

# IPv6概説と導入事例

Matsuzaki 'maz' Yoshinobu

<maz@iij.ad.jp>

# IPv4アドレスの在庫枯渇

- 接続機器の増加
  - インターネットが爆発的に普及
  - だってみんな使っちゃってるでしょ？
- 接続形態の変化
  - ブロードバンドで常時接続とか
  - 利用していないときでも、IPアドレスを保持
  - ダイヤルアップサービスに比べて、ブロードバンドサービスでは10倍程度のIPアドレス数が必要

# なんかやりくりする？

- ISPで巨大なNAT
  - 既存の個人向けISPは延命できる
  - しかもユーザには不利益
- 使っていないIPv4アドレスを回収
  - 回収されたアドレスもある。15.0.0.0/8とか
  - 現在の消費スピードだと /8 は1か月程度
- IPアドレスを売買などで流通させる
  - 買ったアドレスは使えるでしょうか

# じゃあ、やっぱりIPv6

- IPv6に移行すれば枯渇問題は当面解決
  - 今のところ唯一の根本的解決
- ただし道のりは楽じゃない
  - 様々な移行コスト
  - それでも、インターネットが発展を続けるためにはもっとアドレスが必要

# これから起こりうる変化

- 新規IPv4アドレスの割り当てが困難に
- NAPTがより多用される
  - 多段NAPTや双方向NAPTによる複雑化
- IPv4アドレスの取引、登記の書き換え
  - IPv4アドレス価格の上昇
- ルーティングテーブルの肥大
  - 利用率を上げるための細かい経路広報
- 世界的にIPv4の運用コストが上がっていく

# 進む道

- きっといつかはIPv6なインターネット
  - 短期的にはいろいろ手段が利用されるだろうけど
    - 様々なNAT
    - application gateway
    - 未使用IPv4アドレスの取引
  - 長期的にはIPv6に向かうと考えられる
    - 今のところ他に代案が無い
    - 今から準備して間に合うのがIPv6しかない



# というわけで、IPv6来ちゃうよ

- 徐々に広がるIPv6の実装
  - BSD、Linux、WindowsとかMac OSとか
  - ルータとか諸々の機器
  - 知らないうちにIPv6対応の機器がそこらに
- サービスのIPv6対応
  - いくつかのwebサイトなどなど
  - 知らないうちにIPv6対応のサービスがそこらに

# IPv4とIPv6

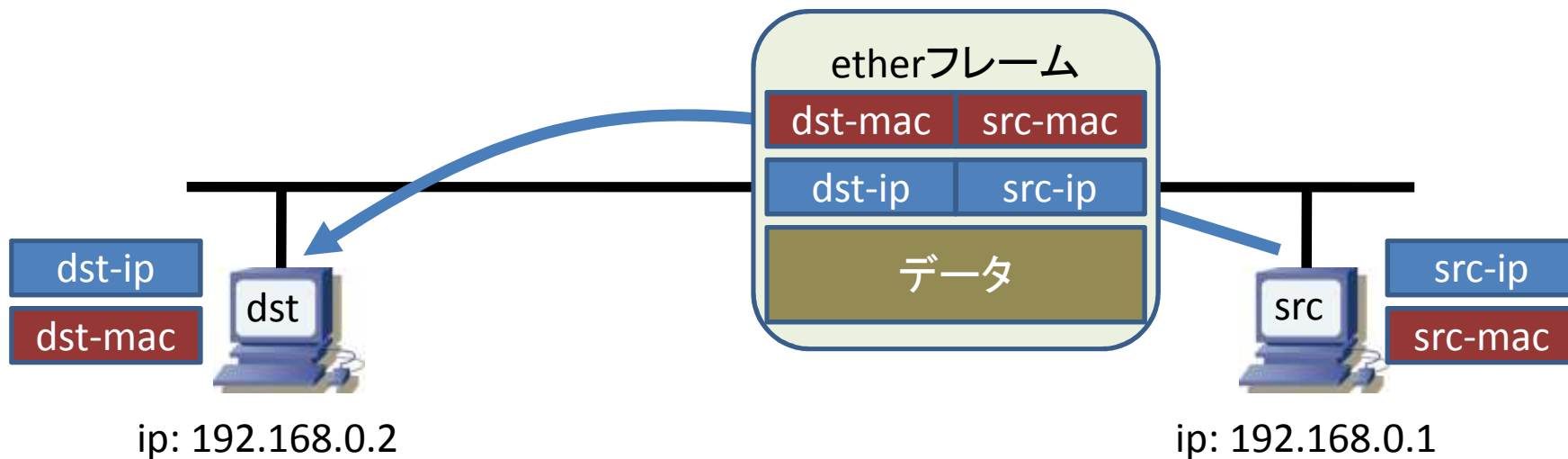
- 基本はほとんど一緒
  - パケット転送などの考え方は一緒
  - IPヘッダがちょっと違う
  - アドレス長が伸びてる
    - IPv4 32bit長 → IPv6 128bit長



# IPv4パケット送信

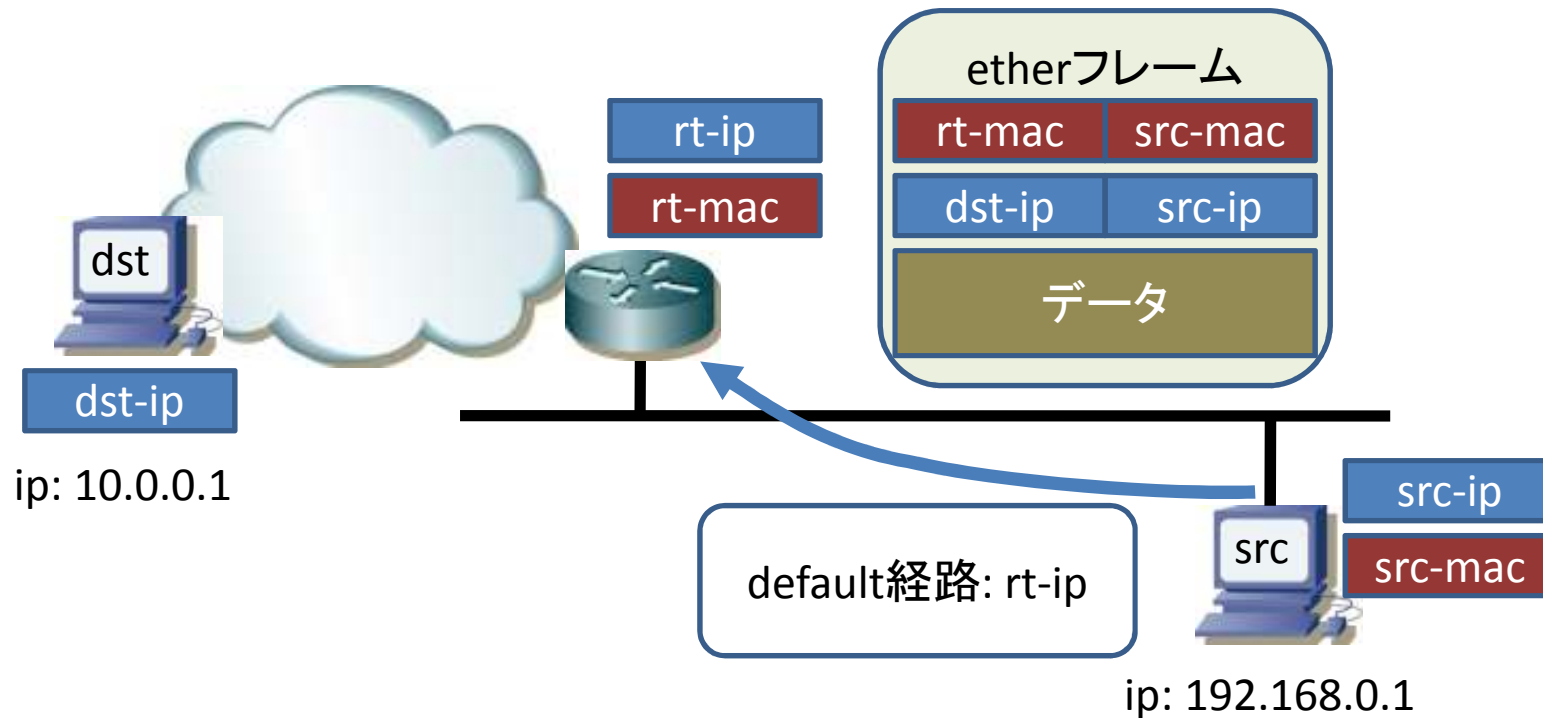
- 同じネットワークに属していれば直接送信

inet 192.168.0.1 netmask 255.255.255.0  
↓  
192.168.0.0~192.168.0.255が同じセグメント上にある



# IPv4パケット送信 2

- 遠くには経路情報に従ってルータに投げる



# arp (Address Resolution Protocol)

- etherではパケット送信にMACアドレスが必要
  - IPv4アドレスは分かってる (ex. defaultの向け先)
  - 機器のIPv4アドレスからMACアドレスを知りたい

- arpで解決

- RFC826

```
arp who-has 192.168.0.2 tell 192.168.0.1
0x0000:  ffff ffff ffff 0019 bb27 37e0 0806 0001
0x0010:  0800 0604 0001 0019 bb27 37e0 c0a8 0001
0x0020:  0000 0000 0000 c0a8 0002
arp reply 192.168.0.2 is-at 00:16:17:61:64:86
0x0000:  0019 bb27 37e0 0016 1761 6486 0806 0001
0x0010:  0800 0604 0002 0016 1761 6486 c0a8 0002
0x0020:  0019 bb27 37e0 c0a8 0001 0000 0000 0000
0x0030:  0000 0000 0000 0000 0000 0000
```

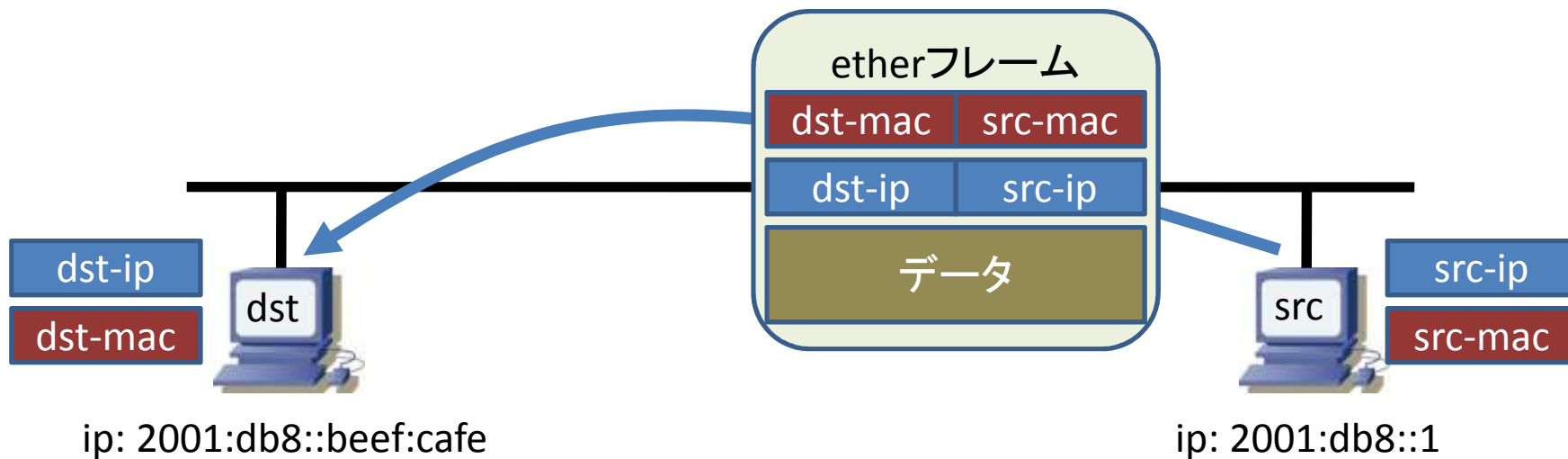
# IPv6パケット送信

- 同じネットワークに属していれば直接送信

inet6 2001:db8::1 prefixlen 64

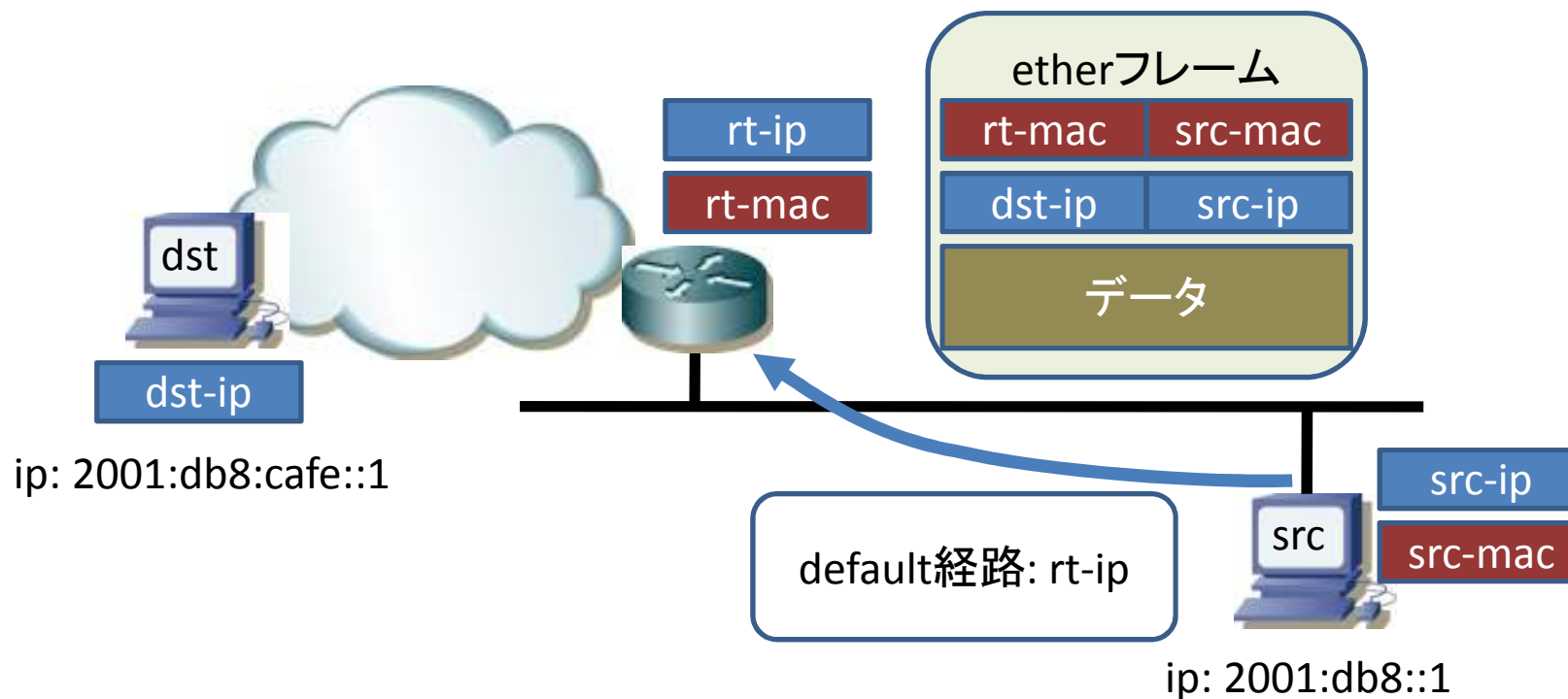


2001:db8::~2001:db8::ffff:ffff:ffff:ffffが同じセグメント上にある



# IPv6パケット送信 2

- 遠くには経路情報に従ってルータに投げる



# ndp (Neighbor Discovery Protocol)

- etherではパケット送信にMACアドレスが必要
  - 機器のIPv6アドレスからMACアドレスを知りたい
- ndpで解決
  - RFC4861
    - ICMPv6を利用してMACアドレスを問い合わせる
  - 送り先を未学習ならmulticastアドレス宛て
    - IP: ff02::1:ff00:0000 ~ ff02::1:ffff:ffff
      - 送信先IPアドレスの下位24bitを利用して生成
    - MAC: 33:33:00:00:00:00 ~ 33:33:ff:ff:ff:ff
      - 送信先IPアドレスの下位32bitを利用して生成

# ndpでMACアドレス解決

```
IP6 2001:db8::1 > ff02::1:ffef:cafe
```

```
ICMP6, neighbor solicitation, who has 2001:db8::beef:cafe  
source link-address option: 00:19:bb:27:37:e0
```

```
0x0000: 3333 ffef cafe 0019 bb27 37e0 86dd 6000  
0x0010: 0000 0020 3aff 2001 0db8 0000 0000 0000  
0x0020: 0000 0000 0001 ff02 0000 0000 0000 0000  
0x0030: 0001 ffef cafe 8700 9a90 0000 0000 2001  
0x0040: 0db8 0000 0000 0000 0000 0000 beef cafe 0101  
0x0050: 0019 bb27 37e0
```

```
IP6 2001:db8::beef:cafe > 2001:db8::1
```

```
ICMP6, neighbor advertisement, tgt is 2001:db8::beef:cafe  
destination link-address option: 00:16:17:61:64:86
```

```
0x0000: 0019 bb27 37e0 0016 1761 6486 86dd 6000  
0x0010: 0000 0020 3aff 2001 0db8 0000 0000 0000  
0x0020: 0000 beef cafe 2001 0db8 0000 0000 0000  
0x0030: 0000 0000 0001 8800 c1fd 6000 0000 2001  
0x0040: 0db8 0000 0000 0000 0000 0000 beef cafe 0201  
0x0050: 0016 1761 6486
```

# IPv6概説



# IPv6の特徴

- アドレスいっぱい
- インタフェースに複数のアドレスが付く
  - 通信元IPアドレスを選択する仕組みがある
- スコープ(有効範囲)の概念がある
  - リンクローカルやグローバル
- Path MTU Discoveryの実装

# IPv6アドレス表記

- 128bit長を16bit毎に16進数表記、「:」で繋ぐ
- 2001:0db8:0000:0000:0000:0000:0000:0001
  - 先頭の0を省略 2001:db8:0:0:0:0:0:1
  - 連続の0を圧縮 2001:db8::1
    - ただし、::は一か所だけ (ex: 2001:db8::1:0:1)
- ネットワークを表す prefix表記は、IPv4と同じく「/」の後にprefix長を10進数で記述
  - 2001:db8::/64

# IPv6アドレスの種類

- unicast (ユニキャスト)
  - 機器に付けられた識別子
  - このアドレスを持つ機器がパケットを受信
- multicast (マルチキャスト)
  - 複数の機器で共用する識別子
  - このアドレスを持つ全機器がパケットを受信
- anycast (エニーキャスト)
  - 複数の機器で共用する識別子
  - IPv4のエニーキャストと同等

# ユニキャスト・アドレス

- リンクローカル(fe80::/10)
  - リンク内のみで有効
  - 今のところ fe80::/64 のみを使用
- グローバル・ユニキャスト
  - 特別じゃないユニキャスト・アドレス全般
- ユニークローカル(fd00::/8)
  - サイトローカルを置き換える目的で策定
  - グローバル・ユニキャストを使う方が安全

# リンクローカルアドレス

- リンク内でのみ有効
  - fe80::/64を利用して各インタフェースに設定
  - ルータは、リンクローカルを宛先または送信元を持つパケットを転送してはならない(must not)
    - RFC4291
- ノードの自動設定や経路制御等に利用
  - ルータ間はリンクローカルのみで経路制御できる
  - グローバルアドレスが変わっても経路の向け先は変わらない

# マルチキャスト・アドレス

- 複数のノードが受信するかも
- リンク内の全IPv6ノード
  - ff02::1
- 要請ノード・マルチキャスト・アドレス
  - ff02::1:ff00:0000/104
  - ユニキャストアドレスの下位24bitを利用して生成
    - 例) 2001:db8::1234:5678 → ff02::1:ff34:5678

# 特別なアドレスとかとか

2001:db8::/32

- 文書記述用アドレス

::

- 未指定アドレス

:::1

- ループバックアドレス

ff02::1

- 全IPv6ノード

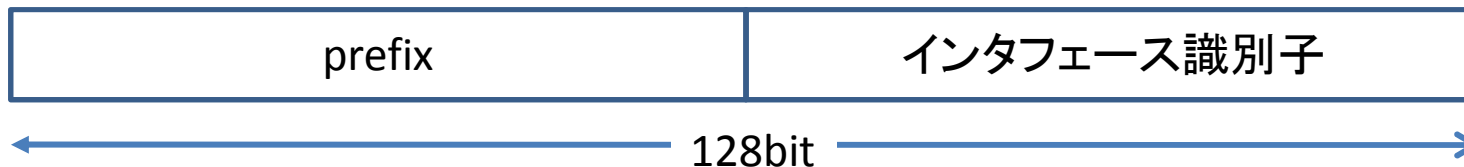
ff02::2

- 全IPv6ルータ

# IPv6のアドレス構造

- 128bitのアドレス空間
- 通常利用する場合、/64が基本
  - 前半64bitがprefix
  - 後半64bitがインタフェース識別子

2001:0db8:0000:0000:0000:0000:0000:0001





# インタフェース識別子

- リンク上でインタフェースを識別するID
  - リンク内で一意
- いくつか自動生成の方法も考えられている
  - MACアドレスから生成するEUI-64っぽいアドレス

例) MACアドレス - 00:19:bb:27:37:e0

1. 3バイトずつに分割
2. ff:feを間に挟む
3. universal/localビットを反転(EUI-64と異なる)

EUI-64形式アドレス - 0219:bbff:fe27:37e0

- ランダムに64bit長のアドレスを生成

# 「挿したら動く」ために

- IPv6アドレス=prefix + インタフェース識別子
  - インタフェース識別子は自動で設定できる
  - あとはprefix
    - リンクローカル(fe80::/64)は生成できる
- リンクローカルアドレスの生成
  - 例えばMACアドレス - 00:19:bb:27:37:e0
    - EUI-64形式 - 0219:bbff:fe27:37e0
    - 仮のIPv6アドレス – fe80::0219:bbff:fe27:37e0

# アドレス衝突検出 (DAD)

- 利用前にndpで衝突が無いことを確認
  - 送信元アドレス「::」
  - 送信先アドレス「要請ノード・マルチキャスト」

```
IPv6 :: > ff02::1:ff27:37e0
ICMP6, neighbor solicitation,
who has fe80::219:bbff:fe27:37e0
 0x0000:  3333 ff27 37e0 0019 bb27 37e0 86dd 6000
 0x0010:  0000 0018 3aff 0000 0000 0000 0000 0000
 0x0020:  0000 0000 0000 ff02 0000 0000 0000 0000
 0x0030:  0001 ff00 0001 8700 5764 0000 0000 fe80
 0x0040:  0000 0000 0000 0219 bbff fe27 37e0
```

# ルータ広告(RA)

- ルータが定期的にICMP6で通知
  - 必要であれば、ホストがルータ要請で広告を依頼

```
IP6 fe80::20b:fdff:fed5:e17f > ff02::1
ICMP6, router advertisement,
  prefix info option: 2001:db8::/64,
  valid time 2592000s
0x0000:  3333 0000 0001 000b fdd5 e17f 86dd 6e00
0x0010:  0000 0040 3aff fe80 0000 0000 0000 020b
0x0020:  fdff fed5 e17f ff02 0000 0000 0000 0000
0x0030:  0000 0000 0001 8600 3aa1 4000 0708 0000
0x0040:  0000 0000 0000 0101 000b fdd5 e17f 0501
0x0050:  0000 0000 05dc 0304 40c0 0027 8d00 0009
0x0060:  3a80 0000 0000 2001 0db8 0000 0000 0000
0x0070:  0000 0000 0000
```

# ルータ広告で分かること

- デフォルト経路を向けるべきルータ
- セグメントのprefixとその有効期間
- つまり、ホストがIPv6で通信できるようになる、  
最小限の情報がわかる
  - 参照用DNSをどう通知するかは議論中。。。
    - DHCPv6を使う(一番有望)
    - RAに新たなオプションを用意する
    - IPv4でDNSアドレスが取得できてるなら、とりあえずそれを利用する

# ホストのアドレス自動設定

## 1. リンクローカルアドレスの生成

例) fe80::219:bbff:fe27:37e0

## 2. アドレス衝突検出(DAD)

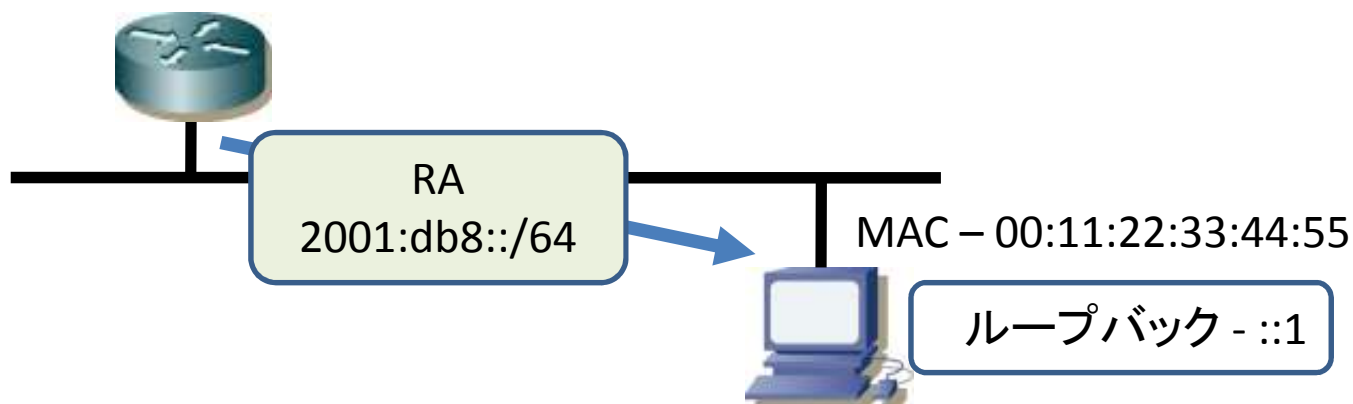
## 3. ルータ広告(RA)の受信

例) 2001:db8::/64

## 4. グローバルアドレスの生成

例) 2001:db8::219:bbff:fe27:37e0

# ホストのIPv6アドレス



リンクローカル - fe80::211:22ff:fe33:4455  
グローバル - 2001:db8::211:22ff:fe33:4455

MAC - 00:11:22:33:44:55を受信

全ノード - ff02::1  
要請ノード - ff02::1:ff33:4455

MAC - 33:33:00:00:00:01を受信

MAC - 33:33:ff:33:44:55を受信

# アドレスとインタフェース

- リンクローカルだと複数インタフェースで困る
  - 例えばping6 ff02::1 したときに、どのインタフェースからパケットを投げればよいか困る
  - OSによっては適当に判断する場合もある
- 送出インタフェースの指定を行うアドレス表記
  - <IPv6アドレス>%<送出インタフェース>
  - 例) ping6 ff02::1%fxp0
  - 例) ping ff02::2%8



# IPv6の押さえどころ

1. しっかり到達性を確保
2. セグメントはとりあえず /64
3. ICMPv6は不用意にフィルタしない
4. サーバのIPアドレス等は手動設定
5. クライアントはRA+DHCPv6で自動設定
6. 経路の向け先はリンクローカルアドレス
7. 9の次にはaが来ます

# IPv6導入事例

# IPv6導入事例 – II/バックボーン

- 最初はIPv6用に別網を構築
- 1998頃
  - PCベースのルータ(kameスタック)
  - tunnel/ethernet
- 2000頃
  - cisco7200に置き換え
  - tunnel/ethernet/T1
- 2005頃
  - IPv4/IPv6デュアルスタックな網に移行開始

# IPv6導入事例 – IIIオフィス環境

- 1998頃 社内利用開始(PCルータで接続)
- 2000頃 Cisco72xxで技術部門を全接続
- 2003頃 オフィス引っ越しを機に全セグメントでIPv4/IPv6デュアルスタック
- 2008頃 IPv6アドレスをリナンバ

# 社内クライアント環境

- ルータはHSRPv2で冗長化
- ルータからRAを広報 (Other Config Flag付き)
- DHCPv6でその他の設定を通知
  - デフォルトの向け先はHSRPの仮想アドレス
  - 参照用DNSのアドレス
  - DNSサフィックス

# ij.ad.jpの公開DNS

- 当然IPv4/IPv6デュアルスタック

ij.ad.jp.	IN	NS	dns0.ij.ad.jp.
-----------	----	----	----------------

ij.ad.jp.	IN	NS	dns1.ij.ad.jp.
-----------	----	----	----------------

dns0.ij.ad.jp.	IN	A	210.138.174.16
----------------	----	---	----------------

dns0.ij.ad.jp.	IN	AAAA	2001:240:bb41:8002::1:16
----------------	----	------	--------------------------

dns1.ij.ad.jp.	IN	A	210.138.175.5
----------------	----	---	---------------

dns1.ij.ad.jp.	IN	AAAA	2001:240:bb4c:8000::1:5
----------------	----	------	-------------------------

# iij.ad.jpなメール環境

- 当然、デュアルスタック
  - 送受信はsendmail (IJJ改)で運用

```
iij.ad.jp.      IN      MX  10  omgi.iij.ad.jp.  
  
omgi.iij.ad.jp. IN      A      202.232.30.70  
omgi.iij.ad.jp. IN      A      202.232.30.144  
omgi.iij.ad.jp. IN      AAAA   2001:240:11e:6300::1:70  
omgi.iij.ad.jp. IN      AAAA   2001:240:11e:6000::1:144
```

# 今後の展望とか



# 技術者、大事

- IPv6をやれる技術者が足りない
  - IPv6対応の機器は増えて来ている
  - 何にせよIPv6の影響は受けるので知識は必要
- だいたい、触ってみないと分からない
  - 検証環境とか実験環境
  - 自分たちを実験台に

# もっと簡単に使える環境を

- とにかく、ネットワークはIPv6対応させちゃう
  - システムの更改とかのタイミングで
  - IPv6導入への準備を整えておく
- 家庭でももっと簡単に使いたい
  - IJはPPTPでIPv6トンネルの提供を始めました
    - <http://www.ij.ad.jp/news/pressrelease/2009/0427.html>
    - Vista SP1以降だとOS標準の機能で利用できます
    - 基本的にはIPv6が利用できるPPTP実装であればOK
    - 申し込み必要なし、しかも無料 ← 大事

# One Internet

- みんなが繋がってるのは良いことだ
  - 新たな利用の可能性が生まれる
  - インターネットの分断や分割は不幸
- もっと多くの参加者が、自由な通信環境を得られればいいなあと
- でもって、もっと楽しいことやりたいでしょ。

おわり