



Internet Week 2016 T2 IPv6トラブルシューティング

株式会社ブロードバンドタワー
國武 功一

IPv6ネットワークを構築・運用する際に

注意すべき観点について解説します

主にサーバセグメントについて解説します。

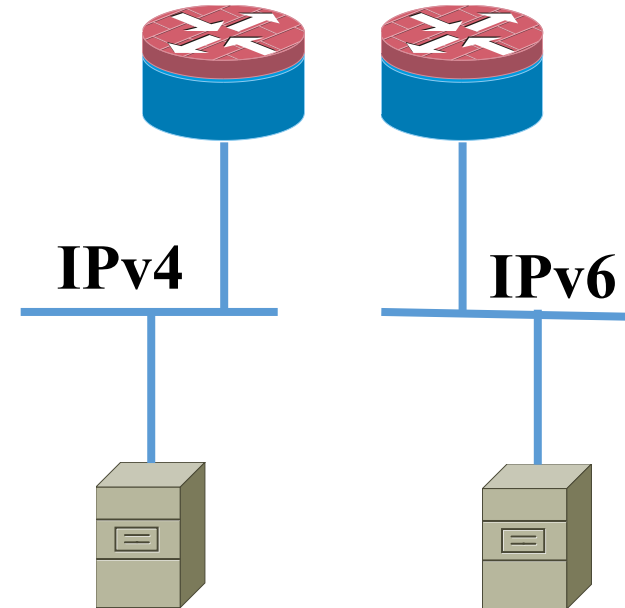
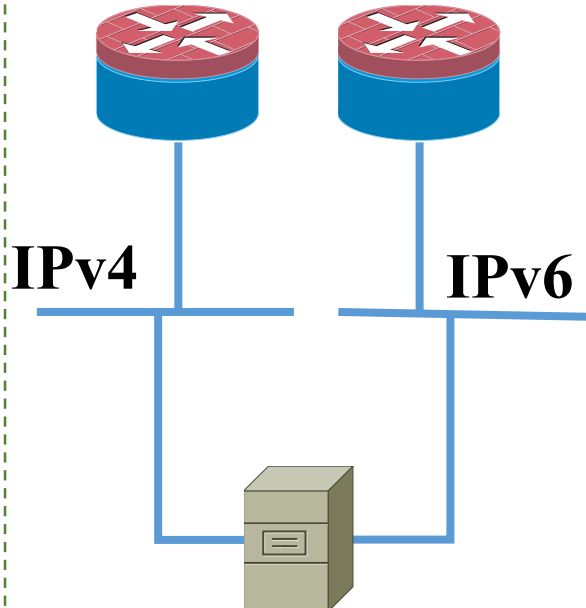
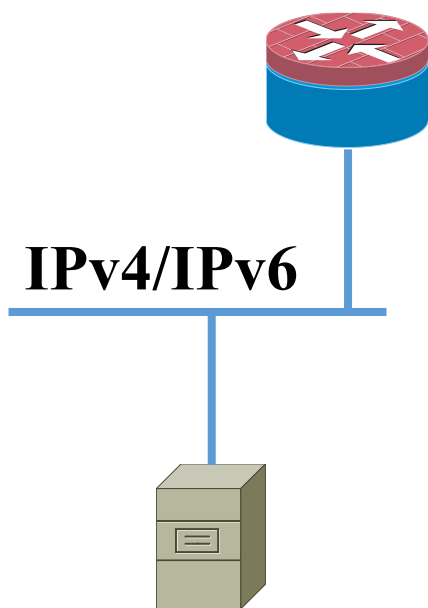
- IPv6ネットワーク概要（基本）
 - ◆ 構成例について
 - ◆ DualStack
 - ◆ Fallbackについて
- IPv6よくある誤解
- IPv6トラブル事例および防止策
 - ◆ DNS関連
 - ◆ Path MTU Discovery Blackholeその原因
- 構築時の注意点

IPv6ネットワーク概要

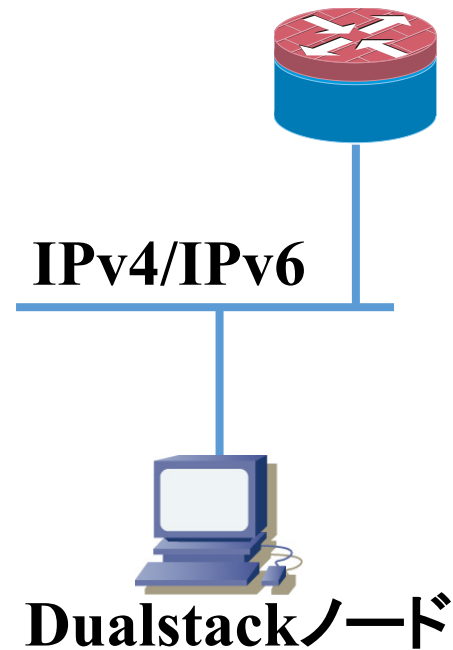
- IPv4とIPv6は別プロトコル
- サーバは、構成も経路も分けることが可能

コスト低

コスト大

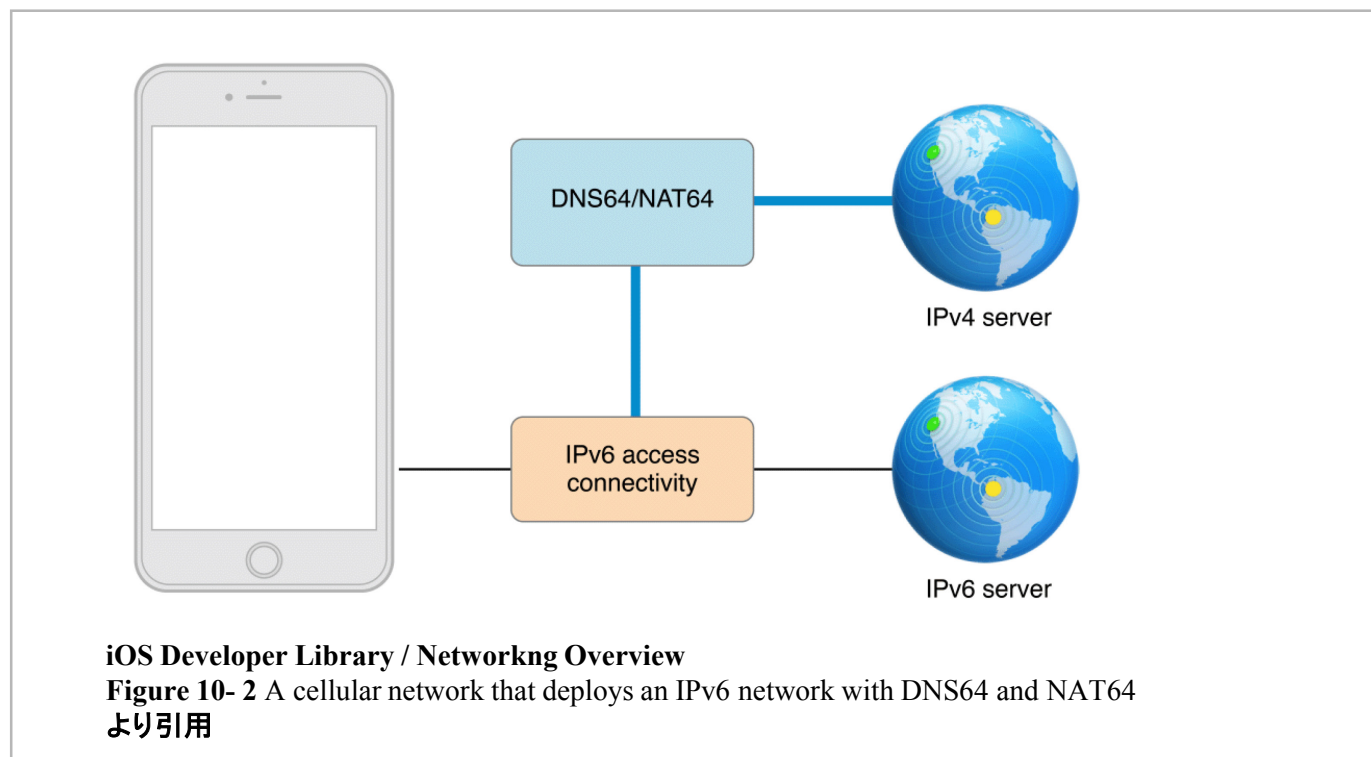


- クライアント側でのIPv6は、DualStackでの対応が多数。
- クライアントに割り当てられるIPv6アドレスはグローバルアドレスが割り当てられることが多い。



■ IPv6 onlyでの運用も想定されつつある。

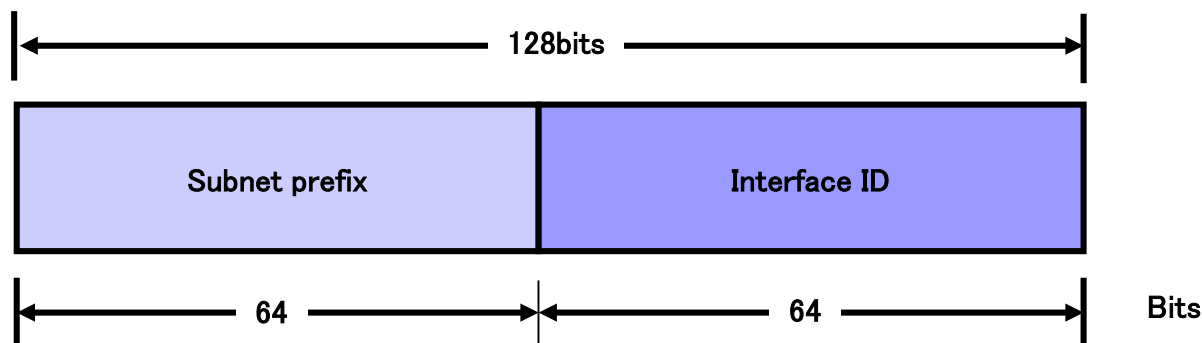
◆ DNS64+NAT64での運用 (例：iOS9のアプリ要件 ※ 1)



※1 <http://goo.gl/0USFlz>

■ Privacy Extension

クライアントに割り当てられるIPv6アドレスは、下位64bitが定期的にランダムに更新され、ユーザの特定が難しいように考慮されている（実装および利用されているかは、個別設定）



⇒ DNSの逆引き設定が困難であり、期待できない。
このため、クライアントに対する名前ベースのACLは利用できないことが多い。


```
1. bash
17:24:11 kunitake@stardust $ ifconfig en0
en0: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    ether 00:00:5e:00:53:0f
    inet6 fe80::200:5eff:fe00:530f%en0 prefixlen 64 scopeid 0x4
    inet6 2001:dc2:cafe:2:200:5eff:fe00:530f prefixlen 64 autoconf
    inet6 2001:dc2:cafe:2:d993:6ace:f62d:4fa4 prefixlen 64 autoconf temporary
    inet 169.254.34.205 netmask 0xffff0000 broadcast 169.254.255.255
    nd6 options=1<PERFORMNUD>
    media: autoselect
    status: active
17:24:37 kunitake@stardust $
```

Modified EUI-64 と呼ばれるMACアドレス由来のアドレスとは別に temporary とついたランダム生成されたアドレスが付いている。

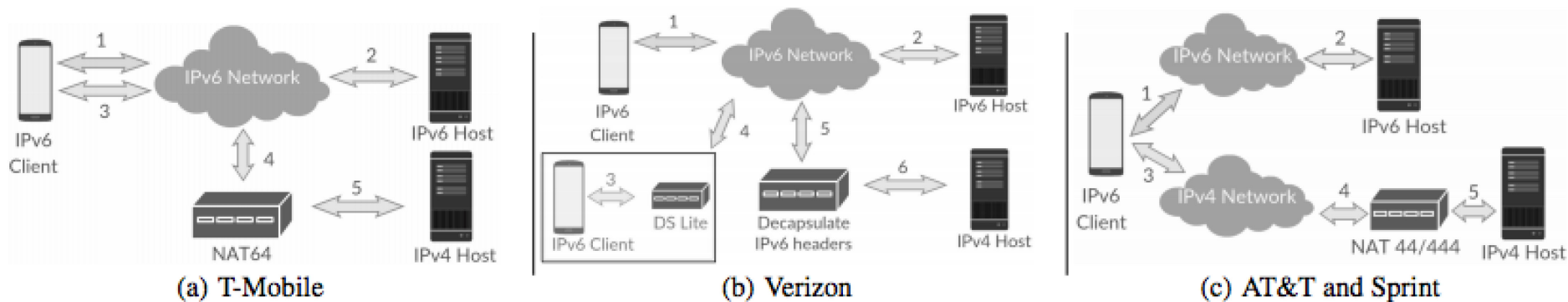
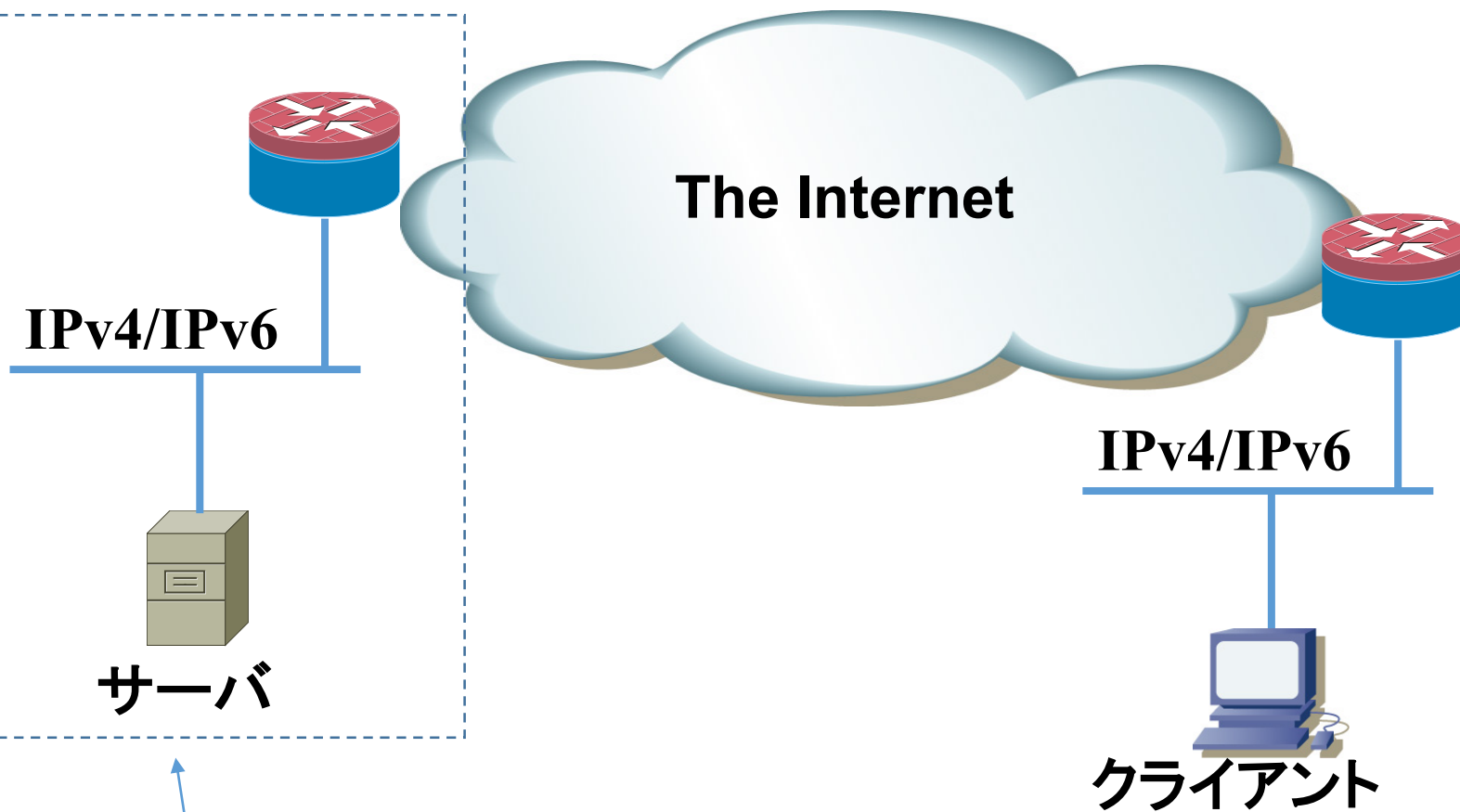


Figure 1: IPv6 infrastructure deployment in different cellular carriers in the US.

- T Mobileは、IPv6対応端末におけるIPv4アクセスに関しては464XLATを選択。基本的に IPv6 only
- Verizon は、LTEネットワークでIPv6対応端末にはDualstackを選択。IPv6対応端末におけるIPv4アクセスに関しては DS-Lite でトンネリング(網はIPv6)
- AT&T and Sprint は、DualStack(IPv4はNAT44/444)

※図は“A Case for Faster Mobile Web in Cellular IPv6 Networks”より引用

<https://www.akamai.com/us/en/multimedia/documents/technical-publication/a-case-for-faster-mobile-web-in-cellular-ipv6-networks.pdf>



基本固定IPアドレス

下位64bitが固定であるケースと、時間経過とともにランダムに変更するもののが存在

- IPv4 stackとIPv6 stackの両方を実装したノードを dualstack ノードと呼ぶ
- 単一のFQDNでIPv4/IPv6サービスを提供するケースが多い。
- FQDNを共有するケースでは、ユーザ側でのフォールバックについて気をつける必要がある。

```
;; Server
```

```
www      IN      A       192.0.2.1
```

```
          IN      AAAA    2001:db8::80
```

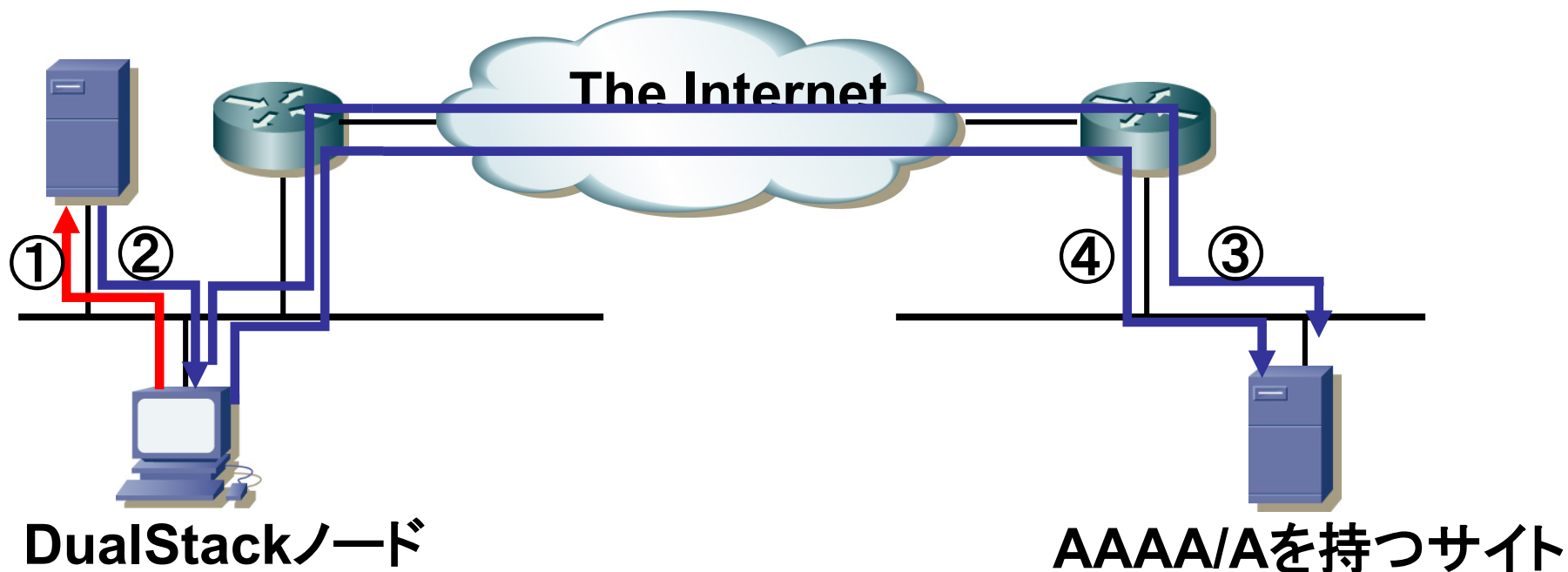
①FQDNの名前解決を行う

IPv4 transport / IPv6 transportでもどちらでもよい)

②AAAA RR, A RRが返る

③IPv6で接続

④IPv6で接続できないと、IPv4へフォールバック



■ IPv6閉域網フォールバック問題

- ◆ IPv6グローバルアドレスを閉域網で利用した場合の問題点 (いわゆるBフレッツ問題)

TCP RSTを網側から返すことで、影響を極小化



■TCPのタイムアウトが長く、ユーザへの影響が大きい

Happy Eyeballsによるブラウザ側での対応が進む

■Happy Eyeballs

最初から、IPv4/IPv6の両方で接続を開始し、先に接続が成功した方で通信を行う。これにより、TCPのタイムアウトを伴うような事象での影響を極小化（*1）

かなり改善されてきたが、Happy Eyeballsはアプリによる対応のため、影響を受ける、受けないは実装依存

(*1)iOS9とEI Capitanでは、IPv4側に25msecの意図的なdelay

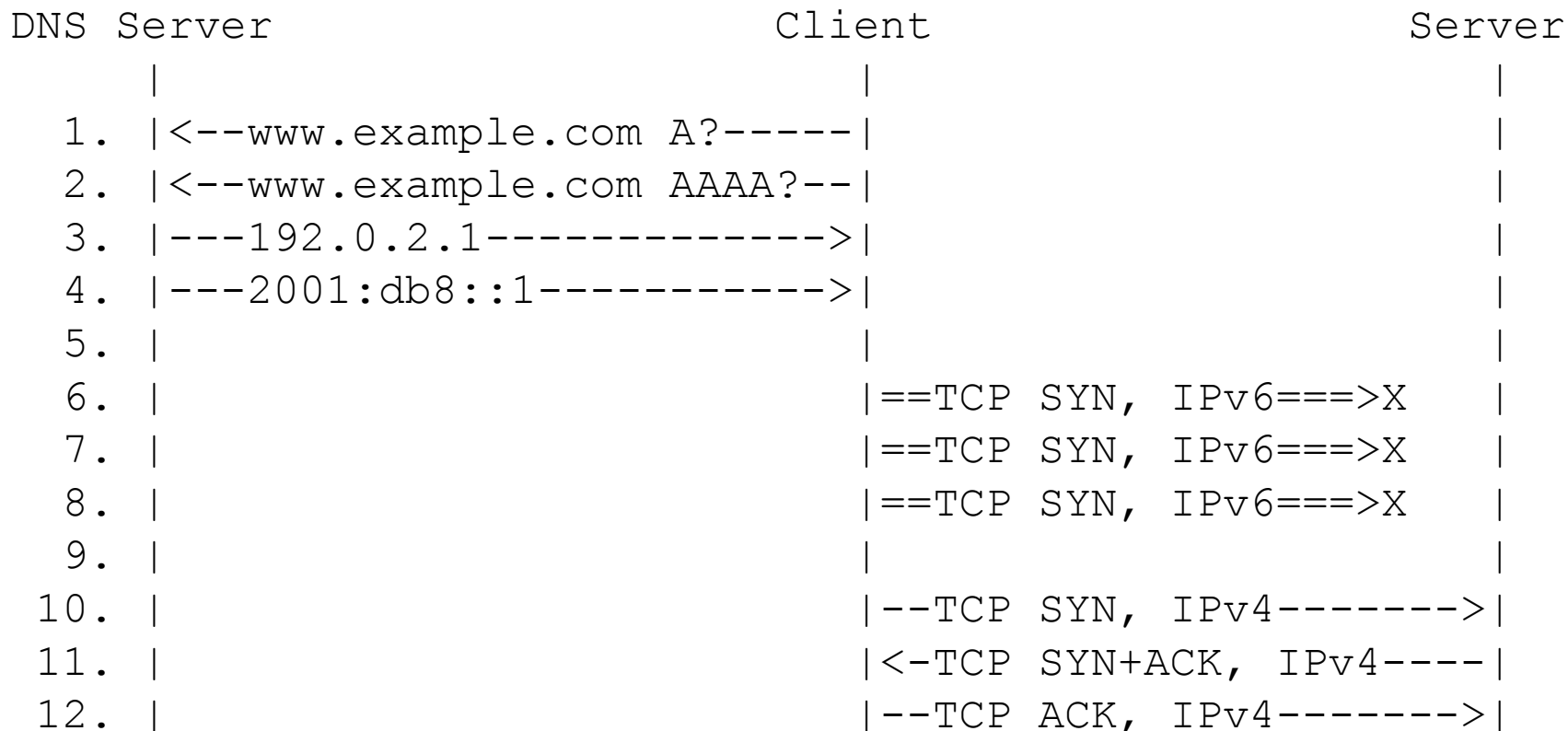


Figure 1: Existing Behavior Message Flow

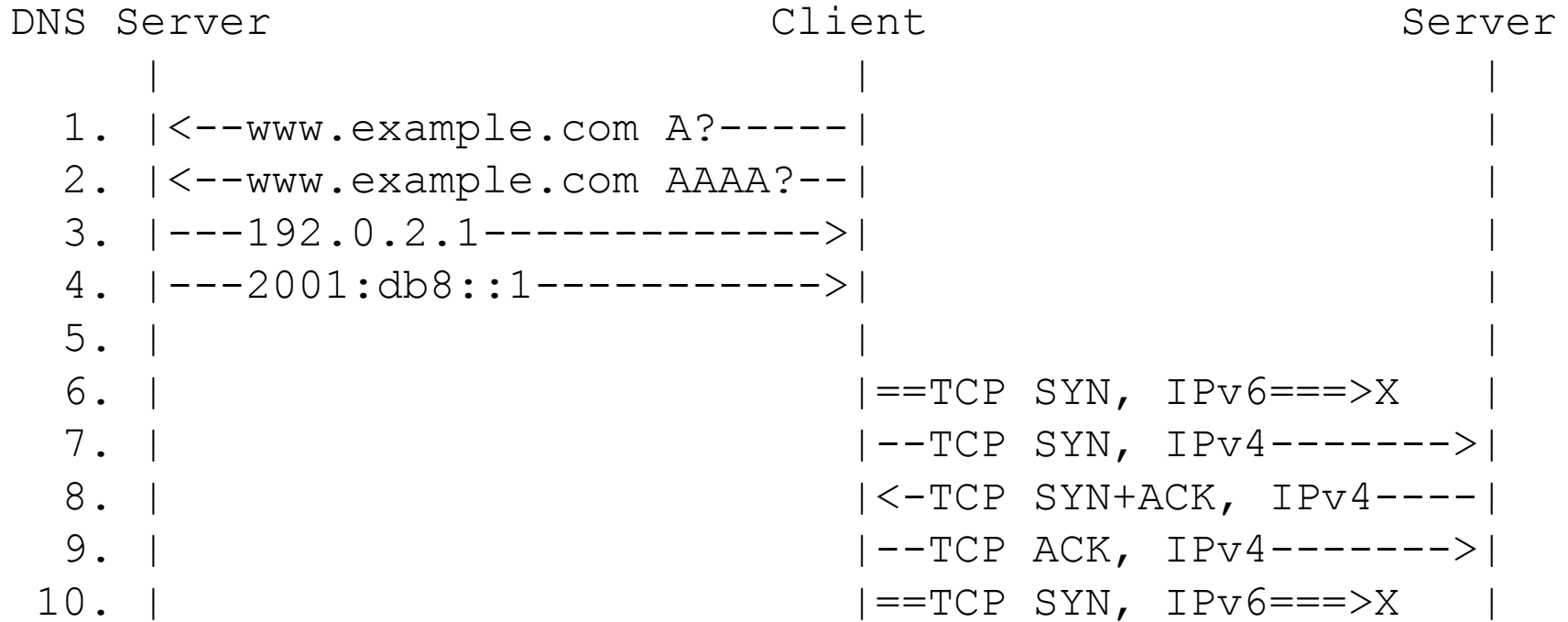


Figure 2: Happy Eyeballs Flow 1, IPv6 Broken

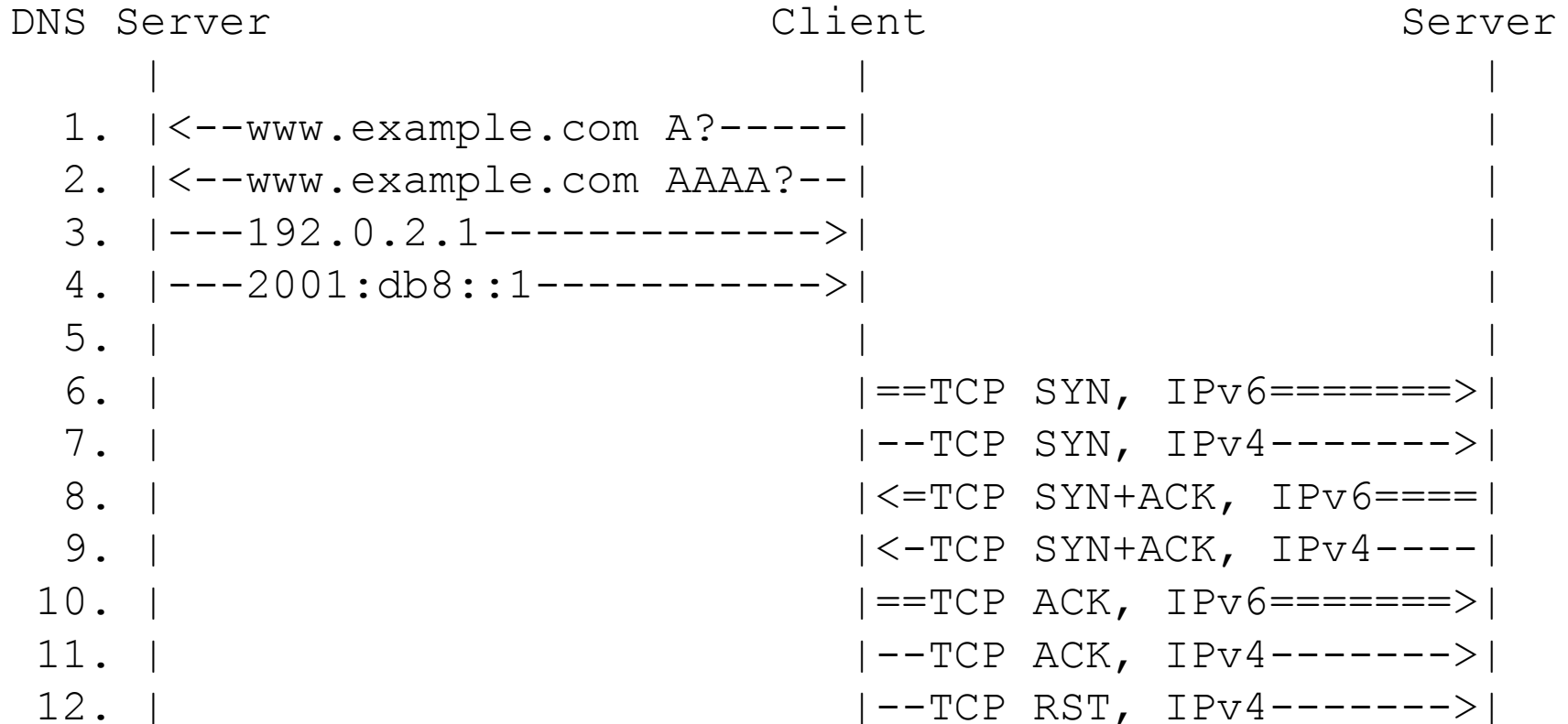


Figure 3: Happy Eyeballs Flow 2, IPv6 Working

(*1) iOS9 and EI Capitan has intentional 25msec delay to IPv4.

■ QUIC でどうなる？

- ◆ QUICは、UDPベース
- ◆ TCP RST での救済は不可。UDPベースのアプリにも、Happy Eyeballs 相当の実装がほぼ必須になると思われる。
- ◆ Squid など、QUICに対応していない Proxy サーバでは、80(UDP), 443(UDP) 宛のパケットが来たら、iptables など で port-unreach を返すことがノウハウとして紹介されている(*1)

(*1) Block QUIC protocol

<http://wiki.squid-cache.org/KnowledgeBase/Block%20QUIC%20protocol>

IPv6によくある誤解

■ IPsec は標準装備でよりセキュアに！

IPsec の実装はオプションではなく、かつて必須になっていたが、現状はIPv4と同じくオプション扱いとなった(RFC 6434)
また、実装が必須であったときでも、利用が必須であったことはない。

RFC 6434 (IPv6 Node Requirements)

Previously, IPv6 mandated implementation of IPsec and recommended the key management approach of IKE. This document updates that recommendation by making support of the IPsec Architecture [RFC4301] a SHOULD for all IPv6 nodes.

■ IPv6 は攻撃が少ないから安全

すでに IPv6 端末をターゲットにした攻撃は観測されている。
IPv4で可能であった攻撃は、その多くはIPv6でも可能。

逆に危険になるという意見もありますが、IPv4 と同じように守ることも可能。過度に恐れることはない(*1)

(*1) RFC 4942 Increased End-to-End Transparency
<https://tools.ietf.org/html/rfc4942#section-2.3>

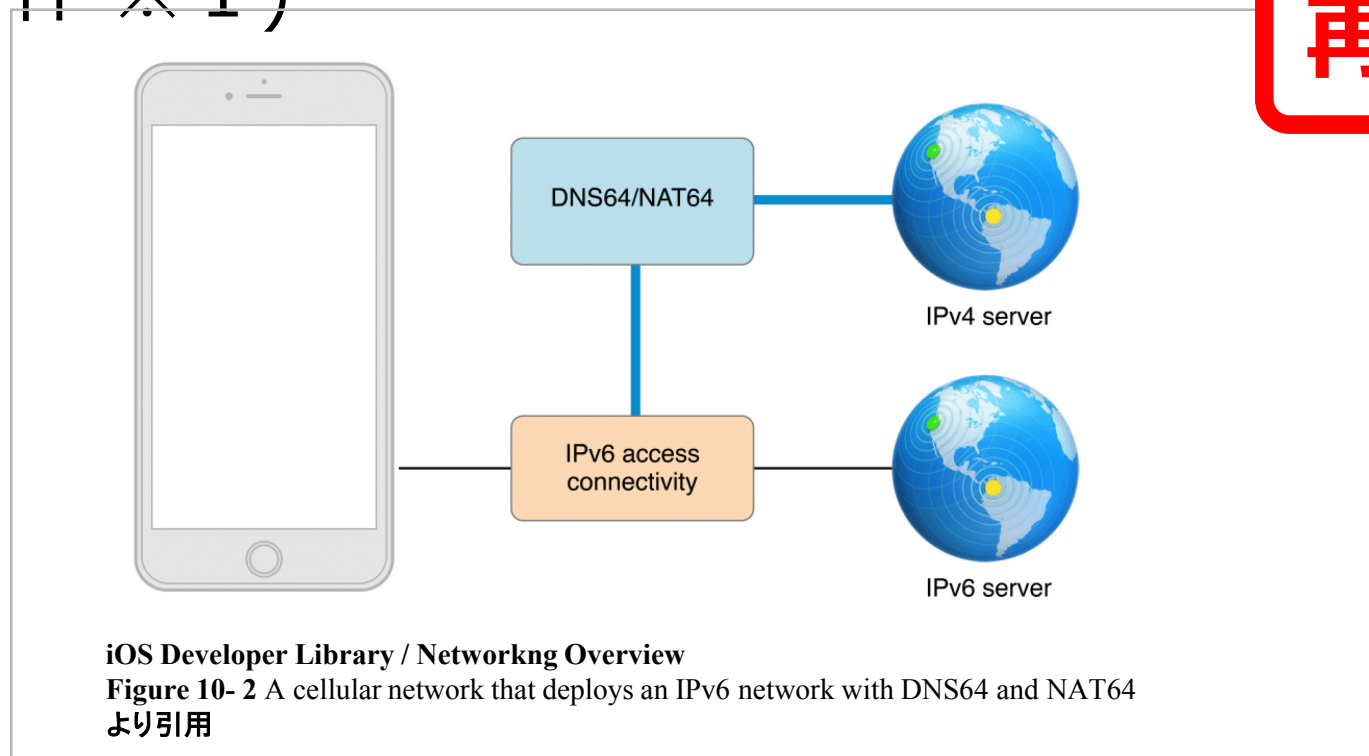
■ iOSでは IPv6対応必須！

v6only ネットワーク(DNS64+NAT64)からでもきちんと動作することが求められるようになったのは本当。
ただ、IPv6通信が必須になったわけではない(アプリ <-> サーバ通信があるようなアプリで、サーバ側のIPv6対応は求められていない。

■ IPv6 onlyでの運用も想定されつつある。

◆ DNS64+NAT64での運用 (例：iOS9のアプリ要件 ※ 1)

再掲



※1 <http://goo.gl/0USFlz>

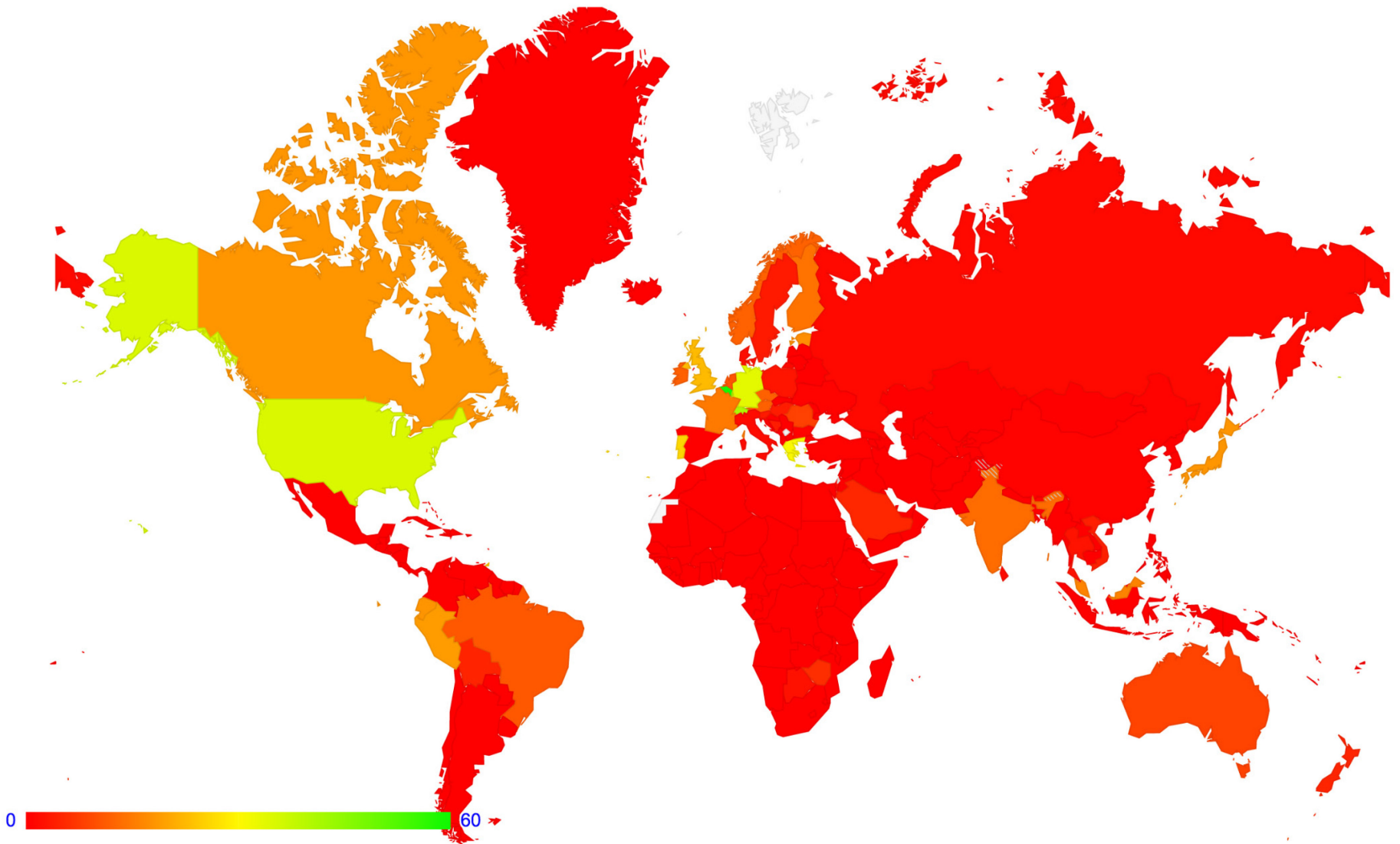
■ IPv6なんて全然流行ってないし、使われていない

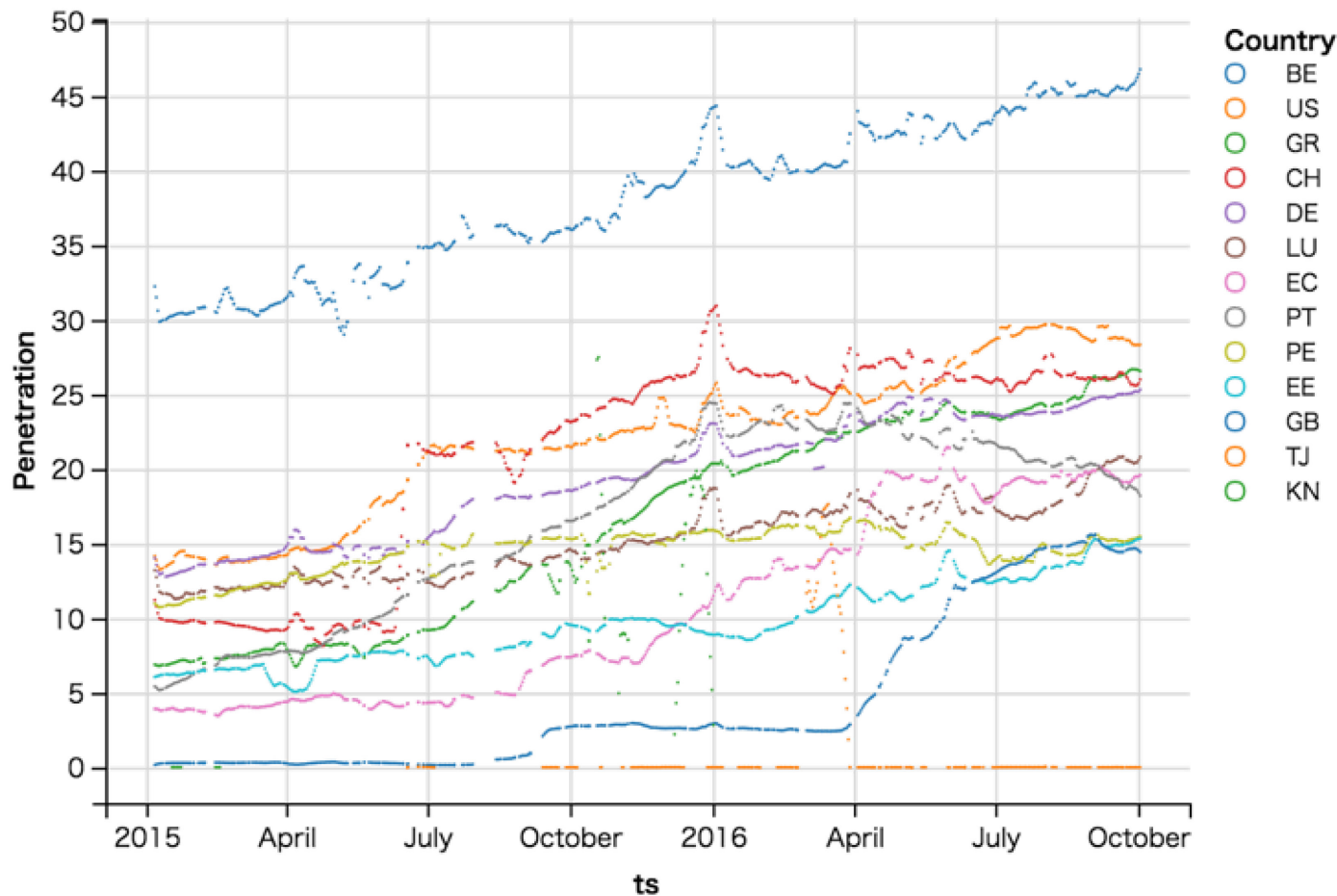
こうした実感のない日本ですら、以前からサービスの裏側でIPv6が使われている(ひかり電話, ひかりTV, VPNサービスなど)

世界的に IPv6への対応が粛々と進みつつある

IPv6 Measurement Maps

IPv6 Capable Rate by country (%)





Google's IPv6 Penetration for countries that have exceeded a minimum
cf <http://mail.coote.org/ipv6/>

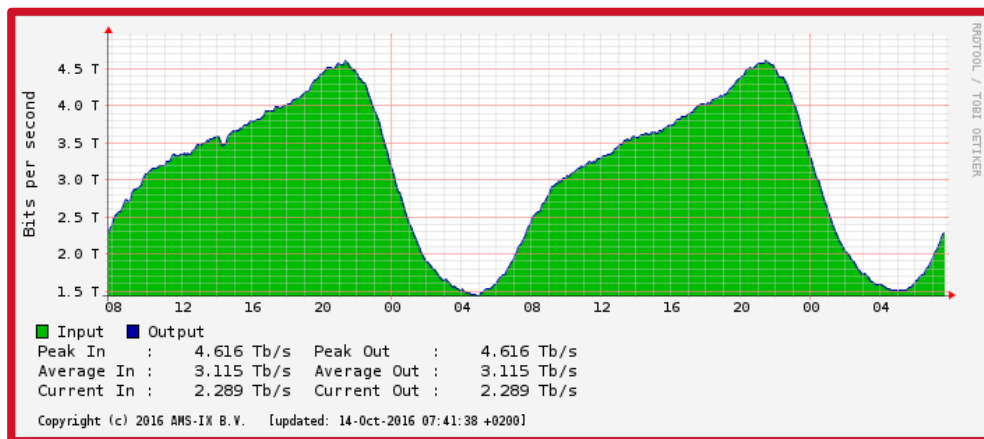
Belgium

IPv6 Deployment: **58.58%** (Prefixes : **38.63%** | Transit AS : **80.39%** | Content : **57.37%** | Users : **45.9%**)

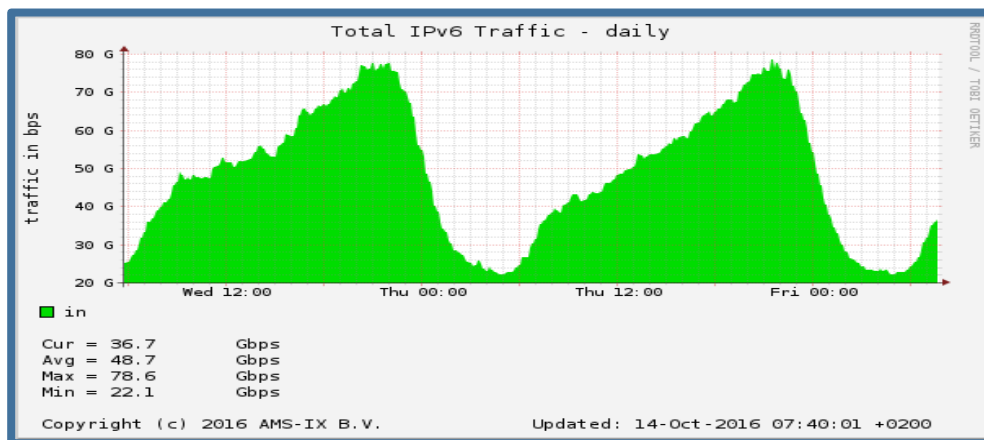
Relative Index: **10 out of 10**



cf. <http://6lab.cisco.com/stats/>



**Aggregated traffic
over 4.616Tbps**



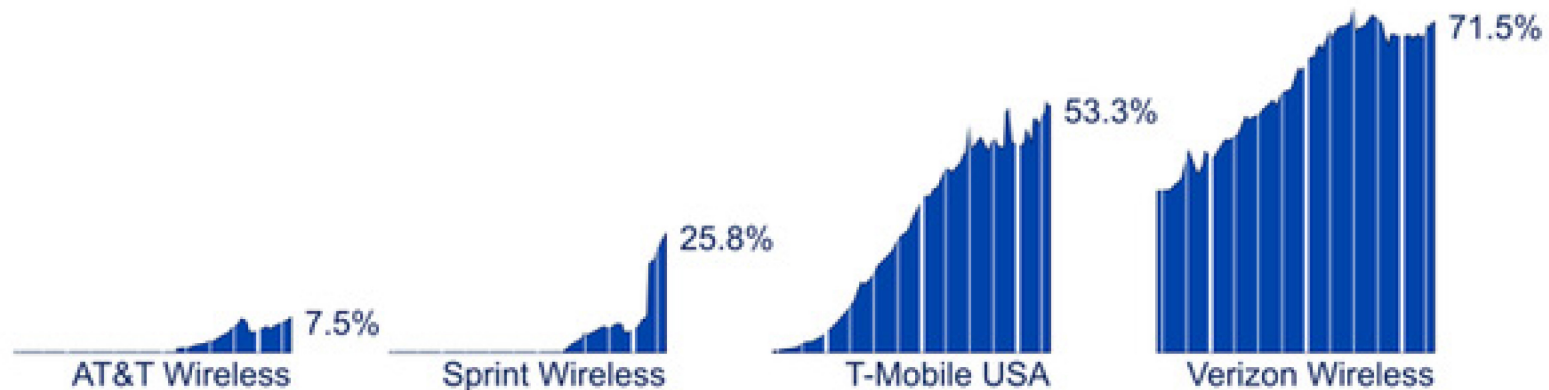
**IPv6 traffic
over 78.6Gbps**

**$78.6 \div (4.616 \times 1000) \times 100 = 1.7\%$
IPv6-based traffic is 1.7% at peak.**

- NTT DOCOMO
- au by KDDI
- Softbank

**モバイルにおけるIPv6の舞台裏と将来に向けて ～総務省
IPv6研究会を通じて～
In JANOG38 Meeting Day 2**

**IPv6 Summit in Tokyo 2016
「携帯キャリアにおけるIPv6対応最新状況」**



Percent of Requests over IPv6 to dual-stack sites on Akamai from June 2013 to May 2016

As of 10th Aug, 2016: IPv6 requests by devices: 70% Android; 30% iPhone

<https://blogs.akamai.com/2016/06/preparing-for-ipv6-only-mobile-networks-why-and-how.html>

<http://www.slideshare.net/apnic/akamai-ipv6-measurement>

http://www.theregister.co.uk/2016/08/22/ipv6_tipping_point/

News Bytes

IPv6 tipping point

22 Aug 2016 at 19:54, [Kieren McCarthy](#)



IPv6 has hit a major tipping point: it now accounts for **more than 50 per cent** of the traffic carried by US mobile networks.

That's [according](#) to the Internet Society's Mat Ford, who has been tracking the figures for the past year. Looking at the AS routing numbers for the four big mobile companies in the United States – AT&T, Sprint, T-Mobile and Verizon – there has been a solid increase in the amount of IPv6 traffic, hitting the 50 per cent mark last month and continuing to increase.

And IPv6 use is still accelerating. This is good news for engineers, who have been desperately trying to encourage take-up of the standard (which is incompatible with IPv4) for a decade. ®

<https://blogs.akamai.com/2016/06/preparing-for-ipv6-only-mobile-networks-why-and-how.html>

<http://www.slideshare.net/apnic/akamai-ipv6-measurement>

http://www.theregister.co.uk/2016/08/22/ipv6_tipping_point/

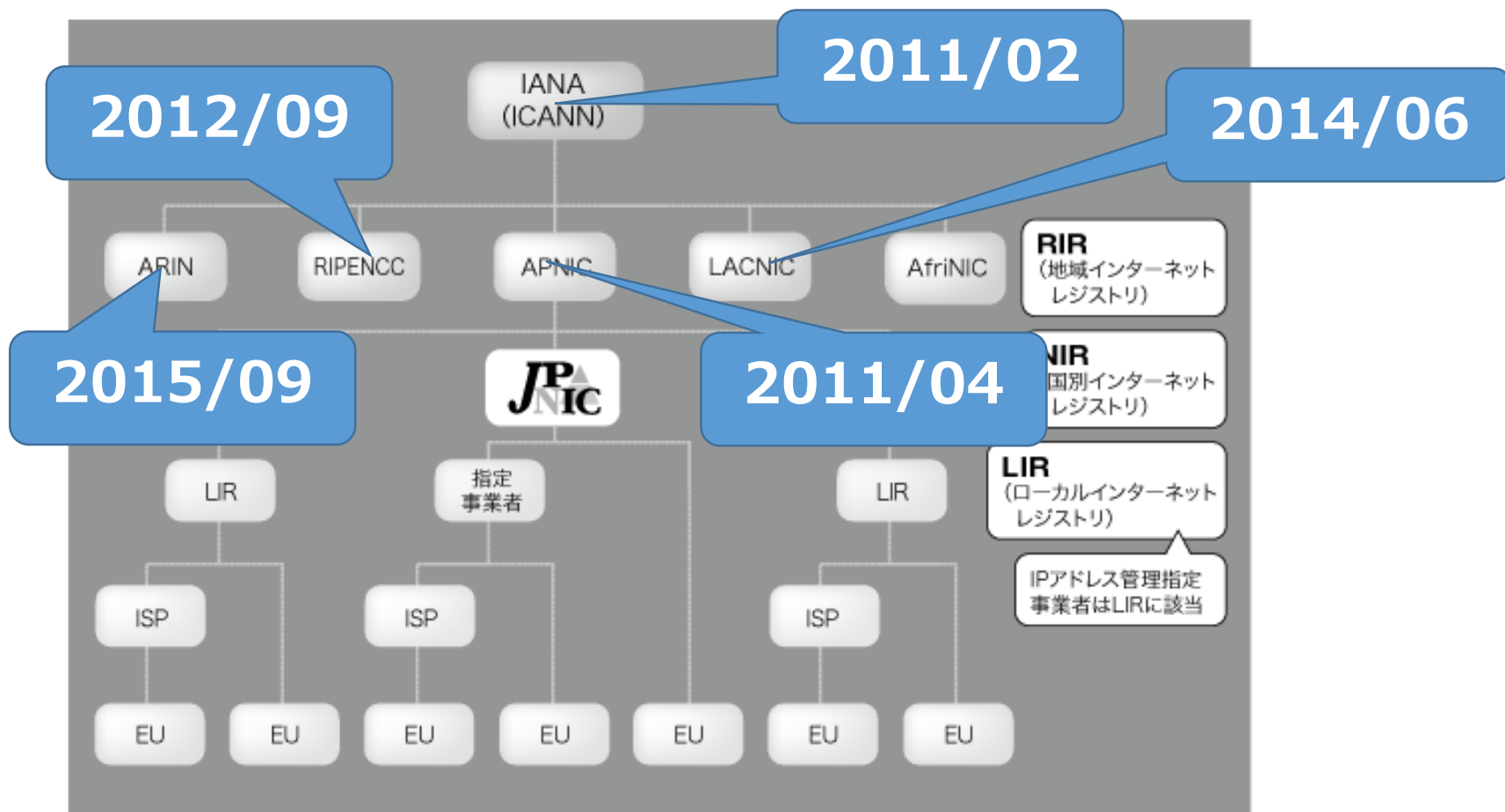
■ IPv6 ready

- Google / youtube.com
- Facebook
- Yahoo!(.com)
- Wikipedia
- Netflix
- LinkedIn
- AOL
- Apple

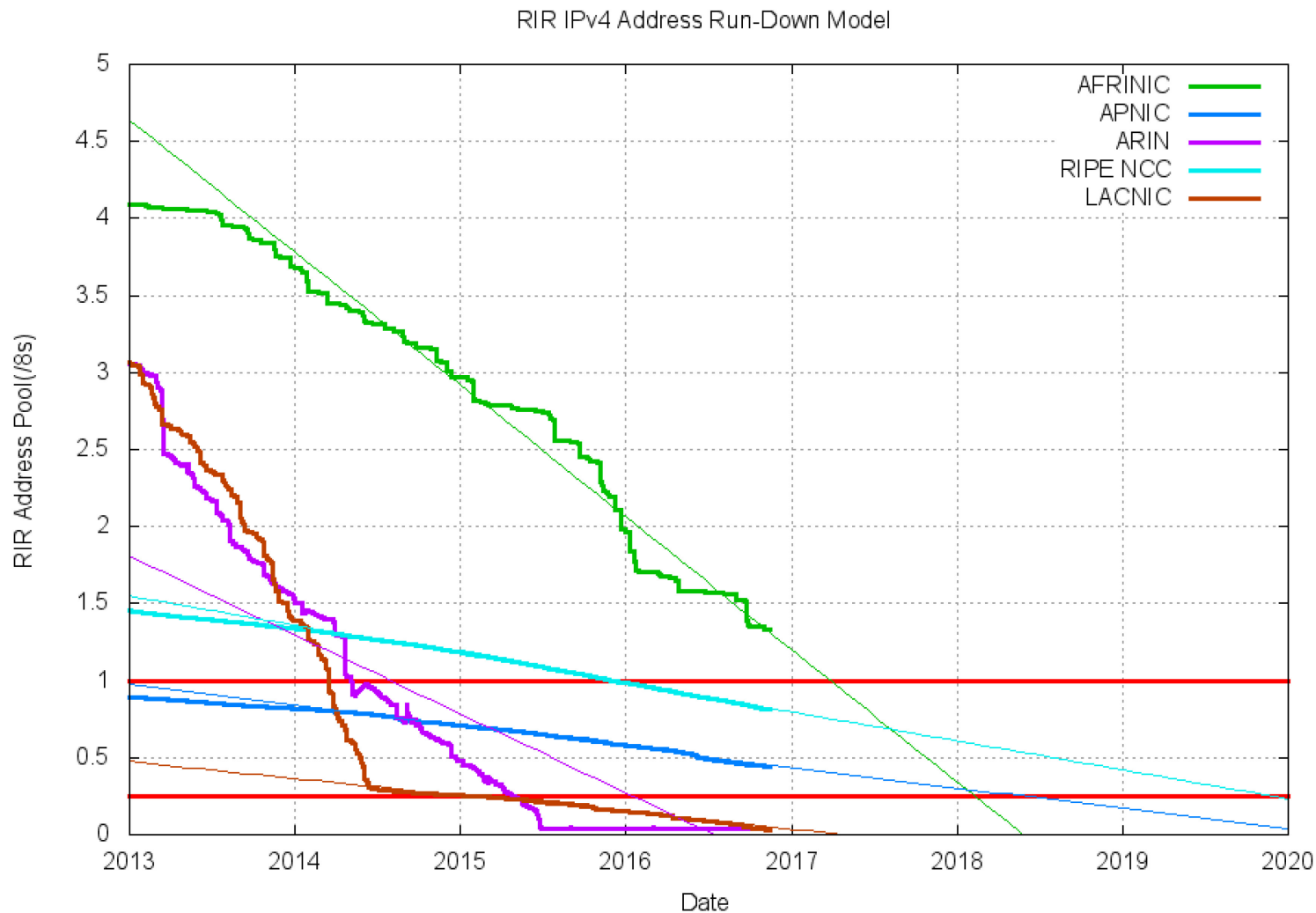
- Akamai
- Cloudflare
- Verizon Digital Media Services
(Edgecastとして知られていました)
- Limelight Networks
- CloudFront

- SoftLayer (Full Support)
- Microsoft Azure (Load Balancer/VM)
- AWS (CloudFront/WAF/S3/Route53)
- IIJ GIO
- Sakura Cloud
- Nifty Cloud (Load Balancer based)

■ IPv4アドレスは枯渇した？



”IPアドレスのポリシー策定とアドレスポリシーフォーラム”より
<https://www.nic.ad.jp/ja/newsletter/No48/0800.html>



出典: <http://www.potaroo.net/tools/ipv4/> 2016年11月現在での予想

■ IPv6移行

- ◆ 2つの文脈で受け取られることがある。
 - IPv6に対応した上での、IPv4/IPv6の共存
 - IPv6への完全移行。IPv4は使わない

IPv6トラブル事例と防止策について

■ DNS関連

■ ネットワーク

■ Path MTU Discovery Blackhole問題

■事象

- ◆支店からウェブアクセスすると早い。本店からアクセスすると、妙に遅い。

■原因

- ◆サーバの再構築後、IPv6アドレスの付与を忘れ、AAAAを残したままだった（フォールバック問題）
- ◆支店にはIPv4環境しかなく、本店には、IPv4/IPv6の接続性があり、IPv6から、IPv4へのフォールバックが発生していた。

```
;; Server
```

```
www      IN      A      192.0.2.1
```

```
IN      AAAA   2001:db8::80
```



移行前にはついていたアドレス

そもそもIPv6アドレスに対して、監視がなされていなかったのも問題。Happy Eyeballs対応のブラウザでは顕在化しづらい。

■サーバの構成変更、移行時には、そもそも既存でIPv6アドレスの利用がないかどうかをチェック

■チェックポイント

- ◆サーバにIPv6アドレスが付いていないか
 - 実際には付与の有無だけでは判断できない
- ◆FQDNにAAAA RRが登録されていないか

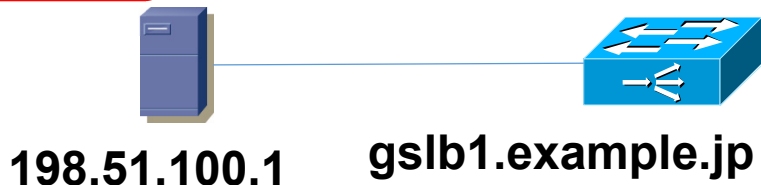
■ 事象

- ◆ よくわかんないけど、重い

■ 原因

- ◆ ある事例では、GSLBなどが、AAAAに応答せず、タイムアウトすることで、AAAAのQueryを投げるクライアントからのアクセスが結果的に遅くなる。
- ◆ 導入前に、Aレコードなどしか利用を想定していない、もしくはテストをしていない。

東京



3. www.example.jp
のIPアドレスは？

4. 192.0.2.1だよ



1. www.example.jp
のIPアドレスは？

大阪

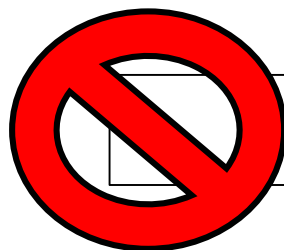


2. gslb{1|2}.example.jpに
聞いてね

Client (DNS Cache Server)

ダメな実装/設定では、AAAA RR Query を受け取った時に、下記のような誤ったレスポンスを返す。

- ◆ 一切応答しない
- ◆ NXDOMAINを返す
- ◆ AAAAレスポンスに、IPv4射影アドレスを入れて返す!!



IN AAAA ::ffff:192.0.2.1

- DNSサーバ、もしくはそれに類するシステム (GSLBなど) が、正しくAAAA RRクエリに反応できるかをチェックしましょう。

あなたが IPv6 サービスを提供している、していないにかかわらず起こり得る。

ユーザはすでに IPv6 ネットワークに接続し始めており、すでにあなたの DNS権威サーバには、AAAAクエリが届いているはず。

■ 事象

- ◆ クラウドのAPI叩いている機能を使うと重い
- ◆ sshでログインする時、妙なひっかかりがある

■ 原因

- ◆ Glibc2.6以降を使っている場合のLinuxのリゾルバの挙動と、Firewallとの総合作用

■クエリ順序はOSで異なる

◆AAAAクエリを先に実施するOS

- Windows XP、Linux

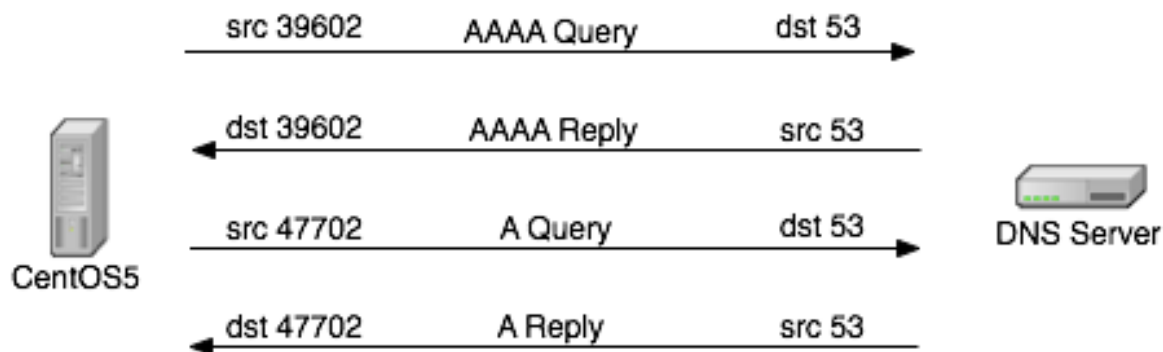
◆Aクエリを先に実施するOS

- Windows Vista、Windows 7、FreeBSD、Mac OS X

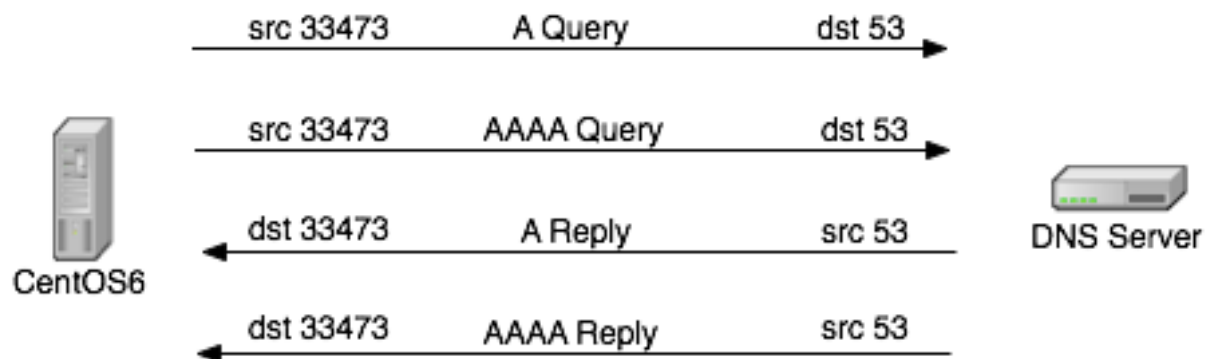
InternetWeek 2010 北口氏資料より一部抜粋 (p.64)

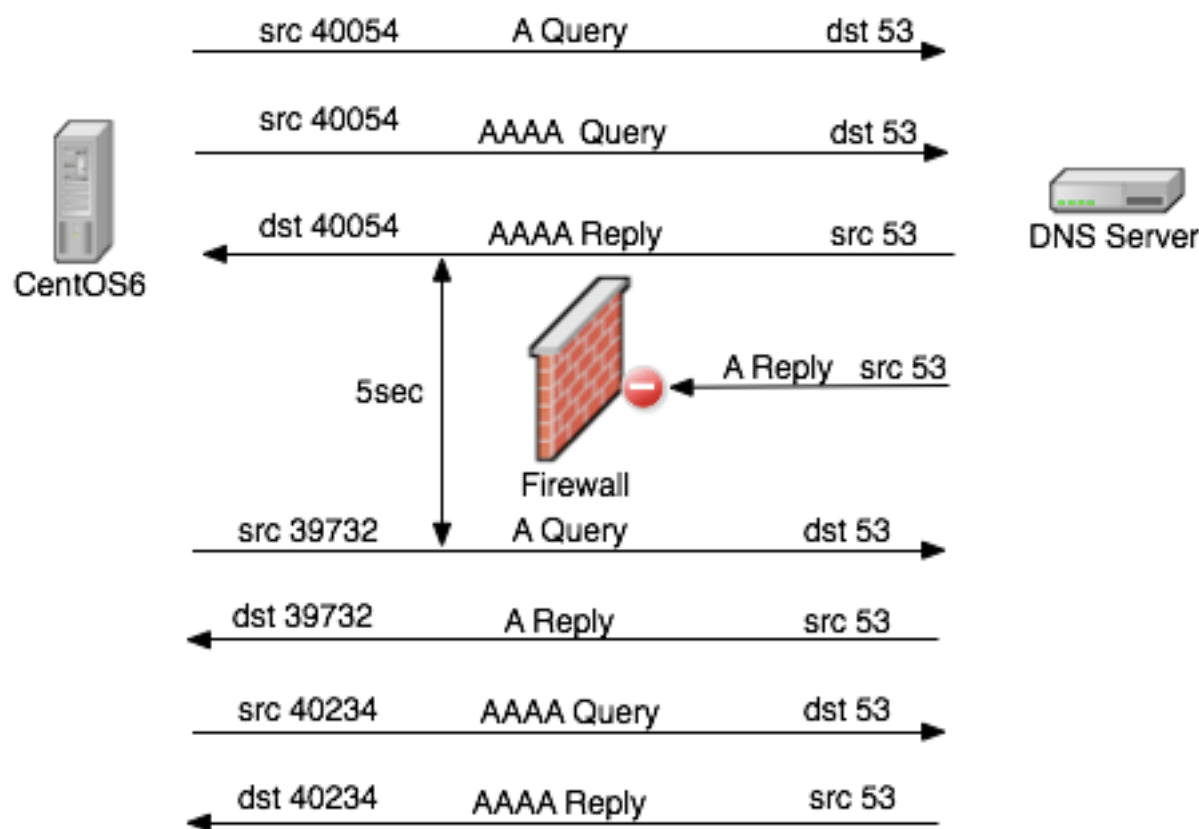
<http://www.nic.ad.jp/ja/materials/iw/2010/proceedings/s2/iw2010-s2-01.pdf>

■ RHEL5/CentOS5



■ RHEL6/CentOS6



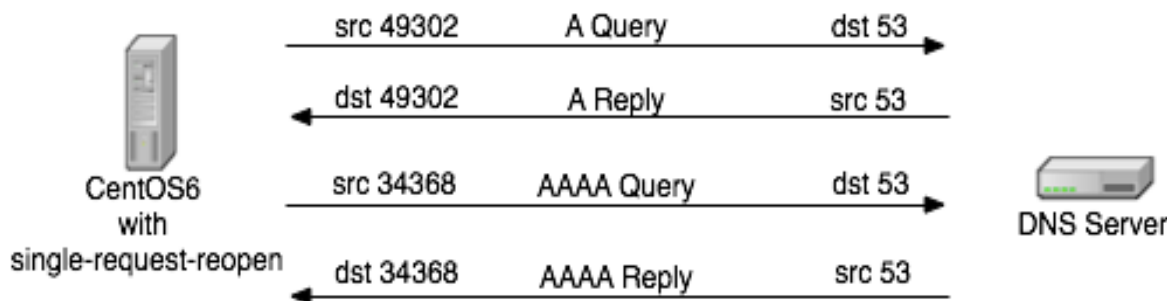


一部のファイアウォールの実装では、同一ポートからのクエリを再送(同一のセッション)とみなし、結果返信が落とされてしまうものがある

- 名前は最終的に引ける
- 若干遅いぐらい（標準設定で 5 秒でfallback)
 - ◆ options timeout:1 なら、もっと短い（そして発覚しづらい）
- 最近のサーバはAPI連携で、DNSを引くことも
 - ◆ 普通にクライアントとして使われる場合には、DNSの結果はキャッシュされないこともあり、ユーザの1リクエストに対して、複数回APIを叩くと……

- /etc/resolv.conf にオプションとして設定すると、クエリ毎にポートを変えるようになる(socketを作り直す)

```
search example.jp
nameserver 2001:db8:0001::53
nameserver 2001:db8:ffff::53
options single-request-reopen
```



■ 事象

- ◆ある日、ULAを使っているネットワークで、突然タイムアウトの嵐。

■ 原因

- ◆ULAに関する逆引きリクエストが Locally Served DNS Zonesの設定漏れで、IANA管理のサーバなどに聞きに行っていた。これが、IANA管理のDNS権威サーバの障害などで、タイムアウトを起こし障害へ発展。
- ◆Locally Served DNS Zones設定漏れに起因する障害（RFC6303）

■ Locally Served DNS Zones (RFC6303) は、この問題における良いBCPを記述している。

- ◆ 自身が管理している DNSキャッシュサーバにこれらの Zone を追加しておきましょう
 - AS112 という外部システムへの依存からの脱却
- ◆ UnboundやBind9.5 では、RFC6303 に対応しています。乗り換えなどを検討しましょう。

事象:

- iOSアプリをアップデートしたら、リジェクトされてしまった。
ちゃんと DNS64+NAT64で v6only での動作検証もしたのに！

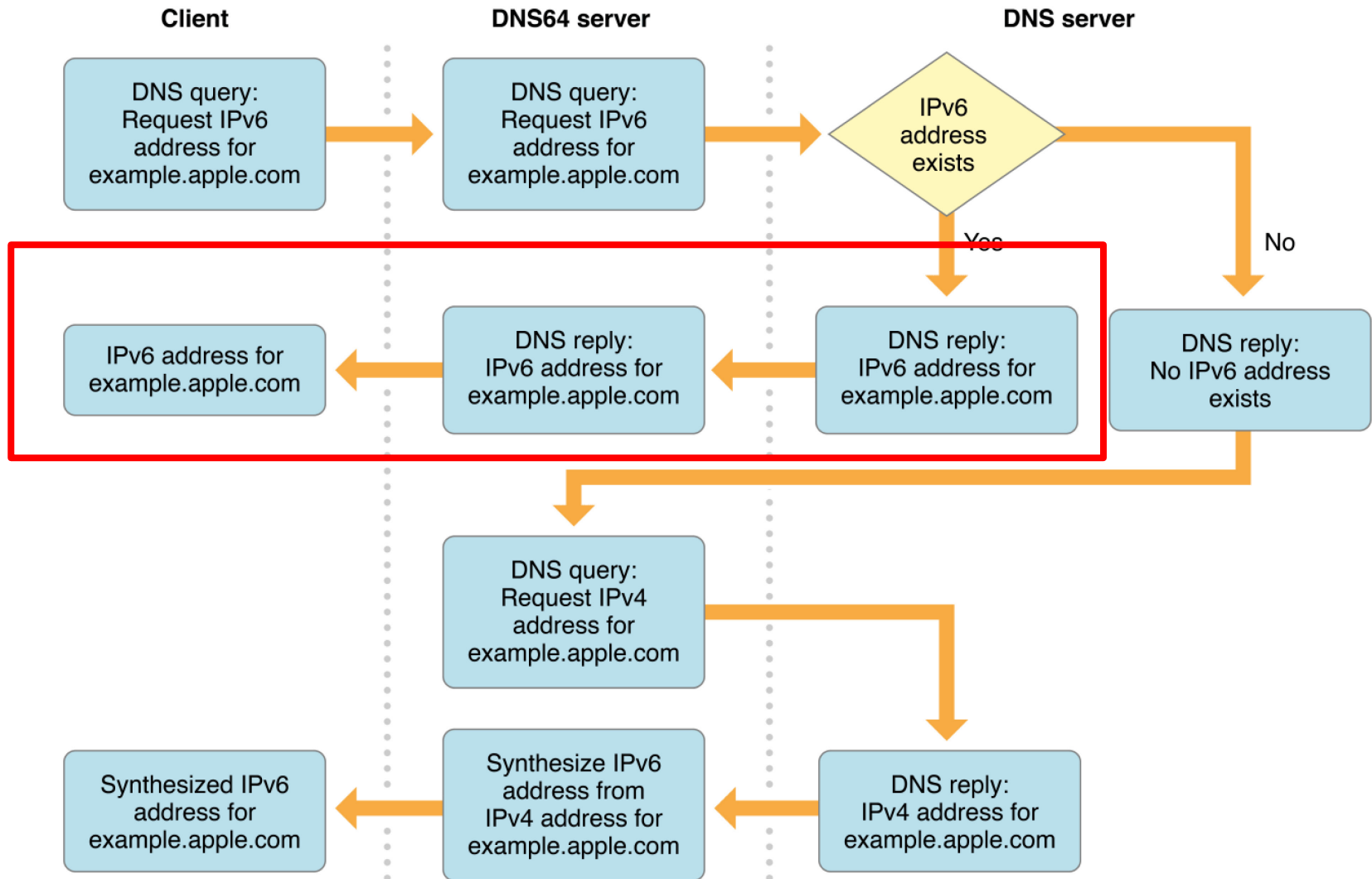
原因:

- レビューとアップルのドキュメントで記述されているテストネットワークとは、完全に同じものではありません。レビューには、実際にIPv6で外部と通信できるネットワークがあります。
- あなたのDNS権威サーバが AAAA Query に対して正しいレスポンスを返しているか確認しましょう。
- AAAAに応答するときに、IPv4射影アドレスを埋めて返すのはダメです。



IN AAAA ::ffff:192.0.2.1

Workflow of a DNS64/NAT64



- /etc/com.apple.mis.unbound.conf

```
dns64-synthal:yes
```

実際のAAAAにかかわらず、AアドレスをAAAAに変換して応答します。

■ dns64-syanhal:yes の実例

```
# dig @localhost www.nic.ad.jp AAAA +short  
64:ff9b::c029:c081
```

■ dns64-synthal: no の実例

```
# dig www.nic.ad.jp AAAA +short  
2001:dc2:1000:2006::80:1
```

ケース2と全く同じです

■ 設定不備がほとんど

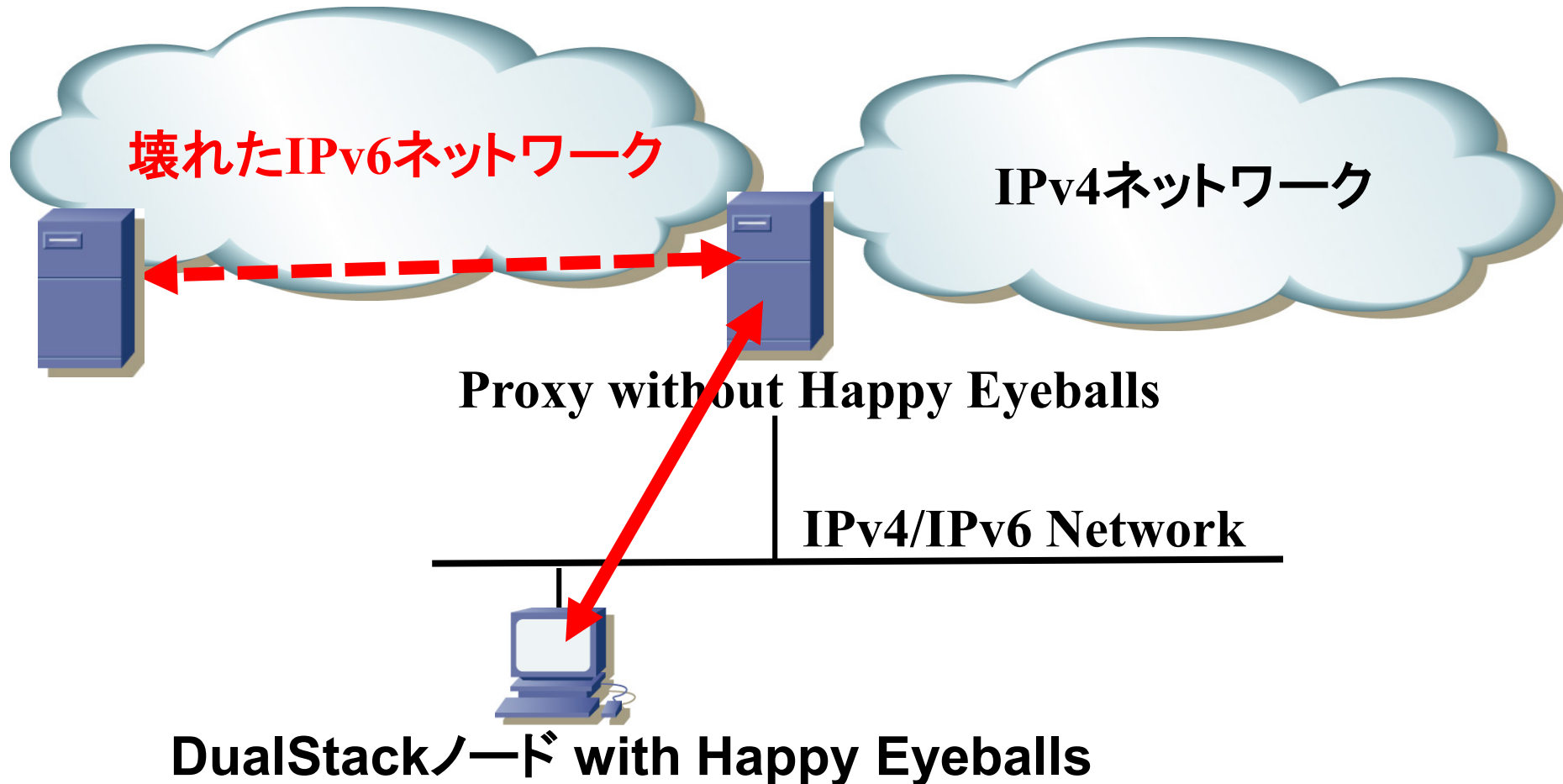
- ◆ DNS権威サーバおよびDNSキャッシュサーバに対する知識の欠如
 - メーカー側、ユーザ側
- ◆ IPv6サービスを提供していることが共有されない、またされ続けない。
- ◆ IPv6でのサービスレベルが、IPv4のものとは比べて、低くなってしまっている（積み重なっている運用経験が、活かされない）
- ◆ 単純なテスト不足

- DNS関連

- ネットワーク

- Path MTU Discovery Blackhole問題

- Proxyは容易にHappy Eyeballs Blockerとなる
 - ◆ これらはTCPを終端し直す



- Squidは現時点で、Happy Eyeballs をフル実装する予定はないと表明している。
- Anti-Virusソフトウェアで、Proxy ベースの実装があるため、該当してしまわないかに注意

■ Firewall内蔵のAnti-Virus Softwareには注意！

- ◆ 意図的に標準でIPv6トラフィックをすべて遮断しようとするものが存在（しかも実際は遮断できていなくて、問題を引き起こす）

Ciscoでは、default でRAを投げます

RAを完全に抑制するには

```
ipv6 nd ra supress
```

だけでは足りなくて

```
ipv6 nd ra supress all
```

と書く必要があります。

“all” をつけないと、定期的なRA送出手は抑制されるものの、RS(Router Solicitation)を受け取ると、RAを送出してしまふ(*1)

従来の実装では、RSは基本的にnodeが I/F を up した時にしか流れない(*2)ので、このRAでついたアドレスを利用すると、valid lifetimeが過ぎたあとに通信ができなくなる危険性がある（なにも設定変更していないと、30日後）

(*1)http://www.cisco.com/c/en/us/td/docs/ios/ipv6/command/reference/ipv6_book/ipv6_07.html

(*2) 最近ではRFC7559ができた

RFC4861 6.3.4

Prefix List - A list of the prefixes that define a set of addresses that are on-link. Prefix List entries are created from information received in Router Advertisements. Each entry has an associated invalidation timer value (extracted from the advertisement) used to expire prefixes when they become invalid. A special "infinity" timer value specifies that a prefix remains valid forever, unless a new (finite) value is received in a subsequent advertisement.

影響を受けるかどうかは、実装に依存する

- 古いIOSでは “all” がなく、下記のようなACLをI/Fに適用することで、フィルタリングする必要がある。

```
ipv6 access-list RS-Filter  
deny icmp any any router-solicitation  
permit ipv6 any any
```

下記より引用

シスコサポートコミュニティ

「IPv6 RAの Suppress動作について」

<https://supportforums.cisco.com/ja/document/100306>

- DNS関連

- ネットワーク

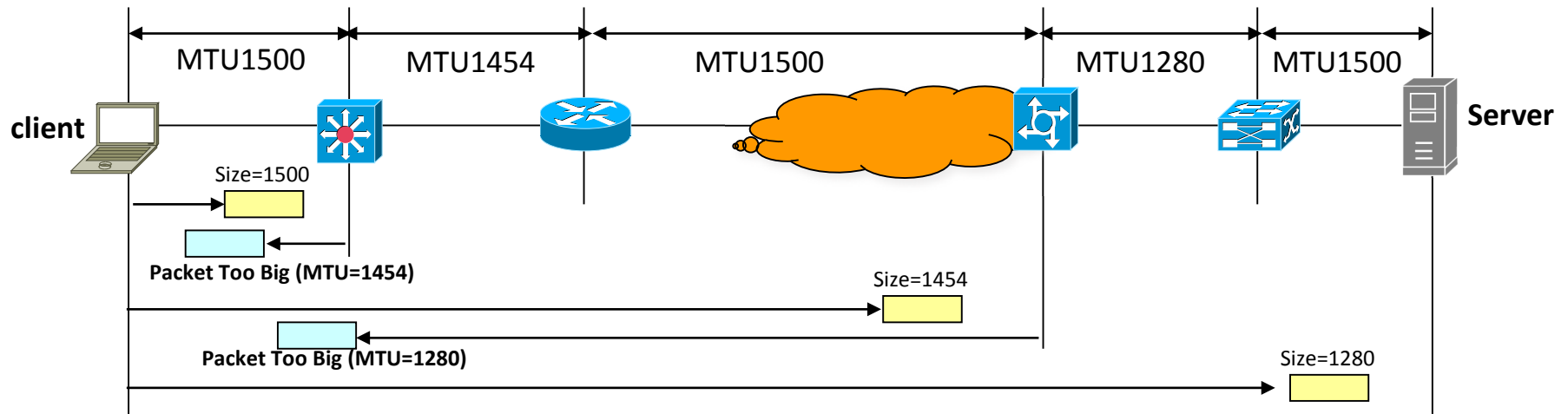
- Path MTU Discovery Blackhole問題

- バグに起因しないネットワークトラブルのほとんどが、 Path MTU Discovery Blackhole 問題

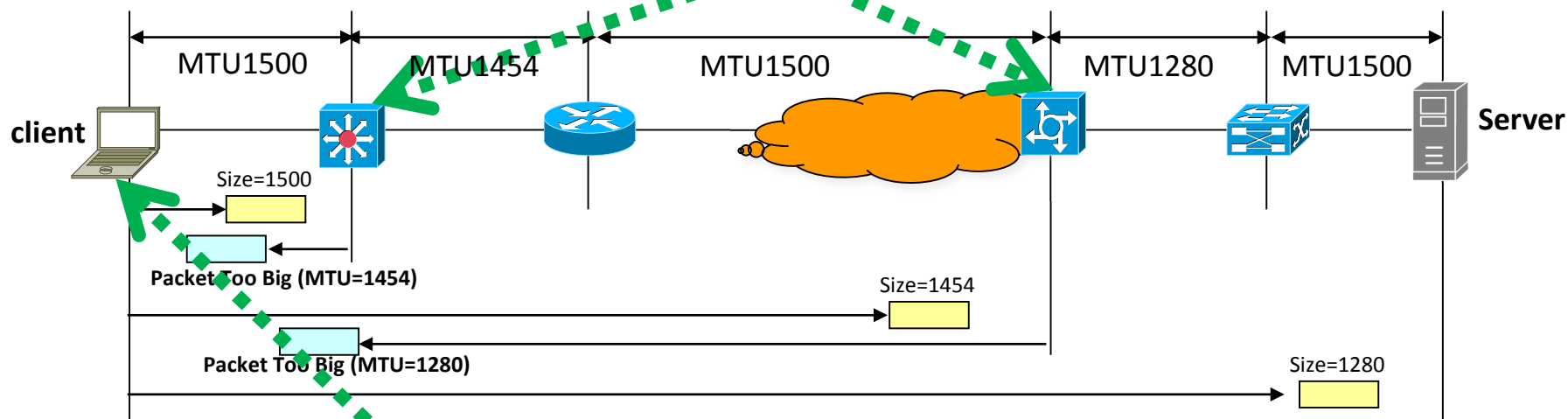
Path MTU Discoveryおさらい

- IPv6 では中継ノードでフラグメントしない(始点ノードが実施)
 - IPv4 ではルータ等の中継ノードがフラグメントを実施
 - 送信パケットに対する ICMPv6 Error Message を受信時、MTU を変更
 - 最初のリンクのMTU が初期値
 - ICMPv6 Packet Too Big Message 受信時、始点ノードでフラグメントして再送
 - IPv6最小MTU は、1280byte
 - L2 SWのMTUにひっかかった場合は破棄される
 - Path MTU Discovery の実装が難しいノードは 1280byte 固定

Path MTU Discovery とは？ (2)



**Too big作る人！
(転送先のMTUが小さい)**



**Too bigを受け取る人！
(大きなデータを送ってるノード)**

1. Too big パケットが作れない
2. Too bigパケットが受信・転送できない

■コンテンツを送信する側

- ◆ウェブサーバ
- ◆メール送信者(大きな添付ファイルとか)
- ◆Dropbox的ななにか

- IPv6のパケットは、IPv4で言えば、すべてDFビットが立っているパケット。つまり経路上のルータがパケットをフラグメントすることは禁止されており、PMTU Discoveryが動作することによるパケットの再送を期待している。

Too bigが届かないと、通信ができない！

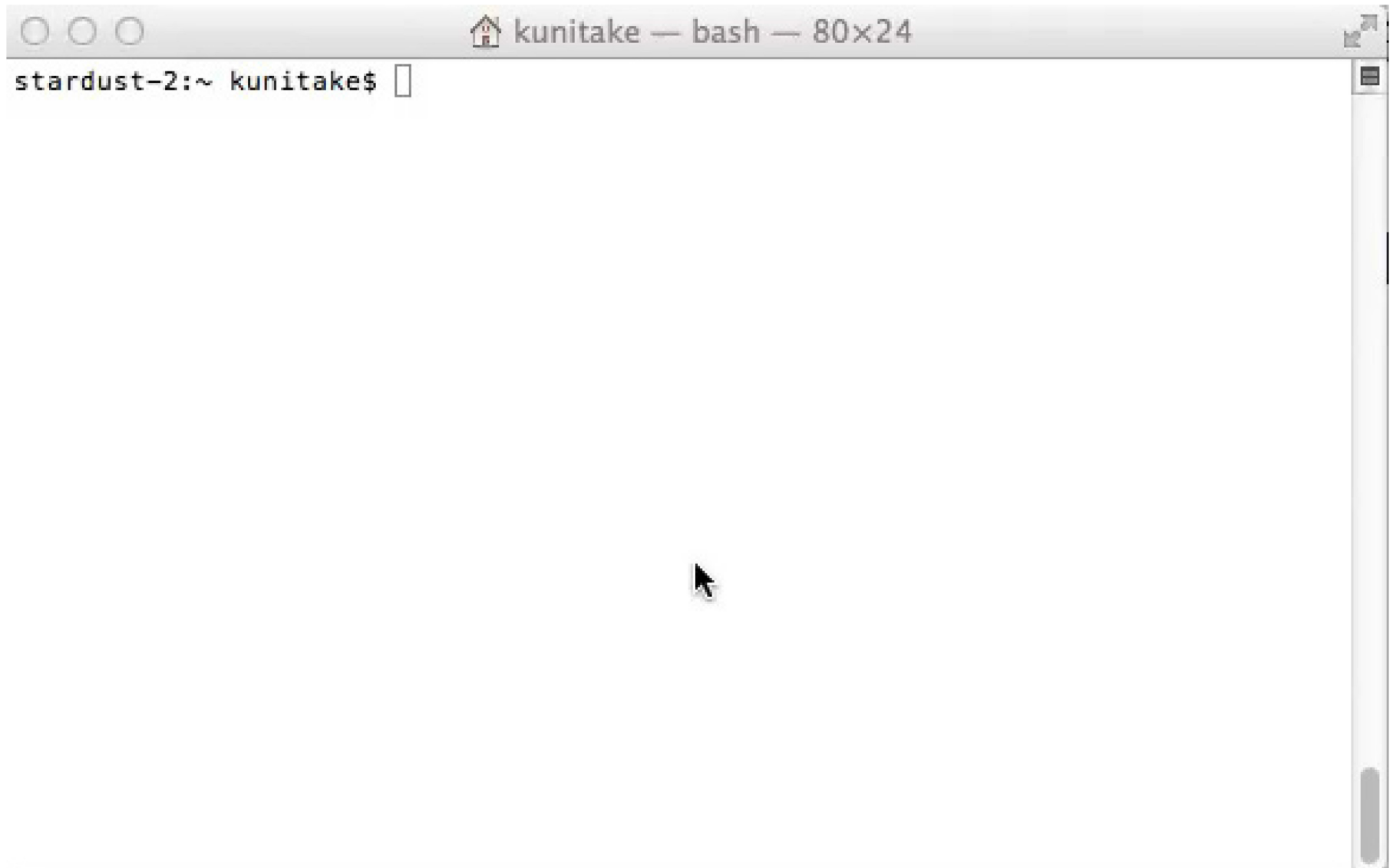
※なおTCPのセッションは張れるため、前述したHappy Eyeballsでは対処できない

Too bigを必ず届ける、受け取る！

or/and

Too bigを発生させない！

がIPv6通信には必須



A terminal window with a title bar containing three window control buttons on the left, a home icon, the text "kunitake — bash — 80x24", and window control buttons on the right. The terminal content shows the prompt "stardust-2:~ kunitake\$" followed by a cursor. A mouse cursor is visible in the center of the terminal area.

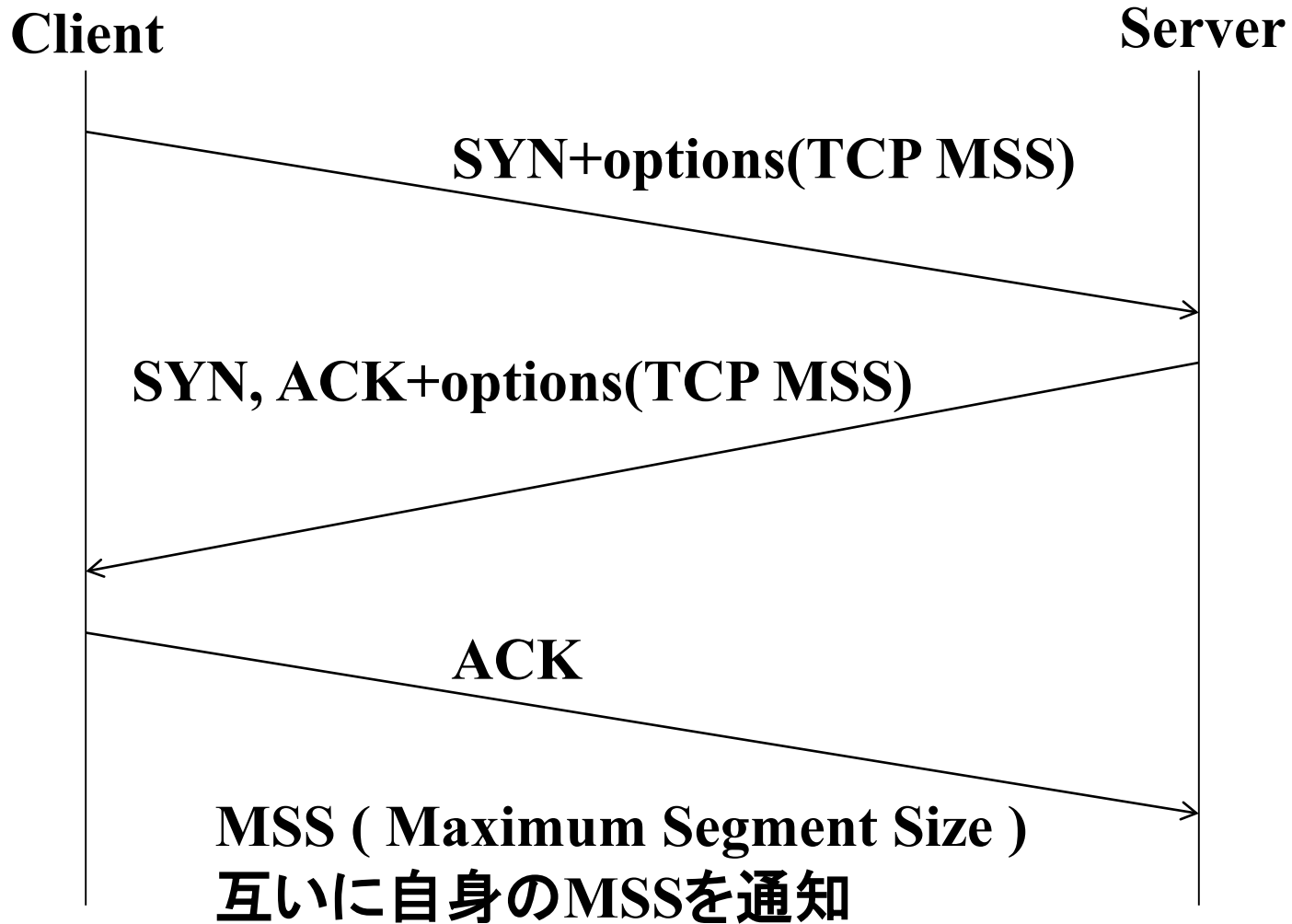
```
stardust-2:~ kunitake$
```


■ MTUが小さくなる環境構築

◆ クライアント環境で以下を実行しテスト

```
# ifconfig en0 mtu 1280
```

**意図的にTCP MSSによる調整が発生させる。
これで問題解消する場合は、
Path MTU Discovery Blackholeが原因**



■L3's icmp rate limit

- ◆L3装置では、ICMPに関してレートリミットがかかっているものがあり、リミットを超えると Too bigを返せなくなる

■Firewall Policy

- ◆本来通信に必要なICMPv6パケットまで落ととしてしまって、通信障害を発生させてしまっていないか

■LB構成でToo bigがちゃんとエンドに届くか

■ネットワーク構成

- ◆Anycastを利用していた場合、too bigが適切なサーバに転送されるか

- 頑張っちゃって link local addressのみでネットワーク作り、**かつ**異なるMTUサイズを混ぜていない(*1)



RFC4291: Routers must not forward any packets with link-local source or destination addresses to other link.

- IPS/UTMやステートを見ないパケットフィルタリング

(*1) Some Design Choices for IPv6 Networks

<https://tools.ietf.org/html/draft-ietf-v6ops-design-choices-12>

- 実はFirewallのポリシーでは、事実上、Too big は落とせない（フロー上、自動的に許可される）

| | Service | Action |
|---|-----------------------------------|---|
| 8 | ICMP6 Packet Too Big ICMP6-ANY |  |
| | ANY |  |

なんと、こんな設定書いてもToo bigは落とせない...

-A INPUT -m state ¥

--state ESTABLISHED,RELATED ¥

-j ACCEPT

RELATED

*meaning that the packet is starting a new connection, but is associated with an existing connection, such as an FTP data transfer, or **an ICMP error.***

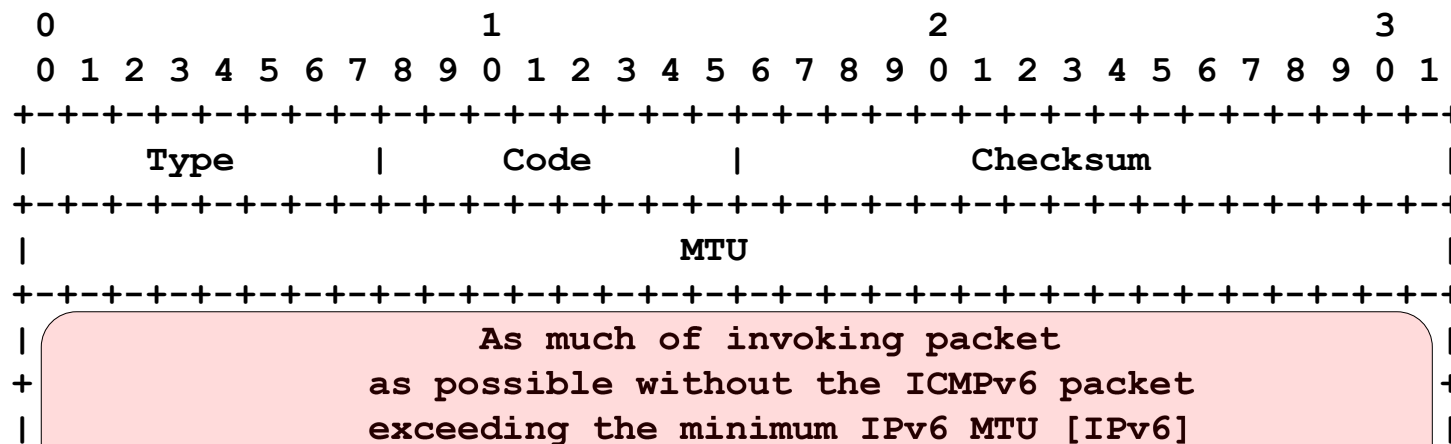
[00001] 2014-10-17 21:00:00 [Root]system-critical-00436: Large ICMP packet! From 2001:db8:ffff::117 to 2001:db8::80, proto 58 (zone Untrust, int ethernet0/1). Occurred 6 times.

■[ScreenOS] Large Size ICMP Packet (size > 1024) in IPv6 environment.

<http://kb.juniper.net/InfoCenter/index?page=content&id=KB26473&actp=RSS>
(<http://juni.pr/QJCruH>)

多くのICMPパケットは大きくないため、1024バイト以上のICMPパケットを攻撃パケットやLoki (ICMP Tunnel)の通信などとみなす。

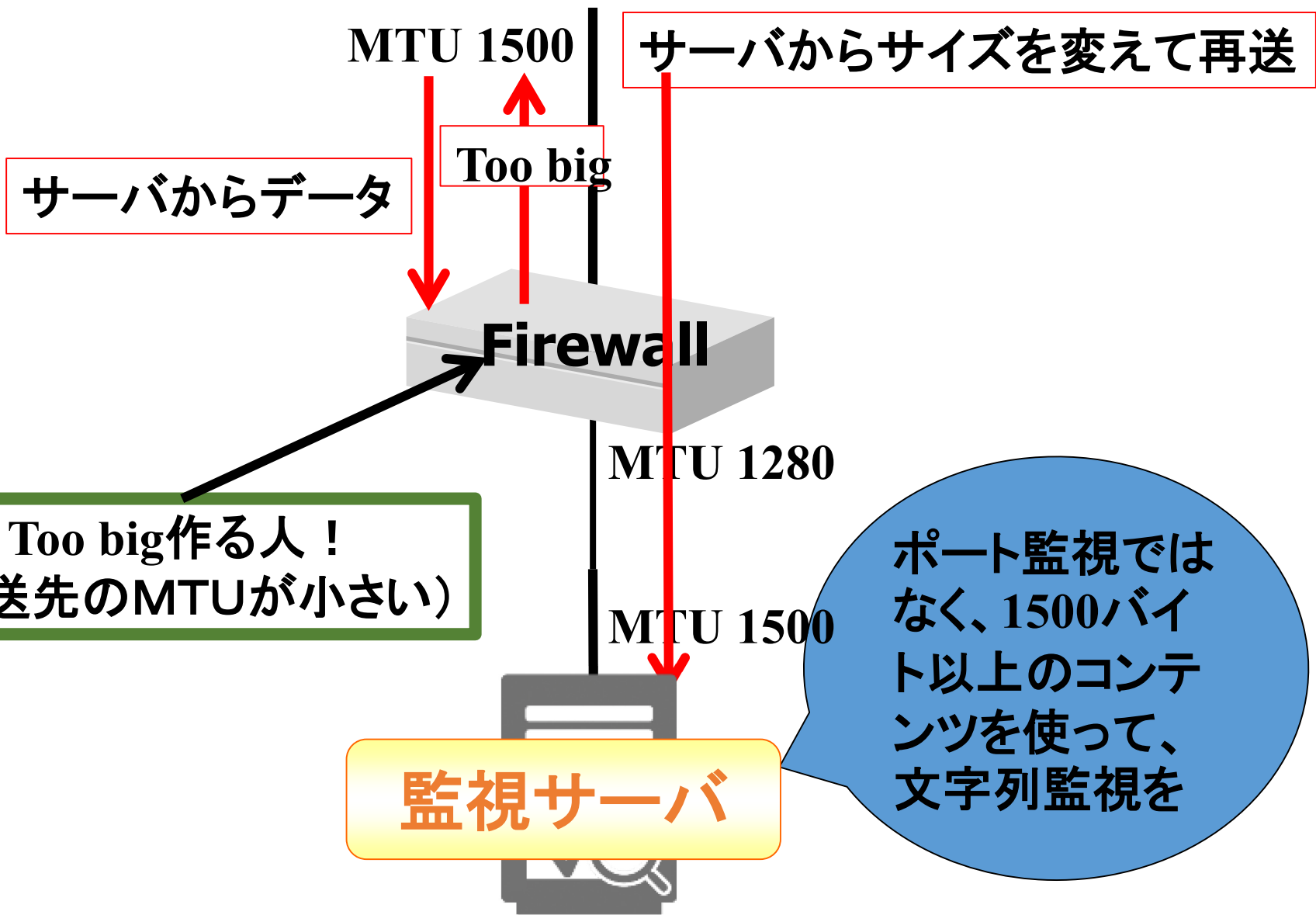
3.2. Packet Too Big Message



1280byte以下であることは仕様で決まっているが、どこまで
 パケットを頑張って詰め込むかは、実装依存

■ Path MTU Blackhole問題対策

- ◆ サーバセグメントのMTUよりもバックボーンのMTUを小さくしない
- ◆ Too big を適切に転送できる構成であるかに注意する。できないなら、サーバのMTUは1280とするか(UDPを利用していないケース)、使っていないなら、適切にtoo bigを転送できるネットワーク構成に変更
- ◆ ステートをみないパケットフィルタリングを利用しているのであれば、Too bigを通すように設定する。



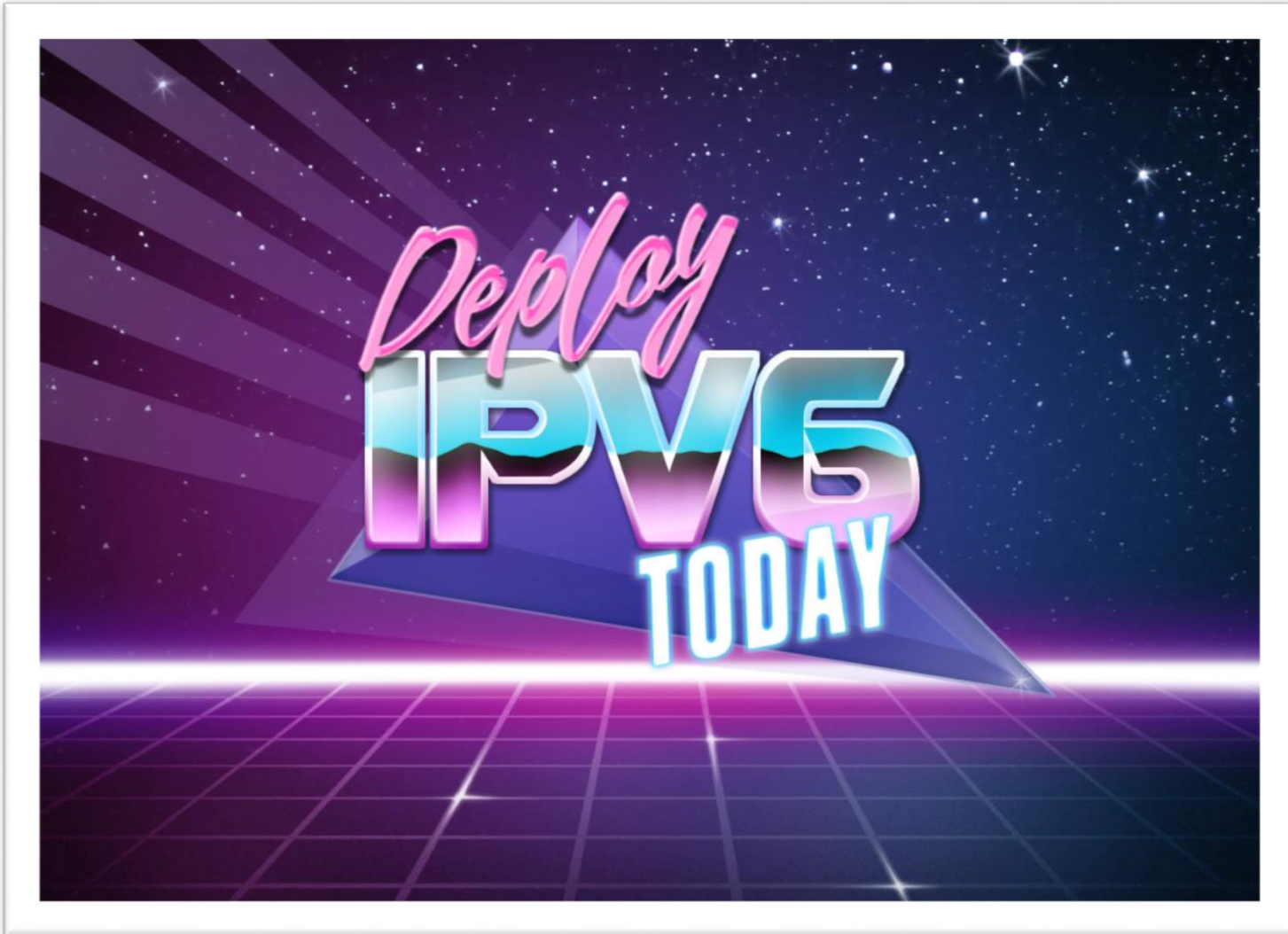


Image source: <https://www.facebook.com/photo.php?fbid=10103868052428748&set=gm.10154445520620540&type=3&theater>