

第十一章

誤り制御技術の基礎

通信システムに要求されること

高速なデータ伝送

- 扱うデータによって最低限必要なデータレートが決まる
- 高速伝送 ➡ 広帯域伝送／多値変調方式が必要

帯域幅

- 運用しようとするシステムの規格によって厳密に規定

品質の高い通信（良好なBER特性）

- 扱うデータの種類によって最低限必要なBERが決まる
- 低BER ➡ 高電力が必要

空中線電力

- 安全性、バッテリーの容量によって制限

交換方式

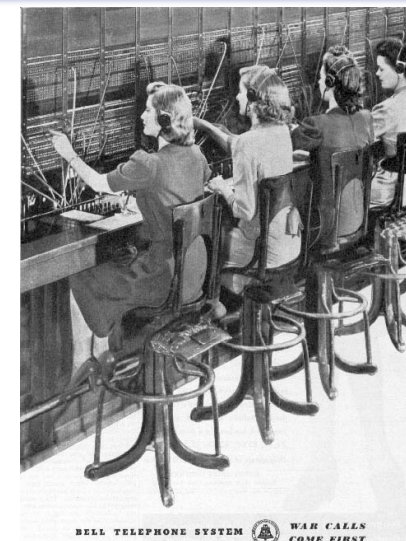


- 通信を始める前に回線を確保
- 一度確保した回線は通信が終了するまで独占
- 情報が間欠的に発生する場合に、情報が流れていなくても回線を占有するため利用効率悪い
- リアルタイム性を優先するため、多少の誤りは許容



- データをパケットと呼ばれる単位に分割して伝送
- パケット単位の通信のため、複数のユーザが回線を共有可
- リアルタイムに回線を確保できる保証がない
- データの確実性を優先するため、多少の遅延は許容

初期の回線交換の様子



メールサービスの違い

📞 ショートメール／SMS (DoCoMo), SkyMail／SMS (SBM, 旧Vodafone), Cメール (au)

- 携帯電話同士で短い文字メッセージを送受信できるサービス（入力方法もポケベル方式が主流）
- 事業者ごとに規格が異なるため、他の事業者の携帯にメッセージは送れない
- 送信には課金されるが、受信は無料

📞 Eメール

- インターネットを通じてメッセージを送るサービス
- 異なる事業者でも、パソコン相手にでも送れる
- 送信も受信も課金（パケット代）

誤り制御方式

📞 誤り訂正 (forward error control/correction: FEC) 方式

📞 誤りを検出し、訂正する能力のある誤り訂正符号を用いて **情報を符号化**して伝送

📞 受信局は復号によって、**誤り検出と訂正**を行う

📞 比較的遅延の少ない伝送が可能／訂正能力には限界有り

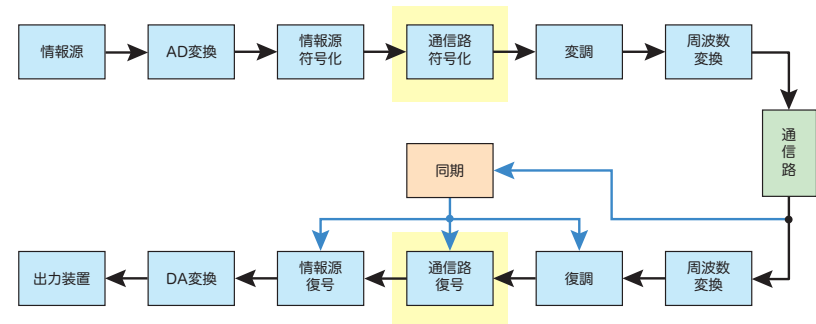
📞 誤り検出再送要求 (automatic repeat request: ARQ) 方式

📞 情報は誤り検出能力のある符号で**情報を符号化**して伝送

📞 受信局は**誤り検出**を行い、受信情報が正しい場合には、(acknowledge) を返送、誤りがあった場合には、(negative acknowledge) 返送し、再送を要求

📞 通信路の状態によっては大幅な遅延が発生

デジタル通信システム



通信路符号化

📞 誤り検出／訂正のための符号化

📞 元の情報系列に冗長（ビット）を付加する

📞 符号化率 r

📞 $r = \frac{k}{n}$ （情報系列 k ビット，符号化後の系列 n ビット）

📞 r が 1 に近く，検出・訂正能力の高い符号化が望ましい

📞 実質の伝送レートが k/n に減少する

📞 符号化率 1/2 の符号化データを 96kbps で伝送

📞 実質の情報レートは 48kbps

繰り返し符号 (repetition code)

9

3重繰り返し符号の例

情報源	1	0	1	1	2値情報系列
符号化器出力	111	000	111	111	符号語, 符号化器は"111"または"000"を出力する
通信路出力	110	110	011	000	通信路を通して受信された符号語
誤り検出	有	有	有	無	比較による誤り検出
誤り訂正	1	1	1	0	多数決原理による誤り訂正



ハミング距離 (Hamming distance)

10

ハミング距離 d_H

二つの同一の長さのバイナリ符号 (符号語) において, 値が異なる桁の数 (ビットの数)

- 符号語 (1, 0, 1), (1, 1, 0) のハミング距離は
- グレイ符号の隣り合う符号語のハミング距離は

最小ハミング距離 d_{min}

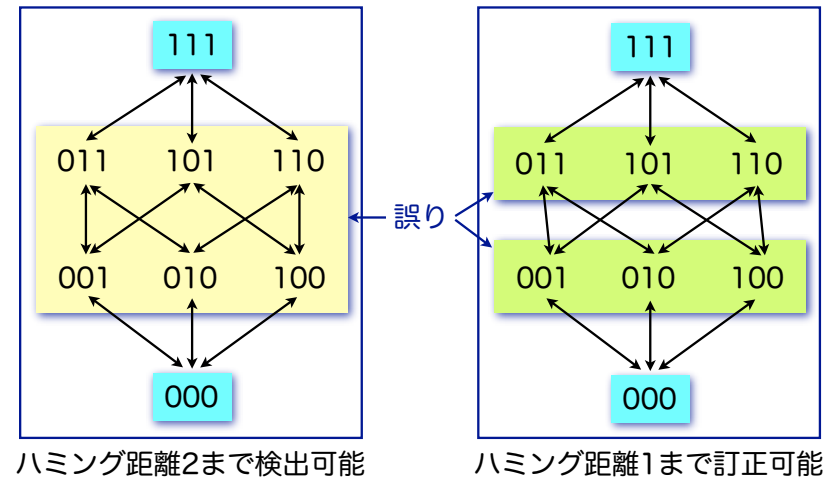
システムで利用できる全ての符号語の対を考えたときの, ハミング距離の最小値

- QPSKシンボルの最小ハミング距離は
- 3重繰り返し符号の最小ハミング距離は



誤り検出能力と誤り訂正能力

11



検出・訂正可能なビット数

12

誤り検出可能なビット数 $d_{min} - 1$

誤り訂正可能なビット数 $\left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$

ただし, $\lfloor x \rfloor$ は x 以下の最大の整数

例) $\lfloor 3.8 \rfloor = 3$

$\lfloor 5 \rfloor = 5$



パリティ検査符号 (parity check code)

13

🕒 単一誤り検出法 1ビットの誤りを検出可能 $d_{min} = 2$

🕒 検査符号を1ビットつける ➡ $n = k + 1$

🕒 奇数パリティ: 「1」の数が奇数になるようにする

🕒 偶数パリティ: 「1」の数が偶数になるようにする

偶数パリティ

情報	パリティ
00	0
01	1
10	1
11	0

靴下が片方足りないわ?



Takahiko Saba

Dept. of Computer Science, C.I.T.



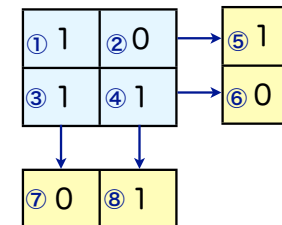
誤り訂正能力を持たせるには?

15

偶数パリティを2次元に配置 ➡ 積符号 (product code)

$d_{min} = 3$ 1個の誤りを訂正可能

1 0 1 1 1 0 0 1



2次元パリティ検査符号

青の靴下が片方足りないわ?



Takahiko Saba

Dept. of Computer Science, C.I.T.



パリティ検査符号 (cont'd)

14

例) 偶数パリティによって情報系列3ビットに対し
1ビットのパリティを付加 (符号化率 = 3/4)

情報系列 1 0 0 1 1 0 1 1 1 ...

符号化系列 1 0 0 1 1 1 0 0 1 1 1 1 ...

伝送路出力 1 0 1 1 1 1 0 0 1 0 0 1 ...

誤り検出

有

無

無

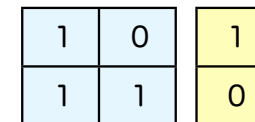
Takahiko Saba

Dept. of Computer Science, C.I.T.

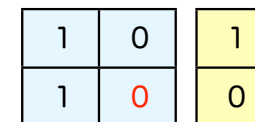


2次元パリティ検査符号

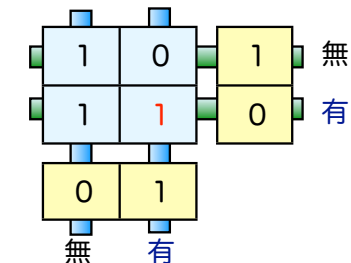
16



送信系列



受信系列



誤り検出/訂正

Takahiko Saba

Dept. of Computer Science, C.I.T.

