

1. データリンクの基礎技術

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

概要

- データリンクの提供している機能は何か
- 使用されている技術
 - デザイン・コンセプト
 - 誤り処理 (error control)
 - フロー制御 (flow control)
 - 基本的なプロトコル

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

データリンクとは

- 機能
 - 同一の伝送メディア上のホスト間の通信
 - 同一の物理媒体を共有しているホスト間通信
 - 異なる伝送メディアとの通信にはゲートウェイが必要
 - Layer 1 gateway: repeater (単純な信号の増幅)
 - Layer 2 gateway: bridge (フレームの再伝送)
 - Layer 3 gateway: router
 - 物理層サービスを利用しての通信機能の提供
 - パケット交換を実現
 - データリンク・フレーム (datalink frame)
 - ネットワーク層に対するインタフェース

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

データリンク層が提供する機能

- Data link control
 - フレーム
 - 誤り検出・訂正(error detection / correction)
 - フロー制御(flow control)
- Addressing
- Logical link control
 - ネットワーク層プロトコルの多重化
- 媒体アクセス制御(MAC)

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

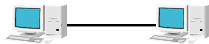
歴史

- 1970年代
 - 公衆データ回線用のデータ転送プロトコルの普及
 - X25(HDLC)
- 1980年代
 - LAN技術の普及
 - Ethernet
 - Token Ring
 - FDDI
 - 無線技術の登場
 - Wireless Ethernet
- 1990年代以降
 - LANトラヒックの公衆データ回線上での収容
 - Frame Relay
 - ATM
 - SONET and POS
 - 無線技術の広がり
 - Wireless Ethernet
 - 無線データ通信
 - LAN技術の広帯域化
 - Fast Ethernet
 - Gigabit Ethernet

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

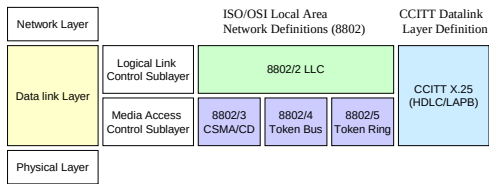
2つの王国 (1)

- Point-to-Point connection
 - CCITT X.25 (HDLC/LAPB)
 - デジタル専用回線やISDNで利用
 - 現在はITU-Tでの活動
- Broadcast Channel (LAN)
 - IEEE 802 (ISO 8802)
 - LLC (Logical Link Control)
 - MAC (Media Access Control)
 - Ethernet, FDDI, Token Ringなどが全て IEEE802 グループとして標準化
 - ISO8802グループとして国際標準化



Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

2つの王国 (2)



Copyright©2002 Suguru Yamaguchi and Youki Kadohayaashi

なぜデータリンク層は必要か

- パケット交換の基本的な機能を提供
 - 物理層が提供するビットレベルでの伝送機能を用いて、パケットレベルでの交換機能を実現
- 物理層の特性を考慮したデータ転送機能を提供
 - 安定したデータ転送処理
 - 遅延 (propagation delay)
 - 誤り (errors)
- ネットワーク層に対するインタフェース
 - 特に共通化したインタフェースが提供される場合、メディアに対する差違を吸収できる
 - IEEE802.2 と IEEE802.x

Copyright©2002 Suguru Yamaguchi and Youki Kadohayaashi

データリンク層のサービスモデル

- Unacknowledged Connectionless Service
 - 送信者が受信者と同期すること無くフレームを送信
 - 誤りが発生しても特別な処理をしない
 - ・ 上位層が誤り制御を実施
 - 高速、エラーの少ないリンク向けの制御方式
 - LAN, real-time traffic (voice/speech)
- Acknowledged Connection-oriented Service
 - データ送信前にコネクションを設定
 - ・ 送信側と受信側が同期した処理が可能
 - 信頼性 (誤り、順番) を保証
 - 処理は一般に複雑
 - 低速、エラーの多いリンクに有効
- Unacknowledged Connection-oriented Service
 - 信頼性 (誤り、順番) を保証しない
 - 高速、エラーの少ないリンク向けの制御方式
- Acknowledged connectionless Service
 - 受信者は各フレーム毎に受信確認 (acknowledge)
 - 誤り制御をデータリンク層で実施
 - 信頼性を向上

Copyright©2002 Suguru Yamaguchi and Youki Kadohayaashi

フレーム

● フレームの定義と同期

- フレーム(frame)
 - 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する可能性がある
 - 適当な長さのブロックに対して冗長な情報を追加して誤りを検出、あるいは、訂正する
 - 誤り制御処理のためのフレーム
- ビット列中からのフレームの取り出し
 - フレームの同期問題
 - » キャラクタ
 - » ビット列

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

フレームの取り出し (1)

● キャラクターベースでのフレームの取り出し

- 物理層としては文字を単位としたデータ伝送
 - CCITT X.21
 - 物理層フレームは8bit
 - コード体系はASCII(A5)
- 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り
 - character stuffing

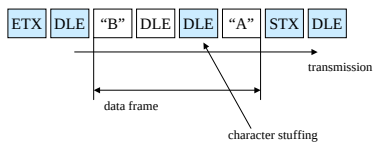
Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

フレームの取り出し (2)

Original Data

"B" DLE "A"

Transmission Sequence



Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

データリンクの基本形態 (2)

- Connection-oriented Service
 - データ送信前にコネクションを設定
 - 送信側と受信側が同期した処理が可能
 - 信頼性 (誤り、順番) を保証
 - 処理は一般に複雑
 - 低速、エラーの多いリンクに有効

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

誤り制御

- Error Control
- 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
- 技術
 - 符号技術
 - 誤り訂正符号 (Error Correction Code)
 - 誤り検出符号 (Error Detection Code)
 - プロトコル技術
 - タイマ (timer)
 - 再送 (retransmission)

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

誤り検出符号

- ビット誤り
 - パリティ (parity) 法が代表的
 - ASCIIでは7ビットでコード表現
 - 8ビット目にパリティ
 - even parity: コード内での1の数が常に偶数
 - odd parity: コード内での1の数が常に奇数
 - 8ビットフレーム内での単一ビット誤りの発見可能
- バースト誤り
 - 連続したビット誤り
 - 交換機やネットワークアダプタでの電氣的なノイズに起因することが多い
 - 巡回符号が広く使われる
 - CRC (Cyclic Redundancy Check code)
 - 幾つかの標準が存在
 - FEC (Forwarded Error Correction) の利用

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

CRC (1)

- フレームの後ろに誤り検出用のCRCチェックビットをつける
- ビット列の多項式表現
 - nビットからなるビット列
 - $(a_{n-1}, \dots, a_1, a_0)$ ただし a_i は0または1
 - 多項式表現
 - $F(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0$
- 多項式演算
 - modulo 2
 - 排他的論理和 (eXclusive OR) と等価

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

CRC (2)

- CRCの計算手順
 - 生成多項式 $G(x)$ ただし次数は r ($r < m$)
 - mビットのフレームを送信するとし、その多項式表現を $M(x)$ (次数 $m-1$) とする
 - フレームに r 個の "0" をつけたもの $x^r M(x)$ を作成
 - 多項式 $x^r M(x)$ を $G(x)$ で割った余り $R(x)$ を計算
 - 送信する符号語 $F(x)$ を作る
 - $F(x) = x^r M(x) + R(x)$

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

CRC (3)

- 誤りの検出
 - $F(x)$ は常に $G(x)$ で割り切れる
 - 割り切れなかった場合には誤りが存在
 - r ビット以下のバースト誤りは検出可能
 - それ以外の誤りは $G(x)$ に依存

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

標準化されたCRC (1)

- 代表的な標準
 - CRC-12
 - $x^{12}+x^{11}+x^3+x^2+x+1$
 - CRC-16
 - $x^{16}+x^{15}+x^2+1$
 - CRC-CCITT
 - $x^{16}+x^{12}+x^5+1$
- 上記以外にも標準化されたCRCがいくつかある

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

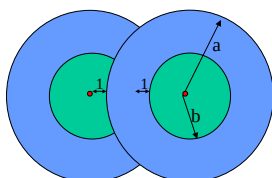
標準化されたCRC (2)

- 用途
 - CRC-12は6ビット文字表現に対して
 - CRC-16およびCRC-CCITTは8ビット文字表現に対し有効
- 検出特性 (CRC-16およびCRC-CCITT)
 - 全ての単一および2ビット連続ビット誤り
 - 全ての奇数個のビット誤り
 - 16ビット以下のバースト誤り
 - 99.997%の17ビットバースト誤り
 - 99.998%の18ビット以上のバースト誤り

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

誤り訂正符号

- Error Correction Code
 - ハミング符号 (hamming code) の利用
 - 符号語間の距離
 - a個の誤り検出: $a+1$ 以上
 - b個の誤り訂正: $2b+1$ 以上



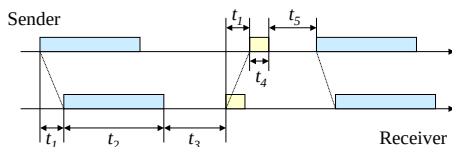
Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

基本的なプロトコル

- 誤り制御のため
 - 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
 - 再送 retransmission
 - 実行方法にはいろいろある
 - ARQ (Automatic Repeat reQuest)
- 基本的なプロトコル
 - Stop-and-wait
 - Go-back-N
 - Selective-repeat

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Stop-and-wait Protocol (1)



Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Stop-and-wait Protocol (2)

- 手法
 - 1フレーム送信毎にACK受信を待つ
 - 送信側タイムは $2t_1 + t_2 + t_3 + t_4$ 以上に設定
 - 送信側タイムがタイムアウトするとフレームを再送
- 特徴
 - 単純
 - バッファは送受信側それぞれで1フレーム分でよい
 - 回線使用効率が非常に悪い

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

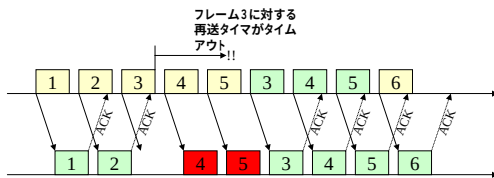
Go-back-N Protocol (1)

● 手法

- Stop-and-Wait Protocolを改善
- 連続してフレームを送出
- 誤りが発生した場合
 - ・ 誤りが発生したフレームを正しく受信するまで受信したフレームを破棄
 - ・ 正しい受信までACKも返さない

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Go-back-N Protocol (2)



Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Go-back-N Protocol (3)

● 特徴

- 誤りが発生すると正しく送られたフレームでも破棄され、無駄が大きい
 - ・ 特に再送タイムを大きく設定する場合不利
- 送信側で再送のためのバッファが必要
 - ・ ACKを受け取っていないフレームについては再送のためにバッファリングしておく必要あり
 - ・ Go-back-N の "N" はACKを待たずに送ることができるフレーム数 (on-the-fly frame数)
 - ・ バッファは再送タイムの値に依存

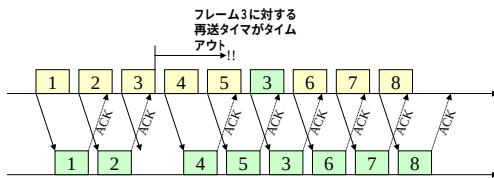
Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Selective-repeat Protocol (1)

- 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
 - 明らかに伝送効率は 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
 - 処理が複雑
 - 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

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Selective-Repeat Protocol (2)



Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

フロー制御

- Stop-and-Wait, Go-back-N では受信側では複雑な処理が発生しない
- Selective-Repeat では、受信側で順番制御
 - 受信側にバッファが存在
 - 送信側がうまく送らないと受信側バッファで溢れが発生する可能性有り
- フロー制御 (flow control)
 - 受信側との間で何らかの手法を用いて、バッファ溢れを抑制する

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

Window Flow Control

- ウィンドウ・フロー制御
 - ウィンドウサイズをWとする
 - 受信側のバッファサイズと同じと考えて良い
 - 送信側で (1) 既を送信したフレームで (2) ACKを受け取っていないフレームの数 (あるいはバイト数) を Nr とする
 - $Nr < W$ の間は送信を行う
 - sliding window flow control

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi

まとめ

- データリンクの基本的な技術を紹介
 - 誤り処理 (error control)
 - フロー制御 (flow control)
 - 基本的なプロトコル
- 実は他の階層でも使われる技術
 - ネットワーク層やトランスポート層などでも使われる
 - 一般的な技術であり、多少変形したものは沢山ある

Copyright©2002 Suguru Yamaguchi and Youki Kadobayashi