階層型プロトコルとTCP/IP

概要

- 階層型プロトコルの概念と構成
  - ISO 7 layer reference model
  - あくまでも概念モデルであるが、「参照モデル」と呼ばれることだけあって、ネットワークを語るときに知らないと恥ずかしい
- TCP/IPの概要
  - 階層型プロトコルであるが、その成立がISO 7 layer reference model と同時期であったために、5層モデルを採用
  - 現状では、7層モデルで語れることもあるが、更に最近はより複雑な状況にある

階層化プロトコル、その概念と構成

概要

- 階層化プロトコル誕生の背景
- OSI 7 Layer Reference Model
  - 通信プロトコルを語るときの基本モデル。このモデルを使って議論されることが多いので、常識として学んでおこう
  - 各層の機能
  - 各層のインタフェース
- 階層化プロトコルの発展
  - さらに近年発展してきている。どんな展開をしているかを学ぼう

歴史的背景（1）

- 1920年代～1950年代
- 通信理論の発展
  - 二つの基礎的理論。この二つの理論は、その後発展する通信理論（伝送理論）の基礎を形成する。特にシャノンの定理は正しく理解することが重要
  - Harry Nyquist (1924)
    - Maximum data rate = $2H \log_2 V$ (bits/s)
      - H: low pass filter bandwidth
      - V: discrete level of signal
  - Claude Shannon (1948)
    - ノイズのある伝送路への拡張
    - Maximum data rate = $H \log_2 (1+S/N)$
      - S/N: signal-to-noise ratio

歴史的背景（2）

- 1960年代
- 大型汎用コンピュータ（例えばIBM370等）が広く使われ、コンピュータにアクセスする手段が求められた
  - 端末 (terminal) とコンピュータ (host computer) の接続
  - 単純な文字情報の伝送路をどのように構築するかが主課題
- 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、特に連携無く発展
  - 伝送・交換（電信・電話）
  - コンピュータネットワーク
- モデム時代
  - アナログ通信路に、低速でディジタル化された情報を伝送
    - 300bit/sec, 1200bit/sec, etc.
  - ビット列をどのように信号化するか
  - 変調技術 (differential Manchester coding etc.)

歴史的背景（3）

- 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 - Metcalfe's Ethernet
  - Ethernet, Token Ring
  - 高速データ通信技術の芽生え
  - モデム技術と完全に異なる技術の出現
    - 1200bit/sec 時代に 1Mbps の技術を提示（初期のEthernet）

歴史的背景（4）

- 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ィジタル専用回線の一般か
    - 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での作業コスト大
  - 「大規模オンラインシステム」
  - 標準化への強い期待
    - ベンダ中心の技術体系化に対する強い不安
    - 国際標準として成立させ、通信ができる

歴史的背景（5）

- 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
  - 階層化プロトコルの原型発生
    - Sync. Link / HDLC / X.25
    - Coax / Ethernet / XNS
    - デジタル専用線 / ARPAnet

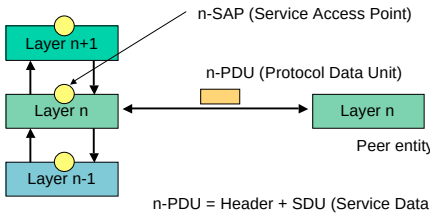
歴史的背景（6）

- 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, ….

OSI 7 Layer Reference Model

- 階層化プロトコルを考える上での基本的なモデル
  - 7階層型
    - Physical / Data Link / Network / Transport / Session / Presentation / Application
  - ISO/OSI標準として規定
    - 階層化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が行われ策定
  - あくまでも参照モデル (reference model) としてまとめることで、通信プロトコルの議論を明確化することに寄与

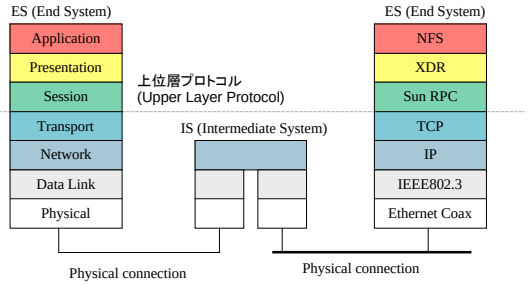
OSI 7 Layer Reference Model



n-PDU = Header + SDU (Service Data Unit)

※ 階層型プロトコルにおける隠蔽の効果を想像してみればよい

OSI 7 Layer Reference Model



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: < +0.5v, 1 > 3.7v)
    - ビット伝送手順 (e.g. 同期方法, 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活かすことが目標
  - 実際には上位層(データリンク層)の特性に大きく影響
    - データリンク層とセットで定義されることが多い

Data Link Layer (1)

- データリンク層 (Layer 2)
- 各伝送媒体に対して定義される、ノード間でのフレーム伝送手順の定義
  - Physical Layerの機能を用いて実現
  - フレーム:ビット列に境界を定義
    - 意味のあるデータの固まり
  - 2つ以上のノードがある場合には調停、協調が必要
- 標準の要素
  - 伝送媒体内での各ノードの識別方法
  - フレームフォーマット (frame format)
  - アクセス手順 (MAC sub-layer)

Data Link Layer (2)

- 数多くの標準が存在
  - HDLCを起源とするディ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 802.3
    - 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
- Physical layerと共に定義されることが一般的
  - 物理的伝送路と伝送手順は不可分と考えるのが普通

Network Layer (1)

- ネットワーク層 (Layer 3)
- データリンクに依存することなく、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

Network Layer (2)

- 標準の要素
  - ES, ISのアドレスの構造
  - パケットの形式
  - パケットの経路制御
  - 同報通信

Transport Layer (1)

- トランスポート層 (Layer 4)
- 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
  - ネットワークでの共通のトランスポートプロトコルが利用される
    - 通信を行う ES は、同一のトランスポートプロトコルを利用

Transport Layer (2)

- トランスポート層の機能
  - パケットより使いやすい通信サービスの提供
  - End-to-Endでの通信の確立
  - パケットの再送
  - エラー処理

上位層プロトコル

- Session Layer以上を上位層プロトコル (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

Session Layer

- セション層
- 通信の意味的な単位の定義
  - Transaction
  - Session
- 通信の単位に対して、処理を定義
  - 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的な処理

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異無く、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 (6 bits), ….
      - 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

Application Layer

- アプリケーション層（応用層）(Layer 7)
- 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
  - 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 (simple mail transfer protocol) が定義されている
  - しかしながら電子メール交換機能を持ったアプリケーションが数多く存在
    - MTA: sendmail, qmail, postfix, etc.…
    - MUA: Eudora, Mozilla Thunderbird, MS/Outlook, etc.…

階層化プロトコルの発展

- 1980年代
- ベンダ独自の階層化プロトコル
  - IBM SNA, DECNET, Xerox XNS
  - AppleTalk, Novell Netware, NetBIOS
- OSI階層化プロトコル
  - CLNP, TP4, IS-IS, X400, …
- TCP/IP
  - RIP, EGP/BGP, OSPF
  - TELNET, SMTP, DNS, FTP, SNMP, NTP, ….
  - 米国DARPAが標準として採用、積極的に推進
    - ARPAnet, MILnet

階層化プロトコルの発展

- 1990年代
- 勝者が決定
- OSIの失敗
  - 標準化が遅すぎた（ISO標準化プロセスの敗北）
  - 仕様が複雑
  - 参照モデルと、X.500だけが残った
- TCP/IPの勝利
  - 標準化が速い（IETF標準化プロセスの勝利）
  - 米国が積極的に使い続けた
  - 仕様が比較的単純、実装重視

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、そして滅亡
- IBM SNAだけが生き残り
  - 大型汎用システム
  - 基幹業務システムでの展開からの脱却が進まない
    - 大規模システムからの足抜けは難しい
    - レガシーシステム問題 (Legacy System)

階層化プロトコルの発展

- 2000年代
- 上位層アーキテクチャの発展
  - セション層
  - プレゼンテーション層技術
- “Demise of protocols”
- 各層の細分化、高度化
  - MPLS
  - J2EE, .NET, GRID….

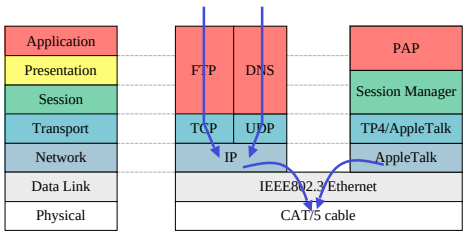
味わうべき技術

- 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 Suite)
  - 世界で最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

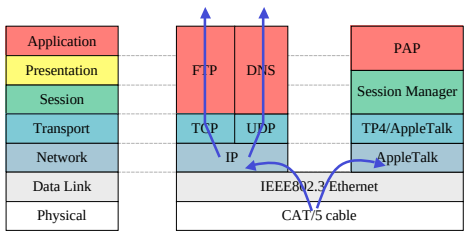
階層化プロトコルの技術的特徴

- 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、同一のEthernetで、複数のプロトコルを同時に使うことができる
    - 受信側では、上位層プロトコルに応じて分離

Multiplexing



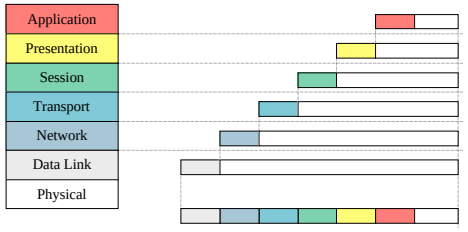
Demultiplexing



階層化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) PDU = (n-1) header + (n)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

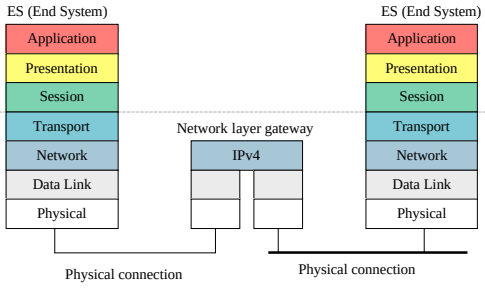
Encapsulation



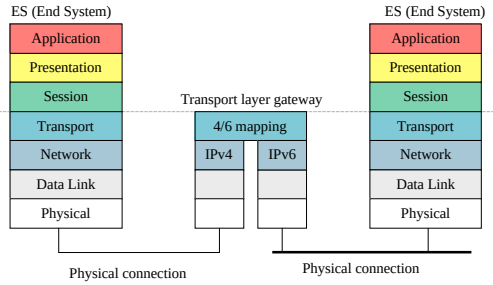
利点・欠点

- メリット
  - 単純明快
  - 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
  - 異種ネットワークの相互接続が可能
- デメリット
  - 階層型の実装をすると、動作時の性能を高くすることができない
  - 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
    - 可変長データの取扱いは大変

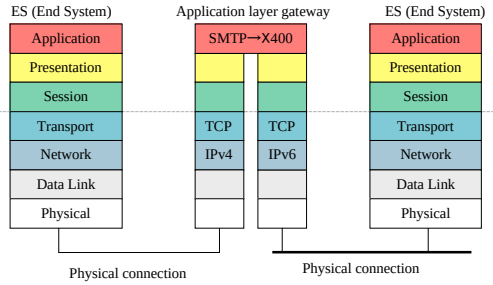
Network Layer Gateway



Transport Layer Gateway



Application Layer Gateway



まとめ

- 階層化プロトコル
  - 問題の段階的解決
  - 各層で提供されるサービスの抽象化
    - 物理層：ビットを提供
    - データリンク層：フレームを提供
    - ネットワーク層：パケットを提供
    - トランスポート層：コネクションを提供
  - インタフェースと実装の分離
  - 異なるデータリンク技術や異なるネットワーク技術の相互接続が可能に
    - さまざまな層でのゲートウェイが可能

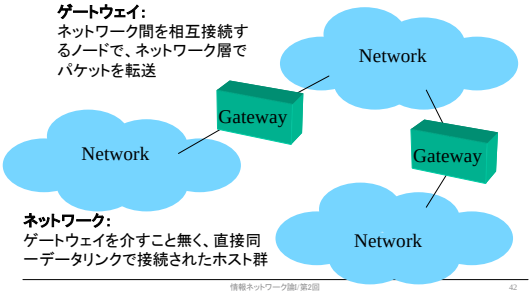
TCP/IPの概要

TCP/IP Protocol Suites

- Internetで使われているプロコル群
- LANから広域ネットワークまで広く利用
- 大部分のシステムで稼動
  - Computers: UNIX workstations、PC、Supercomputers
  - PDA, 温度計, 車, 衛星, 次世代携帯電話, 家電製品,…
- 積極的な研究開発と標準化活動
  - IETF (Internet Engineering Task Force)
  - http://www.ietf.org/
- 階層型プロトコルとして構成
  - 初期のモデルは5階層モデル

TCP/IPでのモデル

ゲートウェイ:
ネットワーク間を相互接続するノードで、ネットワーク層でパケットを転送



ネットワーク:
ゲートウェイを介すこと無く、直接同一データリンクで接続されたホスト群

TCP/IP Protocol Suits

OSI	TCP/IP	処理単位	アドレス
Application	Application	Message / Stream	
Presentation			
Session			
Transport	Transport	Transport Packet	Port
Network	Internet	IP Datagram	IP address
Data Link	Network Interface	Frame	Datalink Address
Physical	Hardware		

Network Interface Layer

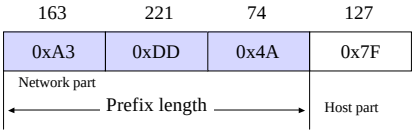
- ネットワーク層の処理単位であるIPデータグラムの転送処理を行う
  - 上位層とのやりとり
    - IPデータグラム
    - 次の転送先（同一データリンク内）のIPアドレス (next hop address)
- 基本的な処理
  - Next hop address とデータリンクアドレスのマッピング
  - データリンク・フレームの組み立て
  - ネットワークハードウェアの制御
  - 送受信処理
- 同一のデータリンク内のホストへのデータ転送
  - ゲートウェイを介して異なるネットワークへデータを転送する機能はネットワーク層

Internet Layer: IP

- IPv4 / IPv6
  - ノードを一意に識別する
  - データリンク層の違いを吸収する
- IP (Internet Protocol)の特徴
  - コネクションレスサービス (connectionless service)
    - IP データグラムが送信単位
    - 送信ホストと受信ホストは特にコネクション設定をしない（郵便型）
    - エラーが発生するとIPデータグラムは捨てられる（Best Effort 型）
  - インターネット内のホスト間の通信機能を提供
    - 異なるネットワークとの通信
    - ゲートウェイにおけるネットワークインタフェースの選択
    - ルーティング (routing)

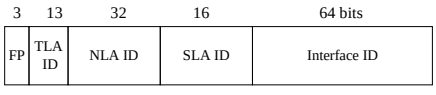
IPv4アドレス

- ネットワーク部とホスト部
- 境界はサブネット毎に異なる
- 163.221.74.127/24
- 203.178.142.0/27



IPv6

- 128bit アドレス
- 今後25年間、インターネットの成長を支えるのに十分なアドレス空間
  - 当面、アドレス空間の1/8を利用
- Aggregatable Global Unicast Address allocation



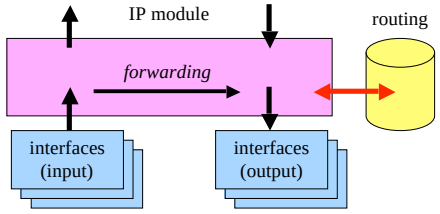
IPv6の特徴

- 大きなアドレス空間
- アドレススコープの概念
  - 構造化されたアドレス
- Classはなくなった
- 集約可能な (aggregatable) アドレス構造
- 拡張性の高さ

IPパケットの転送

- Hop-by-hop, destination oriented
  - 各ゲートウェイで転送先を決定
    - 転送先＝出力インタフェース
  - 各ゲートウェイが何らかの情報を共有していることが前提
    - 経路制御アルゴリズム
- Forwarding and routing
  - 経路制御アルゴリズムが導出した経路情報
  - 経路情報にしたがったパケットの転送

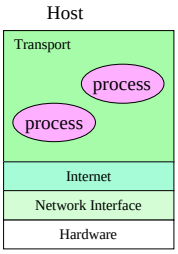
ゲートウェイアーキテクチャ



ネットワークインタフェース選択によるルーティング
output I/F = f(destaddr): f(x) が routing

Transport Layer

- 基本的なモデル
  - プロセス間の end-to-end の通信機能を提供
  - プロセスの識別
    - <IP address, port>
  - IP層の機能を利用
- ポート
  - トランスポート層での識別子
  - 2オクテット
  - トランスポートプロトコル毎に定義
    - TCPとUDPでは別の割当て
    - 同じ番号でも意味が全く違う
    - 例えば 5678/TCP, 5678/UDP
- peer entity の識別
  - (送信IP, 受信IP, プロトコル番号, 送信ポート, 受信ポート)
  - プロトコル番号
    - 標準で規定
    - TCP=6, UDP=17



TCP

- Internet Transmission Control Protocol
  - 電話型通信（全二重）
    - Connection Oriented
    - Virtual Circuit
  - 構造化されていないストリーム
    - メッセージの長さに制限はない
  - 信頼性技術
    - 信頼性の保証
    - メッセージの順番の保証
    - 受信確認 (acknowledgement)
    - 誤り時の再送 (retransmission)

Virtual Circuit

- Byte stream semantics
  - 始点でバイト列を送ると、終点でバイト列が出てくる
  - 双方向：「行き」と「帰り」がある、混ざらない
- パケット境界はアプリケーションには見せない
- Connection setup / release
  - 両端点が合意してコネクションが張られる

UDP

- Internet User Datagram Protocol
  - 郵便型（接続を確立しない）
    - Connectionless
    - データグラム通信
  - 信頼性が保証されない
    - データグラムの到達性保証、順番保証などの機能はない
    - IP層の機能をそのままユーザが利用したいケースに適用
    - Best Effort
  - 最大メッセージ長が規定されている

その他のTransport Layer Protocol

- TCP, UDP以外にも幾つかの Transport layer protocol が開発・標準化
  - VMTP, RTP など
- 研究は続けられているが、有用でわかりやすいセマンティクスがない
  - コネクション、パケットは単純明快
- IPが複数のTransport layer protocol のパケットを多重化する

Application Layer

- ユーザに直接サービスを提供
- トランスポート層サービスを利用
- 例えば、
  - SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), TELENET, POP (Post Office Protocol), HTTP (HyperText Transfer Protocol), ….
- プロトコルに付随してデータフォーマットなどさまざまなものが標準化
  - Webでは、HTTP, HTML

開発が進む Presentation Layer

- MIME - 文字表現の国際化
- ASN.1 - OSIから借りてきた標準
- XDR - Sunが作った業界標準
- XML - W3Cが作った標準
- どれも多くのアプリケーションで使われてきている

アドレスの考え方

- 各層でアドレスが定義
  - アプリケーション層
    - アプリケーション毎の空間定義
    - WWWでのURL
  - トランスポート層
    - peer entity となるプロセスの組を識別
    - ポート (Port)
    - 2 オクテット (unsigned short)
  - ネットワーク層
    - ホストの識別
    - Internet Address, IP host address
    - 4オクテット
  - データリンク層
    - 各標準に従う
    - Ethernetなら、Ethernet Address の定義

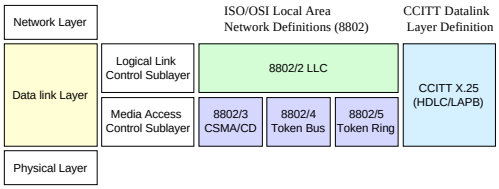
まとめ

- TCP/IP protocol suit
- IP (IPv4 & IPv6)
- Gateway architecture & routing
- TCP and other transport protocols
- Application Layer protocols
- Newly added layer: presentation layer

階層型プロトコルの新たな分化

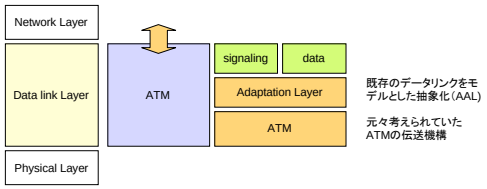
Data Link Layer = multiple sub-layers (1)

- Layer2 (data link layer) は、様々な伝送メディアにおけるフレーム伝送機能を提供する
  - 伝送メディア毎にフレームを何種類も作ると実装が大変
  - IEEE802における 802.2 (LLC layer) が生まれた
    - ※ IEEE802は次回以降詳細説明



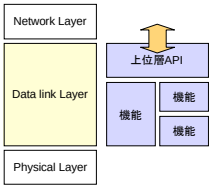
Data Link Layer = multiple sub-layers (2)

- 一度作ったデバイスドライバは何度も活用したい
  - 異なる伝送メディアでも、同じフレームと、同じインタフェースを使うことを考えたい = 開発労力の削減
  - ATMが先駆者
    - ※ ATMについては次回以降詳細説明



Data Link Layer = multiple sub-layers (3)

- サブレイヤ化 (sub-layering)は最近の流行
  - 上位層のインタフェースを変えない
    - 既存のデータリンクを最大限活用する
  - しかしながら、新たな機能をサブレイヤ化して導入
    - 新しいデータリンクによる高機能化、高性能化
    - IEEE802委員会での明確な方向付け
- サブレイヤ化は、実は機能分化を通した機能再定義
  - 階層化した共通API構造を定義することによって開発期間の短縮をねらう



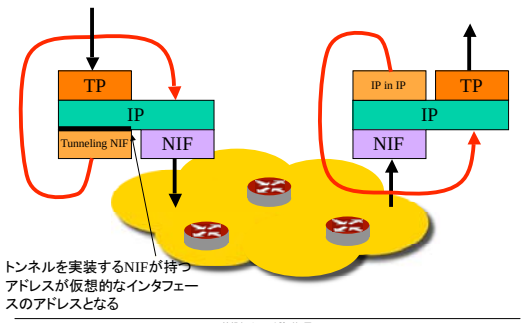
Sub-layering is not only for Layer2

- サブレイヤの考え方は他の層にも展開している
- Layer 3
  - ネットワーク層機能を二階層に分離
    - ネットワーク内での物理的なノード間でのパケット伝送機能
    - 上位層に対するノードの論理的なモデルを提示
      （例1）移動するノードのアドレスを変化させないようにするモデル。Mobile IP 等で採用されている
      （例2）暗号化通信をネットワーク層で実装し、上位層に対しては今までと全く変わらないインタフェースを提供する
- Layer 4
  - 特定サービスについてのプロセス間伝送の多機能化
    - Real-time transmission, multiple data-path management, performance management, etc.…
- 最近のブーム

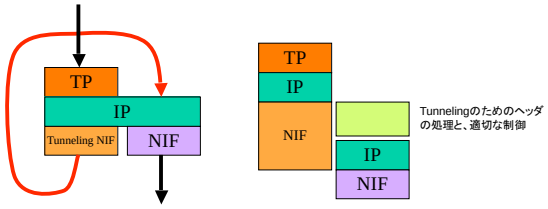
♪ L3 の上に、L3のせてー ♪

- 7階層にとらわれない技術が生まれている
- L3 の上にL3を乗せると何が起きる?
  - “Tunneling” と呼ばれる技術が生まれる
  - トンネリング (tunneling) は最近では広く使われている。
    - IPsec / VPN (Virtual Private Network)
    - IP Multicasting
    - 仮想ネットワーク (Mbone, 6bone), overlay network
  - 新たな課題もたくさん生み出した
    - 経路制御
    - MTU管理

IPトンネリング



IPトンネリングの構造



- IPデータグラムをIP層で運ぶことにより、遠隔ノード間仮想的に直結できる
- ネットワークの仮想的な展開に大きく寄与

階層型プロトコルは自由なモデル

- もやは何でもあり得る
  - 二つの層の間に、新たな層を定義することも言われている
    - 「4.5層サービス」:セション層サービス(アプリケーション毎)の一部をトランスポート層(ノード単位で共通機能)に混ぜる
    - 3.5層サービス
  - 7層に限定されない
  - 複数の層をまとめて議論する
  - 求められる機能が多様化したことから、最終的にモデルも自由度が求められ始めた
    - 本当の意味での reference model になってしまった
    - 設計・実装のモデル化は別のアイディアが必要となった