Chapter 1

インターネットの仕組み

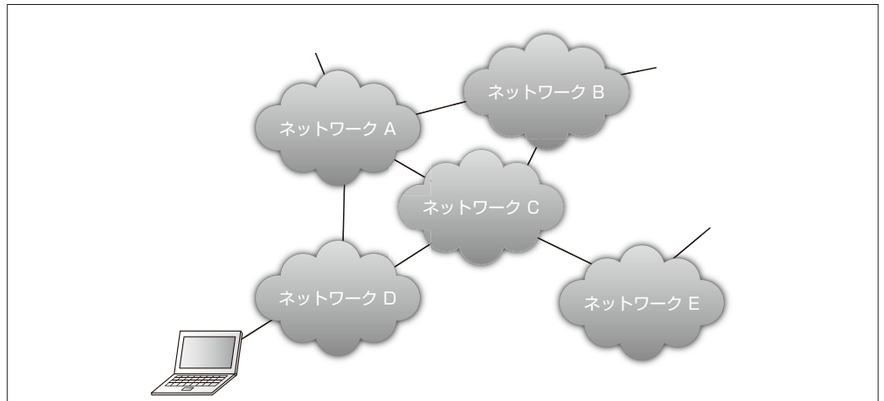
Linuxはさまざまな通信手法をサポートしており、インターネット以外のネットワークプログラミングも可能ですが、「ネットワークプログラミング」という表現は主に「インターネットプログラミング」を指します。

インターネットを使ったネットワークプログラミングでは、インターネットそのものの構造や仕組みを知らなくてもアプリケーションは作成できます。しかし、インターネットの仕組みについて知ることは、スキルを身に付け、本質的な設計ミスを防ぐためには非常に大事です。Chapter 1では、インターネットそのものの仕組みについて解説します。

1-1 インターネットとは

「インターネット(Internet)」とは、一言でいえばネットワークの集合体です。「Inter」という単語は「〜の間」という意味であり、「net」はそのままネットワークを示します。すなわち、ネットワークとネットワークを繋げ合わせて巨大なネットワークを構成しているものがインターネットなのです(図1-1)。

図1-1　インターネットの構造



ネットワークとインターネット

では、「ネットワーク」とは何でしょうか？

複数の機器が接続されていれば、それはすでにネットワークです。家の中にある複数のパソコンを繋いでいるのも、家から「ISP(Internet Service Provider)」まで繋がっているのも、ISP同士が接続しているのも、すべてネットワークといえます。

先ほど、インターネットはネットワークの集合体であると説明しました。ではなぜ、インターネットは単一の大きなネットワークではなく、小さくバラバラなネットワークがくっついている集合体なのでしょうか？　それは、ネットワークには運用・管理する人が必要だからです。

たとえば、会社の社内LANであればネットワーク担当部署(情報システム部など)や(専任ではないかもしれませんが)担当者がいます。あるISPのネットワークを運用しているのはISP事業者の人です。家でパソコンを使っているお父さんも、家庭内ネットワークの管理人といえるでしょう。もしも世界中に広がるインターネットが単一のネットワークだった場合、これら全部のネットワークをまとめて面倒を見る人(あるいは組織)が必要になります。会社や団体が数個のレベルならばまだ何とかなるかもしれませんが、世界規模で考えるとこれは現実的ではありません。

インターネットは、個別に運用・管理された各ネットワークが、必要に応じて相互接続することにより成り立っています。この「それぞれ自分のネットワークを管理する」という自律分散協調的な考え方が、インターネットの根幹にあります。

このような、「自分は自分の責任範囲だけ考える(それ以外は考えない)」というのは重要な要素です。これにより、「規模性(スケーラビリティ)」が確保されています。

初期のインターネット設計思想

ここで、インターネットがどのように作られ、発展してきたか紹介しましょう。

インターネットはもともと、米国における軍の研究で開発されたものです。あまたある基地、軍事施設のどこか1カ所が攻撃されて無力化しても、全体としては生き残れる情報ネットワークを目指して設計されました。それぞれのちほど解説しますが、途中でどこかの回線が切れても何とかして通信を行えること、さまざまな状況で迅速にかつ安価にネットワークを構築できるように多様な種類の機器が相互に接続できること、などの思想もありました。現在インターネットがこれだけ普及したのは、この耐障害性が大きな要因でしょう。

インターネットの根幹となっているTCP/IPを作った研究者たちの一人が書いた1988年の論文[注1-1]によると、インターネットに要求されたゴールには以下のような要件があったそうです。

- **ネットワークやゲートウェイがなくなっても通信は継続できなければならない**
- **さまざまな種類のコミュニケーションサービスをサポートできなければならない**
- **さまざまなネットワークと接続できなければならない**
- **分散管理ができなければならない**
- **適切なコストで構築できなければならない**
- **新しいホストの追加を容易にできなければならない**
- **アーキテクチャに含まれるリソースに対して課金が可能でなければならない**

現在のインターネットを知っていると、これらの条件を見てもあまり感動はないかもしれません。しかし、1970年代前半における設計(アーキテクチャ)としてはすごいことでしょう。

注1-1：D.D.Clark. The Design Philosophy of the DARPA Internet Protocols, Proceedings of ACM SIGCOMM, Pages 106-114, September 1988.

パケットという考え方

初期の設計思想のうち、「何とかしてネットワークを生存させる」という課題を研究する過程で生まれたのが、「パケット」という考え方です。

それまではネットワークといえば、電話のように回線を物理的に繋いでいくという方式が主流でした。この「回線交換方式」では、通信経路のどこかが途中で切れてしまうと特定の通信が完全に切れてしまいます。完全には切れなかったとしても、どこで切れたかを把握して通信を修復するか、通信を行っている両者が考えて対処しなければなりません。しかし、パケットという考え方によって、通信の両端で途中のネットワークが行っていることを把握しなくてもよくなりました。

具体的に説明しましょう。インターネットでのデータのやり取りは、パケットと呼ばれる「小包」で行われます。ひとつのデータがひとつの小包（パケット）とはかぎらず、複数個のパケットに分けられることもあります。このようにデータをパケットに分けているのは、さまざまな情報を小分けにすることで「多重化」ができるようにするためです（図1-2）。

図1-2　パケットを使った通信／使わない通信

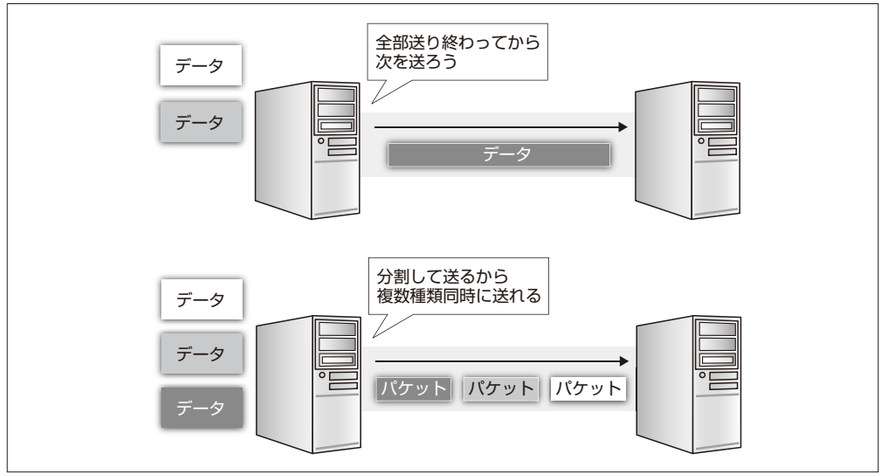


図1-2の上図では、データが全部送信されたあとに次のデータを送っています。一方、図1-2の下図では、データをパケットという単位に小分けにして送っているため、複数種類の通信を（見かけ上）同時にできます。このように、複数種類の通信を同時に行うことを「多重化」といいます。

これらのパケットを転送しているのが「ルータ（router）」と呼ばれる機器です。各ルータは「宛先集」を持っていて、パケットの宛先を見て自律分散的に（自分のわかる範囲で）パケットを転送します（図1-3）。パケットの宛先ごとに「この宛先ならこっちだな」と考えて転送するだけです。「こっちだな」と思った方角にパケットを転送する行為を「フォワーディング（forwarding）」といいます。各ルータがバケツリレー的にパケットを転送していき、最終的には宛先に到達します（図1-4）。

図1-3　パケットのフォワーディング

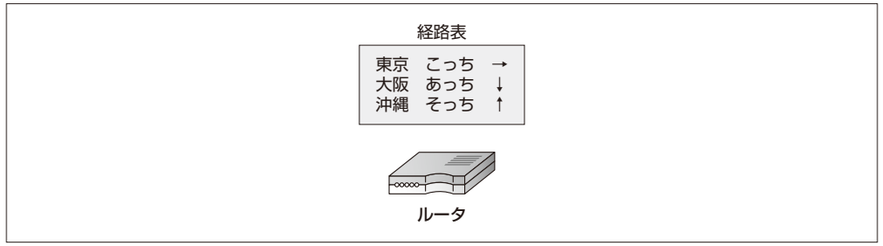
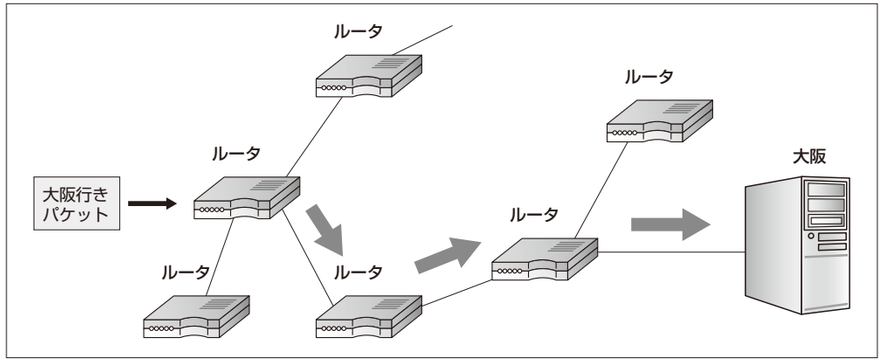


図1-4　パケットが宛先に届くまで



さて、「この宛先ならこっちだな」という説明で「こっち」の方角はどうやってわかるのだろうと不思議に思った人は鋭いです。それは「ルーティング（経路制御）」と呼ばれる機能が実現しています。

ただし、ルーティングに関して解説し始めると非常に長くなるので、本書では割愛します。別途、インターネット技術解説書を参照してください。

1-2 OSI 7層モデル

プロトコルはよく言語に例えられます。たとえば、日本語と英語では話が通じませんが、両方とも日本語を話したり、両方とも英語を話せば話が通じます。言い換えると、共通言語があれば話は通じます。プロトコルとは、機器同士が通じ合えるようにする共通言語ともいえます。

ただし、ネットワークの世界では、共通の言語があるだけでは不十分です。誰がどのような順番でどのように通信を行うかといった、取り決め（約束事）も必要です。

インターネットの基本プロトコルは「IETF（Internet Engineering Task Force）」で決定されています。決定されたプロトコルは「RFC（Request For Comments）」という標準文書にまとめられています。

インターネットではプロトコルは階層化されています。インターネットで通信を行うとき、ユーザは複数のプロトコルを同時に使用していることになります。たとえば、物理的な通信を行うもの、隣の機器までの通信を行うもの、経路を考えながらパケットを転送するもの、データが届いたことを保証するもの、さらにその上、といった感じになります。

このように階層化されているのには理由があります。各層を独立させることで、機器やソフトウェアの相互接続性を上昇させているのです。たとえば、無線LANであっても、光ファイバであっても、ADSLであっても、Webの閲覧は可能ですが、これは物理的な接続形態とインターネットを使ってやり取りされるアプリケーション用データ転送方式が切り分けて構築されているためです。このような接続形態のバリエーションを生み出しているのが「OSI参照モデル」の考え方です。

このOSI参照モデルは、コンピュータの持つべき通信機能を階層構造に分割しています。OSI参照モデルは、インターネットの配線からアプリケーションまでを7階層に分け、各層はそれぞれ自分の責任範囲だけをがんばるという仕組みになっています。

次は、OSI参照モデルを下から順に紹介します。

物理層とリンク層

OSI 7層モデルの一番下は「物理層」です。一番下の層のことを「レイヤ1（Layer 1）」ともいいます。この物理層は、物理的な接続形態を表します。また下から2番目の層は「リンク層」です。「レイヤ2（Layer 2）」ともいいます。リンク層は、隣の機器との「接続性」を確保します。ここでの接続性とは、物理層での物理的接続ではなく、論理的な接続です。たとえば、光ファイバーで2台の機器を接続しても同じ方法で通信をしないと通信は成り立ちません。このような隣の機器との通信方法を決定しているのがリンク層です。

リンク層では、物理層の違いを吸収するという作業も行っています。たとえば、光ファイバー

もしくはLANケーブルで通信できる「イーサネット（Ethernet）」というものがあります。

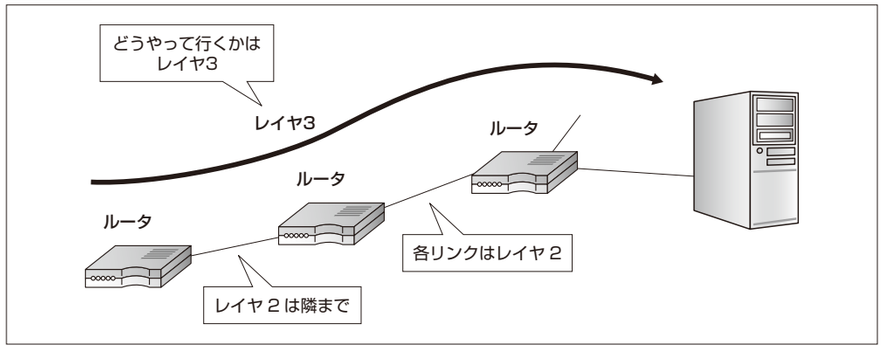
先ほど登場した光ファイバー、電話回線、LANケーブル（Ethernetケーブル）、無線など、下の物理層が異なっていても、リンク層がその違いを隠蔽すれば、それより上の階層はどんな接続形態なのかを考えずにすみます。このような仕組みにより、接続方法にかかわらず、同じようにインターネットが使えるのです。

ネットワーク層

下から３番目は「ネットワーク層」です。「レイヤ3（Layer 3）」ともいいます。ここでは、隣よりもさらに離れた機器との通信を実現します。この表現は多少わかりにくいかもしれません。筆者も、一番最初に勉強をした時点ではレイヤ2とレイヤ3の違いが感覚的にわかりませんでした。

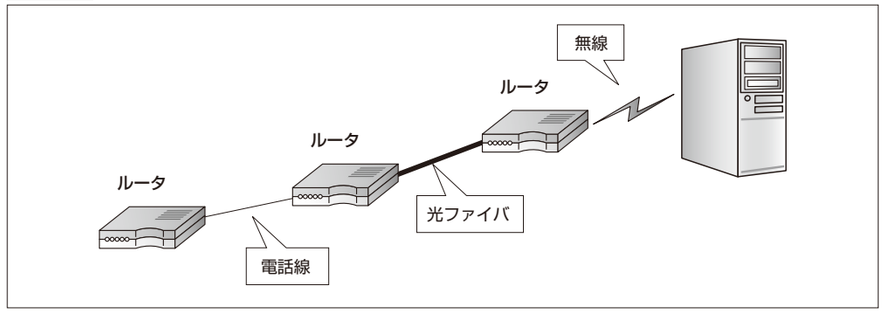
さっそく、レイヤ2とレイヤ3の違いを図1-5に示します。

図1-5　レイヤ２とレイヤ３の違い



この図のように、レイヤ3では離れた地点との通信を実現します。ものすごく乱暴に言ってしまえば、ルータを超えないのがレイヤ2で、ルータを超えるのがレイヤ3です。ここでもOSI 7層モデルの階層構造が重要な役割を果たしています。もしも階層構造がなかったならば、非常に困る状況になるでしょう（図1-6）。

図1-6　さまざまな物理接続で繋がれたネットワーク



階層構造のおかげで、各ルータが接続している物理形態やリンクに左右されずにレイヤ3が動作できるのです。このネットワークを越えた通信を行うのが「IP（インターネットプロトコル）」と呼ばれるものです。IPについては、のちほど詳しく解説します。

トランスポート層

次の「レイヤ4（Layer 4）」は、「トランスポート層」と呼ばれます。レイヤ3まではパケットが届くまでの作業ですが、レイヤ4以上[注1-2]では宛先にパケットが届いてからの作業を行います。

レイヤ4の代表的なプロトコルには「TCP（Transmission Control Protocol）」や「UDP（User Datagram Protocol）」があります。

インターネット通信においてよくTCP/IPという表現が利用されますが、これはレイヤ3にIP、レイヤ4にTCPを使った通信のことを表しています。現在のインターネットでは通信のほとんどがTCPによるものなので[注1-3]、TCP/IPがインターネット通信と同義として使われている場合を見受けますが、厳密にいえば別物なので注意しましょう。

セッション層とプレゼンテーション層、アプリケーション層

最後は一気に紹介しましょう。

注1-2：このレイヤ4自身も含みます。

注1-3：時期や国によって細かい値は異なりますが、たとえば2005～2008年の日本国内における通信ではTCPデータによる通信が95～97%ぐらいを占めており、続いてUDPが1～3%ぐらいを占めています。"Observing Slow Crustal Movement in Residential User Traffic", Kenjiro Cho（IIJ）, Kensuke Fukuda（NII/PRESTO JST）, Hiroshi Esaki（東京大学）, Akira Kato（慶應義塾大学）, ACM CoNEXT2008, Dec 10-12, 2008, Madrid, SPAIN（http://www.iijlab.net/~kjc/papers/kjc-conext2008.pdf）

「レイヤ5（Layer 5）」は「セッション層」と呼ばれ、トランスポート層によって提供されたデータをまとめて制御するための仕組みを提供しています。「レイヤ6（Layer 6）」は「プレゼンテーション層」と呼ばれ、アプリケーション同士がデータの表現に関して合意できるようなサービスを提供することを役目としています。そして「レイヤ7（Layer 7）」は「アプリケーション層」と呼ばれ、各アプリケーションがやり取りするデータ通信を表現しています。

これらレイヤ5～7は役割が非常に似ているため、分けて考えようとすると混乱することがあります。そのため、筆者が簡単に説明するときにはレイヤ5～7はまとめて考え、あまり違いを意識させないようにしています。

このように、レイヤごとに役割が分かれている階層構造になっているため、さまざまな機器の相互接続性が確保しやすくなっています。

1-3 TCPとUDPの概要

TCPとUDPはレイヤ4にあり、レイヤ3であるIPの上の層に存在するプロトコルです。IP層は目的地（宛先）までパケットを運ぶ役目を請け負っています。しかし、実際の通信においては、それだけでは不十分です。

たとえば、ネットワーク上にて転送されているパケットが、何らかの障害でいきなり消えてしまうことがあります。電線の周りに変な電磁波が発生してパケットのデータが修復不可能なほど壊れてしまう場合もあれば、経路に障害が発生してパケットが迷子になることもあります。パケット転送中に機器の電源が落ちることだってあるでしょう。そのようなとき、ルータの集合であるインターネット網は何もしてくれません。インターネットを構成するルータには「パケットを目的地まで送り届ける」という役目は課されていないのです。

インターネットは、耐障害性を重視するために「各機能はできるだけ無責任であること」を目指しているので、途中でどのパケットが失われたかなどを「ルータが考える」ことはありません。では、どうするかというと、インターネットを構成するルータではなく、パケットを送受信するエンドノード（送信側と受信側）が考えることになっています。

インターネットそのものを構成するルータはできるだけ簡素で単純なものとして実現しつつ、足りない部分をエンドノードが各自の責任で補完するという役割を担っているのがTCPやUDPなどのプロトコルです。

TCPとUDPには、それぞれ特徴があります。詳細についてはこのあと解説しますが、簡単に紹介すると、TCPはデータを正しく全部届けたい場合に有効で、UDPは途中で多少のデータがなくなってでも早く届けたい場合に有効です。またTCPは1対1の通信しかできませんが、UDPは一度に大量の受信者に向けてデータを送信できます。

インターネット上を流れるトラフィックの9割以上はTCPによるものであるといわれています。たとえば、Web、メール、FTPなどは全部TCPです。「インターネット」という単語から

思いつく通信の多くがTCPを使っているといっても過言ではないかもしれません。

一方、音声通話、映像配信などではUDPが多く利用されています。音声通話では音を全部正確に伝えるために時間がかかるよりも、多少途切れてでもすぐに相手に伝わることの方が重要な場合が多いでしょう。また、UDPでは一度に複数の相手にデータを送れるため、放送型の映像配信や音声配信によく利用されています(注1-4)。

コネクション型プロトコル～ TCP

TCPは「コネクション型プロトコル」です。コネクション型プロトコルの特徴は、通信をする前に「コネクション」を確立するところにあります。コネクションのイメージとしては、通信を行いたい相手がお互いにプラグを持ってきて「カチャ」と繋げる感じでしょうか。このようなコネクション型の通信では、通信を行う端末同士は仮想的な通信回線を作り上げます。その仮想的な通信回線のことを、「バーチャルサーキット（Virtual Circuit）」と呼びます。

バーチャルサーキットでは、「通信の信頼性」が保証されます。通信の信頼性とは、「送ったデータが必ず届くこと」、および「送ったデータが順番どおりに届くこと」です。これは、通信の両端でお互いに何を送り何が届いたかを繰り返し問い合わせることによって実現しています。

インターネットでは、データが通信途中のどこかでいきなり消失することがあります。このような場合でも、両端で何が届いたかを把握することで足りないパケットに気付き、再度送り直すことが可能です。このように再度送り直すことを「再送」と呼びます。すなわち、再送の仕組みによって、バーチャルサーキットの信頼性を実現しているのです。

バーチャルサーキットによる信頼性の実現は、アプリケーションプログラマにとっても非常にありがたいものです。プログラムを書くにあたり、「ネットワークでパケットがなくなったらどうするか」や「ネットワークの途中でデータの内容が変わったらどうするか」という処理はなるべく考えたくありません。これがTCPというプロトコルを使うことにより、そのようなわずらわしさから解放されます。実際に、メール、WWW、FTPなどといったアプリケーションでは、データの内容が正確に伝わることが要求されます。それらの通信のプロトコルとして、TCPが使われています。

TCPでは、再送だけでなく「輻輳制御」も行います。輻輳制御とは「混雑回避」を行うことです。簡単に言うと、ネットワークが混んでいたら送るデータ量を減らす、混んでいなかったら送るデータ量を増やすという調整をします。この仕組みについての解説は本書では割愛しますので、ネットワークプロトコルの専門書をご覧ください。

注1-4：ただし、YouTubeやニコニコ動画などのWeb動画はHTTP上で動画を実現しているため、UDPではなくTCPで通信が行われています。

コネクションレス型プロトコル～ UDP

TCPがコネクション型プロトコルであるのに対し、UDPは「コネクションレス型プロトコル」となっています。

コネクションレス型の通信は、非常に単純です。簡単に言うと「送りっぱなし」です。コネクションレス型の通信では、コネクション型のようなバーチャルサーキットを構築したりしません。再送も行われず、送ったデータが相手に届くことは保証されません。また、受信側でパケットが送られた順番どおりに届くことも保証されません。

TCPが行う再送や輻輳制御は非常に便利なものです。しかし、破損したデータを送り直したり、送る量を調節したりするため、データの転送には時間がかかってしまうことがあります。それに対してUDPでは、何も考えずひたすらデータを送りつけるだけであり、TCPと比較して即時性があります。そのため、音声通話や映像配信などに利用されます。

TCPとUDPについては、Chapter 2および3でより詳細な説明を行います。

1-4 IPアドレス

広大なインターネット上で任意の機器同士が通信を行うためには、すべての機器同士が互いに「一意性を持つ」必要があります。一意性とは、「その機器が誰から見ても、その機器であるとわかるユニークな特徴、値」ということです。インターネットでは、この一意性を実現するために「IPアドレス」という識別子（番号）が使われています。

IPアドレスは32ビット（4バイト）で表現されます。32ビットで表現できる数値は10進数に直すと0 ～ 4,294,967,295（42億9496万7295、およそ43億弱）です。IPアドレスをこのような大きい数字で扱うのは現実的ではないため、人間が扱うときは8ビットずつを4つに区切り、「`11.22.33.44`」のように表現します。それぞれの区画は8ビット長なので、10進数で0 ～ 255の値をとります。すなわち、IPアドレスとして表現可能な値は、「`0.0.0.0`」～「`255.255.255.255`」になります。

ネットワーク部とホスト部

IPアドレスは、ネットワークを表現する「ネットワーク部」と、そのネットワーク部でのホストを表現する「ホスト部」に区分けされています。ネットワーク部とホスト部の範囲は、ネッ

トワークごとに異なります。そのため、ネットワーク部のビット長をIPアドレスそのものとは別途表現する必要がありますが、この表現方法としては、「/」とともにビット長を付記する記法が多く使われます。

「/」を利用してネットワーク部を示す場合、ホスト部のビットはすべて0にした値が使われることがあります。たとえば、192.168.1.2というIPアドレスのネットワーク部が32ビットのうちの上位16ビットがネットワーク部を表す場合、以下のように表現されます。

```
192.168.0.0/16
```

ホスト部で表されるホストが参加しているネットワークは、「サブネット」と呼ばれますが、255.255.0.0のようにIPアドレスと同じようにネットワーク部の範囲を「サブネットマスク」として表現する方法も一般的です[注1-5]。

IP アドレスの種類

IPには以下の3種類の通信方法があります。

- ■ユニキャスト
- ■ブロードキャスト
- ■マルチキャスト

それぞれの通信で使うIPアドレスの値や範囲が決められているため、IPアドレスの指定によってその通信方法も一緒に指定できることになります。

ユニキャスト

「ユニキャスト」は、もっとも一般的な通信方法である、1対1の通信です。次項以降に紹介するブロードキャストアドレス、マルチキャストアドレス以外のIPアドレスはすべてユニキャストアドレスです。通信相手のノードに割り当てられたIPアドレスを宛先として、通信を行います。

ブロードキャスト

「ブロードキャスト」は、その名前のとおり、すべての「ノード（機器）」に対して送信する通信方法です[注1-6]。宛先として使われるIPアドレスは、現在接続されている「サブネット」にのみ送信される「255.255.255.255」がもっとも一般的です。

サブネットとは、ネットワークの管理単位です。インターネットは多数のネットワークの集合体ですが、サブネットはもっとも小さなネットワークの単位です。IPアドレスはネットワークアドレスによって表現されるネットワーク部と、それ以外の部分で表現されるホスト部に分かれますが、ネットワークアドレスによって表現されるネットワークがサブネットです。

この255.255.255.255というブロードキャストアドレスはさまざまなプロトコルで利用されています。たとえば、空いているIPアドレスを探して割り当てる「DHCP（Dynamic Host Configuration Protocol）」というプロトコルがあります。PCなどの起動直後は自分のIPアドレスが決まっておらず、さらにIPアドレスを割り当ててくれるDHCPサーバのIPアドレスもわかりません。そこでとりあえず、IPアドレス255.255.255.255に対して「どこかにいるDHCPサーバよ、情報をくれ！」とサブネット全体に叫びます。

255.255.255.255はパケットをひとつ送れば、同じサブネットにいる機器すべてに対して送信したことになるので、その中にいるDHCPサーバも「情報をくれ！」のパケットを受け取ります。DHCPサーバは、DHCPクライアントに対して「このアドレスを使ってもいいよ」というメッセージを返し、DHCPクライアントのIPアドレスが決まります。

このように、特定の宛先を明示せずに送るという用途のために多くのプロトコルが「255.255.255.255」を利用しています。

似たような役割を持つものに、自分自身が直接接続されていない遠隔のサブネットに対する「ネットワークブロードキャスト」というものもあります。ただし、セキュリティ上の理由から現在ではローカルサブネットでのブロードキャスト以外はほとんど利用されていません。

なお、ブロードキャストのほとんどは次に紹介するマルチキャストでも実現可能です。そのため、次世代インターネットプロトコルであるIPv6にはブロードキャストの通信方法は規定されておらず、同様の処理にはすべてマルチキャストを使うようになっています。

マルチキャスト

マルチキャストは、「誰かがデータを送り、必要な人にだけデータが届けられる」という通信方法です。マルチキャストの送信者はどこにいてもよいことになっており、受信者は複数存在するという状況が想定されています。

TCPなどによるユニキャストは1対1の通信と呼ばれますが、マルチキャストは多対多の通信と呼ばれます。この「欲しい人に届く」というのがマルチキャストの不思議なところです。特別に受信者を知る仕組みをアプリケーションレベルで作成しないかぎりは、パケットを送信している送信者自身も送ったデータが誰に届いているかわかりません。また、データを中継するルータもすべての受信者を把握しているわけではありません。

マルチキャストでは、送信者は特定の「マルチキャストグループ」に対してパケットを送信します。受信者は欲しいパケットが送信されているマルチキャストグループに「JOIN（参加）」します。ネットワークはマルチキャストグループに参加している受信者を把握し続け、必要な相手へとパケットを伝えます。

データを受け取る必要がなくなった受信者は、グループから「LEAVE」すればデータは届か

注1-5：クラスA ～クラスEのIPアドレスクラスや、「CIDR（Classless Inter Domain Routing）」への移行に関する解説は割愛しています。詳細を知りたい方はTCP/IPの解説書をご覧ください。

注1-6：ルータやホストなど、インターネットに接続される機器は「ノード（node）」として表現されます。

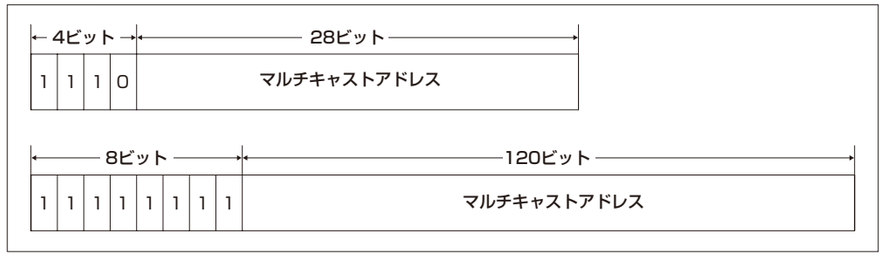
なくなります。マルチキャスト送信者は、送信するマルチキャストグループに参加していなくてもデータの送信が可能です。マルチキャスト送信者は、宛先を「マルチキャストアドレス」にすることによりマルチキャストグループに対してデータを送信します。

特定の通信がマルチキャストかどうかを判断するのは非常に簡単です。マルチキャスト通信はIPヘッダに記述してある宛先IPアドレスがマルチキャストアドレスになっています。

IPv4の場合、IPアドレスの最初の4ビットが「1110」で始まるものはマルチキャストアドレスになります。具体的には、224.0.0.0 ～ 239.255.255.255がマルチキャストアドレスになります。

IPv6では最初の8ビットが「11111111」になるもの、具体的には「ff00::0～ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff」です。

図1-7　マルチキャストアドレス



マルチキャストIPアドレスは、そのままマルチキャストグループを表しています。このマルチキャストグループとは、「そのマルチキャストアドレスのパケットを受信したい機器たち」ということになります。

マルチキャストはインターネット初期から存在していましたが、概念が非常に柔軟であるため、実現も困難な部分があります。近年、さまざまな研究や実績が積み重なり、技術としては成熟してきましたが、運用としてはまだ一般的に広く普及しているとは言い難いのかもしれません。しかし、イントラネット内で社内放送用に利用されるなど、マルチキャストを運用しやすい環境における利用は徐々に増えています。また、サブネット内でのマルチキャストなどはルーティングプロトコルや家庭内で利用するUPnPなどで必要不可欠な技術となっているため、気付かずに利用しているかもしれません。

本書ではこれ以降、特別に記述しないかぎり通信方法はユニキャストとして説明を行います。

IP アドレスの一意性

インターネット上にある個々のノード（機器）を識別するためには、IPアドレスを重複させず、一意（ユニーク）に割り振らなければなりません。そのため「アナタはこのIPアドレスを使ってもいいですよ」という割り振りを一括して行う組織が必要になります。このIPアドレスの割り振りを行うのが「IANA（Internet Assigned Numbers Authority）」です。

しかし、IANAが世界中の細かい割り当てをすべて行うわけではありません。IANAがIPv4で割り当てるのは/8単位のIPアドレスです。/8ということは、16,777,216個のIPv4アドレスを表現できます。32ビット長のIPv4アドレスには256個の/8アドレスブロックがあります。それぞれの/8ブロックは「RIR（Regional Internet Registry：地域レジストリ）」に割り当てられます。 RIRとしては以下の5つの組織があります。

- AfriNIC（アフリカ）
- APNIC（アジア太平洋地域）
- ARIN（アメリカ）
- LACNIC（ラテンアメリカおよびカリブ海地域）
- RIPE NCC（ヨーロッパ、中東、中央アジア）

各RIRは、割り振られた/8アドレスブロックをさらに「NIR（National Internet Registry： 国別レジストリ）」に割り振ります。日本の国別レジストリは「JPNIC」で、APNICに所属している組織という形になっています。

国別レジストリは、新たなIPアドレス割り当てが必要になると、所属しているRIRからIPアドレスを受け取ります。国別レジストリは、RIRから受け取ったIPアドレスを国内の組織に割り当てるとともに「誰に割り当てたのか」の情報を管理します。

このような形でIPアドレスは世界的に管理され、複数の組織が同時に同じIPアドレスを使うという事態を避けています。IANAによる現在の割り当て状況を知るには「IANA IPv4 Address Space Registry」をご覧ください。

プライベートアドレス

インターネットとは関係なく、企業内や家庭内といったローカルなネットワークの場合、世界的に一意なIPアドレスを使う必要はありません。そのような場合に自由に割り振って使える「プライベートアドレス」が用意されています。

以下のIPアドレスは、プライベートアドレスとして閉じたネットワーク内で利用できます。

- 10.0.0.0　～ 10.255.255.255
- 172.16.0.0 ～ 172.31.255.255
- 192.168.0.0 ～ 192.168.255.255

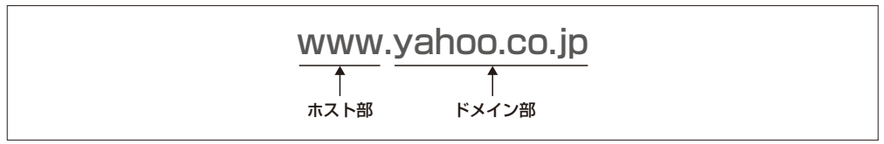
家庭用ルータなどによって自動的に割り振られるIPアドレスはプライベートアドレスであることが多いため、これらのIPアドレスに見覚えのある人もいるでしょう。

プライベートアドレスで使われているIPアドレスは、逆にインターネット上には存在しないIPアドレスともいえます。

1-5 ドメイン名とホスト名

インターネットにおける通信は、すべてIPアドレスを使って通信相手を指定します。このIPアドレスは「10.22.33.44」のように全部数字で表されますが、これでは人間にとって直感的でないうえ、非常に覚えにくいので使いにくい（設定などでミスをしやすい）のが現状です。そこで、人間がわかりやすくする仕組みとして、IPアドレスに「名前を付ける」ことを行っています。たとえば、よく見かける「www.yahoo.co.jp」や「www.google.com」などがその名前です。名前は「yahoo.co.jp」のように「ドメイン」を表している部分と、ドメインの中に存在する「ホスト」を表している「www」という部分に分けられます。

図1-8　ドメインとホスト部



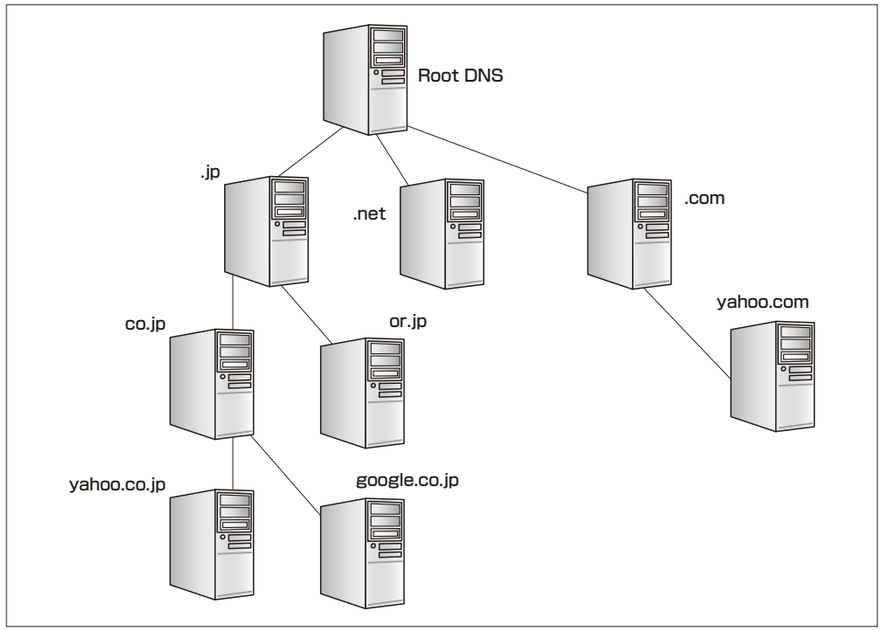
これらの名前は、実際には数字だけのIPアドレスに変換されてから通信が開始されます。この名前からIPアドレスへの変換を行ってくれるのが「ドメインネームシステム（Domain Name System、以下DNS）」です。そして、DNSを実行しているサーバが「DNSサーバ」です。

ドメインネームシステム

名前とIPアドレスの変換は、名前を変換したいと思ったノードがDNSサーバに対して問い合わせを行い、DNSサーバからはその名前に対応するIPアドレスを受け取ります。このIPアドレスを使うことで、目的の相手と通信できるようになります。

ただし、DNSサーバは43億弱あるIPアドレスに対応する名前をすべて知っている魔法の箱というわけではありませんし、そもそも1カ所で把握管理するのは不可能です。そこで名前規則に階層を持たせ、各DNSサーバは自分の守備範囲だけを管理する仕組みになっています（図1-9）。

図1-9　DNS サーバと名前解決の仕組み



DNS階層の一番上には「ルートDNS（Root DNS）」と呼ばれるDNSサーバがあります。そしてそのルートDNSの下には、「TLD（Top Level Domain）」と呼ばれるドメインを管理するDNSサーバがあります。日本のドメインであることを表すTLD「.jp」もルートDNSの下にあります。

日本のドメインである「.jp」の下には「.co.jp」や「.or.jp」などがあります。さらに、「.co.jp」の下に「.yahoo.co.jp」や「.google.co.jp」のドメインがあるという構造になっています。

各階層を管理しているDNSサーバは、それ以下の階層の作り方を自分で管理できます。たとえば「.yahoo.co.jp」を管理している団体は、さらに下にサブドメインを作ったり、「www.yahoo.co.jp」というホスト名を登録したりできます。

このように、インターネットでは各ドメインの管理（名前管理）を分散させることによって、負荷を1カ所に集中させることのない運用を実現しています。

1-6 Webとメールの仕組み

インターネットの用途でもっとも多いのはWebとメールです。ここでは、毎日何気なく利用しているWebの技術的な動作について説明を行うとともに、ここまで説明してきたインターネットの基礎的な仕組みについて復習したいと思います。

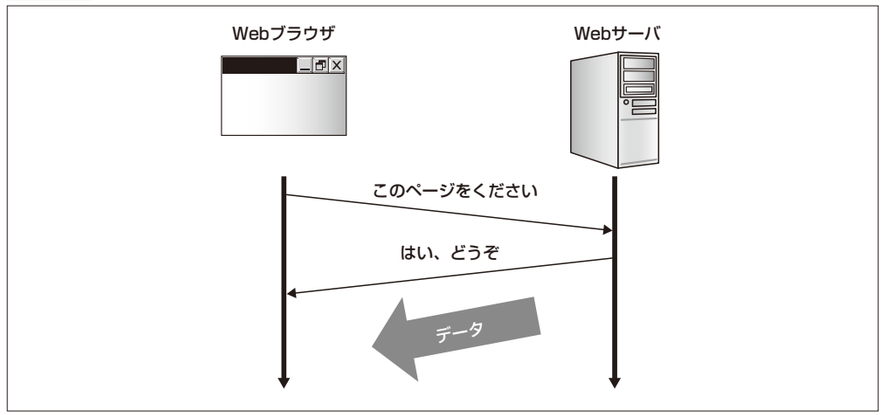
Hyper Text Transfer Protocol

いまや、「http://」という文字列を見たことがない人はいないのではないかというぐらい、この文字列は世の中に溢れています。雑誌や電車のつり革広告には当たり前のように「http://」で始まるURLが書いてありますし、テレビのコマーシャルにも出てきます。いったい「http://」とは何なのだろうと不思議に思ったことはありませんか？

インターネットでWebを閲覧するときには、「Hyper Text Transfer Protocol（以下、HTTP）」というプロトコルを利用して通信が行われます。実は「http://」という文字列は、「プロトコルはHTTPを使いますよ」と説明しているのです。すなわち「インターネットでの通信プロトコルを自ら積極的に指定して通信を行う」という作業を、ほとんどの人が知らないうちに経験しているのです。

以下に単純なHTTPの動作を示します。

図1-10　HTTP の動作



HTTPというプロトコルの基礎的な部分は、非常に単純でわかりやすくなっています。ユーザの手元にあるPC内で動作しているWebブラウザ（Internet Explorer、Firefox、Safari、Opera、Sleipnirなど）は、Webサーバに対してTCPでコネクションを確立します。できあがったTCPコネクションを通じて「このURLの中にあるデータをください」というとサーバは「はい、どうぞ」とデータを渡してくれます。Webブラウザは、受け取ったデータの形式に応じて適切な方法で表示を行います。

「http://」以外にも、「ftp://」や、もう古すぎて使われていない「gopher://」などがあります。通信を暗号化させて途中にいる第三者に通信内容を読み取られないようにしたHTTPとして「https://」というのもあります。このような表現方法を「URL（Uniform Resource Locators）」と呼びます[注1-7]。

Web ブラウザが行う URL の解釈処理

今度は、Webブラウザの「アドレス入力部分」にURLを指定することで、どのような処理が行われているかをもう少し詳しく説明していきます。

まず、ブラウザのアドレス欄にURLとして「http://www.yahoo.co.jp/index.html」を打ち込み、実行ボタンを押したとします（図1-11）。

画面1-1　ブラウザの実行ボタンを押す

まず最初にWebブラウザは、

■「http://」で始まっているので、このURLはHTTPを使って通信を行う

と考えます。そして、「http://www.yahoo.co.jp/index.html」の中にある、

■「www.yahoo.co.jp」部分がWebサーバのホスト名を表している

と認識します。インターネットで通信を行うにはWebサーバのIPアドレスが必要なので、「www.yahoo.co.jp」のIPアドレスをDNSサーバに問い合わせます。DNSサーバからホスト名に対応するIPアドレスを受け取り、WebブラウザはそのIPアドレスに向けてTCPコネクションを確立しようとします。

注1-7：URLと同じように使われる「URI（Uniform Resource Identifier）」があります。URIは、リソース（資源）を示す識別子であり、URLよりも広義な意味を持ちます。URIの厳密な定義に関しては、RFC3986、"Uniform Resource Identifier（URI）: Generic Syntax"、2005年1月、http://www.ietf.org/rfc/rfc3986.txtをご覧ください。

TCPコネクションを確立するためには、ブラウザ側のホストからWebサーバ側のホストまでパケットが届かなければなりません。それには、まずブラウザ側ホストからパケットが送信されなければなりませんが、ブラウザ側ホストに複数のネットワークインターフェースがある場合、「どのネットワークインターフェースを使ってパケットを送信すべきか？」を判断しなければなりません。Webブラウザが動作している機器の「OS（Operating System）」は、手元にある経路情報を確認して、どのネットワークインターフェースを使えばパケットを目的地に送信できるのか調べます。その結果、機器に付属しているイーサネットカード経由でパケットから送信すればWebサーバに送ることができるとわかります。

これらの情報がわかってから、Webブラウザが動作しているノード（機器）は、TCPコネクション確立に必要な情報を含むパケットをネットワークインターフェースに送信します。

TCPコネクションを確立するにはブラウザ側が「TCPのコネクションを確立しましょう！」という要求を含むパケットをWebサーバまで送信しなければなりません。Webサーバ側は「はい、TCPコネクションを確立しましょう」と返し、ブラウザ側が「コネクションを確立し終わりました」という返答をさらにすることでTCPコネクションが確立します。

Webブラウザが動作している機器から送信されたTCPパケットは、まずイーサネット経由でルータに届きます。ルータは、TCPパケットのIPヘッダを見て転送する先を考えます。転送先は、ルータが持っている経路表を基に確定されます。転送先が確定すると、ルータは次のルータに対してTCPパケットを転送します。次のルータはまた次のルータへ転送する、というようにバケツリレー的にパケットが転送されていき、最終的にTCPパケットはWebサーバに到着します。

Webサーバが転送されてきたTCPパケットを受け取ると、Webブラウザがあるノードに対して返事をします。このときWebサーバは、TCPパケットにある送信元のIPアドレスから返信先を指定します。このようにして、ブラウザの機器とWebサーバの間にTCPコネクションが確立します。

Webブラウザは、作成したTCPコネクションを使い、HTTPプロトコルでwww.yahoo.co.jp内の「/index.html」という「パス（path）」で表されるデータを要求します。Webサーバは、それに応答する形でページのデータを送信します。送信されたデータの記述を見て、ブラウザはページを表示する、というわけです。データの記述には「HTML（Hyper Text Markup Language）」という言語が一般的に使われます。ただし、画像やそのほかの言語によるデータがHTTP上に流れたりもします。

メールの仕組み

メールもまた、インターネットで非常によく利用されている、電子データで構成されるメッセージをやり取りするサービスです。普段何げなくメールアプリケーションを利用していると思いますが、なぜメールが相手に届くのか不思議に思ったことはないでしょうか？　ここでは、メールアプリケーションはどのようなプロトコルを使ってネットワーク上で通信を行っているのかについて説明します。

まずは、メールアドレスの意味について説明します。メールアドレスのほぼ中央には、「@（アットマーク）」が入っています。この「@」の左右にある文字列は、それぞれに意味があります。

@の右側は、メールサーバの名前（ホスト名もしくはドメイン名）を表しています。そして@の左側はそのサーバ内でのユーザ名を表しています。たとえば、「username@domainname.or.jp」というメールアドレスがあった場合、「domainname.or.jp」がメールサーバ、「username」がユーザ名を表します（図1-11）。

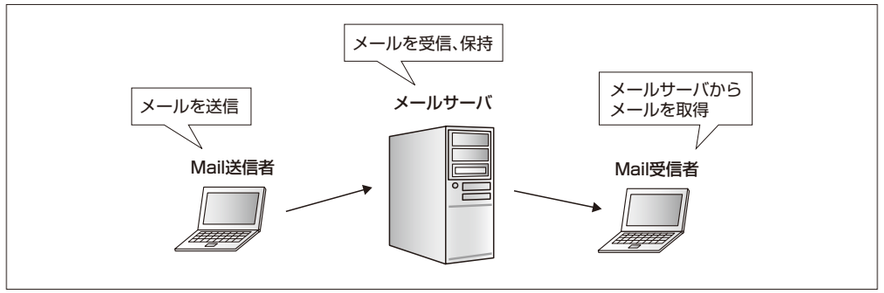
図1-11　メールアドレスの意味

このようにして、メールの宛先を表現しています(注1-8)。

さて、メールアプリケーションはメールの送信と受信ができます。当たり前と思われるかもしれませんが、実はメールの送信と受信ではネットワーク上で利用されるプロトコルはまったく別のものです。受信するプロトコルの代表的なものとしては、「POP3（Post Office Protocol version 3）」や「IMAP（Internet Message Access Protocol）」などがあります。送信するプロトコルとしては「SMTP（Simple Mail Transfer Protocol）」が代表的ですが、本書ではこれらプロトコルについての詳細な解説は割愛します。すなわちメールアプリケーションは、インターネットを利用してメールを送信するプロトコルとメールを受信するプロトコル、その両方を使えることになります。

メールシステム全体を非常に大まかに表したのが図1-12です。

図1-12　メールの仕組み



注1-8：メールアドレスは大文字と小文字を区別しません。「username@domainname.or.jp」でも「USERNAME@DOMAINNAME.OR.JP」でも「UserName@DomainName.OR.jP」でも届きます。

まずメール送信者は、送りたい相手のメール情報を管理しているサーバ（メールサーバ）に向けてメールを送信します。メールサーバは、送られてきたメールデータを受信し、サーバ内に保管します。メール受信者は、メールサーバに保管されたメールがあるかどうかを確認し、メールがあればメールサーバからメールをダウンロードします。

メールサーバという存在がこのメールシステムのポイントです。メールサーバは常時起動していて、常にメールを受け付けます。そして受信したメールを一時的に保持するため、メールの送信者と受信者が同時にネットワークに存在しなくても大丈夫になっています。

またメール受信者側にとっても、メールサーバにアクセスさえできれば、どこにいてもメールを確認、取得できるのが大きなポイントです。家や職場、あるいは旅行先でも、変わらずにメールを受信できます。メール送信者にとっても、受信者の状況を考慮してメールの送り方を変える必要はありません。

1-7 IPv4とIPv6

ここまで解説してきたIPアドレスは、現在広く利用されている「IPv4」によるアドレス（以下、IPv4アドレス）です。IPv4アドレスは32ビット長で、表現できるアドレス数は2の32乗、約43億個です[注1-9]。

インターネット設計当初は、これだけのアドレス空間があれば十分であると考えられていました。しかし、インターネットが世界中に広がると、IPv4で表現できる量では足りないということがわかってきました。これは、人口の多い発展途上国がインターネットを使い始めたり、すでに使っている人でも職場と家など一人で複数のIPv4アドレスを利用したりするためです。あわせて、ダイヤルアップ接続などといった一時的な接続ではなく、ブロードバンドによる常時接続が普及してきたという背景もあります[注1-10]。

世界中の「まだ使われていないIPv4アドレス」はIANAが管理しており、「IANAプール」と呼ばれます。このIANAプールから各国のIPアドレス管理団体へとIPアドレスは付与されていきます。現在、このIANAプールがなくなるのが2011年半ばから2012年初頭と予測されています。ただしそれはIANAのプールの話であり、日本で新規割り当てができなくなるのはもう少しあと、2012年初頭から2013年中と予測されています[注1-11]。

この新規割り当てができなくなると、新しいユーザに対するIPv4アドレス配布ができなくなります。その時点でIPv4アドレスを保持している組織やユーザには影響はありませんが、IPv4アドレス在庫が枯渇すると、インターネットへの新たな接続や、さまざまな事業やサービスの新規参入が阻害、ビジネスの障害になり得ます。そのため、いまのうちにIPv4アドレスが完全に枯渇しないような工夫が求められています。

現在、IPv4アドレスの枯渇に対応するための手段として注目されている技術が、ISPレベルでの「Large Scale NAT（Network Address Translation）」によるIPv4アドレスの節約と、IPv4からIPv6への移行です。

NAT/NAPT（Network Address Port Translation、以後両方を合わせてNAT）技術やLarge Scale NATについては本書では扱わないので、ネットワークの専門書籍をご覧ください。

IPv6への移行

IPv4アドレス枯渇の対策として提案されているISPによるLarge Scale NATは、ほぼIPv4ユーザしかいない現状のインターネットに対しては最善の回避策であると考えられます。

しかし、これは不完全な回避策です。NATはIPアドレスやポート番号を変更してしまうため、P2Pアプリケーションなどの通信を阻害しますし、ネットワーク構成によっては各ユーザが同時に行えるセッション数も制限される場合があります。現状のインターネットは、度重なるつぎはぎによって運営されていますが、さらにそこに新たなつぎはぎを加え、ユーザの利便性を損ねてしまう可能性があります。

現在の多くの通信環境と同様にユーザがインターネットを使えるようにするには、現在のIPv4から「IPv6」へと移行する必要があります。

IPv6のIPアドレスは128ビットによって表現されます。ぱっと見、32ビットの4倍程度で間に合うのかと思うかもしれませんが、ビット数が4倍になると表現できるアドレス空間は2の96乗倍になります。これは天文学的な数値です。実際にはIPv4同様に128ビット空間すべてを満遍なく使えるわけではありませんが、それでも膨大な数になります。この膨大な数のIPv6アドレスを利用することで、いままでと同様の通信環境が実現できます。

しかし、ほぼすべてがIPv4で構成されているインターネットをいきなりIPv6へと移行させるのは不可能です。そのため、IPv4アドレスの枯渇直後はNATによる回避策を実施しつつ、並行してゆっくりとIPv6への移行が行われていくものと思われます。

理想的には、ユーザがまったく意識せずにIPv4からIPv6への切り替えが行われるべきではあります。しかし、実際にはすべてのユーザに対してそのような状況を作り上げることは容易ではありません。世界を大きく変えたインターネットのベースとなっているIPv4をIPv6へとアップグレードする作業は、未知への挑戦といえます。

とはいえ、IPv4アドレスの枯渇そのものは不可避です。根本的な代替案としてIPv6以外には存在せず、IPv6に関する知識はネットワークプログラミングを行ううえでも必須知識になっていきます。そのため本書では、IPv6とIPv4両方を区別せずに扱えるプログラミングコードをベースに解説していきます。従来のIPv4のみが扱えるレガシーなプログラミング手法については、巻末のAppendixに記載します。

注1-9：IPアドレスは、インターネットに接続された機器そのものではなく、機器に付属されている各ネットワークインターフェースに付きます。さらに、ひとつのインターフェースに複数のIPアドレスを付けることも可能です。また、プライベートな利用や、マルチキャストアドレス用途として規定されているIPv4アドレス空間も存在するので、43億個の機器がインターネットに接続できるというわけではありません。

注1-10：細かい時間単位でIPv4アドレスの割り当てと返却を行いにくくなるため、常時接続の方がサービスを提供するために必要となるIPv4アドレス数が多くなります。詳細は「インターネットの円滑なIPv6移行に関する調査研究会 報告書（平成20年）」を参照してください。

注1-11：IPv4アドレス枯渇時期に関しては、"potaroo IPv4 Address Report"（http://www.potaroo.net/tools/ipv4/）を参照してください。

1-8 Chapter 1のまとめ

Chapter 1では、基礎知識としてインターネット仕組みや通信方法を説明しました。実際には、これらはOSやライブラリなどがうまく隠してくれるため、通信プログラムを作成するうえで必須の知識ではありません。しかし、これらを知識として身に付けていることによって、よくわからない症状が現れた場合など、何か困ったときには役に立ちますし、効率の良いプログラムが書きやすくなります。

本章の解説は概要にすぎません。インターネットの仕組みそのものに関しては良書が多々あるので、ぜひそれらもご覧ください。