IPv4 EXHAUSTION

実践！ 初めてのIPv6
～サーバ編～

Internet Week 2010

本セミナーについて

- IPv4アドレス枯渇対応タスクフォースが主催
  - 2008年発足
  - 総務省を含むインターネット関連22団体がIPv4枯渇を乗り切るための方針を立て、啓蒙活動、IPv6サービスの認定、人材育成等、様々な活動を実施中
  - IPv6ハンズオンセミナーもタスクフォースの実施する活動の1つ

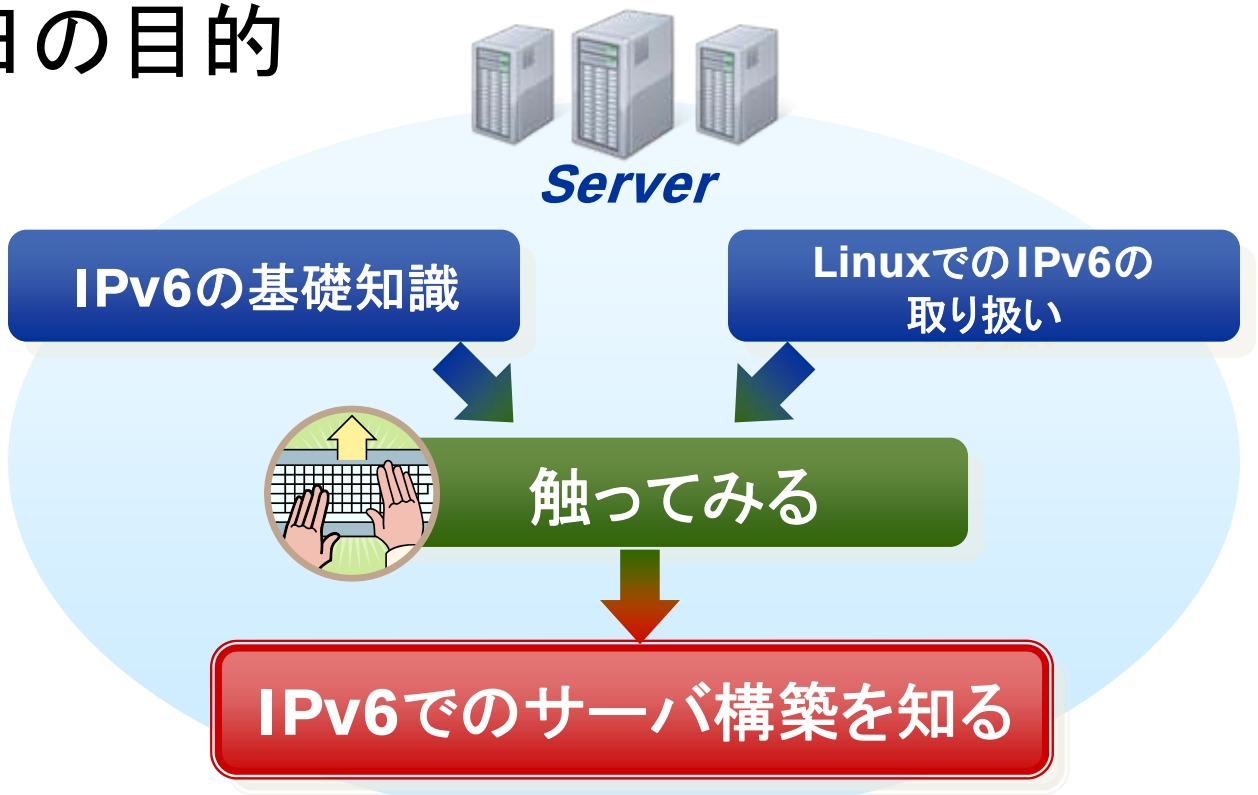
- 本件は、平成22年度総務省施策「IPv6対応に向けたテストベットによる実証実験に係る請負」の一環として実施しております。
  参考URL：http://www.kokatsu.jp/blog/ipv4/event/2011/03/ipv6-handsonseminar.html

サーバ編　講師紹介

- 鈴木一広@NTT Com/OCN
  - 法人向けネットワーク（IP-VPN、広域LAN等）の運用保守を3年間担当
  - 現在、OCNメール等コンシューマ向けサービスを提供するための業務サーバの設計・開発に従事

セッションに先立ち

- 本日の目的



（注）ディストリビューションは、CentOS（RedHat系）を使用します。

セッションのアウトライン

A) 講義

1. IPv6の基礎知識
   - はじめに押さえておきたいIPv6の基礎
2. LinuxでIPv6を扱う
   - ネットワークの設定と確認、各種サーバのIPv6設定概要
3. 【参考】IPv6運用の留意点

B) ハンズオン

– Linuxで、実際のIPv6設定を体験
  - ネットワーク設定、サーバ設定（Web、DNS、メール…）

A）講義

1. IPv6の基礎知識

1-1. 広大なアドレス空間
1-2. アドレスの自動設定
1-3. アドレスの表記
1-4. アドレスの種類

1. IPv6の基礎知識

1-1. 広大なアドレス空間

- IPv4（32bit）　＝約43億個
- IPv6（128bit）　＝約340澗個
  - 億＜兆＜京＜垓＜杼＜穣＜溝＜澗

- IPv4のアドレス空間を1とすると、IPv6は

79,228,162,514,264,337,593,543,950,336

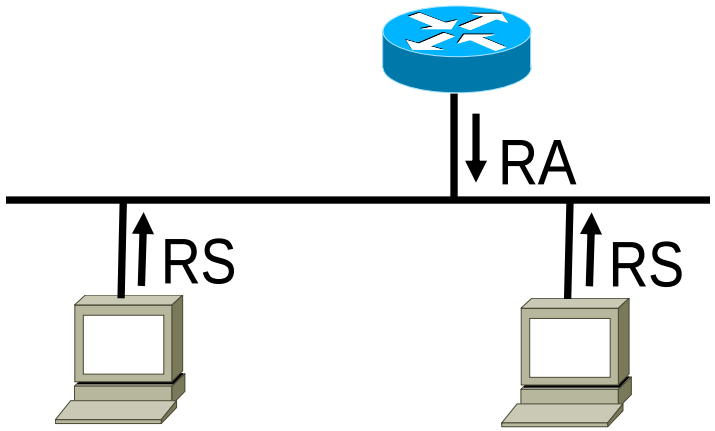
  - 全世界の人ひとりずつにIPv4アドレス空間を配ってもまだまだ余る！
    - 携帯電話、カーナビ、インターネット家電、センサ等にも割当て可能な膨大なアドレス空間

※ただし、ネットマスクの考え方の違いから、上の例ほど単純には比較できない

1. IPv6の基礎知識

1-2. アドレスの自動設定

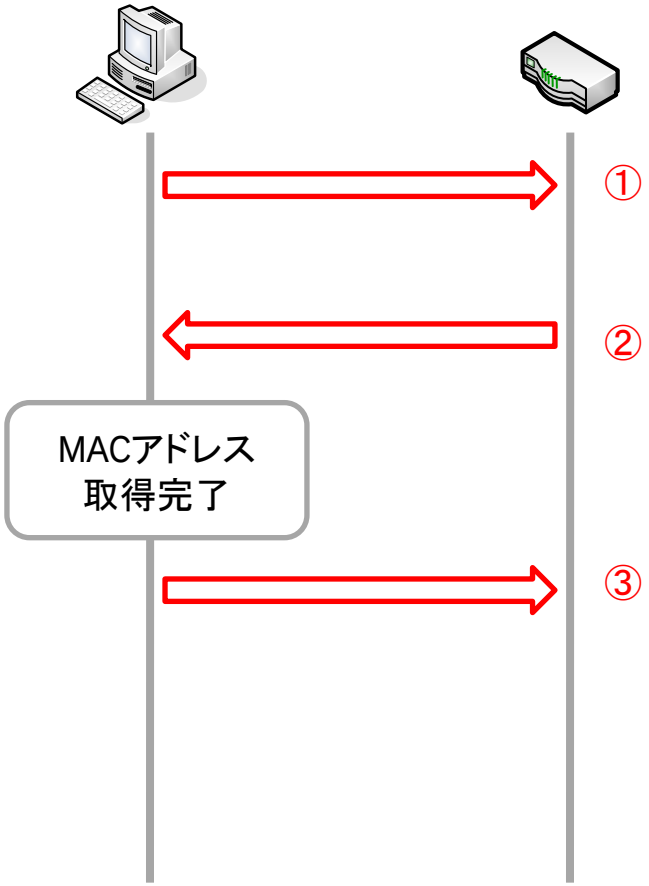
- IPv6では、予想される莫大な数のデバイスに対応するため、接続するだけでインタフェースにアドレスが自動的に設定される機能が標準で用意されている
  - DHCPv6を利用することも可能



【参考】リンクレイヤアドレスの解決の流れ

fe80::211:22ff:fe33:4455
2001:db8::211:22ff:fe33:4455
MAC:00:11:22:33:44:55

fe80::211:22ff:fe66:7788
2001:db8::211:22ff:fe66:7788
MAC:00:11:22:66:77:88



①近隣要請（NS）
通信相手のMACアドレスを探索
近隣広告がない場合はオンリンクでないと判断

Src MAC	00:11:22:33:44:55
Dst MAC	33:33:FF:66:77:88
Src IPv6	fe80::211:22ff:fe33:4455
Dst IPv6	ff02::1:ff66:7788
ICMPv6 Type	135
Target	2001:db8::211:22ff:fe66:7788

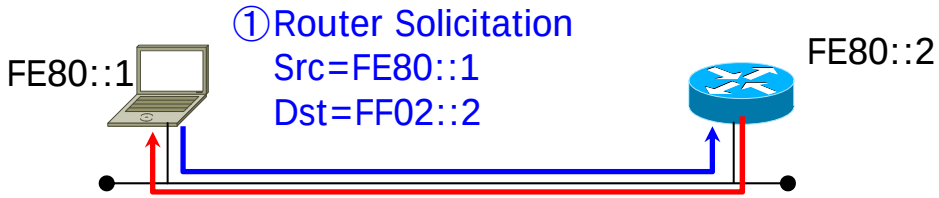
②近隣広告（NA）
ターゲットアドレスを持つノードが回答
ただし誰でもこの応答は可能

Src MAC	00:11:22:66:77:88
Dst MAC	00:11:22:33:44:55
Src IPv6	fe80::211:22ff:fe66:7788
Dst IPv6	fe80::211:22ff:fe33:4455
ICMPv6 Type	136
Target	2001:db8::211:22ff:fe66:7788
Target MAC	00:11:22:66:77:88

③通信開始

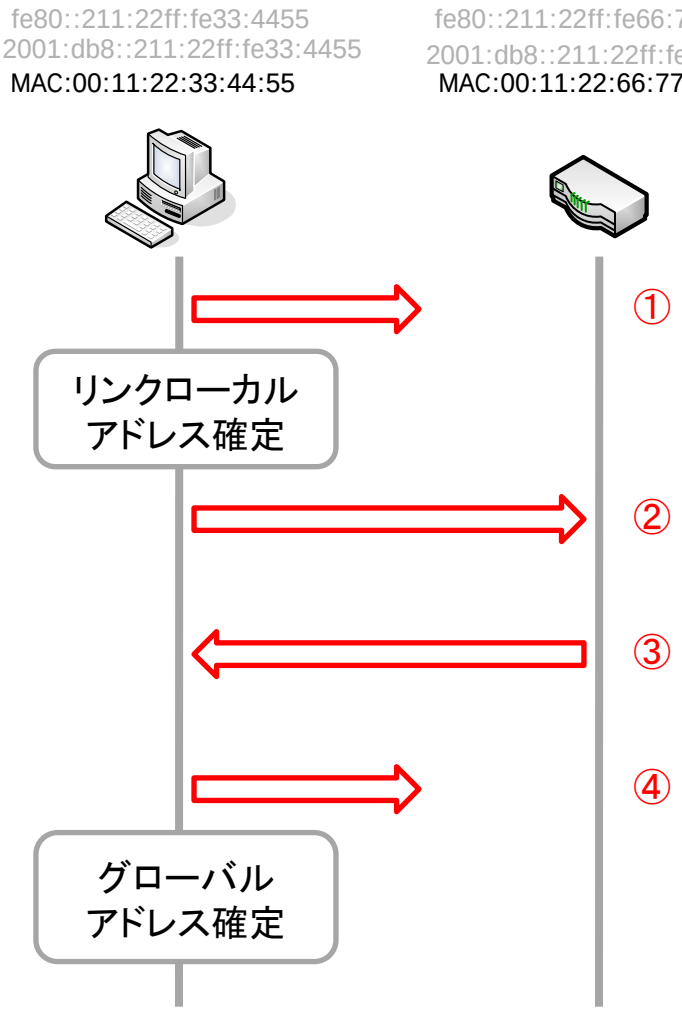
【参考】Router Solicitation/Router Advertisement

- RSの宛先アドレスはFF02::2、Hop Limitは255
- RAの宛先アドレスはFF02::1かRS内の始点アドレス、Hop Limitは255
- RA内のCurrent Hop Limitフィールドでノードが用いるホップ制限を設定
- M-flagが0ならステートレスアドレス自動設定、1ならDHCPv6によるアドレス設定
- O-flagが1ならアドレス以外の情報をDHCPv6により取得
- Router Lifetimeはデフォルトルータのみが1以上（65535以下）を指定
- DRP（Default Router Preference：RFC4191）によってデフォルトルータの優先度の通知が可能
  - High（01）、Medium（00）、Low（11）
  - ノード、ルータ双方がサポートしている必要がある



② Router Advertisement
Src=FE80::2
Dst=FF02::1 (or FE80::1)
Data=Subnet prefix, lifetime, current hop limit
Flag (management, other) , preference

【参考】ステートレス自動アドレス設定の流れ



①近隣要請(NS)
近隣広告がなければ
ターゲットアドレス
の利用が可能
＜重複アドレス検出＞

要請ノードマルチキャスト

Src MAC 00:11:22:33:44:55
Dst MAC 33:33:FF:33:44:55
Src IPv6 ::(未定義アドレス)
Dst IPv6 ff02::1:ff33:4455
ICMPv6 Type 135 Target fe80::211:22ff:fe33:4455

②ルータ要請(RS)
全ルータマルチキャスト
(ff02::2)宛に送信

Src MAC 00:11:22:33:44:55
Dst MAC 33:33:00:00:00:02
Src IPv6 fe80::211:22ff:fe33:4455
Dst IPv6 ff02::2
ICMPv6 Type 133

③ルータ広告(RA)
全ノードマルチキャスト
(ff02::1)宛に送信
取得プレフィックス
を用いてグローバル
アドレスを生成

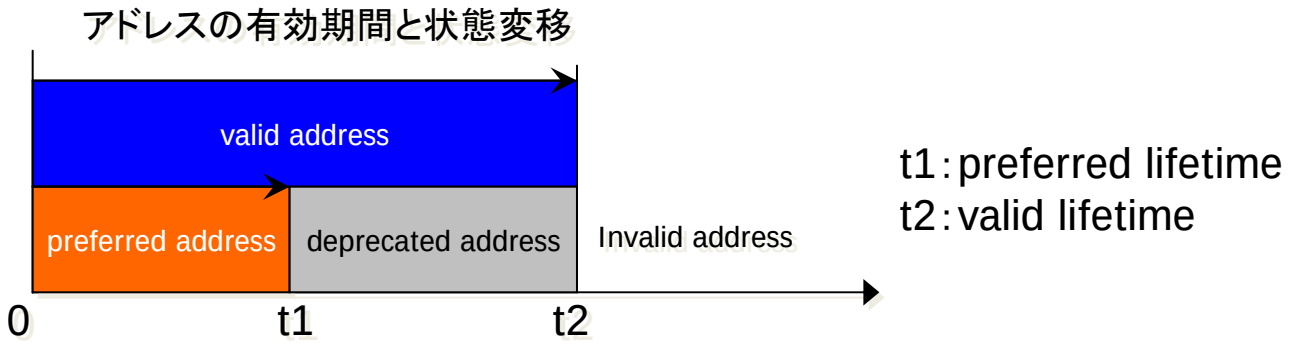
Src MAC 00:11:22:66:77:88
Dst MAC 33:33:00:00:00:01
Src IPv6 fe80::211:22ff:fe66:7788
Dst IPv6 ff02::1
ICMPv6 Type 134 Prefix 2001:db8::

④近隣要請
近隣広告がなければ
ターゲットアドレス
の利用が可能
応答があるとアドレス
を再構成する必要あり
＜重複アドレス検出＞

Src MAC 00:11:22:33:44:55
Dst MAC 33:33:FF:33:44:55
Src IPv6 ::(未定義アドレス)
Dst IPv6 ff02::1:ff33:4455
ICMPv6 Type 135 Target 2001:db8::211:22ff:fe33:4455

【参考】IPv6アドレスの状態、アドレスのlifetime

- tentative address
  - インタフェースに付与されていないアドレスでNDメッセージにしか使用できない。この時点でアドレスの一意性をDADで確認する。
- preferred address
  - インタフェースに付与されたアドレス。アドレスが一意で通信可能な状態
- deprecated address
  - 有効ではあるが、新規通信への使用をしないことが望まれる
- valid address
  - Preferredとdeprecatedのアドレスの双方を指す
- Invalid address
  - 有効アドレスの有効期間が過ぎるとこの無効アドレスになる



1. IPv6の基礎知識

1-3. アドレスの表記

- IPv4のアドレス表記
  - 例）192.0.2.1
    - 10進数で表した数字を“.”で区切って表記

- IPv6のアドレス表記
  - 例）2001:0db8:0000:0000:0206:29ff:fe1e:482e
    - 16進数で表した数字を“:”で区切って表記

1. IPv6の基礎知識 > 1-3. アドレスの表記

アドレスの略記

- 128bitを16bit毎に8分割後、各フィールドを16進数表記にして、":"(コロン)で区切る。

0010000000000001000011011011100000000000000000000000000000000000000000000000000100000011000101001111111111111111000011110010010000010111 0

↓

2001:0db8:0000:0000:0206:29ff:fe1e:482e

- 先行する**0**は省略可能。但し、各フィールドには少なくとも1つの数値を含むこと。

2001:0db8:0000:0000:0206:29ff:fe1e:482e

↓

2001:db8:0:0:206:29ff:fe1e:482e

- 16bitの0、または16bitの0が複数連続するフィールドを、1箇所のみ"**::**" を用いて省略可能。

2001:db8:0:0:206:29ff:fe1e:482e

↓

2001:db8::206:29ff:fe1e:482e

1. IPv6の基礎知識 > 1-3. アドレスの表記

アドレスの略記（Cont.）

- では、以下の場合は？

2001:0db8:0000:0000:fff0:0000:0000:000f

2001:db8::fff0:0:0:f

または

2001:db8:0:0:fff0::f

注）2001:db8::fff0::fとは省略できない

1. IPv6の基礎知識

1-4. アドレスの種類

- IPv6アドレスの分類(一例)
  - アドレスタイプ(挙動)
  - アドレススコープ(適用範囲)
  - 特殊なアドレス

1. IPv6の基礎知識 > 1-4. アドレスの種類

アドレスタイプ（挙動）

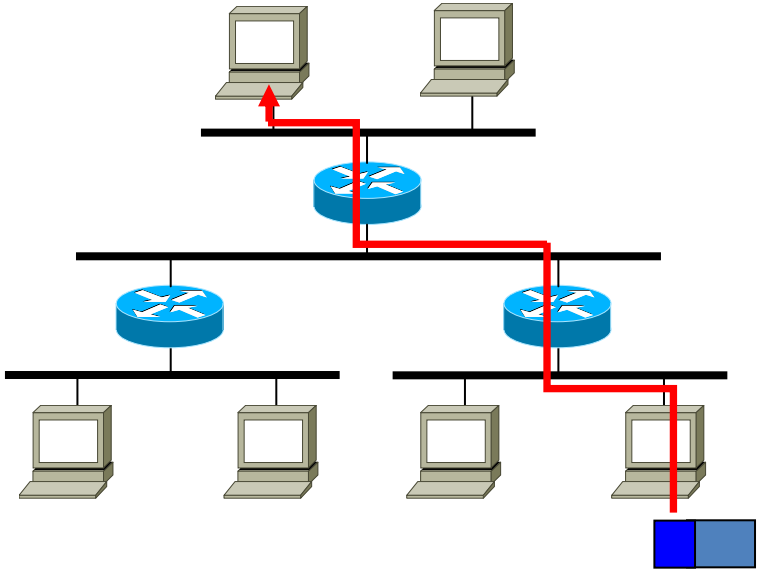
アドレスタイプ	付与対象	通信形態
Unicast	Interface	1：1
Multicast	Group	1：n
Anycast	Service	1：1 ※

※ネットワーク的に最も近い1つを選択

1. IPv6の基礎知識 > 1-4. アドレスの種類

アドレスタイプ(挙動)(Cont.)

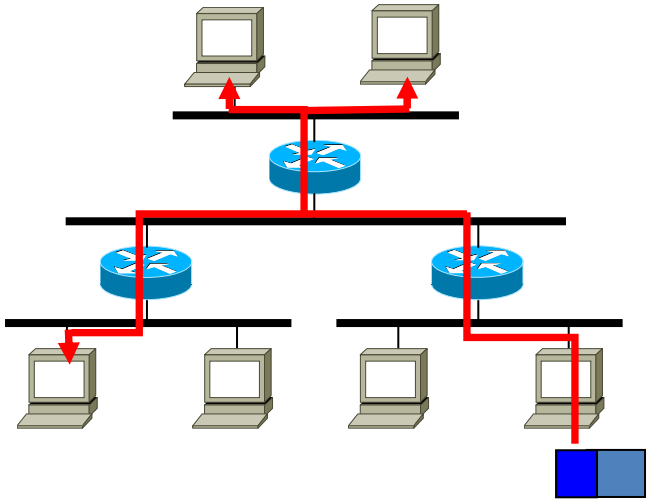
- ユニキャストアドレス
  - 単一のインタフェースに割り当てられるアドレス
  - 1対1の通信に使用される(普段はこのアドレスが使用される)



1. IPv6の基礎知識 > 1-4. アドレスの種類

アドレスタイプ(挙動)(Cont.)

- マルチキャストアドレス
  - あるグループを表すアドレス
    - あるマルチキャストアドレス宛てにパケットを投げると、そのグループに属するすべてのインタフェースに届けられる
    - IPv4におけるブロードキャストは、マルチキャストの1種として取り扱われる



1. IPv6の基礎知識 > 1-4. アドレスの種類

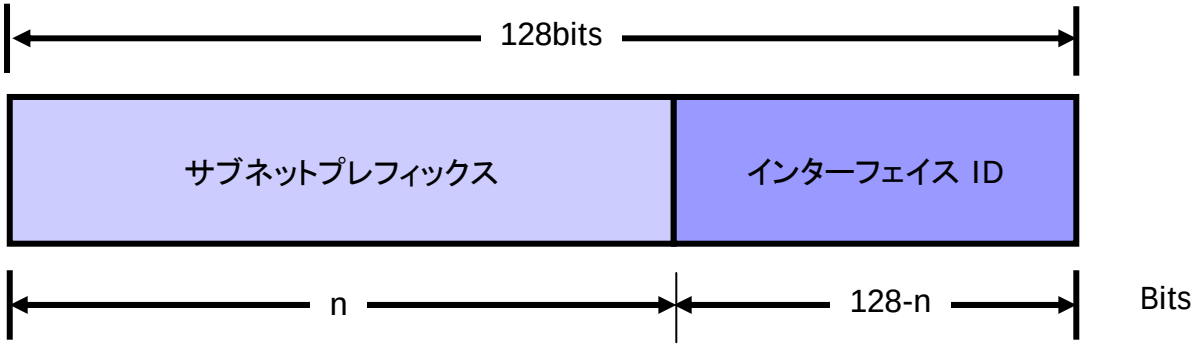
アドレススコープ（適用範囲）

- グローバルアドレス
- リンクローカルアドレス

1. IPv6の基礎知識 > 1-4. アドレスの種類

アドレススコープ（適用範囲）（Cont.）

- グローバルアドレス
  - IPv4でいうところのグローバルアドレスと同義
  - 全世界で一意に決まる識別子



アドレススコープ（適用範囲）（Cont.）

- リンクローカルアドレス
  - 同一リンク上でのみ通信可能（ルータを越える通信はできない）
  - 近隣にRAを投げるルータがなくても自動的に生成される
  - よく知られたリンクローカルアドレス
    - ff02::1 （マルチキャストアドレスでもある）
      - 同一リンク上のすべてのノードが参加している
    - かならずスコープIDを伴って利用する必要がある
      - 例： “ping6 –I eth0 ff03::1”, “ssh fe80::dead:beaf%eth0”

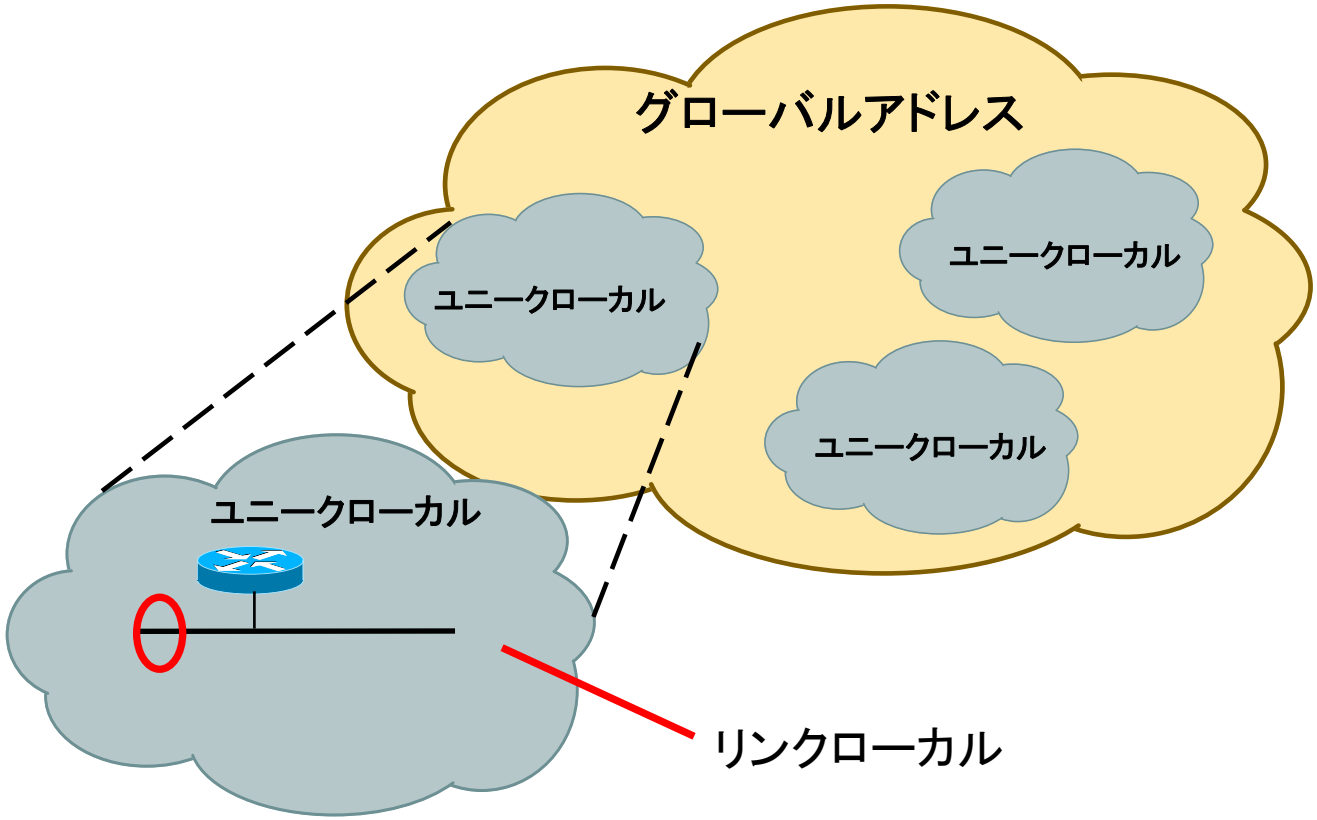
アドレススコープ（適用範囲）（Cont.）

- 【参考】ユニークローカルアドレス（ULA）
  - Globally unique prefix
  - Site boundariesで簡単にfilter
  - 外部（ISP等）との接続なしに使用できる
  - 万が一外部にアドレスを漏らしても、他と重複することがない
  - アプリケーションはグローバルアドレスと同様にこのアドレスを取り扱える

1. IPv6の基礎知識 > 1-4. アドレスの種類

アドレススコープ（適用範囲）（Cont.）

- 各アドレスの適用範囲



1. IPv6の基礎知識 > 1-4. アドレス体系

特殊なアドレス

- 未指定アドレス
  - アドレスが付けられてないことを示す
  - システムの初期化中でまだアドレスがついてないホストがソースアドレスとして使うことがある
  - 0000:0000:0000:0000:0000:0000:0000:0000 ⇒ ::
- ループバックアドレス
  - ノードがパケットを自分自身に送る場合に用いられる（IPv4でいうところの127.0.0.1）
  - 0000:0000:0000:0000:0000:0000:0000:0001 ⇒ ::1

1. IPv6の基礎知識 > 1-4. アドレス体系

特殊なアドレス（Cont.）

- IPv4射影アドレス
  - IPv6ノードが、IPv4しかサポートしていないノードと通信する際に使用するアドレス
  - 利便性を考え、IPv4アドレス部は10進数表記のままとされている

  例）“192.168.0.1” ⇒ “::ffff:192.168.0.1”

フィルタリングの設定に注意！

IPv6 onlyのアプリケーションがIPv4で接続されると、IPv4射影アドレスから接続されたように振る舞うため、フィルタリングの設定には、IPv6のアドレス、IPv4のアドレスの他、IPv4射影アドレスも書かないと、フィルタの対象にならない場合も。

※OS、アプリケーションの実装や設定による。

2. LinuxでIPv6を扱う

2-1. ネットワークの設定
2-2. ネットワーク疎通の確認
2-3. サーバ稼働状況の確認
2-4. サーバのIPv6設定

2. LinuxでIPv6を扱う

2-1. ネットワークの設定

- IPアドレスを設定するには、以下のコマンドが知られている
  - ifconfig
    - 書式： ifconfig *interface [aftype] options | address …*
  - ip
    - 書式：ip *[OPTIONS] OBJECT { COMMAND | help }*

2. LinuxでIPv6を扱う > 2-1. ネットワークの設定

IPアドレスの設定 -ifconfig-

- IPv4アドレスの設定例

```
IPv4 アドレスの設定
# ifconfig eth0 192.0.2.1 netmask 255.255.255.0

設定確認
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:XX:XX:XX:XX
          inet addr:192.0.2.1  Bcast:192.0.2.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:169

インタフェースを上げる
# ifconfig eth0 up

インタフェースを落とす
# ifconfig eth0 down
```


2. LinuxでIPv6を扱う > 2-1. ネットワークの設定

IPアドレスの設定 -ifconfig-（Cont.）

- IPv6アドレスの設定例

```
IPv6 アドレスの設定（事前にinterfaceを upしておく必要がある）
# ifconfig eth0 add 2001:db8::80/64

IPv6アドレスの削除
# ifconfig eth0 del 2001:db8::80/64

設定確認
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:XX:XX:XX:XX:XX
          inet addr:192.0.2.1  Bcast:192.0.2.255  Mask:255.255.255.0
          inet6 addr: 2001:db8::80/64 Scope:Global
               inet6 addr: fe80::2d0:xxxx:xxxx:xxxx/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:169

インタフェースを落とす(IPv6の場合は、落とすとグローバルアドレスが消える）
# ifconfig eth0 down
```


2. LinuxでIPv6を扱う > 2-1. ネットワークの設定

IPアドレスの設定 -ip-

- IPv4/v6アドレスの設定例

```
IPv4/IPv6アドレスの設定
# ip addr add 192.0.2.1/24 dev eth0
# ip addr add 2001:db8::80/64 dev eth0

設定確認
# ip addr show dev eth0
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:xx:xx:xx:xx:xx brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.0.2.255 scope global eth0
    inet6 fe80::202:xxx:xxx:xxx:xxx/64 scope link
       valid_lft forever preferred_lft forever
    inet6 2001:db8::80/64 scope global
       valid_lft forever preferred_lft forever

インタフェースを上げる
# ip link set eth0 up

インタフェースを落とす
# ip link set eth0 down
```

2. LinuxでIPv6を扱う > 2-1. ネットワークの設定

IPアドレスの設定 -ip-（Cont.）

- IPv4/v6 L2アドレスの確認

```
IPv4の場合：ARPコマンドの代わりに
# ip -4 neigh show
192.0.2.1 dev eth0 lladdr 00:00:XX:00:XX:XX REACHABLE

IPv6の場合：
# ip -6 neigh show
2001:db8:0:1::dead:beaf dev eth0 lladdr 00:00:XX:XX:XX:XX router REACHABLE
fe80::XXX:XXXX:XXXX:XXXX dev eth0 lladdr 00:00:XX:XX:XX:XX router REACHABLE
```

2. LinuxでIPv6を扱う > 2-1. ネットワークの設定

経路の設定・確認 -route-

- routeコマンド

```
Default経路の設定
# route add -A inet default gw 192.0.2.254 dev eth0
# route add -A inet6 default gw fe80:x:x:x:x:x dev eth0

ネットワーク別経路の設定
# route add -net 192.0.2.0 netmask 255.255.255.0 gw 192.168.0.1 dev eth0
# route add -A inet6 2001:db8::/64 gw fe80::2d0:b7ff:fea0:beea dev eth0

経路の確認
# route -n -A inet6
# route -n -A inet
```

2. LinuxでIPv6を扱う > 2-1. ネットワークの設定

経路の設定・確認 -ip-

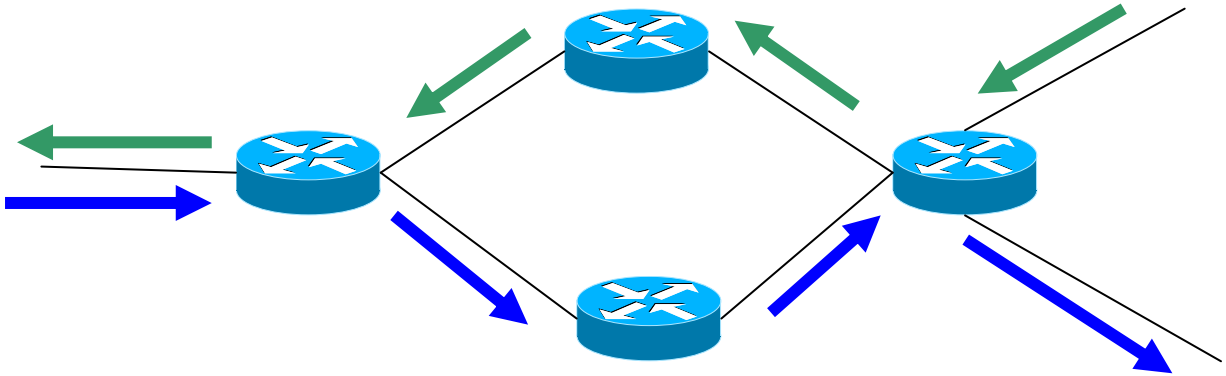
- ipコマンド

```
Default経路の設定
# ip route add default via 10.0.0.1 dev eth0
# ip route add default via fe80::202:b3ff:fe32:faa2 dev eth0

ネットワーク別経路の設定
# ip route add 192.0.2.0/24 via 10.0.0.1 dev eth0
# ip route add 2001:db8::/48 via fe80::202:b3ff:fe32:faa2 dev eth0

経路の確認
# ip -4 route show
# ip -6 route show
```

2. LinuxでIPv6を扱う

2-2. ネットワーク疎通の確認

- 簡単な接続確認方法
  - pingを用いた疎通確認
  - tracerouteを用いた疎通確認
    - tracerouteでわかるのは行きの経路だけ。



行きと帰りで経路が違うことも……

2. LinuxでIPv6を扱う > 2-2. ネットワーク疎通の確認

ICMPによる疎通確認 -ping-

- Default gatewayに対するping（IPv4）

```
IPv4経路の疎通性確認
$ ping -c 10 192.0. 2.254
PING 192.0.2.254 (192.0.2.254) 56(84) bytes of data.
64 bytes from 192.0.2.254: icmp_seq=1 ttl=255 time=2.09 ms
64 bytes from 192.0.2.254: icmp_seq=2 ttl=255 time=2.04 ms
64 bytes from 192.0.2.254: icmp_seq=3 ttl=255 time=4.30 ms
64 bytes from 192.0.2.254: icmp_seq=4 ttl=255 time=2.00 ms
64 bytes from 192.0.2.254: icmp_seq=5 ttl=255 time=2.01 ms
64 bytes from 192.0.2.254: icmp_seq=6 ttl=255 time=2.02 ms
64 bytes from 192.0.2.254: icmp_seq=7 ttl=255 time=2.03 ms
64 bytes from 192.0.2.254: icmp_seq=8 ttl=255 time=2.62 ms
64 bytes from 192.0.2.254: icmp_seq=9 ttl=255 time=3.84 ms
64 bytes from 192.0.2.254: icmp_seq=10 ttl=255 time=4.77 ms

--- 192.0.2.254 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9036ms
rtt min/avg/max/mdev = 2.009/2.776/4.774/1.038 ms
```


2. LinuxでIPv6を扱う > 2-2. ネットワーク疎通の確認

ICMPによる疎通確認 -ping-（Cont.）

- Default gatewayに対するping（IPv6）

```
IPv6経路の疎通性を確認
$ ping6 -c 10 fe80::2000:1 -I eth0
PING fe80::2000:1(fe80::2000:1) from fe80::2d0:b7ff:fea0:beea eth1: 56 data bytes
64 bytes from fe80::2000:1: icmp_seq=1 ttl=64 time=2.38 ms
64 bytes from fe80::2000:1: icmp_seq=2 ttl=64 time=7.71 ms
64 bytes from fe80::2000:1: icmp_seq=3 ttl=64 time=7.47 ms
64 bytes from fe80::2000:1: icmp_seq=4 ttl=64 time=2.41 ms
64 bytes from fe80::2000:1: icmp_seq=5 ttl=64 time=2.39 ms
64 bytes from fe80::2000:1: icmp_seq=6 ttl=64 time=3.91 ms
64 bytes from fe80::2000:1: icmp_seq=7 ttl=64 time=5.00 ms
64 bytes from fe80::2000:1: icmp_seq=8 ttl=64 time=2.29 ms
64 bytes from fe80::2000:1: icmp_seq=9 ttl=64 time=2.30 ms
64 bytes from fe80::2000:1: icmp_seq=10 ttl=64 time=2.32 ms

--- fe80::2000:1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9036ms
rtt min/avg/max/mdev = 2.292/3.822/7.711/2.069 ms
```


2. LinuxでIPv6を扱う

2-3. サーバ稼働状況の確認

- どんなプログラムが起動しているのか？
  - psコマンド

```
# ps aux | less
```

- どんなポートをListenしているのか？
  - netstat
  - fuser

2. LinuxでIPv6を扱う > 2-3. サーバ稼働状況の確認

netstat

- netstatを用いてサーバのListenポートを表示させる

【オプション例】
-l, --listening
　接続待ち状態にあるソケットのみを表示する
-p, --program
　各ソケットが属しているプログラムのPIDと名前が表示される
-n, --numeric
　ホスト、ポート、ユーザなどの名前を解決せずに、数字のアドレスで表示する
-t,--tcp
　tcpに関する情報を表示
-u,--udp
　udpに関する情報を表示

2. LinuxでIPv6を扱う > 2-3. サーバ稼働状況の確認

netstat（Cont.）

- 実行例

netstatでは、表示桁数の制限から、一定長以上のIPv6アドレスは表示が省略される

```
# netstat -ltupn
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:993           0.0.0.0:*               LISTEN      3785/famd
tcp        0      0 127.0.0.1:111           0.0.0.0:*               LISTEN      3175/portmap
tcp        0      0 192.0.2.1:53            0.0.0.0:*               LISTEN      3380/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      3380/named
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      3619/postmaster
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      3596/master
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      3380/named
tcp6       0      0 :::80                   :::*                    LISTEN      11254/apache2
tcp6       0      0 :::53                   :::*                    LISTEN      3380/named
tcp6       0      0 :::22                   :::*                    LISTEN      3659/sshd
tcp6       0      0 :::5432                 :::*                    LISTEN      3619/postmaster
udp        0      0 0.0.0.0:32768           0.0.0.0:*                           3380/named
udp        0      0 127.0.0.1:161           0.0.0.0:*                           3653/snmpd
udp        0      0 192.0.2.1:53            0.0.0.0:*                           3380/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                           3380/named
udp        0      0 127.0.0.1:111           0.0.0.0:*                           3175/portmap
udp6       0      0 :::32769                :::*                                3380/named
udp6       0      0 :::53                   :::*                                3380/named
```

※"-A"でアドレスファミリーを指定することも可能（inet or inet6）

2. LinuxでIPv6を扱う > 2-3. サーバ稼働状況の確認

SS

- 引数はほぼ netstatと同様
- IPv6アドレスが省略されることはない

```
$ ss -ltun
Netid  Recv-Q Send-Q          Local Address:Port            Peer Address:Port
tcp    0      50                 127.0.0.1:3306                 *:*
tcp    0      128                        *:111                  *:*
tcp    0      128                       :::80                  :::*
tcp    0      5                  127.0.0.1:33843                *:*
tcp    0      3                 192.0.2.136:53                 *:*
tcp    0      3                  127.0.0.1:53                   *:*
tcp    0      3                         :::53                  :::*
tcp    0      128                127.0.0.1:631                  *:*
tcp    0      128                127.0.0.1:5432                 *:*
tcp    0      128                      ::1:953                  :::*
tcp    0      128                127.0.0.1:953                  *:*
tcp    0      100                       :::25                  :::*
tcp    0      100                        *:25                   *:*
tcp    0      3                          *:1723                 *:*
```


2. LinuxでIPv6を扱う > 2-3. サーバ稼働状況の確認

fuser

- fuserを用いてサーバのListenポートからListenしているプロセスを特定する

```
# fuser -vn tcp 80

                     USER        PID ACCESS COMMAND
80/tcp:              root       4699 F.... apache
                     www-data   4706 F.... apache
                     www-data   4707 F.... apache
                     www-data   4708 F.... apache
                     www-data   4709 F.... apache
                     www-data   4710 F.... apache
                     www-data   8407 F.... apache
                     www-data   8408 F.... apache
                     www-data   8409 F.... apache
```


2. LinuxでIPv6を扱う

2-4. サーバのIPv6設定

- DNSサーバ - BIND9
- SMTPサーバ - Postfix
- POPサーバ - Dovecot
- HTTPサーバ - Apache
- NTPサーバ

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

DNSサーバ - BIND9

- 「DNSのIPv6対応」には、2つの意味がある
  - 保持できるRRの対応
    - AAAAを返す
      - 逆引きは、同じくPTRだが、”in-addr.arpa”に対して、“ipv6.arpa”を利用。ipv6.arpaではなく、ipv6.int が使われていたこともあったが、今は使われない
    - bind8.4 以上またはbind9で対応
  - IPv6 transportへの対応
    - クエリの送受信をIPv6プロトコルを用いて行う
    - 自身のFQDNに、AAAAが登録される

※AAAAは、IPv4ネットワークを通じて、返信することも可能なので、DNSサーバがIPv6に対応しました！というのは上記のどちらの話なのか（もしくは両方なのか）、に留意する必要がある

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

DNSサーバ - BIND9（Cont.）

- IPv6を利用可能にするには
  - IPv6 transportを有効にする（listen-on-v6設定）

```
options {
            directory "/var/named";
            listen-on-v6 { any; };
...
```

  - AAAAレコードを記述する

AAAA RR登録例

```
;; Server
;;   example.jp
www        IN         A          192.0.2.1
www        IN         AAAA       2001:db8::1
```

AAAA RR確認

```
$ dig www.example.jp AAAA
```

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

DNSサーバ - BIND9（Cont.）

- IPv6を利用可能にするには（Cont.）
  - IPv6の逆引き設定をする

逆引き設定例

```
;; IPv6 PTR for 2001:db8::/64
zone “0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ipv6.arpa” {
          type master;
          file “2001.0db8.0000.000.reverse”;
          allow-transfer { slaves; };
          allow-query { any };
}
```

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN   PTR   www.example.jp.
```

逆引き設定の確認

```
$ dig www.example.jp AAAA
```

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

DNSサーバ - BIND9（Cont.）

- IPv6を利用可能にするには（Cont.）
  - ACLの設定をする
    - IPv4と同様に、生アドレスが記載可能

```
acl “slaves” {
          192.0.2.1;          // slave server
          2001:db8::53;       // slave server
          127.0.0.1;          // for debug
          ::1;                // for debug
};
```

  - リゾルバの設定
    - IPv4と同様、/etc/resolv.confに直接、IPv6アドレスを記述する

```
search example.jp
nameserver 192.0.2.254
nameserver 2001:db8::53
```


2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

SMTPサーバ – Postfix

- Postfixは、2.2からIPv6に対応している
- IPv6を利用可能にするには
  - IPv6を扱う設定を記述

/etc/postfix/main.cf

```
# inet_protocols = ipv4
# inet_protocols = ipv4, ipv6 # allと等価です
# inet_protocols = ipv6
inet_protocols = all
```

これだけ。。。

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

SMTPサーバ – Postfix（Cont.）

- IPv6を利用可能にするには（Cont.）
  - Listenするアドレスを制限

/etc/postfix/main.cf

```
# inet_interfaces = alll
# inet_interfaces = loopback-only
inet_interfaces = 127.0.0.1, [::1], [2001:db8::25]
```

  - 送信時のアドレスを固定

/etc/postfix/main.cf

```
smtp_bind_address6 = 2001:db8::25
```

※master.cfでも利用可能です

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

SMTPサーバ – Postfix（Cont.）

- IPv6の設定（表記）方法についての注意
  - “:”があると区別がつかない項目では[]を付け、それ以外では付けないというのが基本方針
    - mynetworksやdebug_peer_listのように、Postfixマッチリストを設定する項目では、“type:table”形式と混乱しないためにも、IPv6アドレスを[]で囲う必要がある

```
# mynetworks = hash:/etc/postfix/network_table
mynetworks = 127.0.0.0/8 [::1]/128
```

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

POPサーバ - Dovecot

- Dovecotは、IPv6に対応している
- IPv6を利用するための特別な設定は不要。CentOS5系に付属のDovecotでは、Listenアドレスの指定に関して、あまり複雑な指定はできない

/etc/dovecot.conf

```
# “*”を指定すると、すべてのインタフェースのIPv4アドレスを
# Listenする
#Listen = *
#
# “[::]”を指定すると、すべてのインタフェースのIPv6アドレスを
# Listenする。OSによっては、すべてのインタフェースのIPv4も
# Listenする。
#Listen = [::]
```

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

HTTPサーバ - Apache

- Apacheは、2.0からIPv6に対応している
- IPv6を利用するための特別な設定は不要

- IPv6関連設定の注意点は、アドレス表記
  - IPv4の場合と異なり、[]で括る必要がある場合も

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

HTTPサーバ - Apache（Cont.）

- Listen

```
Listen [2001:db8::a00:20ff:fea7:ccea]:80
```

- ACL / アドレスによるアクセス制限
  - 2001:db8:0:1000/64とはできないので注意。きちんとネットワークアドレスを指定する必要がある

```
AuthName “Staff Only”
AuthType Basic
AuthUserFile “/var/www/www.example.jp/.htpasswd”
Require valid-user
Order Deny,Allow
Deny from all
Allow from 192.168.1.1
Allow from 2001:db8:0:1000::/64
Satisfy Any
```

2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

HTTPサーバ - Apache（Cont.）

- アドレスベースのVirtualHost
  - Listenと同様、IPv4の場合と異なり、IPv6アドレスは[]で括る必要がある

```
#<VirtualHost *:80>
<VirtualHost [2001:db8:0:1000::80]:80>
   ServerName www.example.co.jp
  …
  ..
  .
</VirtualHost>
```


2. LinuxでIPv6を扱う > 2-4. サーバのIPv6設定

NTPサーバ

- 上位NTPサーバを指定する
- FQDNおよびIPv6アドレスでの指定が可能

```
server     ntp1.v6.mfeed.ad.jp
server     2001:3a0:0:2005::57:123
```

3.【参考】IPv6運用の留意点

3-1. DNS逆引きについて
3-2. ログ形式について
3-3. アドレス自動設定について
3-4. 障害の切り分け（FQDN）
3-5. IPv6の接続性を得るには

3.【参考】IPv6運用の留意点

3-1. DNS逆引きについて

- クライアントのアドレスは、プライバシー拡張などを用いてアクセスされることなどを考えると、DNSの逆引きによるACLは、事実上使えないと思ったほうがよい

3.【参考】IPv6運用の留意点

3-2. ログ形式について

- syslogなどで記録されるアドレスは、決まった短縮法が適用されるわけではなく、吐き出すアプリケーションに依存する
- このため、grep などで、単純にアドレスをマッチさせることはできないことがある
- 問題の発生を減らすために代表的な表記方法がIETF の 6man WG で議論されている。
  - A Recommendation for IPv6 Address Text Representation [draft-ietf-6man-text-addr-representation-07] (work in progress)
    - RFC5952に発行（2010年8月）

3.【参考】IPv6運用の留意点

3-3. アドレス自動設定について

- 多くのクライアントノードを管理する管理者の作業低減の目的があった
  - しかしながら、サーバ用途では、NICの交換によって、アドレスが変わってしまうこともあるので、自動設定は避けたほうがよい
    - アドレス変更にともなう、フィルタのルール変更なども必要なため

3.【参考】IPv6運用の留意点

3-4. 障害の切り分け(FQDN)

- 監視ツールで、FQDNでノードを登録していた場合、IPv6/IPv4のフォールバックの影響を受けないか注意が必要

3.【参考】IPv6運用の留意点

3-6. IPv6の接続性を得るには

- 上流ISPからIPv6のトランジットを買う
  - 意外とあります
- IPv6対応のiDCに入る
  - 意外とあります
- Tunnel Brokerを探す
  - Feel6（dtcp）
  - OCN IPv6（L2TP）
    http://www.ocn.ne.jp/ipv6/