



今月のテーマ

# IPv4/IPv6 共存技術

今回の10分間講座は、IPv4/IPv6共存技術について解説します。

今回は、IPv4とIPv6を共存させる、また、IPv4からIPv6への移行の際に用いられる技術である、「トンネリング」「デュアルスタック」「トランスレータ」について説明します。これらは、自組織ネットワークや接続先ISPがIPv4のみに対応している状況下でIPv6の通信を行う時、あるいは段階的なIPv6への移行の際に利用できる技術です。

## トンネリング

IPv6/IPv4トンネリング（以下、トンネリング）は、IPv4ネットワーク上でIPv6パケットをルーティングするための方式です。トンネリングは、カプセル化とも呼ばれており、IPv4ネットワークを通してIPv6ノード同士で通信ができるようにするものです。

### ■ トンネルの種類

トンネルにはいくつか方式があります。ここでは以下の二つの方式について説明します。

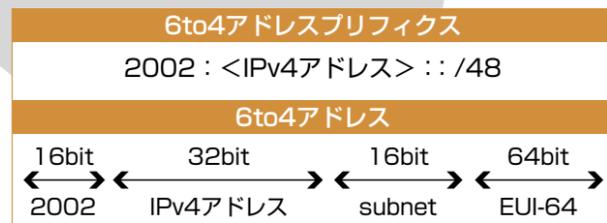
- 6to4 (RFC 3056)
- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) (RFC 4214 (Experimental))

### ■ 6to4

6to4は、ルータ-ルータ間の自動トンネリング技術で、IPv4グローバルアドレスを一つ以上持つサイトに対し、ユニークなIPv6アドレスプレフィックスを割り当て、IPv4ネットワークを通じてカプセル化したIPv6パケットを転送する技術です。

IPv4グローバルアドレスに対して割り当てられる6to4プレフィックスと6to4アドレスは、以下のように決められています。(図1)

図1 6to4アドレスとプレフィックス

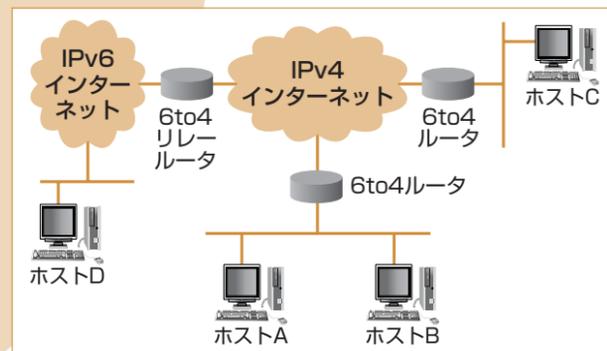


このアドレスプレフィックスを用いて、IPv6に備わっている標準的な方法で、動的もしくは静的に、各サイト内のノードへ6to4アドレスを割り当てることができます。

### ■ 6to4における通信

6to4ホストと6to4ルータによって以下のような通信を行うことができます。

図2 6to4での通信



(1) 同じ6to4アドレスプレフィックスを持つホスト同士の通信

通信先ホストのアドレスプレフィックスが、自身の持つ6to4プレフィックスと同じ場合、6to4ルータは特別な処理を行わず、通常のサイト内IPv6ネットワークを介してIPv6パケットがルーティングされます。(図2 ホストA-ホストB間)

(2) 別の6to4アドレスプレフィックスを持つホスト同士の通信

ある6to4ホストが、別のサイトにある6to4ホストと通信する場合は、6to4ルータが、IPv6パケットをIPv4インターネットを通じて、目的とするサイトの6to4ルータへ転送します。

この時、6to4ルータは、IPv6パケットをIPv4パケットにカプセル化して、相手先の6to4ルータへ送信します。カプセル化の時に用いられる送信先IPv4アドレスは、IPv6パケットにある6to4アドレスプレフィックスから取り出されます。相手先6to4ルータは、受け取ったIPv4パケットのカプセル化を解除し、元のIPv6パケットを取り出します。取り出したIPv6パケットは、通常のサイト内IPv6ネットワークを介して、相手先6to4ホストにルーティングされます。(図2 ホストA-ホストC間)

(3) IPv6インターネット上のネイティブなIPv6ホストとの通信

6to4ホストがIPv6インターネットのホストと通信する場合、6to4ルータがIPv6パケットをカプセル化しますが、上記(2)とは異なり、パケットはIPv4インターネットとIPv6インターネットの両方に接続されている6to4リレールータに対して送信されます。

6to4リレールータは、受け取ったIPv4パケットのカプセル化を解除し元のIPv6パケットを取り出した後、IPv6インターネットにある目的のIPv6ホストへ送信します。(図2 ホストA-ホストD間)

## ■ Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ISATAPは、あるサイト内のローカルIPv4ネットワークにおいて、IPv6通信を実現するトンネリング技術です。

ISATAPで使用されるIPv6アドレスは、以下のような手順で決定されます。

(1) ISATAPインタフェース識別子

まずはじめに、ノードにつけられているIPv4アドレスを元にISATAPインタフェース識別子が作られます。(図3)

図3 ISATAPインタフェース識別子

:: 5efe : <IPv4アドレス>

この時のIPv4アドレスは、グローバルアドレス、プライベートアドレスのどちらでも使用できます。ISATAPインタフェース識別子は、ローカルサイトで一意となります。これを用いてリンクローカルアドレスが作成できます。

この時点で、同一サブネット内に存在する他のISATAPホストとは、IPv4パケットへカプセル化することで、このISATAPインタフェース識別子を元にIPv6による通信を行うことができます。

(2) ISATAPルータ

(1) で得られるアドレスはリンクローカルアドレスのため、同じサブネット内のISATAPホストとは通信できませんが、他のサブネットにあるISATAPホストとは通信できません。

そのため、ISATAPでは、ISATAPルータへカプセル化したパケットを送ることで、サブネットを越えた通信を行います。

ISATAPホストがISATAPルータを見つける手順は、以下のようにして行われます。

- "ISATAP"という名前解決を行います。これによってローカルサイトに存在するISATAPルータのアドレス一覧を得ます。
- a. で得られたアドレスに対し、Router Solicitation (ルータ要請) メッセージを送って、ローカルIPv6アドレスもしくはグローバルなIPv6アドレスプレフィックスを取得します。また、デフォルトルートとしてISATAPルータを登録し、トンネルを構成します。

このISATAPアドレスのIPv6プレフィックスには、リンクローカル、ローカルIPv6アドレス、6to4プレフィックス、グローバルユニキャストアドレスのどれでも使うことができます。例えば、上記で説明した6to4プレフィックスと組み合わせることも可能であり、ISATAPルータと6to4ルータを設置すれば、自組織内ネットワークと接続ISPの両方がIPv4のみの環境であっても、IPv6通信が行えるようになります(図4)。

図4 ISATAPと6to4を組み合わせた例

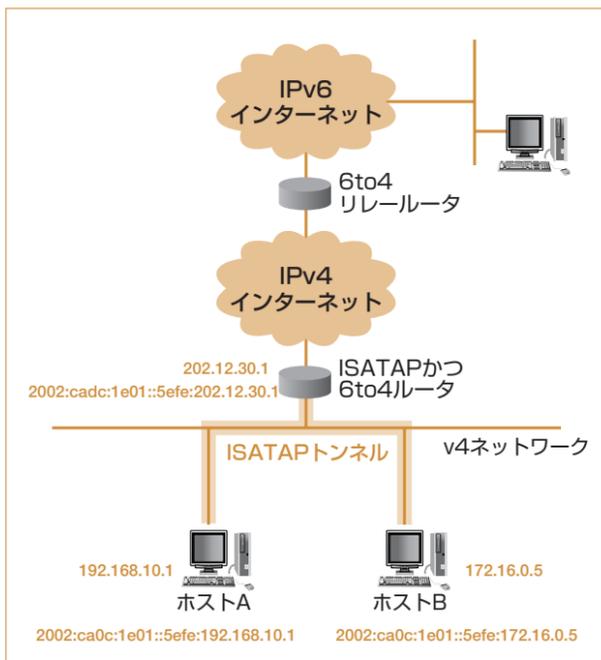


図6-1 デュアルスタックノード間の通信

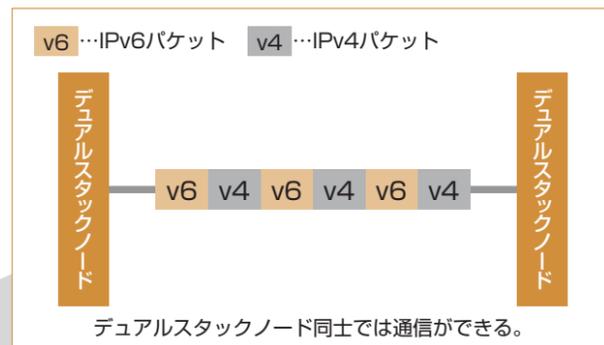


図6-2 IPv4ノード間の通信

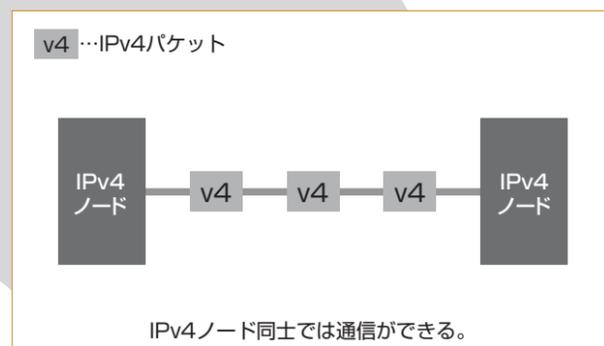


図6-3 IPv6ノード間の通信

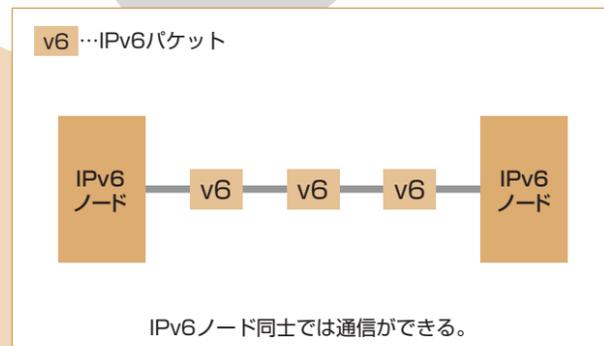
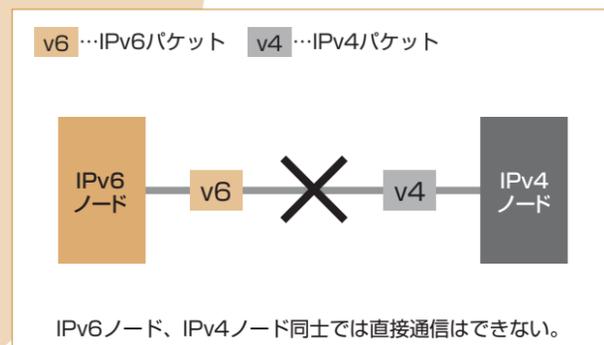


図6-4 IPv4ノード、IPv6ノード間の通信

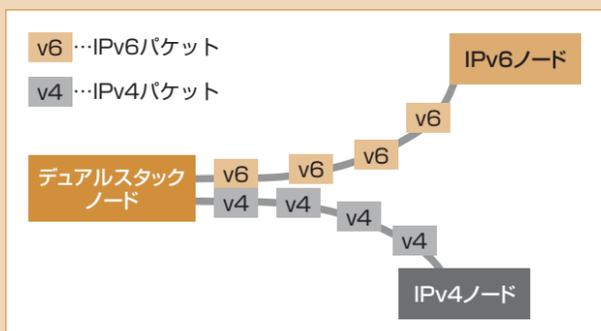


## デュアルスタック

### ■概念

「IPv4/IPv6デュアルスタック」(以下、デュアルスタック)とは、単一機器にIPv4とIPv6という仕様の異なるプロトコルスタックを共存させる仕組みです。この技術により、単一の機器においてIPv4とIPv6を同時に動作させることができます。IPv4対応機器と通信を行う際にはIPv4を使用し、IPv6対応機器と通信を行う際にはIPv6を使用することになります。(図5、図6-1、図6-2、図6-3、図6-4)

図5 デュアルスタック概念図



### ■長所

アプリケーションによっては、IPv4のみでしか動作しないものもあります。そのような場合に、デュアルスタックでは旧環境を残しつつIPv6に対応させることが可能なため、IPv6への移行がしやすくなります。

### ■短所

デュアルスタック実装時には、単一の機器でIPv4とIPv6という二つのプロトコルが同時に動作することになりますので、機器の負荷も相応に増大します。具体的には、IPパケットの行き先を決定する経路制御処理が、IPv4とIPv6それぞれで行われることとなります。そのことによりCPUやメモリにも負荷がかかります。

### ■DNSについて

ホスト名(ドメイン名)とIPアドレスの名前解決について、IPv4アドレスではAレコードを用いますが、IPv6アドレスではAAAA(クアッド)レコードが用いられます。基本的には名前解決のクエリをDNSリゾルバが送信する時点で、AレコードもしくはAAAAレコードのどちらで回答を求めるのか決まっています。

ただし、1回の名前解決でAレコードとAAAAレコードの双方の情報が得られることがあります。例えば、NSレコードの名前解決を実施した際に該当ネームサーバのIPアドレスがグルーレコードとして付加的に送られてくる場合がありますが、そのネームサーバについてAレコードとAAAAレコードの双方にて記述されている状況があります。(図7)そのような場合、どちらのIPアドレスを利用するのかについては、ホストのアプリケーションやDNSリゾルバでの実装によります。

図7 dig出力例

```
% dig ns example.jp
;<<> DIG 9.2.3 <<> ns example.jp
:: global options: printcmd
:: Got answer:
:: ->HEADER<<- opcode: QUERY, status: NOERROR, id: 29469
:: flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2
:: QUESTION SECTION:
;example.jp.                IN      NS
:: ANSWER SECTION:
example.jp.                 86400  IN     NS      ns1.example.jp.
:: ADDITIONAL SECTION:
ns1.example.jp.            86400  IN     A       192.168.0.1
ns1.example.jp.            86400  IN     AAAA    fe80::53
(後略)
```

### ■経路制御について

経路制御プロトコルについて、IPv4とIPv6とでそれぞれ別の経路制御プロトコルを用いる必要があります。

IPv4経路制御プロトコルを基にした各IPv6経路制御プロトコルが開発されています。具体例としては(図8)をご参照ください。

図8 経路制御プロトコルの対応表

IPv4	IPv6
RIP	RIPng
OSPF	OSPFv3
BGP	BGP4+

## トランスレータ

IPv6からIPv4の通信や、IPv4からIPv6への通信を行うには、IPv6とIPv4の間を取り持つトランスレータが必要になります。

トランスレータは、通信元や通信先がIPv6、IPv4の差異を意識しなくてすむ仕組みを提供します。代表的なトランスレータを実装する技術として以下の3種類について解説します。

### (1) Proxy方式

Proxy方式は、IPv4で利用されるProxyと同様に、アプリケーション毎に送信元の代理となって送信先へ通信を行う方式です。

### (2) NAT-PT方式

NAT-PT方式は、プロトコルのアドレスやポート番号を変換し、IPv4、IPv6間を相互に通信する方式です。

### (3) TCP Relay方式

TCP Relay方式はトランスポート層で、セッションを横取りしてTCPやUDPプロトコル間の通信を行う方式です。

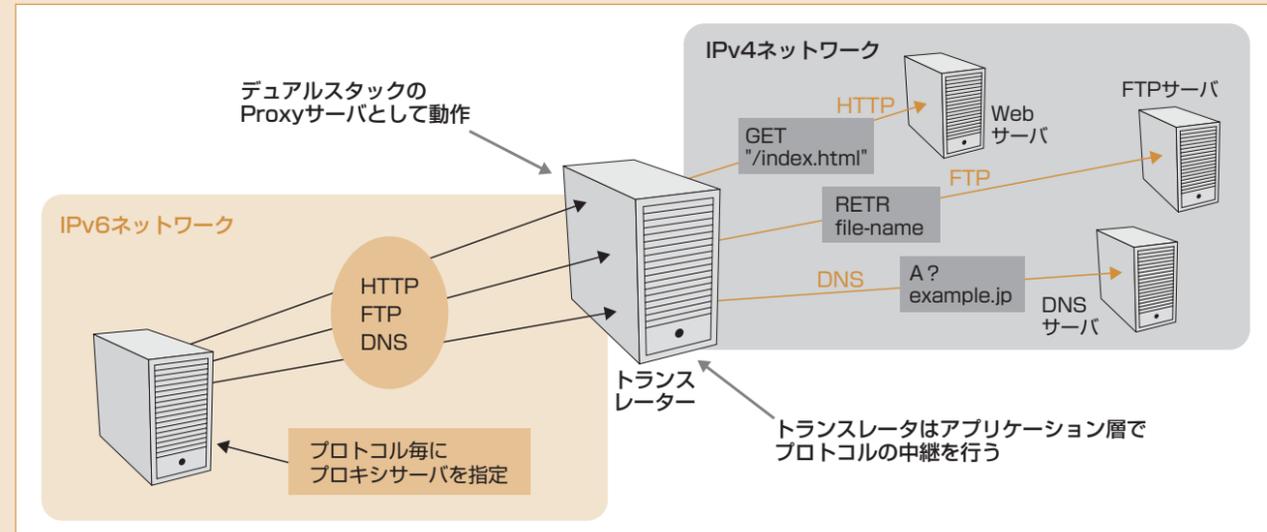
■ 詳解

(1) Proxy方式

Proxy方式は、デュアルスタック方式のサーバへ、HTTPやFTPなど、アプリケーション層でのデータ中継を行う方式です。実装は他のトランスレータ技術と比較して

容易ですが、アプリケーション層でのプロトコル毎にトランスレータを用意する必要があります。アプリケーションゲートウェイ方式とも呼ばれます。

図9 Proxy方式



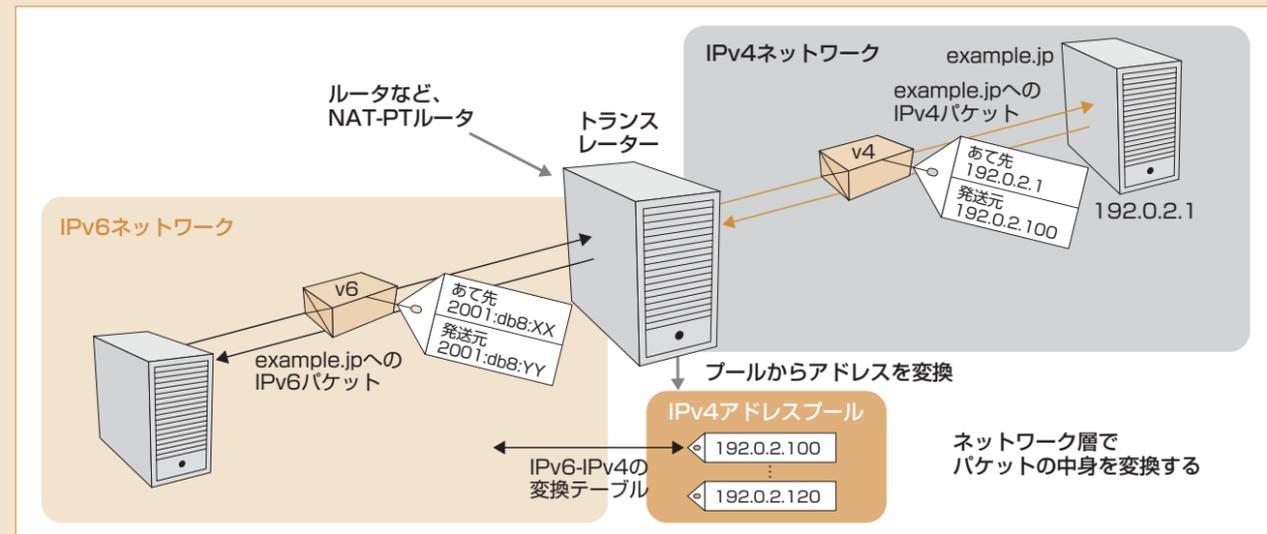
(2) NAT-PT方式 (RFC2766)

IPv4でのNATやIPマスカレードのように、プロトコルのヘッダ部分を変換する方式をNAT-PT方式と呼びます。RFC2766で規定されるNAT-PT方式では、DNS-ALG

(RFC2764) を使い、DNSを使ってIPv6~IPv4間の通信をトランスレータで横取りします。

NAT-PT方式は、アプリケーション層で解釈されるペイロード部分の変換ができないなどの問題があります。

図10 NAT-PT方式

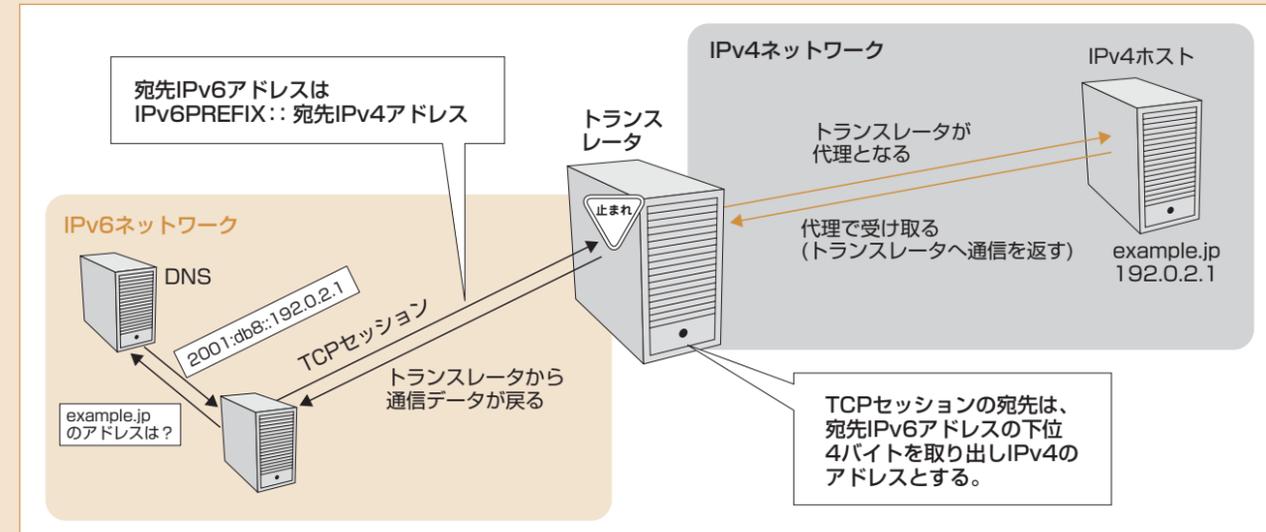


(3) TCP Relay方式 (RFC3142)

トランスレータが、TCPやUDPの通信を横取りし、代理となってトランスポート層のデータを取得する方式です。実際には、TCPコネクションをトランスレータが送

信元へ終端し、IPv6であればIPv4、IPv4であればIPv6のネットワークにトランスレータ自身が代理となって送信先へ通信を行い、データの中継を行います。

図11 TCP Relay方式



(トンネリング: JPNIC 技術部 小山祐司)  
(デュアルスタック: JPNIC 技術部 澁谷 晃)  
(トランスレータ: JPNIC 技術部 岡田雅之)