

2. 階層型プロトコルとは

概要

- 階層型プロトコルとは何か
 - » メリット・デメリット
 - » 現在使われている標準
 - » 用語

歴史的背景

- 1960年代
- 通常の通信サービス (例えば電話など) の技術とコンピュータ通信の技術は、特に連携なく発展
 - » 伝送・交換
 - » コンピュータネットワーク
- モデム時代
 - » ビット列をどのように信号化するか
 - » 変調技術

Copyright ©2000, Suguru Yamaguchi, All right reserved

歴史的背景

- 1960年代後半から1970年代初頭
- データ通信サービスは存在。しかしながら、非常にプリミティブなサービスのみを提供
 - » X.25
 - » IBM HDLC から発展的に標準化
 - » コンピュータネットワークへの第一歩
- LAN誕生
 - » Ethernet, Token Ring
 - » 高速データ通信技術の芽生え

Copyright ©2000, Suguru Yamaguchi, All right reserved

歴史的背景

- 1970年代中旬から1970年代後半
- システム相互接続の標準手順への期待
 - » さまざまなコンピュータネットワークシステムの登場
 - ・ 長距離通信技術としてのデジタル専用回線
 - ・ データ交換網 X.25
 - ・ LANシステム
 - » ベンダ毎、システム毎、納入先毎のカスタマイズ
 - ・ 導入コスト 開発コストの肥大化
 - ・ アプリケーション以外の部分での作業コスト大
 - » 標準化への強い期待

Copyright ©2000, Suguru Yamaguchi, All right reserved

歴史的背景

- 1970年代中旬から1970年代後半
- 複雑化した通信システム開発
 - » 機能的な分割とパッケージ化
 - ・ 構造化プログラミングと同じ考え方
 - ・ 開発の分割、実装検証
 - » 階層型プロトコルの原形発生
 - ・ Sync. link / HDLC / X.25
 - ・ Coax / Ethernet / XNS
 - ・ デジタル専用線 / ARPAnet

Copyright ©2000, Suguru Yamaguchi, All right reserved

歴史的背景

■ 1970年代後半から1980年代初頭

■ 標準化活動の開始

» 三つの大きな流れ

- 国際標準としての ISO/OSI
- 学術研究の成果としての TCP/IP
- 各コンピュータベンダによる標準化
 - IBM/SNA, Digital/DECNet, Xerox/XNS, ...

Copyright ©2000, Suguru Yamaguchi, All right reserved

OSI 7 Layer Reference Model

■ 階層型プロトコルを考える上の基本的なモデル

» 7階層型

- Physical / Data Link / Network / Transport / Session / Presentation / Application

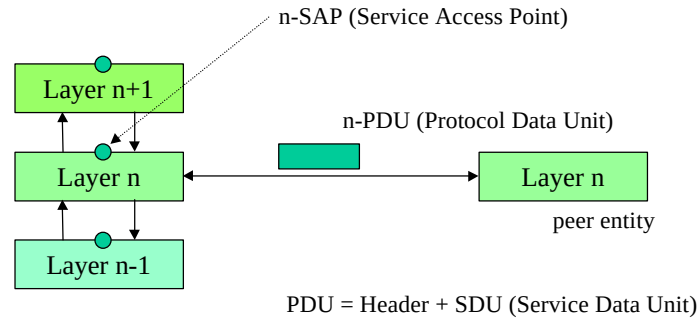
» ISO/OSI標準として規定

- 階層型プロトコルの概念を明確に定義
- 少なくとも多くの人に受け入れられた

■ 1970年代終わりから1980年代初頭にかけて標準化作業

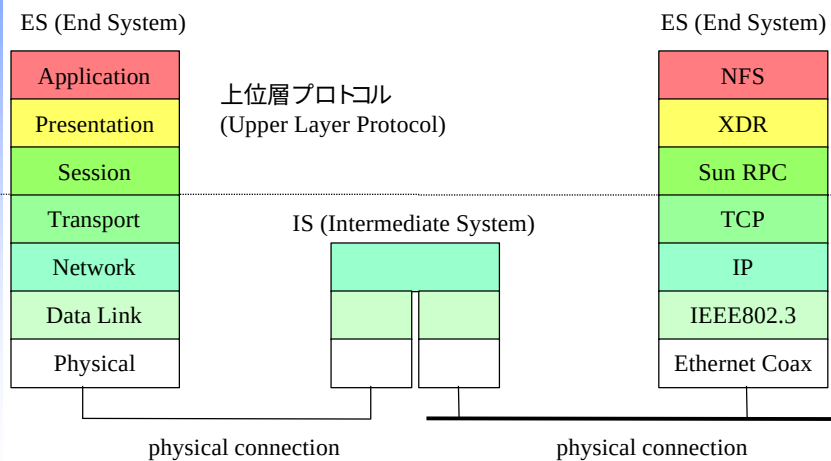
Copyright ©2000, Suguru Yamaguchi, All right reserved

OSI 7 Layer Reference Model



Copyright ©2000, Suguru Yamaguchi, All right reserved

OSI 7 Layer Reference Model



Copyright ©2000, Suguru Yamaguchi, All right reserved

Physical Layer

- 物理層
- 各伝送媒体 (communication media) に対して、ビット伝送手順を定義
 - » ノード間でのビット列伝送手順
 - » 例
 - 電気信号レベル (e.g. 0: $< +0.5\text{v}$, 1 $> +3.7\text{v}$)
 - ビット伝送手順 (e.g. エラー発見など)
- 各伝送媒体毎に定義
 - » 伝送媒体毎の特性を生かすことが目標
 - » 実際には上位層 (データリンク層) の特性に大きく影響

Copyright ©2000, Suguru Yamaguchi, All right reserved

Data Link Layer

- データリンク層
- 各伝送媒体に対して定義されるノード間でのパケット伝送手順の定義
 - » Physical Layer の機能を用いて定義
 - » パケット: 意味あるデータの固まり
- 標準の要素
 - » パケットフォーマット
 - » アクセス手順

Copyright ©2000, Suguru Yamaguchi, All right reserved

Data Link Layer

■ 数多くの標準が存在

- » HDLCを起源とするデジタル専用回線を用いた 1対1通信でのデータ交換手順
- » LANを中心とするIEEE802.x シリーズ
 - Ethernet 802.3x
 - Token Ring 802.5
 - FDDI 802.9
- » ISDNなどの今日的電話技術もDLLとして定義
 - I.100シリーズでは、発呼手順、データフレームの定義などが含まれる

Copyright ©2000, Suguru Yamaguchi, All right reserved

Network Layer

■ ネットワーク層

- データリンクに依存すること無くネットワーク内のノード間でのデータ交換機能を提供
 - » ES (終端システム: End System) 間の通信定義
 - » ノードに対するアドレスの付与
 - » ゲートウェイ機能をIS (中間システム: Intermediate System) として実装
 - » パケットの定義

Copyright ©2000, Suguru Yamaguchi, All right reserved

Transport Layer

- トランスポート層
- ネットワーク内のプロセス間の End-to-End 通信を定義
 - » ES の中にはプロセスがある
 - » プロセス = サービスの処理機構
 - » プロセスにとって、一番基本となるプロトコル
 - » ネットワークで共通のトランスポートプロトコルが利用される

Copyright ©2000, Suguru Yamaguchi, All right reserved

上位層プロトコル

- Session Layer 以上を上位プロトコル (Upper Layer Protocol) と呼ぶ
 - » 各アプリケーション毎に定義される
 - » アプリケーションからの要求に強く拘束される

Copyright ©2000, Suguru Yamaguchi, All right reserved

Session Layer

- セッション層
- 通信の意味的な単位の定義
 - » Transaction
 - » Session
- 通信の単位に対する処理の定義
 - » Transaction Logging & Roll-back operation
 - » Session Termination
- アプリケーションから見た通信の概念的な定義と基本処理

Copyright ©2000, Suguru Yamaguchi, All right reserved

Presentation Layer

- プレゼンテーション層
- データの表現形式
 - » システム・プラットフォームの差違なくデータが正しく表現・理解されるための基盤
 - » 例
 - 数値 1 の表現
 - 何バイト使って表現するか
 - » 1, 2, 4, less than 1 byte (6bit) ...
 - バイトオーダーはどうするのか
 - » Little Endian / Big Endian
 - 送信する場合のビットオーダーはどうするのか
 - » MSB first, LSB first

Copyright ©2000, Suguru Yamaguchi, All right reserved

Application Layer

- アプリケーション層 (応用層)
- 具体的なアプリケーションを構成するためのプロトコル定義
 - » 注意：アプリケーションそのものではない
 - » 電子メールの交換では SMTP が定義
 - » しかしながら電子メール交換が可能なアプリケーションは数多く存在
 - MTA: sendmail, qmail, Sun Enterprise Mail Server, etc....
 - MUA: Eudora, Netscape Messenger, MS/Outlook, etc....

Copyright ©2000, Suguru Yamaguchi, All right reserved

華の1980年代

- 1980年代は各コンピュータベンダが独自に開発してきたネットワーク技術が多かった
 - » IBM's SNA, DECNET, Xerox's XNS
 - » TCP/IP
 - » AppleTalk, Novel's NetWare, NetBios, etc...
- 数多くの要素技術が開発

Copyright ©2000, Suguru Yamaguchi, All right reserved

統合に向けて

■ 1990年代になるとプロトコルのより一層の統合が始まった

» ISO/OSIの失敗

- 動かないものが多すぎた
- 結局 X.400, X.500 以外の技術は不発
- X.400, X.500 はアプリケーション
- トランスポート以下は誰も使わなかった

» Internet Protocol の急激な広がり

- 動くプロトコルであった
- 米国の積極的な開発・展開
- Internet が世界中をカバー

Copyright ©2000, Suguru Yamaguchi, All right reserved

ベンダ独自プロトコルの衰退

■ 標準化による優位性の欠如

- » DECNET → OSI Protocol
- » AppleTalk → TCP/IP上のアプリケーションへ
- » Netware, NetBios → TCP/IP上のアプリケーションへ
- » Xerox XNS: 衰退からほぼ滅亡へ

■ IBM SNAだけが生き残り

- » 大型汎用機
- » 業務用システムでの展開
- » なかなか足抜けできない

Copyright ©2000, Suguru Yamaguchi, All right reserved

味わうべき技術

- データリンク技術
 - » Ethernet に代表されるLAN技術
 - » HDLCに代表される古典的パケット交換技術
- TCP/IP プロトコル群 (TCP/IP Protocol Suites)
 - » 世界でもっとも使われているコンピュータネットワークプロトコル
 - » 特に IP とTCP
- ネットワーク管理技術
- アプリケーション技術

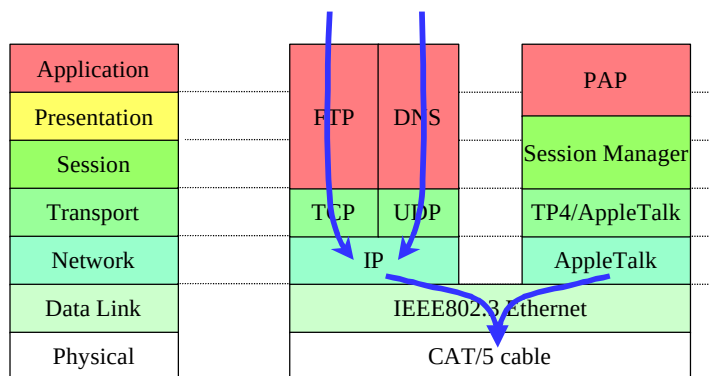
Copyright ©2000, Suguru Yamaguchi, All right reserved

階層型プロトコルの技術的特徴

- 多重化 (Multiplexing) と分離 (Demultiplexing)
 - » 下位層は、複数ある上位層プロトコルを多重化する
 - ・ 例
 - データリンクプロトコルである IEEE802.3 (Ethernet) は、上位層プロトコルとしてIP, AppleTalk を同時に取り扱うことができる。したがって、IEEE802.3はネットワークプロトコルを多重化して取り扱うことができる
 - » 受信側では、上位層プロトコルに応じて分離

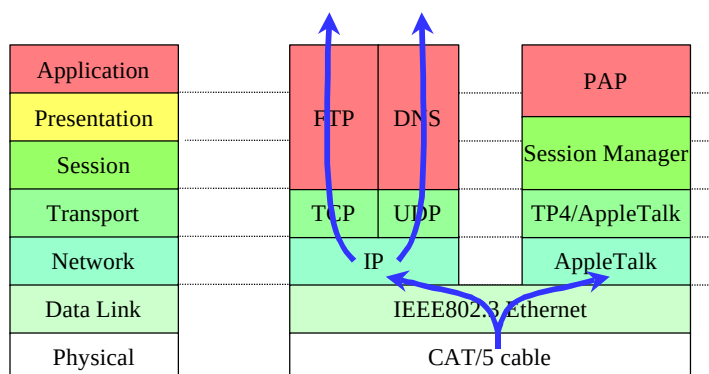
Copyright ©2000, Suguru Yamaguchi, All right reserved

Multiplexing



Copyright ©2000, Suguru Yamaguchi, All right reserved

Demultiplexing



Copyright ©2000, Suguru Yamaguchi, All right reserved

階層型プロトコルの技術的特徴

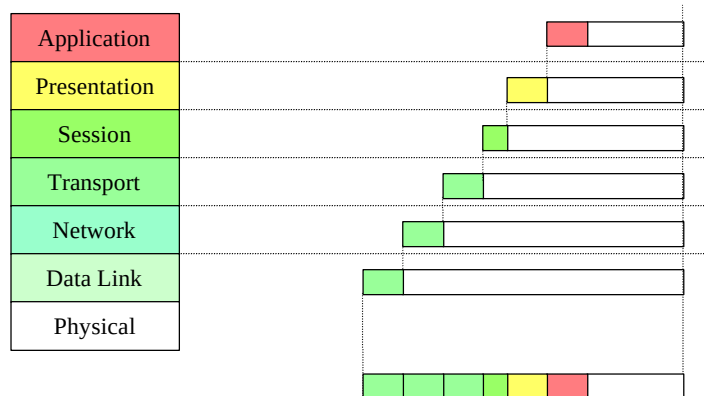
■ カプセル化 (encapsulation)

» 下位プロトコルは、上位プロトコルのパケットをカプセル化する

- $(n-1) \text{ PDU} = (n-1) \text{ Header} + (n) \text{ PDU}$
- 上位のプロトコルのパケットを、データ領域として格納する。

Copyright ©2000, Suguru Yamaguchi, All right reserved

Capsulation



Copyright ©2000, Suguru Yamaguchi, All right reserved

メリット・デメリット

■ メリット

- » 単純明解
- » 理解しやすい
- » 分割開発が可能
- » 機能隠蔽と独立したバージョンアップ

■ デメリット

- » 階層型の実装をするとパフォーマンスがでない
- » カプセル化処理によって、メモリ管理とデータ管理が難しい
 - 可変長データの取り扱いは大変

Copyright ©2000, Suguru Yamaguchi, All right reserved

まとめ

■ 階層型プロトコル

- » OSI 7 Layer Reference Model
- » 理解をする上でのモデル
 - 実装のモデルとはちょっと違う
 - 特に性能を考えると単純ではない

Copyright ©2000, Suguru Yamaguchi, All right reserved