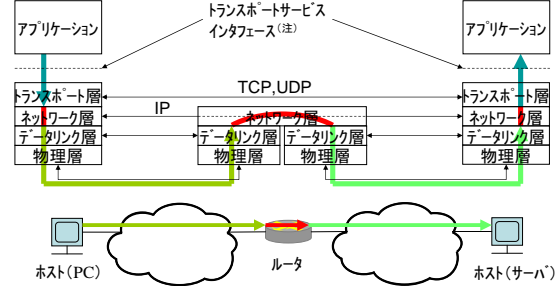


第14回 ネットワーク制御

OSにおけるネットワークの位置づけ

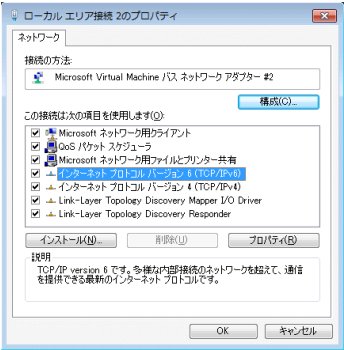
- ネットワーク=入出力装置 (通信)
  - 端末や他ホストとの間で、通信回線を介して入出力する
  - OSは通信制御機能を提供する
- ネットワーク=論理的な通信路
  - OS参照モデル(ネットワークアーキテクチャ)に基づくAP間通信
  - OSはAPIとしてトランスポートサービスを提供(ソケットインタフェース)
- ネットワーク=共用資源
  - LANを介してプリンタやファイルを共用
  - OSは共用資源へのアクセス機能を提供(狭義のネットワークOS)
- ネットワーク=プログラム+計算機能
  - プログラムが、他コンピュータに処理をさせ、結果を返してもらう
  - OSは遠隔手続き呼び出し(RPC:Remote Procedure Call)を提供
    - 注:ミドルウェア(OSとAPの間に位置するプログラム)で実現されることも多い

通信プロトコルとプログラミングインタフェース

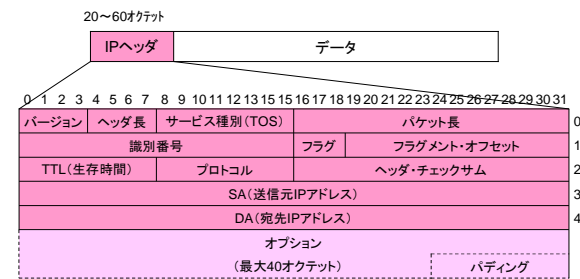


注:トランスポートサービスインタフェースは、通信プロトコルとしては、規定対象外。しかし、オペレーティングシステムでは、APIとして、明確に定義する必要がある。  
TCP/IPのトランスポートサービスを提供するAPIが、ソケットである。最初に、Unixにおいて提供され、広く使われるようになった(Windowsでも採用されている)

Windows7におけるIPv6



IPv4パケットヘッダの構造



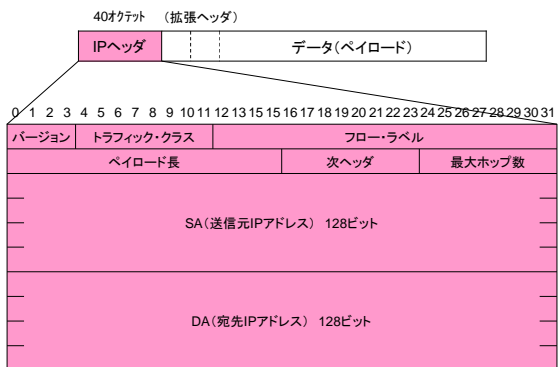
ヘッダ長が可変なので、ハードウェア処理が難しい。  
ヘッダチェックサムを各ルータで計算する(TTL値が変わるため)必要があり、処理が重い。

IPv4ヘッダのフィールド

フィールド名	幅	意味および説明
バージョン	4bit	IPv4を表す「0100(2進数表示)」になる
ヘッダ長	4bit	IPv4ヘッダの長さ。先頭から、オプションの最後までの含むサイズ(を4で割った値をセットする)
サービス・タイプ(TOS)	8bit	パケットの優先度などを表すために用意されているが、実際にはほとんど使われていない
パケット長	16bit	IPv4ヘッダも含む、IPパケット全体の長さ
識別子	16bit	フラグメント化されたパケットの識別用番号。パケットをフラグメント化すると、すべて同じIDを持つ。これを元に、元のパケットを識別・復元する
フラグ	3bit	制御用フラグ。フラグメントの存在やフラグメント禁止などを表す
フラグメント・オフセット	13bit	フラグメント化されたパケットにおいて、元のオフセットを表す
TTL	8bit	パケットの寿命を表す数値。ルータを1つ通過することに1ずつ減算され、0になるとパケットが破棄される。ルーティングがループになるのを防ぐ
プロトコル番号	8bit	上位プロトコルの番号
ヘッダ・チェックサム	16bit	IPv4ヘッダのチェックサム。TTLが変わると再計算する必要があるため、ルータの負荷は軽くない
送信元IPアドレス	32bit	送信元ノードのIPv4アドレス
宛先IPアドレス	32bit	送信先ノードのIPv4アドレス

## 第14回 ネットワーク制御

## IPv6パケットヘッダの構造



## IPv6ヘッダのフィールド

フィールド名	幅	意味および説明
バージョン	4bit	IPv6を表す「0110(2進数表示)」になる。
トラフィック・クラス	8bit	IPv4のサービス・タイプ(TOS)と同じで、パケット送信時のQoS (Quality of Service) を指定する。パケット送信時の優先度を確保する。
フロー・ラベル	20bit	マルチキャスト通信などにおいて、通信経路の品質を確保したり、経路を優先的に選択させたりするために使用する。IPv4ヘッダには該当するものはない(IPv4の場合は、より上位のプロトコルと組み合わせて通信路の経路を確保するのが一般的)。
ペイロード長	16bit	拡張ヘッダとペイロード・データの合計サイズ。IPv4と異なり、IPv6の基本ヘッダ部分(40bytes)は含まない。
次ヘッダ	8bit	IPv6基本ヘッダに続く、拡張ヘッダや上位プロトコルのタイプ。拡張ヘッダが複数ある場合は、最初の拡張ヘッダのタイプを表す。以後のヘッダは、数値つなぎで並べる。IPv4の「プロトコル・タイプ」に相当するが、拡張ヘッダもペイロード(上位プロトコルのヘッダとそのデータ)もこのネクストヘッダを渡して面倒に並べる。
ホップ・リミット	8bit	IPv4のTTLと同じで、通過可能なルータの最大数を制限する。ルータを1つ通過するたびに1ずつ減算される。0になると「クォンタム」は破棄され、送信元に対してICMPv6の「hop limit exceeded in transit」が返信される。
送信元IPv6アドレス	128bit	送信元ノードのIPv6アドレス
あて先IPv6アドレス	128bit	送信先ノードのIPv6アドレス

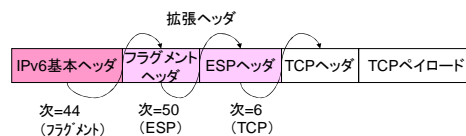
## IPv6の拡張ヘッダ

拡張ヘッダタイプ	番号	意味
ホップ・バイ・ホップ・オプション・ヘッダ	0	通常、中継ルータは拡張ヘッダを無視するが、このタイプが指定されている場合は、すべてのルータでこの内容进行处理する。このヘッダには追加のオプションを格納する
ルーティング・ヘッダ	43	途中で経由する必要があるノード(ルータ)を列挙する。パケットが通過するルータを限定する。IPv4のソース・ルーティングに相当
フラグメント・ヘッダ	44	フラグメント化(断片化)されたデータが含まれることを示す。IPv6ではフラグメント化は送信元でのみ行い、途中のルータでは行わない。IPv4のヘッダにあったIDフィールド(フラグメント化されたパケットを識別するための番号)は、この拡張ヘッダ中に存在する
ESPヘッダ	50	パケットの暗号化情報。IPv4ではIPやTCPなどは別に定義されていた暗号化/認証などの機能があり、IPv6ではIPレベルで統合されている
AHヘッダ	51	パケットの認証情報(同上)
ICMPv6ヘッダ	58	IPv6用のICMP
ネクストヘッダ	59	次のヘッダが存在しない、つまりヘッダ・チェーンの終了を表す
あらかじめオプション・ヘッダ・ヘッダ	60	通常、拡張ヘッダ部分は中継ルータでは関与しないが、このタイプが指定されている場合は、あらかじめノードと、(ルーティング・ヘッダで指定されている)すべてのルータでこのヘッダの内容进行处理すること
(プロコル番号)	上記以外	指定されたプロコルのヘッダとそのデータを表す。例えばTCPなら6、UDPなら17などとなる(IPv4のプロコル番号と同一)。

## IPv6パケットヘッダの特徴

- 以下の方法により、中継ルータの負荷軽減を図っている。
- ・ヘッダ・サイズ（基本ヘッダ）の固定長化（40bytes）
- ・ヘッダ・チェックサムの廃止
- ・ルータが処理すべきヘッダ情報種別の削減
- いつかのヘッダが基本ヘッダから外れ、拡張ヘッダ中へ移行している
- ・ルータによるフラグメンテーションの廃止（バスマTUを取得して送信元ホストが実施）

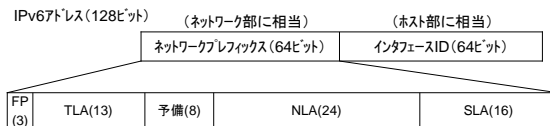
## 基本ヘッダと拡張ヘッダの関係



## Windows 7が管理するプロトコル番号

```
# protocol = 名称
# filename(ファイル名) | 機能名 | 形式記法 | 表示方法 | ヘルプ項目
# Copyright (c) 1993-2006 Microsoft Corp.
# This file contains the Internet protocols as defined by various
# RFCs. See http://www.iana.org/assignments/protocol-numbers.
# Format:
# <protocol name> <assigned number> [aliases...] [#comment:]
ip      0    IP          # Internet protocol
icmp    1    ICMP        # Internet control message protocol
igmp     2    IGMP       # Gateway-to-gateway protocol
tcp     6    TCP         # Transmission control protocol
udp     7    UDP         # User datagram protocol
st       8    ST         # Stream transport protocol
pawm    12   PAWM        # PASC universal packet protocol
vtp     17   VTP         # User Datagram Protocol
ospf    22   OSPF        # Open Shortest Path First
esp     24   ESP         # Encrypted Security Payload Protocol
xwp     25   XWP         # Xerox NS DAP
ns-ispd  26   NS-ISDP     # "NAT's Inside" protocol
ospf6   41   IPv6        # Internet protocol IPv6
ipv6-route 42 IPv6-Route  # Routing header for IPv6
ipsec-frag 43 IPsec-Frag  # Fragmentation for IPv6
ospf6   50   OSPF        # Open Shortest Path First
ah       51   AH         # Authentication Header
ipsec-icp 52 IPsec-ICP   # Internet Cryptographic Protocol
ipv6-nonet 59 IPv6-NoNet  # No next remote for IPv6
ospf6   60   OSPF        # Open Shortest Path First
vrrd    66   VRRD        # Designation options for IPv6
vrrd    66   VRRD        # Next remote virtual disk
```

## IPv6アドレスの構成とネットワークプレフィックス



FP: Format Prefix (アドレスの種類)  
 001 (2000:3): グローバル・ユニキャスト・アドレス (GUA) グローバルIPアドレスに相当  
 1111 110 (fc00:7): ユニーク・ローカル・ユニキャスト・アドレス (ULA) プライベートIPアドレスに相当  
 1111 1110 10 (fe80:10): リンク・ローカル・ユニキャスト・アドレス (隣接レベルでの通信のみ使用)  
 1111 1111 (ff00:8): マルチキャスト・アドレス (制御情報のためIPv6に使用)  
 TLA: Top Level Aggregator (これを取らずには大規模IP)  
 NLA: Next Level Aggregator (TLAの下位IP. 24bitを分割して階層構成が可能)  
 SLA: Site Level Aggregator (サイト内でのサブネットアドレス)

但し、この構成で運用すると、TLAに広大なアドレス空間を割り当てることになる。  
(経験が浅い現時点では、将来起こりうる、運用上の問題を見落としている懸念)

暫定的に、下記の構成で、TLA=0について、sTLA(sub TLA)を割り当てる運用を行っている。

FP (3)	TLA(13)	sTLA(13)	予備(6)	NLA(13)	SLA(16)
-----------	---------	----------	-------	---------	---------

WindowsにおけるIP情報の確認

C:\>ipconfig ...**IP情報の確認**

Windows IP 構成

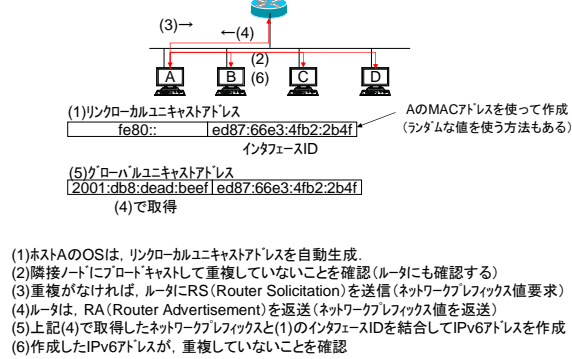
イーサネット アダプター ローカル エリア接続 2: ...**ネットワーク・インターフェイス**

接続固有の DNS サフィックス . . .  
リンクローカル IPv6 アドレス . . . : fe80::ed87:66e3:4fb2:2b4f%13 ...**IPv6情報**  
自動構成 IPv4 アドレス . . . . . : 169.254.43.79 ...**IPv4情報、APIPA**  
サブネット マスク . . . . . : 255.255.0.0  
デフォルト ゲートウェイ . . . . . :

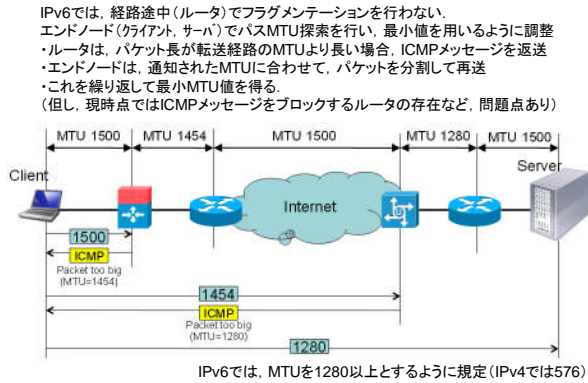
IPアドレスの割当てや手動設定がされていない場合、OSが、リンクローカルIPv6アドレスおよびAPIPAを自動的に生成して使用する。

APIPA: Automatic Private IP Addressing  
管理されていない小規模ネットワークで、OS同士でIPアドレスを自動割当てする機能。  
(169.254.0.0/16のホスト部をランダムに使用し、ARPで他との重複を自動検出)

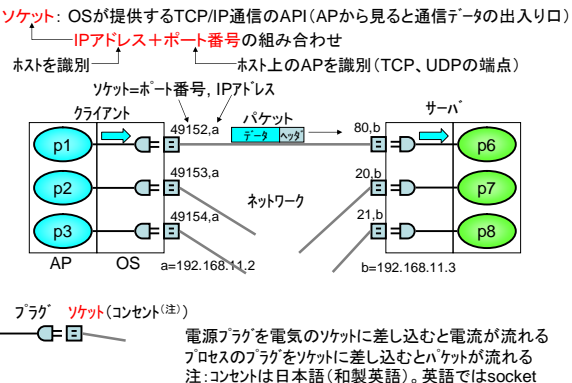
IPアドレスのステートレス自動設定



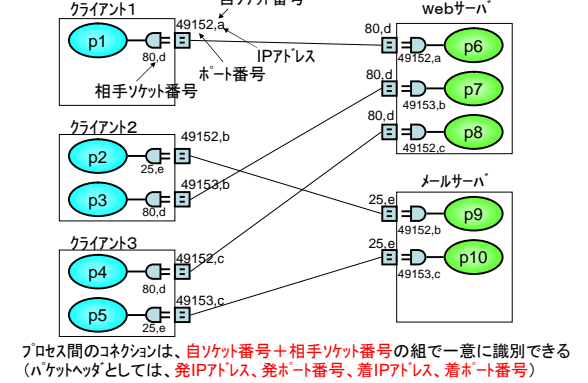
パスMTU探索



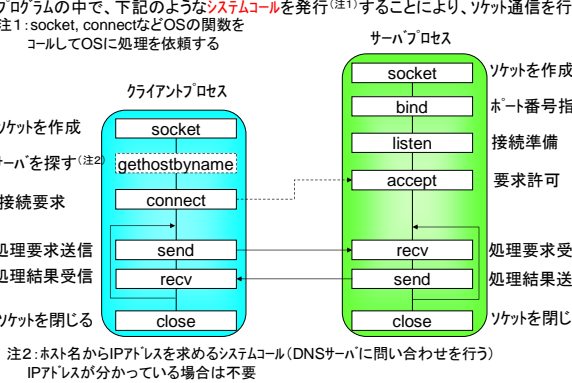
重要:ソケットを用いたプロセス間通信



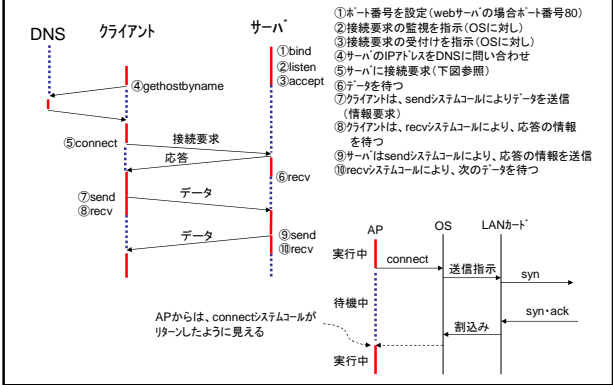
ネットワーク内での接続の識別



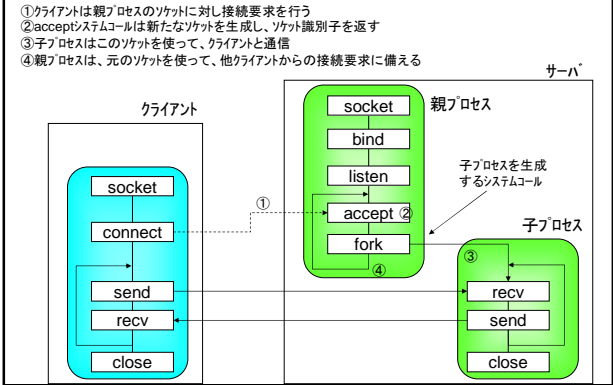
ソケットを用いた通信プログラムの流れ



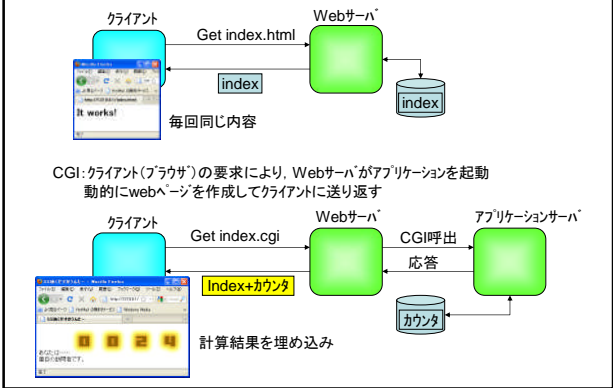
通信シーケンスと状態遷移



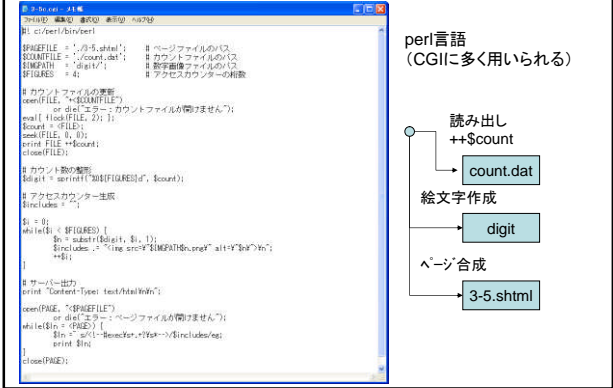
並行サーバのプログラムの流れ



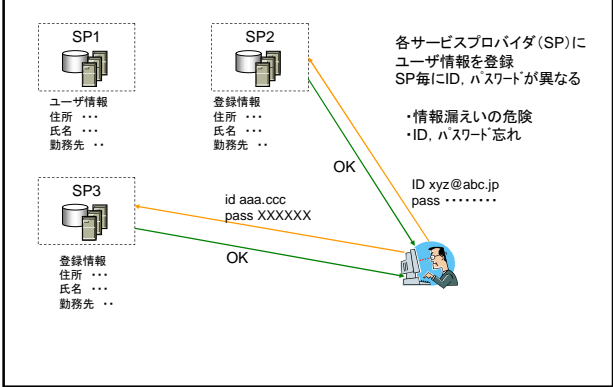
CGI (Common Gateway Interface)



アプリケーションのプログラム



サイト毎の登録と認証



OpenIDによる認証

