

Internet Week 2014 T3

IPv6ネットワーク運用とセキュリティ

スカイリンクス株式会社

技術本部

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(90分)
主にサーバセグメントについて解説します。

※IPv6 onlyネットワークについては取り上げません

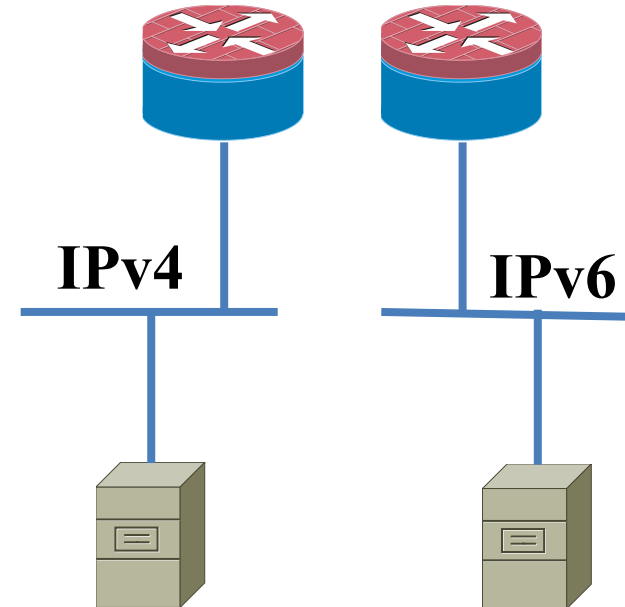
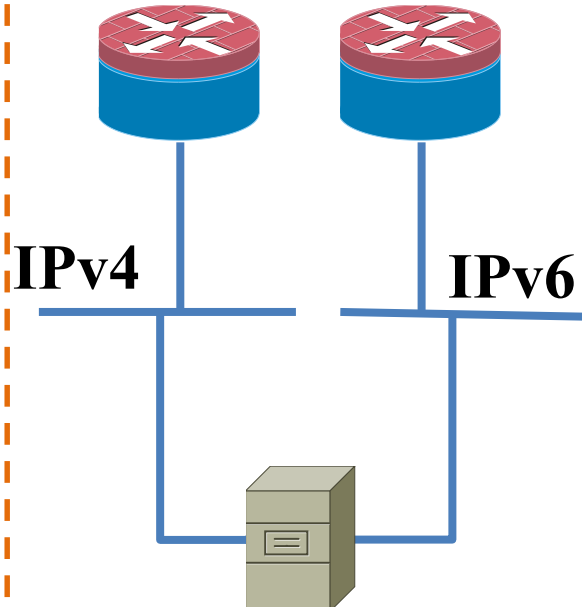
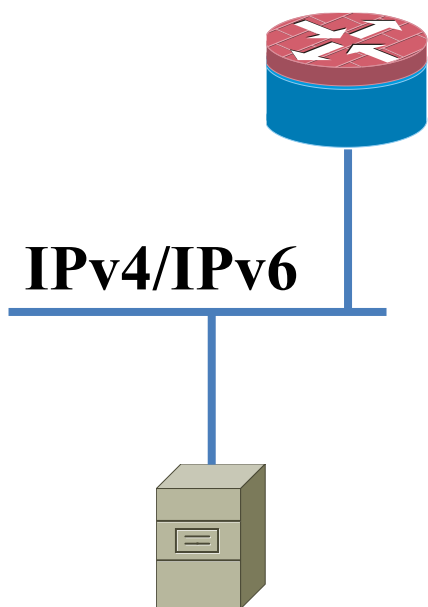
- IPv6ネットワーク概要（基本）
 - 構成例について
 - DualStack
 - Fallbackについて
- IPv6トラブル事例および防止策
 - DNS関連
 - Path MTU Discovery Blackholeその原因
 - 構築時の注意点

IPv6ネットワーク概要(1)

- 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

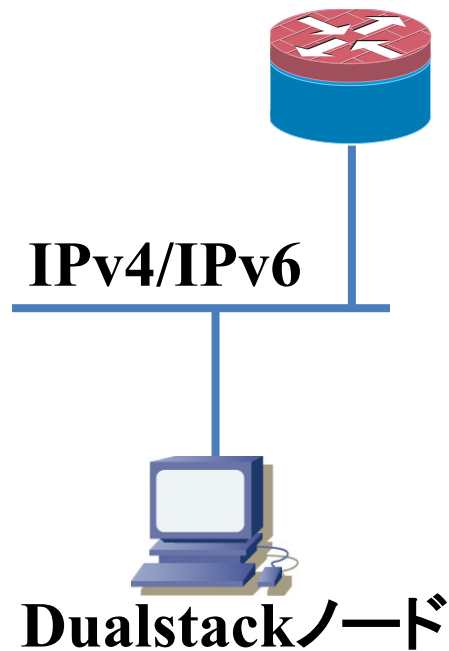
コスト低

コスト大



※Tunnelは除外してます。

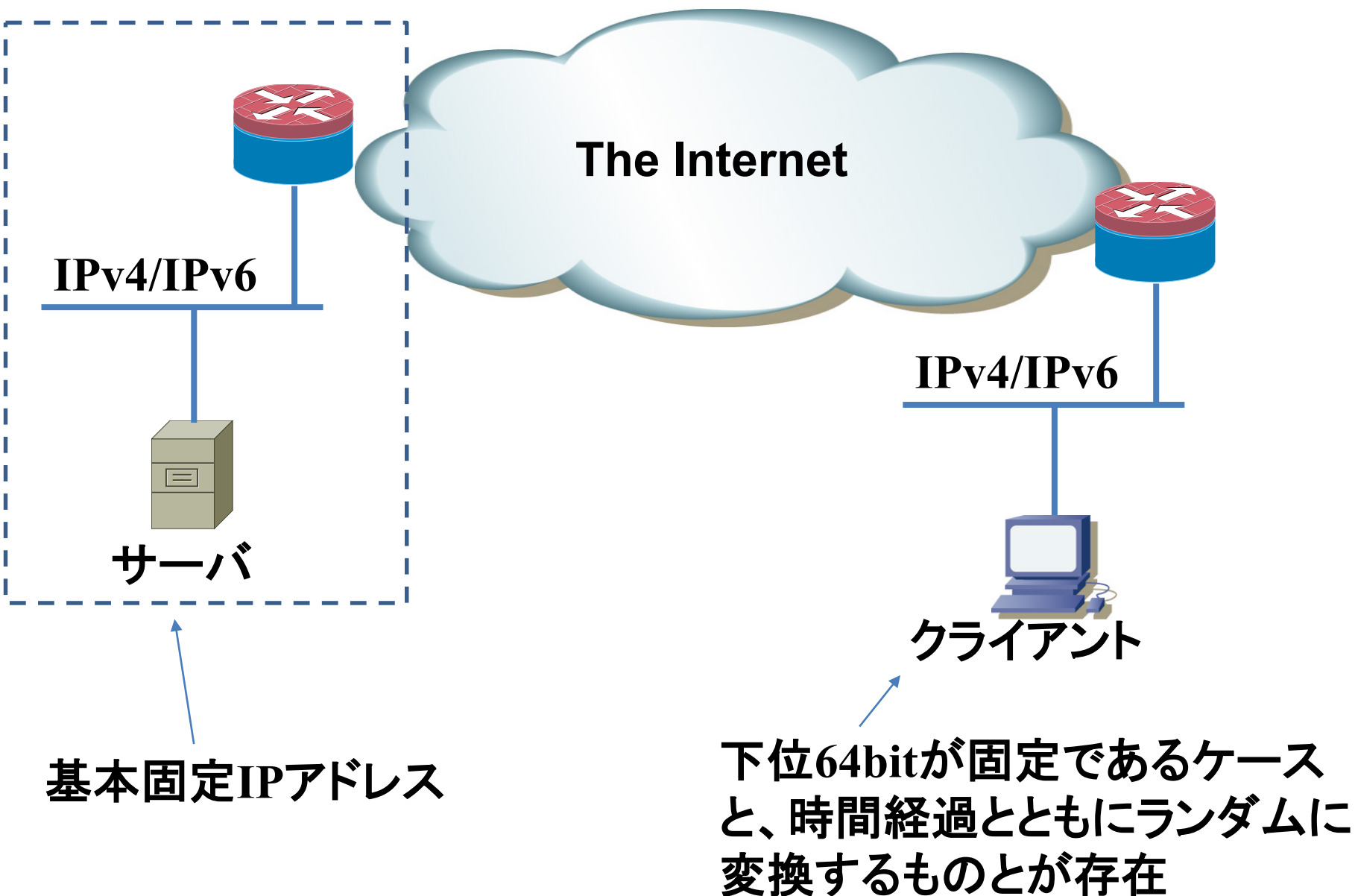
- クライアント側でのIPv6は、DualStackでの対応が多数。
- クライアントに割り当てられるIPv6アドレスはグローバルアドレスが割り当てられることが多い。



- 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。



- 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
```

IPv6 -> IPv4へのフォールバック

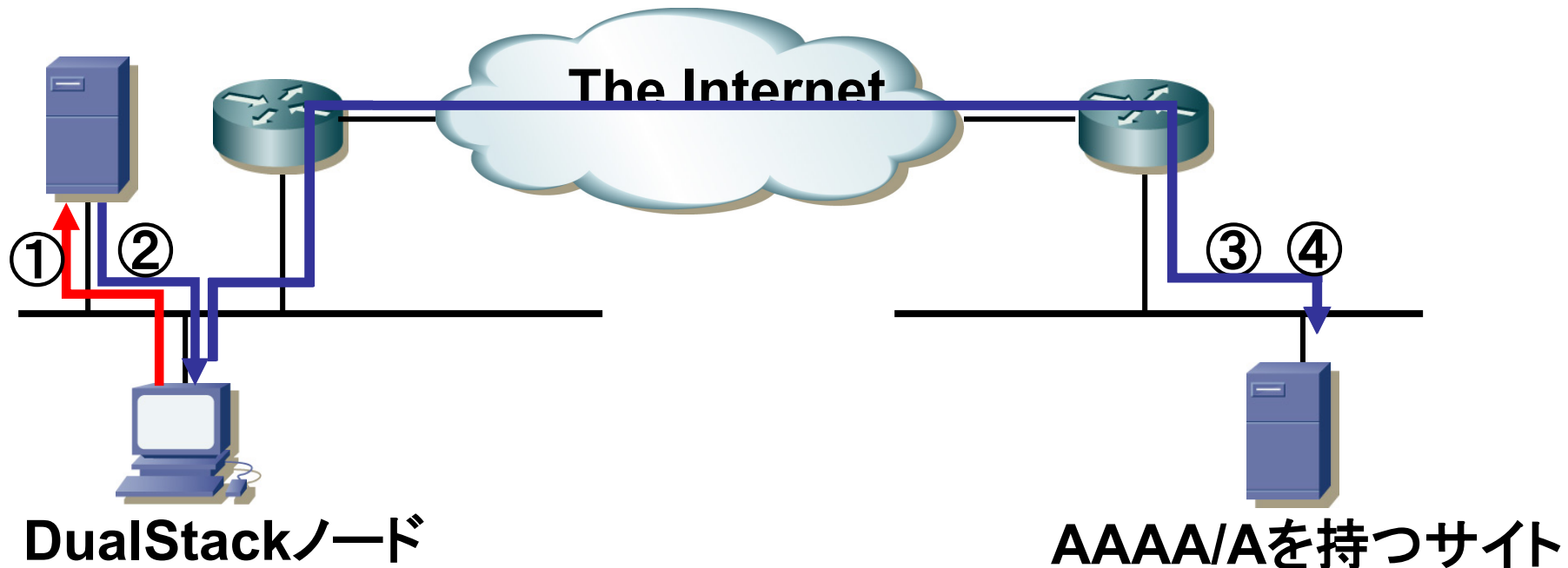
①FQDNの名前解決を行う

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

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

- 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

- 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

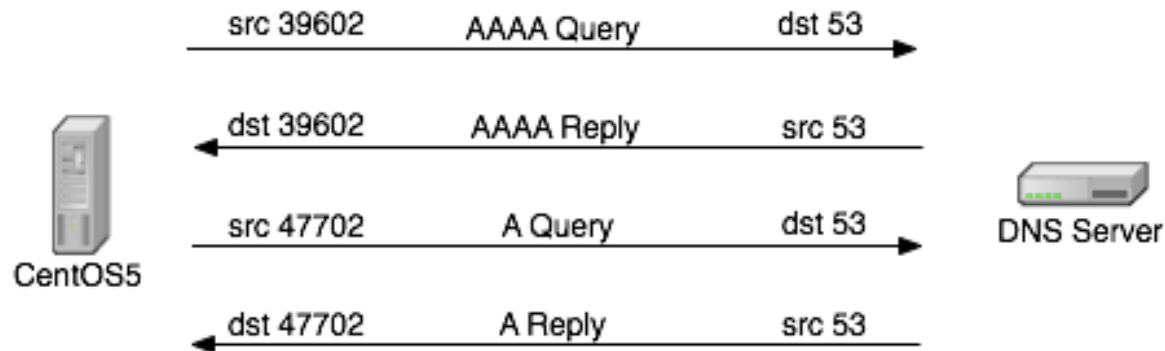
- クエリ順序はOSで異なる
 - AAAAクエリを先に実施するOS
 - Windows XP、Linux
 - Aクエリを先に実施するOS
 - Windows Vista、Windows 7、FreeBSD、Mac OS X

InternetWeek 2010 北口氏資料より一部抜粋 (p.64)

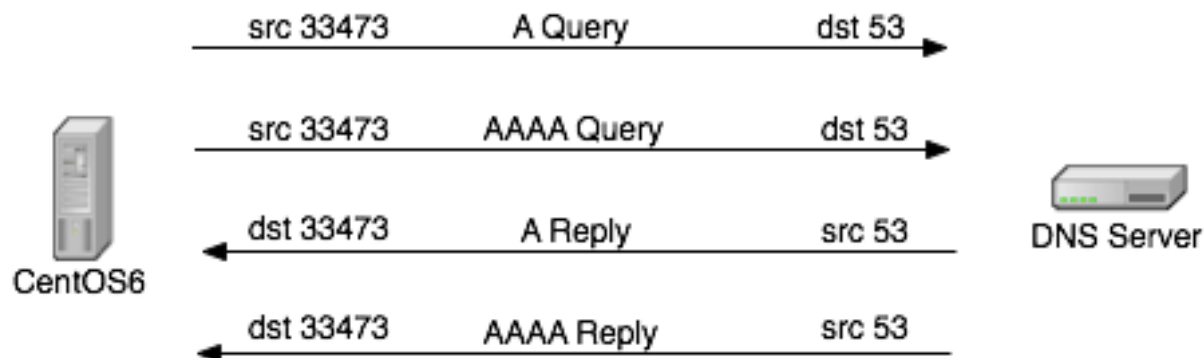
<http://www.nic.ad.jp/ja/materials/iw/2010/proceedings/s2/iw2010-s2-01.pdf>

が、Glibcのバージョンによって...

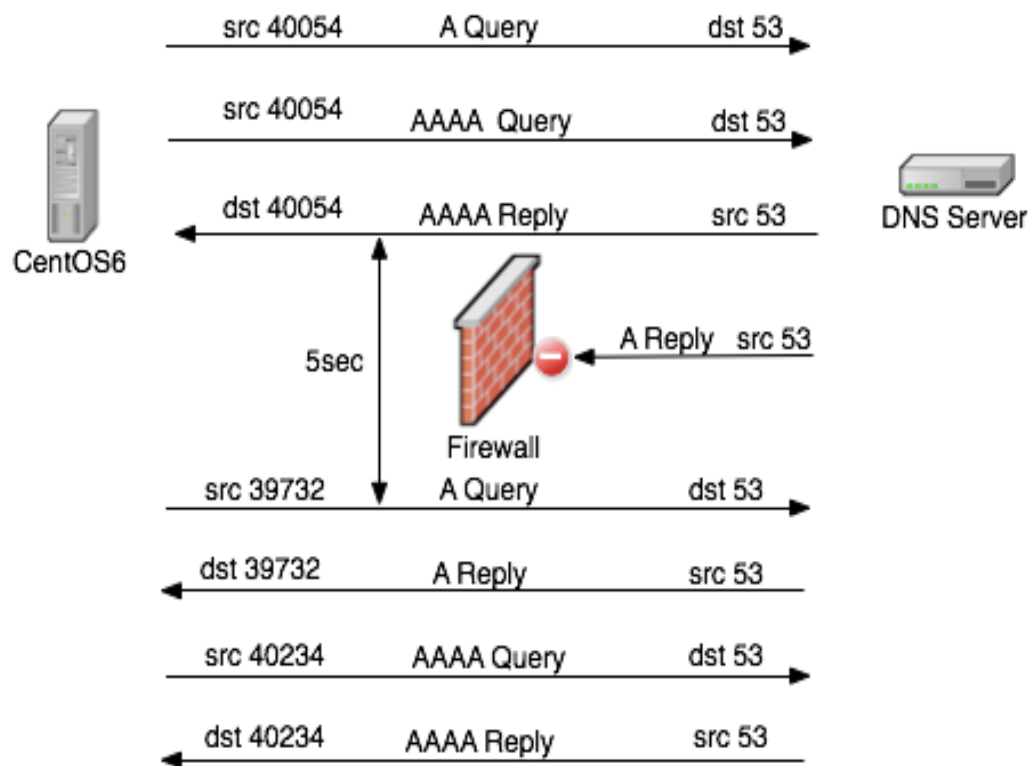
- RHEL5/CentOS5



- RHEL6/CentOS6 and so on.



基本的にはよい方向だが...



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

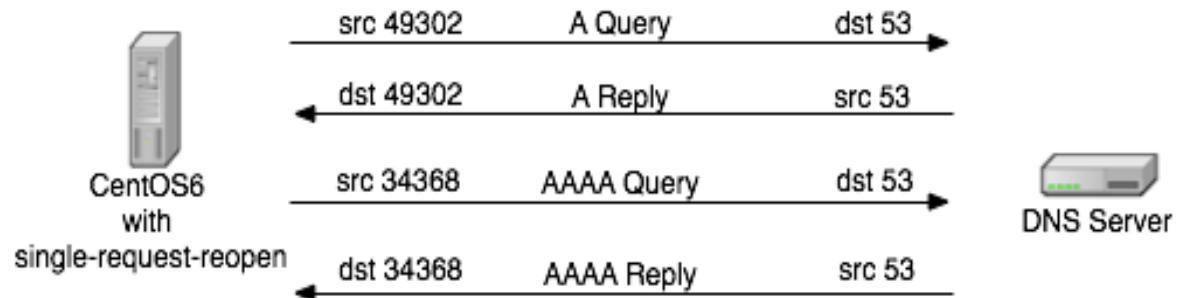
この問題の罪なところ

- 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.....

single-request-reopenを設定する

- /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

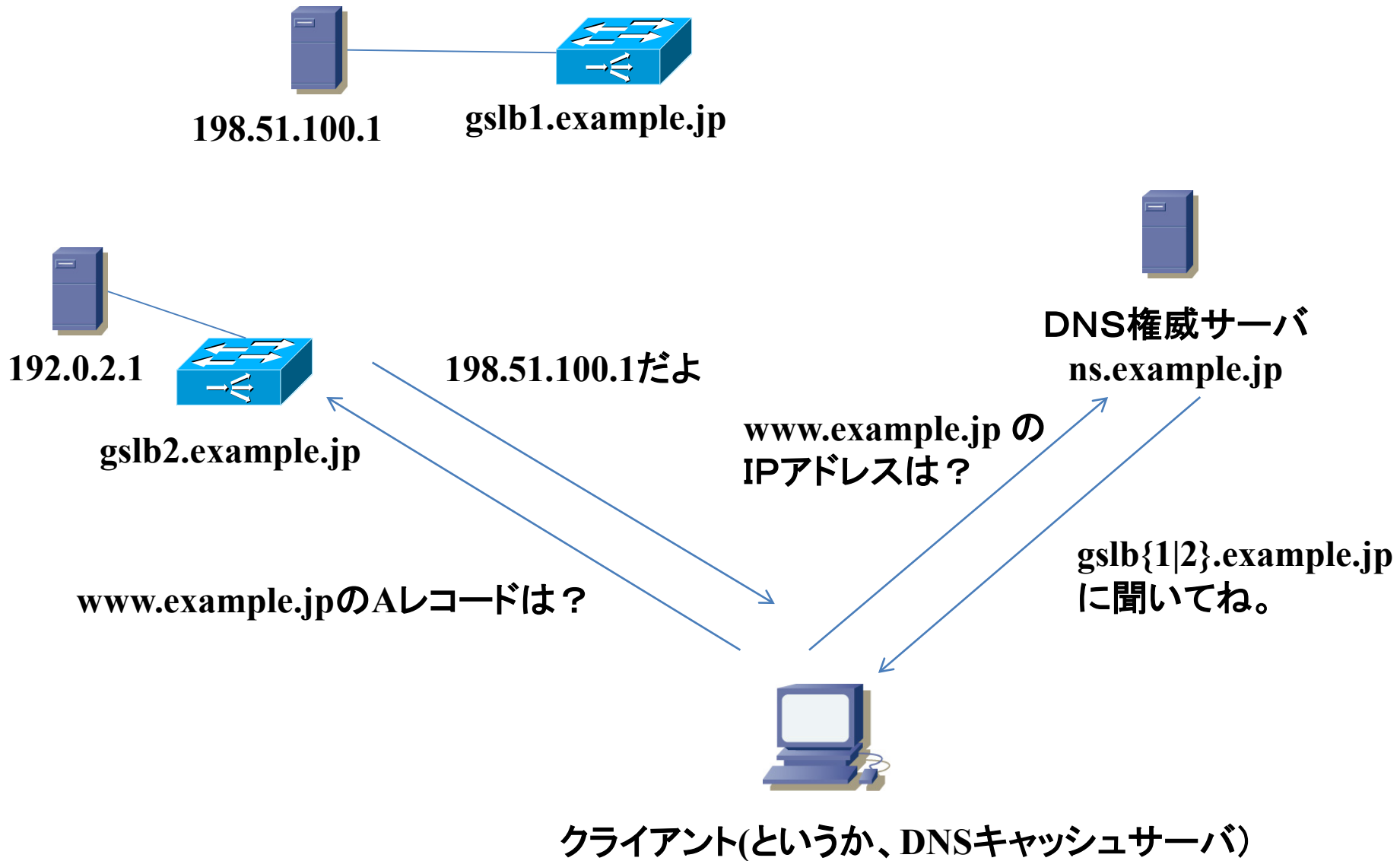
- よくわかんないけど、重い

- 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。

GSLBのざっくりとした仕組み



- 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)

- 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のものに比べて、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

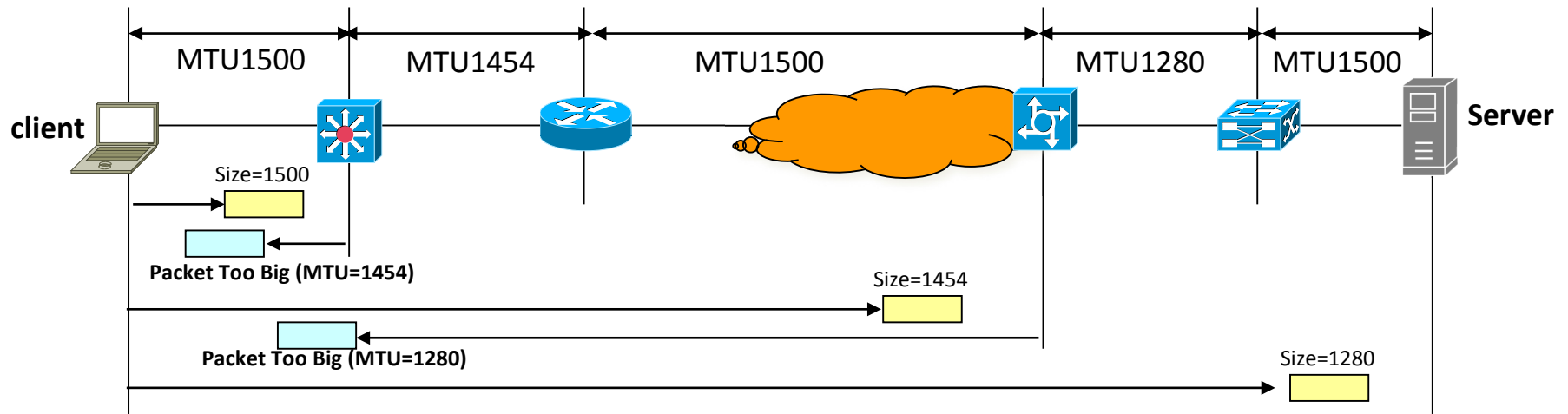
- バグに起因しないネットワークトラブルのほとんどが、Path MTU Discovery Blackhole問題

Path MTU Discoveryおさらい

Path MTU Discoveryとは(1)

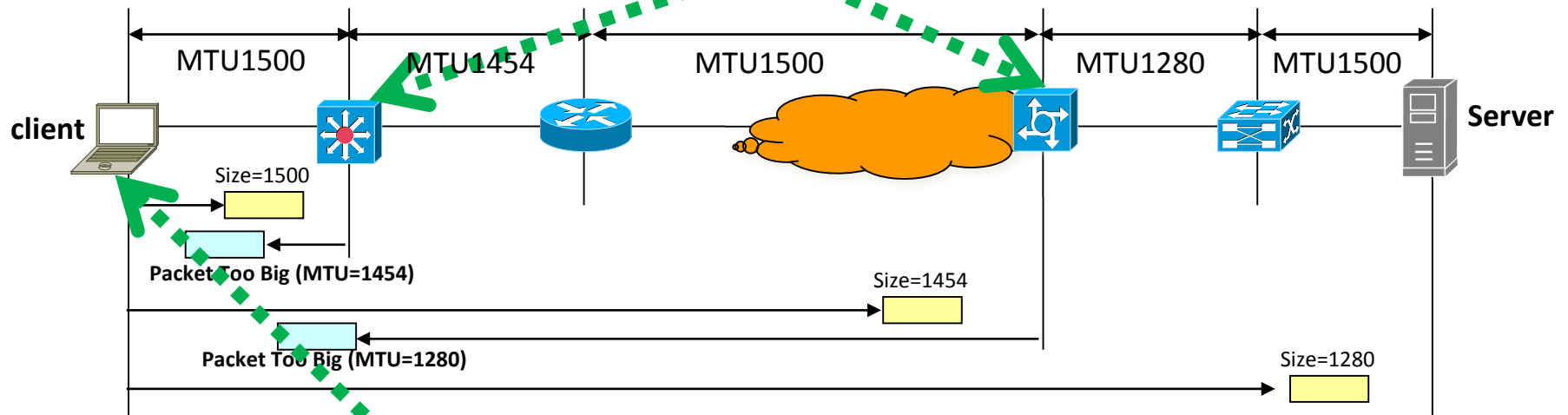
- IPv6 では中継ノードでフラグメントしない(始点ノードが実施)
- IPv4 ではルータ等の中継ノードがフラグメントを実施
ただし、DFビットが立っているパケットはフラグメントされず(この場合はIPv6の場合と同様)
- 送信パケットに対する ICMP Error Message を受信時、MTU を変更(最初のリンクのMTU が初期値)
 - IPv4の場合
 - ICMP Type 3/Code 4のパケットを受信時、始点ノードでフラグメントして再送
 - IPv6の場合
 - ICMPv6 Packet Too Big Message 受信時、始点ノードでフラグメントして再送
- L2 SWのMTUにひっかかった場合は破棄される

Path MTU Discoveryとは(2)



Path MTU Discoveryとは(3)

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

Too big 受け取るのはだれ？

- 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

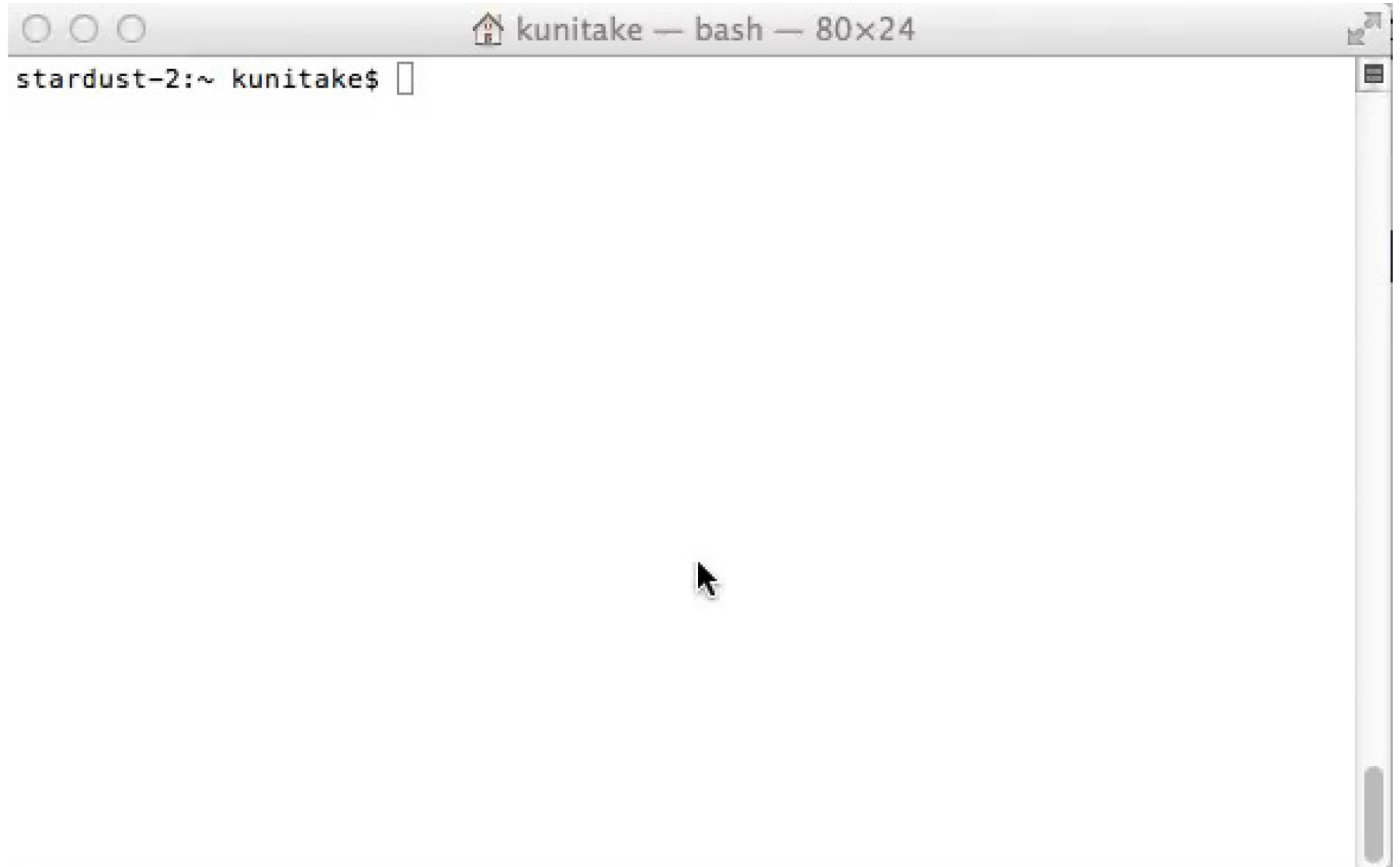
Too bigを必ず届ける、受け取る！

or/and

Too bigを発生させない！

がIPv6通信には必須

PMTUD Blackhole demo

A screenshot of a terminal window. The title bar shows three window control buttons (red, yellow, green) on the left, a home icon followed by the text "kunitake — bash — 80x24" in the center, and a maximize button 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. The terminal has a vertical scrollbar on the right side.

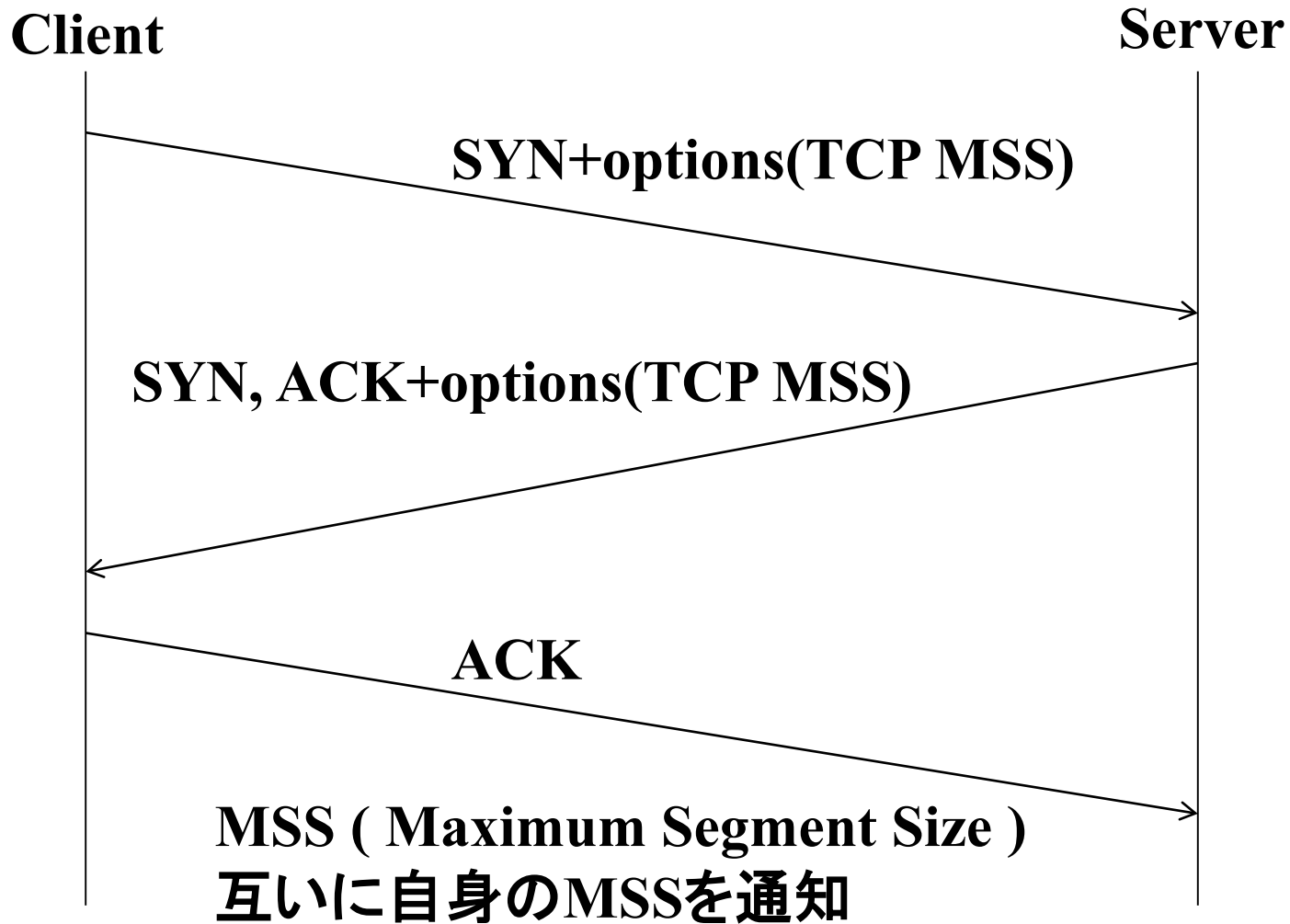
```
stardust-2:~ kunitake$
```

- 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**

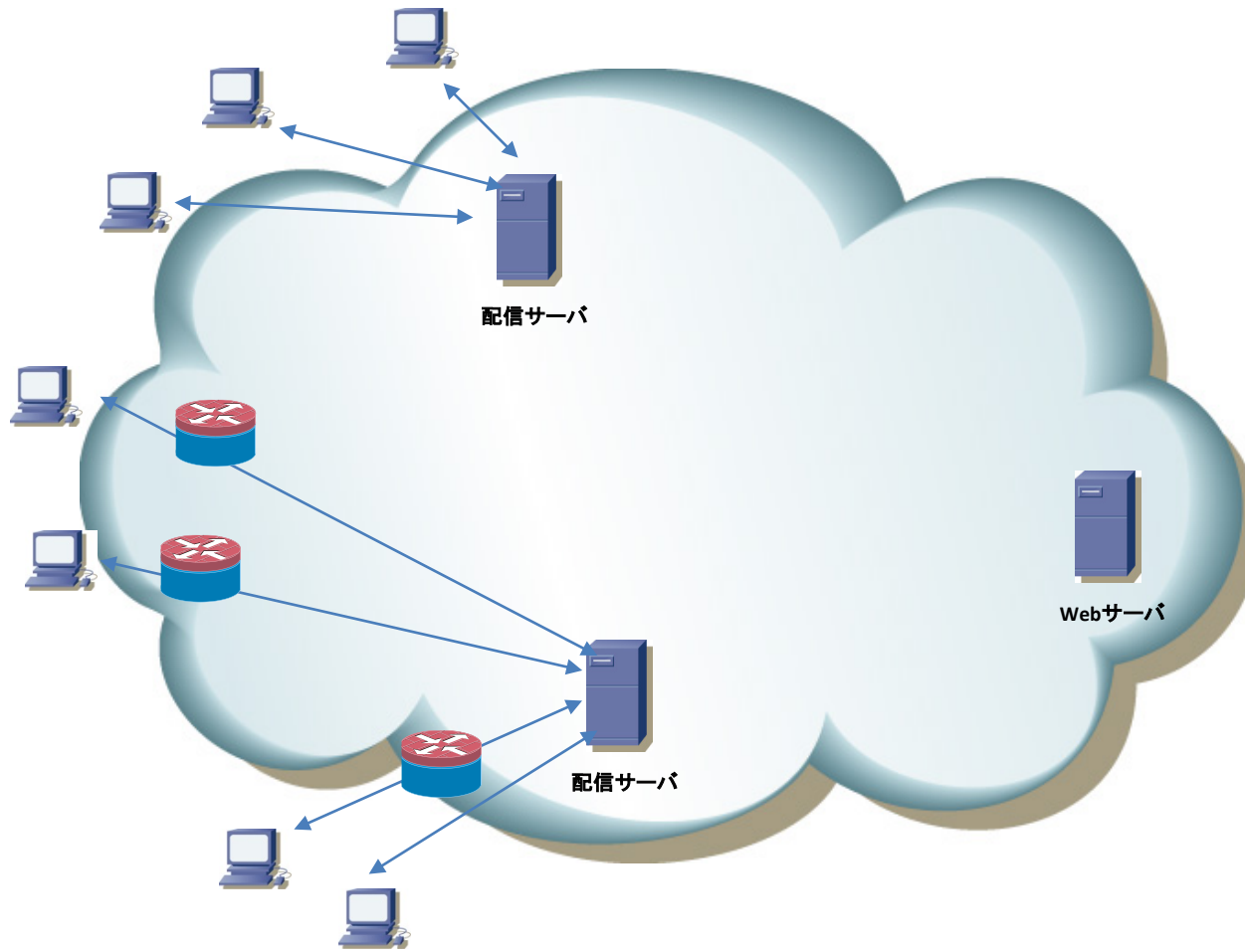
TCP 3way handshake復習



起きてしまったら、ここを疑え！

- 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

意図したサーバにToo bigは返る？





- 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か

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

補足：Firewallポリシーについて

- 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.

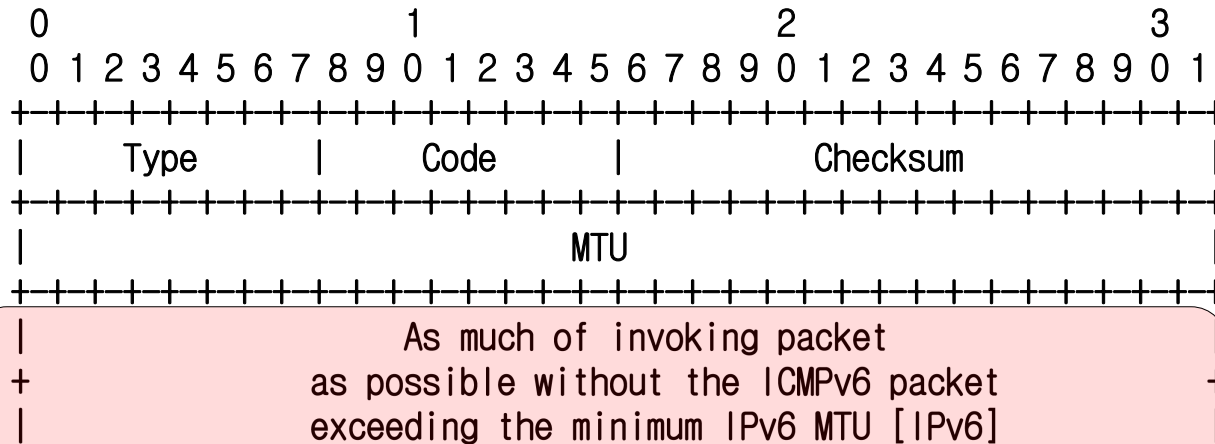
<http://kb.juniper.net/InfoCenter/index?page=content&id=KB26473&actp=RSS>

(<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。

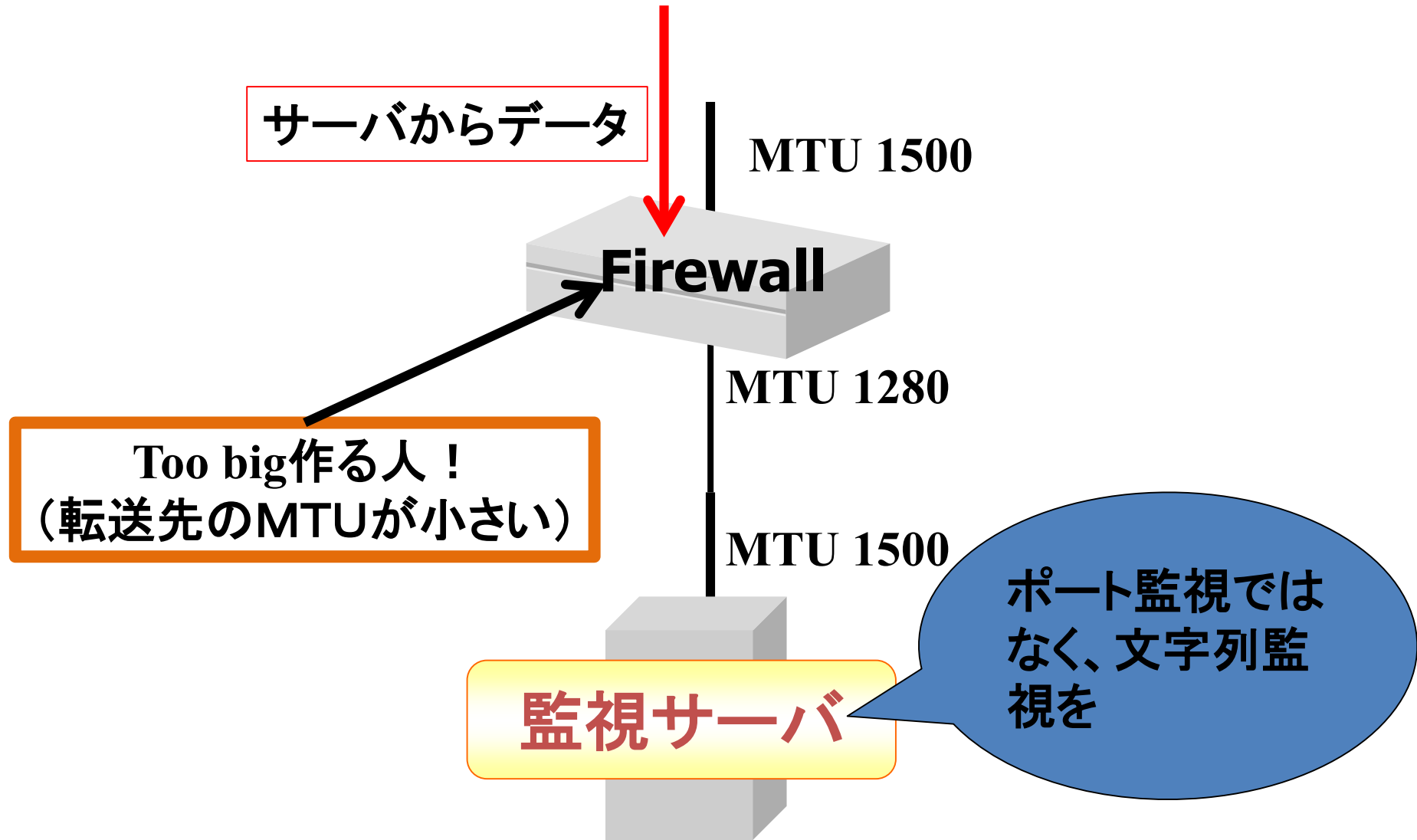
なぜひっかかるのか？

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。



Q&A?
