

# 本資料について

本資料は下記著書を基にして作成されたものです。著書の内容の正確さは保障できないため、正確な知識を求める方は原本を参照してください。

監修：平松直樹

著書名：「IPv6 ネットワーク構築実験技法」

出版社：(株)オーム社

発行日：2001年8月24日 第1版第1刷



# IPv6 ネットワーク 実践構築

渡邊研究室

00J125 増田 真也

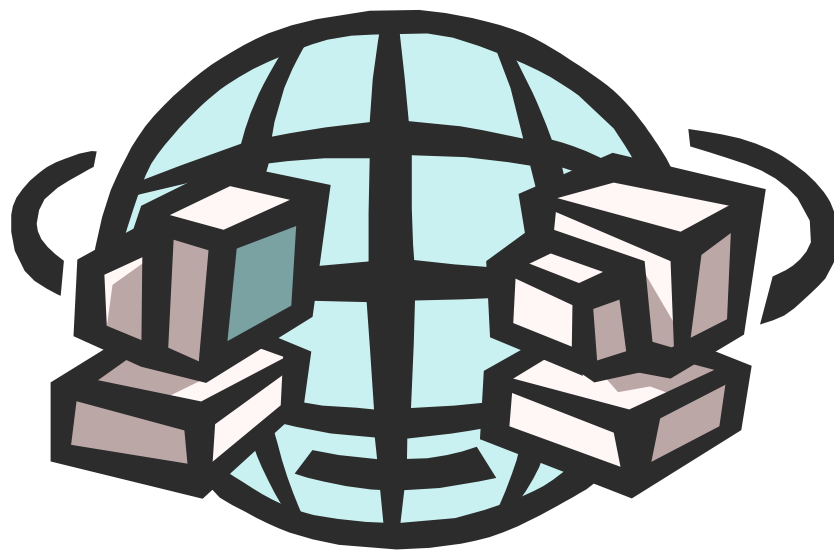


# はじめに

## ○ 全体の流れ

- 第1章 IPv6 概要
- 第2章 プロトコル概要
- 第3章 移行技術と移行ツール
- 第4章 IPv6 ネットワーク構築実験

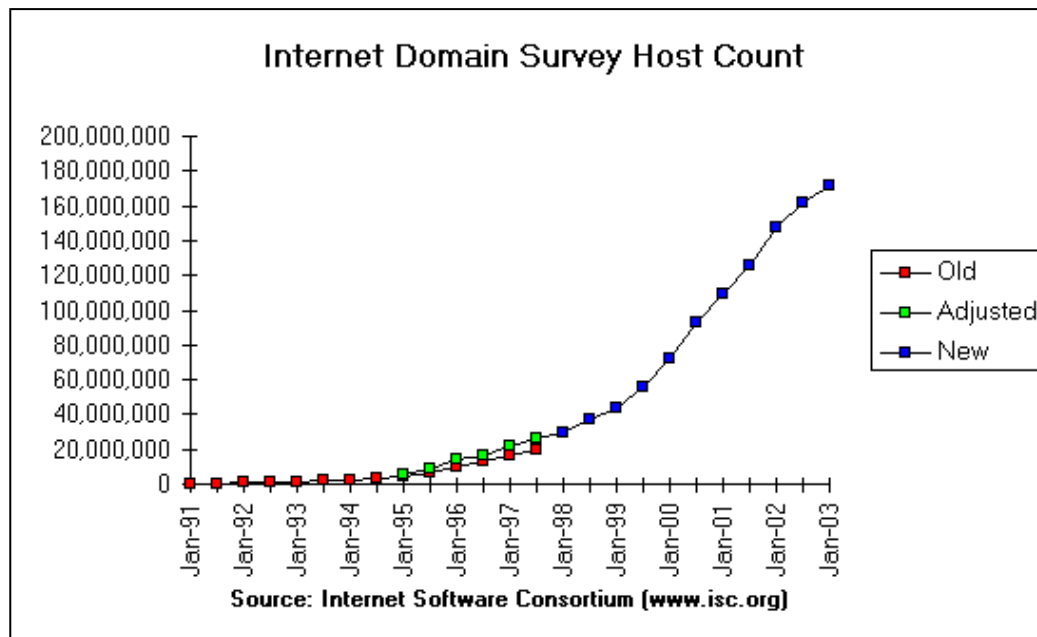
# 第1章 IPv6概要



# 1-1 . IPv6 とは

## ○ IPv6検討の背景

- インターネットの急速な普及

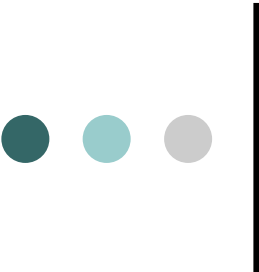


出典: Internet Software Consortium



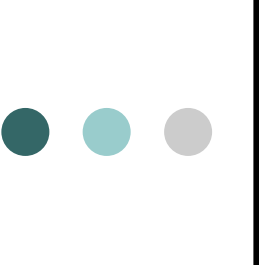
# 1-1 . IPv6 とは

- アドレスの枯渇問題
  - 短期解としてクラスの廃止 (CIDR) を検討 .
  - 長期解として Bigger Address を検討 .
- 長期解に向けた検討 IPng (IP Next Generation)
  - TUBA (TCP and UDP with Bigger Address)
  - TP/IX
  - SIPP (Simple Internet Protocol Plus)
- IPv6へ
  - SIPPベースの次世代プロトコルにバージョン6が割り当てられる .



## 1-2 . プライベートアドレスとNAT

- アドレスの枯渇問題に対応し, 大きく貢献
- 諸問題
  - プライベートアドレスの弊害別々に管理されているネットワークを相互接続する場合にアドレスが重複する可能性がある.
  - NATの限界
    - IPアドレスを運ぶプロトコルは, 運ぶIPアドレスも含めて変換する必要があり, 専用の処理が必要で汎用性がない.
    - IPsecで暗号化されたアドレスは変換ができない.



## 1-3 . IPv6の影響範囲

- IPに関わるプロトコルは変更が必要
  - ネットワーク層のプロトコル
  - FTPとUDPはIPヘッダの一部を利用した擬似ヘッダを用いる
  - IPアドレスを直接扱うようなアプリケーション
  - IPを直接処理する機器





# 1-4 . IPv6 が開く可能性

- アドレスの増大 様々な機器がネットワークにつながる
- パフォーマンスの向上
  - ヘッダ構造の簡素化によるルータの負荷軽減
  - ルータに分割処理をさせない
- プラグ&プレイ機能を必須に 端末導入手順の簡便化
- 品質保証機能 (QoS / CoS) 音声, 映像通信への対応
- IPセキュリティ
  - IP層での機能なので, アプリケーションによらず本機能を適用できるようになる
- マルチキャスト, Mobile IPv6 新たなアプリケーションの可能性

# 第2章 プロトコル概要



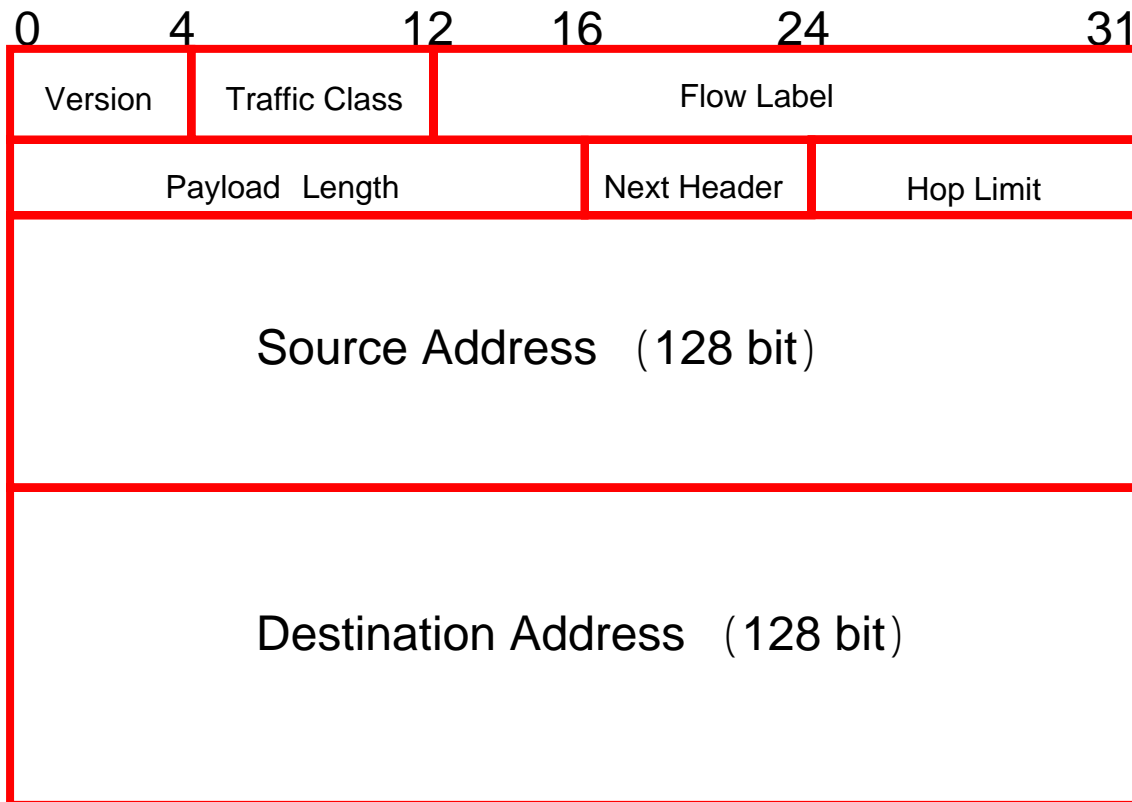


## 2-1 . IPv6 ヘッダ

- IPv6 プロトコルの概要
  - 基本的な考え方はIPv4と変わらない
  - IPヘッダの単純化
    - 基本ヘッダが固定長に
    - ヘッダ長とチェックサムの領域の削除
  - 拡張ヘッダによる拡張性
  - アドレス空間の拡大
  - IP層でのセキュリティ
    - IPSecが標準になる

# 2-1 . IPv6 ヘッダ

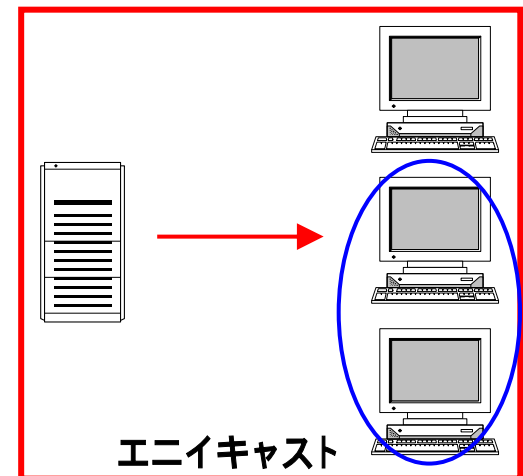
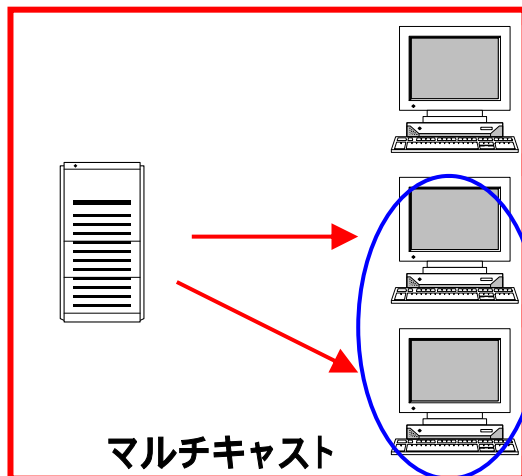
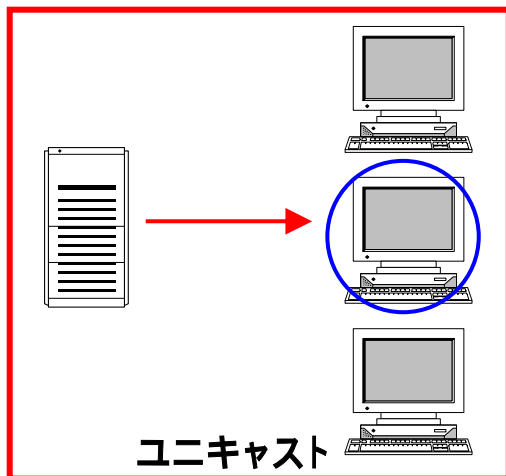
## ○ IPv6 基本ヘッダ



## 2-2 . IPアドレス

### ○ IPv6 アドレスの種類

アドレスの種類	アドレス割り当て インタフェース	パケット転送される インタフェース
ユニキャスト	1	1
マルチキャスト	任意数	任意数の全て
エニイキャスト	任意数	任意数のうちの一つ





## 2-2 . IPアドレス

### ○ アドレスの表記法

- 128ビットのアドレスを16ビット単位に区切り, それぞれを16進数で表記し, “:”で区切る.

例) FF01:0:0:0:0:0:0:101

- 連続する0については1度だけ“::”として短縮して表記できる.

例) FF01::101

### ○ スコープ

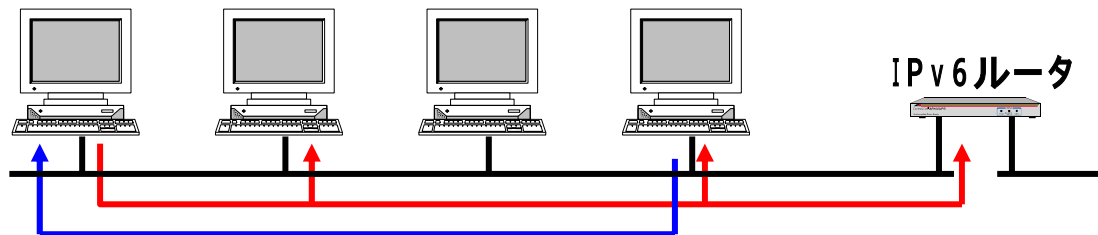
- IPv6アドレスは「スコープ内」で一意
  - link-local scope : あるリンク内でのみ一意
  - site-local scope : 組織内で一意
  - global scope : 全世界で一意
- インタフェースにスコープの異なる複数のアドレスをつけることが可能

プライベートアドレスより柔軟なセキュリティポリシーの実現  
しかし, DNSデータベースやルータでの経路管理が面倒になる

## 2-3 . ICMPv6 と近隣探索

### ○ ICMPv6の役割

- IP層の制御メッセージが規定されている
  - エラーメッセージ
  - 情報メッセージ
- 近隣探索メッセージ (Neighbor Discovery)
  - IPv6アドレスとMACアドレスの対応関係を調べる (IPv4でのARPに該当) .



近隣要請メッセージをマルチキャストで送り, MACアドレスを問い合わせる.  
近隣告知メッセージでMACアドレスを通知する.



# 第3章

## 移行技術と移行ツール





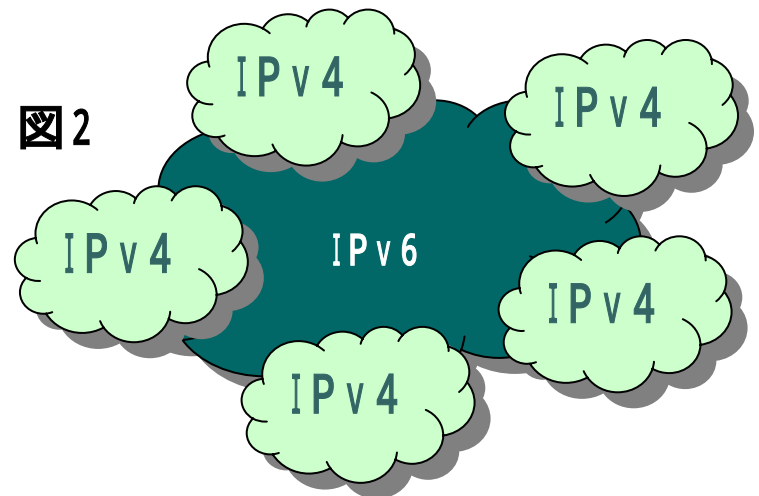
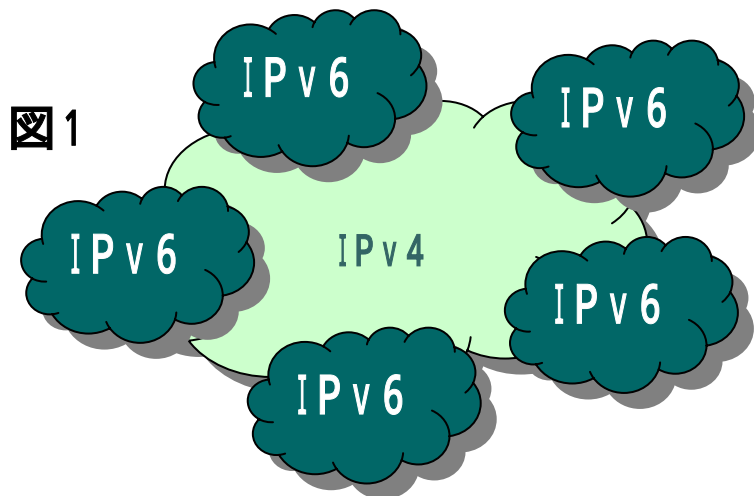
# 3-1 . IPv6への移行

- IPv6への移行は、段階的に進んでいく。

IPv4のみの時代

IPv6普及初期(図1)

IPv6普及後期(図2)





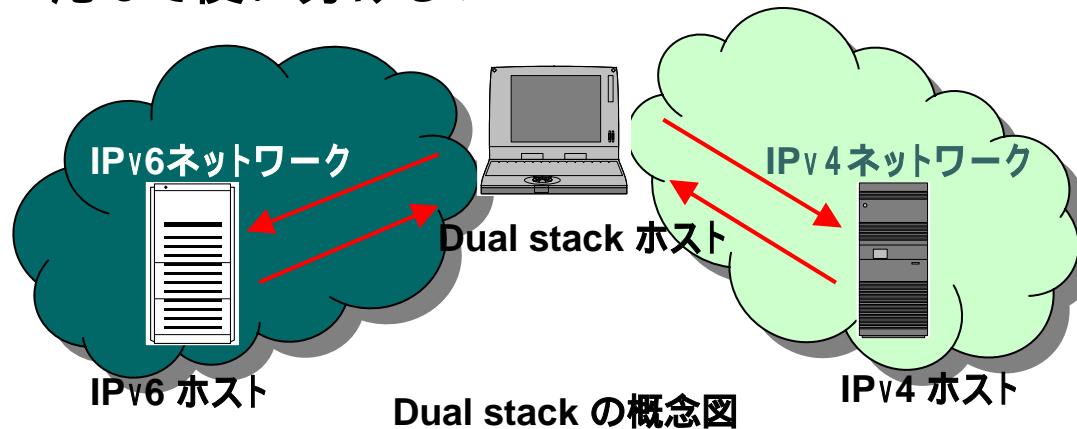
## 3-2 . 移行技術

- IPv4からIPv6へ移行するための技術
  - IPv6/IPv4 Dual stack
  - トンネル
  - IPv6/IPv4 Translation

## 3-2 . 移行技術

### ○ IPv6/IPv4 Dual stack

- IPv6とIPv4のプロトコルスタックの両方を持ち、通信相手に応じて使い分ける。

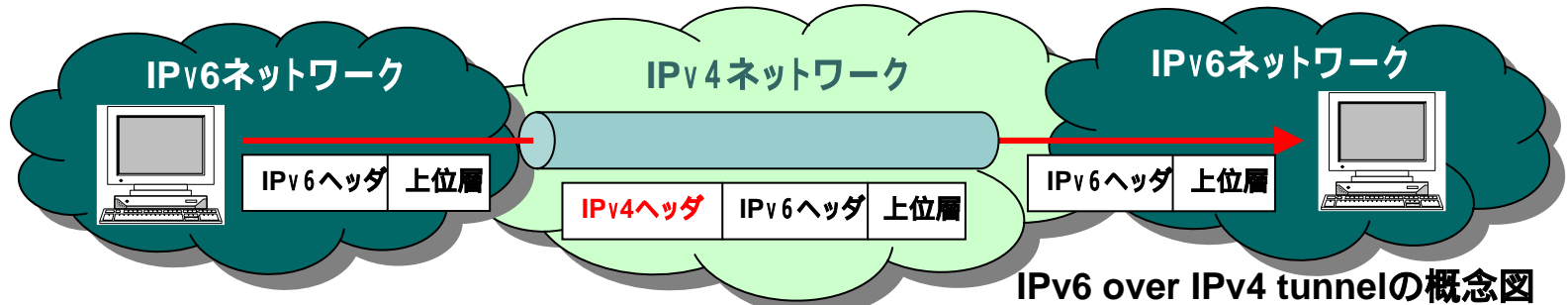


- Dual stackの利点
  - IPv4ホストと通信する場合もネイティブな通信が行える。
- Dual stackの欠点
  - メモリなどのリソースがSingle stackより多く必要。
  - 運用・管理が複雑になることがある。

## 3-2 . 移行技術

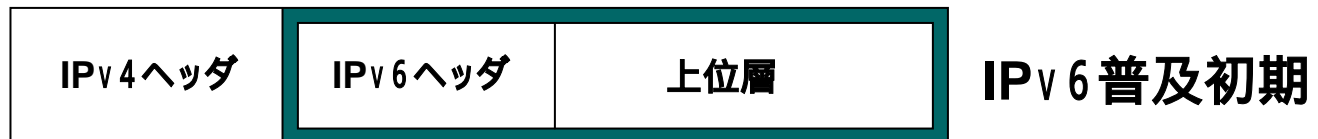
### ○ トンネル

- IPv6のホスト同士がIPv4のネットワークを介して通信



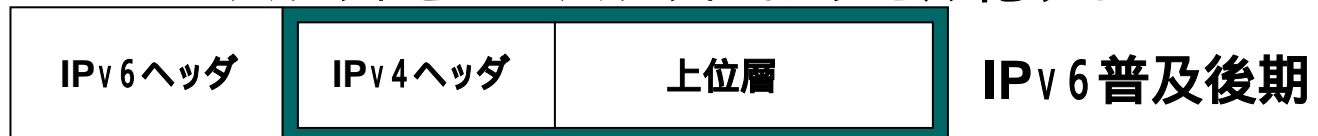
- IPv6 over IPv4 tunnel

- IPv4パケットをIPv4パケットでカプセル化する



- IPv4 over IPv6 tunnel

- IPv4パケットをIPv6パケットでカプセル化する





## 3-2 . 移行技術

- トンネルの技術

- Configured Tunnel : 管理者が手動で設定するもの
- Automatic Tunnel : 自動でトンネルを構築する機能
- 6 to 4 : 自動でトンネルを構築する機能

- トンネルの利点

- ISPがIPv6に対応していない場合に有効

- トンネルの欠点

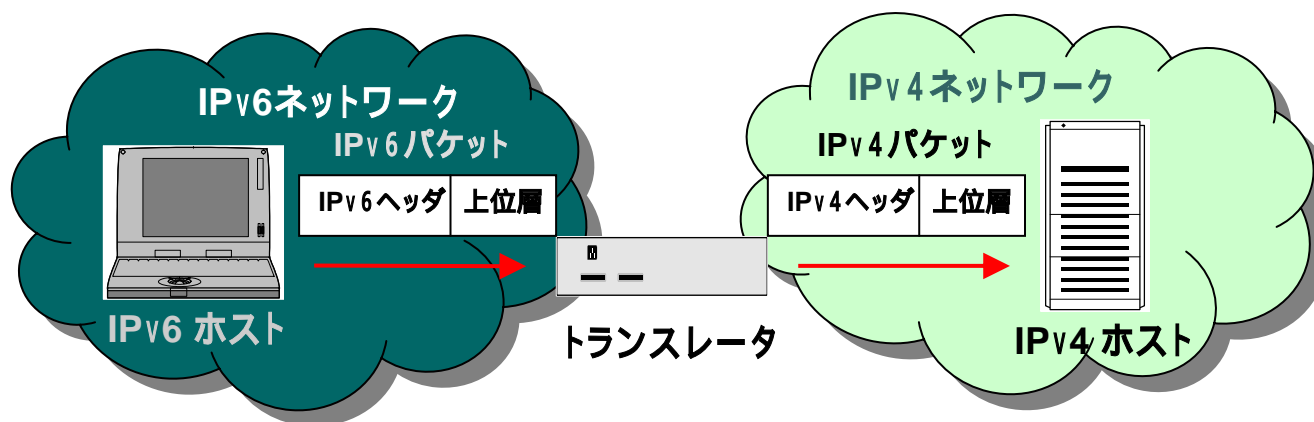
- Dual stackが前提なので, リソース制限のあるホストなどには不向き
- カプセル化する分データは大きくなり, 処理も遅くなる

## 3-2 . 移行技術

### ○ IPv6/IPv4 Translation

- IPv6のホストとIPv4のホスト間の通信をサポート
- リソースの制限などから, 全てのIPv6ノードがDual Stack になれるとは限らない.

IPv6/IPv4 Translationを利用



トランスレータの概念図



## 3-2 . 移行技術

- IPv6/IPv4 Translationの技術
  - Stateless IP/ICMP Translation Algorithm (SIIT)
  - An IPv6-to-IPv4 transport relay translator (TRT)
  - Network Address Translation - Protocol Translation (NAT-PT)
- IPv6/IPv4 Translationの利点
  - Single stackで利用できる  
リソース制限のあるホストに有効
  - ISPがIPv6に対応していない場合に有効
- IPv6/IPv4 Translationの欠点
  - IPアドレスを扱う上位プロトコル部分もアドレスを変換しなければならない場合がある.
  - トランスレータにアドレスの対応をマッピングする必要がある  
利用の制約



## 3-3 . 移行ツール

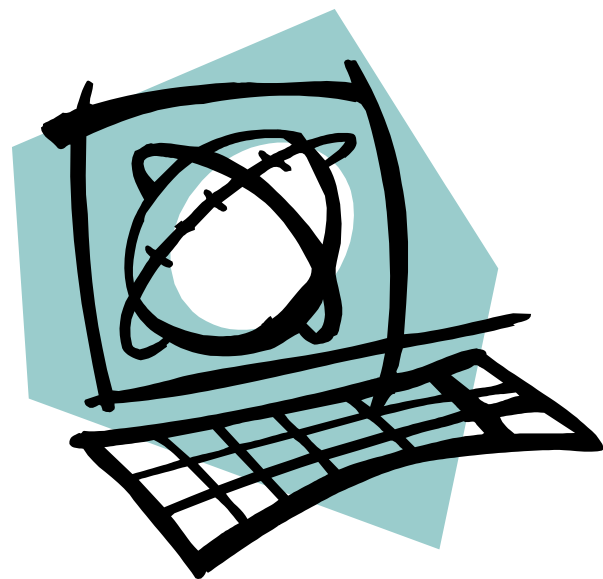
- Dual stack
  - BSD系OS: KAMEパッチ
  - Linux: USAGIパッチ
  - Windows XP: 標準搭載のコンポーネント
- トンネル
  - Configured Tunnel
    - KAMEスタックをはじめ, IPv6対応の大部分で提供
  - Automatic Tunnel
    - LinuxやWindows XPのプロトコルスタックで利用できる
  - 6 to 4
    - KAMEスタックやBSD系OS, Windows XPのプロトコルスタック
- トランスレータ
  - TRT方式: TBB(横河電気の製品)など
  - NAT - PT方式: KAMEスタックにあるNAPT-PTなど



# 第4章

## IPv6 ネットワーク

### 構築実験





## 4-1 . はじめに

- 実験の目的

- とにかく触れて覚える  
基本的な知識・技術の習得

- 実験項目

Windows XP , LinuxのDual stackによるIPv6化

ping6によるWindowsとLinuxの通信

Apache (2.0) の動作確認

IPv6でインターネット

## 4-2 . Dual stackによるIPv6化

### ○ Windows XP

#### ● 導入法

- IPv6コンポーネントが標準搭載なので、コマンドプロンプトから“ipv6 install”と入力するだけ。

#### ● 動作確認

- コマンドプロンプトの「ipconfig」コマンドで、IPv6アドレスが表示されることを確認する。

```
C:¥Documents and Settings¥dasshi>ipconfig
```

```
Windows IP Configuration
```

```
Ethernet adapter ローカル エリア接続:
```

```
Connection-specific DNS Suffix . :
```

```
IP Address. . . . . : 192.168.1.14
```

```
Subnet Mask . . . . . : 255.255.255.0
```

```
IP Address. . . . . : fe80::230:1bff:fead:b664%4
```

```
Default Gateway . . . . . : 192.168.1.1
```

## 4-2 . Dual stackによるIPv6化

- Linux (Red Hat Linux 9)
  - 導入法
    - USAGIパッチをあて(詳細は別資料), ユーザーランドアプリケーションを作成する.
  - 動作確認
    - 「ifconfig」コマンドで, IPv6アドレスが表示されることを確認する.

```
[root@dasshi-srv root]# ifconfig
eth0  Link encap:Ethernet HWaddr 00:00:E2:12:58:C3
      inet addr:192.168.1.13 Bcast:192.168.1.255 Mask:255.255.255.0
      inet6 addr: fe80::200:e2ff:fe12:58c3/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:83 errors:0 dropped:0 overruns:0 frame:0
      TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
```

## 4-3 . ping6による通信

### ○ 応答確認

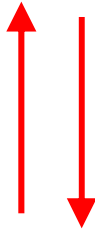
- LinuxからWindowsへ「ping6」コマンドで応答確認する。

Linux



fe80::200:e2ff:

fe12:58c3



Windows



fe80::230:1bff:

fead:b664

```
[root@dasshi-srv root]# ping6 fe80::230:1bff:fead:b664%eth0
```

```
PING fe80::230:1bff:fead:b664%eth0 (fe80::230:1bff:fead:b664%eth0) from fe80::200:e2ff:fe12:58c3%eth0 : 56 data bytes
```

```
64 bytes from fe80::230:1bff:fead:b664%eth0: icmp_seq=1 ttl=128 time=0.286 ms
64 bytes from fe80::230:1bff:fead:b664%eth0: icmp_seq=2 ttl=128 time=0.295 ms
64 bytes from fe80::230:1bff:fead:b664%eth0: icmp_seq=3 ttl=128 time=0.295 ms
64 bytes from fe80::230:1bff:fead:b664%eth0: icmp_seq=4 ttl=128 time=0.289 ms
64 bytes from fe80::230:1bff:fead:b664%eth0: icmp_seq=5 ttl=128 time=0.297 ms
```

```
--- fe80::230:1bff:fead:b664%eth0 ping statistics ---
```

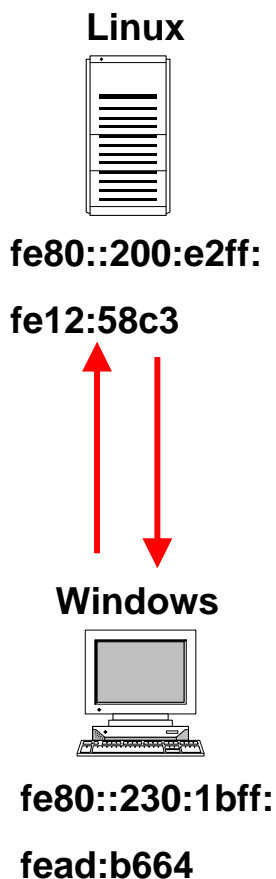
```
5 packets transmitted, 5 received, 0% loss, time 4002ms
```

```
rtt min/avg/max/mdev = 0.286/0.292/0.297/0.015 ms
```



## 4-4 . Apache (2.0) の動作確認

- telnet による動作確認
  - WindowsからLinux (Apache導入済) へtelnetで80番ポートに接続してApacheの動作確認をする。



```
C:¥Documents and Settings¥dasshi>telnet fe80::200:e2ff:fe12:58c3%4 80
```

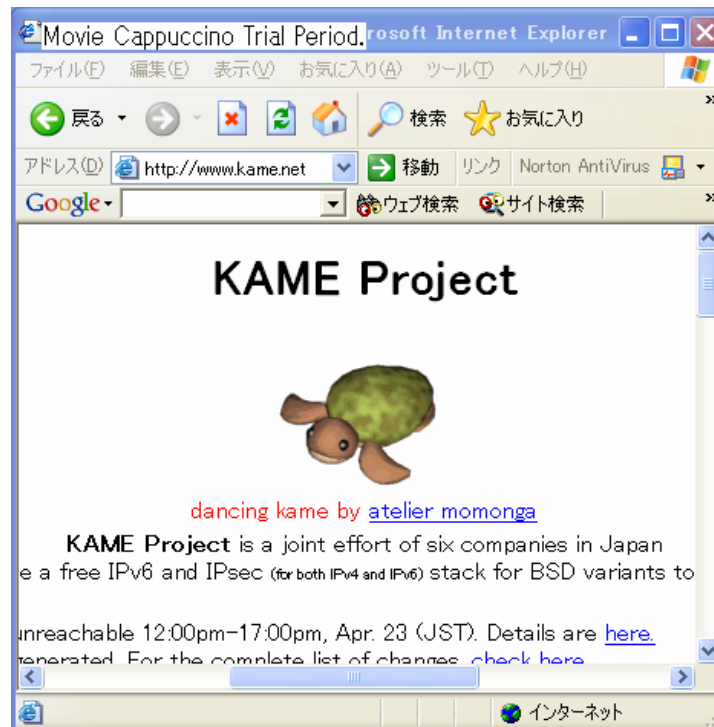
```
GET /
```

```
<div align="center">  
    
  
  <h3>Apache HTTP Server Version 2.0</h3>  
</div>
```

ホストとの接続が切断されました。

## 4-5 . IPv6でインターネット

- ISPのIPv6サービスはまだまだトンネリングで実現するIPv6の世界
  - 比較的簡単な6to4を採用 . その中でも容易なWindows XPで実験 .



アニメーションのカメ . IPv4では静止画のカメである .

<http://www.kame.net>



## 4-6 . 実験の考察・感想

### ○ 考察

- グローバルIPv4を取得できる環境であれば、Dual stack + トンネリングで、IPv6のインターネットが実現可能。
- 動的グローバルIPの場合は、IPアドレス変更時に設定の書き換えが必要。  
DPRPを利用することで、動的に設定変更が可能ではないだろうか。

### ○ 感想

- 実際にIPv6ネットワークに触れたことで、体感を得ることができ、IPv6の実態がつかめた。



# まとめ

- IPv6による新たな可能性
- ネイティブなIPv6が実現するまでは移行技術を用いる
- 実際に触れることで、IPv6が身近なものとなる

詳細ドキュメント

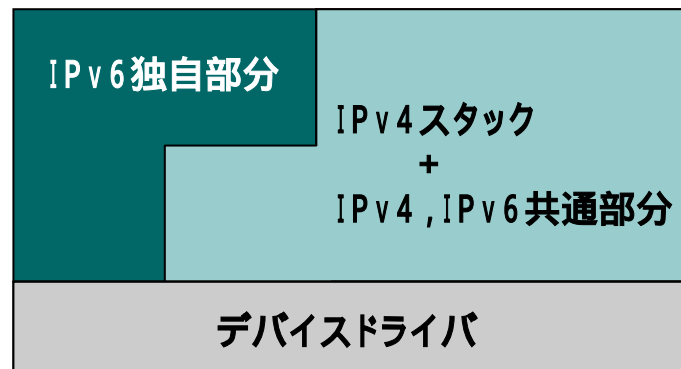
¥¥Wata\_lab-srv¥documents¥IPv6実践構築

おわり

# LinuxのIPv6実装状況

- FreeBSDなどのKAMEプロジェクトの環境と比べると、あまり芳しくない。
- スタックの未分離問題
  - IPv6のプロトコルスタックは、IPv4のプロトコルスタックと別になっていない。

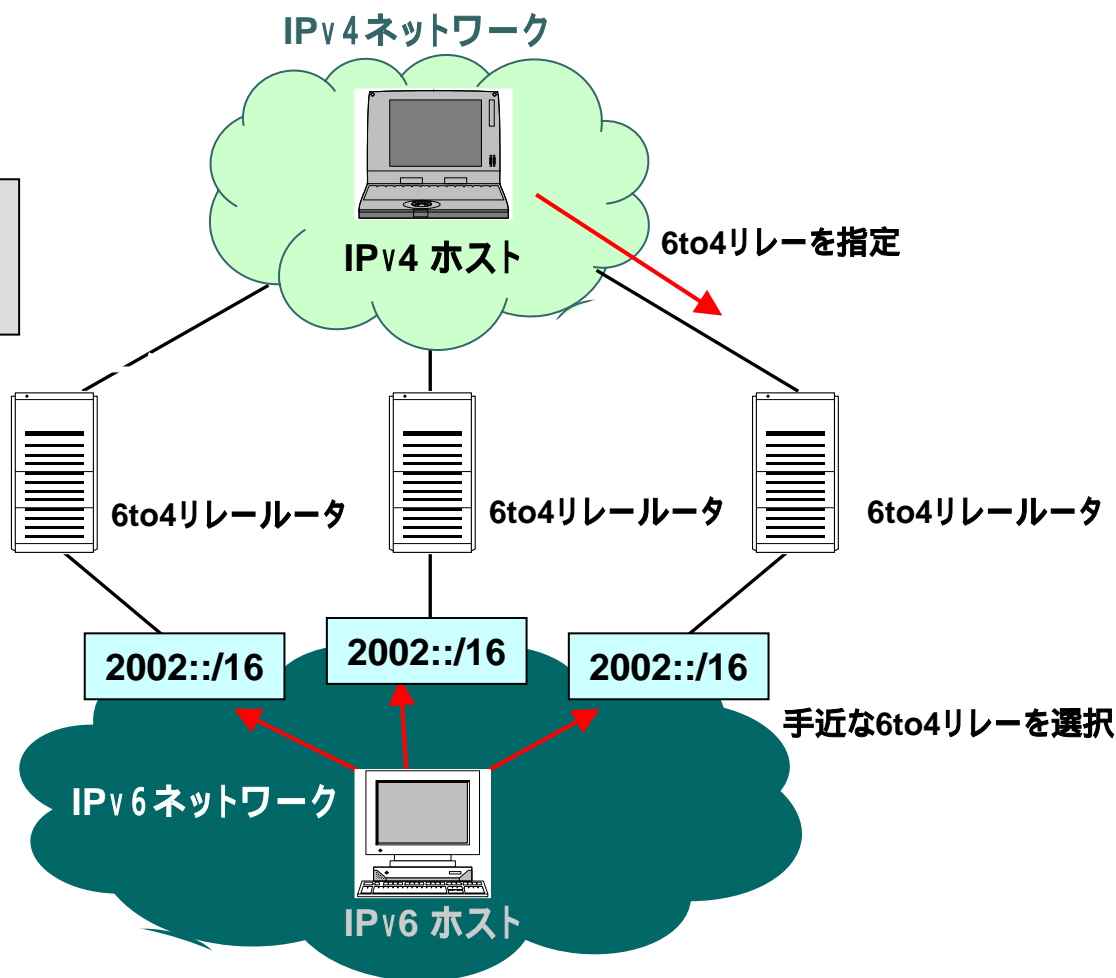
IPv6 onlyのマシンを作ることができない。



カーネル内部のプロトコルスタックの構造

# 6to4によるトンネリング

本実験では, KDDI研究所のkddilab.6to4.jp をリレールータとして指定



6to4 ネットワーク の概念図



# 参考文献・サイト

- 「IPv6 ネットワーク構築実験技法」  
監修:平松直樹 出版:(株)オーム社
- 「マスタリングTCP/IP 入門編」  
竹下隆史・村山公保・荒井透・苅田幸雄 共著 出版:(株)オーム社
- 「IPv6 次世代インターネットプロトコル」  
クリスチャン・ウイテマ 著 出版:(株)ピアソン:エデュケーション
- Linuxで作るIPv6ネットワーク環境  
<http://www.atmarkit.co.jp/flinux/special/usagi/usagi01a.html>
- 6to4  
<http://www.6to4.jp/>