頑張れフォールバック

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

フォールバック

- ダメだったら次に試す先
  - 代用とか代替とか予備
- インターネットでは多用
  - 冗長性や可用性の確保
  - 複数台のDNS、複数個のAレコード

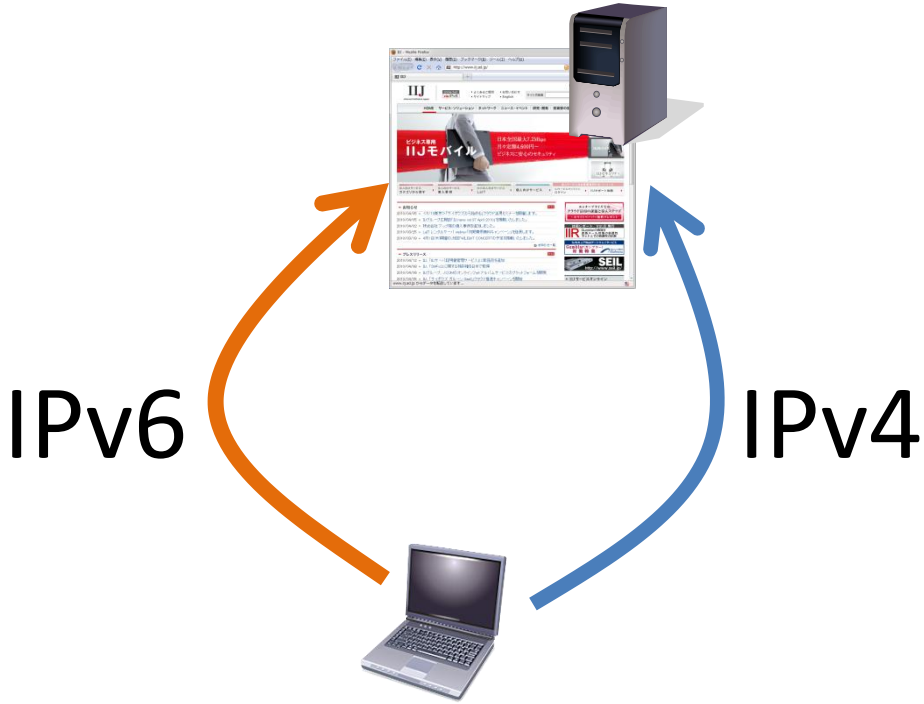
- 端末が通信できる様によろしくやってくれる
  - ユーザは気にしなくてよい

今回の話題

- IPv6→IPv4フォールバック
  - 通信プロトコルを選ぶ
- Path MTUフォールバック
  - 最大Path MTUを選ぶ

v6→v4 フォールバック

- IPv6/IPv4デュアルスタックで選択肢が増える
  - どっちか通信できる方でよろしく通信する



RFC3484
Default Address Selection for IPv6

Policy Table

	Prefix	Precedence	Label
	::1/128	50	0
IPv6 →	::/0	40	1
6to4 →	2002::/16	30	2
IPv4 →	::/96	20	3
	::ffff:0:0/96	10	4

- 要は、基本的にIPv6がIPv4より優先

closed IPv6 network

- インターネットにアクセスできないIPv6網
  - 端末はIPv4とIPv6のIPアドレスを持つ
  - IPv4だとインターネットにアクセスできる
- 宛先もIPv6/IPv4デュアルスタックになる
  - webサイトとかとか

- 必ずIPv6→IPv4 フォールバックが発生する

ちなみにNTT NGNの仕様

- IP通信網サービスのインタフェース
  - NTT東
    - フレッツシリーズ- 第三分冊 – 第11版
    - 2.4 レイヤ3仕様
  - NTT西
    - フレッツシリーズ – 第37版
    - 4.5.4 ネットワークレイヤ(レイヤ3)仕様
- IP通信網内に存在しない宛先に送信されるパケットについては、IP通信網において応答なくパケット破棄される場合や、RFC793に規定されるRSTビットをセットしたTCPパケットを返信する場合があります。

v6→v4 フォールバックの実施

- アプリケーションがそれぞれ頑張ってる
  - メールクライアントとか、webブラウザとか
- v6→v4 フォールバックの成否は実装次第
  - アプリケーション毎に挙動が違う可能性

事例1　メールクライアント

- POPサーバにAAAAレコードが付いていると、メールが受信できない
  - v6→v4 フォールバック以前の問題。かなりダメ
  - IPv6でPOPを試しもしない
  - でも送信はIPv6でも問題なくできる。不思議
- 開発元には報告済み。対応をお話中
  - 古いバージョンなので、アップグレード推奨かも

事例2　メールクライアント

- v6→v4 フォールバックできない
  - IPv6アドレスに送受信を試みるが、失敗したらそのままあきらめる
  - IPv4アドレスは試さない。悲しい
- 開発元には報告済み。対応をお話中
  - パッチが出るかも・・・

事例3　webブラウザ

- AAAAが多いとv6→v4 フォールバックできない
  - 与えられたIPアドレスを上から順に5つまで試す
  - AAAAを5つ以上書いてあるとAにたどり着かない
  - 再読み込みさせると、続きから試すようでアクセスできるようになる
- 開発元に報告済み
  - KB2148580 発行。regeditで接続試行回数を変更
    - Thanks to Microsoft & NTT東

考えどころ

- closedな網はやっぱり邪悪かも
- v6→v4 フォールバックは各開発元でちゃんと実装してもらう必要がある
  - アプリケーションの開発者
- v6→v4 フォールバックすると事前に分かってるなら、IPv6でアクセスを試さなきゃいい
  - OSレベルで何かできるかも

closed IPv6網でのアクセス制御

- 案1．Policy Tableの変更
  - closedなIPv6網のIPアドレスは、closedな所と通信するときだけ使うポリシを追加
  - 宛先がインターネットだったらIPv4で通信する
  - ○すぐ動く。×各端末で設定変更が必要
- 案2．端末がインターネットへの接続性を診断
  - インターネットへアクセスできないアドレスファミリはイントラネットモードに落とす
  - ○設定変更必要なし。×実装まで時間がかかる

Policy Tableの変更

- 要は、closedなIPv6網のIPv6 prefixを別ラベルで登録しておけばOK
  - 通信元、通信先で同じラベルを優先
- http://www.attn.jp/maz/p/i/policy-table/

```
netsh interface ipv6 delete prefixpolicy ::1/128
netsh interface ipv6 delete prefixpolicy ::/0
netsh interface ipv6 delete prefixpolicy 2002::/16
netsh interface ipv6 delete prefixpolicy ::/96
netsh interface ipv6 delete prefixpolicy ::ffff:0:0/96
netsh interface ipv6 delete prefixpolicy 2001::/32
netsh interface ipv6 add prefixpolicy ::1/128 50 0
netsh interface ipv6 add prefixpolicy ::/0 40 1
netsh interface ipv6 add prefixpolicy 2001:c90::/32 40 6
netsh interface ipv6 add prefixpolicy 2001:d70::/32 40 6
netsh interface ipv6 add prefixpolicy 2001:d71::/32 40 6
netsh interface ipv6 add prefixpolicy 2001:d72::/32 40 6
netsh interface ipv6 add prefixpolicy 2001:d73::/32 40 6
netsh interface ipv6 add prefixpolicy 2001:a000::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a100::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a200::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a300::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a400::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a500::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a600::/24 40 6
netsh interface ipv6 add prefixpolicy 2001:a600::/24 40 6
netsh interface ipv6 add prefixpolicy 2404:1a8::/32 40 6
netsh interface ipv6 add prefixpolicy 2408::/24 40 6
netsh interface ipv6 add prefixpolicy 2408:100::/24 40 6
netsh interface ipv6 add prefixpolicy 2408:200::/24 40 6
netsh interface ipv6 add prefixpolicy 2408:300::/24 40 6
netsh interface ipv6 add prefixpolicy 2002::/16 30 2
netsh interface ipv6 add prefixpolicy ::/96 20 3
netsh interface ipv6 add prefixpolicy ::ffff:0:0/96 10 4
netsh interface ipv6 add prefixpolicy 2001::/32 5 5
```

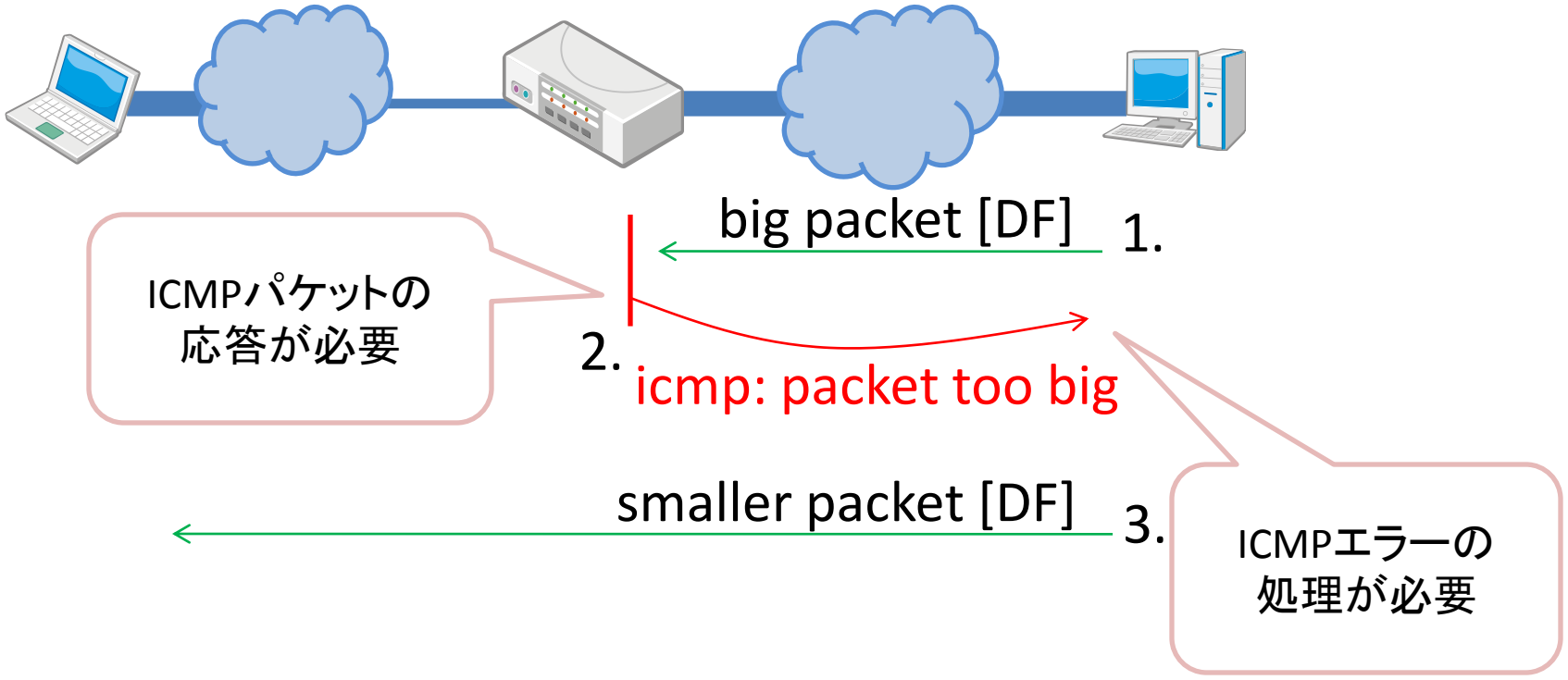
Windowsではこんなコマンド

管理者権限が必要

Path MTU discovery

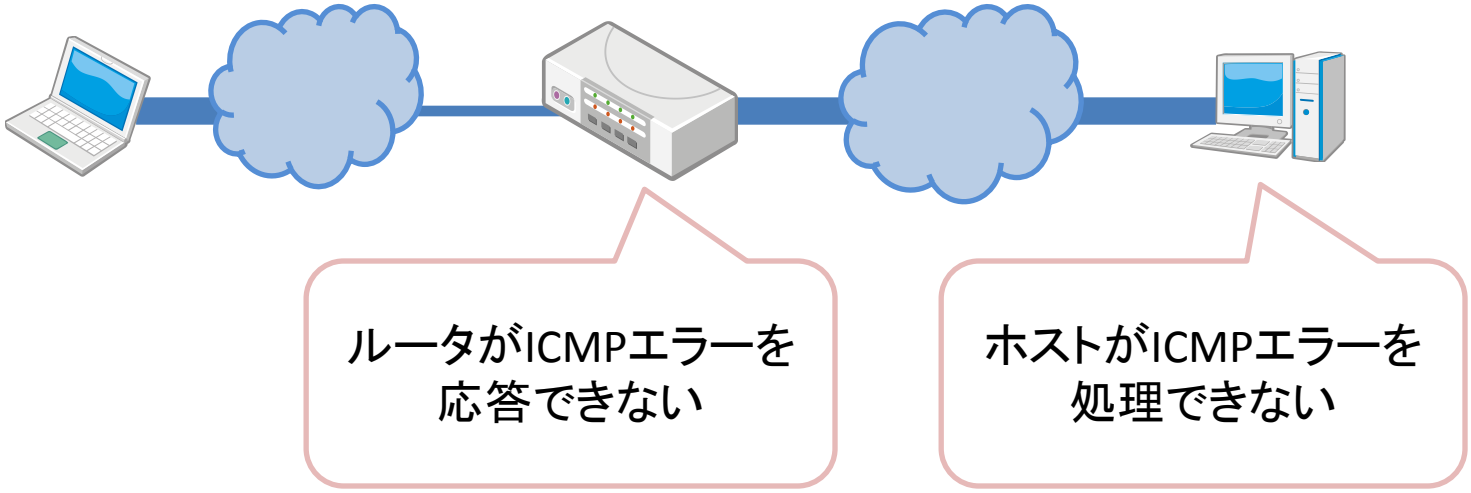
- 大きなIPパケットで効率的な通信
  - 利用できるPath MTUを検出する
  - packet/secを減らせる
- IPv6での実装は”SHOULD”
  - Path MTU discovery for IPv6 [RFC1981]
  - 利用しない場合は 1280byteを利用

Path MTU discovery のシナリオ



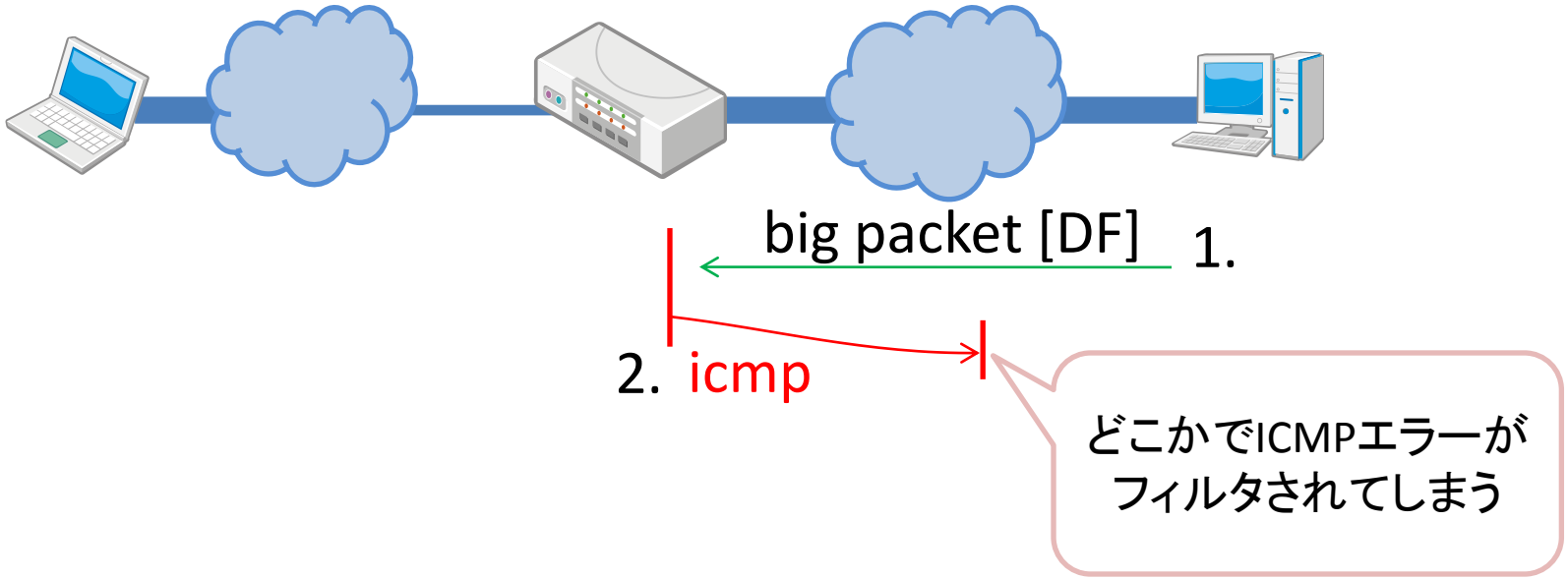
失敗事例 #1: 未対応

- PMTUd ブラックホールルータ
- ICMPエラーの処理ミス



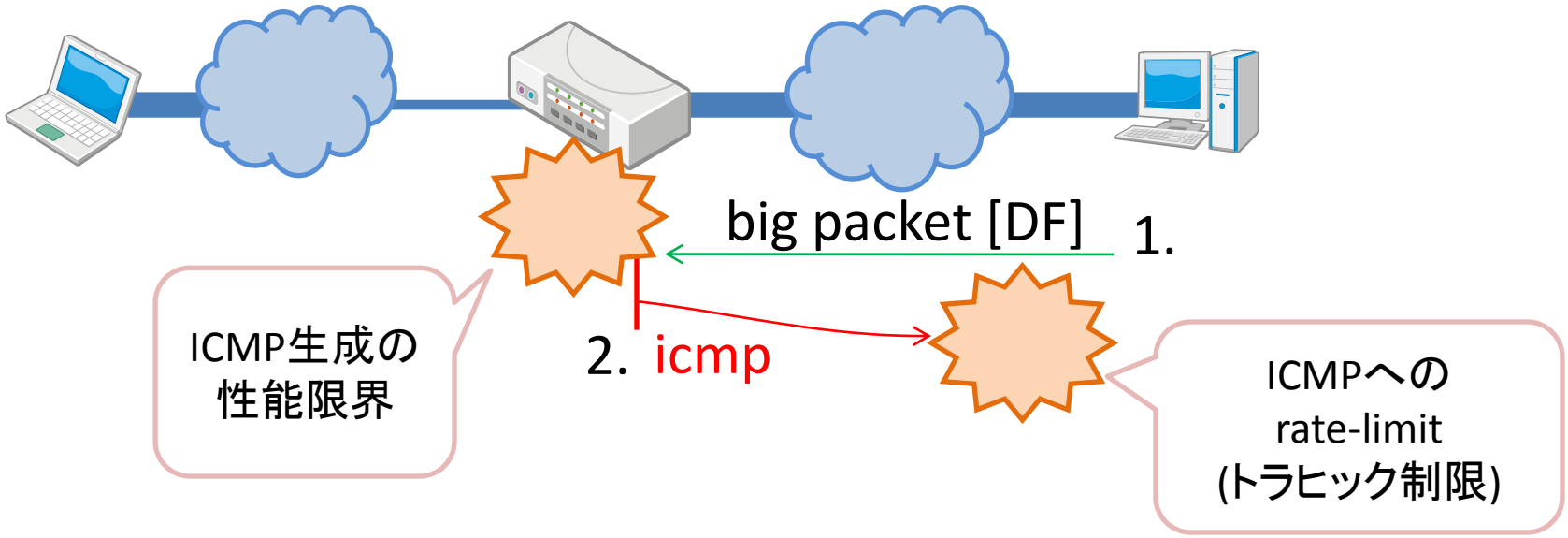
失敗事例 #2: フィルタ

- 安易なパケットフィルタ
- ダメなセキュリティポリシ



失敗事例 #3: 制限

- ICMPの流量を制限しているネットワーク
- ICMPメッセージを生成するルータの性能



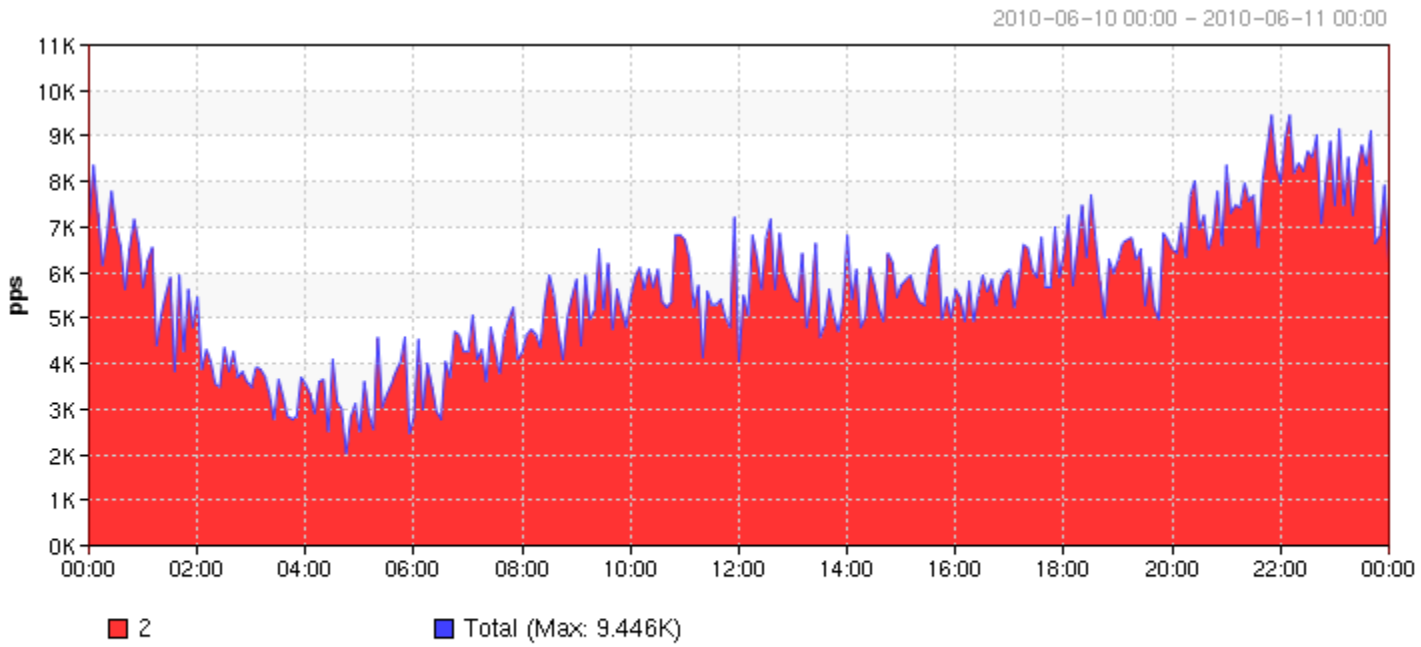
ICMP生成の制限

- cisco ios
  - ip icmp rate-limit unreachable 500
    - means icmp errors are limited to one every 500msec
  - ipv6 icmp error-interval 100
    - means icmp errors are limited to one every 100msec
- juniper junos
  - icmpv4-rate-limit {packet-rate 1000;};
    - means max 1000pps for icmp to/from RE
  - icmpv6-rate-limit {packet-rate 1000;};
    - means max 1000pps for icmp to/from RE

失敗事例から分かること

- Path MTU discoveryは、各機器が対応していたとしても失敗する可能性がある
  - 性能問題
  - ICMPがフィルタされちゃう
- Path MTU discoveryは何だか例外処理
  - ルータの性能評価であまり気にされていない

TCP SYNレート



IPv4の事例に学ぶなら

- ほとんどのブロードバンドルータがTCP MSSに関する機能を持つ
- トンネルリンク等で、TCP MSSの値を書き換え
  - PPPoEとか、何らか1500byteより小さなMTU
  - 必要のないフォールバックを避けるため
- うまく動いている！
  - ユーザからの苦情ってないよね

IPv6での選択肢

- 案1. RAでMTUを通知
  - でもでも、家庭でGbEが導入されてきている
  - たぶん、家庭内でjumbo frame使いたくなるよね
  - ○全般の通信に有効 ×家庭内の通信に影響
- 案2. TCP MSSをブロードバンドルータに実装
  - TCPにしか有効ではないけれども
  - ○IPv4ではうまくいってる ×TCP以外の通信

考えどころ

- TCPはMSSの調整で救うのが無難
  - 主要な通信はこれで救える
  - RAでのMTU通知は行きすぎ感がある
- それでもPath MTU discoveryは必要
  - より小さなMTUのリンクがあるかも
  - ルータや端末がちゃんとサポート

フォールバック全般

- 時間がかかる
  - 基本はトライ＆エラー
- ユーザに通知していないことが多い
  - うまく通信できているなら、みんな気にしない
  - 通信できていない時の通知が不十分
    - 何がエラーか、対策があるのか

考えどころ

- 不要なフォールバックを避けられるように
  - 端末/アプリケーションで頑張る？
  - ネットワークで頑張る？
- 状況をうまくユーザに通知できるように
  - アプリケーションのエラー表示？
  - 診断サイト/ツール？