忘れがちなIPv6のアドレス構成

Matsuzaki ‘maz’ Yoshinobu
<maz@iij.ad.jp>

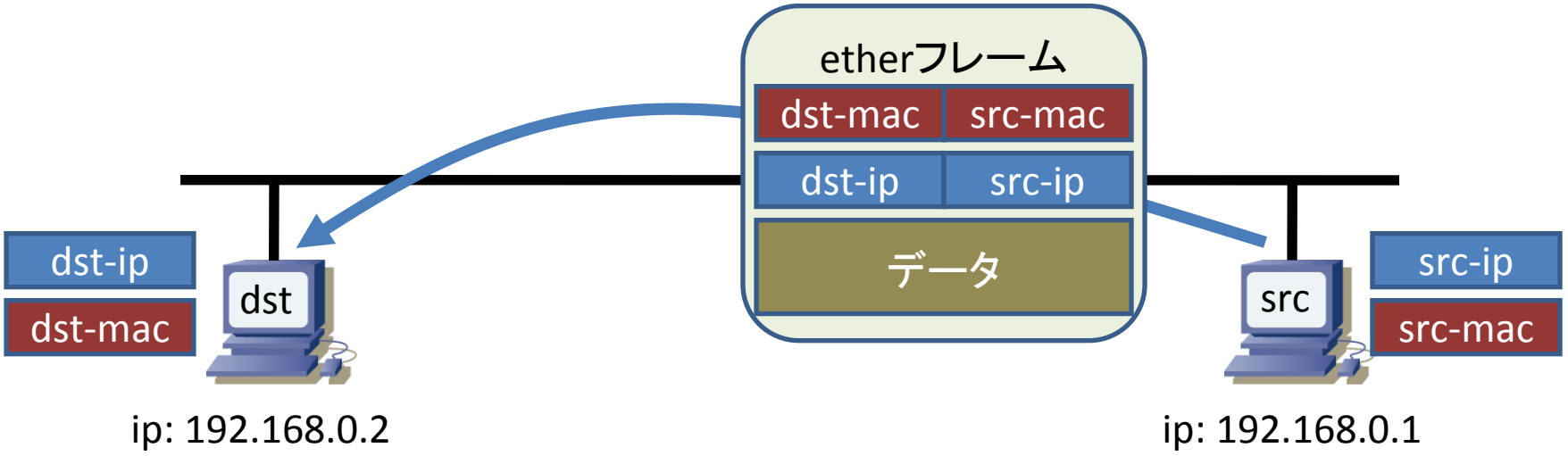
IPv4とIPv6

- パケット転送などの考え方は同じ
- つまり基本はほとんど一緒
  - 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



arp (Address Resolution Protocol)

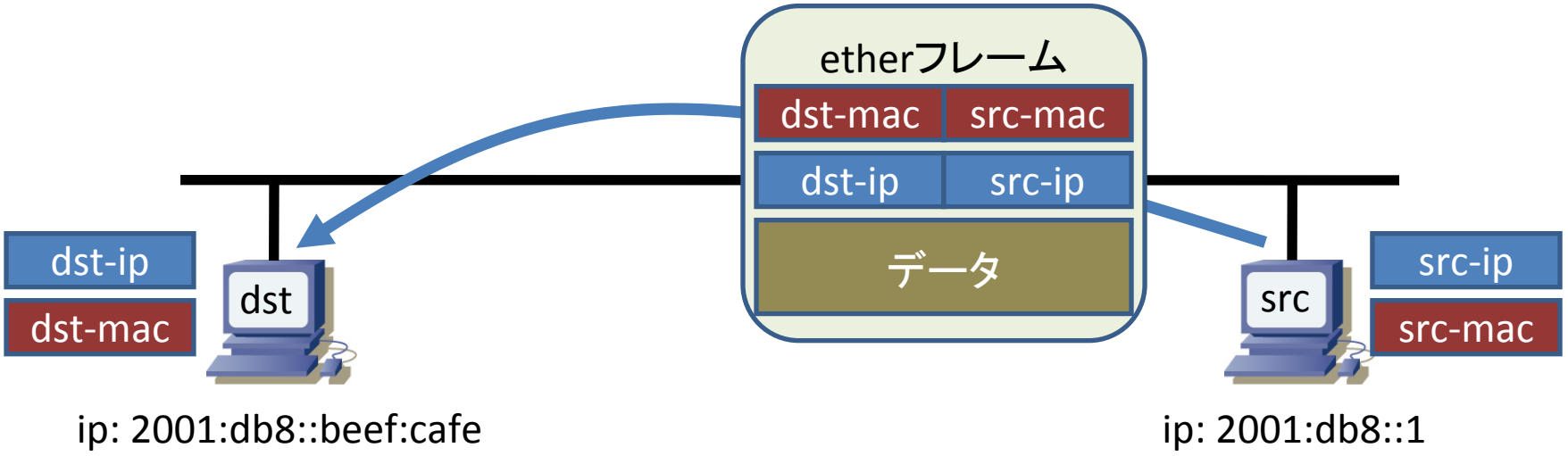
- etherではパケット送信にMACアドレスが必要
  - IPv4アドレスは分かってる (ex. defaultの向け先)
  - 機器のIPv4アドレスからMACアドレスを知りたい
- arpで解決
  - RFC826

```
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



ndp (Neighbor Discovery Protocol)

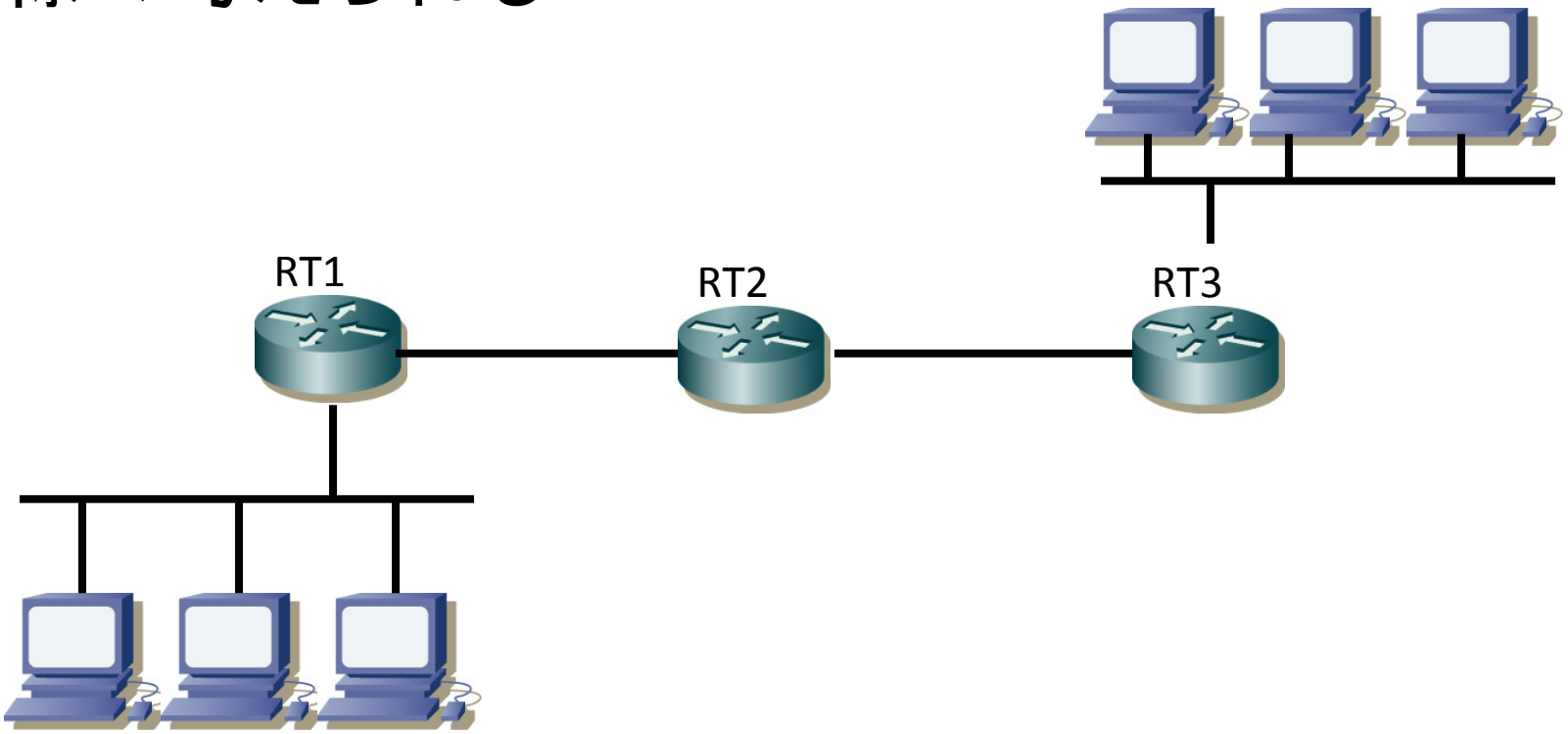
- etherではパケット送信にMACアドレスが必要
  - 機器のIPv6アドレスからMACアドレスを知りたい
- ndpで解決
  - RFC4861
    - ICMP6を利用してMACアドレスを問い合わせる
  - 送り先を未学習ならmulticastアドレス宛て
    - IP: ff02::1:ff00:0000 ～ ff02::1:ffff:ffff
      - 送信先IPアドレスの下位24bitを利用して生成
    - MAC: 33:33:00:00:00:00 ～ 33:33:ff:ff:ff:ff
      - 送信先IPアドレスの下位32bitを利用して生成

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 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 beef cafe 0201
        0x0050:  0016 1761 6486
```

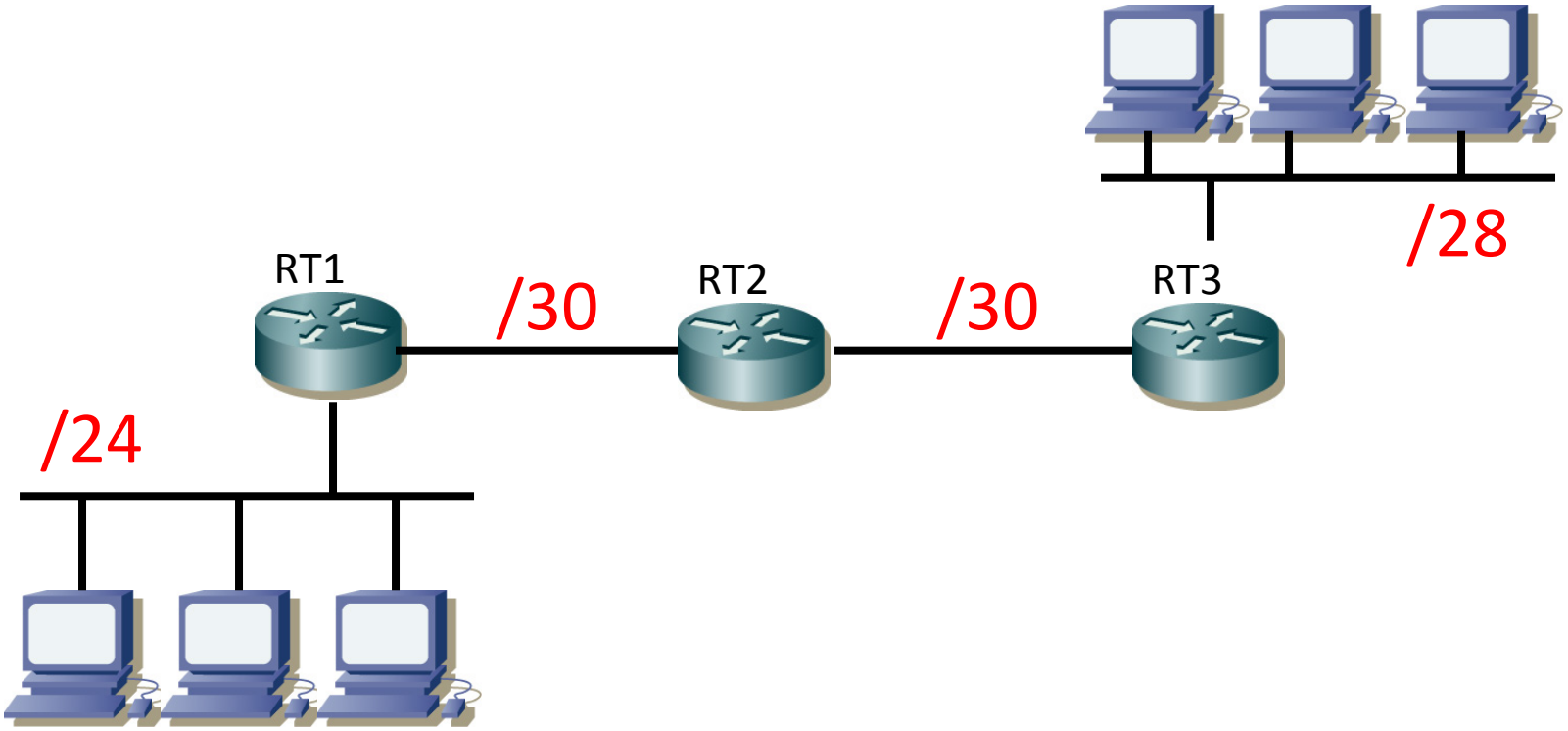
ネットワーク構成

- ルータ間やホストの接続するセグメントなど様々考えられる



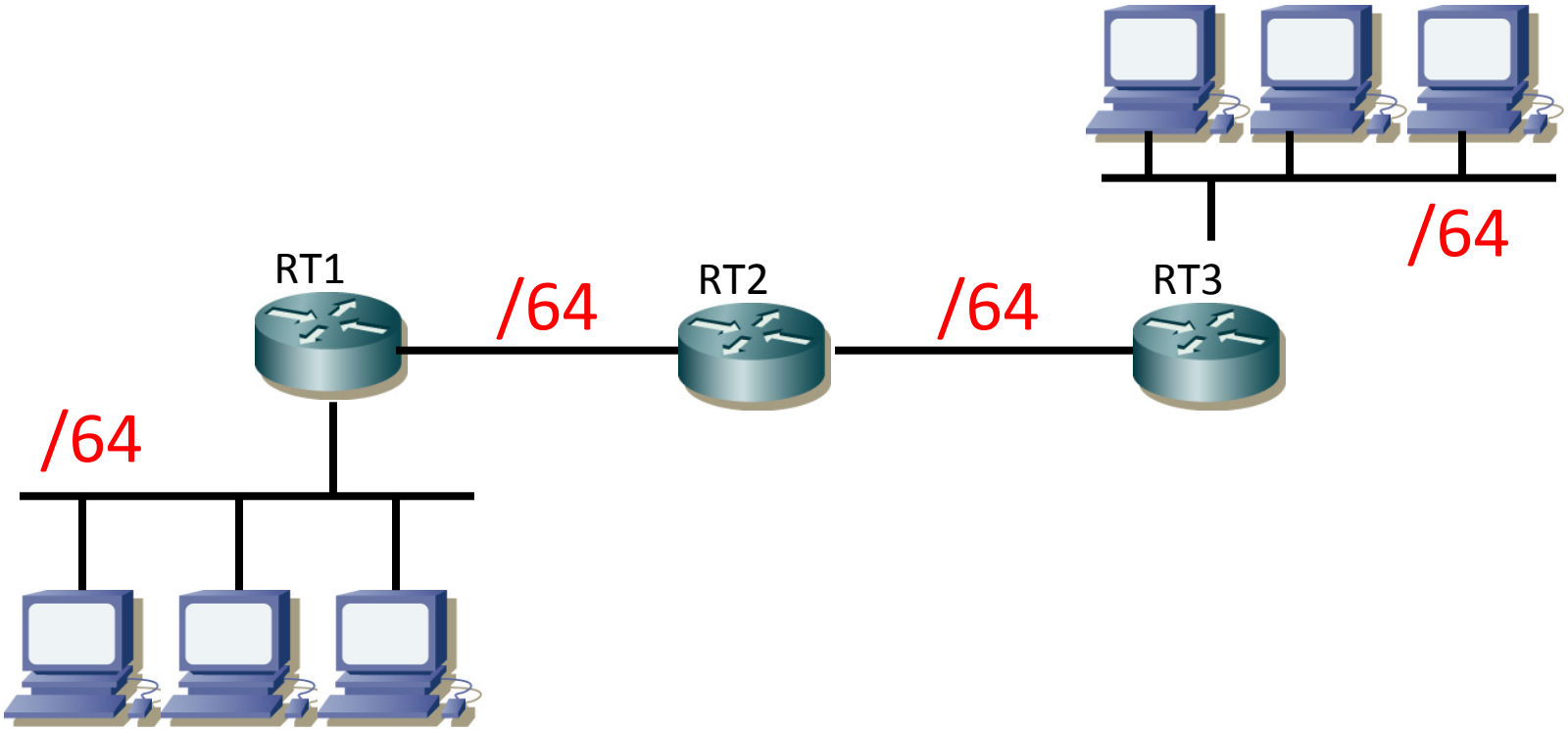
IPv4の場合

- 必要に応じてマスク長を変えて運用



IPv6の場合

- 気にせず/64で運用するのが基本



point-to-pointリンク

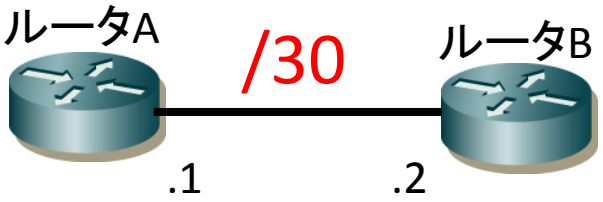
- 主にルータ間で多用されている回線
  - POS、Serialなどなど
    - 障害切り分けとかしやすい
  - 実はtunnelもpoint-to-pointリンク
- パケットを回線に投げれば対向に届く
  - Layer2アドレス解決のためのarpとかいらない

point-to-pointリンクとアドレス

- 古は対向のアドレスをいちいち指定する
  - remote-addressとかdest_addressとかの指定
  - 今でもこんな設定をできるルータもあるが、すでにできないルータもある
- 今はリンク上に/30とか/64のセグメントがあるかの様に設定している
  - Ethernetだろうが、POSだろうが気にしてない

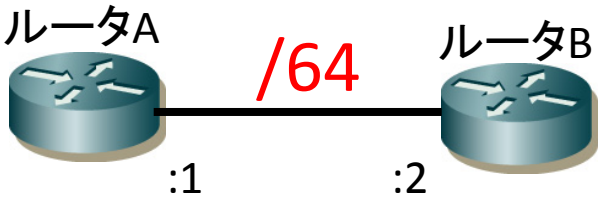
/30でルータ間

- 10.0.0.0/30
  - 10.0.0.0 ← network address
  - 10.0.0.1 ← ルータA
  - 10.0.0.2 ← ルータB
  - 10.0.0.3 ← broadcast address



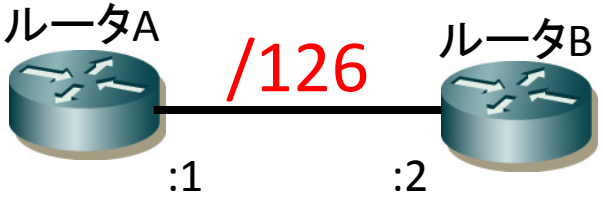
/64でルータ間

- 2001:db8::/64
  - 2001:db8::0 ← Subnet Router-anycast address
  - 2001:db8::1 ← ルータA
  - 2001:db8::2 ← ルータB
  - 2001:db8::3-2001:db8::ffff:ffff:ffff:ffff ← 空き



/126で使ってる場合でも

- 2001:db8::/126
  - 2001:db8::0 ← Subnet Router-anycast address
  - 2001:db8::1 ← ルータA
  - 2001:db8::2 ← ルータB
  - 2001:db8::3 ← 空き



宛先のいないパケット

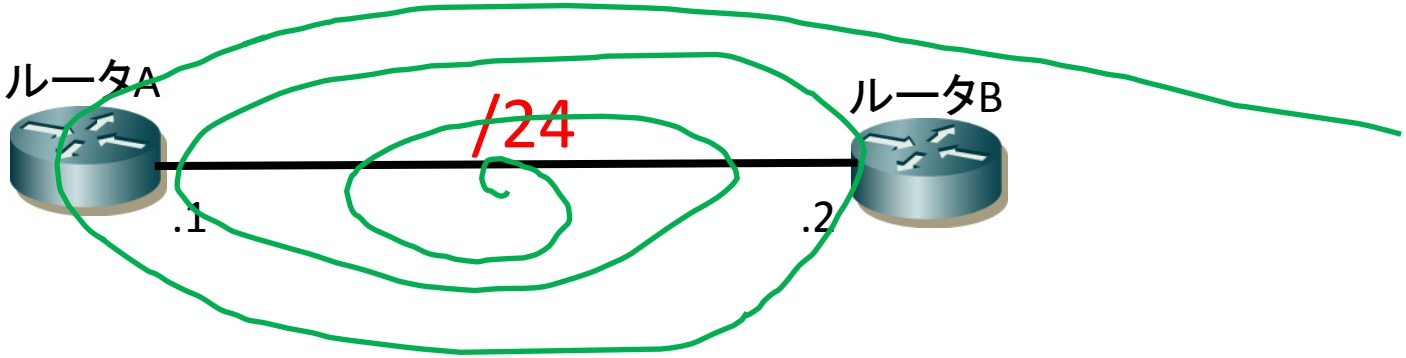
- セグメント上で、どの機器も利用していない様な‘空き’のアドレス宛てにパケットが届いたら
- Ethernet
  - arpやndpで確認し、居なければhost unreach
    - これはこれで別途懸念点はあるけど
- point-to-pointリンク
  - 回線に投げてみる・・・

point-to-pointリンクと空きアドレス

- 10.0.0.0/24をリンクに割り当て
  - 10.0.0.1をルータA、10.0.0.2をルータBに

Q. 10.0.0.13宛てのパケットはどうなる？

A.



ping pong

- point-to-pointリンク上に空きアドレスがあると、そこ向けのパケットがループする
  - IPv4では/30とか/31とか利用して空きがなかった
  - IPv6だと/64とか/126とか使ってるので空きがある
- もちろんIPv6では古くからこの問題は認識されていて、対策が議論されてきた
  - 実はよーく読むとRFCにも書いてある

RFC4443 – ICMPv6

3. ICMPv6 Error Messages

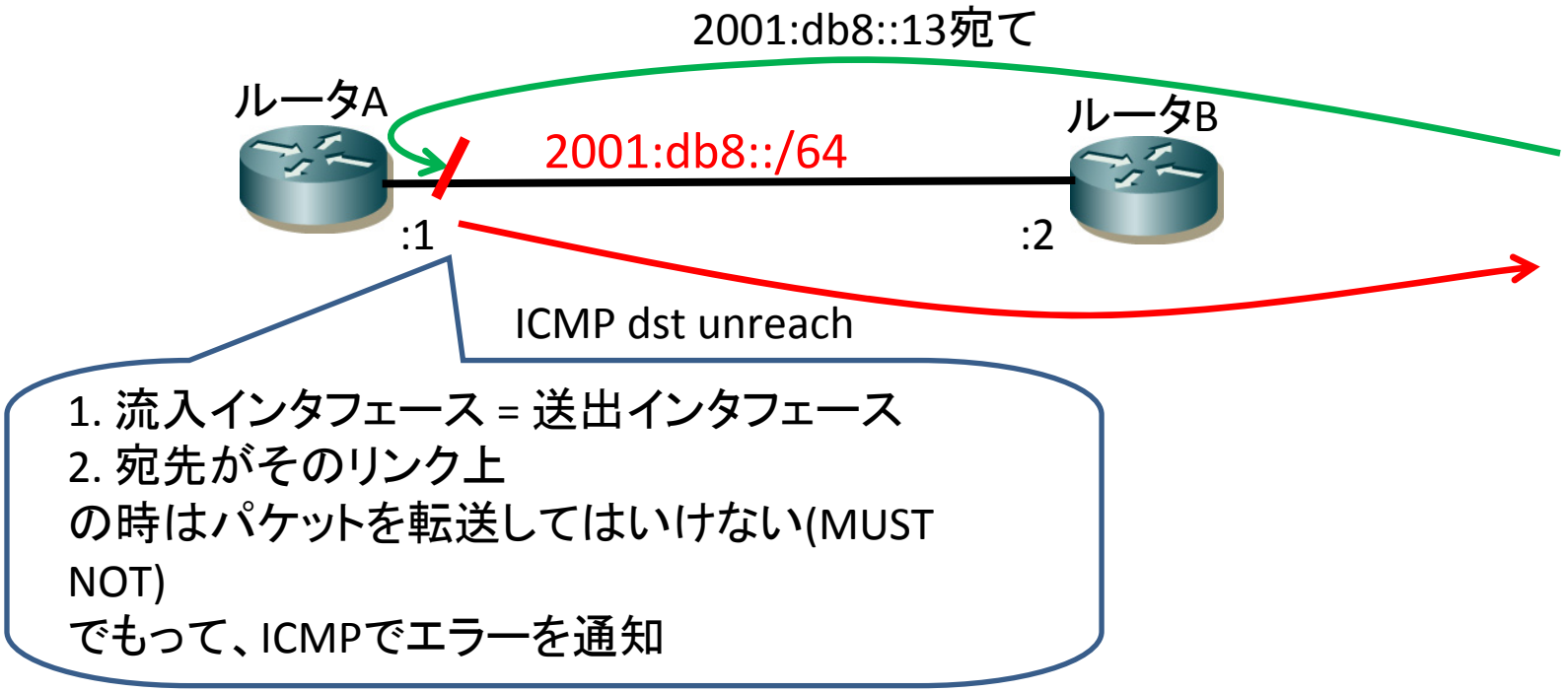
3.1. Destination Unreachable Message

<snip>

One specific case in which a Destination Unreachable message is sent with a code 3 is in response to a packet received by a router from a point-to-point link, destined to an address within a subnet assigned to that same link (other than one of the receiving router's own addresses). In such a case, the packet MUST NOT be forwarded back onto the arrival link.

つまり、RFC曰く

- 2001:db8::/64をpoint-to-pointリンクで使っている時、2001:db8::13宛てのパケットが来たら



知ってるのが重要。

- 仕様上は問題なし、でも実装は違うかも
  - 良くある話
- こういった、‘特別な場合’は忘れられがちなので、使う前にはキチンと確認を
  - 僕たちだって忘れそう
  - ベンダだって忘れそう

念のための方策の検討

- 万が一に備えるのが重要
  - 事前に複数の対策を講じておくとか
  - いよいよとなった時に逃げの方策があるかとか

1. link-localで頑張る
2. 泣きながらパケットフィルタ
3. /127で生きてみる

案1 ルータ間はlink-localのみ

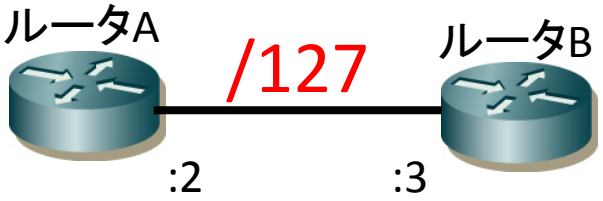
- IPv6ではルータ間はlink-localのみでも動く
  - loopbackにだけアドレスを振っておけばルーティングプロトコルも大丈夫
  - 隣接ルータ間はlink-localで経路のやりとり
- 問題もいくつか
  - リモートからのping監視など
  - eBGPでのnexthop書き換えが面倒
  - tracerouteではどのリンクを通ったか分かりにくい

案2 泣きながらパケットフィルタ

- いわゆるinfrastructure ACLと一緒
  - point-to-point用にアドレスを確保して、そこ宛てのパケットをネットワーク境界でフィルタ
  - IPv6アドレスはいっぱいあるから
- 問題もある
  - フィルタのメンテナンス
  - 他のASとの接続点とか

案3 /127で生きてみる

- 2001:db8::2/127
  - 2001:db8::2 ← ルータA
  - 2001:db8::3 ← ルータB



- 空きアドレスがなくなって、ばっちり
- でも実は既にこんな事を検討した人はいた

/127とRFC

- ‘use of /127 considered harmful’ [RFC3627]
- 現時点では/127の利用は有害らしい
  - subnet-router anycast addressが使えない
    - 実はまだほとんどのルータには実装されてないけど、実装されるとDADで問題になる
    - /31の時の様に使わないことにするRFCも書けるけど、‘特別な場合’を増やす問題もある
  - /64より長いprefix長での問題
    - u/g bitとか

まとめ

- 仕様と実装は違うかも
  - 使う前にはちゃんと検証を
    - 問題を見つけたら、ベンダにも教えてあげましょう
  - 特に特別な場合は要注意
    - point-to-pointリンクでのICMPv6とか
    - tunnelを張るときも同様なので注意
- ちょっとした違いに注意
  - 思い込みとか、IPv4では大丈夫だったとか

おわり