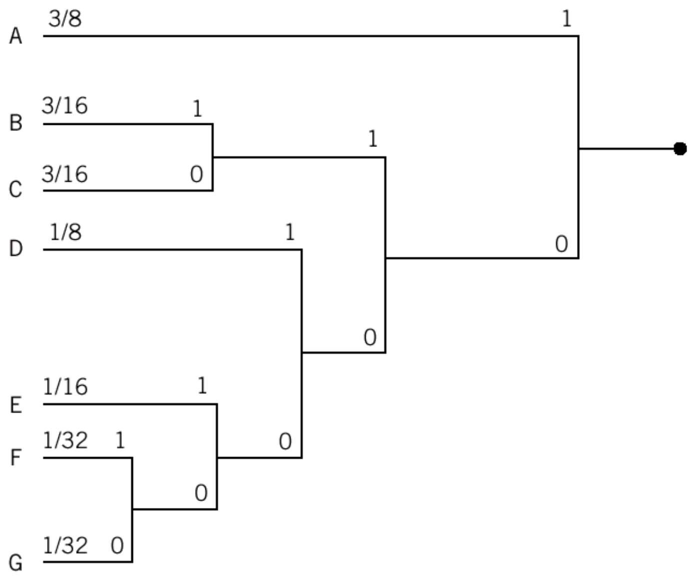
– Calcule a variância para os dois exemplos anteriores.

Exemplo

- Decodifique sequência 00010110111000 usando a árvore do exemplo no slide 32.
- O algoritmo de Huffman produz um código prefixo com comprimento médio próximo da entropia e inequivocamente decodificável.

Exemplo

- A Figura exibe uma árvore de Huffman. Qual é a palavra código para os símbolos A, B, C, D, E, F e G representados por ela? Quais são seus tamanhos de palavra-código individuais?



Extensão de uma Fonte Discreta sem Memória

- Considere $\mathbf{S} = \{s_0, s_1, s_2\}$ com
  - $P(\mathbf{S} = s_k) = \{p_0; p_1; p_2\}$.
- Extensão da fonte $\mathbf{S}$ em $\mathbf{S}^2$.

Blocos	s_0s_0	s_0s_1	s_0s_2	s_1s_0	s_1s_1	s_1s_2	s_2s_0	s_2s_1	s_2s_2
Símbolos	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8
Probabilidades	$p_0 \cdot p_0$	$p_0 \cdot p_1$	$p_0 \cdot p_2$	$p_1 \cdot p_0$	$p_1 \cdot p_1$	$p_1 \cdot p_2$	$p_2 \cdot p_0$	$p_2 \cdot p_1$	$p_2 \cdot p_2$

$$\mathbf{S}^2 = \{\sigma_0, \sigma_1, \ldots, \sigma_8\}$$

$$H(\mathbf{S}^n) = nH(\mathbf{S})$$

Exemplo

- Suponha uma fonte discreta sem memória $\mathbf{S} = \{s_0, s_1, s_2\}$ com probabilidades $P(\mathbf{S} = s_k) = \{0{,}6; 0{,}3; 0{,}1\}$. Admita que essa fonte deva ser codificada com um código prefixo com eficiência igual ou maior do que 0,95.
  a) Codificação da fonte S
  b) Codificação da fonte S^2

Codificação de Lempel-Ziv

- O algoritmo de Huffman pressupõe o conhecimento prévio da estatística da fonte.
- Para aplicações tais como compactações de textos existem códigos que se adaptam à estatística da fonte.
- O algoritmo de Lempel-Ziv é adaptativo e não necessita do conhecimento prévio da estatística da fonte.
- Usa palavras de comprimento fixo para representar um número variável de símbolos da fonte.
- Basicamente, a codificação no algoritmo de Lempel-Ziv é realizada analisando-se o fluxo de dados da fonte em segmentos que sejam as subsequências mais breves não encontradas anteriormente (no dicionário).

Exemplo

- Codificação da sequência 000101110010100101. . .
  – Inicialmente os bits 0 e 1 são previamente armazenados nas posições numéricas 0 e 1, conforme apresentado a seguir. Tais posições numéricas compõem o *livro de códigos* (*codebook*).

Posição numérica	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Subsequências	0	1							
Blocos codificados	-	-							

Exemplo

- Codificação da sequência 000101110010100101. . .
  – Em seguida, a subsequência mais curta, ainda não armazenada em nenhuma posição binária, é armazenada na posição 2 pelo codificador. Esta subsequência é **00.** O processo de codificação consiste em transmitir a posição de memória onde se encontra a parte da subsequência **00** já armazenada anteriormente, na forma binária, e o que é *novidade* simplesmente é repetido. Ou seja, da subsequência **00**, o primeiro **0** se encontra na posição numérica **0** (**000** em binário) e o segundo **0** simplesmente é repetido.

Posição numérica	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Subsequências	0	1	00						
Blocos codificados	-	-	000 0						

Exemplo

- Codificação da sequência 00010111001010010 1. . .
  - Novamente a próxima subsequência mais curta ainda não armazenada em nenhuma posição binária, que é a subsequência **01**, é armazenada na posição 3. A posição de memória onde se encontra a parte da subsequência **01** já armazenada é a posição 0 (**000** em binário) e a parte *novidade*, ou seja **1**, é repetida. Logo, o bloco codificado agora é **0001**.

Posição numérica	0	1	2	3	4	5	6	7	8
Subsequências	0	1	00	01					
Blocos codificados	-	-	0000	000 1					

Exemplo

- Codificação da sequência 00010111001010010 1. . .
- Este procedimento é repetido conforme apresentado na sequência a seguir.

Posição numérica	0	1	2	3	4	5	6	7	8
Subsequências	0	1	00	01	011				
Blocos codificados	-	-	0000	0001	011 1				

Posição numérica	0	1	2	3	4	5	6	7	8
Subsequências	0	1	00	01	011	10			
Blocos codificados	-	-	0000	0001	0111	001 0			

Posição numérica	0	1	2	3	4	5	6	7	8
Subsequências	0	1	00	01	011	10	010		
Blocos codificados	-	-	0000	0001	0111	0010	011 0		

Exemplo

- Codificação da sequência 000101110010100101. . .

Posição numérica	0	1	2	3	4	5	6	7	8
Subsequências	0	1	00	01	011	10	010	100	
Blocos codificados	-	-	0000	0001	0111	0010	0110	101 0	

Posição numérica	0	1	2	3	4	5	6	7	8
Subsequências	0	1	00	01	011	10	010	100	101
Blocos codificados	-	-	0000	0001	0111	0010	0110	1010	101 1

Codificação de Lempel-Ziv

- O codificador e o decodificador possuem o mesmo número de memórias no livro de códigos e, portanto, o decodificador conhece o tamanho do bloco codificado.
- O decodificador também inicia o processo de decodificação com os bits 0 e 1 armazenados nas posições numéricas 0 e 1.
- O decodificador compõe o seu próprio livro de código usando as memórias na mesma sequência que o codificador.
- Em comparação com a codificação de Huffman, o algoritmo de Lempel-Ziv usa códigos com tamanho fixo para representar um número variável de símbolos da fonte.

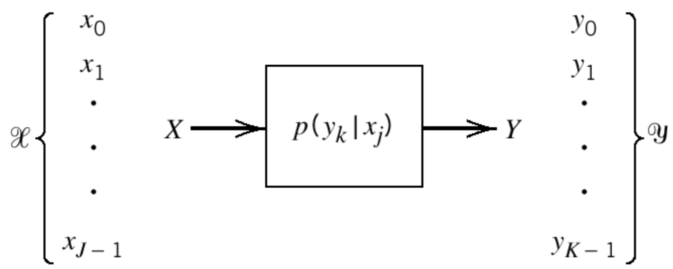
Codificação de Lempel-Ziv

- Em uma primeira análise, podemos chegar à conclusão que este algoritmo não compacta nada, pelo contrário. Essa é uma idéia falsa uma vez que a sequência utilizada é muito curta para o algoritmo *reconhecer* padrões de repetição. Conforme mencionado no livro texto, na prática, são utilizados blocos com 12 bits, o que resulta em 4096 posições no *livro código*. Além disso, o algoritmo de Lempel-Ziv reconhece redundância entre caracteres, o que não ocorre com o algoritmo de Huffman. Particularmente eficiente para compactação de textos, o algoritmo de Lempel-Ziv pode atingir uma taxa de compactação de 55% contra 43% do algoritmo de Huffman, para textos em inglês

Exemplo

- Considere a seguinte sequência binária 11101001100010110100101101000 1...Use o algoritmo de Lempel-Ziv para codificar essa sequência. Suponha que os símbolos 0 e 1 já estejam no *codebook*.

Canais Discretos sem Memória



$$p(y_k \mid x_j) = P(Y = y_k \mid X = x_j)$$

- *Discreto:* os dois alfabetos possuem tamanhos finitos e o termo
- *Sem memória:* o símbolo corrente presente na saída depende apenas do símbolo corrente presente na entrada e de nenhum outro anterior.
- $p(y_k|x_j)$ representa o conjunto das probabilidades de transição

Matriz de canal ou matriz de transição

$$\mathbf{P} = \begin{bmatrix} p(y_0 \mid x_0) & p(y_1 \mid x_0) & \cdots & p(y_{K-1} \mid x_0) \\ p(y_0 \mid x_1) & p(y_1 \mid x_1) & \cdots & p(y_{K-1} \mid x_1) \\ \vdots & \vdots & & \vdots \\ p(y_0 \mid x_{J-1}) & p(y_1 \mid x_{J-1}) & \cdots & p(y_{K-1} \mid x_{J-1}) \end{bmatrix}$$

$$\boxed{\sum_{k=0}^{K-1} p(y_k \mid x_j) = 1}$$

$$p(x_j, y_k) = p(y_k \mid x_j) p(x_j)$$

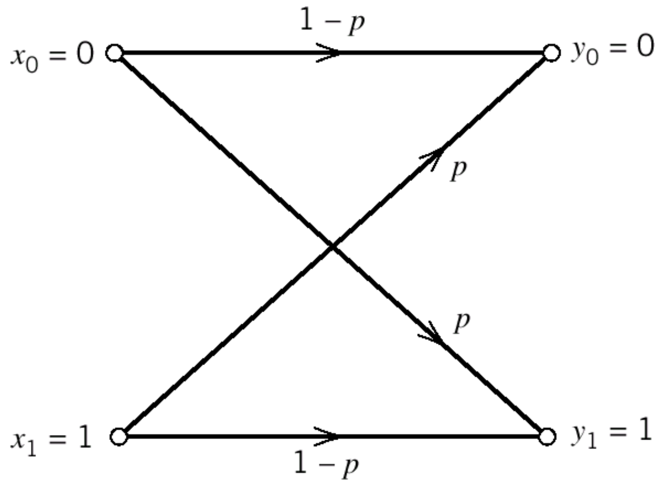
$$\boxed{p(y_k) = \sum_{j=0}^{J-1} p(y_k \mid x_j) p(x_j)}$$

- Cada linha de **P** corresponde a um símbolo fixo de entrada e cada coluna a um símbolo fixo de saída.

Exemplo

- *Canal binário simétrico (BSC)*
- $J = K = 2$
- Alfabeto de entrada: $x_0 = 0, x_1 = 1$
- Alfabeto de saída: $y_0 = 0, y_1 = 1$

$$p_{10} = P(y = 1 \mid x = 0)$$
$$p_{01} = P(y = 0 \mid x = 1)$$
$$p_{10} = p_{01} = p$$

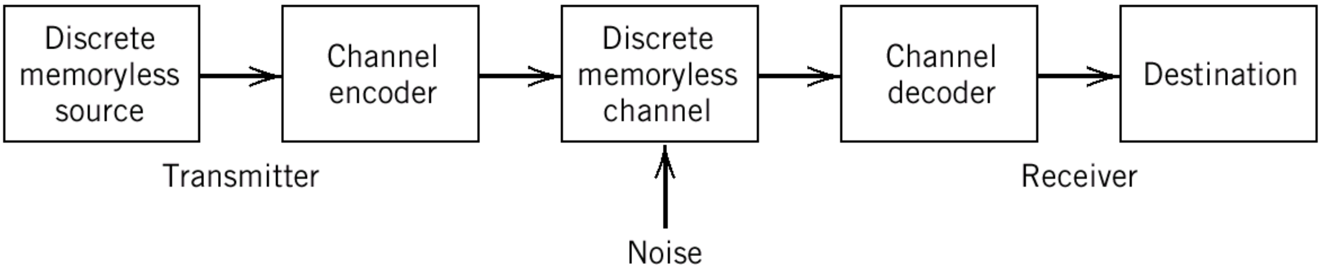


Exemplo

a) Seja o canal BSC do slide anterior. Considere que os símbolos 0 e 1 têm a mesma probabilidade de ocorrer. Encontre a probabilidade de os símbolos binários 0 e 1 aparecerem na saída do canal.

b) Repita o cálculo anterior, supondo que os símbolos binários de entrada 0 e 1 tenham probabilidades ¼ e ¾ de ocorrer, respectivamente.

Teorema da Codificação de Canal



- ***Será que existe um esquema de codificação de canal eficiente o suficiente para tornar a probabilidade de erro de bit arbitrariamente baixa sem que a quantidade de redundância introduzida seja demasiadamente alta?***

Terceiro Teorema de Shannon

- Teorema da Capacidade de Informação ou Teorema da Capacidade de Canal
- A capacidade de informação de um canal contínuo com largura de faixa de B hertz, perturbado por ruído Gaussiano branco aditivo de densidade espectral $N_0/2$, é dada por

$$\boxed{C = B\log_2\left(1+\frac{S}{N_0 B}\right)}$$

dada em bits por segundo, onde P é a potência média do sinal transmitido.

Se um canal discreto sem memória tem uma capacidade C e uma fonte gera informações a uma taxa menor que C, então existe uma técnica de codificação tal que a saída da fonte pode ser transmitida ao longo do canal com uma probabilidade arbitrariamente baixa de erro de símbolo[1].

[1] Simon Haykin, Michael Moher. "Sistemas Modernos de Comunicações Wireless" Bookman, 2008.

Terceiro Teorema de Shannon

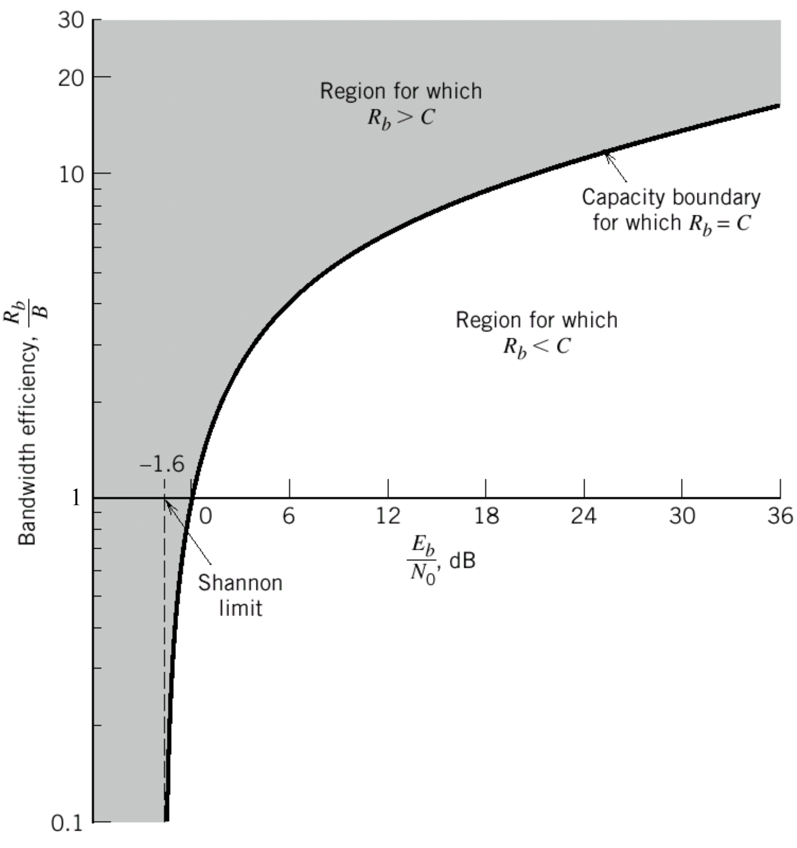
- Para um sistema ideal $R_b = C$.
- Então $S = E_b/T_b = E_bR_b = E_bC$.

$$\frac{C}{B} = \log_2\left(1+\frac{S}{N_0B}\right) = \log_2\left(1+\frac{E_b}{N_0}\frac{C}{B}\right) = \log_2\left(1+\frac{S}{N}\right)$$

$$\frac{E_b}{N_0} = \frac{2^{C/B}-1}{C/B}$$

- $N = N_0 B$

Diagrama de Eficiência de Largura de Banda



- O limite é representado por uma fronteira entre a região em que é possível uma transmissão livre de erros ($R_b < \mathrm{C}$) e a região em que isso não é possível ($R_b > \mathrm{C}$) em um plano da *eficiência de largura de faixa* em função da relação entre a *energia de bit* e a *densidade espectral de ruído*.

Observações

- Limite de Shannon para um canal AWGN (Largura de Banda Infinita): *The ratio of energy per data bit E_b to one-sided noise power spectral density N_0 in an AWGN channel above which errorless transmission is possible when bandwidth limitations are not placed on the signal and transmission is at channel capacity*[1].

$$\lim_{B\to\infty}\left(\frac{E_b}{N_0}\right)=\lim_{B\to\infty}\left(\frac{2^{C/B}-1}{C/B}\right)=\log 2=0.693=-1.6\text{ dB}$$

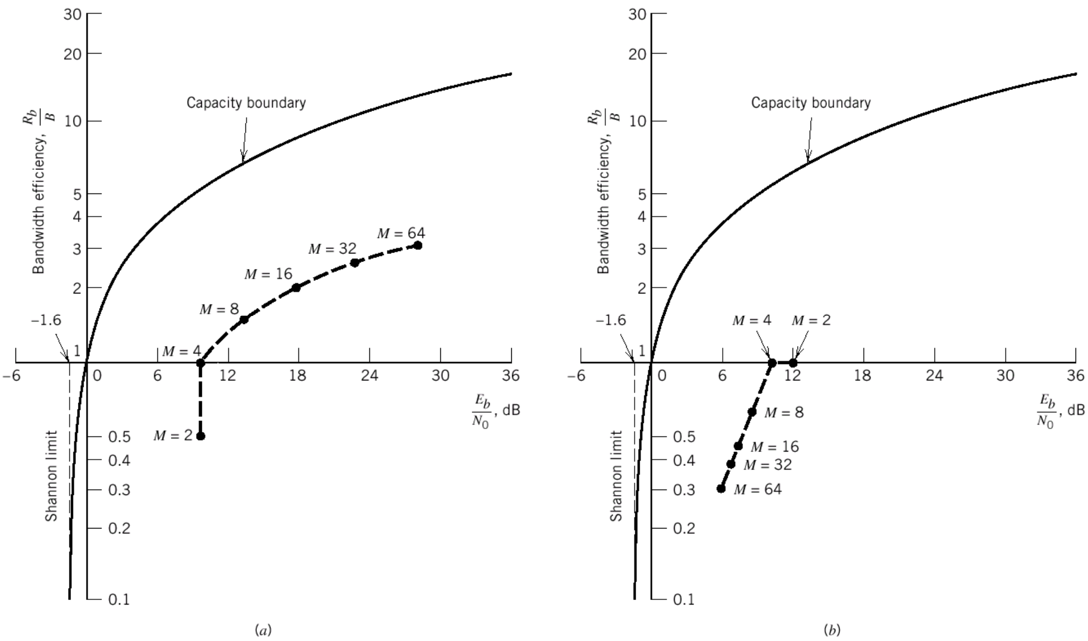
- Compensações entre E_b/N_0, R_b/B e a probabilidade de Erro de símbolo P_e.

[1]Bhargava, V.K. "Forward Error Correction Coding" *Mobile Communications Handbook* Ed. Suthan S. Suthersan Boca Raton: CRC Press LLC, 1999

Exemplo

- Um canal de faixa de voz da rede telefônica tem uma largura de banda de 3.4 kHz.
  a) Calcule a capacidade de informação do canal telefônico correspondente a uma relação sinal-ruído de 30 dB.
  b) Calcule a relação sinal-ruído mínima necessária para suportar transmissão de informação através do canal telefônico a uma taxa de 9600 bps.

PSK *M*-ário e FSK *M*-ário



(*a*) Comparison of *M*-ary PSK against the ideal system for $P_e = 10^{-5}$ and increasing *M*.
(*b*) Comparison of *M*-ary FSK against the ideal system for $P_e = 10^{-5}$ and increasing *M*.

Exemplo

- Considere a necessidade de transmissão a uma taxa de 100 kbps em um sistema de voz (com largura de banda de 3 kHz). É possível a transmissão nesta taxa (livre de erros) com um SNR de 10 dB? Justifique sua resposta. Se não for possível, quais as possíveis modificações no sistema?

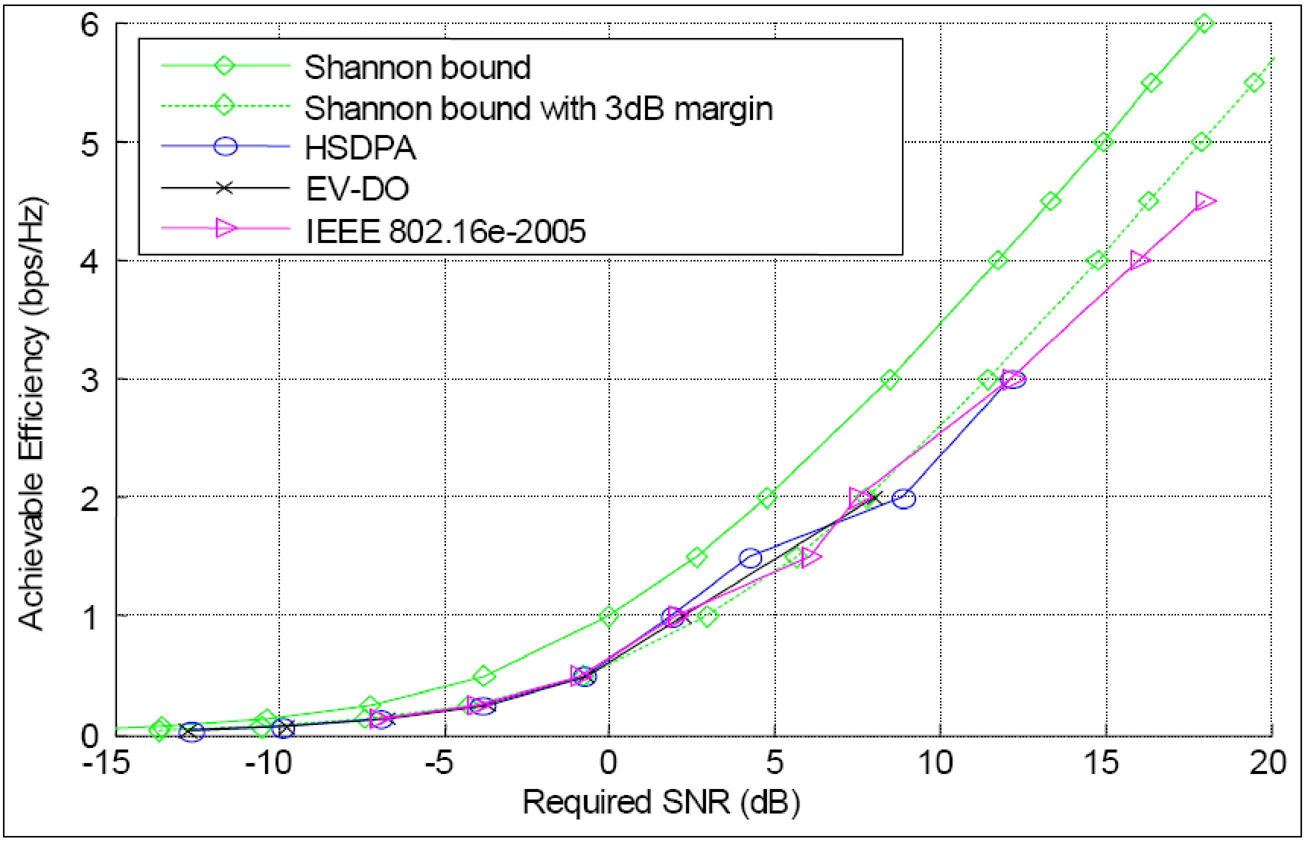
Exemplo

- Uma imagem de televisão em preto e branco pode ser vista como uma imagem que consiste em aproximadamente 3×10^5 elementos, em que cada um tem a mesma probabilidade de ocupar um dos 10 níveis de brilhos distintos. Suponha que (1) a taxa de transmissão seja de 30 quadros de imagem por segundo e que (2) a relação sinal-ruído seja de 30 dB. Usando o teorema da capacidade de informação, calcule a largura de banda mínima necessária para suportar a transmissão do sinal de vídeo resultante.

Futuro

- "When determining the best area on which to focus future technology enhancements, it is interesting to note that HSDPA, 1xEV-DO, and IEEE 802.16e-2005 all have highly optimized links—that is, physical layers. In fact, as shown in Figure 16, the link layer performance of these technologies is approaching the theoretical limits as defined by the Shannon bound. (The Shannon bound is a theoretical limit to the information transfer rate [per unit bandwidth] that can be supported by any communications link. The bound is a function of the Signal to Noise Ratio [SNR] of the communications link.) Figure 16 also shows that HSDPA, 1xEV-DO, and IEEE 802.16e-2005 are all within 2 to 3 decibels (dB) of the Shannon bound, indicating that there is not much room for improvement from a link layer perspective. Note that differences do exist in the design of the MAC layer (layer 2) and this may result in lower than expected performance in some cases as described previously". From: http://www.3gamericas.org/documents/EDGE_HSPA_and_LTE_Broadband_Innovation_Rysavy_Sept_2008.pdf

Figure 16: Performance Relative to Theoretical Limits for HSDPA, EV-DO, and IEEE 802.16e-2005[66]



Exercício Adicional

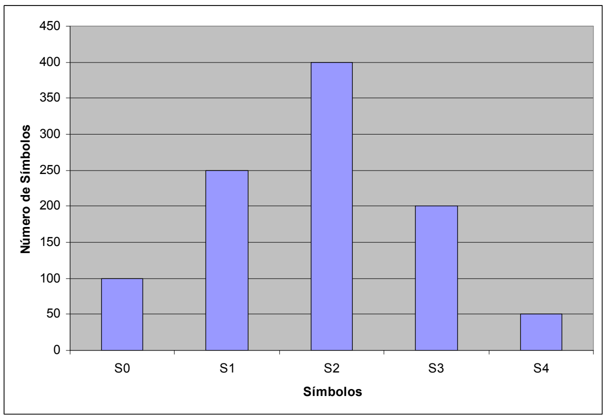
- O alfabeto de uma fonte discreta sem memória é composto por 5 símbolos. Os símbolos emitidos pela fonte devem ser codificados por meio de um código prefixo a partir do histograma da frequência de emissão apresentado abaixo. O número total de símbolos do histograma foi emitido em 1 ms. Pede-se:

a) A entropia da fonte.

b) As palavras códigos de um código prefixo para esta fonte.

c) A eficiência de codificação.

d) Determine qual é a taxa de bit na saída do codificador de fonte.



Exemplo

a) Encontre a capacidade média, em bps, que seria necessária para transmitir um sinal de TV em preto e branco de alta-resolução à taxa de 32 quadros (imagens) por segundo. Cada imagem é composta de até 2×10^6 elementos e 16 diferentes níveis de brilho. Todos os elementos são independentes e tem a mesma probabilidade de ocorrência.

b) Para um TV em cores, este sistema adicionalmente fornece 64 diferentes combinações de cores. Quanto será o adicional de capacidade necessária comparada com o sistema em preto e branco?

c) Encontre a capacidade necessária se, das combinações cor - níveis de brilho, 100 elementos ocorrem com probabilidade 0.003, 300 das combinações ocorrem com probabilidade 0.001 e 624 das combinações ocorrem com probabilidade 0.00064.