講義日程予定

- 第1回「ガイダンス」
- 第2回「次世代ネットワーク技術：概論」
- 第3回「ゲスト講演(1)：富士通のユビキタス事業」
- 第4回「次世代ネットワーク技術：ホームネットワーキング技術」
- 第5回「次世代ネットワーク技術：インターネットワーキング技術」
- 第6回「次世代ネットワーク技術：移動体通信, IPv6, P2P
- 第7回「センシング技術：RFIDと携帯端末」
- 第8回「センシング技術：測位技術」
- 第9回「ゲスト講演(2)：スカイリーネットワークスの技術」
- 第10回「センシング技術：センサーネットワーク技術」
- 第11回「サービスアーキテクチャ：基盤ソフトウェア」
- 第12回「ゲスト講演(3)：内田洋行のユビキタス技術」
- 第13回「サービスアーキテクチャ：XML技術」
- 第14回「サービスアーキテクチャ：プライバシとセキュリティ」
- 第15回「期末定期試験」

2007年度前期
情報システム構成論2
第3回「インターネットワーキング技術」

西尾 信彦

nishio@cs.ritsumei.ac.jp

立命館大学 情報理工学部

把握しておきたい事項

- IPアドレスの枯渇問題
  - プライベートアドレスおよびNATの利用
  - IPv6への移行
- ネットワークセキュリティ
  - 本人認証と暗号化
  - IPsecとIKE
  - SSL/TLS
  - ファイアウォール
  - VPN
  - セキュリティOS (SELinux, AppArmor)

TCP/IP = インターネット技術の総称

- 狭義の意味でのTCP/IP
  - TCPとIPの二つのプロトコルのみを指す
- 広義の意味でのTCP/IP
  - IPを利用する通信で利用されるプロトコルの総称
    - IP,ICMP,ARP,TCP,UDP,RIP,HTML,SMTP,FTP,etc
- インターネットの標準プロトコル群

TCP/IPネットワークの歴史

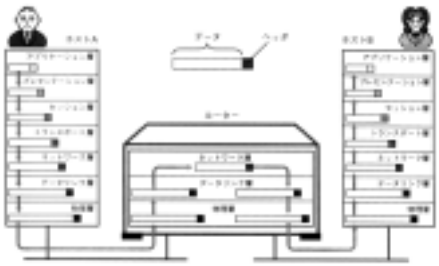
1960年代後半	アメリカ国防総省による軍事ネットワーク研究開始
1969年	ARPANET誕生 パケット交換技術の確立
1972年	ARPANET実験成功 50ノード以上にまで拡大
1975年	TCP/IP誕生
1982年	TCP/IP仕様決定 BSD UNIXの無料提供開始 BSD UNIXがTCP/IPを実装していたことにより急速に拡大
1983年	ARPANETの正式プロトコルにTCP/IPを採用
1989年ごろ	LAN上でTCP/IPの利用が急拡大
1990年ごろ	LAN、WAN共にTCP/IPを利用する方向へ
1995年ごろ	インターネットが一般的になる　ISPが多数発足
1996年	次世代IP、IPv6の仕様が決定　RFCに登録

ISO 7階層モデル

アプリケーション層	特定のアプリケーションで利用されるプロトコル
プレゼンテーション層	機器やネットワークの固有データフォーマットを制御
セッション層	コネクションの確立 下位層の管理
トランスポート層	両端ノード間のデータ転送管理 信頼性の提供
ネットワーク層	アドレスの管理 経路選択
データリンク層	直接接続された二つの機器間での通信制御
物理層	電圧や光の明滅と0、1のデジタル信号間の変換

ISO 7階層モデルの動き

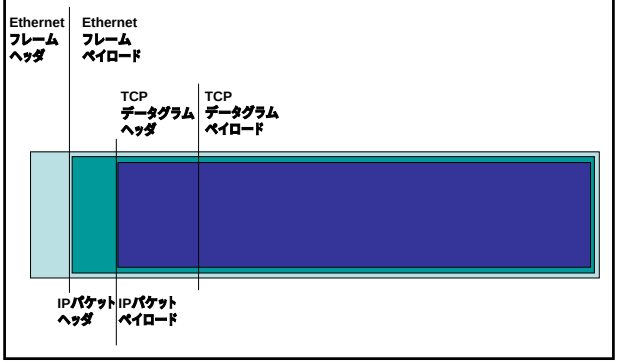
- ISO 7階層モデル上での動き



TCP/IPモデル

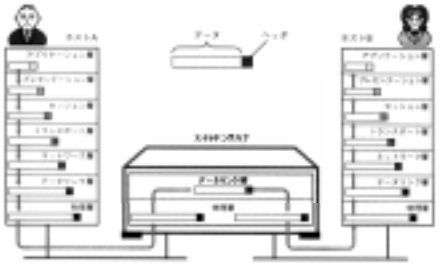
ISO7階層モデル	TCP/IPモデル
アプリケーション層	アプリケーション層 HTTP,SMTP,TELNET, FTP,SNMP,MIME,HTML,MIB
プレゼンテーション層	
セッション層	
トランスポート層	トランスポート層 TCP,UDP
ネットワーク層	インターネット層 ARP,IP,ICMP
データリンク層	データリング・物理層付近 イーサネット、WiFi、ATM、FDDI、etc
物理層	

TCP/IPネットワークパケット構造



ISO 7階層モデルの動き

- ISO 7階層モデル上での動き

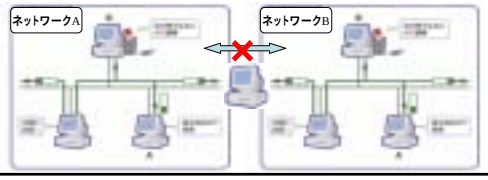


MACアドレス

- IPを利用するネットワーク機器全てに割り当てられた、世界で一意なアドレス
- ROM内に焼きこまれている48bitアドレス
- 例：00-90-CC-20-93-D0

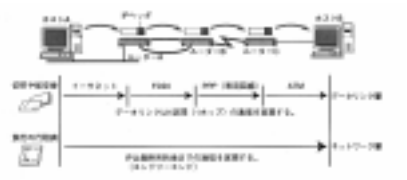
EthernetとMACアドレス

- 同一ネットワーク内でMACアドレスを利用して通信を行う
- 複数のネットワークを越えて通信できない
  - ゲートウェイ(複数のネットワークに接続したホスト)を超えては通信できない



IP (Internet Protocol)

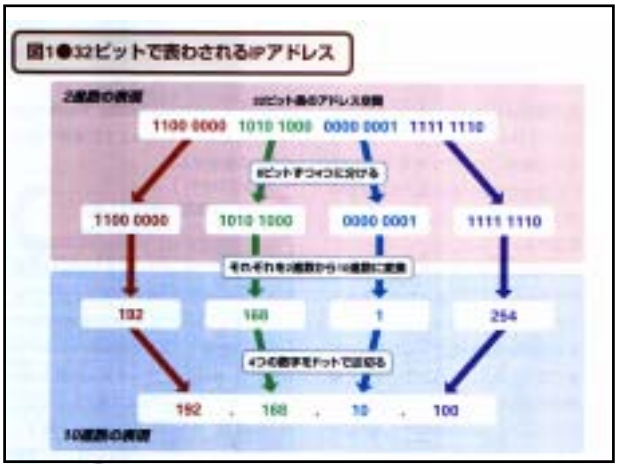
- 直接接続されていない(複数のネットワークを経由する)機器同士の通信の確立
- インターネット上のホストからホストまで届ける
  - 目的地のホストに辿りつくまで情報を管理し、
  - そこまで届ける手法を用いて到達する



IPアドレス

- 世界中のホストからホストまで届ける「宛名」
- 上位のネットワークアドレスと下位のホストアドレスから構成される
- 32bitの二進数整数値
- 8bitずつに区切って10進数で表現する方法が一般的
- 最大数 2^{32} = 4,294,967,296

	2^8	2^8	2^8	2^8
二進数	10101100	00010100	00000001	00000001
十進数	172	.20	.1	.1



IPアドレスのクラス(1)

- 配布するネットワークの規模に合わせてクラスがある
- クラスA
  - 先頭1bitが0で始まるアドレス
  - 0.0.0.0 〜 127.0.0.0まで
  - 16,777,214個の割り当てが可能
- クラスB
  - 先頭2bitが10で始まるアドレス
  - 128.0.0.0 〜 191.255.0.0まで
  - 65,534個の割り当てが可能

IPアドレスのクラス(2)

- クラスC
  - 先頭3bitが110で始まるアドレス
  - 192.0.0.0 〜 223.255.255.0まで
  - 254個の割り当てが可能
- クラスD
  - 先頭4bitが1110で始まるアドレス
  - 224.0.0.0 〜 239.255.255.255まで
  - ホストアドレスではなくIPマルチキャストに利用

ネットワークアドレス

- ネットワーク自体を表す
- クラス内のホストアドレスbitが全て0のアドレス
- 逆に全て1ならブロードキャストアドレス
  - それ以外が実際にホストにつけられるアドレス
- 10.0.0.0 (クラスA)のネットワークのネットワークアドレスは
  - ホストアドレスである下位24ビットがすべて0なので
  - 10.0.0.0

ブロードキャストアドレス

- クラス内のホストアドレスbitが全て1のアドレス
- クラス内の全てのアドレスに送信される
- 例：172.20.0.0 クラスB(下位16bitがホスト)
  10101100.00010100.00000000.00000000
  - ブロードキャストアドレス 172.25.255.255
  10101100.00010100.11111111.11111111
- クイズ：133.19.7.0/27(下位5ビットがホストアドレス)のネットワークのブロードキャストアドレスは？

マルチキャストアドレス

- 先頭4bitが1110の場合、下位28bitによって規定されたマルチキャストが行われる
- 例：
  - 224.0.0.0　予約(利用できない)
  - 224.0.0.1　サブネット内の全てのシステム
  - 224.0.0.2　サブネット内の全てのルータ
  - 224.0.0.14 DHCPサーバ/リレーエージェント

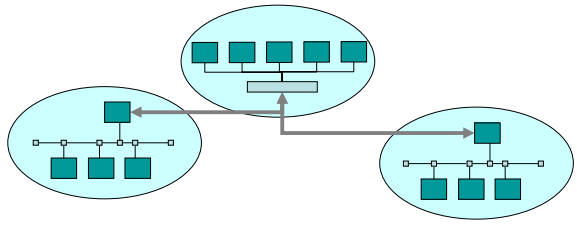
IPアドレスの分配

- ICANNが全世界で一元管理
  - (Internet Corporation for Assigned Names and Numbers)
- 日本ではJPNICが管理
  - (Japan Network Information Center)
- 申請してIPアドレスを取得する
- 立命館
  - クラスB 133.19.0.0
  - 210.166.170.128/26

サブネットワーク

- クラスのみで管理するのは現実的ではない
- クラスは冗長
  - 必ず、一つのリンク内に全てのマシンを接続する必要がある。
  - クラスBを利用すると、3台しかつながない場合でも、必ず65.534分のIPアドレスを消費
- 可変長サブネットマスクの導入
  - VLSM (Variable Length Subnet Mask)
  - サブネットマスクをクラスの代用として利用

ネットワークのイメージ



- インターネットはネットワークの単位で処理されるが
- ネットワーク内がフラットだとこれもまた大変

サブネットマスク利用例

IPアドレス　　172. 20. 1. 0 (クラスB)

10101100.00010100.00000001.00000000

ネットワーク　サブネットワーク　ホストアドレス

サブネットマスク 255.255.255. 0

11111111.11111111.11111111.00000000

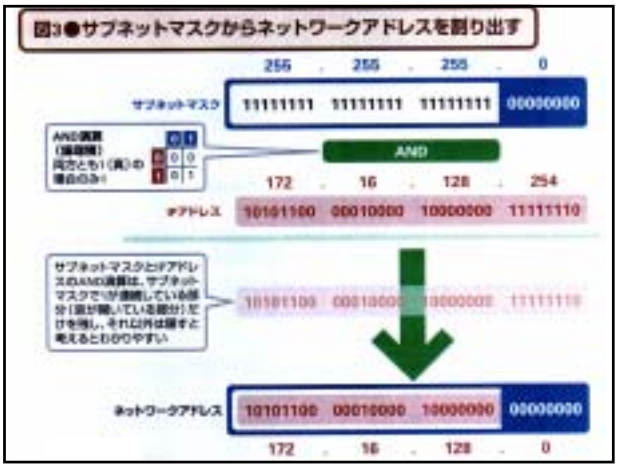
– 利用可能範囲 172.20.1.0〜172.20.1.255

– Maskで0の部分がホストアドレスとして利用可能部位

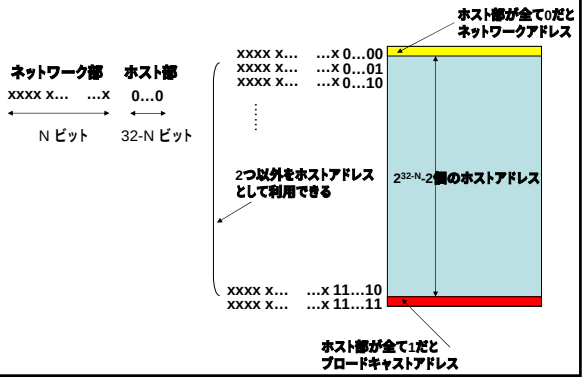
表記法

172.20.1.0/255.255.255.0

172.20.1.0/24 (こちらが一般的によく使用される)



一般化して X.Y.Z.W/Nのサブネットワークでは

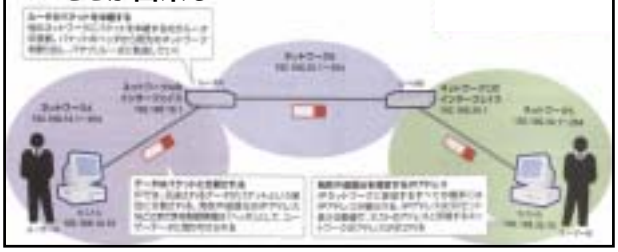


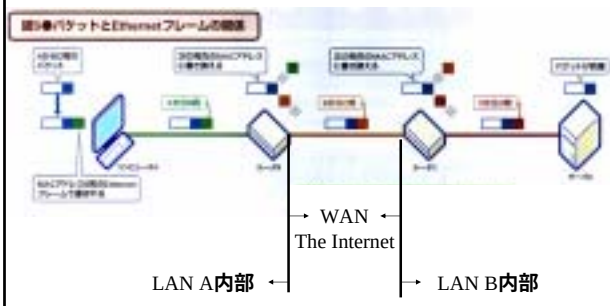
例題:133.19.7.176/28のサブネットワーク

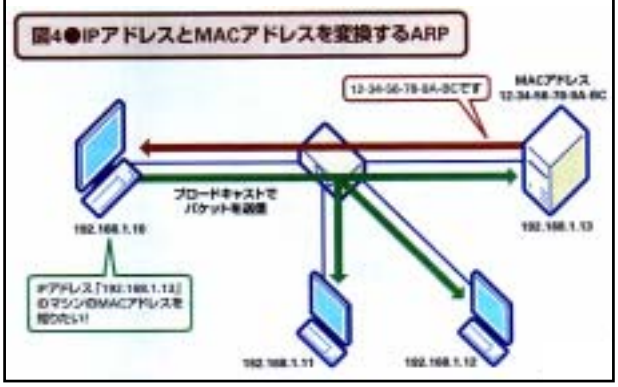
- 第4バイトの176の上位4(=4*8-28)ビットはネットワークアドレス部分
  - 128-64-32-16-8-4-2-1なので176=128+32+16
  - よって 第4バイトの176は1-0-1-1-0-0-0-0
  - 133.19.7.1-0-1-1-0-0-0-0の　がネットワーク部分
- ネットワークアドレス133.19.7.176
  - ホストアドレス部分がすべて0
  - 133.19.7.1-0-1-1-0-0-0-0
- ホストアドレスは133.19.7.177から133.19.7.191まで
  - ホストアドレス部分が1から14までの14ホストを収容可能
  - 133.19.7.1-0-1-1-0-0-0-1から133.19.7.1-0-1-1-1-1-1-0
- ブロードキャストアドレスは133.19.7.192
  - ホストアドレス部分がすべて1
  - すなわち 133.19.7.1-0-1-1-1-1-1-1

ネットワークを越えた通信:IPの機能

- IPアドレスを利用して通信を行う
- 複数のセグメントを通じて、マシンを特定することが出来る

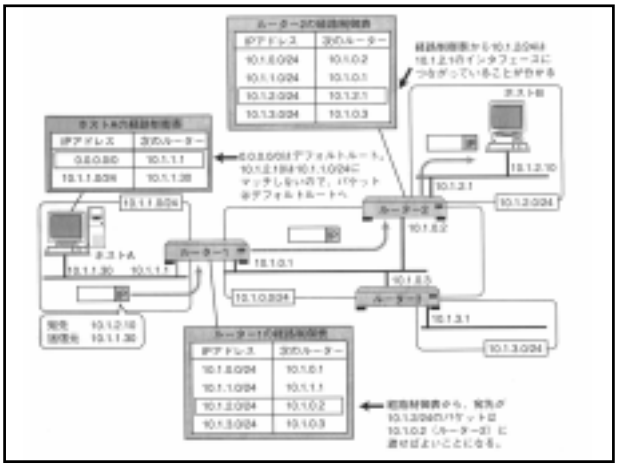






IPルーティング

- 経路制御表(ルーティングテーブル)
  - IPアドレスと配送先ネットワークの相対表
  - Radixテーブル方式(アドレスの最長一致検索が可能)
- デフォルトルート
  - 経路制御表にないアドレスの配送先
- ループバックアドレス
  - 自分自身を指すアドレス
  - 実際のネットワークにはパケットは流れない



netstat –rn：経路情報出力

```
-bash-2.05b$ netstat -rn
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif Expire
default            133.19.7.30        UGS         0 22127138   bge0
127.0.0.1          127.0.0.1          UH          0 3846233724    lo0
133.19.7/27        link#1             UC          0         0   bge0
133.19.7.1         00:0d:0b:09:1c:d3  UHLW        0       533   bge0   1020
133.19.7.2         00:09:6b:a3:9e:30  UHLW        0      2464    lo0 =>
133.19.7.2/32      link#1             UC          0         0   bge0
133.19.7.3         00:09:6b:a3:9e:30  UHLW        0 11796854    lo0
133.19.7.4         00:03:2d:00:80:88  UHLW        0 288802491   bge0   1196
133.19.7.7         00:a0:b0:70:c8:aa  UHLW        0  3462308   bge0    169
133.19.7.27        00:08:74:1c:9c:d8  UHLW        0       25   bge0   1189
133.19.7.30        00:a0:de:3a:04:31  UHLW        1      600   bge0   1024
192.168.0/16       link#2             UC          0         0   bge1
192.168.0.7        00:09:6b:a3:9e:31  UHLW        0   124286    lo0
192.168.0.11       00:e0:18:59:d6:f5  UHLW        0   136791   bge1    173
192.168.1.10       00:19:d1:45:b2:50  UHLW        0        3   bge1    822
192.168.2.31       00:17:31:38:4a:0f  UHLW        0        2   bge1    599
192.168.2.50       00:0a:e4:82:57:af  UHLW        0     2116   bge1   1101
192.168.4.3        00:0d:56:05:aa:86  UHLW        0     1926   bge1    851
192.168.5.50       00:01:80:61:67:a4  UHLW        0      169   bge1   1193
192.168.5.98       00:13:02:40:52:cc  UHLW        0       20   bge1   1131
192.168.5.114      00:13:d1:e8:bc:98  UHLW        0     1001   bge1   1130
```

IPルーティング

- AS (Autonomous System)単位で分割
  - ritsumei.ac.jpは一つのASの例
- AS内のルーティング
  - RIP, OSPF, etc.
- AS間のルーティング
  - BGP4 (Boarder Gateway Protocol version 4)

名前解決とDNS

- www.ritsumei.ac.jpのようなドメイン名前からIPアドレスへの変換作業を名前解決という
- DNSというサーバ群が全世界を階層的に管理している

IPアドレスの配布方式

- 個々に設定
  - IPアドレス
  - サブネットマスク
  - デフォルトルート
  - プライマリDNS，セカンダリDNS
- 自動設定
  - DHCP (Dynamic Host Configuration Protocol)
  - ネットワークに接続したホストはDHCPサーバを探し，そこからアドレスなどをリース
  - IP通信が成立する前に通信できる必要があるので，IPの横にならぶプロトコルである

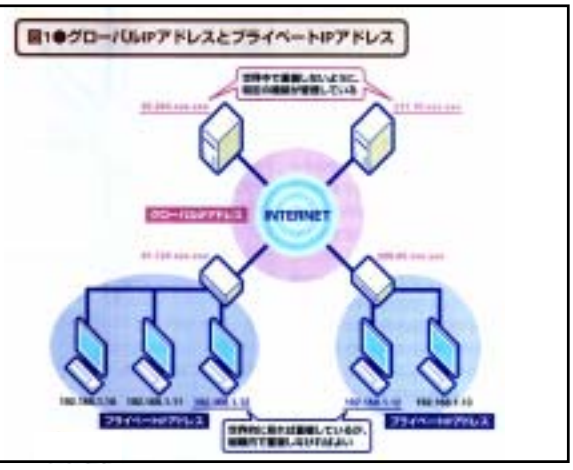
IPアドレス枯渇問題

- 全世界でたったの43億個弱しかない
- すでにIPアドレスは枯渇済み

- 解決策
  - プライベートIPアドレスの利用
  - 新しいプロトコルの開発(IPv6)

プライベートIPアドレス

- インターネットに『直接接続していない』ネットワークで自由に利用可能なアドレス
  - 10. 0. 0. 0 ～ 10.255.255.255 (10.0.0.0/8)
  - 172. 16. 0. 0 ～ 172. 31.255.255 (172.16.0.0/12)
  - 192.168. 0. 0 ～ 192.168.255.255 (192.168.0.0/16)
- プライベートネットワーク内でのサブネッティングも可能
- それぞれクラスABCに相当するが
  - クラスCに注目
  - CIDR (Classless Inter-Domain Routing)



NAT(Network Address Translator)

- プライベートIPアドレスを広域ネットワークに接続する方法
- 本当はポートも変換するNAPTを利用する

IPv6 (Internet Protocol version 6)

- 128bit長のアドレスを利用
- 利用可能個数 約3.40×10^{38}

- IPv4の機能改善
  - パフォーマンスの向上
  - IPアドレスの自動割当
  - 認証機能、暗号化機能の提供

TCP
(Transmission Control Protocol)

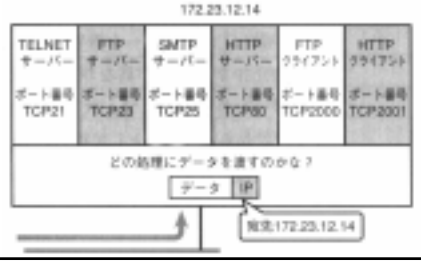
- コネクションを確立する
- 信頼性のある通信を実現
- データの破壊、パケットの損失、到着順序の整合性などを検出、補正する
  - シーケンス番号
  - 確認応答処理
  - 再送
- 信頼性の必要なサービスで利用される

UDP
(User Datagram Protocol)

- コネクションを確立しない
- 信頼性のない自由な通信を実現
- 信頼性を確保しないため、高速に動作
- 利用例
  - 動画、音声などのマルチメディア通信
  - 同報性を必要とする通信
    (マルチキャスト、ブロードキャスト)

IPアドレスとポート

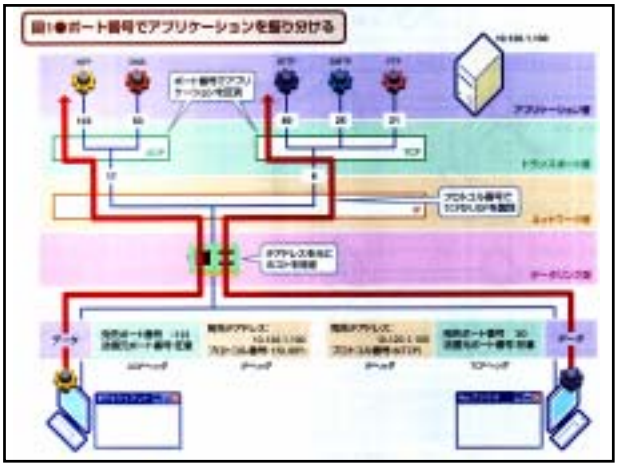
- TCP/UDPで上位層のアプリケーションを特定するために利用

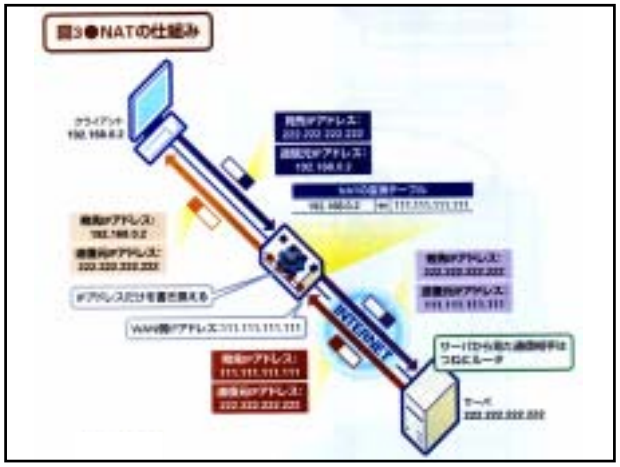


Well-known Port Number

- サービスが公知のポート
- 例

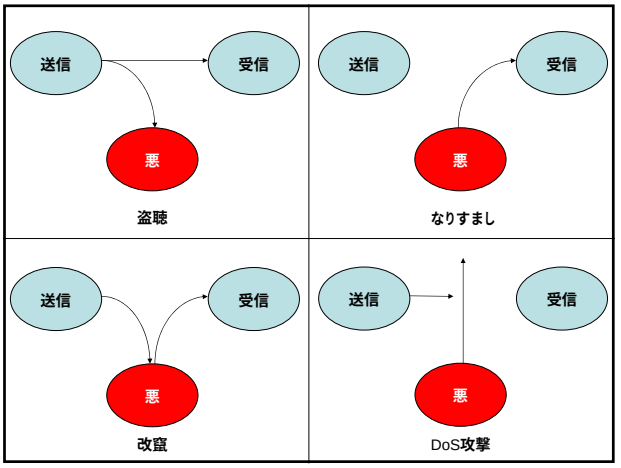
Port	名称	内容
20	ftp-data	File Transfer [Default Data]
21	ftp	File Transfer [Control]
22	ssh	SSH Remote Login Protocol
23	telnet	Telnet
25	smtp	Simple Mail Transfer Protocol
80	http	World Wide Web HTTP





ネットワークセキュリティ問題

- 盗聴
  - 通信の秘匿性(暗号化)で対応
- 詐称(なりすまし)
  - 本人性の保証(認証)で対応
- 改竄
  - 通信の完全性(改竄検出)で対応
- DoS攻撃
  - アクセスコントロール機能
- コンピュータウィルス
  - ウィルス･セキュリティ機能



ネットワークは信用できない

- LANはホストの追加が非常に簡単
  - 無線LANは自由に接続が可能
  - 正規のクライアントのみが接続している保証がない

有線LANもコネクタを繋ぎ直せば、
誰でもネットワークに参加できる

- インターネットはそもそも信用できない

DHCPの問題

- DHCP
  (Dynamic Host Configuration Protocol)
  - IPアドレスを動的に配布する
  - 要求があり、空きがあれば自動的に配布され、正規ユーザ、不正規ユーザを見分ける機構をもっていない
  - MACアドレスを指定して限定することは可能

ネットワーク参加時にユーザ認証をする方法

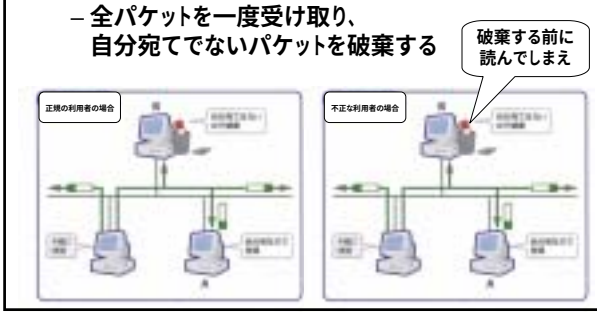
- MACアドレス詐称は可能
  - DHCPの機構では、参加を防ぎきれない
- IEEE802.1xを利用する
  - IDとパスワードを利用
  - 正規のユーザの場合のみIPアドレスを配布
  - ユーザ認証情報を保持するサーバを利用
    RADIUS(Remote Access Dial In User Service)

盗聴・改ざん・詐称問題

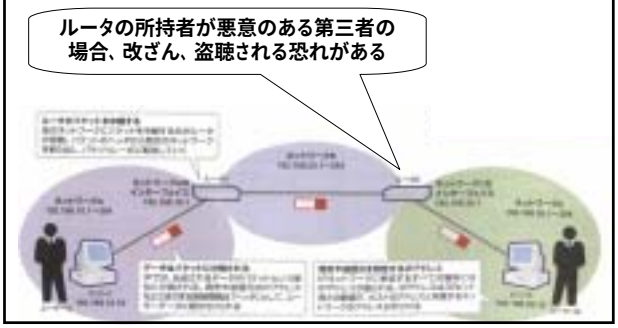
- Ethernetの問題
  - IEEE802.1xを利用しても、IPアドレスを手動で取得、設定されれば意味がない
- IPの問題
  - インターネットでは、ルーティング経路が全て悪意のないホストではない可能性がある
  - 途中で改ざんされても、それを知る機構が存在しない

Ethernet盗聴

- Ethernetの動き
  - 全パケットを一度受け取り、自分宛てでないパケットを破棄する



IPパケット改ざん・詐称

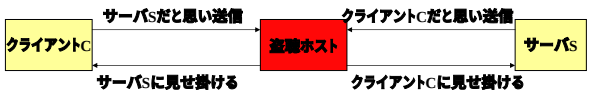


盗聴・改ざん・詐称の被害

- パスワードなどを平文で送信した場合に、パスワードを見ることが出来る
  - パスワードアタックに利用される
    - もっとも原始的で、非常に有効な攻撃方法
- メールの内容や、見ているWebページなどの内容を全て取得することが出来る
  - プライバシーや機密の問題

盗聴・改ざん・詐称の被害

- システムを詐称して，パスワードを入力させる．
- ユーザのリクエストを変更する．
- Man-in-the-middle攻撃という



Switching Hubによる盗聴防止

- 近年のネットワークのほとんどがSwitching Hub方式
- 共有バスではないので盗聴されない
- 実はARP spoofing(poizoning)により盗聴の可能性もある
- さらにVLAN(仮想LAN)の構築も可能
- おかげでSniffingやtcpdump, ethereal などが利用できなく(Switchを越えられなく)なっている.

WEP (Wired Equivalent Privacy)

- 無線LANでの通信路の暗号化
- RC4アルゴリズムを利用した秘密鍵暗号方式
- 40bit 方式で制定されたが, 近年では128bit方式が採用されている
- さらにWPAが制定されている

IPSecを利用する

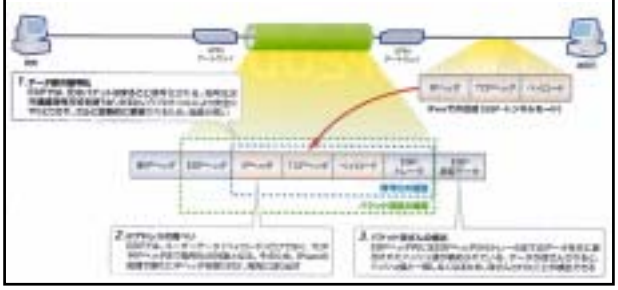
- ESP(Encapsulation Security Payload)とAH(Authentication Header)
- 主流はESP, 以下ESPについて
- IPレベルでパケットを暗号化
- IPSecで可能なこと
  - データ部の暗号化
  - IPアドレスの隠蔽
  - パケット改ざんの検出
  - 送信元成りすましの回避
- IPv6にはデフォルトで組込まれている

暗号鍵方式

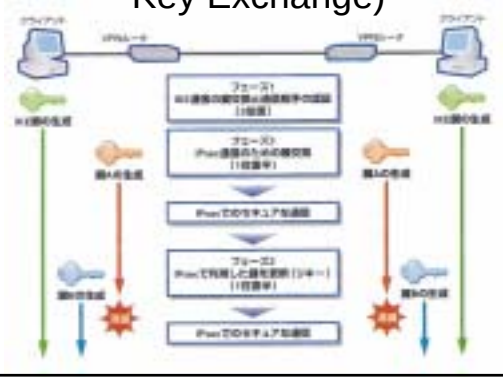
- 秘密鍵暗号
  - 事前に秘密鍵を共有しておく必要がある
  - 暗号化複合化が高速
  - IPsecの通信路暗号化に適用
- 公開鍵暗号
  - 秘密鍵と公開鍵のペアを生成
  - 一方の鍵で暗号化した場合、もう一方の鍵で複合化する
  - 秘密鍵を転送する必要はない
  - 暗号化のみならず本人認証にも利用できる
  - 公開鍵から秘密鍵を推定するのは困難
  - 処理が重い
  - IPsecの鍵交換に適用 IKE

IPSecの動き

- 暗号化とIPアドレス隠蔽(トンネリング)



IPSecの鍵の動き(IKE: Internet Key Exchange)

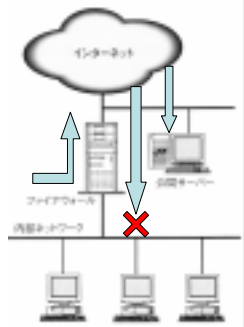


IPSecがあれば大丈夫？

- 特定ルート間の暗号化は可能だが、不特定ルート間の暗号化は出来ない
  - 例
    メールは不特定多数のメールサーバと通信するため、IPSecで暗号化することが出来ない（対象サーバが対応している保証が無いため）
- アプリケーション・プロトコルレベルでのセキュリティ対策も必要になる

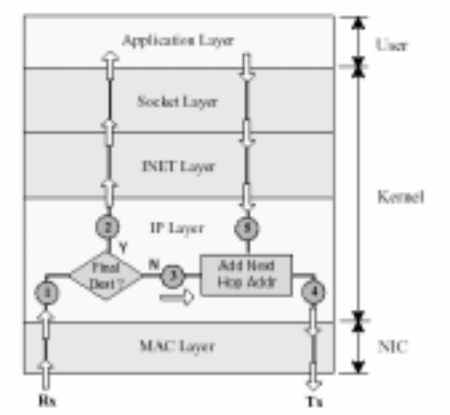
ネットワークの外と内

- 公開されている部分(外)
  - 外部からアクセス可能
  - 不正なアクセスや攻撃に遭う可能性有り
- 公開されない保護された部分(内)
  - 外部からのアクセスは不可能
  - 不正なアクセスや攻撃は不可能
  - ただし「基本的には」という話



ファイアウォールの種類と特徴

- パケットフィルタリング型
  - 第三層(TCP/IP)によるフィルタリングにより高スループットが期待できる
  - 第三層によるフィルタリングのため、上位層に依存しない(どのようなサービスでも対応可能)
  - ほぼ全てのLinuxで標準搭載
  - 設置が簡単
  - 文献が非常に豊富



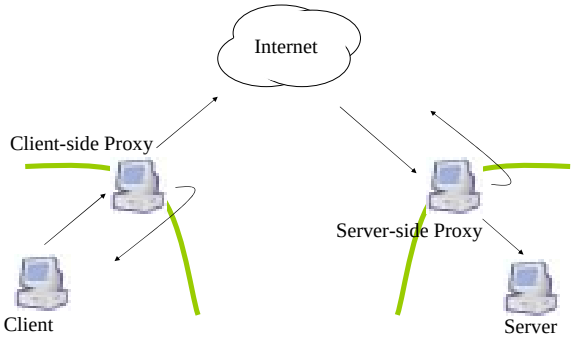
ファイアウォールの種類と特徴

- アプリケーション・ゲートウェイ型
  - 第7層でのアプリケーションの需要に適応したフィルタ
  - 多種多様、選択の幅は広い
  - 要求の内容、要求対象の内容などを利用しての細かいフィルタリングが可能
  - 要求を一つ一つ高層まであげてチェックするため、スループットは低い
  - ユーザ認証などを利用することも可能
  - サービス毎に対応するため個々にゲートウェイが必要
    - 複数のサービスを設置する場合、複数のゲートウェイが必要になる場合もある
  - 多くが商用のものであり、
  - サポートが期待できるが高価

ファイアウォールの種類と特徴

- Proxy型
  - アプリケーション・ゲートウェイ型の一つ
    - 基本的な利点・欠点は同じ
    - 通常唯一portを開放しているwebサービスに適用する
  - 簡易的なものから商用まで多種多様、選択の幅は広い
    - Apacheなどの簡易Proxyモードも利用できる
    - Proxyから専用のソフトを利用したもの
    - 複合タイプも存在する
  - HTTPなどの場合、要求をキャッシュすることで高スループットが期待できる

Dual-sided Proxy



ファイアウォール比較

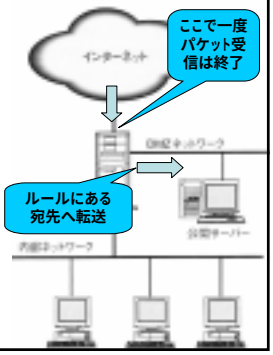
	パケットフィルタリング型	Proxy型	アプリケーション･ゲートウェイ型
スループット	◎	◎	△
クライアントから見たの透過性	◎	○	△
管理の簡便性	△	△	◎
ユーザ認証などによる制御	△	○	◎
閲覧内容や要求による制御	△	○	◎
ログの見易さ	△	△	◎
各種サービスへの適合度	◎	△	△
設置の安易さ	◎	○	△
文件数	◎	○	△

パケットフィルタリング型

- Linux kernel 2.2系列
  - ipchains
- Linux kernel 2.4系列以降
  - iptables
- BSD系
  - ipfw
- 以降のnetfilterはkernel 2.4系以降の話
- 基本的な機能
  - IP forwarding
  - パケットフィルタリング

IP Forwarding

- ネットワークインターフェイスから受信したIPパケットを、他のインターフェイスへ転送すること
- パケットの内容(port情報など)によってフォワード先を変更することでパケットフィルタリングを実行している
- 特定portに来たパケットのみ他のマシンへ転送することなども可能
- ネットワークの負荷分散などにも利用される
- NATの一種

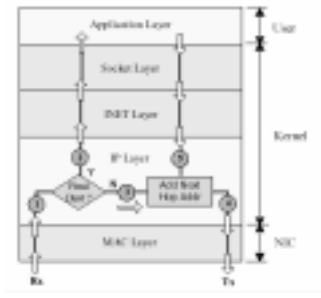


パケットフィルタリング

- ネットワークインターフェイスから受信したパケットを、事前に与えた条件と比較して無効なパケットであれば破棄する
- 基本的なフィルタリングポリシー例
  - すべてのパケットは危険であるので通さない
  - 明確に安全だと指定されたパケットは通過させる
    - ssh, imap/pop over sslなど
  - 既知の攻撃用ポートは閉じる(in-coming)
    - 445(blaster)など
  - 内部が汚染された場合のために外部の既知の攻撃ポートへのアクセスは禁止する(out-going)

netfilterのhookポイント

- 以下の五つのhookポイントでパケットを加工
  1. Pre Routing
  2. Input
  3. Forward
  4. Post Routing
  5. Output



Hookポイントの使い方

- Pre Routing
  - 最終受け取り判定前のため、最終受け取りアドレスを変更(NAT)し、他のマシンへ転送したりすることが可能
  - In-coming全体に対するフィルタリングなどはこのポイント
- Input (自分へのフィルタリング)
  - 最終受け取り後の入力時のhook
  - 不正アクセスなどはこの時点で判別し破棄する
- Forward (内部ホストへのフィルタリング)
  - 転送時のフィルタリングポイント
  - 他のマシンへ転送したくない(不正アクセス)パケットなどはこのポイントで破棄する

Hookポイントの使い方

- Post Routing
  - 最終出力前加工ポイント
  - 全体的な出力パケットに対するフィルタリングはこのポイントで行う
  - IPマスカレードの設定はこのポイントで行う
- Output
  - 標準出力時のhookポイント

NAT(Network Address Translation)

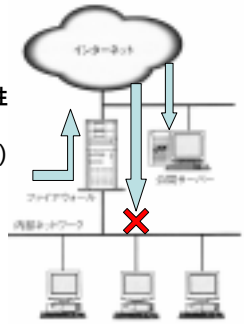
- ネットワークアドレス変換機能を提供する
- SNAT(Source NAT)
  - 送信元を書き換える
  - IPマスカレードはSNATの特化したもの
  - 返答パケットを他のマシンに受け取らせたりすることができる
- DNAT(Destination NAT)
  - 送信先を書き換える
  - 経路制御やパケットの種類による振り分けなどに利用する

IPマスカレード(NAPT)

- プライベートアドレスを利用している内部ネットワークから外部へ出て行くパケットの戻り先アドレスを、ファイアウォールの外部アドレスに変更し、パケットのセッション情報を保持することによって、内部のマシンをあたかもグローバルに存在しているかのように見せる方法
- 家庭でNATを導入する場合には必須の機能
- パケットのセッション情報(通過記録)が必要となる
- Linux kernel 2.4以降はnetfilterの設定が簡単に

ネットワークの外と内

- 公開されている部分(外)
  - 外部からアクセス可能
  - 不正なアクセスや攻撃に遭う可能性有り
- 公開されない保護された部分(内)
  - 外部からのアクセスは不可能
  - 不正なアクセスや攻撃は不可能

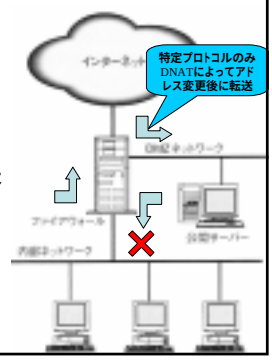


内外完全分離する上での問題

- 外部に置くサーバは全て、公開するサービス以外を完全に防御する必要がある
  - ファイアウォールや設定項目の増加
  - 作業量の増加
  - メンテナンス性の低下
- 外部からアクセス可能だがある程度守られたサーバが欲しい
- セキュリティ上、内部ネットにおくことには抵抗がある
  - 陥落した場合、内部ネットワーク全てが危険にさらされるため

DMZ(DeMilitarized Zone)

- 非武装地帯
- 外でも内でもない領域
  - 一部のプロトコルのみフォワーディングにより通過
  - DMZから内部ネットへのアクセスは基本的に禁止
  - 全体的なフィルタリングは大本のファイアウォールによって実行
    - メンテナンス性の向上
    - 設置作業量の減少



外部からのネットワーク利用

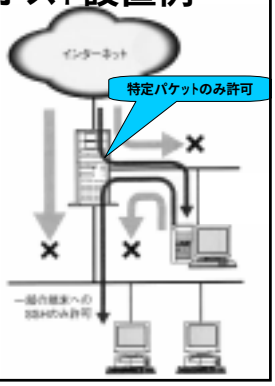
- サービスを設置する = セキュリティレベル低下
  - 不要なサービスは設置しない
  - 一番硬い守りは、何もサービスを稼動しないこと
- 特に、ネットワーク内部に侵入するリモートアクセスを可能にすることは、セキュリティレベルを大幅に下げることになる
- 利便性とトレードオフ

設置するべきではない危険な主要サービス

- 通信経路が暗号化されていない、かつパスワードなどが必要になる
  - リモートアクセス
    - telnet, rsh, rlogin, etc
    - 対策:ssh, VPNなどを利用する
  - メール関連
    - smtp, pop, imap
    - 対策:smtp over SSL, pop/imap over SSLを利用する
  - ファイル転送
    - ftp
    - 対策:sshを利用しscpを活用

リモートアクセスホスト設置例

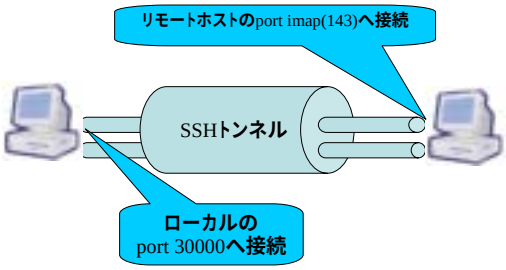
- リモートアクセス用ホストをDMZに置き、DMZから内部へのアクセスを制限
  - 侵入者にとっては障害が二倍になる
  - 可能ならば、DMZ内のリモートアクセス用ホストと内部にあるリモートアクセス用ホストは異なるパスワードを利用する



リモートアクセス

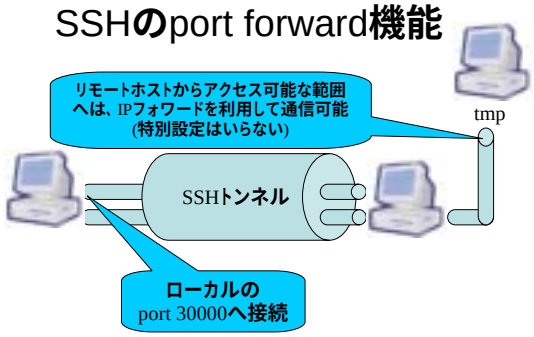
- SSH
  - お手軽、Unix系はほぼ全て利用可能
  - 通信経路暗号化
  - Windowsクライアントからは専用ソフトが必要
    - TTSSH(フリーソフト), PuTTY(フリーソフト), 他多数
  - Port forwarding機能を利用して、他の多くのプロトコルをバイパス可能
    - 外部からのsmtp/popなどで利用
    - ただし、設定が複雑
    - サーバ(sshd)側が許可している必要有り
      - RAINBOWは許可していないため利用不可能

SSHのport forward機能



- ローカルのport 30000へアクセスすることで、リモートのimapへ接続可能
- 通信はSSHトンネルを通るため、暗号化される

SSHのport forward機能



- ローカルport 3000へ接続するとtmpへつながる

VPN(Virtual Private Network)

- 特徴
  - 普通にLANに接続するのと同様の感覚でネットワーク内の資源を利用可能
  - 末端にVPN対応マシン(もしくはルータ)が必要
  - Windows2000/XPはL2TPを利用したVPNを単体で構築可能
  - 一般にはIPSecを利用(RAINBOWはこちら)
- 問題点
  - IPパケットそのものを暗号化, カプセリングするため、処理コストが高い
  - NATなどを併用した場合、設定が難しかったり特別なNAT機構が必要になる場合があるー>IPSecスルー機能

VPNの接続例

- クライアントをネットワークに接続する

- ネットワークをネットワークに接続する

VPNのプロトコル

- IPSec
  - IPパケットの暗号化と認証
- PPTP(Point to Point Tunneling Protocol)
  - PPPを拡張し、トンネリングを可能にしたもの
  - PPTP自身に暗号化機能は無い
  - 認証機能は無い
- L2F
  - PPP通信をベースとしたトンネリングプロトコル
  - 現在はL2TPに統合されている
- L2TP
  - L2FとPPTPを統合した物
  - PPTPの機能に加え、認証機能、1仮想トンネルを複数のセッションで共有することなどが可能
  - 暗号化機能は無い
  - WindowsはIPSecと組み合わせて暗号化機能も提供

Soft Ether

- 第2層トンネリングプロトコル
  - 既存ソフトを修正することなく利用可能
  - 特殊パケット(IPSecやL2TP(GRE))を利用しないため、ファイアウォールの設定や特殊なNATが必要ない
- Ethernet over TCP/IP
  - 上位層ではTCP over TCP
    - 『TCP over TCP is bad Idea』だが、独自の最適化でパフォーマンスを維持
- 多くのプロトコルに擬態することが可能
  - 途中のサーバには通常の通信に見える
  - 終点についた時点でトンネリングが外れ、本来の通信へ戻る

Soft Ether 通信方法

- ssh, socks, httpなどのproxyを越えて通信が可能
- 認証機構有り
- SSLによる通信経路の暗号化

- 既存のVPNと同様のセキュリティを確保可能
  - これは利用者にとってであり、必ずしもネットワーク管理者にとってではない
  - ソフトをインストールするだけで簡単に利用可能

諸刃の剣

- Soft Etherは第2層のトンネリングを行うため、通信内容によるフィルタリングが掛けられない
- 利用者によるネットワーク崩壊の可能性がある
- 例
  - ネットワーク同士を接続した場合、接続先のネットワークに進入された場合、接続もとのネットワークまで簡単に接続することが出来る
- 強力すぎ当初のバグフィックスにからんで，経済産業省(情報処理振興事業協会（IPA）)から公開停止要請が出たこともある(現在は再開)