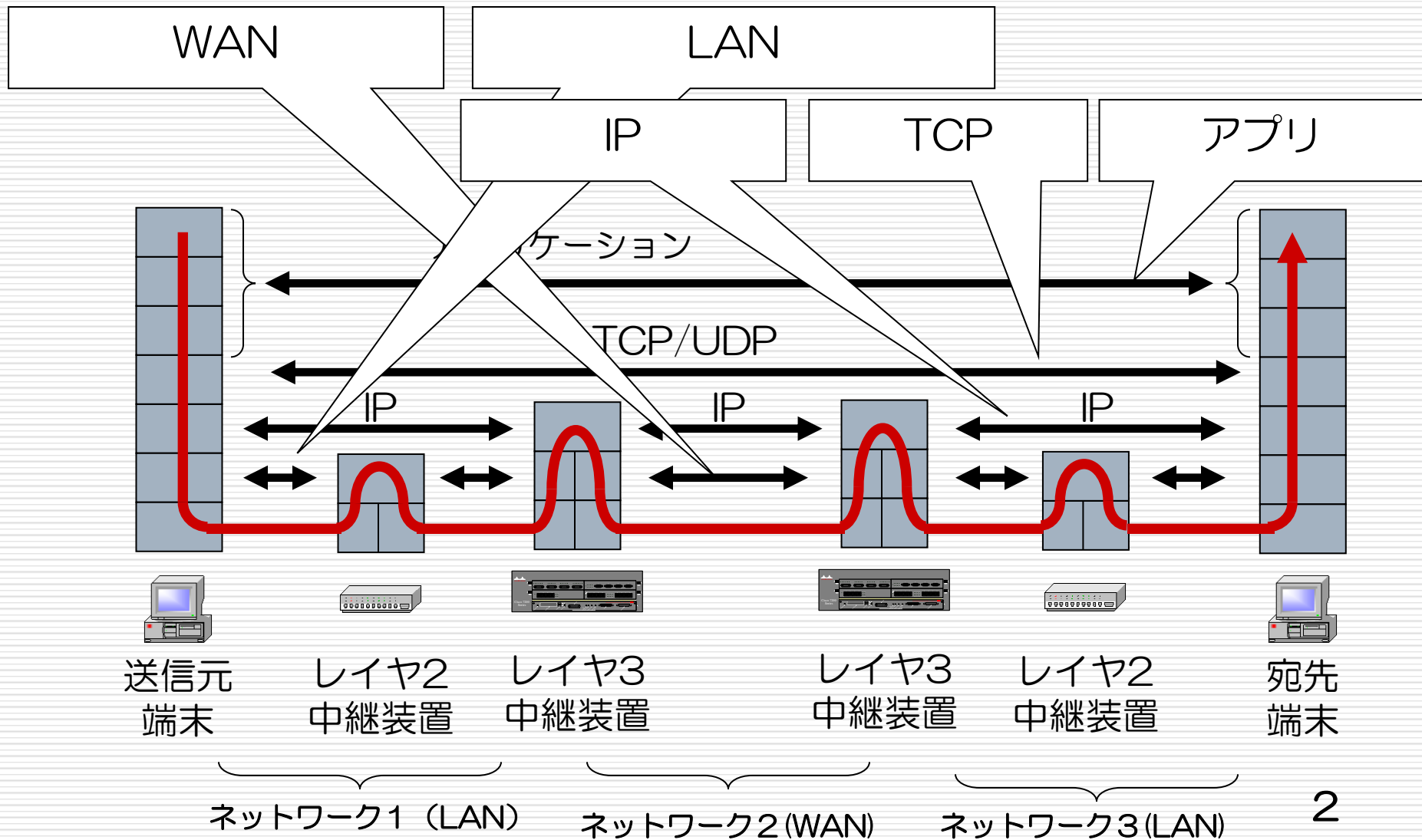

ネットワーク技術の基礎

データリンク制御

データリンク



データリンク層とは

- 同一の伝送媒体に対して，隣接するシステム（ノード間）で媒体に適した通信方法による通信路を提供
- 主要な機能：
 - フレーム同期：パケットやキャラクターの識別
 - データリンクの設定・開放
 - 順序制御：フレームに順序記号をつけて，正しい順序を相手に伝える。
 - エラー制御：フレームにエラーチェック用の符号をつけて，送信途中にエラーが混入していないかを相手側にチェックさせる。
 - フロー制御：相手側が受信できる状態であることを確認して，フレームを送信する。



データリンクの歴史

- 1970年代
 - 公衆データ回線 X.25 (HDLC)
- 1980年代
 - LANの普及：Ethernet, Token Ring, FDDI
- 1990年代
 - LANトラヒックの公衆データ回線上での収容
 - ISDN、ATM、ADSL、フレームリレー
 - 無線技術の普及 Wireless LAN (IEEE802.11)
 - LAN技術の広帯域 Fast Ethernet, Gigabit Ethernet
- 2000年代
 - LANトラヒックの公衆データ回線上での収容
 - ATM, 広域LAN, FTTHの普及
 - 無線技術の普及 Wireless LAN (IEEE802.11)
 - 携帯電話ネットワークの利用
 - 仮想化技術の普及

フレーム

- データリンク層での処理単位
 - 物理層はビット誤りを発生
 - データブロックに対して冗長な情報を付加し、誤りを検出・訂正
 - フレームの境界を定義
- フレームヘッダ
 - 誤りを検出・訂正のための情報
 - 制御情報

例：HDLC



F (Flag) : パケットの識別

A (Address) : 多重通信する際の識別用

C (Control) : フレーム種別、コマンド/レスポンス

I (Information) : 情報

FCS (Frame Check Sequence) : 誤り検出用のCRC符号

誤り検出の対象はA/C/Iフィールド。2バイトまたは4
バイト（最大フレーム長による）

フレームの取り出し

- キャラクタベース
 - 物理層としては文字を単位としたデータ伝送
 - データ転送の最初と最後に特別な文字の挿入。STX, ETX
 - DLE (DataLink Escape) によって特別な文字をデータと識別
 - バイナリデータの転送では特別な文字がパターン出現する可能性有
 - character stuffing
- ビット列ベース
 - ビット列をベースとしたフレームの取り出し
 - 特定のビット列をデータの最初と最後に挿入
 - データ中に現bit stuffing：特定のビット列がフレームの最初と最後だけに常に存在するように「同じ」ビット列を細工
 - 例：
 - 特別なビット列：01111110
 - 送信側ではデータ中に5ビット“1”が続いたら“0”を挿入
 - 受信側ではデータ中に5ビット“1”が続いたら次の“0”を削除

誤り制御方式

- 誤りとは
 - 伝送路における電氣的な雑音等により内容が変化
 - 中継ノードでバッファあふれ(輻輳, congestion)が発生し, 情報全体が消失
- デジタル通信路の品質はビット誤り率BER (Bit Error Rate)で表わす
 - $BER = r/n$ (n:送信ビット、r:誤りビット)
 - $BER(\text{移動無線}) \div 10^{-2}$ 、 $BER(\text{光ファイバ}) \div 10^{-9}$
- 単に誤り率だけでなく、実際には誤りの発生パターンも問題
- 誤りパターンによる分類
 - ランダム誤り (random error) : 伝送ビット中でバラバラに発生する。波形の歪みや雑音、記録媒体では製造誤差など。
 - バースト誤り (burst error) : 一部に連続して発生する。無線におけるフェージング、同期ずれ、パルス性雑音、記録媒体では傷など

誤り制御方式

- 誤り制御方式の目的
 - 通信路上での伝送誤りを発見・訂正
 - 送信フレームが正しく送られているかどうか
 - 送信フレームの順番が正しいかどうか
- 大きくは2種類
 - 符号技術を用いた誤り訂正と検出
 - 誤り訂正符号(Error Correction Code)
 - 誤り検出符号(Error Detection Code)
 - 再送制御：ARQ (Automatic Repeat Request)

誤り制御方式の分類

- 符号技術を用いた誤り訂正と検出
 - 受信側で付加ビットをチェックし、誤り検出と誤り訂正
 - 冗長な情報が多くなり、複雑な符号器が必要
 - 誤り率が大きな伝送路や伝播遅延が大きな極めて大きな伝送路に適用（衛星、無線）
- ARQ再送制御
 - 誤りパケットに誤り検出符号を付加して誤りを検出し、伝送中に誤りの生じたパケットは再度送信
 - 付加される情報量が小さい
 - 送受信回路が簡単
 - パケット誤り率が十分小さいことが条件
 - ほとんどのコンピュータネットワークで使用

符号技術：(n,k)符号

- (n,k)符号※：一般に、誤り検出・訂正符号は、情報を表すkビット（桁）に余分のmビットを追加して、全体 $n=k+m$ ビットの符号にして送信。
- 情報を表すkビットを情報ビット、追加した（冗長な） $m=(n-k)$ ビットを検査ビット
- 情報を表すkビットに対して、 $m=n-k$ ビットを適当に決めることによって、誤りの検出や訂正を可能に

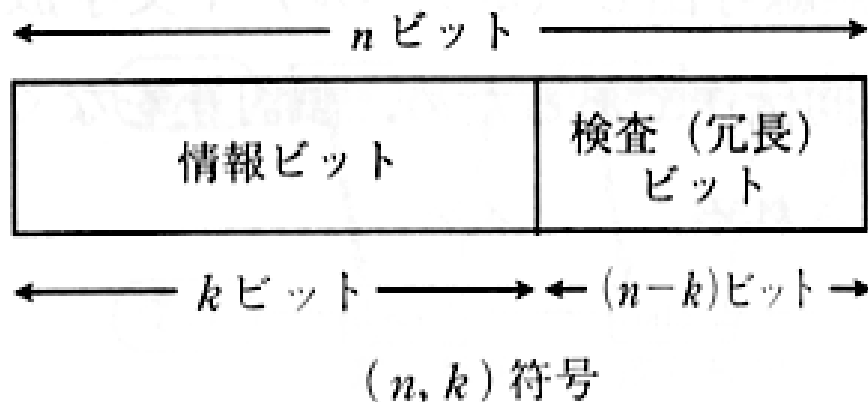


図 5.2 通信路符号の構成

※情報を表すkビットをそのまま符号語の一部とすることに本質的な意味はない。ただ、そのほうが符号化が簡単であり、そうでない場合に比べて符号の性能が劣ることはない。

符号技術：符号語と非符号語

- 全体的には $\eta = k/n$ の大きさに、誤り・訂正能力は制限

$$\eta = \frac{k}{n}, \quad \rho = 1 - \eta$$

- η を符号化効率、 $\rho = 1 - \eta$ を冗長度
- 誤り検出・訂正能力を高くする（信頼性を上げる）ためには、ある程度 η を小さく（ ρ を大きく）することが必要
- 符号化効率を下げないでどれだけ信頼性を上げることができるかが、通信路符号化の課題

符号技術：パリティチェック符号による誤り検出

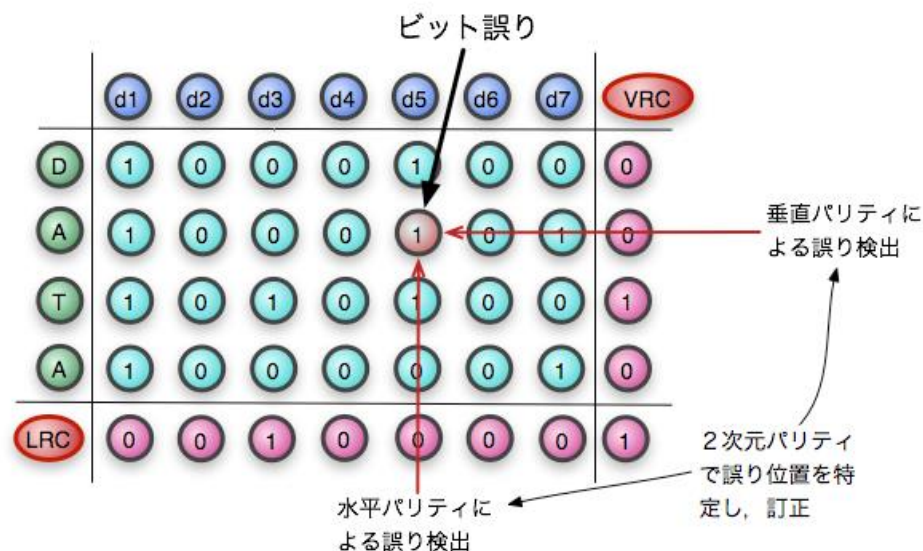
- 情報ビットの数：k

$$x_0 + x_1 + x_2 + \cdots + x_k = 0$$

- 検査ビット：x0 → 上記の等式が成り立つように付与
- 誤りの検出：通信路における1ブロック中の1の数が奇数になる

- 水平・垂直パリティ検査符号

- 情報ビットをブロックに分割して2次元に並べ、垂直および水平方向のビット列に対してパリティビットを付加する符号
- 誤り箇所を検出



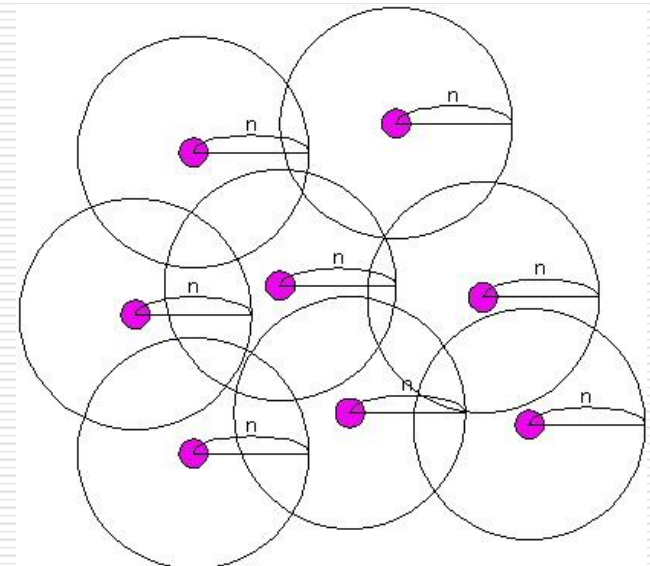
符号技術：ハミング符号による誤り検出・訂正

□ ハミング距離

- 等しい文字数を持つ二つの文字列の中で、対応する位置にある異なった文字の個数
- 伝送路を流れる二つの文字がどれだけ似ているかを表す
- 例：1011101 と 1001001 の間のハミング距離は 2
- 例：2143896 と 2233796 の間のハミング距離は 3
- 文字数 n の1ビット文字列間のハミング距離は、それらの文字列間の排他的論理和のハミング重み（文字列内の 1 の個数）

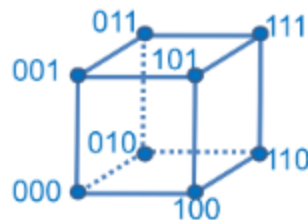
□ 誤り検出

- この体系内の任意の符号間のハミング距離は n 以上
- 誤って狂ってしまったビット数が n より少ない場合、誤りが検出できる

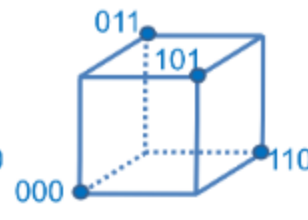


符号技術：ハミング符号

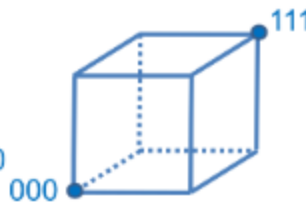
- 誤り検出・訂正用のビットを付加
- ハミング距離が3以上
 - 2ビットまでの誤りは検出できる
 - 1ビットの誤りは訂正できる
- ハミング距離の例
 - どちらの方向に誤っているかで検出・訂正可能



ハミング距離 1



ハミング距離 2



ハミング距離 3

	x_1	x_2	x_3	x_4	y_1	y_2	y_3
0 →	0	0	0	0	0	0	0
1 →	0	0	0	1	0	1	1
2 →	0	0	1	0	1	0	1
3 →	0	0	1	1	1	1	0
4 →	0	1	0	0	1	1	0
5 →	0	1	0	1	1	0	1
6 →	0	1	1	0	0	1	1
7 →	0	1	1	1	0	0	0
8 →	1	0	0	0	1	1	1
9 →	1	0	0	1	1	0	0
A →	1	0	1	0	0	1	0
B →	1	0	1	1	0	0	1
C →	1	1	0	0	0	0	1
D →	1	1	0	1	0	1	0
E →	1	1	1	0	1	0	0
F →	1	1	1	1	1	1	1

符号技術：巡回符号での誤り検出

- 巡回符号：cyclic code あるいは cyclic redundancy code (CRC)
 - ある線形符号の符号語を巡回シフトしたのもまた符号語であるもの
- 特徴
 - 機械的な代数演算で処理が可能で、ハードウェア実現が容易
 - バースト誤りに強い
 - 実用上、広く利用
- 各コンピュータが共通の「生成多項式」を持ち、CRCを生成
 - 生成多項式の例： $G(x)=x^3+1$

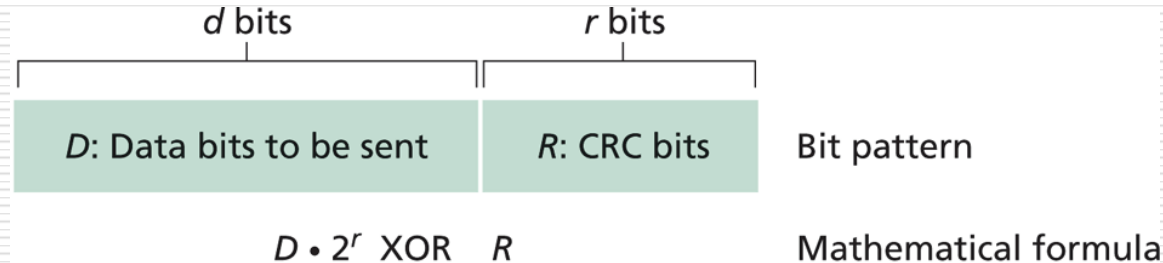


Figure 5.7 ♦ CRC codes

符号技術：巡回符号の生成法

- dビットの情報 ($a_{d-1}, a_{d-2}, \dots, a_0$) から $d+r-1$ 次の多項式 $D(x)$ を次のように生成

$$D(x) = a_{d-1}x^{d+r-1} + a_{d-2}x^{d+r-2} + \dots + a_0x^r$$

例: (1,0,1,1,1,0) かつ $n=6, r=3$ の時, $D(x) = x^8 + x^6 + x^5 + x^4$

- r 次の「生成多項式」 $G(x)$ で $D(x)$ を割り余り $R(x)$ を求める.

$$G(x) = x^3 + 1 \text{ のとき, } D(x) = (x^5 + x^3 + x + 1)G(x) + x + 1$$

つまり, $R(x) = x + 1$

- チェックビットを ($b_{r-1}, b_{r-2}, \dots, b_0$) で与える. ここで,
 $R(x) = b_{r-1}x^{r-1} + \dots + b_0$

上の例: (0,1,1)

符号技術：巡回符号の生成法

□ 2を法(mod 2)とする演算

$0+0=0$, $0+1=1$, $1+0=1$, $1+1=0$

$0-0=0$, $0-1=1$, $1-0=1$, $1-1=0$

$0\times 0=0$, $0\times 1=0$, $1\times 0=0$, $1\times 1=1$

二進数の計算.

繰り上がりナシ

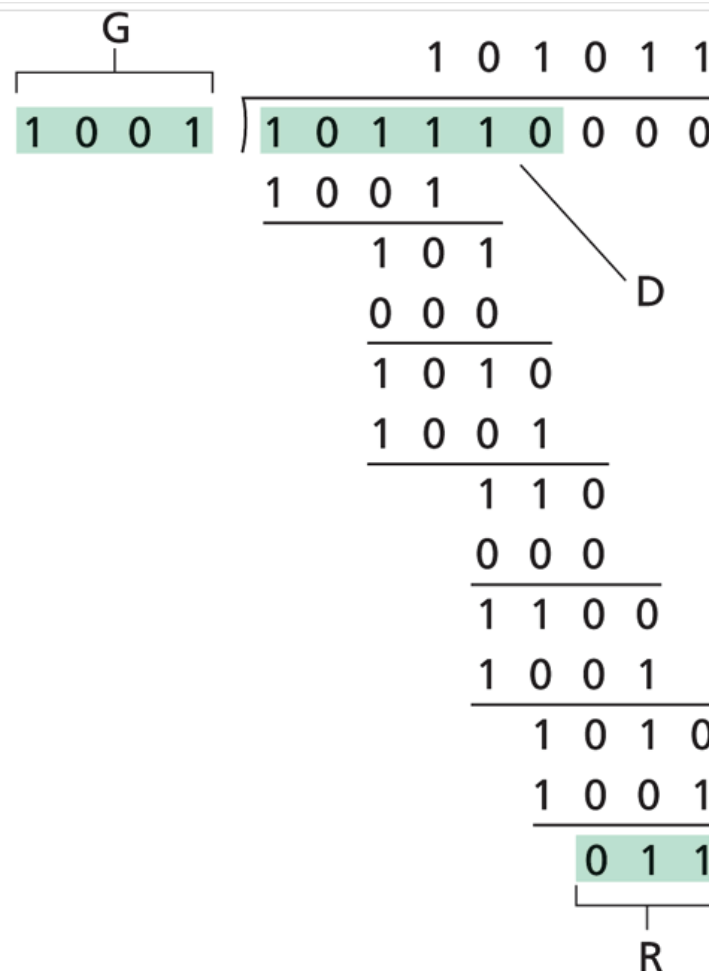


Figure 5.8 ♦ A sample CRC calculation

符号技術：巡回符号による誤りの検出手順

- 送信側は情報 $(a_{n-1}, a_{n-2}, \dots, a_0)$ を多項式にして, $G(x)$ で割る, あまり $(b_{r-1}, b_{r-2}, \dots, b_0)$ を後につけて送信する.

- 受信した $n+r$ ビットの情報 $(a_{n-1}, a_{n-2}, \dots, a_0, b_{r-1}, b_{r-2}, \dots, b_0)$ から $n+r-1$ 次の多項式 $D^*(x)$ を次のように生成する.

$$D^*(x) = a_{n-1}x^{n+r-1} + \dots + a_0x^r + b_{r-1}x^{r-1} + \dots + b_0$$

- 受信側は, $G(x)$ で $D^*(x)$ を割り, 余りが生じれば, 伝送誤りが検出されたことになる.

CRCで使う生成多項式

- CRCの生成多項式は任意の数なので、具体的なCRCは無数に存在
⇒ 生成多項式の桁数が比較的短くて、誤り制御能力が高いものを利用

- 代表的な生成多項式

名 称	生成多項式	主な用途	CRC長
CRC-12	$x^{12}+x^{11}+x^3+x^2+x^1+1$	6bit通信データの誤り検出	12bit
CRC-16	$x^{16}+x^{15}+x^2+x^1+1$	8bit通信データの誤り検出	16bit
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1$	FCS-32 (PPP)	32bit
CRC-ITU-T	$x^{16}+x^{12}+x^5+1$	8bitデータの誤り検出 (FDC)	16bit

再送制御方式

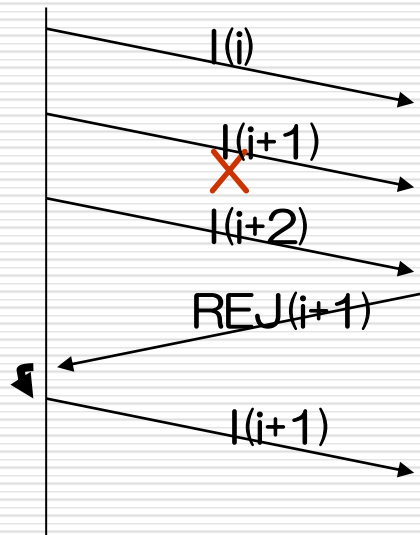
- ARQ (Automatic Repeat reQuest)
- 誤り制御
 - 受信側でフレームに誤りを発見
 - 送信者側にもう一度フレームを送り直してもらう
- 代表的な方式
 - Stop-and-wait
 - Go-back-N
 - Selective-repeat

再送メカニズム

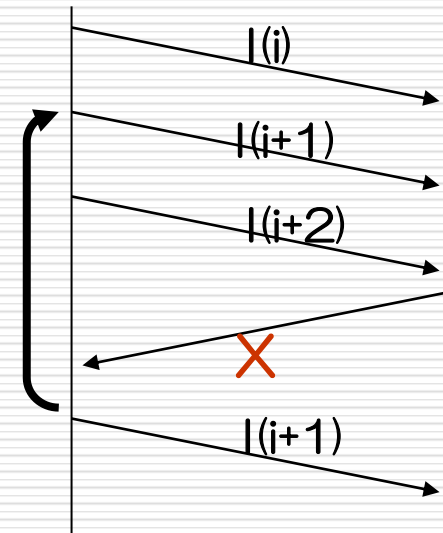
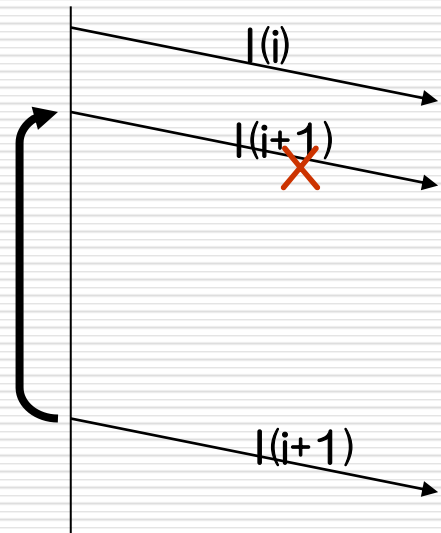
- 誤りを含むパケット (X) を受信したノードは、単に廃棄するのみ
- 2種類の再送方法
 - 受信フレームのシーケンス不一致による受信ノードからの再送要求(ACKあるいはNACK)による再送
 - 送信側でのタイムアウトによる再送

acknowledgement, ACK

negative-acknowledgement : NACK

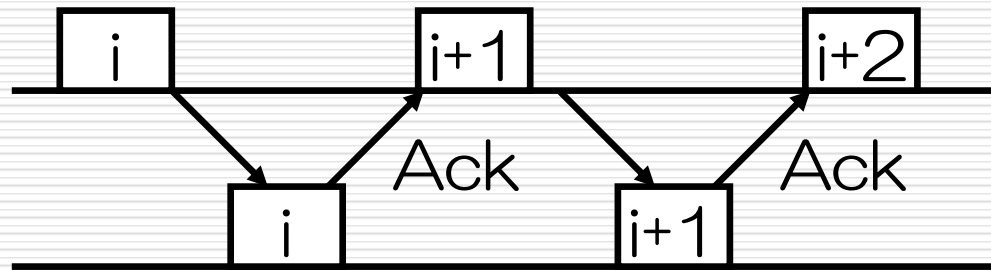


受信シーケンス不一致
による再送



タイムアウトによる再送

Stop-and-wait



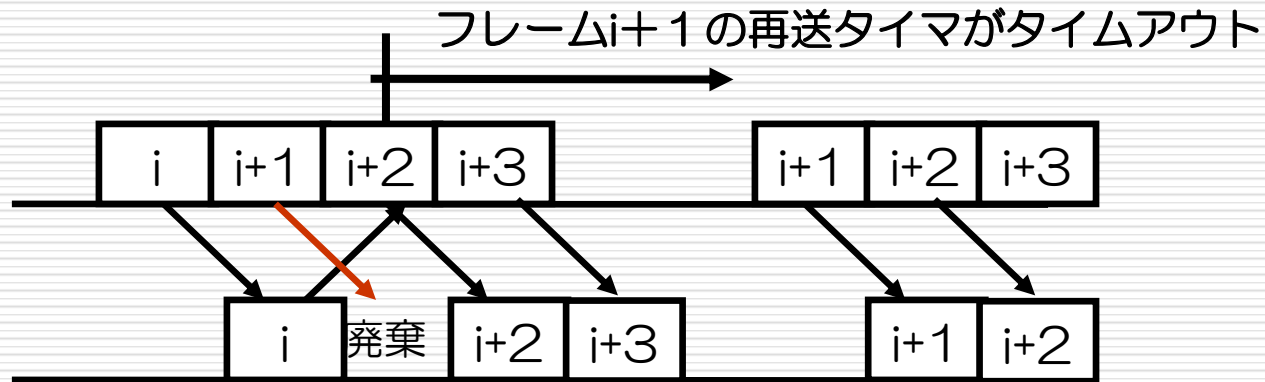
■ 方式

- 送信局に対してフレームを一つ送信すること、ACKが返送されるまで次のブロックを送信せずに待機
- 送信側タイマがタイムアウトするとフレームを再送

■ 特徴

- バッファは送受信側それぞれ1フレーム
- 制御が簡単
- 1ブロック伝送するのに往復の伝送が必要

Go-back-N



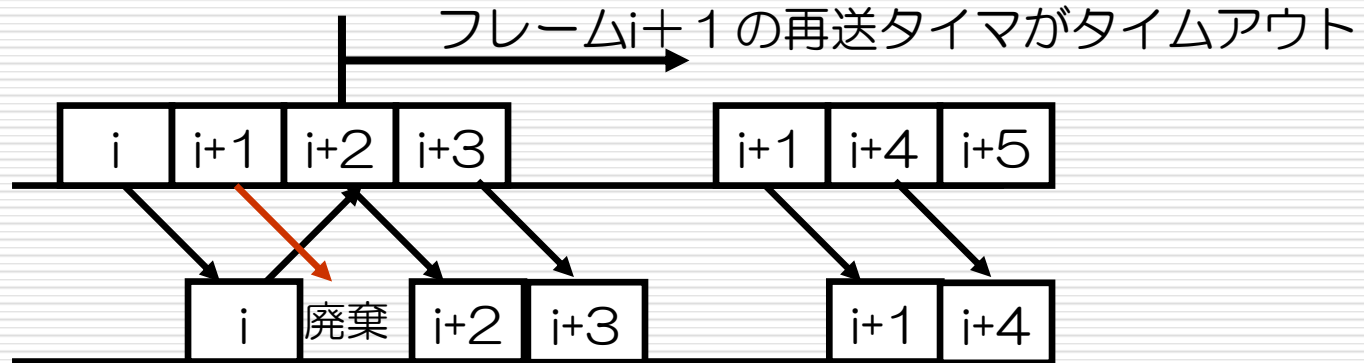
■ 方式

- 連続してN個のフレームを送出
- 誤りが発生した場合
 - 誤ったフレームを正しく受信するまでフレームを破棄
 - 正しい受信までACKを返さない

■ 特徴

- 正しく送られたフレームでも破棄され、無駄
- 送信側で再送のためのバッファが必要
 - ACKの未返送フレームは再送のためにバッファリング
- バッファは再送タイマの値に依存

Selective-repeat



■ 方式

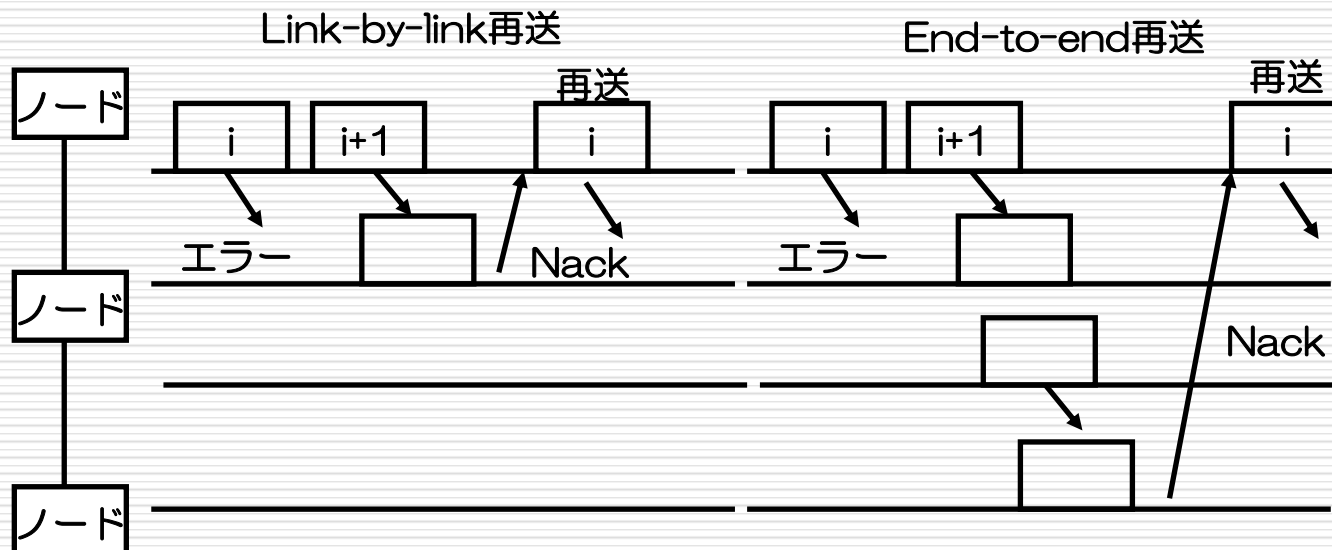
- Go-back-N プロトコルを改良
- 必要なフレームだけを再送

■ 特徴

- 伝送効率はGo-back-N よりも良い
 - 正しく受信されているフレームを無駄に破棄しない
- 複雑な制御
 - 受信側でバッファが必要
 - 順番制御（フレームシーケンス制御）が必要

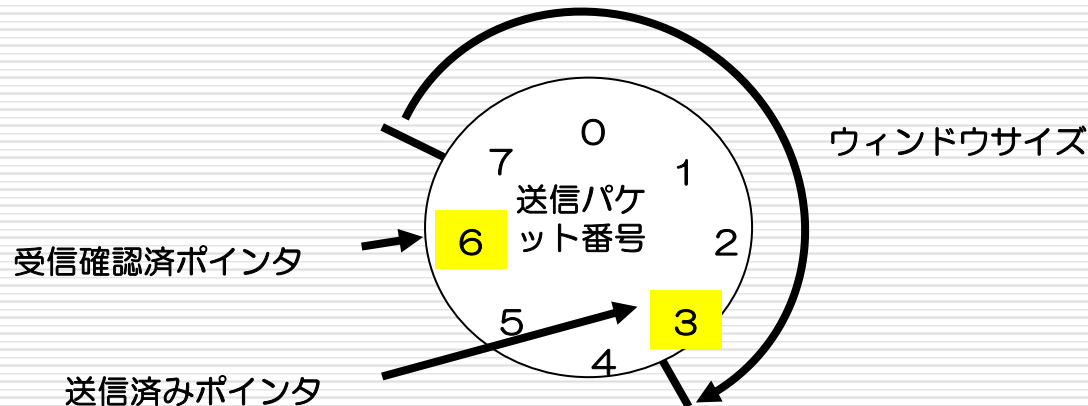
Link-by-link再送とEnd-to-end再送

- Link-by-link再送：誤りの発生した伝送区間でパケットを再送
 - 途中のノードは送信後もデータを保持して再送
 - 利点：再送に時間がかからない。
 - 欠点：途中ノードでの処理が複雑。大きなメモリが必要
- End-to-end再送：誤りが発生すると中間ノードはパケットを廃棄
 - 端末でパケットのシーケンス異常を検出して、端末相互間で再送
 - 利点：途中ノードでの処理が簡単。大きなメモリも不要
 - 欠点：時間がかかる。トラフィックが増大



フロー制御(Selective-repeatのnによる制御)

- 受信ノードのバッファ溢れやビジー時、あるいは次リンクの輻輳時に、送信ノードからの送出パケット数を抑制
- ウィンドウサイズの通知(Selective-repeatのnのこと)
 - 先行して送ることのできるパケット数を送信ノードに通知
 - 伝播遅延の長い伝送路、高速な回線、短いパケットを送るシステムではウィンドウサイズを大きくしないとスループットが出ない
 - アウトスタンディングフレーム数とも呼ぶ
- パケット数のみではリンク容量（帯域）との乖離があるため、パケット数ではなく、データ量で制御する場合もあり



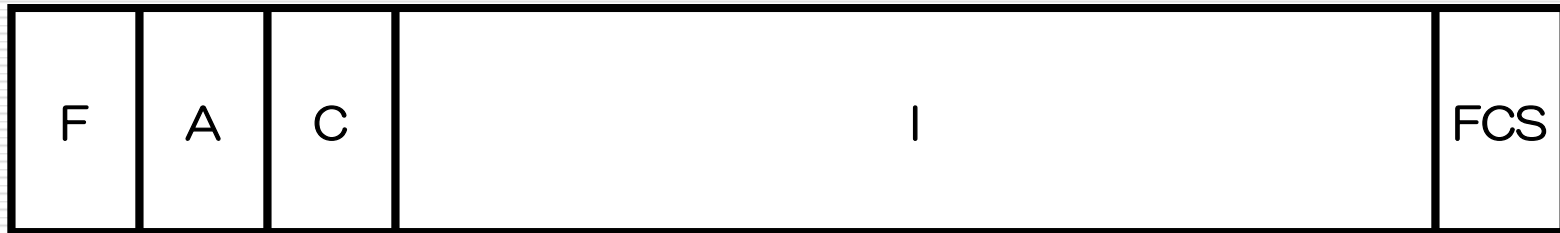
伝送制御手順

- データリンクの確立と開放
 - 通信相手の指定
 - 相手側がデータ交換可能か確認
- データ転送制御機能
 - データリンクが確立した後の実際にデータ交換する手順
 - ビット誤りが生じた時の誤り訂正手順
- 種類
 - 無手順
 - 有手順
 - ベーシック手順
 - HDLC手順 (High-level Data Link Control Procedure)

HDLC手順(High-level Data Link Control Procedure)

- JIS C-6363, JIS C-6364, JIS C-6365
 - IBMのSDLC (Synchronous data link control) 手順に基づいてISOが標準化
 - HDLCは歴史あるプロトコルで、現代の各種プロトコルの原型
- 特徴
 - 同期方式：フラグ同期
 - Bit Oriented Protocol
 - 0連続は発生しない（アイドル状態は1の連続）
 - 誤り制御：CRC方式
 - Go-back-NまたはSelective repeat
 - Bit Transparency
 - 任意のパターンのデータが送信できる
 - FAX等の符号化イメージ情報、バイナリーデータ
 - ビットパターンによってフレーム長が伸びる
 - X.25 N-ISDN Frame Relay, …..で使われている

HDLCのフレーム構造



F (Flag) : パケットの識別

A (Address) : 多重通信する際の識別用

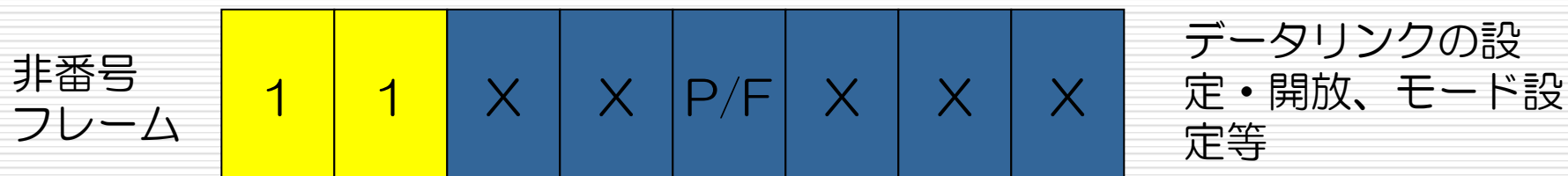
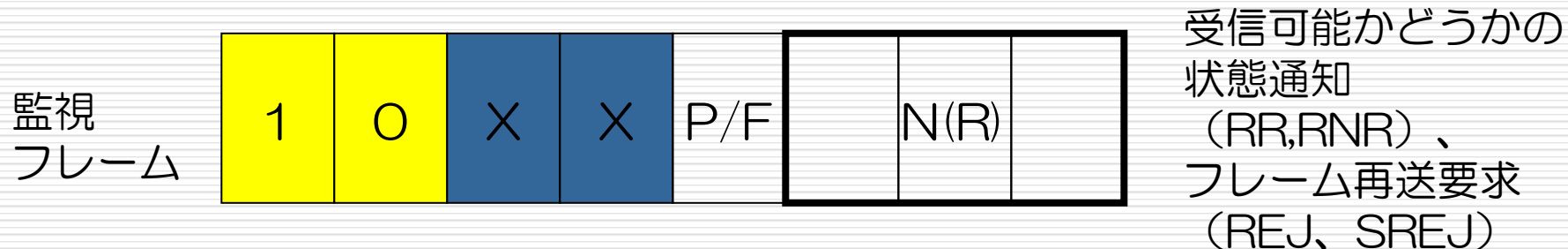
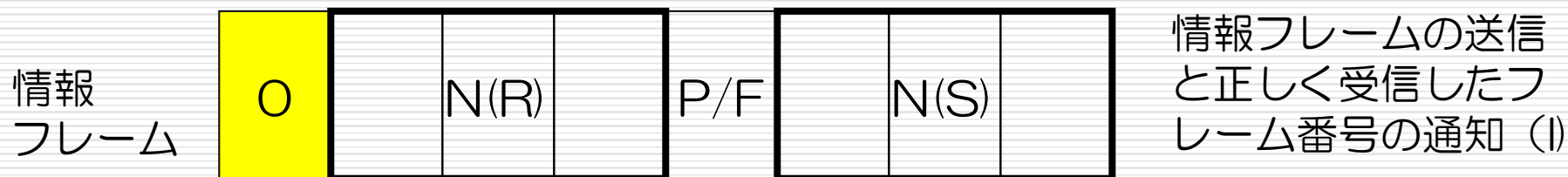
C (Control) : フレーム種別、コマンド/レスポンス

I (Information) : 情報

FCS (Frame Check Sequence) : 誤り検出用のCRC符号

誤り検出の対象はA/C/Iフィールド。2バイトまたは4
バイト (最大フレーム長による)

(補足) Cフィールドのコーディング例



N (R)/N(S):受信/送信情報フレーム番号

P/F:ポール/ファイナル (コマンドレスポンスの対応をとる)

（補足）HDLCを使うデータリンク制御

LAPB：X.25パケット交換サービス

link-by-link制御とend-to-end制御の組み合わせ

LAPD：ISDNの信号（Dチャネル）プロトコル（ITU標準Q.921）

宅内系でバス接続インタフェースに端末を自動インストールするためのTEI（端末ID）設定機能の追加

LAPF：フレームリレープロトコル（ITU標準Q.922）

誤り検出・フロー制御機能のみとし、再送制御機能・順序制御機能を削除（端末相互間）

MTP2：回線交換におけるNo.7共通線信号

リンク監視機能の強化

PPP：Point-to-point

プロバイダへのダイアルアップ接続、ADSLやFTTH、専用線
プロバイダバックボーン（POS：Packet Over SONET/SDH）