第 8 章 アプリケーション層のプロトコル

8.1 サーバプロセス

8.1.1 クライアント・サーバ（C/S）モデル

サーバプロセス（サーバ）はネットワーク上で色々なサービスを提供するプロセスであり，クライアントプロセス（クライアント）はそのサービスを利用するプロセスである。クライアントがサーバのサービスを利用する場合は，サーバに対して**リクエスト**を送り，その**レスポンス**としてサービスを受ける（図 8.1.1）。

この通信形態は**クライアント・サーバ（C/S）モデル**と呼ばれ，現在のネットワークでは最も一般的な通信形態である。

一方最近では，お互いに対等の関係で通信を行い，サービスを提供し合う **Peer to Peer（P2P）モデル**の通信形態（図 8.1.2）を採るサービスも存在する。P2P 型のサービスは技術的には興味深いものがあるが，しかしながら，今日の実際のサービスでは不正なファイル交換型のものが多く，一般的な使用に関してはまだそれ程の普及はない。

なお，この章では**クライアント・サーバ（C/S）モデル**のアプリケーションについてのみ解説を行う。P2P モデルおよびそのアプリケーションについては，第 2 部の 5 章**「P2P ネットワークとグリット」**を参照すること。

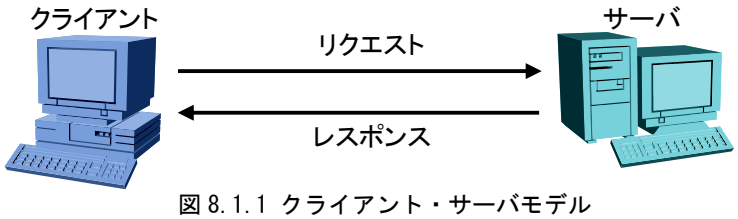

図 8.1.1 クライアント・サーバモデル

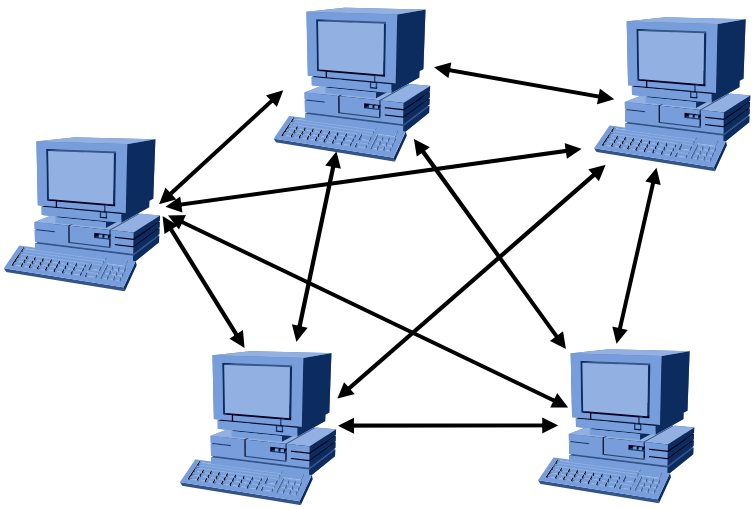

図 8.1.2 P2P モデル

8.1.2 デーモン（daemon）

Linux や Unix において，OS の中核（カーネル）とは独立して作動し，リクエストに対して様々なサービスを提供するプロセスを**デーモン（daemon）**と呼ぶ。デーモン（daemon）は精霊や小鬼を意味し，悪魔を意味する demon とは綴りが違う。デーモンは殆どの場合，サーバプロセスとして動作する。

なお，MS Windows では，デーモンと同等の働きをするプロセスを**サービス**と呼ぶ。

8.2 DNS（Domain Name System）　UDP の 53 番ポート（TCP が使用される場合もある）

DNS（Domain Name System）はインターネット上のリソースの名前とその実態の対応付けを行うサービスである。サービスの内容として最も代表的なものは，ドメイン名（正確には **FQDN**：次節参照）と IP アドレスとの対応付けである（簡単に言えば，ドメイン名を IP アドレスに変換する）。それ故，DNS サーバは**ネームサーバ**とも呼ばれる。

またネームサーバに問い合わせを行うクライアントは**リゾルバ**と呼ばれる。クライアントアプリケーションが FQDN の名前解決を行う（IP アドレスに変換する）場合は，リゾルバを利用してネームサーバに問い合わせを行う。

通常 DNS では UDP の 53 番ポートが使用されるが，返答が UDP セグメントの最大セグメント長（MSS）を越える場合には，TCP が使われることもある。ネームサーバの実装としては **bind** と呼ばれるソフトウェアが最も有名である。

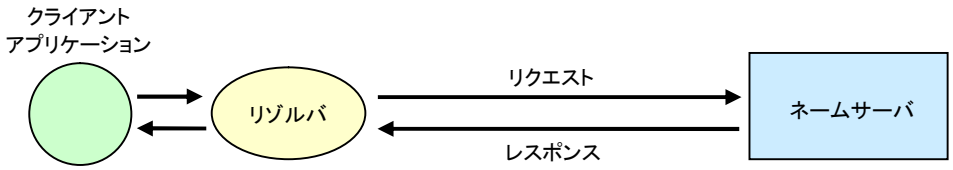

図 8.2.1 ネームサーバとリゾルバ

8.2.1 FQDN

一般的に使用されるドメイン名という言葉は非常に曖昧である。例えば，WEB の URL である http://www.tuis.ac.jp/ の **www.tuis.ac.jp** もドメイン名であるし，メールアドレスの anyone@tuis.ac.jp の **tuis.ac.jp** もドメイン名と呼ばれている。しかしながらこの両者は全く別のもので，**www.tuis.ac.jp** の方は，ネットワーク上にある実際のノード名を表している。

このように，実際のノード名を表すドメイン名を，他の種類のドメイン名と区別して **FQDN（Fully Qualified Domain Name：完全修飾ドメイン名）**と呼ぶ。

ネットワーク上で通信を行う場合，ソフトウェアは IP アドレスを使用する。しかし人間にとっては，一々ノード毎の IP アドレスを覚えるは大変であり，覚えやすい FQDN を使用

する方が簡単である。とは言え，結局通信を行うのはソフトウェアなので，どこかでFQDNをIPアドレスに変換（またはその逆変換）しなければならない。この変換を行うのが**DNSサーバ（ネームサーバ）**である。

FQDN，IPアドレスおよびMACアドレス間のアドレス変換の様子を図8.2.2に示す。

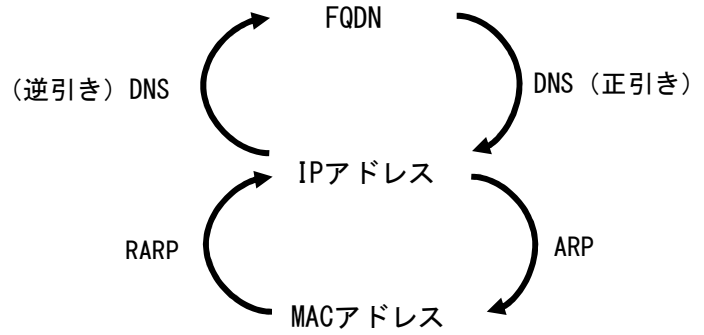

図8.2.2 ネットワークで使用されるアドレスの変換図式

8.2.2 FQDNの形式。

FQDNは一般的には図8.2.3の様な形式を採る。

jupiter.rsch.tuis.ac.jp.

図8.2.3 FQDNの形式

FQDNは階層構造になっており，一番右の．（ドット）は（殆どの場合は省略されるが）世界のトップ(Root Domain)を表している。右端の． に続く jp は**TLD(Top Level Domain)**と呼ばれている。

TLDは，com や org，net などのINANA（ICANN）が管理を行う**gTLD（generic Top Level Domain）**と jp，uk，kr，us などの2文字で表され，国毎に管理を行う**ccTLD（country code Top Level Domain)**，インターネット発祥の地である米国で使用される**特殊ドメイン**（gov，edu，mil,...）などに分類される。

図8.2.3で jp に続く **ac** は **SLD（Second Level Domain）**と呼ばれている。SLDは一般的には，組織の種類や名称，所在を表す。

ちなみにFQDNの記述では，大文字小文字は関係なく（ケース・インセンシティブ），大文字で書いても小文字で書いても，大文字小文字を混ぜて書いても全く問題は無い。

IPアドレスをドメイン名の様に記述することも可能である。 FQDNは右側に行くに従って，SLD，TLD，ルートドメインとアドレスの示す範囲が広がるが，逆にIPアドレスは右側に行くに従ってアドレスの示す範囲が縮小する。そこで，IPアドレスをFQDNの様に記述す

る場合には，IP アドレスを逆に書き，最後に **.in-addr.arpa.** を付加する。

例えば，IP アドレスが 202.26.159.139 のドメイン名表記は

139.159.26.202.in-addr.arpa.

となる（最後のドットは省略可)。この表記は DNS を使用して，IP アドレスから FQDN へ変換（**IP アドレスの逆引き**）を行う場合などに使用される。

8.2.3 DNS の階層構造

DNS はインターネットにおいて最も重要なサービスであり，インターネットに接続する場合には必ず使用するネームサーバの指定を行わなければならない（**DHCP** により自動的に行われる場合もある)。

DNS は FQDN（ドメイン名）と同様に階層的に管理されている。世界のトップであるルートドメインを管理する**ルートドメインサーバ（ルートサーバ）**は世界中に A～M の 13 台しか無く（ただし多数の予備が世界中に分散している)，欧米以外では唯一日本が M のルートドメインサーバを管理している。各ルートドメインサーバは一階層下の TLD の管理のみを行っており，問い合わせに対して各 TLD を管理するネームサーバの IP アドレスを返す。

各 TLD は自らが管理する SLD の情報のみを保持し，問い合わせに対して各 SLD を管理するネームサーバの IP アドレスを返す。このように，各階層の管理を行うネームサーバは自分の管理する，一階層下の情報のみを提供する（非再帰モード)。

FQDN を IP アドレスに変換（正引き）するために，DNS を検索する場合の動作例を図 8.2.4 に示す。この図では，WEB ブラウザが http://WWW.TUIS.AC.JP/ の WEB ページを参照するために，組織内のネームサーバが WWW.TUIS.AC.JP の名前解決（IP アドレスへの変換）を行っている様子を示している。詳しい動作内容は以下の通りである。

① 自組織内のネームサーバに WWW.TUIS.AC.JP. の IP アドレスを問い合わせる。
② ルートドメインサーバに問い合わせを行う。
③ JP. を管理するネームサーバの IP アドレスが返信される。
④ JP. を管理するネームサーバに問い合わせを行う。
⑤ AC.JP. を管理するネームサーバの IP アドレスが返信される。
⑥ AC.JP. を管理するネームサーバに問い合わせを行う。
⑦ TUIS.AC.JP. を管理するネームサーバの IP アドレスが返信される。
⑧ TUIS.AC.JP. を管理するネームサーバに問い合わせを行う。
⑨ WWW.TUIS.AC.JP. の IP アドレスが返信される.
⑩ WWW.TUIS.AC.JP. の IP アドレスが返信される。
⑪ WWW.TUIS.AC.JP. へリクエストを送る。
⑫ WWW.TUIS.AC.JP. からレスポンス(HTML)が返る。

なお，図では分り易いように JP を管理するサーバと AC.JP を管理するサーバは別に記載しているが，実際には同じサーバのようである。また，DNS システムでは，通信量を抑えるために，一度問い合わせた結果は，24 時間ほどキャッシュされるのが普通であ

るが，図ではキャッシュが全く行われていない状態を想定している。

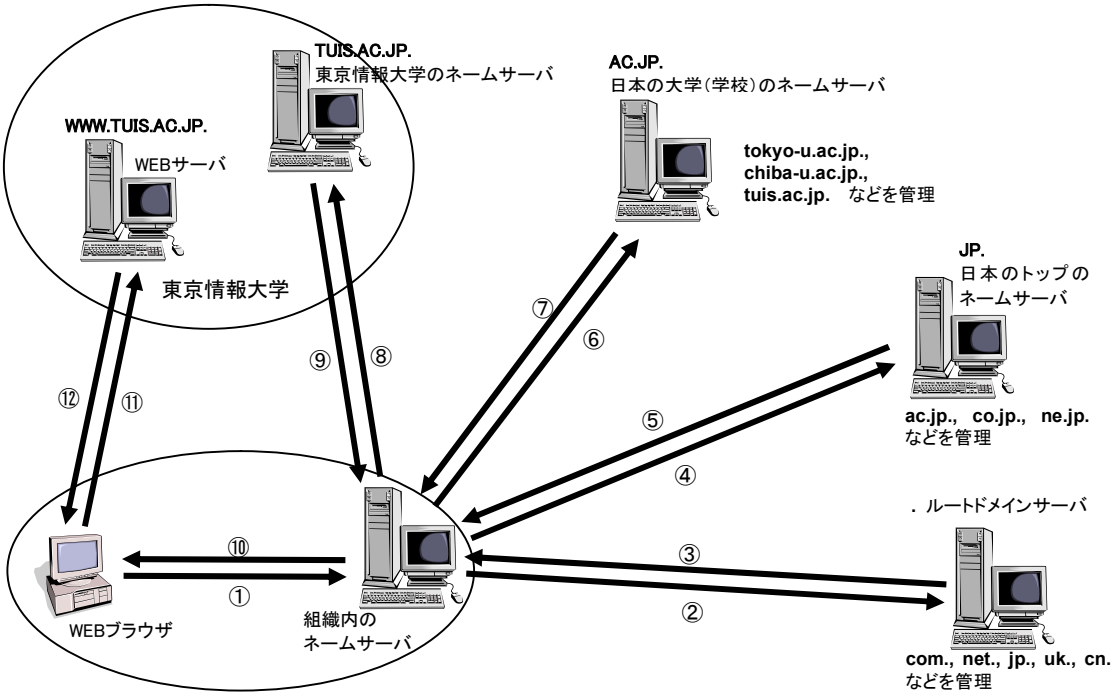

図 8.2.4 ネームサーバによる検索

8.2.4 再帰モードと非再帰モード 【中級】

リゾルバとネームサーバの動作モードには**再帰モード**と**非再帰モード**がある。通信を行っているリゾルバとネームサーバは同じモードで動作していなければならない。

再帰モードのネームサーバでは，問い合わせがあった場合に，自分が保持しない情報については上位のネームサーバに問い合わせを行う。一方，**非再帰モード**では，上位のネームサーバへは問い合わせを行なわず，自分の管理する情報についてのみ返答を行う。

図 8.2.4 では，WEB プラウザのリゾルバと組織内のネームサーバ（①，⑩の通信）は再帰モードで動作している。組織内のネームサーバは，名前を解決するために，非再帰モードのリゾルバとして各ドメインを管理する上位のネームサーバに問い合わせを行なう（②，③などの通信)。従ってこの場合，各ドメインの管理サーバは非再帰モードで動作することになる（図 5.2.5)。

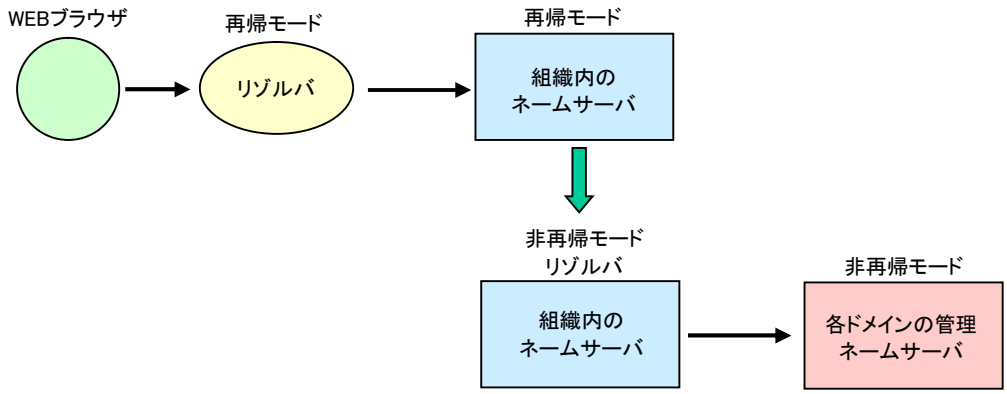

図 8.2.5 再帰モードと非再帰モード

8.2.5 DNS レコード 【中級】

DNS ではレコードと呼ばれる単位で情報が管理される。例えば，**A レコード**は FQDN から IP アドレスを検索（正引き）するための情報であり，**PTR レコード**は IP アドレスから FQDN を検索（逆引き）するための情報である。

その他にも，ドメイン名からその管理ネームサーバの名前を検索するための **NS レコード**，メールのドメイン名からそのメールを受信するメールサーバを検索するための **MX レコード**，ノードの別名から FQDN を検索するための **CNAME レコード**などがある。

8.2.6 nslookup コマンド 【中級】

nslookup コマンドは，ネームサーバへの問い合わせを行うコマンドである。対話モードを持つ非常に使い易いコマンドで，MS Windows でも使用可能である。また，nslookup はネームサーバから得た情報を，ユーザに分かり易い形式に変換して表示を行う。

その一方で，ネームサーバからの情報をどのように加工しているのか，ユーザからは不透明な部分もあり，信頼性に欠ける面もある。また，非再帰モードのサーバへの問い合わせでは，問題が生じることが知られている。それ故，nslookup 自身が（起動時に），将来的にはリリースされなくなる恐れがあることを警告し，代わりに **dig** や host コマンドを使用することを推奨している。

しかしながら，nslookup コマンドは初心者にとっては非常に使いやすいコマンドであるので，dig コマンドの紹介の前に簡単に解説を行っておく。

nslookup コマンドの対話モードは，MS Windows ではコマンドプロンプトから，Linux/Unix ではコンソールから **nslookup** と入力することにより起動する。nslookup の対話モードでは，**server** コマンドでネームサーバの指定や，**set type** コマンドで検索タイプとして DNS のレコード名を指定することが可能である（デフォルトは A レコード）。ただし，検索対象として IP アドレスを入力した場合は，自動的に逆引きモードになり PTR レコードの検索が行われる。

nslookup の対話モードを終了するには，Ctrl+C でブレイクさせるか，**exit** コマンドを入力する。

図 8.2.6 に Linux での nslookup の対話モードの実行例を示す（MS Windows でも殆ど同じである)。

```
$ nslookup
> server 202.26.157.10                  ネームサーバの指定
Default server: 202.26.157.10
Address: 202.26.157.10#53
> www.tuis.ac.jp                        www.tuis.ac.jp の IP アドレスを検索（A レコード）
Server:         202.26.157.10
Address:        202.26.157.10#53

www.tuis.ac.jp  canonical name = webhost.tuis.ac.jp.
Name:   webhost.tuis.ac.jp
Address: 202.26.157.15
> set type=NS                           NS レコードの検索モードへ移行。
> .                                     ルートドメインサーバを検索
Server:         202.26.157.10
Address:        202.26.157.10#53

Non-authoritative answer:
.       nameserver = J.ROOT-SERVERS.NET.
.       nameserver = K.ROOT-SERVERS.NET.
.       nameserver = L.ROOT-SERVERS.NET.
.       nameserver = M.ROOT-SERVERS.NET.
.       nameserver = A.ROOT-SERVERS.NET.
.       nameserver = B.ROOT-SERVERS.NET.
.       nameserver = C.ROOT-SERVERS.NET.
.       nameserver = D.ROOT-SERVERS.NET.
.       nameserver = E.ROOT-SERVERS.NET.
.       nameserver = F.ROOT-SERVERS.NET.
.       nameserver = G.ROOT-SERVERS.NET.
.       nameserver = H.ROOT-SERVERS.NET.
.       nameserver = I.ROOT-SERVERS.NET.

Authoritative answers can be found from:
B.ROOT-SERVERS.NET      internet address = 192.228.79.201
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
D.ROOT-SERVERS.NET      internet address = 128.8.10.90
E.ROOT-SERVERS.NET      internet address = 192.203.230.10
G.ROOT-SERVERS.NET      internet address = 192.112.36.4
H.ROOT-SERVERS.NET      internet address = 128.63.2.53
H.ROOT-SERVERS.NET      has AAAA address 2001:500:1::803f:235
I.ROOT-SERVERS.NET      internet address = 192.36.148.17
L.ROOT-SERVERS.NET      internet address = 199.7.83.42
M.ROOT-SERVERS.NET      internet address = 202.12.27.33
> 202.26.159.139                        IP アドレスを検索。自動的に PTR レコードを検索。
Server:         202.26.157.10
Address:        202.26.157.10#53

139.159.26.202.in-addr.arpa     canonical name = 139.128/28.159.26.202.in-addr.arpa.
139.128/28.159.26.202.in-addr.arpa      name = pleiades.solar-system.tuis.ac.jp.
```

```
> set type=MX                          MX レコードを検索するモードへ移行
> edu.tuis.ac.jp                       @edu.tuis.ac.jp のメールサーバを検索
Server:         202.26.157.10
Address:        202.26.157.10#53

edu.tuis.ac.jp  mail exchanger = 10 mercury.edu.tuis.ac.jp.
>exit                                  対話モードを終了
```

図 8.2.6 Linux での nslookup の対話モードの実行例

8.2.7 dig コマンド 【中級】

dig(domain information groper)は nslookup コマンドに代わるコマンドである。nslookup コマンドがネームサーバからの情報を加工して表示するのに比べ，dig コマンドはサーバへの問い合わせ（QUESTION SECTION）やサーバから返答（ANSWER SECTION）などの情報をそのままの形で表示する。

dig は，出力する情報は正確だが，ネームサーバに詳しいユーザでないと情報の意味を完全に理解することは難しいなどの欠点も持つ。また MS Windows では，標準の状態で dig はインストールされていない。

dig コマンドの基本的な書式は

dig [@ネームサーバの IP] リソース名 検索レコード [ ]内は省略可

である。@オプションが省略された場合は，/etc/resolv.conf の中に記述されたネームサーバへ問い合わせが行われる。

以下に Linux での dig コマンドの実行例を示す。

```
$ dig @202.26.157.10 www.nsl.ac.jp A
.....
;; QUESTION SECTION:
;www.nsl.tuis.ac.jp.             IN      A

;; ANSWER SECTION:
www.nsl.tuis.ac.jp.      39162   IN      CNAME   pleiades.solar-system.tuis.ac.jp.
pleiades.solar-system.tuis.ac.jp. 39162 IN A    202.26.159.139

;; AUTHORITY SECTION:
solar-system.tuis.ac.jp. 39162   IN      NS      jupiter.solar-system.tuis.ac.jp.

;; ADDITIONAL SECTION:
jupiter.solar-system.tuis.ac.jp. 39162 IN A     202.26.159.135
.....
```

図 8.2.7 202.26.157.10 のネームサーバを利用して www.nsl.ac.jp の A レコードを検索

```
$ dig jp NS
.....
;; QUESTION SECTION:
;jp.                             IN      NS

;; ANSWER SECTION:
```

```
jp.                     86400   IN      NS      a.dns.jp.
jp.                     86400   IN      NS      b.dns.jp.
jp.                     86400   IN      NS      d.dns.jp.
jp.                     86400   IN      NS      f.dns.jp.
jp.                     86400   IN      NS      c.dns.jp.
jp.                     86400   IN      NS      e.dns.jp.
jp.                     86400   IN      NS      g.dns.jp.
.....
```

図 8.2.8 日本の TOP のネームサーバの検索

```
$ dig 1.158.26.202.in-addr.arpa PTR
.....
;; QUESTION SECTION:
;1.158.26.202.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
1.158.26.202.in-addr.arpa. 86400 IN     CNAME   1.0/24.158.26.202.in-addr.arpa.
1.0/24.158.26.202.in-addr.arpa. 86400 IN PTR    www.infosys.tuis.ac.jp.

;; AUTHORITY SECTION:
0/24.158.26.202.in-addr.arpa. 86400 IN  NS      www.infosys.tuis.ac.jp.

;; ADDITIONAL SECTION:
www.infosys.tuis.ac.jp. 73513   IN      A       202.26.158.1
.....
```

図 8.2.9 IP アドレス 202.26.158.1 の逆引き

```
$ dig nsl.tuis.ac.jp MX
.....
;; QUESTION SECTION:
;nsl.tuis.ac.jp.                        IN      MX

;; ANSWER SECTION:
nsl.tuis.ac.jp.         86400   IN      MX      10 sol.nsl.tuis.ac.jp.

;; AUTHORITY SECTION:
nsl.tuis.ac.jp.         86400   IN      NS      jupiter.solar-system.tuis.ac.jp.

;; ADDITIONAL SECTION:
sol.nsl.tuis.ac.jp.     86400   IN      A       202.26.159.196
jupiter.solar-system.tuis.ac.jp. 86400 IN A     202.26.159.135
.....
```

図 8.2.10 @nsl.tuis.ac.jp のメールサーバを検索

8.3 SMTP (Simple Mail Transfer Protocol)　TCP の 25 番ポート

SMTP (Simple Mail Transfer Protocol) はインターネットでのメール転送用のプロトコルである。本来は，MTA (Message Transfer Agent) 注)間，即ちメールサーバ間のメール転送に用いられるべきプロトコルであるが，(その名前が示す通り) 非常にシンプルなプロトコルであるため，MUA (Message User Agent) 注)と MTA 間，即ちメーラソフトとメールサーバ間でもメールの転送が可能となってしまう。

以上の事と，SMTP ではメールの認証機能が無いことなどから，送信元を偽装した SPAM メール（迷惑メール）などをメーラソフトから直接メールサーバに投函することができ，様々な問題を引き起こしている。

注）MTA,MUA の M を Message ではなく Mail の略号として説明が行われる場合もあるが，これは RFC 毎に M の説明が違っているためである。従って，どちらを用いても間違いではないし，意味もそれ程違わない。つまりどちらの解釈においても，MTA はメールを転送するプログラム，つまりメールサーバを表し，MUA はユーザの操作によりメールの読み込み，送信（投函）を行うメーラソフトを表す（図 8.3.1）。

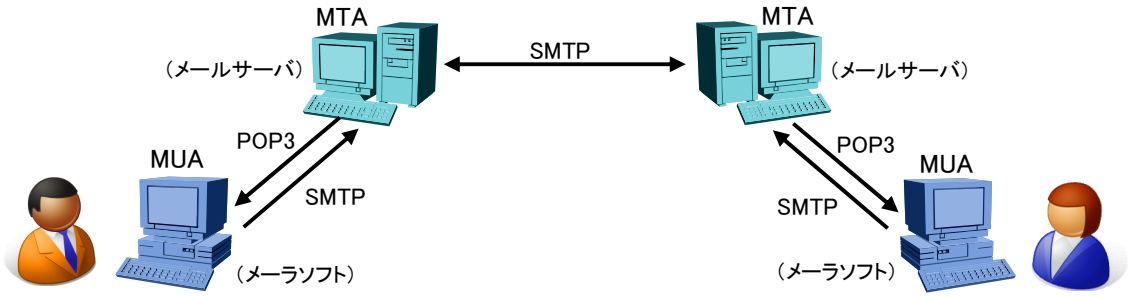

図 8.3.1 MTA と MUA の動作

SMTP は，インターネット初期の頃は各サーバ間をバケツリレー方式で転送されていたが，現在では TCP のコネクションにより直接受信サーバへ転送されるようになっている（ただし転送制御によるメールの中継は間々ある）。

また，SMTP では本来 7bit の文字コード（7-bit byte）しか転送できないが，SMTP の拡張仕様である **ESMTP（Extended SMTP）** では 8bit の文字コード（8-bit byte）も転送可能になっている。ただし，ESMTP であってもバイナリデータを直接転送することはできないので，画像などをメールに返付して送る際には，バイナリデータは **Base64** などの手法によりテキスト化されて転送される。

SMTP サーバの実装としては **sendmail**(セキュリティ的に問題が多い)，**postfix**(sendmail 互換），**qmail** などがある。

8.3.1 エンベロープ 【中級】

SMTP ではメール本体は**エンベロープ**と呼ばれる封筒に挿入されて転送される(図 8.3.2)。メール本体はヘッダとボディから成るが（ヘッダとボディは空行によって分けられる），SMTP から見るとヘッダもボディと同じアプリケーションデータに過ぎない。ヘッダにはメールの送受信および中継を行ったメールサーバにより情報が書き込まれる。

一方，エンベロープこそが SMTP の配送対象であり，エンベロープに書かれた宛先（**RCPT**

TO）にメールが届けられる。メール本体のヘッダの宛先（To や Cc など）は，メールサーバがデータとして書き込んだもので，実際の配送対象ではない。従って，エンベロープの宛先とヘッダの宛先には特に決まった関係はなく，両者が違っていても何の問題もない。つまり，ヘッダの宛先以外にメールが届けられても，何の不思議もないと言うことである。

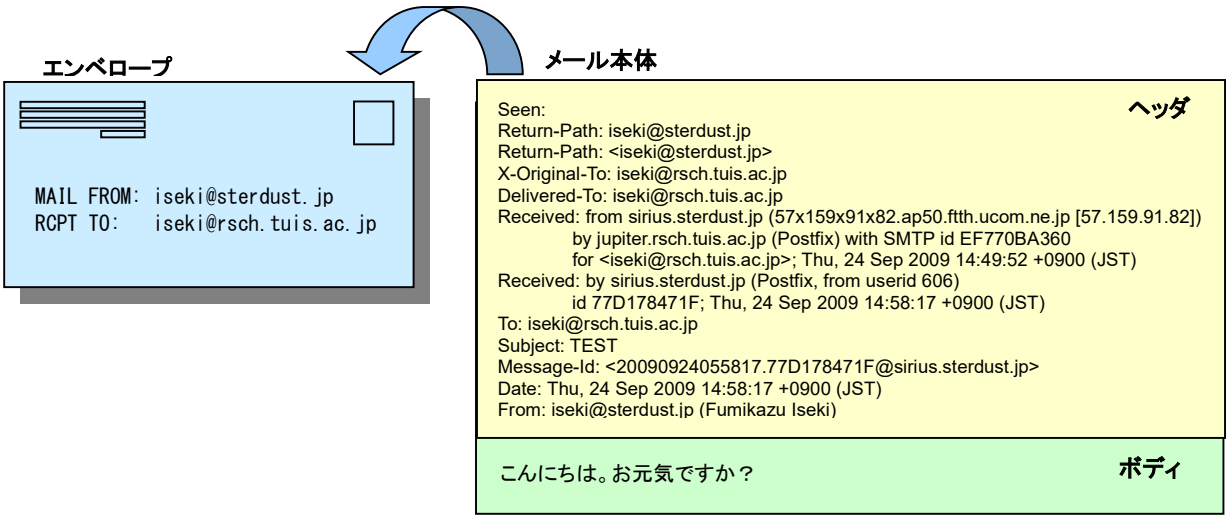

図 8.3.2 エンベロープとメール本体

メール本体のヘッダには，メールの送受信および中継を行ったメールサーバによる情報が書き込まれている（図 8.3.3）。これらの情報を見ると，メールがメールサーバによってどのように処理されたかを知ることができる（ただし，途中のメールサーバが正しい情報を書き込んだという保障は何処にもない）。

例えばヘッダの Received 行を見ると，メールがどのように転送されて来たかを推測することができる。図 8.3.3 のヘッダを例に採ると，Received 行が 3 行存在するが，後の方が時間的に先に処理された情報であり，from から by にメールが転送されたことを表している。つまりこの場合は

192.168.1.153[122.26.106.54] → iemsa204.i.softbank.jp → icmsa204.i.softbank.jp → jupiter.rsch.tuis.ac.jp

とメールが転送されたことが推測される。

最初の 192.168.1.153［122.26.106.54］は NAT（NAPT）により 192.168.1.153 のプライベート IP アドレスが 122.26.106.54 のグローバル IP アドレスに変換されたことを示している。また，最初の Received 行より icmsa204.i.softbank.jp の実際の名前は imsa204.mailsv.softbank.jp であり，IP アドレスが 126.240.66.103 であることも推測される（icmsa204.i.softbank.jp はメールサーバが自己申告した名前である）。

```
Seen:
Return-Path: iseki@i.softbank.jp
```

```
Return-Path: <iseki@i.softbank.jp>
X-Original-To: iseki@rsch.tuis.ac.jp
Delivered-To: iseki@rsch.tuis.ac.jp
Received: from icmsa204.i.softbank.jp (imsa204.mailsv.softbank.jp [126.240.66.103])
    by jupiter.rsch.tuis.ac.jp (Postfix) with ESMTP id E1C88BA360
    for <iseki@rsch.tuis.ac.jp>; Thu, 24 Sep 2009 15:00:45 +0900 (JST)
Received: from iemsa204.i.softbank.jp by icmsa204.i.softbank.jp with ESMTP
    id <20090924060045814.NJUQ.5086.icmsa204.i.softbank.jp@icmsa204.mailsv.softbank.jp>;
    Thu, 24 Sep 2009 15:00:45 +0900
Received: from [192.168.1.153] ([122.26.106.54] [122.26.106.54])
    by iemsa204.i.softbank.jp with ESMTP
    id <20090924060045777.EMCE.14522.iemsa204.i.softbank.jp@iemsa204.mailsv.softbank.jp>;
    Thu, 24 Sep 2009 15:00:45 +0900
Message-Id: <C94315B8-DCEF-4822-A68A-80063624EE97@i.softbank.jp>
From: "Fumi.Iseki" <iseki@i.softbank.jp>
To: =?iso-2022-jp?B?GyRCMGYOWBsoQiAbJEJKODBsGyhC?= <iseki@rsch.tuis.ac.jp>
Content-Type: text/plain;
    charset=us-ascii;
    format=flowed
Content-Transfer-Encoding: 7bit
X-Mailer: iPhone Mail (7A341)
Subject: =?iso-2022-jp?B?GyRCJUYlOSVIJWEhPCVrGyhC?=
Mime-Version: 1.0 (iPhone Mail 7A341)
Date: Thu, 24 Sep 2009 15:00:24 +0900
X-SB-Service: Virus-Checked
```

図 8.3.3 メールヘッダの例

8.3.2 MIME 【中級】

SMTP(ESMTP)は通常の方法ではバイナリデータを転送することはできない。バイナリデータを転送する場合にはバイナリデータをテキストデータに変換（エンコード）し，ヘッダに新たな項目を追加することによって，添付ファイルとして転送しなければならない。このように，SMTP でバイナリデータを転送するための一連の規格を **MIME(マイム：Multipurpose Internet Mail Extension)** と呼ぶ。

MIME において，バイナリデータをテキストデータに変換（エンコード）する最も一般的な手法に **Base64** がある。Base64 ではバイナリデータを 6bit 毎に区切り，2^6 = 64 個の文字（キャラクタ）で表現し直す。この場合 3Byte のデータが 4 個のキャラクタ(4Byte)で表現されることになる。例えば，0x00，0x10，0x83 のバイナリデータは，Base64 によるエンコードでは“ABCD”というテキストデータに変換される（図 8.3.4)。

なお，Base64 は符号化方式であって，暗号化方式ではないので，この点をよく注意すべきである。

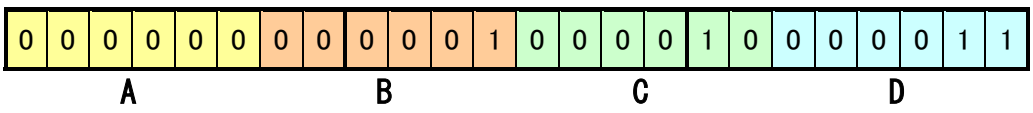

図 8.3.4 Base64 によるエンコード例

8.3.3 OP25B 【中級】

先に述べた様に，SMTP ではメーラソフト(MUA)からメールサーバ(MTA)へ，ユーザ認証なしに直接メールを投函することが可能である。この事が SPAM メール（迷惑メール）の増加の一因になっていることは確かである。

そこで，最近日本においては，**OP25B (Outbound Port 25 Blocking)** と呼ばれる手法により，メーラソフト(MUA)からメールサーバ(MTA)へのメールの直接投函を禁止する ISP（Internet Service Provider）が増え始めている。

OP25B では，ISP の境界にあるルータで，固定 IP アドレスを持たない外部ノードからの内部ノードの 25 番ポートへの接続パケットを全て遮断する。また，ISP 内部でも，固定 IP アドレスを持たないノードからのメールサーバの 25 番ポートへの接続を禁止してしまう。つまり固定 IP アドレスを持たないノードは，全て MUA であると認識する。

しかしながらこの方法では，ISP 内部の MUA はメールを送信できなくなってしまう。そこで，25 番ポートの変わりに **587 番の Submission Port（メール投函用ポート）** を用意する。ただし，587 番ポートで 25 番ポートと同じサービスを提供しては，ポートを変更した意味が無いので，587 番ポートでは **SMTP Auth** と呼ばれるパスワードによるユーザ認証機能を作動させ，そのパスワードを保護するために **SSL (Secure Socket Layer)** や **TLS (Transport Layer Security)** で通信の暗号化を行う（図 8.3.5)。SSL/TLS で SMTP を暗号化するプロトコルを **SMTPS (SMTP over SSL)** と呼ぶ。

国内 ISP の OP25B による対策により，国内のメールサーバを経由した SAPM メールは減少したようであるが，海外ではこれらの対策がまだ進んでいない国も多く，結局日本国内の対応だけでは，SPAM メールの対策としては限界がある。

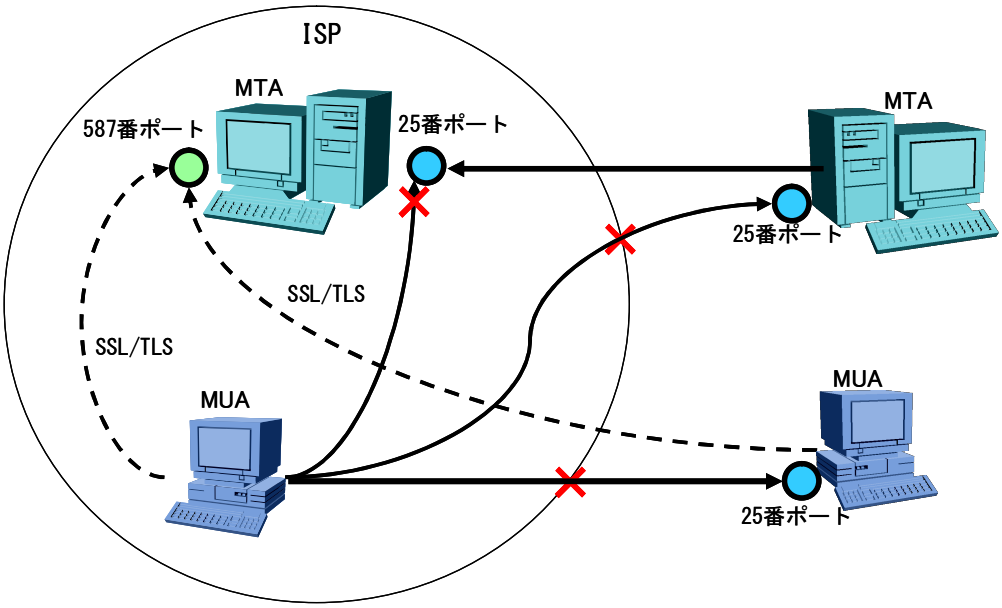

図 8.3.5 OP25B

8.4 POP3 (Post Office Protocol Version 3)　TCP の 110 番ポート

POP3（ポップスリー：Post Office Protocol Version 3）は MUA（メーラソフト）が MTA（メールサーバ）からメールを読み出すためのプロトコルであり，TCP の 110 番ポートが使用される。

POP3 サーバは，メールサーバのスプールに溜まったメールを POP3 のクライアント（メーラソフト：MUA）へ送信する機能を持つ。送信後，通常の設定では，送信済みメールはスプールから削除される（スプールに残す期間を設定することも可能）。

POP3 通信ではその内容は全く暗号化されないため，メールの本体は元より，POP3 サーバにログインするためのユーザ ID とパスワードも暗号化されずに平文のままネットワークを流れる。そのため，現在ではインターネット越しに POP3 を使用することは推奨されない。

APOP は POP3 の環境において暗号化通信が可能なプロトコルであるが，ユーザ認証に生パスワードを必要とするため，サーバの OS が Linux/Unix の場合は使用できないという欠点を持つ（Linux/Unix ではユーザの生パスワードを保持しないため）。

そのため，最近では SMTPS と同様に，POP3 を SSL/TLS で暗号化する **POP3S（POP3 over SSL：ポート番号 995）**が使用される場合が多い。

また，POP3(S)は SMTP のユーザ認証の一手法である **POP before SMTP** に利用される場合もある。これは SMTP によるメール送信を行う前に POP3(S)によりユーザ認証を行い，ユーザ認証が成功したノード（マシン）からの SMTP 接続のみを許可するものである。ただしこれはノード単位（IP アドレス単に）の認証となるため，確実性を欠く手法である。

前節で説明したように，最近の SMTP では POP before SMTP ではなく，**SMTP Auth** を用いてメール送信時にユーザ単位の認証が行われる場合が多い。

8.5 IMAP4（Internet Message Access Protocol Version4）　TCP の 143 番ポート

POP3 では，ローカルノード（ローカルマシン）に読み込まれたメールは，基本的にサーバ上のメールスプールから削除される.このことは，使用するマシンが度々変わる環境では問題となる場合がある。つまり，あるマシンで POP3 によりローカルマシンに読み込んでしまったメールは，他のマシンでは読むことができないのである。

メールサーバからメールを読み込んだ後も，一定期間サーバのスプールにメールを残すように設定することも可能であるが，根本的な解決方法ではない。

IMAP4（アイマップフォー：Internet Message Access Protocol Version4）では，メールをサーバ上のメールスプールからローカルマシンに転送せずに，サーバ上のユーザ専用領域に移動させる。メールを読んだ後も，メール本体はメールサーバ上に残るので，使用するローカルマシンが度々変わっても同じように過去のメールを読むことができるのである。

IMAP4 は POP3 と同様に通信内容は暗号化されないので，インターネット越しに使用する場合は，POP3S と同様に SSL/TLS で暗号化する **IMAP4S（IMAP4 over SSL）**を使用する方が好ましい。

8.6 FTP（File Transfer Protocol）　TCP の 20,21 番ポート

FTP はネットワークにおいて，ファイル転送を行うためのプロトコルである。データ転送に TCP の 20 番ポート，制御用コマンドの転送に 21 番ポートを使用する。

FTP のアクティブモードではデータ転送時に，FTP サーバの 20 番のポートから FTP クライアントへ接続が行われる。しかし，ファイアウォールなどにより外部から内部のノードへの接続を遮断している場合には，この接続は禁止される。この場合，FTP のコマンドは正常にサーバに届くが，ファイル転送だけができないという現象が発生する。

このような場合には，FTP をパッシブモードに切り替える。パッシブモードでは，ファイル転送時にも，FTP クライアントから FTP サーバの 20 番ポート（クライアントとサーバのネゴシエーションによって 20 番以外のポートが選択される場合もある）へ接続が行われるので，ファイアウォールを通過できる可能性が高くなる。

FTP サーバへの接続は POP3 などと同様に，ユーザ ID やパスワードが暗号化されない（勿論ファイルの内容も）。従って，現在では，POP3 と同様にインターネット越しに FTP を使用することは推奨されない。インターネット越しに FTP を使用する場合には，**SSH のポートフォワーディング**による **SFTP（SSH FTP）**や **SCP（Secure CoPy）**または SSL/TLS で FTP を暗号化する **FTPS（FTP over SSL）**など使用する方が望ましい（MS Windows では，SSH を利用

する **WinSCP** と呼ばれるソフトが有名である)。

一方，誰でもがログイン可能な **Anonymous（匿名）FTP サーバ**もある。Anonymous FTP は通常はダウンロード専用であり，Web ブラウザで URL が **ftp://～** となる場合は Anonymous FTP サーバへの接続を表す。Anonymous FTP では，ユーザ ID として **anonymous** または **ftp**，パスワードとして**メールアドレス**が使用されるので，インターネット越しの接続であっても，データの内容が機密性の必要の無いものであれば特に暗号化する必要はない。

8.7 TELNET　TCP の 23 番ポート

Telnet（テルネット）はリモートノードへ接続するための仮想端末プロトコルである。遠隔地にあるノード（マシン）をあたかも手元にあるかのように操作可能であるが，POP3 や FTP と同様に通信路が暗号化されないため，リモートノードへのログイン方法としては，現在では殆ど使用されない。代わって通信路を暗号化可能な **SSH（Secure Shell）**が，リモートノードへのログイン方法として専ら使用されている。

ただし，**telnet** コマンドを使用すると，任意の TCP ポートに接続できるため，TCP ポートの状態をチェックする場合などにはよく使用される。

例えば，IP アドレス 202.26.158.1 のマシンの 80 番ポートに接続するには，

```
telnet  202.26.158.1  80
```

とすれば良い。telnet コマンドは Linux/Unix は勿論，MS Windows XP のコマンドプロンプトからでも使用可能である（MS Windows Vista の場合は若干の設定が必要)。

8.8 SSH（Secure SHell）　TCP の 22 番ポート

SSH(Secure SHell)は TELNET(仮想端末)の暗号化バージョンであり，暗号化には **SSL/TLS** を利用している。

また，SSH は仮想端末機能の他に**ポートフォワード機能**を有している。ポートフォワード機能では，SSH ポートを他のアプリケーションのポートへ直結させる，この機能と暗号化機能を組み合わせると，暗号化機能を持たないプロセスでも SSH を経由して暗号化通信を行う事が可能となる（図 8.8.1)。

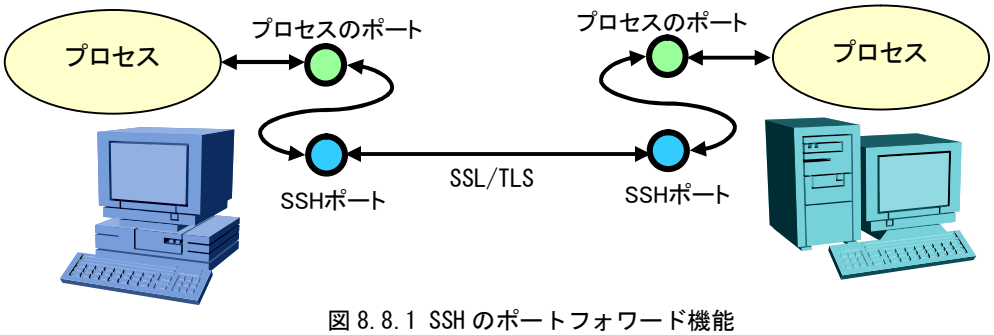

図 8.8.1 SSH のポートフォワード機能

8.9 HTTP (Hyper Text Transfer Protocol)　TCP の 80 番ポート

HTTP (Hyper Text Transfer Protocol) は HTML 文章を転送する **WWW (World Wide Web)** のプロトコルである。**Hyper Text** とは，文章中に他の文章へのリンクを持つような物を指し，HTML によって記述された WEB ページがその代表的なものである。

1991 年，**CERN（セルン：欧州合同素粒子原子核研究機構）**において，当初は研究者間の情報交換用に **Tim Berners-Lee** によって開発された。今日では，WWW はインターネットの代表的なサービスであり，インターネットの代名詞ともなっている。

HTTP サーバの実装としては Linux/Unix では **Apache（アパッチ)**，MS Windows では **IIS（Internet Information Server）**が特に有名である。

8.10 HTTPS (Secure Hyper Text Transfer Protocol)　TCP の 443 番ポート

HTTPS (Secure HTTP) は，Netscape 社によって開発された **SSL (Secure Socket Layer)** を使用して HTTP の暗号化とサーバの認証を行うプロトコルである。

SSL は**公開鍵暗号方式**を用いた暗号化および認証用のプロトコルであり，現在主に使用されているのは SSL Version 2 と Version 3 である。SSL Version 3 は後に若干の改良が加えられ，**TLS (Transport Layer Security)** として標準化された。従って，SSL Version3 と TLS はほぼ同じものと見なしても問題は無い。なお SSL Version 1 にはセキュリティホールが発見されており，現在では使用されることは無い。

HTTPS を使用する場合，サーバ側ではサーバの身元を証明する X.509 形式の**サーバ証明書**が必要となる。ただし，正式なサーバ証明書は有料で，URL の FQDN 毎に必要であるため，小規模なサイトでは，自らサーバ証明書を作成して使用しているところも多い。

最も，正式なサーバ証明書を使用しているからと言って，そのサイトが不正行為を行わないという保障は何処にもない。また，自前のサーバ証明書を使用しているからと言って，そのサイトが不正サイトであるとは限らない。

HTTPS の暗号化も通信路が暗号化されているだけであり，サーバ上でクライアントからのデータを処理する場合には，当然暗号化は解除された状態で処理されることになる。ユーザは HTTPS の機能を正しく認識し，その効果を過信しない事が大切である。

8.11 DHCP (Dynamic Host Configuration Protocol)　TCP/UDP の 546,547 番ポート

DHCP (Dynamic Host Configuration Protocol) サーバは IP アドレスが設定されていないノードに対して，IP アドレスとサブネットマスクの自動割り当てを行う。また，デフォルトゲートウェイ（ルータ）やネームサーバなどの通知を行う事もでき，オプションとして Proxy サーバの通知も可能である。

IP アドレスの自動設定を行う場合は，DHCP のクライアントプロセスを起動する必要があ

る。MS Windows では，ネットワーク設定のコントロールパネルのプロパティで該当インターフェイスに対して，「IP アドレスを自動的に取得する」のラジオボタンをクリックすれば良い。

DHCP による IP アドレスの自動設定は以下の手順で行われる。なお，これらの通信はデータリンク層の MAC アドレスを使用して行われるので，IP アドレスは必要とされない。

① DHCP クライアントは，**DHCPDISCOVER** のブロードキャストをネットワーク全体にリクエストする。なお，このリクエストには割り当て希望の IP アドレスを含ませることもできる。
② **DHCPDISCOVER** のブロードキャストを受信した DHCP サーバは，IP アドレスプール（割り振り可能な IP アドレスの集合）の中から，割り当てる IP アドレスの**候補**を選択し **DHCPOFFER** のレスポンスに含ませて DHCP クライアントに返す。
③ ネットワーク上に複数の DHCP サーバがある場合，DHCP クライアントは複数のオファーを受信するがその内一つを選択し，正式な IP アドレスの割り当て要請である **DHCPREQUEST** を，選択した DHCP サーバに送信する。
④ **DHCPREQUEST** を受信した DHCP サーバは，割り当てる IP アドレスを **DHCPACK** のレスポンスに含ませて DHCP クライアントに返す。
⑤ DHCP クライアントは，**DHCPACK** により割り当てられた IP アドレスが既に使用されていないことを確認するために，ARP のブロードキャストを送信する。
⑥ ARP のブロードキャストに対して返答が無ければ，そのアドレスは使用されていないことになるので，自己の IP アドレスとして設定する，
⑦ DHCP サーバにより割り当てられた IP アドレスには有効期限（リース期限）がある。DHCP クライアントは，有効期限（リース期限）の半分が過ぎた時点で，**DHCPREQUEST** によりリース期限の更新要求を何度でも HCP サーバに行うことができる。
⑧ DHCP クライアントは IP アドレスの使用が終わった場合には，DHCP サーバに対して **DHCPRELEASE** を送信する。**DHCPRELEASE** を受信した DHCP サーバは割り当ていた IP アドレスを IP アドレスプールに返却する。

以上をまとめると図 8.11.1 のようになる。

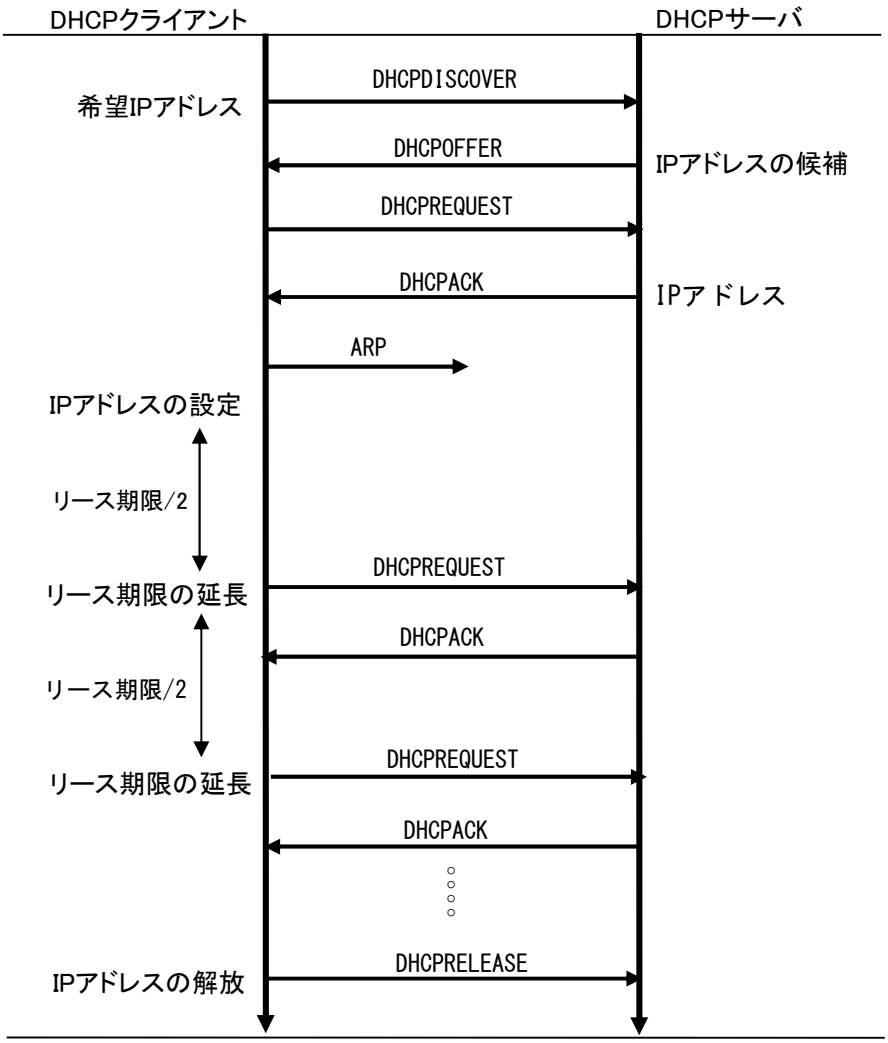

図 8.11.1 DHCP の通信動作

DHCP を利用するネットワークにおいて，不正な DHCP サーバ（不正な IP アドレスを返すサーバ）を作動させた場合，簡単にネットワークは混乱してしまう。例えば，家庭用 **BB（ブロードバンド）ルータ**の LAN ポートを，DHCP を利用するネットワークに（誤って）接続した場合，家庭用 BB ルータの LAN ポートでは通常の状態では DHCP サーバが作動しているため，この BB ルータからの DHCPOFFER を選択した DCHP クライアントには不正な IP アドレスが割り当てられることになる。結果としてそのノードはネットワークに接続することができなくなる。

このような場合，不正な DHCP サーバを見つけ出す作業は骨の折れる仕事となる。

8.12 SIP（Session Initiation Protocol）　UDP の 5060 番ポート

SIP（シップ：Session Initiation Protocol）は OSI 参照モデルのセッション層でセッションを管理する UDP のプロトコルである。**VoIP（Voice Over IP**:IP 電話）などでダイヤ

ル接続に使用される。

よく VoIP 専用のように思われがちだが，汎用性があり，他のプログラムからも利用可能である。ポート番号は，通常は 5060 番が使用されるが，特に決まったものではなく，他のポート番号を使用することも可能である。

8.13 RTP, RTCP (Real-time Transport Protocol, RTP Control Protocol)

RTP (Real-time Transport Protocol) は VoIP での音声伝送や映像のストリーミング配信において，リアルタイムにデータを転送するための UDP のプロトコルである。SIP と組み合わせた VoIP のシステムは特に有名である。

使用するポート番号は特に決まっておらず,VoIP では先行して実行される SIP によって，セッション毎にその都度決定される。

RTCP (RTP Control Protocol) は RTP を制御するための UDP のプロトコルで，RTP と組になって使用される。使用するポートは，RTP の使用ポートに 1 を足したものとなる。

8.14 RPC (Remote Procedure Call)　TCP/UDP の 111 番ポート

RPC (Remote Procedure Call) は Sun Microsystems 社が開発したセッション層のプロトコルであり，リモートノード上のプログラム（プロシージャ）を起動することのできる，強力なプロトコルである。非常に強力な機能を持つ反面，使い方を誤ると危険であるため，通常は組織内（イントラネット）でのみで使用される。よって，このプロトコルがインターネット上で使用されることは稀である。

RPC は，同じく Sun Microsystems 社が開発した **NIS** や **NFS** などで使用されている。

8.15 NIS (Network Information System)

NIS (ニス：Network Information System) は Sun Microsystems 社が開発した，ネットワーク上で情報を共有するためのシステムである。下位プロトコルとして RPC を使用している。

当初は Yellow Page（YP）という名称であったが，他社の版権に違反するため，途中で NIS という名称に変更された。それ故，NIS 関連の殆どのコマンドは **yp** という 2 文字で始まっている（ypserv，yppasswd など）。

サーバはマスタサーバとスレーブサーバの 2 段構成で，スレーブサーバでマスタサーバが持つ情報の完全なコピーを保持することにより，信頼性を高めている。

主にネットワーク上でユーザアカウント情報（ユーザ ID やパスワード: /etc/passwd, /etc/group, /etc/shadow）を共有するために用いられたが，幾つかのセキュリティホールが存在していた。その後 NIS はより安全性の高い **NIS+**に改良されたが，結局，現在では **LDAP** に取って代わられてしまっている。

使用するポートは RPC の portmapper（ポートマッパー） によって自動的に割り当てられ

るため，手動で登録する必要はない。

8.16 NFS (Network File System)

NFS (Network File System) は Sun Microsystems 社が開発したネットワーク上でファイルシステムを共有するためのシステムである。簡単に言えば，ファイルサーバ用のシステムで，ネットワーク上のリモートなファイルシステムをローカルなシステムにマウントすることができる。下位プロトコルとして RPC を使用している。

NFS は**ステイトレス**（サーバがクライアントの状態を保持しない）なプロトコルで，ファイルシステムを使用する時だけ，自動的にマウントすることも可能である（オートマウント機能)。

NFS は，Linux/Unix では標準的なファイル共有システムである。なお，MS Windows で使用されるファイル共有システムは **SMB (Server Message Block)** と呼ばれ，こちらは**ステイトフル**（サーバがクライアントの状態を保持する）なプロトコルである。

NFS が使用するポートは RPC の portmapper（ポートマッパー）によって自動的に割り当てられる

8.17 SAMBA　TCP の 137～138 番ポートなど

SAMBA(サンバ) は MS Windows のファイル･プリンタ共有プロトコル(**SMB: Server Message Block**）をエミュレートするシステムである.従って，SAMBA を利用すると，Linux/Unix マシンを MS Windows のファイルサーバやプリンタサーバにすることができる。

SAMBA は MS Windows のドメインコントローラ機能をエミュレートすることも可能で，Linux/Unix のマシンだけで MS Windows のサーバシステムを構築することが可能となる。

ちなみに **SAMBA** は **SMB** のもじりである。

8.18 LDAP (Lightweight Directory Access Protocol)　TCP/UDP の 389 番ポート

LDAP（エルダップ：Lightweight Directory Access Protocol）は，簡易ネットワークデータベースシステムであり，ネットワーク上でデータを共有するプロトコルである。

ネットワークデータベースシステムとしては，X.500 の DAP と呼ばれるディレクトリサービスが存在したが，このサービスがあまりにも複雑で重いプロトコルであったため，これを軽量化した LDAP（Lightweight DAP）が開発された。

MS Windows のアクティブディレクトリィとの連携も可能で，現在では主に NIS の代用として，ネットワーク上でのユーザアカウント情報（ユーザ ID やパスワード）の共有に用いられる。

8.19 NTP (Network Time Protocol)　TCP/UDP の 123 番ポート

NTP (Network Time Protocol) はネットワーク上での時刻合わせに使用されるプロトコ

ルである。ネットワーク上で色々なデータを交換する場合に，それぞれのノードの時刻が合っていないと，「未来に作成されたデータ」などという矛盾したデータが発生する可能性もあるため，各ノード間の時刻合わせは重要な問題である。

PC などではハードウェアによる時刻合わせは誤差が多く，信頼性に欠ける場合が少なくない。NTP は，NTP クライアントが NTP サーバに対して時刻の問い合わせを行い，通信による遅れも考慮して徐々に時刻を合わせていくプロトコルである。従って，時刻が大幅にずれている場合は NTP でも修正が困難になるため，ある程度時刻を合わせてから NTP を使用する方が通例である。

NTP クライアントは，同時に NTP サーバになることも可能である。従って，組織内の全てのノード（PC）を外部の NTP サーバに接続させるのではなく，外部の NTP サーバに接続させた NTP クライントのノード（PC）を組織内の NTP サーバとするべきである（図 8.19.1）。

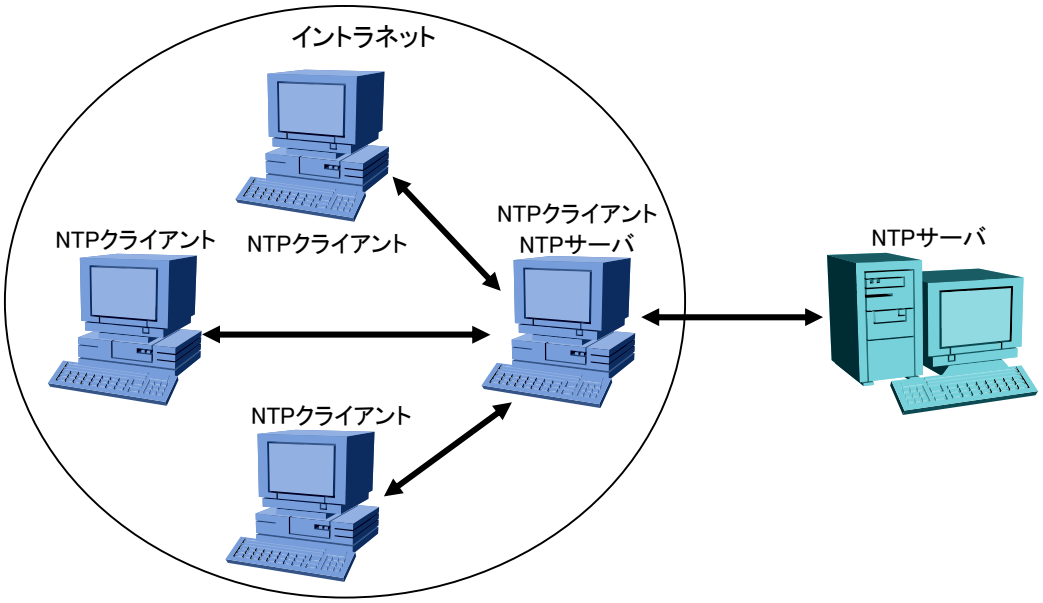

図 8.19.1 NTP システム

8.20 PROXY サーバ

Proxy（プロキシ）サーバは，イントラネットなどにおいて，ファイアウォールによりインターネットへ信号を送れないマシンに代わって，インターネット側との通信を行うサーバである。代理サーバとも呼ばれる。

WWW でよく使用され，WEB ページのキャッシュサーバとしても利用される（図 8.20.1）。WWW で使用する場合には，各ブラウザの設定画面において Proxy の設定を行うが，組織内部の WWW サーバへのアクセスは直接アクセス可能であるので，通信効率の観点から，Proxy の例外サイトとして登録する方が良い。

Proxy サーバは通常は外部へのアクセスのために用いられるが，外部からのアクセスに対

して，負荷分散やセキュリティ強化の目的として用いられる場合もある。この場合は特に**Reverse Proxy（リバースプロキシ）**と呼ばれる（図 8.20.2）。

実装としては **squid（スクイド）**や **delegate（デレゲート）**が有名である。

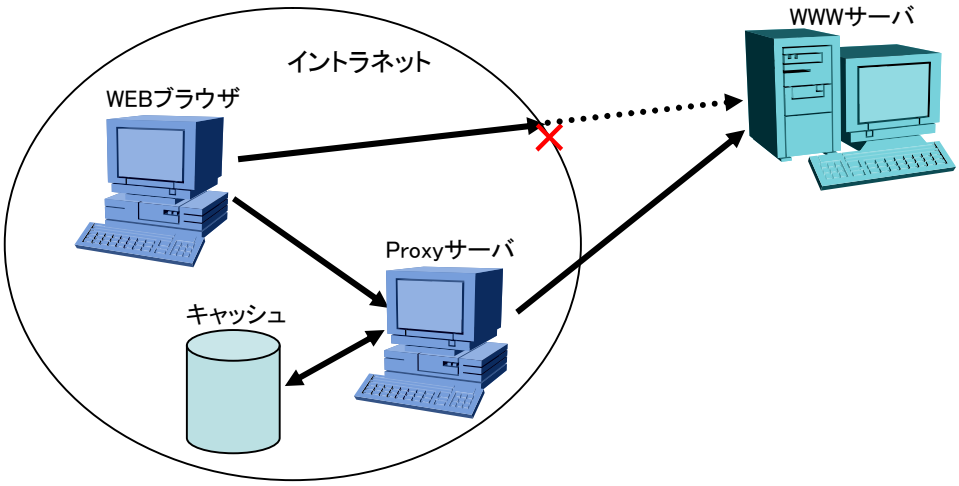

図 8.20.1 Proxy サーバ

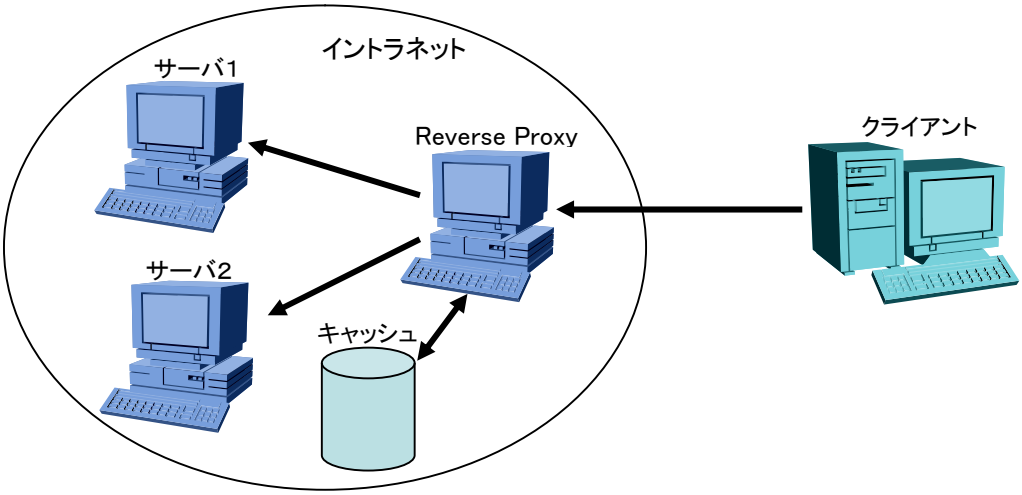

図 8.20.2 Reverse Proxy

8.21 スーパーデーモン

デーモンの中には，WWW サーバとは違い，それ程頻繁にサービスを要求されないものもある。このようなデーモンを常時作動させておくのはリソースの無駄使いになるので，それらのデーモンは停止させ，必要時にそれらのデーモンを呼び出すデーモンのみ作動させておけば良い。

このような監視用デーモンは**スーパーデーモン**と呼ばれ，会社の受付係のような仕事を

行う。スーパーデーモンは，自分の管理するデーモン（サービス）へのリクエストを監視し，リクエストがあった場合は，該当するデーモンを起動して通信の引継ぎを行う。該当デーモンにリクエストの処理を任せた後は，再びリクエストの監視に戻る。

スーパーデーモンの実装としては **inetd（アイネットディー）**，**xinetd（エックスアイネットディー）**が有名である。

inetd は設定が簡単な反面，細かい設定を行うことができない。また，inetd 自身はアクセス制御も無いため，通常はアクセス制御を行う **TCPWrapper（TCP ラッパー）**と組み合わせて使用する（図 8.21.1）。

一方，**xinetd** では，アクセス制御は勿論，サービスが有効な時間帯を指定できるなど細かい設定が可能であるが，その分設定が難しくなるのが欠点である。

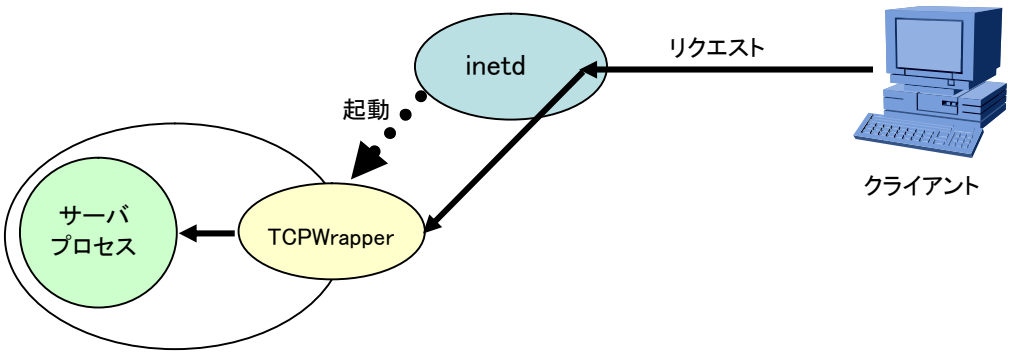

図 8.21.1 inetd

8.22 練習問題

【問題】

1. DNS のルートドメインサーバは世界中に何台あるか？ ただしバックアップ用サーバの数は除く。

2. 完全修飾ドメイン名（FQDN）を IP アドレスに変換するサーバは何か？
   ○ a. ネームサーバ　○ b. SMTP サーバ　○ c. IP サーバ
   ○ d. DHCP サーバ　○ e. ARP サーバ　○ f. SIP サーバ

3. クライアントマシンに自動的に IP アドレスを割り振るサーバは何か？
   ○ a. ネームサーバ　○ b. SMTP サーバ　○ c. IP サーバ
   ○ d. DHCP サーバ　○ e. ARP サーバ　○ f. SIP サーバ

4. POP の説明として正しいものはどれか？（複数選択）
   □ a. 現在の POP の最新バージョンは 2 である
   □ b. 盗聴の危険性がある
   □ c. 設定にもよるが，複数のマシンから同一のメールを読むのが不可能になる場合がある
   □ d. メールサーバ間でメールを転送するプロトコルである
   □ e. 暗号化されているので盗聴の心配はない

5. SSH のサーバが使用するポート番号は幾つか？
   ○ a. 20　○ b. 21　○ c. 22　○ d. 23　○ e. 25　○ f. 80

6. MS Windows でファイル共有を行うプロトコルは何か？
   ○ a. NFS　○ b. SMB　○ c. NTFS　○ d. SAMBA　○ e. Windows File Sahre

7. ネットワーク上で時刻を合わせるプロトコルは何か？
   ○ a. NFS　○ b. NetTime　○. NTFS　○ d. INET　○ e. NTP　○ e. NTT

8. ネットワーク上でユーザの ID とパスワードを共有することが可能なプロトコルはどれか？（複数選択）
   □ a. LDAP　□ b. NIS　□ c. NIS+　□ d. DHCP　□ e. SIP

9. 【中級】nslookup または dig コマンドを使用して，よく使用する WEB サイトのサーバの IP アドレスを調べてみよ。

10. 【中級】nslookup または dig コマンドを使用して，自分の携帯メールを受信するメールサーバの IP アドレスを調べてみよ。

11. 【中級】サポート WEB ページ（http://el.nsl.tuis.ac.jp/）にある NetProtocol プログラムを使用して，HTTP，HTTPS，POP3，SMTP などのプロコルを調べてみよ。

【解答】

1. 13 台
2. a
3. d

4. b, c
5. c
6. b
7. e
8. a, b, c
9. -
10. -
11. -

コラム

【シミュレーションとエミュレーションの違いは何か】

シミュレーションとエミュレーションの違いは何であろうか？ 結論から言えば，エミュレーションとは完全に物事を模倣することであり，ユーザから見ると本物かどうかの区別がつかない状態を言う。一方シミュレーションとは，ある条件の下で物事を模倣することである。条件としては「難しい要素は考えに入れない」といのが最も一般的である。

従って，飛行機のシミュレータは存在するが，エミュレータは存在しない。もし飛行機のエミュレータが存在するとすれば，エミュレータを操縦している人は，それが本物の飛行機なのか，それともエミュレータなのかを判別できないと言うことになる。加速度や重力を考えれば，それは現在の技術では到底不可能なことである。

少し古い辞書で，「シミュレーションはソフトウェアで行い，エミュレーションはハードウェアで行う」というのを見たことがあるが，このような解説は間違いである。確かにエミュレーションは速度が要求されるので，ハードウェアで行ったほうが有利である。しかしそれは速度の問題だけであり，ソフトウェアで実現不可能という訳ではない。

辞書と言えば，ISO を International Standard Organization の略であるとしているものを見たこともある。辞書だからと言って，全て正しいとは限らないということである。