

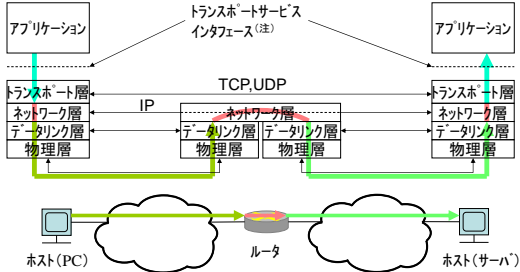
第14回 ネットワーク制御

期末試験は電卓のみ持ち込み可
(PC、スマートフォンなどは不可)

OSにおけるネットワークの位置づけ

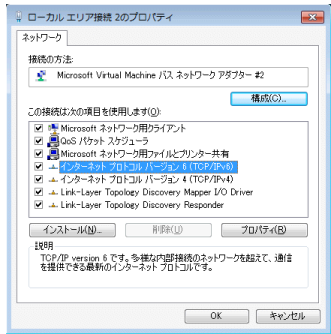
- ネットワーク＝入出力装置 (通信)
 - 端末や他ホストとの間で、通信回線を介して入出力する
 - OSは通信制御機能を提供する
- ネットワーク＝論理的な通信路
 - OS参照モデル (ネットワークアーキテクチャ) に基づくAP間通信
 - OSはAPIとしてトランスポートサービスを提供 (ソケットインタフェース)
- ネットワーク＝共有資源
 - LANを介してプリンタやファイルを共有
 - OSは共有資源へのアクセス機能を提供 (狭義のネットワークOS)
- ネットワーク＝プログラム＋計算機能
 - プログラムが、他コンピュータに処理をさせ、結果を返してもらう
 - OSは遠隔手続き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

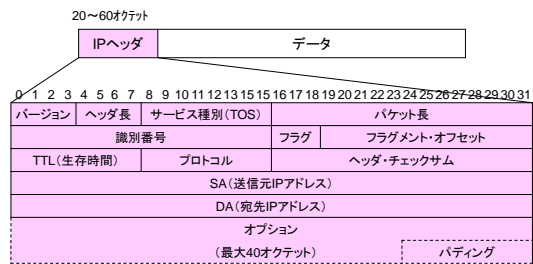


注: トランスポートサービスインタフェースは、通信プロトコルとしては、規定対象外。しかし、オペレーティングシステムでは、APIとして、明確に定義する必要がある。
TCP/IPのトランスポートサービスを提供するAPIが、ソケットである。最初に、Unixにおいて提供され、広く使われるようになった (Windowsでも採用されている)

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つ通過するごとに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
あて先IPアドレス	32bit	送信先ノードのIPv4アドレス

WindowsにおけるIP情報の確認

C:\>ipconfig ...IP情報の確認

Windows IP 構成

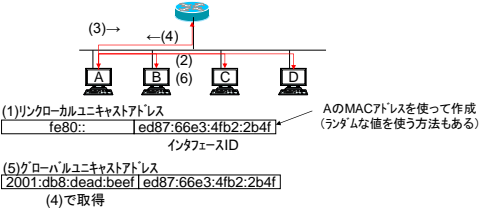
イーサネット アダプター ローカル エリア接続 2: ...ネットワーク・インターフェイス

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アドレスの割当てで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)

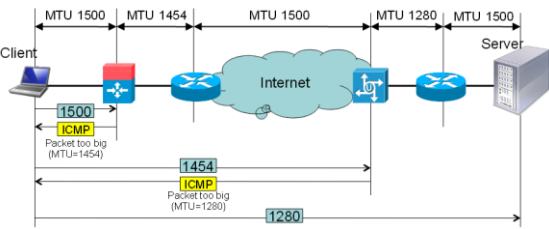
IPアドレスのステートレス自動設定



- (1)ホストAのOSは、リンクローカルユニキャストアドレスを自動生成。
- (2)隣接ノードにブロードキャストして重複していないことを確認(ルータにも確認する)
- (3)重複がなければ、ルータにRS(Router Solicitation)を送信(ネットワークプレフィックス値要求)
- (4)ルータは、RA(Router Advertisement)を送信(ネットワークプレフィックス値を返送)
- (5)上記(4)で取得したネットワークプレフィックスと(1)のインタフェースIDを結合してIPv6アドレスを作成
- (6)作成したIPv6アドレスが、重複していないことを確認

パスMTU探索

IPv6では、経路途中(ルータ)でフラグメンテーションを行わない。
エンドノード(クライアント、サーバ)でパスMTU探索を行い、最小値を用いるように調整
・ルータは、パケット長が転送経路のMTUより長い場合、ICMPメッセージを返送
・エンドノードは、通知されたMTUに合わせて、パケットを分割して再送
・これを繰り返して最小MTU値を得る。
(但し、現時点ではICMPメッセージをブロックするルータの存在など、問題点あり)



IPv6では、MTUを1280以上とするように規定(IPv4では576)

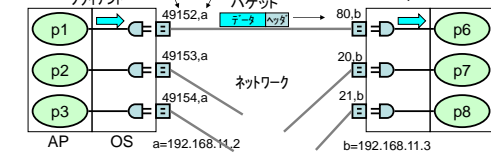
重要:ソケットを用いたプロセス間通信

ソケット: OSが提供するTCP/IP通信のAPI(APから見ると通信データの出入り口)

IPアドレス+ポート番号の組み合わせ

ホストを識別

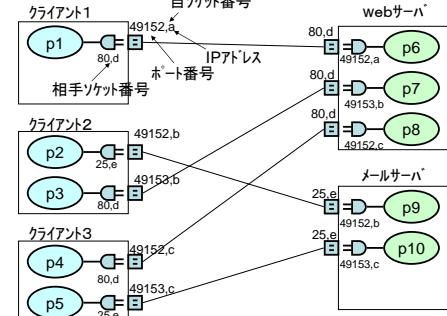
ホスト上のAPを識別(TCP、UDPの端点)



プラグ ソケット(コンセント)

電源プラグを電気のソケットに差し込むと電流が流れる
プロセスのプラグをソケットに差し込むとパケットが流れる
注:コンセントは日本語(和製英語)。英語ではsocket

ネットワーク内での接続の識別



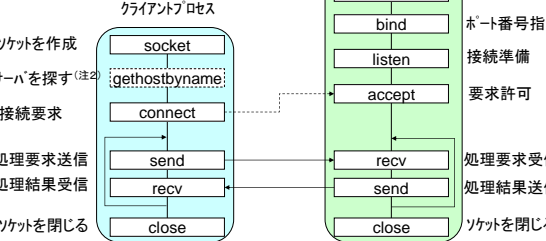
プロセス間の接続は、自ソケット番号+相手ソケット番号の組で一意に識別できる
(パケットヘッダとしては、発IPアドレス、発ポート番号、着IPアドレス、着ポート番号)

ソケットを用いた通信プログラムの流れ

プログラムの中で、下記のようなシステムコールを発行(注1)することにより、ソケット通信を行う

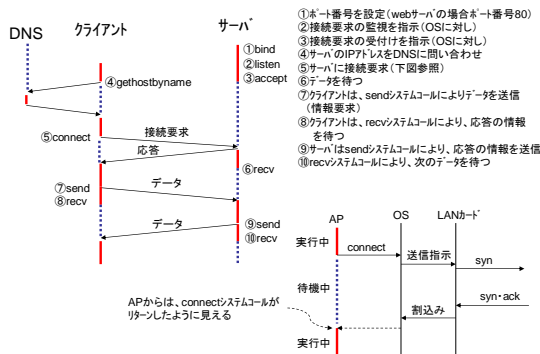
注1: socket, connectなどOSの関数を

コールしてOSに処理を依頼する

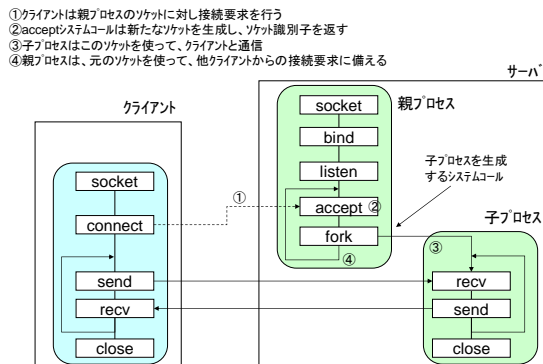


注2: ホスト名からIPアドレスを求めるシステムコール(DNSサーバに問い合わせを行う)
IPアドレスが分かっている場合は不要

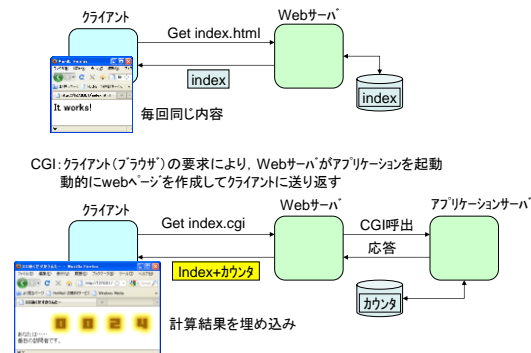
通信シーケンスと状態遷移



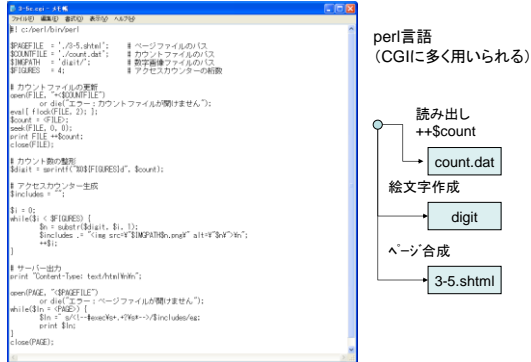
並行サーバのプログラムの流れ



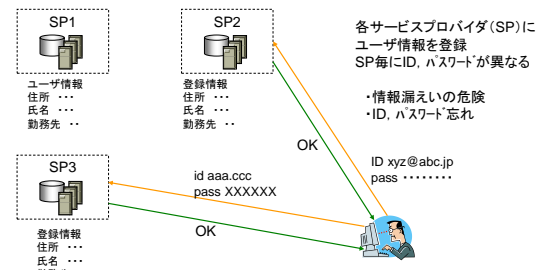
CGI (Common Gateway Interface)



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

