

初級ネットワーク講座

コンピュータのデータが
きちんと届くまで

遠藤

(m-enndou@active-sita.com)

目次

1. デジタル情報はどこから来る？
2. データの先頭はどうやって見分ける？
3. データの間違いをどう見つける？
4. 間違った部分は送り直してもらう
5. 送り直しできない場合はどうする？
6. 途中であふれないように送るには？
7. 付録

デジタル情報はどう送られる？

- ◆ データの正体は0と1
- ◆ 電気信号に乗せて送る
- ◆ データとは0と1の集合である
- ◆ 0と1の信号を電気信号に変換して送信
- ◆ 電気のプラスとマイナスに0と1を割り当てる

データ(デジタル)の基礎(1)

- ◆ 0か1 … 2進数
- ◆ 2進数を4桁で括り、数値化 … 16進数
- ◆ 例 JISコード(キャラクタコード)の場合

英字A 2進数 01000001

 16進数 4 1

伝送時は10000010と後から

データ(デジタル)の基礎(2)

16進数	2進数	16進数	2進数
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

例題

「z」「t」の文字をJISコードで

「 」

送信される信号は

「 」

二つ目のビットから読み始める

「 」

文字では「 」



データの先頭はどうやって見分ける？

- ◆ データの先頭に目印として特殊な文字やパターンを置く
- ◆ ブロック同期の技術
- ◆ 読み始める位置が違くと意味が変わってしまう
- ◆ キャラクタ同期のしくみ
- ◆ フレーム同期のしくみ

同期のしくみ(1)

◆ **キャラクタ同期** JISコードのSYNキャラクタは「00010110」

◆ 「01101000」を送る 01101000を探そう！

◆ 11001101000011010000101111000101110

SYNじゃない

11001101000011010000101111000101110

SYNだね！

11001101000011010000101111000101110

これもSYNだね！

「z」

「t」

同期のしくみ(2)

◆ **フレーム同期** データの先頭は01111110(フラグ)が目印！

0111111000110100101010110100110001111110110011000010



1が6個並んだフラグのパターンが、フレーム中に発生することを防ぐため、1が連続して5個送られると、強制的に0をその後に挿入する。受信側では、データ部分で連続して5個の1が受信されるとその直後の0を無視する。

データの間違いをどう見つける？

- ◆ 検算用のデータを付けて送り正しく届いたか確かめる
- ◆ データが相手に正しく伝わらない理由は？
- ◆ ブロックに含まれる「1」の個数を数える
- ◆ 単純なパリティ・チェックはバーストに弱い
- ◆ 効率的に誤りを発見するCRC

エラー検出(1)

パリティチェック方式

偶数パリティ

パリティビット

データ	1000100	0
	1010001	1
	0101011	0
	1000100	0
パリティビット	1111010	1
	水平パリティ	
		垂直パリティ

エラー検出(2)

CRC方式



例

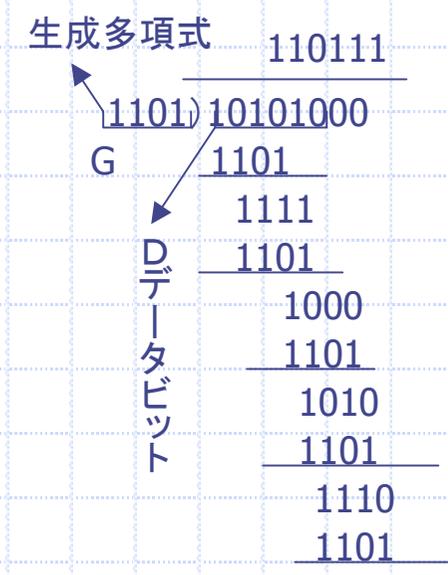
D=101010 (d=6)

G=1101 (x=3)

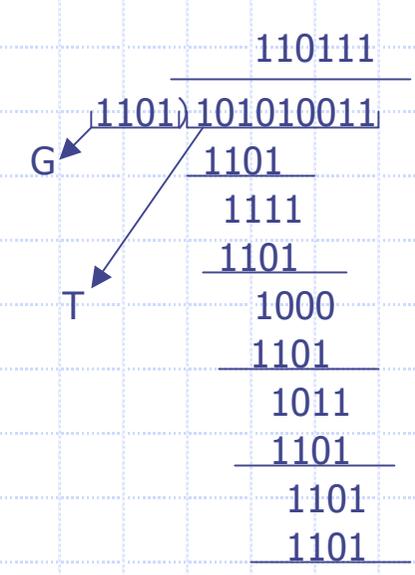
生成多項式 = $X^3 + X^2 + 1$

計算例

CRC生成



CRC検査



余り 0 ← 000

データOK D=101010

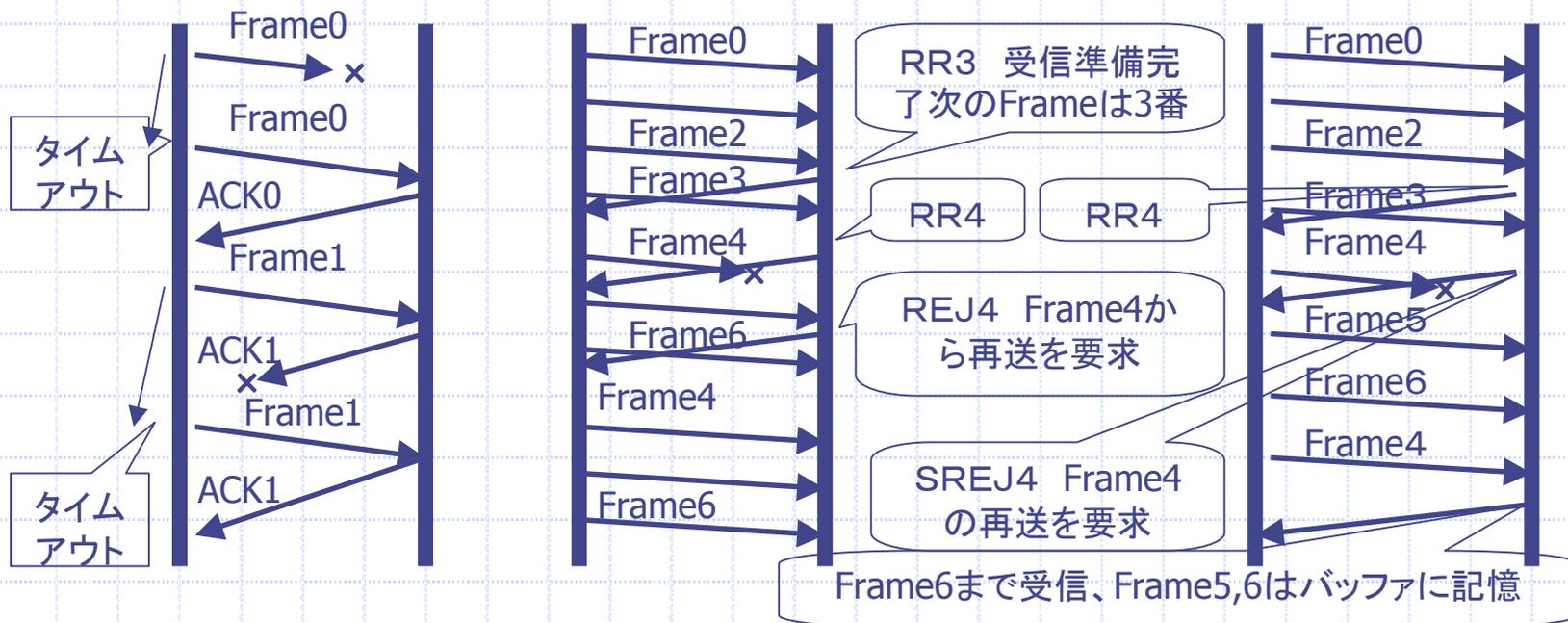
間違った部分は送り直してもらおう

- ◆ 送信側がフレームに番号をつけエラーのあるものだけを再送
- ◆ 再送してもらうには約束事が必要
- ◆ うまくいったときも相手に知らせる
- ◆ 一つずつ送ると効率が悪い
- ◆ 常にいくつかのフレームを送っておく

再送方式(エラー制御)

逐次確認方式

スライディング・ウィンドウ方式



Stop-and-wait ARQ

一括再送方式

選択的再送方式

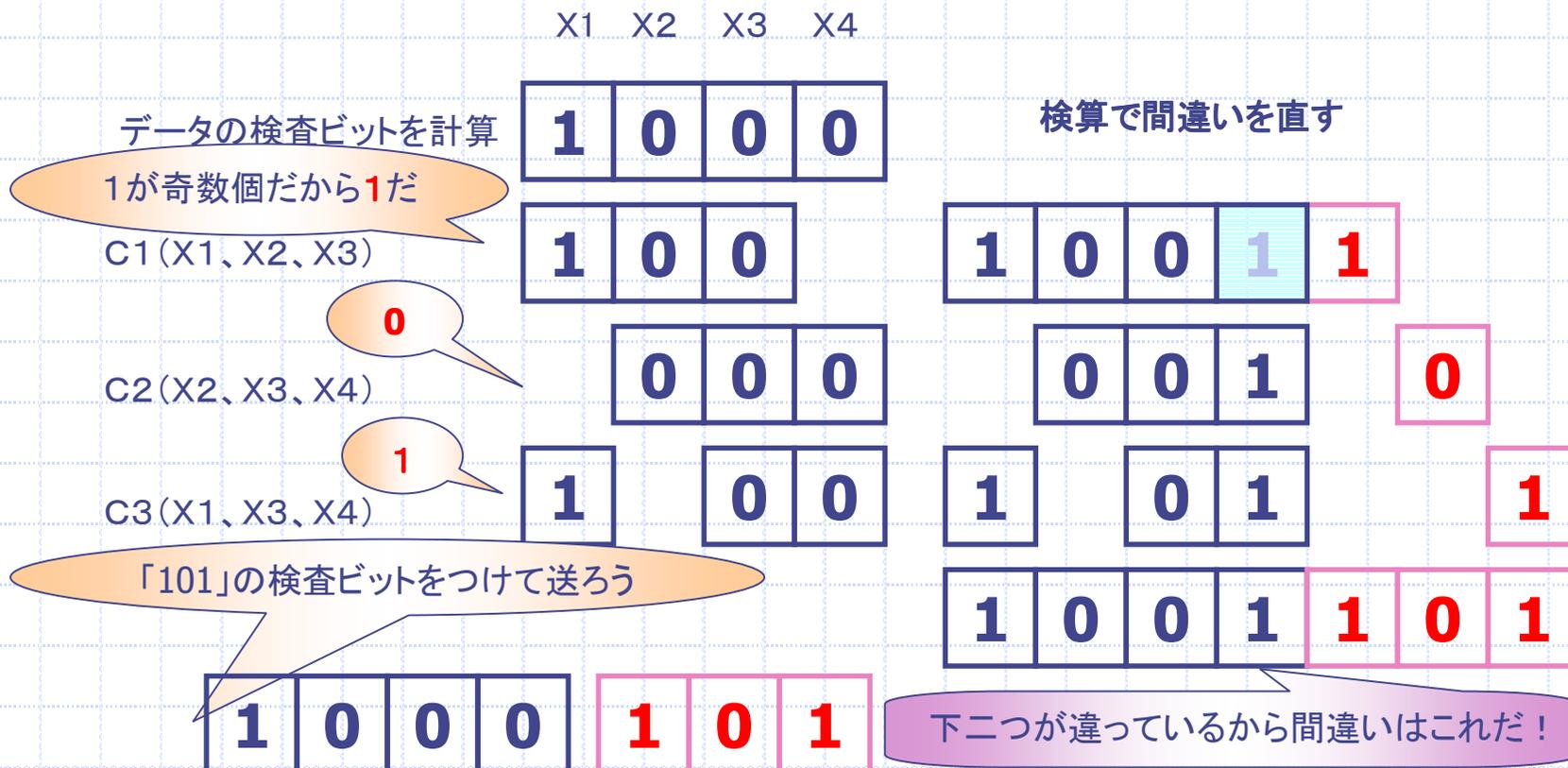
Sliding window ARQ

送り直しできない場合はどうする？

- ◆ 情報を付け加えて送り受け取り側で誤りを修正する
- ◆ 再送方式では間に合わない用途もある
- ◆ 誤りを回復するための情報をデータに付け加える
- ◆ 同じビットを連続して送り、多数決で決める
- ◆ 通信の効率が高いハミング符号
- ◆ 1ビットの誤りしか訂正できない

誤り訂正符号

ハミング符号



途中であふれないように送るは？

- ◆「受け取るよ」の合図を見て決められた量だけ送る
- ◆受け取れる量より多いデータが来るとあふれる
- ◆受信側の能力はどうやって決まる？
- ◆送信側の立場を優先したウィンドウ方式
- ◆受信側の立場が優先されるクレジット方式

付録 キャラクタコード表

上位4ビット→

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DE		0	@	P	`	p			-	タ	ミ			
1	SH	D1	!	1	A	Q	a	q			。	ア	チ	ム		
2	SX	D2	"	2	B	R	b	r			「	イ	ツ	メ		
3	EX	D3	#	3	C	S	c	s			」	ウ	テ	モ		
4	ET	D4	\$	4	D	T	d	t			、	エ	ト	ヤ		
5	EQ	NK	%	5	E	U	e	u			・	オ	ナ	ユ		
6	AK	SN	&	6	F	V	f	v			ラ	カ	ニ	ヨ		
7	BL	EB	'	7	G	W	g	w			ア	キ	ヌ	ラ		
8	BS	CN	(8	H	X	h	x			イ	ク	ネ	リ		
9	HT	EM)	9	I	Y	i	y			ウ	ケ	ノ	ル		
A	LF	SB	*	:	J	Z	j	z			エ	コ	ハ	レ		
B	HM	EC	+	:	K	[k	{			オ	サ	ヒ	ロ		
C	CL	→	,	<	L	¥	l	!			ヤ	シ	フ	ワ		
D	CR	←	-	=	M]	m	}			ユ	ス	ヘ	ン		
E	SO	↑	.	>	N	^	n	~			ヨ	セ	ホ	"		
F	SI	↓	/	?	O	_	o				ツ	ソ	マ	'		

←16進数

↑7≒164位下

漢字コードの第1バイト目 (8, 9)

漢字コードの第2バイト目 (A-F)

※00～1Fまでは制御用文字であり、スペース表示されるか、あるいは各コードの機能が働きます。上記コード表ではコードの意味を表記しています。

※81～8F、90～9F、E0～EF、F0～FCはシフトJIS漢字コードの第1バイト目を示します。漢字用コードの第2バイト目は、40～7E、80～FCのコードで必ず第1バイト目と対になっている必要があります。

コード	符号	意味
00		Null
01	SH (SOH)	Start of Heading
02	SX (STX)	Start of Text
03	EX (ETX)	End of Text
04	ET (EOT)	End of Transmission
05	EQ (ENQ)	Enquiry
06	AK (ACK)	Acknowledge
07	BL (BEL)	Bell
08	BS	Backspace
09	HT	Horizontal Tabulation
0A	LF	Line Feed
0B	HM	Home
0C	CL	Clear
0D	CR	Carriage Return
0E	SO	Shift Out
0F	SI	Shift In
10	DE (DLE)	Data Link Escape
11	D1 (DC1)	Device Control 1
12	D2 (DC2)	Device Control 2
13	D3 (DC3)	Device Control 3
14	D4 (DC4)	Device Control 4
15	NK (NAK)	Negative Acknowledge
16	SN (SYN)	Synchronous Idle
17	EB (ETB)	End of Transmission Block
18	CN (CAN)	Cancel
19	EM	End of Medium
1A	SB (SUB)	Substitute Character
1B	EC (ESC)	Escape
1C	→	
1D	←	
1E	↑	
1F	↓	

(詳細についてはJIS C6220を参照してください)



参考資料

- ◆ 日経NETWORK
- ◆ TOSHIBA(LAPTOP) オペレーションガイドより
- ◆ 通信とネットワークの
基礎知識 森本喜一郎著 (昭晃堂)