IPv6 入門

～Internet Protocol Version 6～

1999 年 12 月 15 日

社団法人　日本ネットワークインフォメーションセンター
宇井　隆晴

Internet Week 99　パシフィコ横浜
（社）日本ネットワークインフォメーションセンター編

目 次

1 概要

本チュートリアルでは、導入として「インターネットへの道のり」と題し、歴史的な話を踏まえたうえでインターネットが現在までにどういう発展をしてきたのか、いまどういう形になっているのかということを少し簡単に説明します。

次に、IPv4 の話に移り、現在の IPv4 がどのような問題を抱えているのかということを踏まえたあとで、それらの問題を解決する次なるプロトコルとして（本チュートリアルの中心である）IPv6 の説明に入ります。

2 インターネットへの道のり

ご存知の方も多いかと思いますが、現在のインターネットの元となったのはアメリカ国防総省の「ARPA Project」というものです。これは、核戦争にも耐えられるネットワークを作るという計画に端を発しています。

この ARPA Project は、コンセプトとしては

- 障害に強い分散型のネットワークであること
- 単純だが確実な転送のできるプロトコル

という 2 つが求められていました。

そして、このプロジェクトの中で 1970 年代に Ethernet という技術が開発されました。この Ethernet を基盤としたネットワーク、これが現在までの約 30 年間に及ぶインターネットの歴史の根源となっているものです。

2.1 中央集中から分散へ

では、この“核戦争にも耐える”という要求のコンセプトを確認しましょう。

当時のコンピュータのネットワークは、（現在でも用いられていますが）中央に機能を集約したホストコンピュータが存在し、ユーザーは遠隔地から端末と回線を通してホストコンピュータ上の機能を利用するという形をとっていました。

このような形は利用という面では便利ですが、戦争のようなトラブルが起こったとき

に中央のコンピュータに攻撃が与えられ破壊されるとすべての機能が停止してしまうという問題を含んでいました。

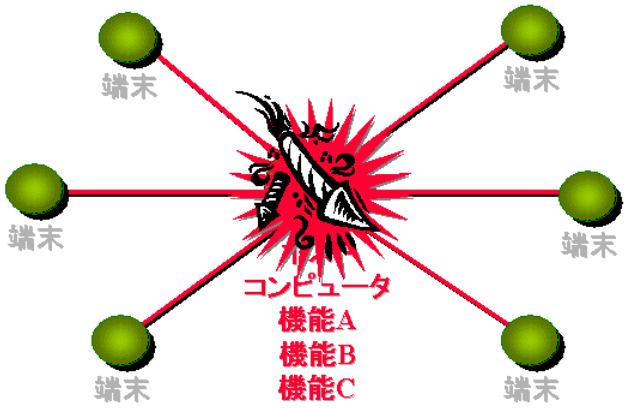

図 1　中央に機能を集中したネットワーク

このように、中央にすべての機能を集めているような構成だと攻撃に対して非常に弱くなってしまいます。そこで、攻撃に弱いシステムではなく、攻撃に強いシステムとして機能を分散させる方向に動いていきました。

2.2　分散化のメリット

そのためには、まず、ホストと、それを利用する端末という考え方をなるべく捨てていきます。すべてのコンピュータは互いに同等の位置に付き、機能はなるべくネットワーク上に分散させるようにします。

ネットワークは中央に集まる一方で、全体に網の目のように張り巡らされます。そのうえで機能を分散させておくと、どこかが攻撃を受けてダメージを受けたとしてもすべての機能が停止するという事態は避けられます。

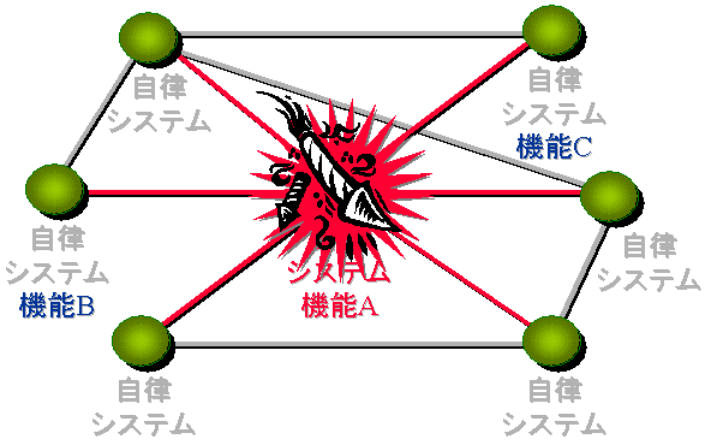

図 2　機能の分散化

実際にはネットワークの張り巡らし方によって変わりますが、たとえばここで紹介したような構成にしておくと、以前の状況と同じように中央にあるコンピュータが破壊されても使えなくなるのは機能Aだけで、生き残ったコンピュータは機能Bや機能Cを使い続けることができます。

このように、攻撃を受けても生き残った他の部分で継続できることを目標に研究が進められていきました。

2.3 internet から The Internet へ

このようにして始まったインターネットは、ARPAが軍事プロジェクトであったために、当初はアメリカの研究所や大学などの相互接続から始まりました。しかし、時代とともにその利便性が一般に認められるようになると、企業とか一般団体へとそのネットワークが拡大していきます。

そして、現在のように国境を越えて全世界的なネットワークへと発展してきました。目的も、当初の軍事利用から大学などの学術研究、そして現在に至る商用などへの一般化が変化として見られてきています。

それに伴い、ネットワークの相互接続という「inter-networking」を語源とした「internet」から固有名詞的な「The Internet」と呼ばれるようになり、それが定着しています。

2.4 技術の標準化 —— RFC

インターネットの普及を語るうえで忘れてならないのが、技術の標準化です。

インターネットを支える技術的な発展は、RFC（Request for Comments）という仕組みによって支えられています。RFC は、インターネットにかかわる新しい技術やアイディアを文書化したもので、一般に公開され、誰もが閲覧することができます。

公開された文書を基にした技術は誰でも利用できますから、好きなように実装して好きなように利用するということが可能になります。すばらしい技術であれば広範囲で使われるようになりますし、そうして広まっていった技術が結果として標準化の道をたどることになります。どこからか、この技術を使いなさいとか、このプロトコルを使いなさいと強制されるわけではありません。それで、いわゆる「インターネットは常に実験場である」となるわけです。

2.4.1 代表的な RFC

ここで、代表的な RFC をいくつか紹介しましょう。

821 番で「SMTP」というプロトコルが定義されていますが、これは皆さんがメールをやりとりするときに使っています。959 番の「FTP」はファイルを転送するときに使い、1661 番の「PPP」は家から電話をかけて ISP に接続するときに使っています。

メールボックスから自分のコンピュータにメールを取り込む「POP3」というものも RFC の 1939 番で決められていますし、WEB サーバとブラウザがやりとりするデータの形式も RFC2068 で決まっています。

また、このようなプロトコル系の RFC だけでなく、たとえば 2564 番のセキュリティハンドブックのように、セキュリティを確保するときにはこういうことをしなさい、こういうことを考えなさいといった情報提供をするものもあります。

2.4.2 joke RFC

RFC の中には、いわゆる「joke RFC」も存在します。たとえば 1149 番には「鳥類キャリアによるIPデータグラムの転送に関する標準」という名前が付いたものがありますが、これはデータを運ぶときに鳥を使おうというものです。また、2324 番では「HTCPCP(Hyper Text Coffee Pot Control Protocol)」として、遠隔からいかにコーヒーポットを操作するかといったことが記述されています。

RFC1149 は、のちに RFC2549 で QoS(Quality of Service)を実現するための拡張が提案されたりもしていますが、こうした joke RFC は毎年 4 月 1 日に公開されるといった恒例行事になっています。内容は RFC の形に添った真面目なふりをしていますが、ではまるまるジョークかというとそうではありません。それを見るために、伝書鳩プロトコルをちょっと見てみましょう。

2.4.3 伝書鳩プロトコルの心

伝書鳩プロトコルは、簡単に言うと紙にデータを書き出して巻物のようにし、それを伝書鳩の片足に結んで飛ばしてしまおうというものです。その結果として、高い遅延性と低いスループット、低い高度でのサービスが提供できるとしています。

このあたりはいかにもジョークなわけですが、1 次元ではなく 3 次元の空間を利用できるとか、早春を除くという繁殖期の条件がかかるとはいえ干渉させることなく使用できるとか、本能的に衝突回避システムを持っているといったメリットも述べています。半分冗談で、半分はプロトコルの面で真面目なことが書いてあるわけですが、こうした発想はインターネットの思想から外れているわけではありません。

3 インターネットプロトコル（IPv4）

では、今回のテーマである IP の話として、まずは現在用いられている IPv4 がどのようなものであるかについて説明していきます。

3.1 プロトコルとは何か

先ほどから話の中で何回も「プロトコル」という言葉が出てきました。この「プロトコル」というものを簡単に説明してしまうと、「通信を行うための手順や決め事」ということになります。

たとえば日常社会において、手紙といったものを考えてみます。手紙を出す場合には、表に郵便番号、住所、氏名を書いてポストに投函するという一連の作業を行います。これは、手紙という手段を用いて通信を行うための手順であり、決め事であり、プロトコルであると言えます。

また、電話であれば相手の電話番号を入力し、たとえば「もしもし、○○ですが」と話し出します。これも、相手と電話を使って話をするときのプロトコルです。

誰かと会談を行いたいときは、あらかじめ相手とアポイントを取り、時間と場所を取り決め、最低限 5 分前には到着しましょうということを約束しますが、これも相手と会談を持つことに関するプロトコルになるわけです。

これらは日常社会的なプロトコルですが、ここでネットワークでのプロトコルというものを考えてみたいと思います。

3.1.1 ネットワークでのプロトコル

先ほどの考え方を使って説明すると、ネットワークにおけるプロトコルとは「情報をネットワークを通した相手とやりとりするための手順」であるということになります。

たとえば電子メールを相手のメールボックスに届けるとき、その中ではどういったやりとりを行うのか、どういう情報を流すのかといったことを決めるのが電子メールのプロトコルですし、ブラウザから WWW サーバにアクセスしてホームページを見るとき、ブラウザはサーバと何をどのようにやりとりするのかを決めるのが Web のプロトコルです。

また、ネットワーク通信ですから、ネットワークケーブルを通じていろいろとやりとりをしていますが、その中にはどういった信号が流れるのかということもネットワーク

での通信プロトコルということになります。

3.1.2 電子メール送信の裏側

電子メールアプリケーションを用いて電子メールを作成し、インターネットを使って相手の電子メールアプリケーションまで届けることを考えてみます。この図では、いわゆる「電子メール」という情報が流れるわけですが、イーサネットと書かれている部分は実際には FDDI であったり電話回線であったりします。

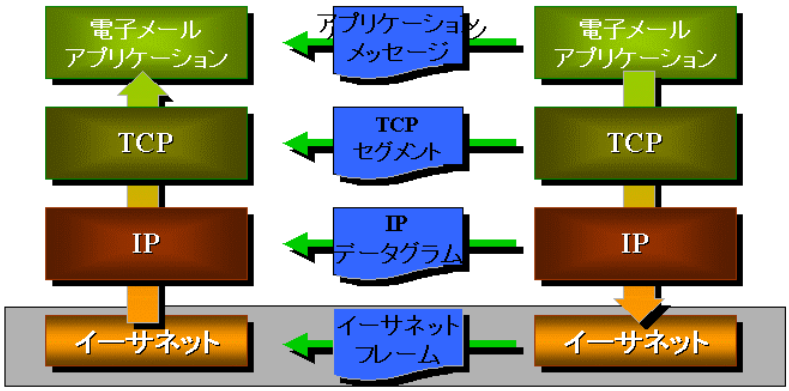

図 3 電子メール送信の裏側

ある電子メールアプリケーションが別の電子メールアプリケーションに電子メールを届けるといった作業をするときに、ネットワーク部までいったん情報を落とします。そして、ネットワークを通して相手の電子メールアプリケーションまで電子メールを運ぶという一連の作業を行うわけですが、電子メールアプリケーションがネットワークに直接データを渡すわけではありません。

3.1.3 階層的プロトコル

プロトコル的に考えると、電子メールアプリケーションはアプリケーションメッセージを「TCP」に送ります。TCP はアプリケーションメッセージを受け取ると、自分のデータの形式である「TCP セグメント」というものに作り変えます。そして、これを相手の TCP まで送るわけです。

しかし、この TCPもネットワークと直接話をするわけではありません。その下に今回お話しする「IP」が存在しています。IP は TCP から TCP セグメントを受け取ると、自分のデータ形式である「IP データグラム」に作り変えます。そしてこの IP データグラムをネットワークに渡し、ネットワークは自分のデータ形式である「フレーム」と呼ばれる形式にして相手のネットワークまで送るという作業を行います。こういった形を「階層的プロトコル構造」といいます。

このように階層構造を使う理由は、インターネットで用いられるアプリケーションが

電子メールだけではないからです。Web もありますし、Real Player、Real Server といったようなストリーム配信もあります。

そうしたときでも、たとえば IP というプロトコルが 1 枚入ることによって、アプリケーションはネットワークが実際に何であるかを意識する必要はなくなります。同時に、ネットワーク側から見れば、上位で動いているアプリケーションが何であれ IP だけを見ればよくなります。

図 4　階層的プロトコルを使う理由

階層構造を取ることによって、このように大きな柔軟性が生まれます。

3.1.4　インターネットのプロトコル

では、インターネットにおいて、この階層的なプロトコルがどうなっているかということを示します。

インターネットでは、この層が大きく 4 つに分かれています。一番上がアプリケーション層、次がトランスポート層、そして今回お話しする IP はインターネット層というところに位置しています。そして、一番下にネットワークインタフェース層があります。

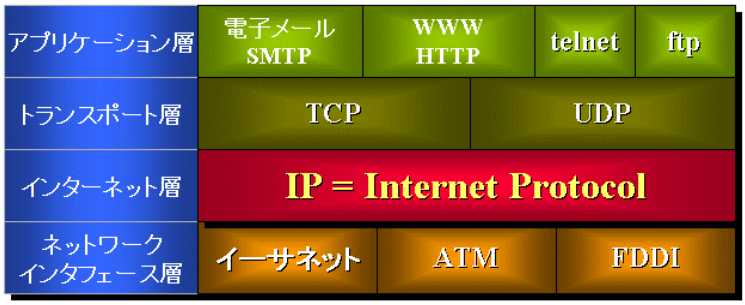

図 5　インターネットのプロトコル

アプリケーション層は、その名が示すようにさまざまなアプリケーションのプロトコルが並びます。そのアプリケーションメッセージを受け取るトランスポート層は、TCP

とUDPというプロトコルがあります。インターネット層にはIPがあり、一番下のネットワークインタフェース層にはネットワークの種類だけプロトコルが存在します。たとえば、Ethernet であるとか、ATM や FDDI であるということになります。

この図を見ると、さまざまなプロトコルが各レイヤにありますが、インターネット層のところだけ IP というひとつのプロトコルだけになっています。この IP こそがインターネットの基盤を支えているプロトコルです。

3.2 インターネットプロトコル（IP）

インターネットプロトコルが行う作業は、アプリケーションからの情報をネットワークフレームサイズに収まるように分割し、情報の送り先や送り元などのヘッダ情報を付加し、分割した情報とヘッダ情報を IP データグラムとして構成することです。図の中に TCP/UDP ヘッダがあるのは、IP にくる前にトランスポート層を通ってくるからです。

図 6 インターネットのプロトコル

IP の目的は、IP データグラムを始点ホストから終点ホストまで転送することです。そうすると、インターネット上で送り先をどのように特定するのかという問題を解決しなければいけません。

ご存知のように、現在のインターネットには非常に数多くのホストが存在しています。したがって、この送り先がどこにあるのかということをヘッダの中にどのように書き表すのかが重要になります。このために用いられるのが「IP アドレス」と呼ばれるものです。

3.2.1 IPv4 アドレス

電話は、電話番号で相手を一意に特定します。手紙の場合は、住所と名前で相手を特定します。IP ネットワークにも相手を特定するための何かが必要になります

が、その IP ネットワークの住所と呼べるものが IP アドレスです。

現在用いられている IPv4 アドレスは、インターネット上でホストを一意に特定するための番号として 32 ビットのゼロと 1 の並びを使っています。そしてこの IP アドレスは、IP で通信を行う全てのホストに付けられます。もし、ある IP アドレスを複数のホストに付けてしまうと一意に特定できなくなり通信がうまくできなくなりますから、それはルール違反ということになります。

ところで、IPv4 アドレスは 32 ビットのゼロと 1 の並びだということを説明しましたが、それではコンピュータにとっては問題がなくても人間が簡単に覚えられるような性質のものではありません。そのために、人間のための表記方法が決められています。

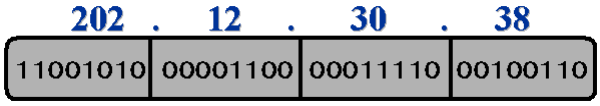

図 7 IPv4 アドレス

人間のための表記は、32 ビットを 8 ビットずつに区切り、それぞれを 10 進数で表わしてピリオドで区切ります。8 ビットは 0 から 255 までの数を示せますから、図の例では“202.12.30.38”となります。

3.2.2 ネットワークの識別

実は、相手を IP アドレスによって特定しただけでは実際に通信を行うことはできません。理由は、送り先がわかっても、どのような経路を通っていけばその相手にたどり着けるのかがわからないからです。インターネットはひとつのネットワークではなく数多くのネットワークの集合体ですから、相手がどのネットワークにいるのかがわからなければいけません。

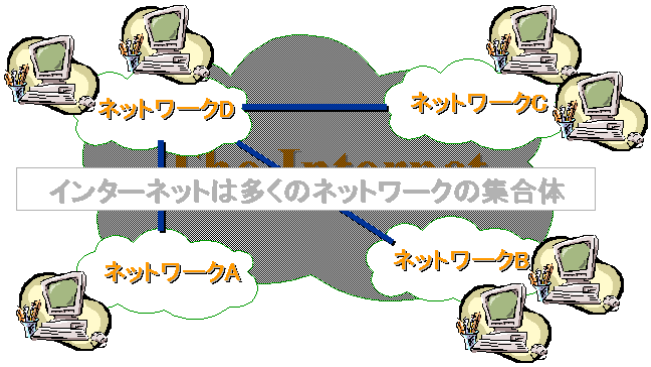

図 8 インターネットはネットワークの集合体

そのためには、IP アドレスによって、相手が属しているネットワークの識別と、相手のホストを特定できるようにする必要があります。

こうしたことは、たとえば電話では実際に行われています。電話をかけるとき、まず市外局番を入れると地域が特定されます。“03”と入れれば東京ですし、“0565”と入れれば愛知県豊田市になります。同様に、IP アドレスでもそのアドレスの一部を見ることでネットワークを特定することができ、全体を見ることでホストを特定できるようになっています。

3.2.3 IP アドレスの意味付け

この 32 ビットのアドレスに対する意味付けを見てみましょう。

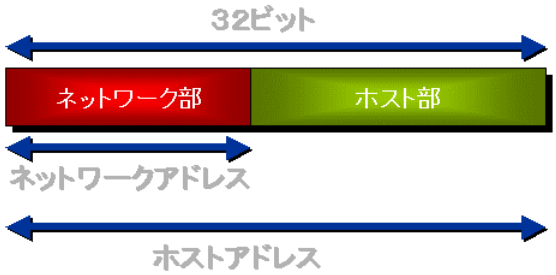

図 9　IP アドレスアドレスの意味付け

IPv4 アドレスは図のように「ネットワークを表わす部分」と「ホストを表わす部分」に分けることができます。先頭から n ビットのネットワーク部を示す部分をネットワークアドレスとし、全体を見たときの 32 ビットをホストアドレスとして、この 2 つに対して意味付けを行います。

3.2.4 ネットワーク部の大きさ

ネットワークアドレスを説明では単に「先頭から n ビット」と紹介しましたが、このネットワーク部分の大きさをいくつにするかによって使えるネットワークアドレスの数とホストの数というものが決まりますから、実際にはどんな数でもいいというものではありません。

電話の例で言えば、市外局番を 2 桁として固定したとすると“00”から“99”まで 100 通りの地域を特定することができます。しかし、100 通りという数字が適正かどうかは事情によります。

たとえばこのネットワーク部分を 8 ビットとして考えてみます。

8 ビットあれば、ネットワークは 256 パターンを定義できます。つまり、インターネット上にネットワークが 256 個まで存在することができるということになります。全体が 32 ビットだとすると残りが 24 ビットになりますから、これだと約 1,600 万個のホストを定義することができるようになります。

インターネット上に256個のネットワークというのは非常に少ない数でしょうから、ではネットワークを16ビットとしてみます。16ビットでは、ネットワークとホストの双方をそれぞれ約 65,000 個ずつ定義できます。

では、ネットワーク部の長さはいったいいくつにすればいいのかという話になるわけですが、インターネットにはさまざまな規模のネットワークが存在します。たとえば、ホストが 3 つしかないネットワークもあれば、百万以上のホストを持っているネットワークがあるかもしれません。

こうしたさまざまなパターンに対応するためには、このネットワーク部の大きさというのは固定にしないほうがよいということになります。

そこで、IPv4 ではこのネットワークの部分の大きさというものを何パターンか用意しています。その識別は「クラス」を示すビットで示され、最初のビットがゼロであれば、この IPv4 アドレスは「クラス A」であるということになります。

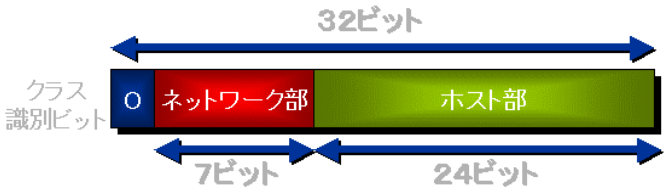

図 10　クラス A アドレス

クラスAの場合は、先頭の1ビットに続く7ビットがネットワーク部となり、インターネット全体で 128 個のネットワークを定義できます。また、このネットワークの中には約 1,600 万ものホストを持つことができます。したがって、これは非常に大規模なネットワーク用のアドレスクラスということになります。

次に、「クラス B」というものを紹介します。

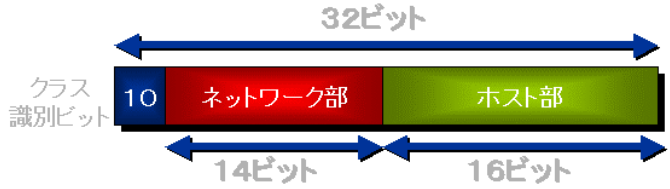

図 11　クラス B アドレス

これは、先頭の 2 ビットが“10”となっていることで識別されます。このとき、続く 14 ビットがネットワーク部となり、インターネット全体で 16,384 個のネットワークを定義することができます。残りの16ビットを使って定義できるホストの数は65,536になります。

さらにもうひとつ。「クラス C」というものがあります。

図 12　クラス C アドレス

これは、先頭から 3 ビットが“110”のパターンとなることで識別されますが、続く 21 ビットがネットワーク部となります。このパターンだと、インターネット全体で約 200 万ものネットワークを定義できます。ただし、ホストは8ビットで表わせる範囲しか使えませんので、ひとつのネットワークで定義できるホスト数は 256 個に限定されます。したがって、これは小規模なネットワーク用のアドレスということになります。

このように、IPv4 にはクラスA、クラスB、クラスCというアドレスクラスが存在します。そのうえで、ネットワークアドレスというものをもう一度見てみます。

IP アドレスの先頭に「クラス識別ビット」というものがあるわけですが、これを見ることによってアドレスクラスが識別可能になり、ネットワーク部を何ビットで示しているかが判別できます。ネットワークアドレスとは、ホスト部をゼロで埋めてネットワーク部だけを示したものです。

このネットワークアドレスによって、IP データグラムを送りたいのはどのネットワークであるのかを示せるようになりました。

3.3　IP データグラムの転送

異なるネットワーク上にあるホスト間で IP データグラムを転送する場合を考えます。

図 13　IP データグラムの転送

異なるネットワークをつなぐときには、「ルータ」を使います。このルータは、たとえ

ばネットワークＡから受け取ったIPデータグラムをネットワークＢに転送するという作業を行います（IP を使う上で、ルータは絶対に考慮しなければいけない装置です）。

次に、ルータに関連して、IP データグラムの転送経路はどのように決まるかを考えます。図のように、複数のネットワークがつながっている状態を想定しましょう。

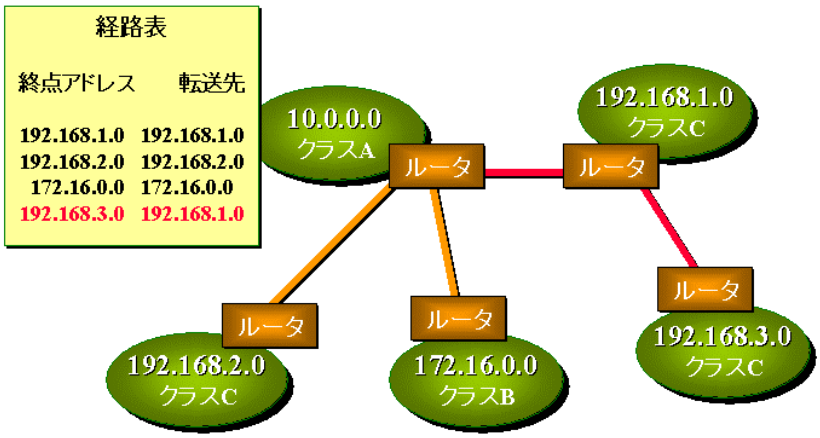

図 14　IP データグラムの転送経路は？

ネットワークはルータによって接続され、それぞれのルータは「経路表」というものを持っています。この経路表は非常に重要なものですが、この中には最終的な目的地はどこにあるのか、そして、それに対応する転送先のネットワークがどこであるのかといった対応一覧が入っています。図の中では“10.0.0.0”というネットワークのルータが持っている経路表を示していますが、ポイントは隣接していないネットワークの扱いです。

隣接したネットワークであれば直接指定できますが、たとえば“192.168.3.0”というネットワークに送る場合は“192.168.1.0”というネットワークを経由しなければいけません。そうしたとき、「“192.168.3.0”というネットワークに送る場合は“192.168.1.0”というネットワークに送りなさい」という情報が追加されることになります。

3.3.1　経路制御

このようにルータが経路表を持つことでネットワーク同士を接続することができるようになるわけですが、「IP データグラムを次にどのルータに投げればよいか」ということを「経路制御」といいます。経路制御は非常に複雑な話になりますからここでは簡単に済ませます。

各ルータは、最終宛先ネットワークと、それに対応する中継ルータの一覧表を保持します。その一覧表のことを「経路表」といい、経路表はネットワークが増えれば

経路表のエントリ数も増えることになります。このことを頭に置いてください。

IP データグラムを最終宛先ネットワークに送り届けることが経路制御の目的ですが、直接つながっていない場合には IP データグラムはルータによりネットワークを転々とすることになります。

3.3.2 IP データグラムの細分化

ここでひとつ考えなければいけないのは、ネットワークには、その種類によって扱える「最大フレームサイズ（MTU：Maximum Transmission Unit）」というものが存在するということです。たとえば、イーサネットであれば 1,500 オクテット（1 オクテットは 8 ビット）であり、FDDI では 4,470 オクテットと決められています。

仮に、送ろうとする IP データグラムがこの最大フレームサイズより大きかったらどうなるのでしょうか。ネットワークの制限として、そのネットワークで決められている以上の大きさの単位では送れませんから、フレームサイズに収まるように分割することになります。

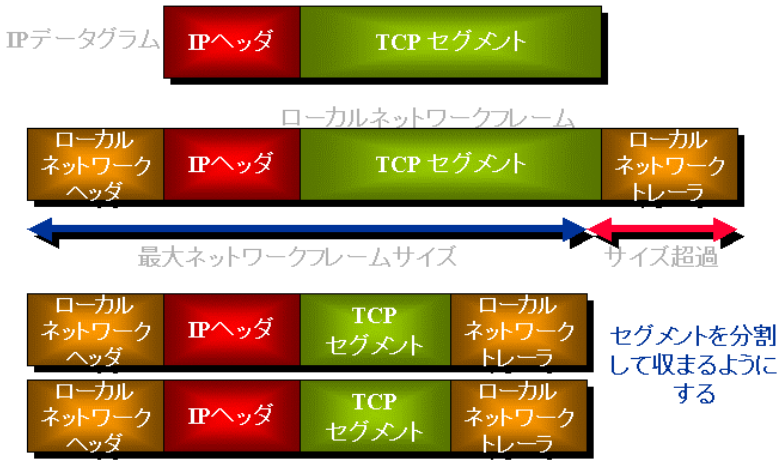

図 15 IP データグラムの細分化

これを「IP データグラムの細分化」または「フラグメンテーション」といいますが、これは、IP データグラムがネットワークインタフェース層で使われる「ローカルネットワークフレーム」に作り変えられるときに、最大フレームサイズに収まるような形で処理を行うということです。この際に分割できる部分はローカルネットワークフレームにとってのデータ部分となる TCP セグメントしかありませんから、処理としてはサイズが収まるまでどんどん小さくし、結果として分割されたそれぞれの TCP セグメントに IP ヘッダを付けてネットワークに渡します。これをもう少し具体的に表わしたのが次の図です。

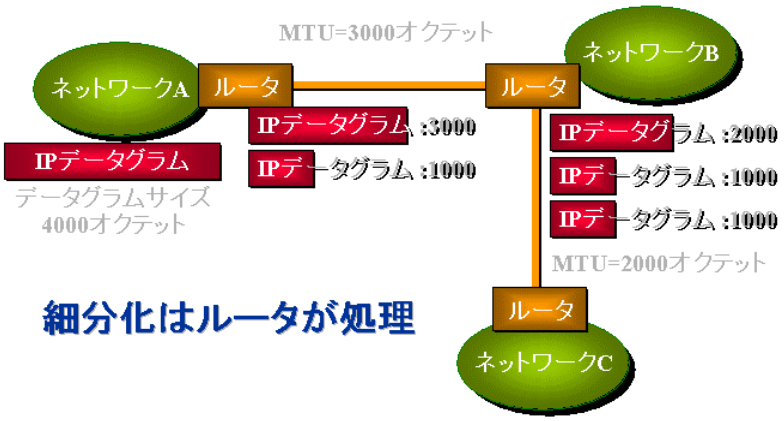

図 16　細分化の例

ネットワーク A からネットワーク C まで 4,000 オクテットの IP データグラムを送るとき、ネットワーク A とネットワーク B を結ぶ回線が持つ最大フレームサイズの 3,000 で分割して送出したとします。この、セグメントを分割して最大フレームサイズ内に収まるようにする作業を行うのはルータです。さて、次にネットワーク B で受け取った IP データグラムをネットワーク C に送るためにはその間にある回線の最大フレームサイズが 2,000 ですので 3,000 の大きさを持つ IP データグラムの方をさらに分割しなければいけません。元はひとつの IP データグラムでしたが、このような経過によって最終的には 3 つの IP データグラムになります。

IP にはこのようにして分割されたデータグラムを統合する機能がありますから、分割されても問題はありませんが、この細分化はルータによって行われるために処理が多くなるとルータにとって重荷となります。

3.4　IPv4 が抱える問題

まず、経路制御という面から見てみましょう。これまでの説明で示したとおり、ネットワークの数が増えると経路情報が増えます。経路情報が増えると情報を持つためのメモリ容量や宛先ネットワークを検索する処理能力も必要になるので、ルータの負荷が増大します。

ルータは機械ですから機能や処理能力に限界があります。そのためには経路情報をできるかぎり抑制する必要があります。では、経路情報を抑制するためにはどうすればよいのでしょう。

当初のアイディアは、「ひとつのネットワークにひとつのネットワークアドレスを割り当てる」というものでした。この考えに従うと、たとえば 1,000 個のホストを持つネットワークがあったとすると、クラス C（ホスト数 256）では足らず、クラス B（ホスト数約 65,000）では多すぎるということになります。しかし、クラス C を 4 つ渡すと経路情報

も４つになってしまいますから、経路情報を抑制する目的のためにはクラスＢを渡すしかありません。ですが、インターネットの成長によって今度はクラス B が足らなくなるという事態に陥りました。

ここで危機感が起こり、クラス B が無くなってしまうという問題を回避するために仕方なくクラスＣを複数個渡すということになりました。当然の結果として経路情報が急増してしまったのです。

3.4.1 CIDR

こうした事態に陥る諸悪の根源は、クラスという概念を表わすためのルールが 8 ビット単位でしか定義できないという点にあります。そのために無駄が多く、アドレスを有効に利用できません。そこで、この「クラス」という概念を無くしてしまおうということになり、「CIDR」(Classless Inter-Domain Routing:サイダー)という技術が生み出されました。

この基本的な考え方は、ネットワーク部を１ビットごとにどこでも区切れるようにするためにクラスという概念を廃止し、先頭からどこまでがネットワーク部かを示せるようにするというものです。これに伴い、IP アドレスの先頭ビットでクラスを識別するという手法は廃止されました。ここから、クラスという概念を使った「クラスフル」な時代からクラスの無い「クラスレス」な時代に移っていきます。

たとえば次の図のようなアドレスを見てみましょう。

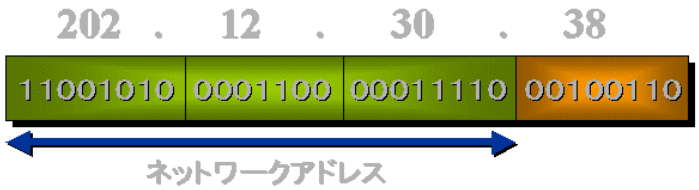

図 17　従来のクラス C アドレス

これは、従来のクラスフルな考え方では先頭ビットが“110”ですからクラス C であり、ネットワークアドレスは 24 ビットでホスト数は 256 までということになります。これが、CIDR では次のようになります。

図 18　CIDR 表記

アドレスの後ろに付いている“/22”は、先頭から何ビット目までがネットワーク部かを示す数字です。この例では 22 となっていますから、先頭から 22 ビット目までがネットワークアドレスであるということになります。そして残る 10 ビットがホスト部となりますから 1024 のホストを定義することができます。

CIDR を使うことによって、ネットワークの規模に応じた適切な大きさのアドレス空間を割り当てることができるようになりました。また、そうしてできた空間はひとつのネットワークとしてのアドレスブロックになりますから経路情報もひとつで済みます。また、うまく割り振りを行うことで、さらに経路情報を減らすこともできます。次からは、その話に移りましょう。

3.4.2 経路情報の集約

たとえば次の図のようなネットワークがあったとします。それぞれのネットワーク部は 23 ビットもしくは 22 ビットで、それを束ねるネットワークがひとつ存在します。

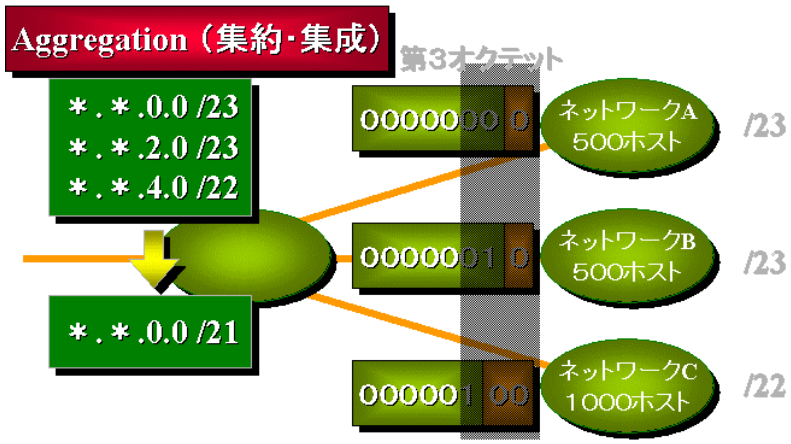

図 19 経路情報の集約

図で示されている第 3 オクテットの緑の部分が実際のネットワーク部です。このとき、第 3 オクテットの先頭から 5 ビット目までは共通ですから、このネットワーク A とネットワーク B、そしてネットワーク C はネットワーク部が 21 ビットのネットワークとして集約できることになります。

これを郵便に例えれば、別の市からある市の中にある各団地に手紙を配達するときにそれぞれ個別に経路を設定するのではなく、いったん市内の郵便局に一括して転送し、そこから仕分けして送ることに似ています。

つまり、ネットワーク A とネットワーク B、そしてネットワーク C のいずれに送る場合でも、とりあえず経路的にはそれぞれのネットワークを束ねているネットワークまで送り届ければいいわけですから、それを行うことで外部から見た経路情報を集約することができるということです。

このように、複数の経路情報をひとつにまとめることを「集約」とか「aggregation」といいます。

3.4.3 32 ビットのアドレス

CIDR の導入によってアドレスを割り当てるときの無駄は減少し、経路情報もうまく集約することができました。しかし、それでもインターネットの発展に対して 32 ビットのアドレスは小さすぎるということが言われ続けています。

32 ビットで表わすことができる数字は約 40 億です。そのため、世界の人口を 60 億人としても全員には行き渡りません。もちろん、一人にひとつずつアドレスを割り当てることは現実的ではないにしても、たとえば携帯電話や家庭用ゲーム機、インターネットの常時接続の増加といったことを考えると、今後ますます IP アドレスに対する要求は大きくなると考えられます。

ですから、アドレスの無駄はできるだけ出さないとか、アドレスを節約しようということになります。そこで、世界で一意となる IP アドレスが必要なのはインターネットと直接接続するときですから、インターネットに直接接続することのないホストには不要だという考え方がされるようになります。これには、社内ネットワークのみに接続している場合とか、プロキシサーバを介してインターネットにアクセスしているホストとかが該当します。

そういった考え方の基に、IP アドレスには「グローバルアドレス」と「プライベートアドレス」といった分け方があります。

3.4.4 グローバルアドレスとプライベートアドレス

インターネットで使われる IP アドレスは、ホストの特定のためにインターネット上で一意に割り当てられる必要があります。このためのアドレスを「グローバルアドレス」といいます。

一方で、閉じた環境の中で使うことを条件に、自由に使ってよいアドレスとして「プライベートアドレス」というものが存在します。

プライベートアドレスは、インターネットに直接接続して使うことはできません。その代わり、異なるネットワークで同じアドレスを用いることができます。これはグローバルアドレスと同様に 32 ビットのアドレスですが、どれがプライベートアドレスになるかということが RFC1918 で決められています。具体的には、以下のアドレスの範囲です。

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)

ところで、プライベートアドレスを使っているホストは直接インターネットに接続することはできませんが、間接的に行うことは可能です。それを実現するものに、プロキシサーバや NAT(RFC1631:Network Address Translator)という技術があります。

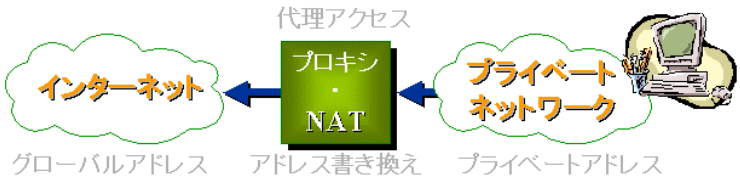

図 20 プライベートアドレスの活用

この仕組みを簡単に説明すると、プロキシサーバはアプリケーションレベルで実装され、プライベートからのリクエストをプロキシサーバが代理アクセスし、得られた結果を要求元に返すことで間接的にインターネットとやりとりができるようにしています。一方で、NAT はネットワークレベルで実装され、プライベートアドレスとグローバルアドレスを一対一に対応させて、その対応表を使って相互変換を行うことでインターネットに接続するというものです。

こうした技術の普及によってプライベートアドレスは企業内ネットワークなどで一般的に使われるようになり、結果として、部分的ではあってもアドレスの節約をすることができました。

3.4.5 インターネットの一般化と要求

インターネットが発展して一般化したことにより、新しい適用分野や、新しい機能要求が出てきました。たとえば、プラグ＆プレイ、モバイルコンピューティング、セキュリティの確保、リアルタイム通信といったさまざまな要求があります。

IP としては、これらの要求を満たすためにさまざまな技術開発が行われてきました。プラグ＆プレイとしては、DHCP というアドレスなどの自動割当を行う仕組みがあり、モバイルコンピューティングでは Mobile IP、セキュリティに対しては IPsec、リアルタイム通信には Diffserv とか RSVP といった技術が用意されています。

3.4.6 IPv4 の問題のおさらい

このように、いままでいろいろな技術開発が行われてきましたが、ここでもう一度 IPv4 が抱える問題を振り返ります。主なものとしては、

- アドレスの不足
- 経路制御情報の増大
- 細分化、ヘッダ処理によるルータ負荷
- サービス要求に対する機能拡張

がありました。しかし、それらのために提示されたさまざまな方法で IPv4 が抱える問題が必ずしも解決したわけではありません。

CIDR は有効に機能し、アドレスの節約とか経路情報の集約にはある程度成功しましたが、根本的なアドレス不足には対処できません。プライベートアドレスや、NAT とかプロキシサーバを有効に使うことによってアドレスはかなり節約されています。しかし、NAT やプロキシサーバを利用できないアプリケーションも多く存在します。また、IP の細分化やヘッダ処理は依然として存在し、拡張された機能も IP という基盤技術の上に後付けされた機能であるために互換性や普及という点で問題があります。

では、そういった問題があるのなら、もっと根本的な部分から見直しをかけようということになり、IP というプロトコルの見直しから始めることになりました。ということで、IPv6 という技術が開発されていくわけです。

4 次なるプロトコルへ —— IPv6

いままで IPv4 の問題点をあげてきましたが、IPv4 が劣った技術だったわけではありません。IPv4 は 20 年以上にわたってインターネットを支えてきた基盤技術であり、また基本が優れていたからこそ 20 年以上にわたって使われ続けてきた技術であるわけです。

しかし、いままでお話してきたとおり、さまざまな問題が出てきていることも事実であり、いつまでもしがみついているわけにはいきません。まさに、危機はいまそこにあるのです。したがって、IPv4 には感謝しつつ、根本的な解決を目指すためには次なるプロトコルである IPv6 に向かう必要があります。

4.1 IPv6 への道

当初、この IPv6 は「IP Next Generation」という名前で開発が進められていました。

4.1.1 IP Next Generation

IP Next Generation の発端は次のようなものでした。まず、1991 年 7 月に IPv4 のアドレスが足りなくなるという研究結果を受けて IETF（Internet Engineering Task Force）が調査を始めます。翌年の 1992 年 11 月、IETF の調査結果としてアドレスの先行き調査結果が RFC1380 として公開されました。そして、その RFC をもとに次世代のインターネットプロトコルの検討が開始されました。

そこでは、次世代のインターネットプロトコルはどういう機能を持っていなければならないかということが検討され、1993 年の 12 月に RFC1550 として IPng（ng は Next Generation の略）に対する機能的な要求がまとめられました。

ここで、その機能的要求のいくつかを取り上げてみましょう。

まず、たくさんのアドレスが持てること。次に、経路情報を集約できること。ヘッダの構成が単純であること。ルータの負荷の原因となっている細分化が防止できること。プラグ＆プレイによる簡単な設定ができること。セキュリティやリアルタイム性の確保といったことがプロトコルのレベルで標準として実装されていること。そして最も重要視される部分として、現在使われている IPv4 から単純で柔軟に移行できることがあります。

4.1.2 IPv6 の誕生

前出の機能的要求を受けていくつもの提案がなされましたが、IETF が最終的に選択したのは「CATNIP」、「SIPP」、「TUBA」の 3 つです。これらは

- CATNIP は、上位層のプロトコルとして、TCP や OSI、Novell といったようなさまざまなプロトコルを IP レベルで共通的に利用できるようにしたもの。しかし、壮大な理想ではあるけれども仕様が固まっていない。
- SIPP は、IPv4 を拡張し、問題のひとつであったアドレス空間を 64 ビットにすることで解決しようとしたもの。アドレス空間は拡大されたが、64 ビットではまだ足りない。
- TUBA は、IP を大きく作り変えてしまい、アドレスの長さも可変長にしてしまうといった、かなり野心的なもの。しかし、重要な要求のひとつである IPv4 との整合性に問題がありそう。

というようにそれぞれ問題を抱えていました。結果的には 1995 年 1 月に SIPP をベースにしてアドレスを 128 ビット化したものが RFC1752 として採用されました。そして、この中で初めて「IPng」から「IPv6」（IP Version 6）へと正式に改名されています。

そして、1995 年 12 月に RFC1884「IP Version 6 Addressing Architecture」として IPv6 のアドレス構造が決まりました。のちに、これは RFC2373 として改定されています。

その後、1998 年末に IPv6 関係の RFC に大改定が加わりました。現在 IPv6 として用いられるのはこのときのもので、RFC2460「Internet Protocol, Version 6 (IPv6) Specification」から始まる一連の IPv6 関係の RFC です。

4.2 機能的要求

では、IPv6 の特長をひとつずつ見ていきましょう。

4.2.1 たくさんのアドレス

アドレスがたくさんあることは当初の重要な目的のひとつでした。IPv6 ではアドレスの長さが 128 ビットもありますから、そこで示せる数は 3.4 に 10 の 38 乗を掛けた数ということになります。34 という数字のあとにゼロが 37 個並びますが、その数を容易に想像することはできません。仮にこの数を地球上の陸地にばら撒いたとしても、1 平方センチメートルあたり 2.2×10^20 もの数になります。

4.2.2 IPv6 のアドレス表記

数が多いということははっきりしましたが、一方で 128 ビットという長いアドレスをどのように表記するかという問題が起こります。IPv4 のときは 8 ビットずつピリオドで区切って書くということになっていましたが、これに従うと“123.123.123.123.123.123.123.123.123.123.123.123.123.123.123.123”というように非常に長くなってしまいます。

いずれにせよ、128 ビットの長さのものを記述しなければいけませんから、少しでも短くするために IPv6 では 16 進数で書き、区切り文字に“:”(コロン)を使います。したがって、“FFDC:BA98:7654:3210:FEDC:BA98:7654:3210”というように 4 文字のフィールドが 8 個並ぶことになります。

それでもまだ長いわけですが、IPv4 と異なるのは省略表記ができるようになっていることです。ただし、ゼロが連続していて、かつ 1 か所だけというように省略できるパターンは限られています。具体的には、“1080:0:0:0:8:800:200C:417A”というような場合には“1080::8:800:200C:417A”というように表記できます。

4.2.3 IPv6 アドレスの種類

IPv4 のときにはグローバルアドレスとプライベートアドレスという分け方がありました

が、IPv6 にもいくつかの種類があります。

図 21　アドレス形式プリフィックス

IPv4 のときにはアドレス範囲のどこからどこまでがプライベートアドレスだという決め方でしたが、IPv6 では先頭にあるアドレス形式プリフィクスというものを設けることによって自動的に判別できるようになっています。

先頭にある「アドレス形式プリフィクス」が“001”の場合は「集約可能なグローバルユニキャストアドレス」になり、これは、IPv4 におけるグローバルアドレスにあたります。

また、“1111 1110 10”の場合には「リンクローカルユニキャストアドレス」になり、“1111 1110 11”の場合には「サイトローカルユニキャストアドレス」になります。これは、IPv4 におけるプライベートアドレスにあたります。

そして、先頭の 8 ビットがすべて 1 の場合は「マルチキャストアドレス」となります。

アドレス形式プリフィクスは、この他にもさまざまに定義されていますが、定義されていない空間も数多くあります。そのような空間は予約領域として現在は使われていません。将来的に集約可能なグローバルユニキャストアドレスが足りなくなった場合にそこから割り当てるとか、そういった使い方になると考えられます。

4.2.4　経路情報の集約

次に、128 ビットのアドレスを用いたうえで経路が集約できる必要があります。IPv4 のときを振り返ってみると、経路情報を集約するためには

- アドレスがクラスレスであること
- ネットワークの構造に応じた割り振りをする

ということがありました。前者はネットワーク部をうまく使うためで、後者は経路情報をまとめるためでした。具体的には、同じネットワークには連続したアドレスブロックを割り振るとか、上位において下位のアドレス経路情報をまとめることで経路情報

を集約します。

4.2.5 アドレス構造

IPv6 では IPv4 におけるこのような議論や経験を元に最初から集約可能(Aggregatable)なアドレス構造になっています。

集約可能なグローバルユニキャストアドレスはネットワークトポロジに応じた階層構造を持っています。

図 22 集約可能なグローバルユニキャストアドレス

3 ビットの「FP」は「アドレス形式プリフィックス」で、集約可能なグローバルユニキャストアドレスを表わすときにはここは“001”になります。続く「TLA ID」は 13 ビットの長さを持つ「最上位階層集約子」です(階層集約子は、下位のネットワーク情報をまとめるために使います)。8 ビットの予約領域を挟んで続く 24 ビットの「NLA ID」は「次階層集約子」であり、16 ビットの「SLA ID」は「サイト階層集約子」です。最後の 64 ビットの長さを持つ「インタフェース ID」は、ネットワークインタフェースが持つ固有のアドレスです。

IPv6 では、128 ビットのうちの先頭から SLA ID までの 64 ビットがネットワークアドレスになります。IPv4 のように、クラスだとかネットワーク部がどこまでだといった可変の形にはなっていません。IPv6 では 64 ビットという固定の長さでネットワークアドレスが書かれます。

4.2.6 アドレスの割り振り

では、この集約可能なグローバルユニキャストアドレスが実際にどのように使われるのかを確認します。

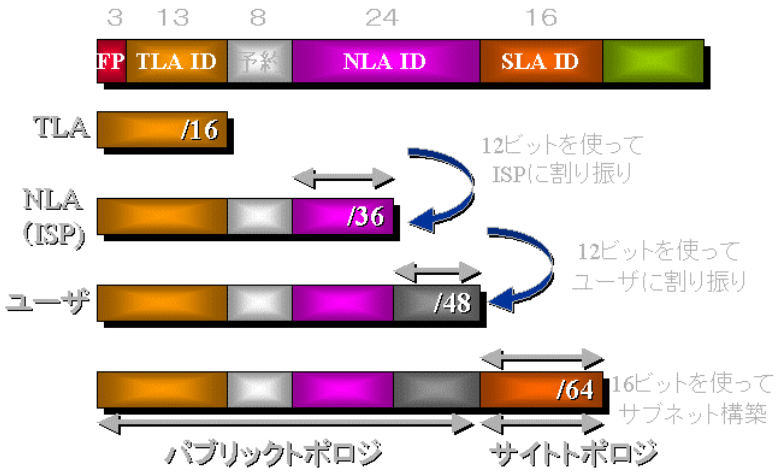

図 23　アドレス割り振りの例

まず、TLA　の割り振りが行われます。割り振りは階層的に上位から順に大きな単位を割り振るということになっていますから、この最初の 16 ビットがインターネットの基幹を構成する組織に割り振られます。

TLA を割り振られた組織は、次の NLA ID の部分を下位の NLA 組織（一般には、ここが ISP などになります）に割り振ります。この NLA　ID の使い方は各 TLA 組織の裁量に任されることになりますが、この図では前半分の 12 ビットを ISP に割り振って、ISP は残りの 12 ビットを使ってさらにユーザーに割り振りを行っています。

このようにして、順次、アドレスが下位組織に割り振られていきますが、IPv6 ではユーザーに割り振ることのできる最小単位は/48 と決められています。/49 とか/50 という割り振りを行うことはできません。

IPv6 では、SLA　ID 以上の 48 ビットをインターネットのグローバルな世界で用いられるネットワークアドレスとして「パブリックトポロジ」と呼び、さらにそれより小さい SLA ID までを用いた/64 を「サイトトポロジ」と呼びます。

ユーザーは、この/48 を受け取ると、残っている SLA　ID の 16 ビットを用いて自組織内のサブネットを構築することができます。

このようにして、上位階層で下位のネットワーク情報をまとめていくことで経路情報を集約できるということ、16　ビットを用いて組織内のサブネットを構築でき、かつ最小割り振り単位が/48　であるといった広大なアドレス空間の割り振りを受けることができるということが IPv6 の特長になっています。

4.2.7 IPv4 のヘッダ

IPv4 で問題となっていたルータの負荷を軽減するために要求されたことは、

- 単純な構成のヘッダを持っていること
- 経路の途中での細分化が防止できること

の 2 点でした。まず、IPv4 のヘッダ構成を確認します。

図 24 IPv4 のヘッダ

先頭の「Ver」はバージョンを示します(この場合では IPv4)。次にある「IHL」は“Internet Header Length”の略で、ヘッダ自身の長さを示します。なぜこのようなものがあるかというと、最後の方に「オプション」という可変長のオプションフィールドがあるためです。IP データグラムの先頭には IP ヘッダが必ず付きますから、そのヘッダが可変長であるということはデータグラムごとに長さの判断のための処理を強要しますのでルータにとってとても大きな負荷となります。ですから、可変長をやめ、IHL 自体を無くしてしまうことを考えます。

「TOS」は“Type of Service”の略で、このIPデータグラムが運んでいる情報の種類を定義するために作られたフィールドです。しかし、実際にはほとんどのアプリケーションが自分の情報は重要で最優先だとしてしまうため、ほとんど意味のないものとなっています。ただし、リアルタイム性が必要な場合とかを宣言したい場合とかも考えられるため、機能としては必要なものです。

「データグラム長」は、どのくらいの大きさのデータグラムかを示します。

その下にある「ID」、「フラグ」、「オフセット」の 3 つは細分化のために用意されているフィールドです。データグラムの細分化が行われたとき、元となったデータを特定するためにこの ID の部分には同じ値が入ります。フラグは、どのようなポリシーでそのデータグラムを細分化してよいかといったことを書き込みます。オフセットは、元のデータグラムのどこに位置していたのかを示します。もし細分化が必要なければ、これらのフィールドは必要ありません。

「TTL」は“Time to Live”の略で、あと何回ルータを通過できるかが示されます。

「プロトコル」は、その IP データグラムが運んでいるデータに含まれる上位のプロトコルが何であったかを示します。たとえば TCP とか UDP であるということですが、IP というプロトコルとして見れば始点から終点まで無事に送り届ければいいわけで、上位のプロトコルが何であるかを知っている必要はありません。したがって、不要ということになります。

「ヘッダチェックサム」は、このヘッダ自体のチェクサムを記憶するようになっています。送信途中で細分化が行われたり TTL の値が変わったりするたびに再計算されますが、本来データの正しさは IP レベルで確保されるものではありませんし、保証するものでもありません。問題が起これば捨てるしかなく、元に戻せるものではありませんから IP レベルでは必要ないということになります。

4.2.8 IPv6 ヘッダ

IPv4 のヘッダに対して行った検討結果を IPv6 のヘッダに適用すると、本当に必要な情報はそれほど多くないということになります。

図 25 IPv6 のヘッダ

IPv6 のヘッダは IPv4 に比べて非常に単純です。「Ver」はバージョンを示します。次に、TOS を拡張した「クラス」と「フローラベル」があり、この IP データグラムの優先度を設定する目的で使われます。そして、「ペイロード長」は運んでいるデータの長さを表わします。

IPv6 の特徴的な部分として「次ヘッダ」がありますが、これは IPv4 にあったようなオプション機能を拡張ヘッダとして IPv6 ヘッダの後ろに続けられるようにするものです。したがって、ここには拡張ヘッダが続く場合にその種類を示す識別子が書き込まれることになります。

続く「中継制限」は、IPv4 でいうところの TTL と同様です。そして、ヘッダとして重要な「発信元 IP アドレス」と「送信元 IP アドレス」があります。

4.2.9 拡張ヘッダ

IPv6 のヘッダは、ルータの処理を単純化するために基本情報のみとしてサイズを固定しています。そのため、IPv4 にあったオプションフィールドのような拡張機能は独立したヘッダとして IPv6 ヘッダの後ろに連結することになります。

図 26 拡張ヘッダ

拡張ヘッダは、それ自身の中にも次ヘッダフィールドを持ちますから、それによって拡張ヘッダをいくつもつなげることができます。また、拡張ヘッダはオプショナルですから付けなくてもかまいませんし、新しいタイプを定義していくことで柔軟な拡張を実現することができます。

4.2.10 経路中での細分化禁止

IPv6 では、ルータ負荷の原因となっている経路中での細分化を禁止しています。細分化が起きるのはネットワークごとに MTU（最大フレームサイズ）というものが存在するからですが、そのための対策として IPv6 には経路上の最小 MTU を探索する機能が用意されています。

つまり、始点から終点までで経由する全てのネットワークの MTU を調べて、その最小値を送出する IP データグラムの大きさとします。

4.2.11 プラグ＆プレイ

IPv6 では、プラグ＆プレイは非常に簡単です。ネットワークインタフェースをネットワークに接続すると、ルータが設定に必要な情報を教えてくれます。

図 27 プラグ＆プレイでのアドレス設定

ネットワークインタフェースは、受け取ったルータからのネットワーク情報（64 ビット）と自分自身が持っているインタフェース ID（64 ビット）をつなぎ合わせて IP アドレスを生成します。

なぜこんなに単純にいくかというと、ネットワークインタフェースにはもともと世界で一意となる番号が割り振られているからです。イーサネットの例でいえば MAC アド

レスが該当しますが、これを利用すればインターネット上で一意なアドレスを簡単に作ることができます。また、インタフェース ID だけで 64 ビットの長さがありますから、個々のインタフェースを世界的に一意にする場合でも十分な数が用意できていることになります。

大事なことは、この方法を使うとネットワーク ID で経路を明確にでき、インタフェース ID で機器を特定できるということです。また、DHCP でアドレスをもらう場合には毎回異なるアドレスが割り当てられる可能性がありますが、この方法だと同じネットワークなら固定化できるというメリットもあります。

4.2.12 セキュリティやリアルタイム性の確保

セキュリティは、いまや重大な関心事です。また、オーディオ配信やビデオ配信、オンラインシステムなどの場合には応答性と通信速度の確保が重要になります。こうしたことをどのように実現するかを簡単に説明します。

IP における「セキュリティ」は、通信するインタフェース間における認証と機密性を保持することです。たとえば、通信する相手が、自分が意図している正しい相手であるかどうかの確認ができます。これを「認証」と言いますが、これによって「なりすまし」のようなものが防げます。また、通信内容を経路の途中で覗き見されないようにするために暗号化の機能が提供されます。

これらは IPv4 で用いられた IPsec という機能を IPv6 の拡張ヘッダを用いて実装し、標準の機能として提供されています。

通信の優先度に関しては、リアルタイム通信が必要なものには高い優先度を設定することになりますが、利用方法などはまだ研究段階にあります。

4.3 IPv4 から IPv6 への移行

IPv4 から IPv6 へ移行する場合、何が必要かということを考えてみます。そのためには、

- この変更によって必要となるサービス的な変更は何か
- 変更のタイミングをどのように設定すればよいのか
- いま使っている IPv4 のアドレスは使えなくなってしまうのか

といったようにさまざまなことを考える必要があります。

4.3.1 IP の変更による影響範囲

まず、IP の変更がどこに影響を与えるかを確認しましょう。

図 28 IP の変更による影響

IP はインターネット層に位置しています。たとえば、身近なコンピュータを例にとれば、このインターネット層はオペレーティングシステムが管理しています。その下にはネットワークインタフェースカードがあったり、その上ではアプリケーションが動いていたりします。

ルータでは、IP はルータの本来の機能である経路制御部分にあたります。その上では経路制御部分を制御するための OS が動いていたり、さらに上ではアプリケーションが動いていたりします。

IP に変更が加わることで、必然的に OS やルータの経路制御部に影響が及びます。しかし、それだけではなく、その上全部に影響があります。直接的に IP にかかわっていなくても IP を用いている全ての部分に大小の差こそあれ何らかの変更が必要です。

アプリケーションから見ると直接 IP を使うわけではありませんが、たとえば

- 直接アドレスを入力することもある Telnet や FTP などでは IPv6 のアドレス表記を理解できなくてはならない
- TCP/IP とのアプリケーションインタフェース部分で IPv6 が理解できなくてはいけない

といったことがあります。また、オペレーティングシステムでは

- IPv6 のアドレス表記を理解できなくてはならない
- IPv6 固有の機能（セキュリティのように標準実装となったものなど）というものを実装しなければいけない

- ネットワークインタフェースとやり取りをするドライバの部分で IPv6 に対応しなければならない

といったことがあります。ルータでは、

- 自分が管理する経路情報においてIPv6の経路が管理できなければならない

といったことがあります。

4.3.2 DNS

インターネットにおける基幹サービスは DNS です。DNS はドメイン名と IP アドレスの変換を行いますが、その変換には 2 つのパターンがあります。

ドメイン名からIPアドレスを検索するための仕組みが正引きです。このためにDNSはドメイン名と IP アドレスの対応表を持っていますが、この IP アドレスの部分にIPv6の情報を登録できるようにする必要があります。従来はIPアドレスを記述するために A レコードというものがありましたが、IPv6 のために“AAAA”というレコードが用意されています。

IP アドレスからドメイン名を検索するための仕組みが逆引きです。しかし、IPv4 とは違うアドレス体系ですから、IPv6 のためのアドレス空間を新しく作成する必要があります。

ちなみに、DNSの実装であるBINDは、現在のバージョン8系列ではIPv6への対応がほぼ終了しています。したがって、最新の BIND であれば、設定さえ行えばIPv6の管理が行えるようになっています。

4.3.3 移行の流れ

IPv6はさまざまなメリットを与えてくれますが、だからといってあるタイミングでインターネットの全てのネットワークを一度に置き換えることはできません。したがって、少しずつでも確実に移行していくことを考えます。そのためには、

- インターネットはネットワークの集合であるから、移行する単位はネットワークごとにする
- それぞれのネットワークの中でもIPv4とIPv6を少しずつ置き換えていく
- 少しずつ切り替えていくことで、いずれは世の中の全てが IPv6 になるだろう

といった移行プランが必要です。では、ネットワークの中でどのように移行していく

かから見ていきましょう。

図 29　デュアル IP スタック

ホスト単位での移行では「デュアル IP スタック」という実装を使います。これは、インターネットプロトコル層に IPv4 と IPv6 という 2 つのプロトコルを同居させることで、IPv4 と IPv6 の両方を扱うことのできるホストやルータを実現するものです。もちろん、IPv4 が使えますから従来のネットワークの中で使うことができます。このようにして IPv6 を扱うことのできる機器を増やしていき、すべてが IPv6 を動かすことにできるようになった時点でそのネットワークは完全に IPv6 に移行したことになります。

そして、ネットワーク単位で IPv6 への移行ができたとします。しかし、世の中の主流はまだ IPv4 ですから、生まれたての IPv6 ネットワーク同士が通信を行うためにはちょっとした工夫が必要です。そのために用意されるのが「IPv6 over IPv4 トンネリング」という技術です。

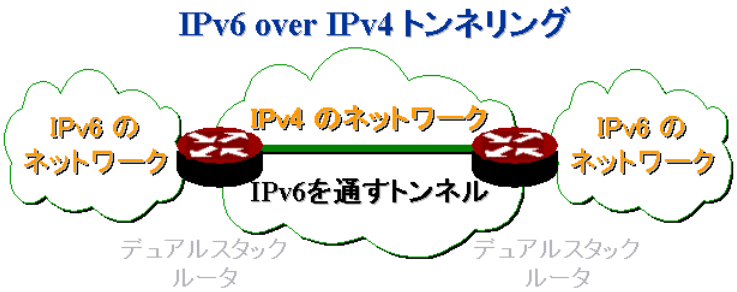

図 30　IPv6 over IPv4 トンネリング

IPv4 と IPv6 は違うプロトコルですから、IPv4 と IPv6 のネットワークを直接つなぐことはできません。このために、IPv4 のネットワークに IPv6 をつなぐパイプを作ります。このような仕組みを用意することで、離れた IPv6 のネットワーク同士を接続します。

この「トンネリング」という技術は、異なるプロトコルパケットを通信経路のプロトコルでカプセル化して相手のネットワークまで転送するために用いられます。

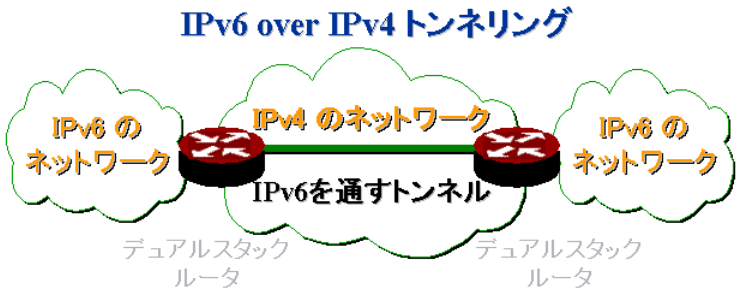

図 31 トンネリング

このケースでは、IPv6 の IP データグラムをデュアルスタックルータのところで IPv4 の IP データグラムとしてカプセル化し、IPv4 のネットワークに投げる形になります。IPv4 データグラムの中には 2 種類の IP ヘッダが見えますが、IPv4 のネットワークから見れば IPv4 ヘッダの後ろは基本的に IPv4 データとしか見えませんから問題にはなりません。もちろん、このデータグラムの届け先は相手のネットワークにあるデュアルスタックルータとなり、受け取った IPv4 データグラムから IPv6 の IP データグラムを取り出すことになります。

4.3.4 移行のストーリー

では、具体的にインターネットにおいてどのように移行していくのかといったストーリーを見ていきます。

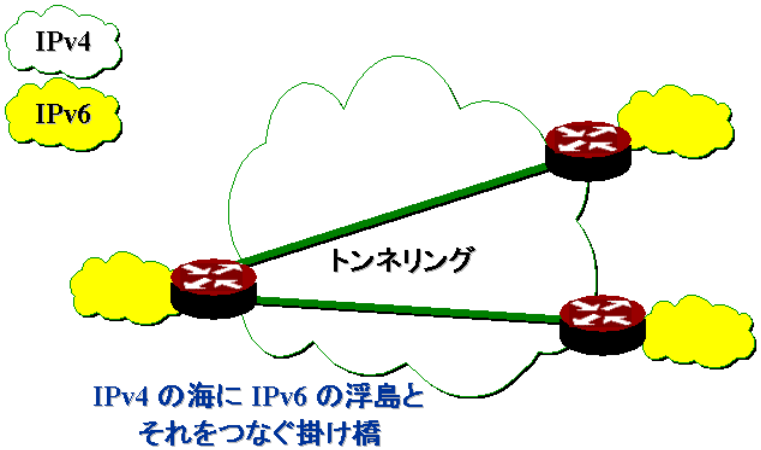

図 32 IPv6 over IPv4 トンネリング

生まれたての IPv6 ネットワークはインターネット上に点在する形になり、中央には IPv4 のネットワークが存在しています。この時点では、IPv6 over IPv4 トンネリングの技術を使って点在する IPv6 同士をつなげることになります。

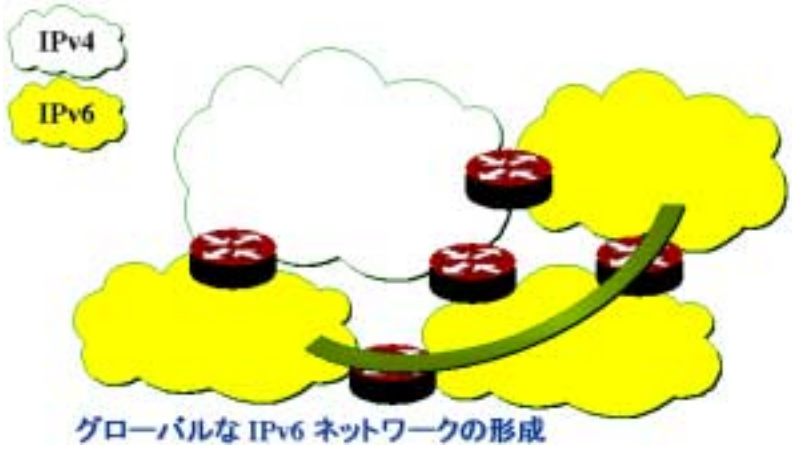

図 33　IPv6 としての通信が可能になる

IPv6 のネットワークが増えてくると、IPv6 のネットワーク同士が接続されて IPv4 のネットワークを通さなくてもよくなります。こうなるとトンネリングは不要になり、IPv6 だけを用いてグローバルな通信が可能になります。

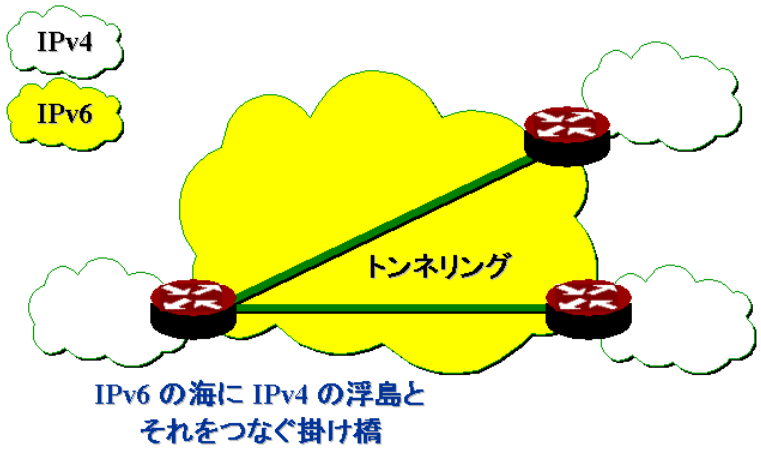

図 34 残った IPv4 のネットワーク同士がトンネリングによって結ばれる

IPv6 が主流となった時点で立場は逆転し、今度は残った IPv4 のネットワーク同士がトンネリングによって結ばれることになるでしょう。

4.3.5　IPv4 の扱い

ここでの心配は、「IPv4　は使えなくなってしまうのか」というものです。その回答としては、「IPv4 は時間とともに IPv6 に置き換わっていくと考えられるが、まったく使えなくなるわけではない」というものです。

もちろん、その理由としては、技術を選択するのはユーザーであり、誰かが「こうする」と言ったからそうなるものではないということと、移行のスピードの予測ができないということがあるからです。

また、仮にインターネットの基幹が全てIPv6になったとしても、トンネリングといった技術を使うことによってIPv4を使いつづけることができるという部分も実際にはあります。

とはいえ、IPv6 への移行はいずれ必ず行われるものですから、それに対する備えは十分にしていくべきだと考えます。そのためには、たとえばIPv4に特化しないサービスおよびネットワーク運用というものを視野に入れておく必要があります。

5 IPv6 の今

ここでは、IPv6 の現状について説明します。

まず、インターネットプロトコルとしての基本的な部分は昨年末の大改定によって基本的な部分はほぼ固まりました。ただし、複雑な経路制御に関する部分やセキュリティに関する部分、またデータグラム転送の優先順位に関する部分といった拡張機能に関してはまだ検討中のものが多く残っています。

5.1 ハード・ソフトの IPv6 対応

では、そういう仕様に対してハードウェア、ソフトウェアがどういう状況にあるかを紹介します。

5.1.1 OS（オペレーティングシステム）

世界中で最もユーザー数が多いと思われる Windows 系を見てみると、

Windows 2000 は IPv6 に対応しています
Windows NT ではマイクロソフトがテスト実装を WEB 上で公開しています
Windows 98/95 では日立製作所がテスト実装を公開しています

となっています。

また、インターネットの世界で主流の OS として使われている UNIX では、日本の WIDE プロジェクトが行っている「KAME Project」による実装が世界における事実上の標準として使われています。

5.1.2 ルータ

ルータに関しては、現在のインターネットの基幹を構成しているものにはすでに

IPv6 に対応した製品も多く、今後もサポート対象は広がっていくと予想されます。

5.2 インフラの IPv6 対応

インターネットを構成するインフラストラクチャには通信会社やプロバイダが持っているさまざまなものがあります。それらの対応は

- 電線や光ファイバといった物理的なネットワークラインはインターネット層の影響を受けないので現状の資産を生かせます
- ルータなどは IPv6 への対応が必要になります（ソフトの変更か、ハードウェアの更新のいずれかが必要）

ということになります。ただし、IPv6 対応はデュアルスタックといった技術によっても行うことができるため、IPv4 を運用している中で作業を進めることができます。

5.3 6bone

IPv6 に関係する活動の中で「6bone」の話題を欠かすことはできません。

6bone は、6bone プロジェクトによって IPv6 のプロトコルの設計段階から運用されている実験用のネットワークです。この 6bone では、実験用のアドレスブロックである「pseudo-TLA」（pTLA）を割り振っています。

ここでは、IPv6 のプロトコル仕様の検証とか各実装の相互接続性、ソフトウェアやハードウェアの動作検証などさまざまなことが行われ、その果たした役割は非常に大きいものです。IPv6 の現在は、この 6bone の運用から生まれているといっても過言ではありません。

日本でも、この 6bone に接続するという形で 6bone-jp が WIDE プロジェクトの主導の元で運用されています。6bone-jp は WIDE プロジェクトが 6bone から pTLA を取得して運用し、この 6bone-jp に接続する組織は WIDE から NLA を取得するという形になっています。

また、「NSPIXP-6」というものが存在します。NSPIXP は、WIDE プロジェクトが主体となって運用しているインターネットの相互接続点です。NSPIXP は、1 と 2 が東京にあり、3 が大阪にあるというようにいままで 3 つ存在していました。そこに IPv6 を示す 6 という番号の NSPIXP ができ、1999 年 8 月に運用が開始されています。これは IPv6 だけの IPv6 による相互接続点で、IPv6 による基幹ネットワークを構成するための基盤技術の研究が行われています。

5.4 IPv6 正式運用へ

このようにさまざまな実験が行われていますが、IPv6 に関しては今年(1999 年)に動きがありました。6bone で運用されているのは実験用の pTLA というアドレスですが、IPv6 のアドレス割り振り基準が今年の 5 月に公開され、それに基づいた正式運用アドレスの割り振りが 7 月に開始されています。今後は、この正式なアドレスの運用が広まっていくことで IPv6 のネットワークが成長していくものと思われます。

5.5 IP アドレスの階層的配分

ここで、IP アドレスの割り振りに関する基本的なことを確認します。

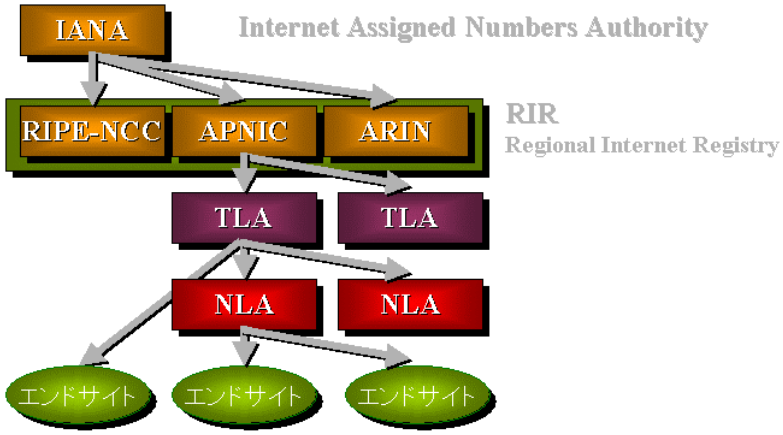

図 35 IPv6 アドレスの割り振り階層

まず、インターネット上で一意になる IP アドレスを割り振るという目的のために、階層的なレジストリ構造が存在しているということを理解してください。IP アドレスは、ユーザーが好き勝手に使うことができないもので、必ずレジストリにアドレスの利用申請を行って、それにより配分されたアドレスを利用しなければいけません。

図のトップにある「IANA」(“Internet Assigned Numbers Authority”の略)という組織は、アドレス全体を管理をする組織として頂点にいます。その下に位置するのが「RIR」(“Regional Internet Registry”の略)という組織で、大陸レベルで存在してアドレスの管理を行います。「RIPE-NCC」がヨーロッパ方面、「APNIC」がアジア・環太平洋方面、「ARIN」がアメリカ大陸を担っています。

IANA は、まずこの RIR に対してアドレスの分配を行います。さらにこの RIR は割り振られたアドレスを TLA 組織に対して割り振ります。そして TLA は自分のアドレスの中から NLA に対して割り振りを行います。多くの場合、ここが ISP になると思われます。ユーザーには、この NLA から最終的な割り振りが行われるという形になります。

5.6 IPv6 アドレスの割り振り

実際にIPv6のアドレスの割り振りが開始されていますが、IPv6の立ち上がりである現状では一番大きい割り振りの単位であるTLAは大きすぎるということで、現時点では/35 の sub-TLA(sTLA)という単位で割り振りが行われています。

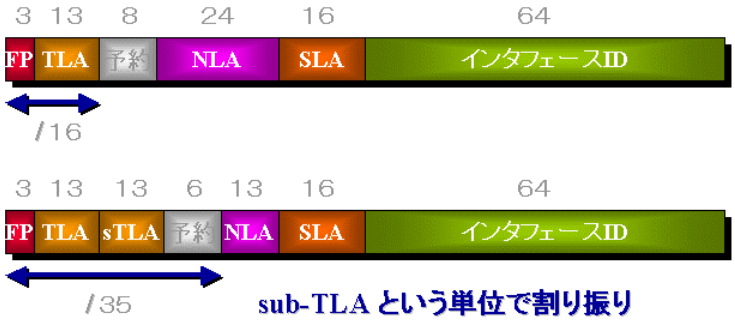

図 36 sub-TLA

これは、いままで説明してきた階層構造と少し異なっています。具体的には、TLAの後ろに sTLA というフィールドを設けたために NLA のフィールドが小さくなっています。

ただし、sub-TLA が欲しいとしても無条件にもらえるわけではありません。RIR は、この割り振りに関して条件を付けています。このためには、

- すでに IPv6 ネットワークを運用していること
- 他の 3 つ以上の sub-TLA 組織の IPv6 ネットワークとピアリングしていること

という条件に加えて、さらに

- すでに IPv6 アドレスを 40 の SLA 顧客サイトに割り当てていること
- sub-TLA を割り振られてから 12 か月以内に IPv6 サービスを提供すること

のいずれかの条件を満たすことを示さなければいけません。

ただし、よく考えてみると IPv6 の運用は始まったばかりですから、条件として示されたような状況が生まれるわけはありません。そのために、現在の状況に応じた初期の割り振り条件というものが設定されています。それは、

- IPv4 ネットワークが他の 3 つ以上のネットワークとピアリングしていること
- sub-TLA の割り振りを受けてから 12 か月以内に IPv6 サービスを提供すること

という条件に加えて、さらに

- 40 以上の顧客に IPv4 アドレスを割り当て済みであること

- 6bone に 6 か月以上参加し、pTLA を 3 か月以上運用していること

のいずれかの条件を満たすこととなっています。

実際に sub-TLA を割り振ることができるのは RIR になりますから、それが必要なときには直接申請を行うことになります。しかし、APNIC は APNIC 会員にしか sub-TLA の割り振りを行っていないので、APNIC から割り振りを受けるためには APNIC 会員になる必要があります。

ちなみに、JPNIC は APNIC 会員であり、JPNIC 会員である組織は JPNIC を通して申請を行うことで APNIC 会員になることなく sub-TLA を申請できます。もしこうした点に興味がある方は、JPNIC の WEB ページなどをご覧ください。

さて、sub-TLA をもらうための条件は非常に厳しく、申請資格があるのはインターネットの基幹を構成するような大手のプロバイダのような組織になってしまうと考えられます。しかし、だからといって IPv6 が使えないかというとそうではなく、一般的な規模の ISP などは NLA として sub-TLA を持つ組織からアドレスの割り振りを受けることになります。

実際、sub-TLA を割り振られた組織が下位の NLA 以下にどのように割り振りを行うかはその組織のポリシーによります。ただし、その場合でもグローバルな割り振りポリシーに従う必要はありますので、必ずその範囲に収まるようにしなければいけません。

6 IPv6 のこれから

最後に、IPv6 のこれからを簡単にまとめて終わりとします。

インターネットが IPv4 から IPv6 へと移行するのは必然の流れであり、広く共通の認識となりつつあります。ただし、IPv4 のネットワークがいつまで主流であるのか、どのタイミングで IPv6 に移行するのか、どういった形で IPv6 になっていくのかといったことは予測の範囲でしか考えることができず、誰にもわかるものではありません。

IPv6 は機能拡張などで検討中の部分も多いが、基本的なプロトコル部分だけでも移行する価値は大きいと考えられます。ですから、いつまでも IPv4 にしがみつくのではなく、早い時期から準備を進めることが肝要ではないでしょうか。

結論としては、「IPv6 の未来はバラ色ではないが明るい」ということだと思います。