

**Multimedia Communications (INT-3305): Final exam - HK1 - 2017-2018**

*Thời gian:* 90 phút

*Không được sử dụng:* Internet

*Được sử dụng:* máy tính cầm tay, cheat sheet

*Tên:* .....

*MSSV:* .....

*Ngày sinh:* .....

*Chữ ký giám thị 1:* .....

*Chữ ký giám thị 2:* .....

**Bài 1**

Các phát biểu sau đúng hay sai:

1. Thuật toán nén LZW không làm giảm Entropy của dữ liệu đầu vào
2. Một encoder mã hoá các symbol thành các codeword (từ mã) như ở dưới.  
 $a=0010$   
 $b=0111$   
 $c=00$   
 $d=11$   
 Encoder này là không phải là một prefix code
3. Sampling (subsampling) có thể coi là một cách nén dữ liệu
4. Các hệ thống đảm bảo QoS theo kiểu IntServe sẽ có khả năng mở rộng (scalability) cao hơn các hệ thống theo kiểu DiffServe
5. Huffman, Arithmetic Coding, và LZW là các mã hoá Entropy coding
6. Entropy coding không khiến dữ liệu bị mất mát
7. Huffman, Arithmetic Coding, và LZW cần biết tần suất (xác suất) xuất hiện của từng symbol
8. Đánh dấu và phân loại gói tin là đủ để đảm bảo thực hiện được đảm bảo QoS ở lớp mạng
9. Các cách tiếp cận để giải quyết bài toán đảm bảo QoS là IntServe, DiffServe, và Laissez Faire
10. Mã hoá kênh (Channel coding) sẽ làm giảm lượng dữ liệu đi
11. Adaptive Playout Delay là một kỹ thuật đảm bảo QoS ở lớp mạng
12. Trong các ứng dụng Multimedia trên mạng Internet, tăng playout delay sẽ làm giảm mất mát gói tin.
13. Weighted Fair Queueing là một thuật toán có thể được áp dụng ở lớp mạng để đảm bảo QoS.
14. Mã sửa lỗi trước (Forward Error Correction - FEC) là một cách mã hoá nguồn (Source coding)
15. Giao thức TCP chạy trên mạng Long-fat (long-fat network) sẽ không đạt được hiệu quả cao

ĐẠI HỌC QUỐC GIA HÀ NỘI

... xuất xuất hiện p(x), thì nội dung thông tin của nó (thông tin chứa) ...  
 ... giá trị biểu tượng, chỉ phụ thuộc vào xác suất xuất hiện ...  
 ... symbols trong alphabet của người:  $H(x) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$  ...  
 ... short codes ...  
 ... (1,2) (1,4) ...  
 ... 3,1) ...  
 ... symbol ...

Bài 2

Hãy encode chuỗi LALAL\* thành nhị phân sử dụng thuật toán LZW 9-bit.

Lưu ý:

- Ký tự \* là ký tự kết thúc, không cần mã hoá.
- Trình bày các bước mã hoá trung gian
- Trình bày chuỗi bit đầu ra thành từng khúc 9 bit sao cho dễ đọc

Gợi ý: Bảng mã ASCII có 256 ký tự. Trong đó 128 ký tự đầu tiên cho ở hình dưới

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	000	NUL (null)	64	40	100	€#64: 8	96	60	140	€#96: 8
1	1	001	SOF (start of heading)	65	41	101	€#65: A	97	61	141	€#97: a
2	2	002	STX (start of text)	66	42	102	€#66: B	98	62	142	€#98: b
3	3	003	ETX (end of text)	67	43	103	€#67: C	99	63	143	€#99: c
4	4	004	EOT (end of transmission)	68	44	104	€#68: D	100	64	144	€#100: d
5	5	005	ENQ (enquiry)	69	45	105	€#69: E	101	65	145	€#101: e
6	6	006	ACK (acknowledge)	70	46	106	€#70: F	102	66	146	€#102: f
7	7	007	BEL (bell)	71	47	107	€#71: G	103	67	147	€#103: g
8	8	010	BS (backspace)	72	48	110	€#72: H	104	68	150	€#104: h
9	9	011	TAB (horizontal tab)	73	49	111	€#73: I	105	69	151	€#105: i
10	A	012	LF (NL line feed, new line)	74	4A	112	€#74: J	106	6A	152	€#106: j
11	B	013	VT (vertical tab)	75	4B	113	€#75: K	107	6B	153	€#107: k
12	C	014	FF (NP form feed, new page)	76	4C	114	€#76: L	108	6C	154	€#108: l
13	D	015	CR (carriage return)	77	4D	115	€#77: M	109	6D	155	€#109: m
14	E	016	SO (shift out)	78	4E	116	€#78: N	110	6E	156	€#110: n
15	F	017	SI (shift in)	79	4F	117	€#79: O	111	6F	157	€#111: o
16	10	020	DLE (data link escape)	80	50	120	€#80: P	112	70	160	€#112: p
17	11	021	DC1 (device control 1)	81	51	121	€#81: Q	113	71	161	€#113: q
18	12	022	DC2 (device control 2)	82	52	122	€#82: R	114	72	162	€#114: r
19	13	023	DC3 (device control 3)	83	53	123	€#83: S	115	73	163	€#115: s
20	14	024	DC4 (device control 4)	84	54	124	€#84: T	116	74	164	€#116: t
21	15	025	NAK (negative acknowledge)	85	55	125	€#85: U	117	75	165	€#117: u
22	16	026	SYN (synchronous idle)	86	56	126	€#86: V	118	76	166	€#118: v
23	17	027	ETB (end of trans. block)	87	57	127	€#87: W	119	77	167	€#119: w
24	18	030	CAN (cancel)	88	58	130	€#88: X	120	78	170	€#120: x
25	19	031	EM (end of medium)	89	59	131	€#89: Y	121	79	171	€#121: y
26	1A	032	SUB (substitute)	90	5A	132	€#90: Z	122	7A	172	€#122: z
27	1B	033	ESC (escape)	91	5B	133	€#91: [	123	7B	173	€#123: {
28	1C	034	FS (file separator)	92	5C	134	€#92: \	124	7C	174	€#124:
29	1D	035	GS (group separator)	93	5D	135	€#93: ]	125	7D	175	€#125: }
30	1E	036	RS (record separator)	94	5E	136	€#94: ^	126	7E	176	€#126: ~
31	1F	037	US (unit separator)	95	5F	137	€#95: _	127	7F	177	€#127: DEL

Source: www.LookupTables.com

Bài 3

Cho chuỗi dữ liệu đầu vào (hàng thứ 2)

vị trí	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
symbol	A	B	A	A	C	A	B	A	A	B	B	C	A	A	A	B	B	A	A	A

1. Hãy tính lượng tin của từng symbol A, B, C
2. Hãy tính entropy của chuỗi dữ liệu trên
3. Hãy vẽ cây mã hoá theo thuật toán mã hoá Huffman
4. Hãy cho biết mã Huffman tương ứng cho mỗi symbol A, B, C

Họ và tên thi: \_\_\_\_\_  
 Số báo danh: \_\_\_\_\_  
 Phòng thi: \_\_\_\_\_

• Symbol x với xác suất xuất hiện...  
 ... không phụ thuộc vào giá trị...  
 ... trung bình lượng tin đầu với mỗi sym...

... Frequent symbols are encoded with...  
 ... 004 => 9 12 5 4  
 ... (3,1) (3,2) (4,7) (1,5) (2,6) ...  
 ... (lossless)  
 ... "hot" (codeword)  
 ... bel. usually (0,1)  
 ... = sym

**Bài 4**

Xét một token bucket với tham số  $r_{token} = 2000 \text{ tokens/s}$ , và bucket size  $b = 1000 \text{ tokens}$ . Kích thước hằng đợi đầu vào của token bucket là vô hạn.

1. Người ta cho 1 luồng dữ liệu chảy liên tục vào đầu vào của token bucket trên trong một thời gian dài, tốc độ dữ liệu đầu ra  $r_{out}$  của token bucket sẽ là bao nhiêu, nếu:

A. tốc độ của luồng dữ liệu đầu vào là  $r_{in} = 3000 \text{ packets/s}$   
 B. tốc độ của luồng dữ liệu đầu vào là  $r_{in} = 1500 \text{ packets/s}$

2. Nếu có 1 burst dữ liệu rộng 500ms chứa 5000 packets áp đến đầu vào của token bucket. Sau bao lâu toàn bộ 5000 packets sẽ được gửi toàn bộ đi. Giả sử rằng tại thời điểm bắt đầu của burst dữ liệu thì bucket đang chứa đầy  $b$  tokens.

$$\frac{5000}{0.5} = 10^4 \text{ packet/s} + b$$

$$\frac{r_{in} t + b}{r_{out}}$$

$$r_{max} = \frac{r_{in}}{r_{out}}$$

$$8 \cdot \frac{1}{48} \cdot 10^3$$

$$\frac{8}{48} \cdot 10^3$$

**Bài 5**

Cho tín hiệu  $s(t)$  có dạng sóng

$$s(t) = 4 \times \sin(2\pi 8 \times 10^3 t) + 5.2 \times \cos(6\pi 8 \times 10^3 t)$$

1. Tần số lấy mẫu tối thiểu của tín hiệu  $f_{min}$  là bao nhiêu?  
 2. Giả sử tín hiệu  $s(t)$  được lấy mẫu với tần số  $f_{min}$ , sau đó từng mẫu được lượng tử hoá (quantize) 8 bit/mẫu. Hỏi tốc độ bit của luồng dữ liệu (sau khi số hoá tín hiệu  $s(t)$ ) là bao nhiêu?  
 3. Giả sử tín hiệu  $s(t)$  được lấy mẫu với tần số  $f_{min}$  rồi mã hoá DPCM. Cụ thể là, cả luồng được chia thành từng frame 1500ms. Với mỗi frame mẫu đầu tiên được lượng tử hoá thành 8bit/mẫu, các mẫu tiếp theo được mã hoá thành hiệu giữa mẫu hiện tại với mẫu trước đó rồi lượng tử hoá 4bit/mẫu. Hỏi, tốc độ bit của luồng dữ liệu DPCM là bao nhiêu?

$$8$$

$$\frac{10^4 t + b}{r_{out}}$$

8000 Hz

8 bit/mẫu

48 ms/mẫu

**Bài 6**

Người ta sử dụng một cơ chế mã hoá FEC cho một chuỗi các gói đầu vào. Cứ  $n$  gói dữ liệu đầu vào (gói tin gốc - original packet) được mã hoá ra  $k$  gói dữ liệu mã hoá theo công thức dưới

$$X_n = A x_n$$

Trong đó:

- $x$ : vector (cột) các gói tin gốc (original packets)
- $X$ : vector (cột) các gói tin mã hoá (encoded packets)
- $A$ : Ma trận mã hoá và

$$X_1 \dots X_6$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$6 = x_1 + x_5 \quad x_2 + x_5 + x_6$$

$$n > k$$
  

$$n > m$$

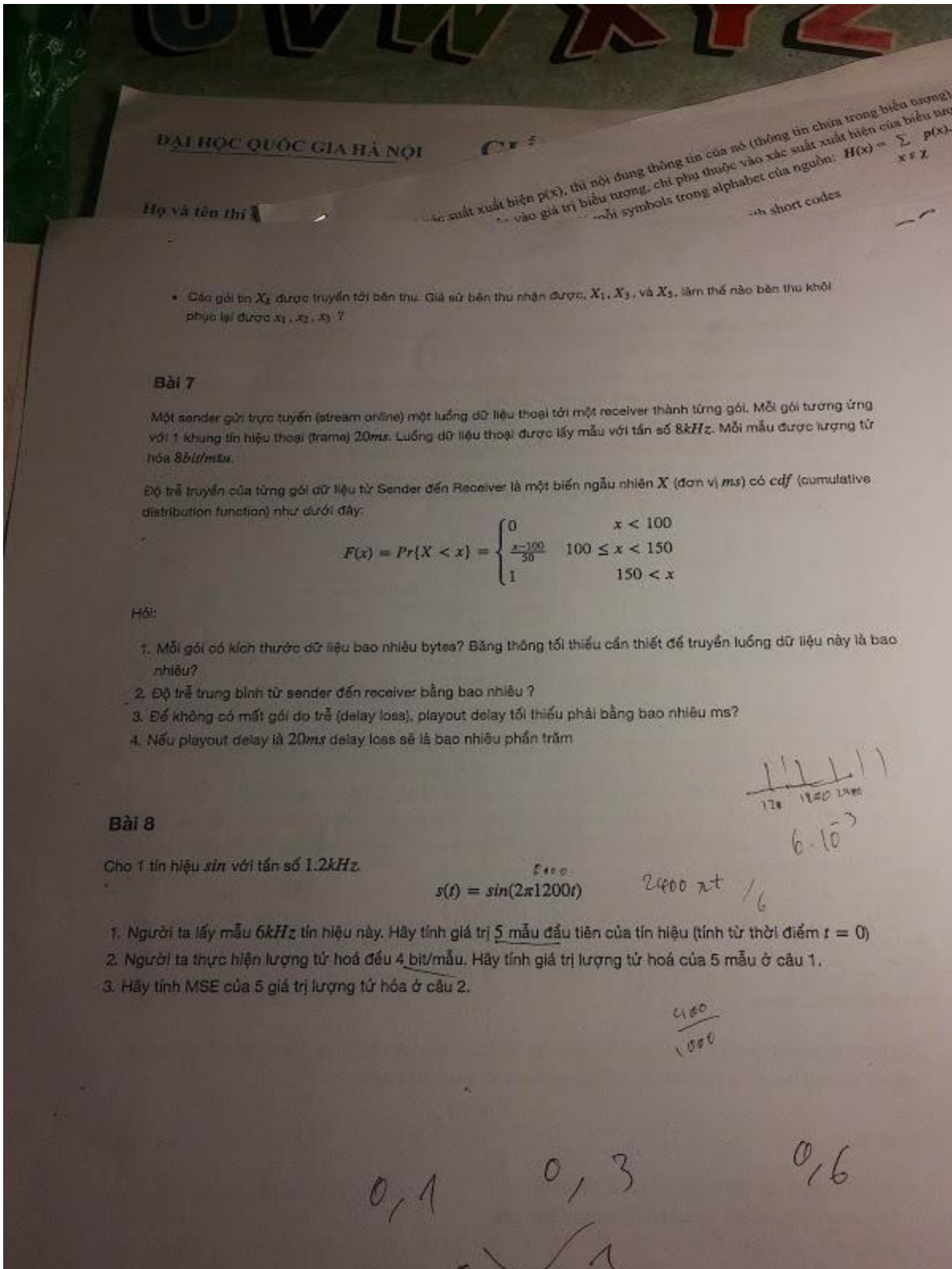
mẫu/s

8/

Câu hỏi:

- Với cơ chế mã hoá ở trên  $n$  bằng bao nhiêu?  $k$  bằng bao nhiêu?
- Viết công thức tính các gói mã hoá  $X_1, X_2, \dots, X_k$





Làm thử truyền thông đa phương tiện:

Thấy sai chỗ nào "Nhận xét" phát nhé: để chuột vào rồi bấm Ctrl + Alt + M

Status:

Bài	Loading	remaining
1	done	none
2	done	none
3	done	none
4	done	none
5	done	none
6	done	none
7	25%	2. 3. 4.
8	done	none

**Bài 1:**

Câu	Đúng/ Sai	Giải thích
1	Đúng	Entropy của dữ liệu đầu vào không đổi, nếu thuật toán nén cho output có entropy nhỏ hơn => lossy hoặc lỗi, ngược lại là lossless (entropy là lossless)
2	Đúng	input = 00 không thể biết chắc chắn là a hay c
3	Đúng	tất cả những gì liên quan đến việc tăng giảm dữ liệu đều có thể được coi là nén
4	Sai	IntServe yêu cầu thay đổi gần như toàn bộ hạ tầng mạng
5	Đúng	vì bản chất entropy coding đều là mã hóa để giảm dữ liệu nguồn.
6	Đúng	Entropy coding là lossless (đọc lại câu 1)
7	Sai	LZW đ cần
8	Sai	Packet classification (marking + classifying) + Traffic Policing + Scheduling
9	Đúng	Lý thuyết dạy thế

10	Sai	channel coding làm tăng dl gửi đi (add redundant), nhưng sửa lỗi tốt hơn
11	Sai	lớp ứng dụng
12		
13		
14	Sai	FEC là channel coding. Cứ cái gì có sửa lỗi là channel coding
15	Đúng	A high bandwidth-delay product is an important problem case in the design of protocols such as <a href="#">Transmission Control Protocol</a> (TCP) in respect of <a href="#">TCP tuning</a> , because the protocol can only achieve optimum throughput if a sender sends a sufficiently large quantity of data before being required to stop and wait until a confirming message is received from the receiver, acknowledging successful receipt of that data.

**Bài 2:**

step	w	k	wk	output	dictionary	DEC
1		L	L		ASCII	0->255
2	L	A	LA	L	+<LA>	256
3	A	L	AL	A	+<AL>	257
4	L	A	LA			
5	LA	L	LAL	LA	+<LAL>	258
6	L	%	L	L		

=> output: <L> <A> <LA> <L>

=> DEC: <76><65><256><76>

=> BIN: <001001100><001000001><100000000><001001100>

Các bước làm:

### Encoding Algorithm: Produce bit stream for stream of ASCII string

1. Load ASCII table into dictionary *dict*. Let  $w \leftarrow \text{NULL}$  // empty string
  2. Read a character  $k$
  3. If  $wk \in \text{dict}$ , then  $w \leftarrow wk$   
Else
    - ▶ *dict.push(wk)*
    - ▶ emit bit string (code)  $c$  where  $c \leftarrow \text{dict.getCode}(w)$
    - ▶ Let  $w \leftarrow k$
  4. repeat step 2 until no more symbol to read
- 

#### Bài 3:

1.

$$P(A) = 12/20 = 0.6$$

$$P(B) = 6/20 = 0.3$$

$$P(C) = 2/20 = 0.1$$

$$\Rightarrow I(A) = -\log(P(A)) = 0.736965594$$

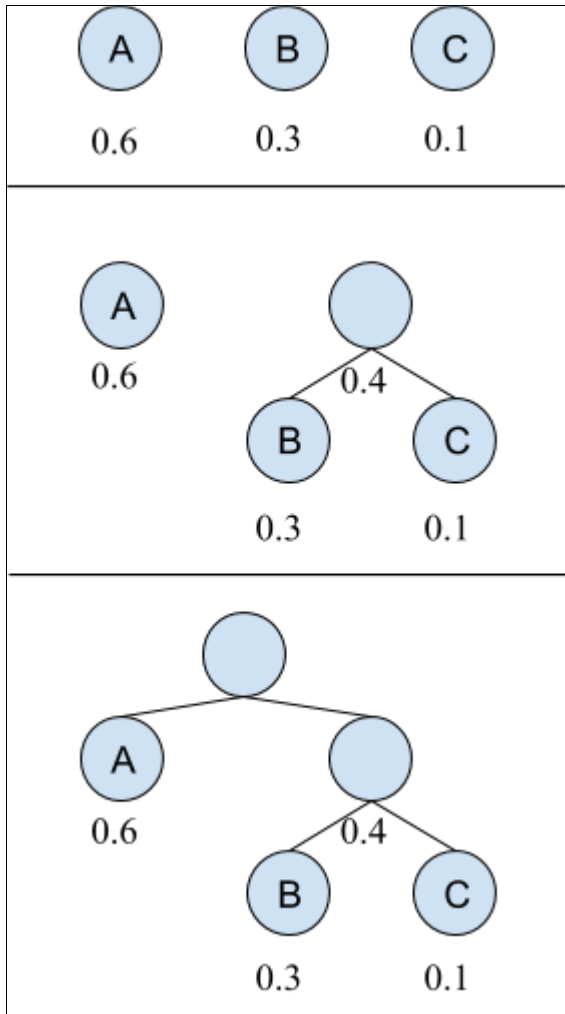
$$I(B) = -\log(P(B)) = 1.736965594$$

$$I(C) = -\log(P(C)) = 3.321928095$$

2.

$$e = I(A)*P(A) + I(B)*P(B) + I(C)*P(C) = 1.295461844$$

3.



4. (có phải là cái này k nhỉ?)

symbol	P	codeword
A	0.6	0
B	0.3	10
C	0.1	11

**Bài 4:**

1.

$r_{\text{token}} = 2000 \text{ tokens/s}$

bucket size  $b = 1000 \text{ tokens}$

kích thức queue  $q = \text{vô hạn}$

A.

với tốc độ luồng dữ liệu vào  $r_{\text{in}} = 3000 \text{ packets/s}$ :

$\Rightarrow$  tốc độ luồng dữ liệu ra chính bằng tốc độ nạp token:  $r_{\text{out}} = r_{\text{token}} = 2000 \text{ packet/s}$

B.



với tốc độ vào  $r_{in} = 1500 \text{ packets/s} < 2000 \Rightarrow$  tốc độ  $r_{out} = r_{in} = 1500 \text{ packets/s}$

2.

0.5s burst 5000 packets

1000 packets đầu tiên sẽ được đẩy đi ngay (do bucket đang chứa 1000 tokens)

$\Rightarrow$  còn 4000 packets cần xử lý

ngay lập tức token được nạp vào với tốc độ 2000 tokens/s

Do có 4000 packets nên cần 4000 tokens  $\Rightarrow$  thời gian để cấp đủ token là 2s

$\Rightarrow$  Tổng thời gian để đẩy hết burst là 2s

#### Bài 5:

1.

tần số lấy mẫu phải bằng ít nhất 2 lần  $f_{max}$

$f_{min} = f_{sampling} = 2 * f_{max} = 24 * 2 = 48 \text{ kHz}$

2.

$f_{min} = 48 \text{ kHz}$  có nghĩa là mỗi giây lấy mẫu 48000 lần

mỗi mẫu 8 bit

$\Rightarrow$  tốc độ luồng bit dữ liệu:  $48000 * 8 = 384000 \text{ bits/s}$

3.

frame 1500 ms = 1.5 s

$\Rightarrow$  tổng cộng tất cả là  $1.5 * f_{sampling} = 1.5 * 48000 = 72000$  mẫu

mẫu đầu tiên 8 bit, các mẫu sau 4 bit

$\Rightarrow$  tổng bit trong 1.5 s:  $8 + 71999 * 4 = 288004 \text{ bits}$

$\Rightarrow$  tốc độ luồng bit dữ liệu:  $288004 / 1.5 = 192002.67 \text{ bits/s}$

#### Bài 6:

1.

Dễ thấy:

$n = 4$

$k = 6$

2.

$X_1 = x_1$

$X_2 = x_2$

$X_3 = x_3$

$X_4 = x_4$

$X_5 = x_1 + x_2 + x_3$

$X_6 = x_2 + x_3 + x_4$

3.

nếu bên nhận chỉ nhận được  $X_1, X_3, X_5$

$\Rightarrow$  phương trình thu được:

$X_1 = x_1$

$X_3 = x_3$

$X_5 = x_1 + x_2 + x_3$

$\Rightarrow x_1 = X_1$

$x_3 = X_3$

$x_2 = X_5 - X_1 - X_3$

**Bài 7: <tự tin không thi>**

- 1 frame = 20 ms = 0.02 s  
 $f_{\text{sampling}} = 8\text{kHz} \Rightarrow T_{\text{sampling}} = 1/8000 \text{ s}$   
 $\Rightarrow$  mỗi frame được lấy mẫu  $0.02 \cdot 8000 = 160$  lần  
 $\Rightarrow$  mỗi frame tương đương với  $8 \cdot 160 = 1280$  bit  
 $\Rightarrow$  tốc độ truyền tối thiểu:  $1280/0.02 = 64000 \text{ bit/s}$

2.

**Bài 8:**

1.  
 $f_{\text{sampling}} = 6\text{kHz}$   
 $\Rightarrow T_{\text{sampling}} = 1/f_{\text{sampling}} = 1/6000$   
 $\Rightarrow$  5 mẫu đầu tiên được lấy ở thời điểm:  
 $t_1 = 0/6000 \text{ (s)} \Rightarrow s(t_1) = 0$   
 $t_2 = 1/6000 \text{ (s)} \Rightarrow s(t_2) = 0.951$   
 $t_3 = 2/6000 \text{ (s)} \Rightarrow s(t_3) = 0.588$   
 $t_4 = 3/6000 \text{ (s)} \Rightarrow s(t_4) = -0.588$   
 $t_5 = 4/6000 \text{ (s)} \Rightarrow s(t_5) = -0.951$

2.

Mỗi mẫu 4 bit:

Interval	Giá trị lượng tử	s(t1)	s(t2)	s(t3)	s(t4)	s(t5)
(0.875, 1.000)	0.9375		0.951			
(0.750, 0.875)	0.8125					
(0.625, 0.750)	0.6875					
(0.500, 0.625)	0.5625			0.588		
(0.375, 0.500)	0.4375					
(0.250, 0.375)	0.3125					
(0.125, 0.250)	0.1875					
(0.000, 0.125)	0.0625	0				
(-0.125, -0.000)	-0.0625					
(-0.250, -0.125)	-0.1875					
(-0.375, -0.250)	-0.3125					

(-0.500, -0.375)	-0.4375					
(-0.625,-0.500)	-0.5625				-0.588	
(-0.750,-0.625)	-0.6875					
(-0.875, -0.750)	-0.8125					
(-1.000, -0.875)	-0.9375					-0.951

3.

$$\text{MSE} = ((0.9375 - 0.951)^2 + (0.5625 - 0.588)^2 + (0 - 0.0625)^2 + \dots) / 5 = 0.00111425$$

Link sưu tầm:

[https://docs.google.com/document/u/0/d/1RGM9xfvfxiIH2HA\\_w2KRxPM6pwnzakE0IW\\_NzIX4I/mobilebasic?fbclid=IwAR3t0Gm3GoyelilLMuSLQFbb-V-WAtyrht-nTb\\_pmn9\\_k0IOppOpnUxoQg](https://docs.google.com/document/u/0/d/1RGM9xfvfxiIH2HA_w2KRxPM6pwnzakE0IW_NzIX4I/mobilebasic?fbclid=IwAR3t0Gm3GoyelilLMuSLQFbb-V-WAtyrht-nTb_pmn9_k0IOppOpnUxoQg)