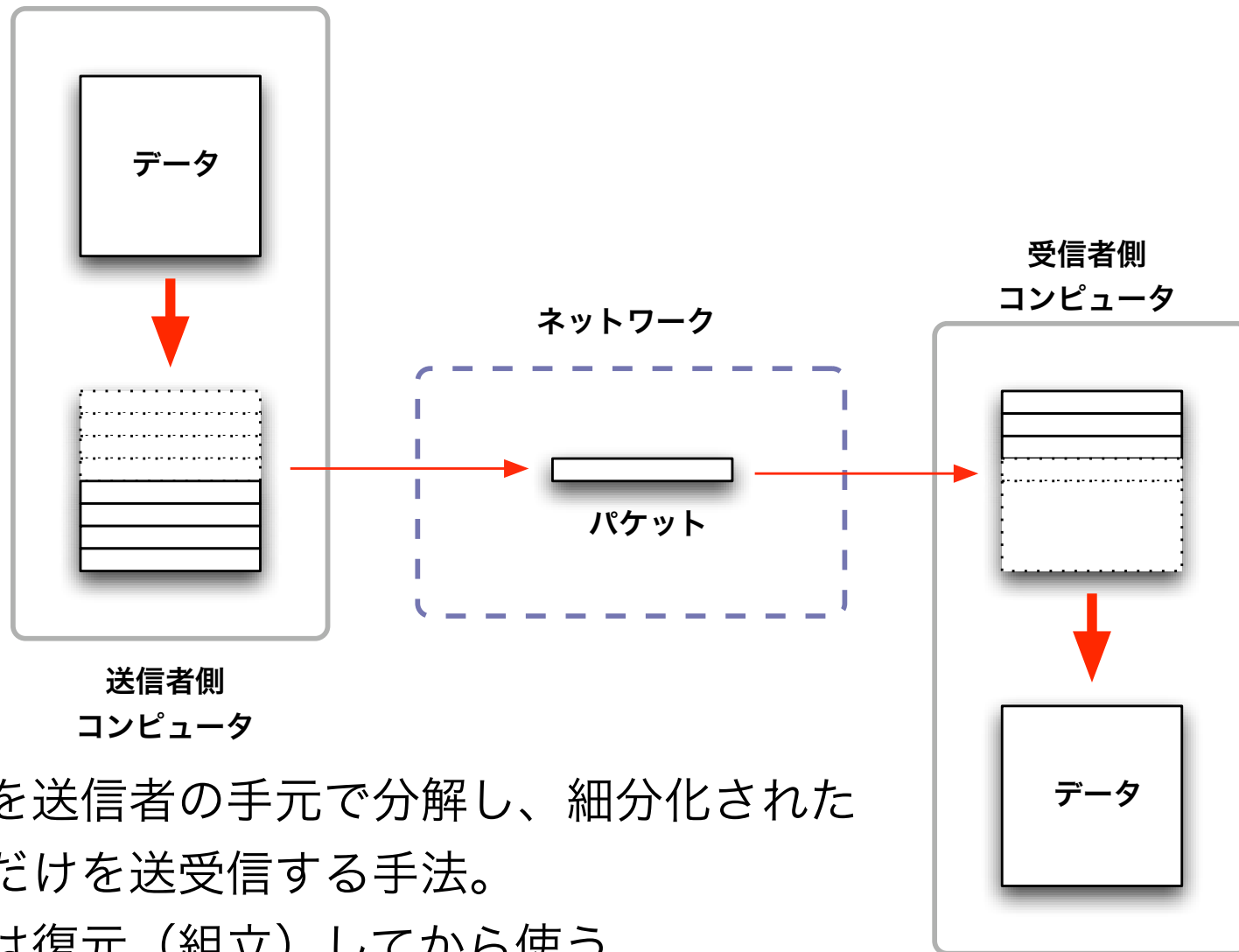


コンピュータシステムA - ハードウェアを中心に -

#14 IPv6

パケット通信



データを送信者の手元で分解し、細分化されたデータだけを送受信する手法。

受信側は復元（組立）してから使う。

パケットの構造 (の単純な例)

- ヘッダ
宛先アドレス、送り元アドレス、長さ、データ種類など
- ボディ
データそのもの
- エラー検出符号
SUM, CRC など、誤りが含まれていないことを調べるためのデータ



パケット全長は Ethernet (一般的な LAN) で 1.5KBytes 程度

インターネットにおけるアドレッシング

- IP アドレス

接続されている全てのコンピュータに個別に割り当てられた番号

例：133.101.32.84 = 4 Bytes = 32bits

- グローバルアドレス：世界で唯一になるように階層管理されて割り当ててるアドレス

- 互いにIPアドレスを指定して通信する

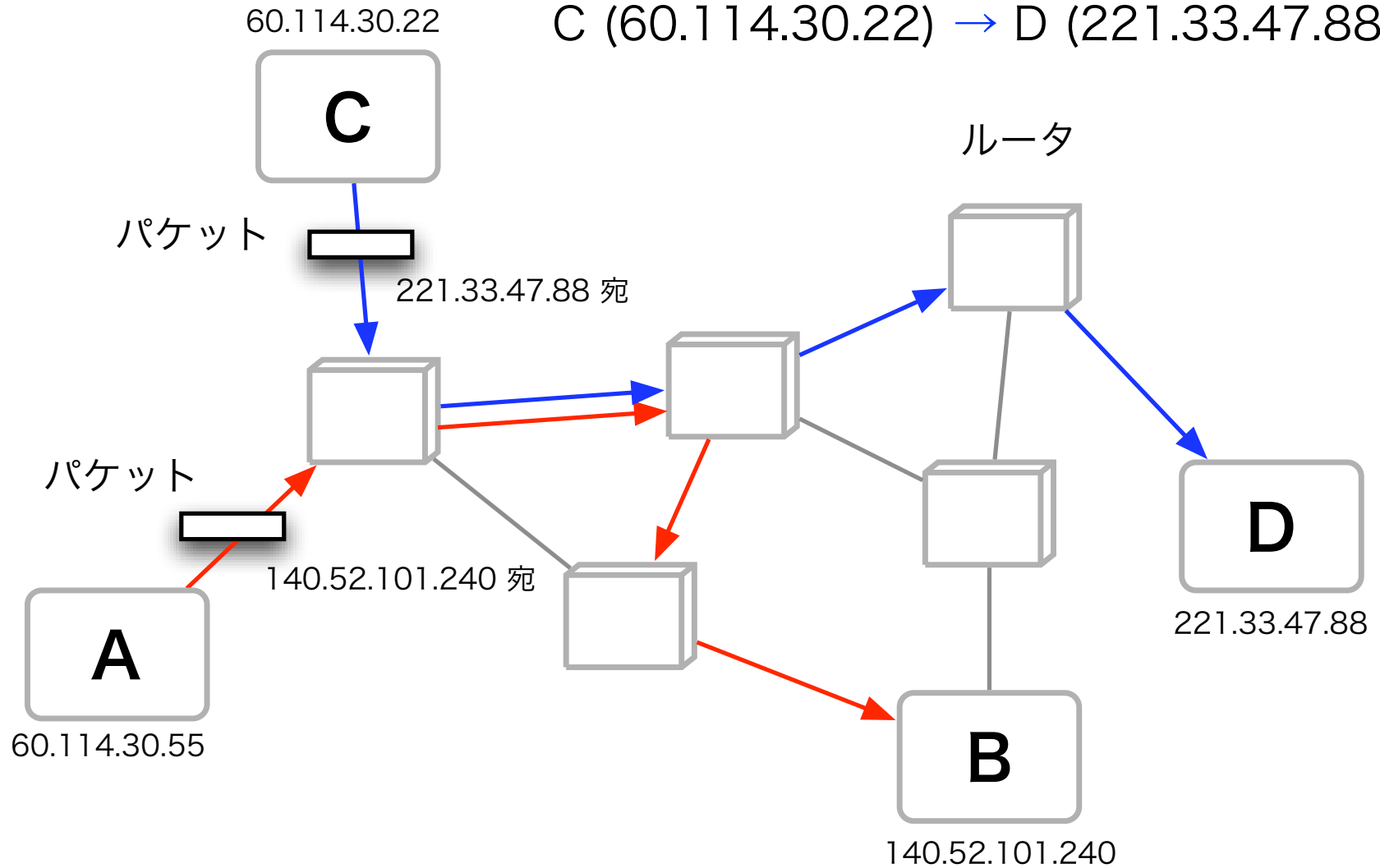
www.yahoo.com も机の PCも同じく持っている

- 対等な接続

ルーティング (再掲)

A (60.114.30.55) → B (140.52.101.240)

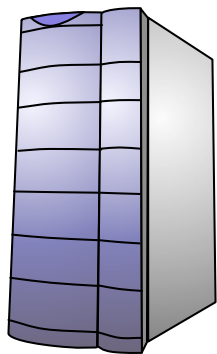
C (60.114.30.22) → D (221.33.47.88)



ドメイン名によるアクセス

- IPアドレスによるアクセス
不便+ネットワーク構成変更
- 名前によるアクセス（`www.kyoto-su.ac.jp` など）
ドメイン名：階層的に管理されたインターネット上の名前空間
- DNS (Domain Name System)
ドメイン名からIPアドレスを検索

61.194.20.x



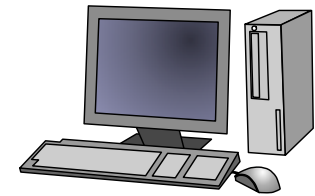
Server

`http://www.kyoto-su.ac.jp/` のデータが欲しい

GET

リクエスト

133.101.x.x

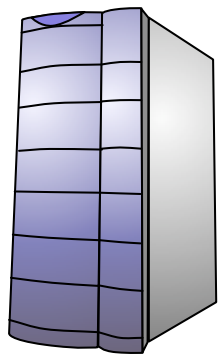


Client

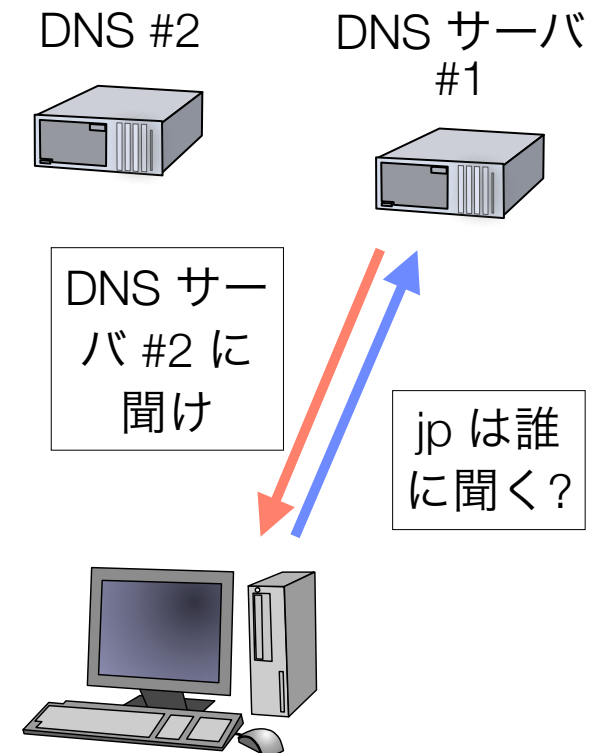
データ

大学のWebサーバにアクセスする

1. `http://www.kyoto-su.ac.jp/` の IP アドレスは何か？
2. DNS サーバに `.jp` は誰に聞けば良いかを聞く
(最初の訊ね先は PC が知っている)
3. DNS サーバは訊ね先の DNS サーバの IP アドレスを答える

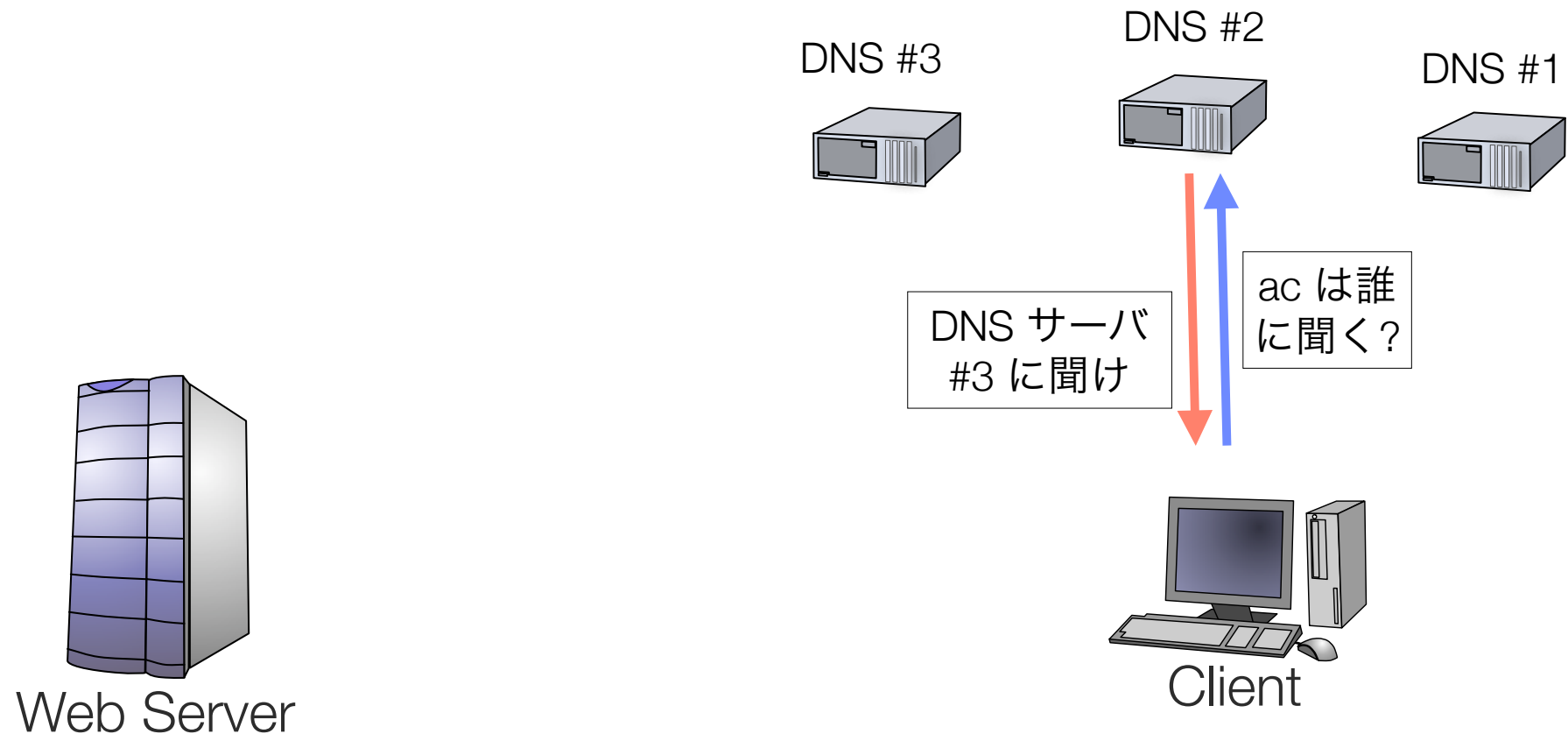


Web Server



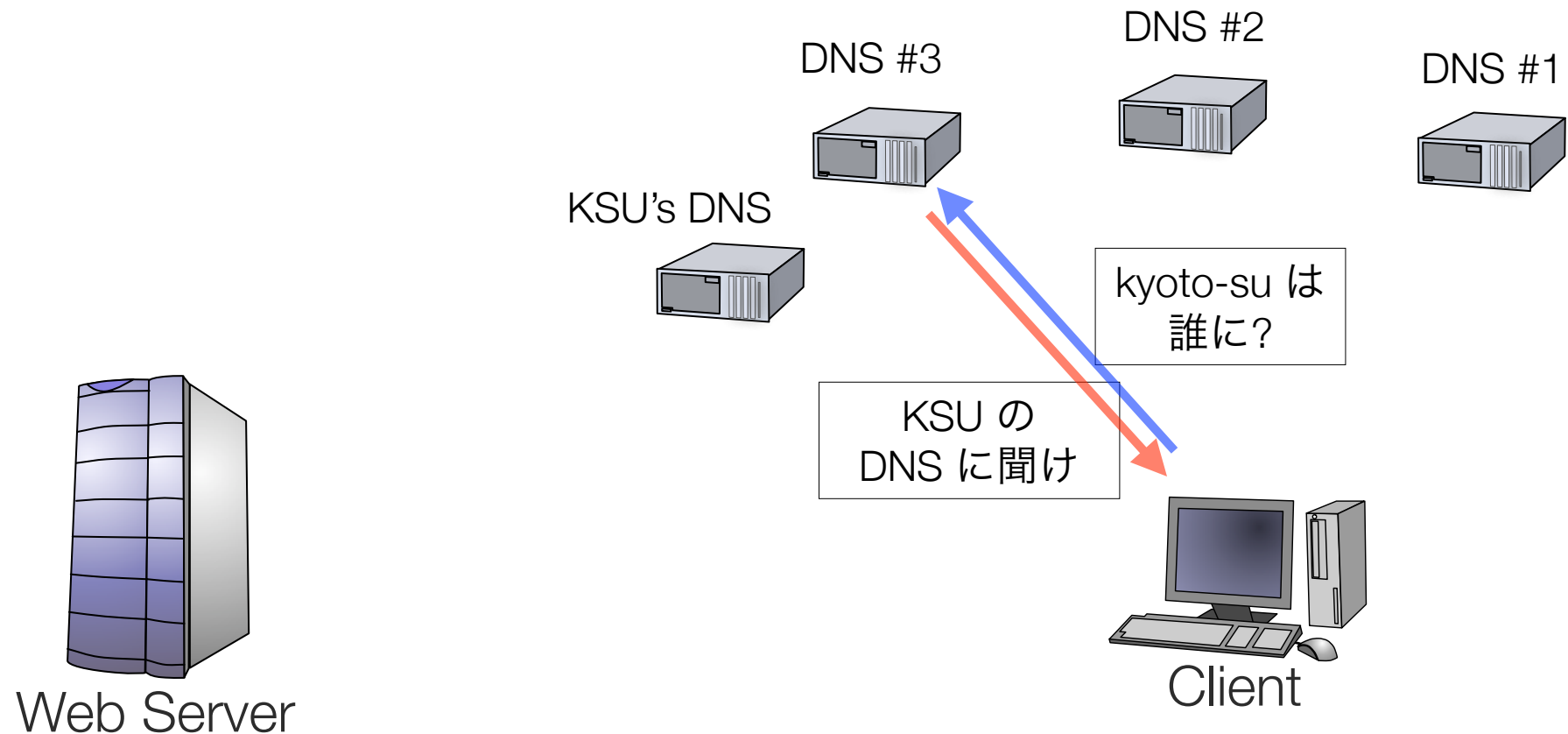
大学のWebサーバにアクセスする

4. DNS サーバ#2 に ac は誰に聞けば良いかを聞く
5. #2 は #3 を答えた



大学のWebサーバにアクセスする

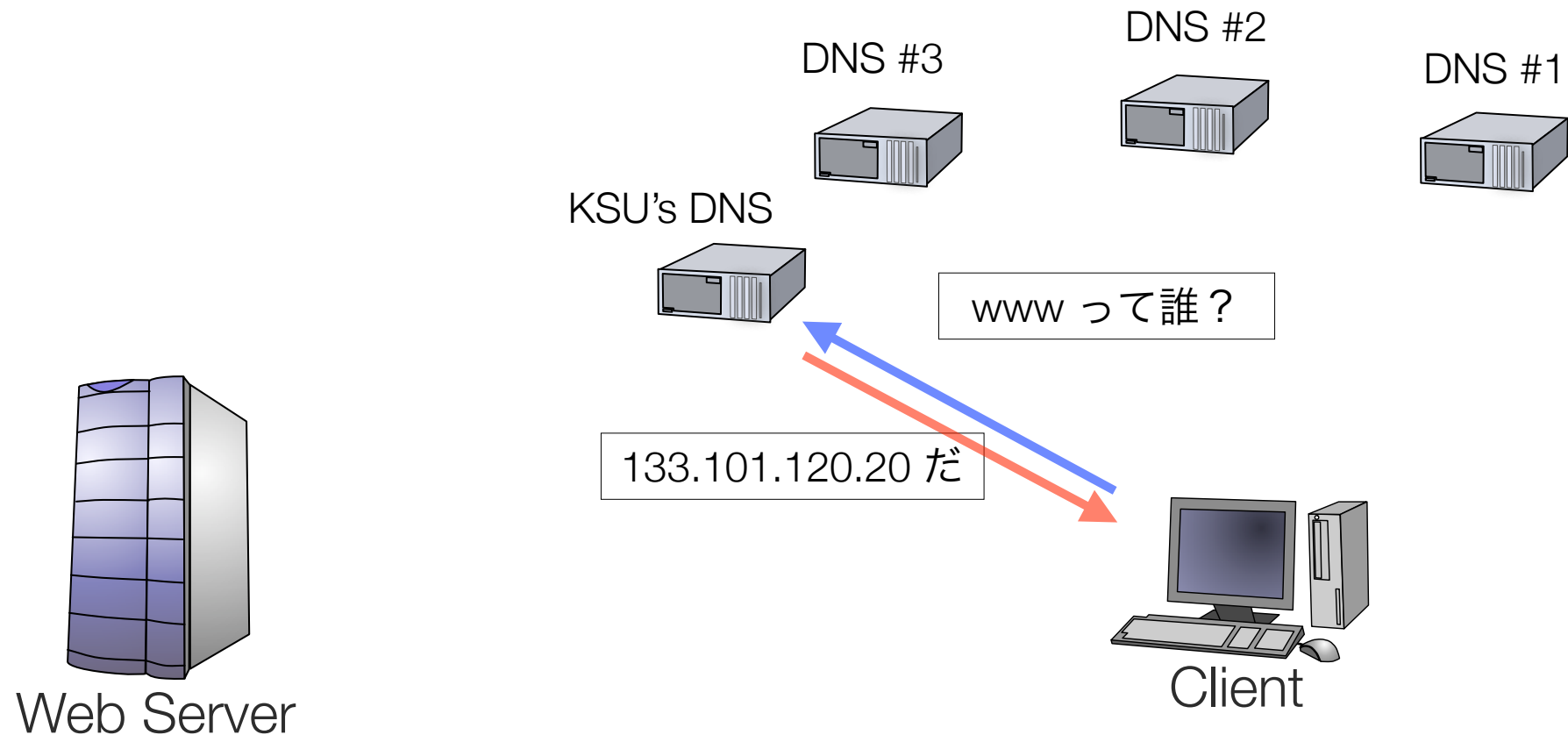
6. DNS サーバ#3 に kyoto-su は誰に聞けば良いかを聞く
7. #3 は KSU の DNS を答えた



大学のWebサーバにアクセスする

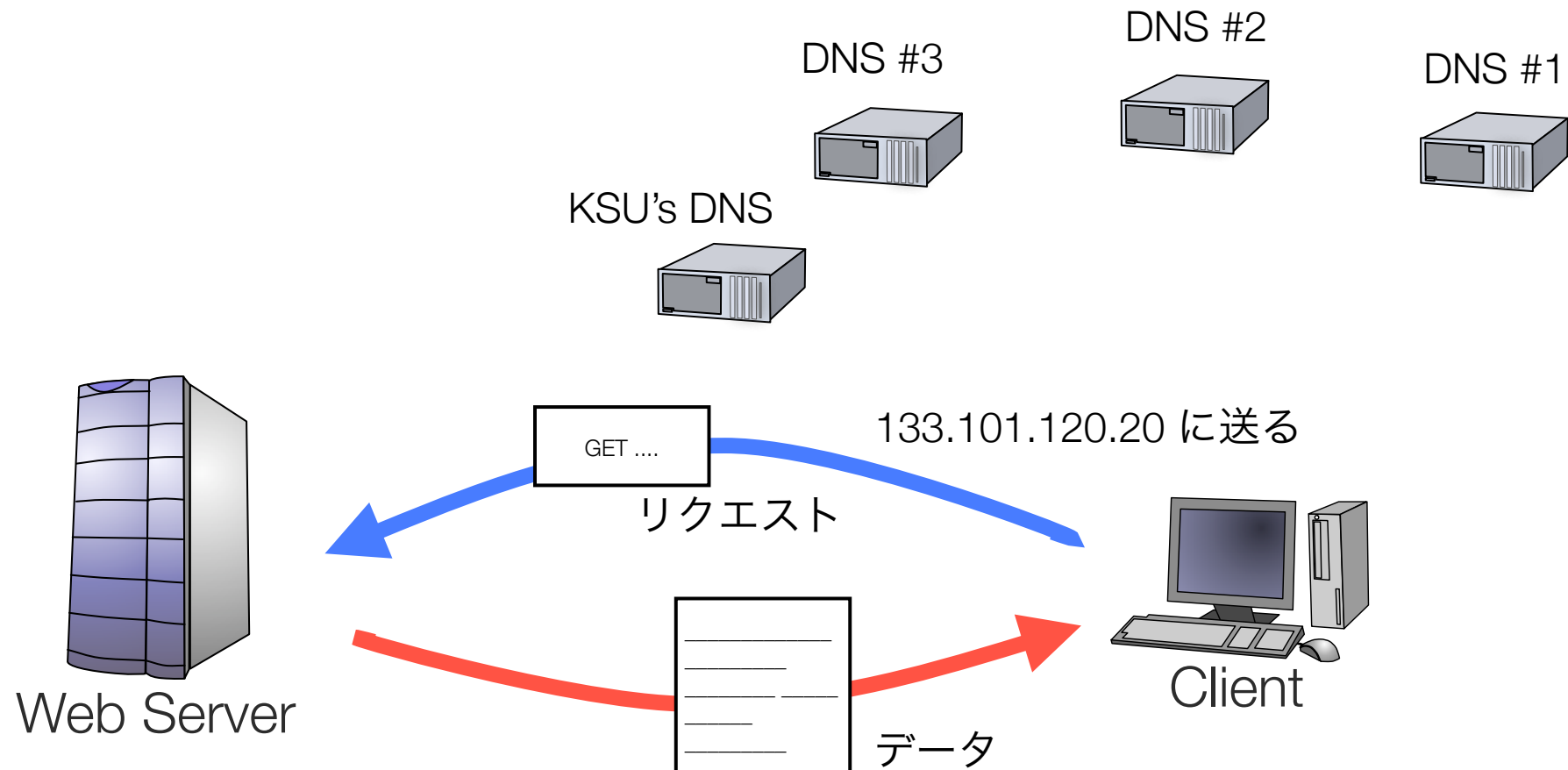
8. KSU の DNS に www の IP アドレスを聞いた

9. アドレスは 133.101.120.20 だと答えた



大学のWebサーバにアクセスする

10. (ようやく) アドレス 133.101.120.20 に Web ページ閲覧の依頼を送る
11. 依頼したページのデータが返ってきた



インターネットにおけるアドレッシング（再掲）

- IP アドレス

接続されている全てのコンピュータに個別に割り当てられた番号

例：133.101.32.84 = 4 Bytes = 32bits

- グローバルアドレス：世界で唯一になるように階層管理されて割り当てるアドレス

- 互いにIPアドレスを指定して通信する

www.yahoo.com も机の PCも同じく持っている

- 対等な接続

インターネットにおけるアドレッシング（再掲）

- IP アドレス

接続されている全てのコンピュータに個別に割り当てられた番号

例：133.101.32.84 = 4 Bytes = 32bits

- グローバルアドレス：世界で唯一になるように階層管理されて割り当てるアドレス

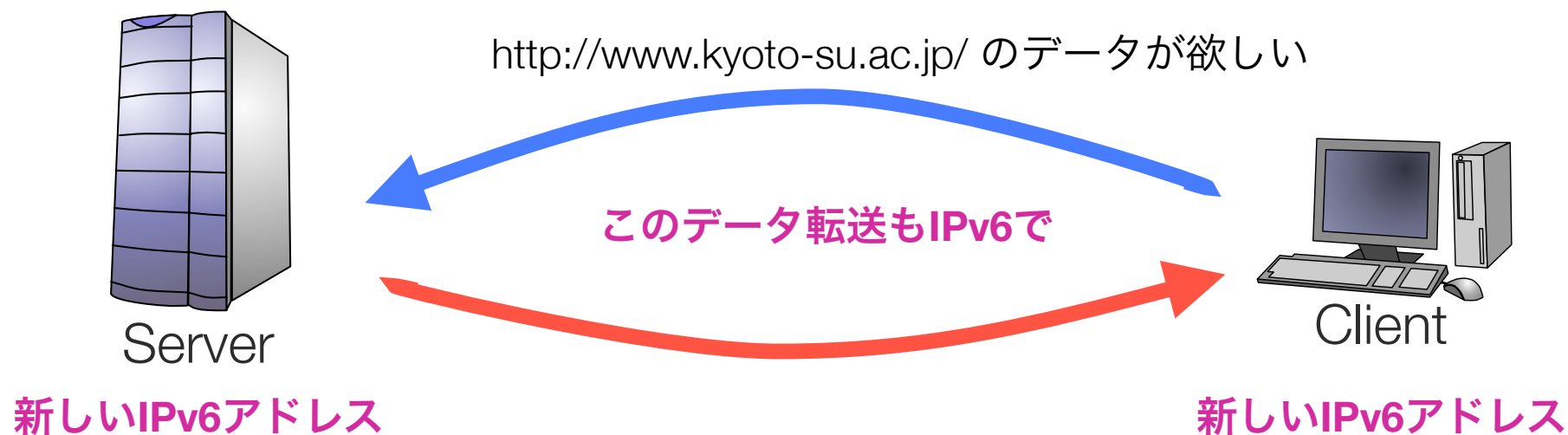
- 互いにIPアドレスを指定して通信する
 - 接続できるコンピュータの数には限りがある
 - その上限は $2^{32} = 40$ 億程度
 - 対等な接続
- www.yahoo.com も机のPCも同じく持っている

IPv6

- 新しいプロトコル

従来の IP version 4 とは互換性がない


- IPv4 アドレスの枯渇問題を背景に開発



IP アドレス


IPv4 アドレス

133.101.38.42


8bit x4 = 32bit

IPv6 アドレス

2404:6800:400a:0802:1234:2124:2142:1010


16bit x8 = 128bit



パケットヘッダのフォーマット



IPv4 ヘッダ

bit
0

version	Length	type	Length	
Indication			flg	offset
TTL		Proto.	Check sum	
Source Address (32bit)				
Destination Address (32bit)				
Option...				

bit
31

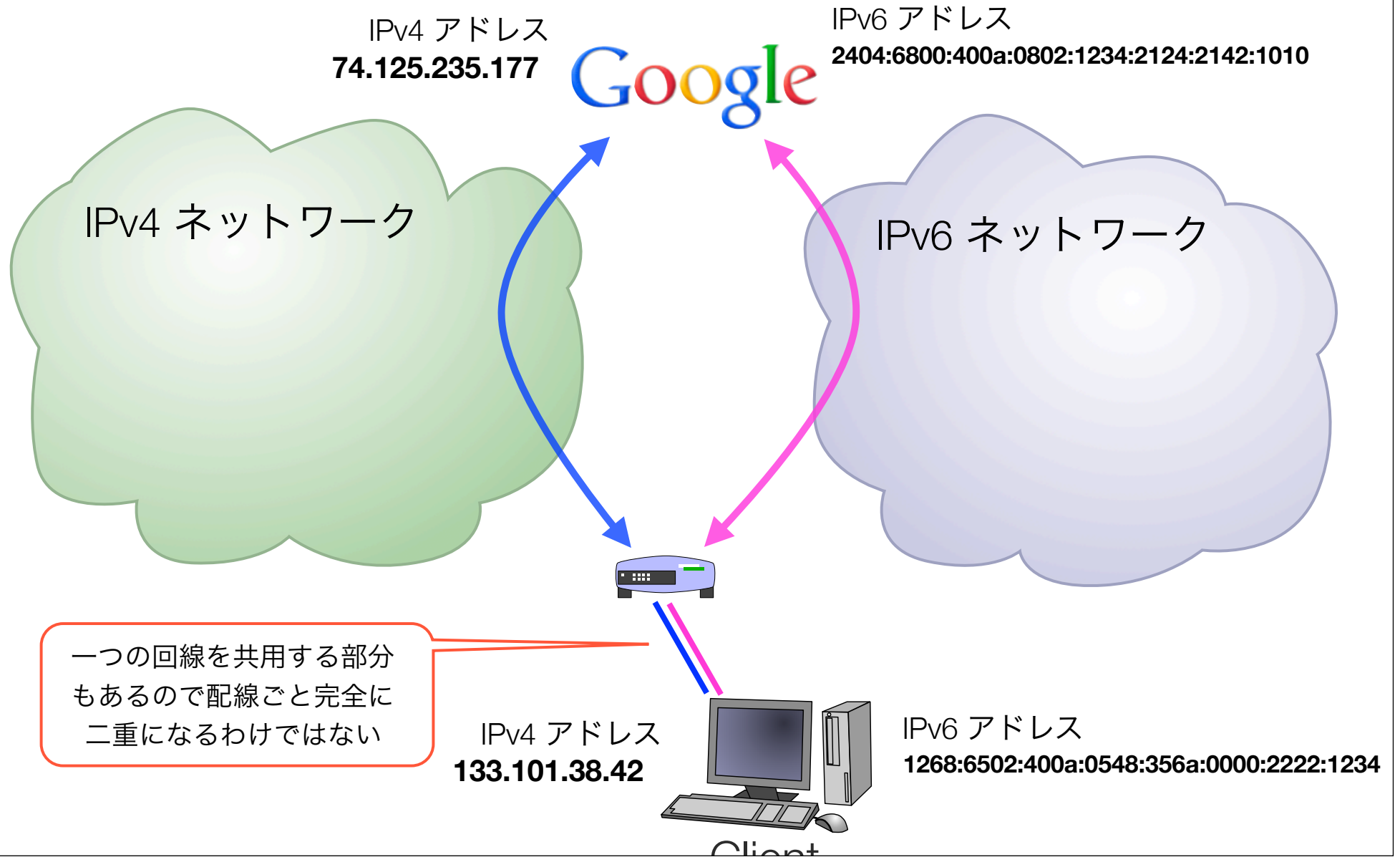
IPv6 ヘッダ

bit
0

bit
31

version	traffic class	Flow Label		
Length		next header	hop limit	
Source Address (128bit)				
Destination Address (128bit)				

二重のインターネット



ではもう IPv6 で良いではないか

- 多くのサーバは IPv6 ではサービスを提供していない
- つまりクライアントの IPv4 アドレスは外せない
 - 誤：v6 利用者が増えれば v4 枯渇問題は解決する
 - 正：v4 でアクセスしなくて済むようにしなければ
- しかし v6 に対応したところで儲からないではないか
 - アクセスが増えるわけではない
 - v6化しなければアクセスが減るわけでもない
- ユーザが特に v6 を好む理由はない (技術的好奇心だけ?)

では v6 化なんてしなくて良いではないか

- 長い時間を掛けてでもやらなければならない

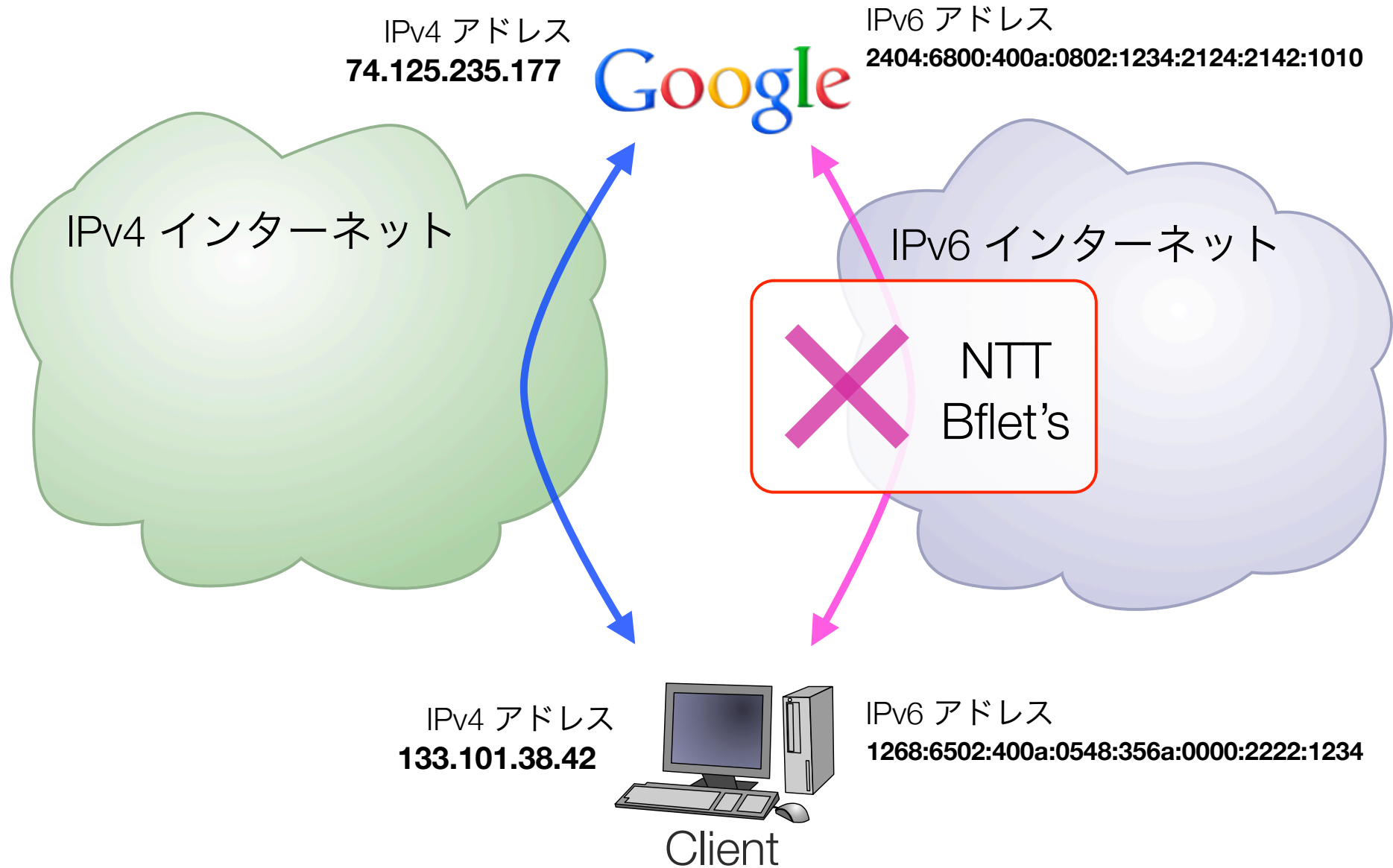
IPv4 アドレス価格の（意味のない）高騰
（稀少財は高く売れますから）

技術的に最適でないネットワーク構造への偏向
（＝無意味なコスト増大）

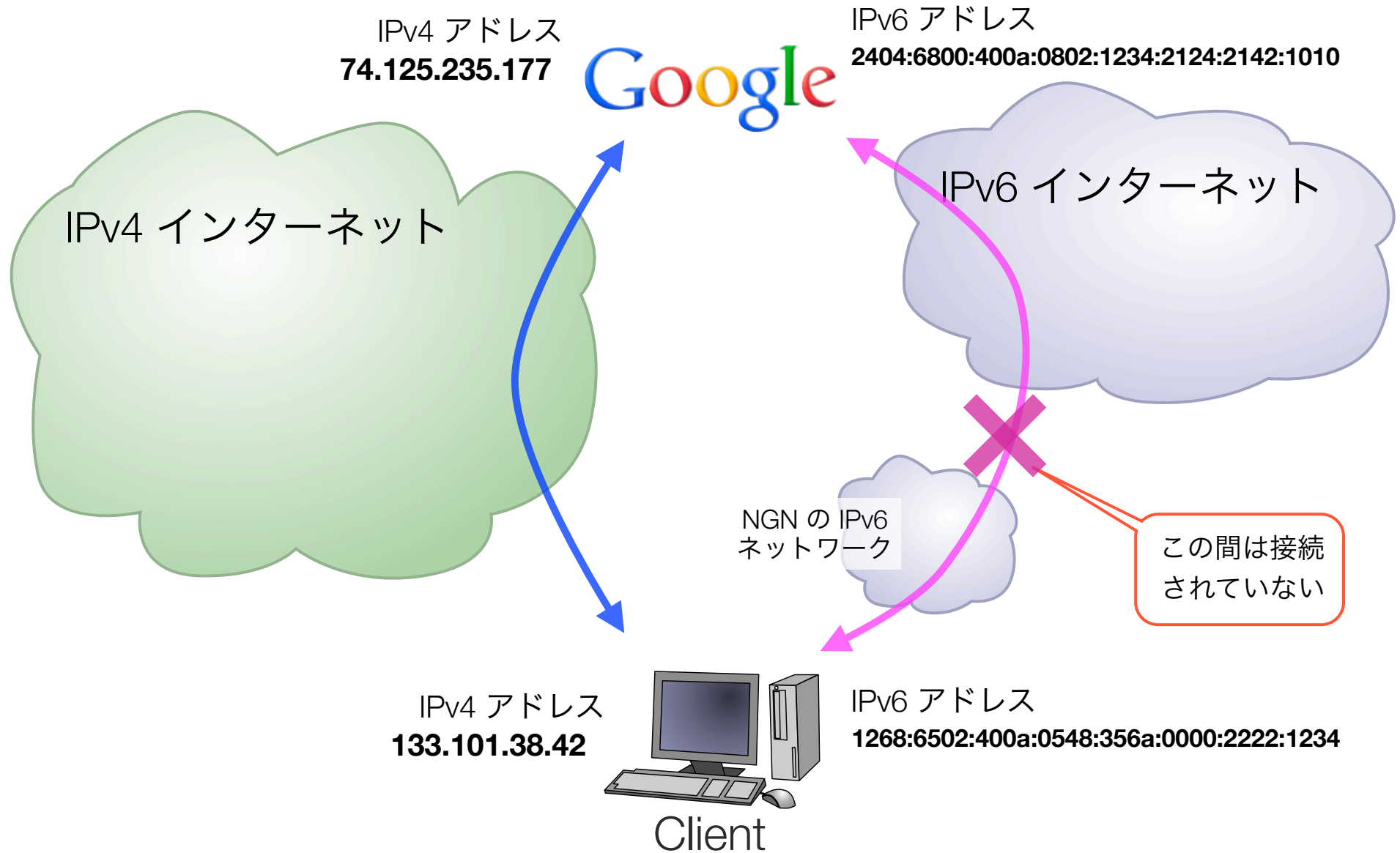
World IPv6 Launch, 2012.6.6



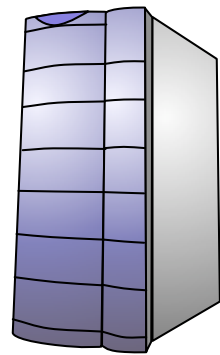
NTT NGN 網での "フォールバック" 問題



NTT NGN 網での "フォールバック" 問題



DNS の振る舞い



Server

www.google.com

などの v4, v6 両方で
通信できるサービス

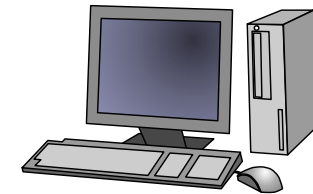
v4=173.194.38.114

v6= 2404:6800:4004:800::1013

IPv6



IPv4

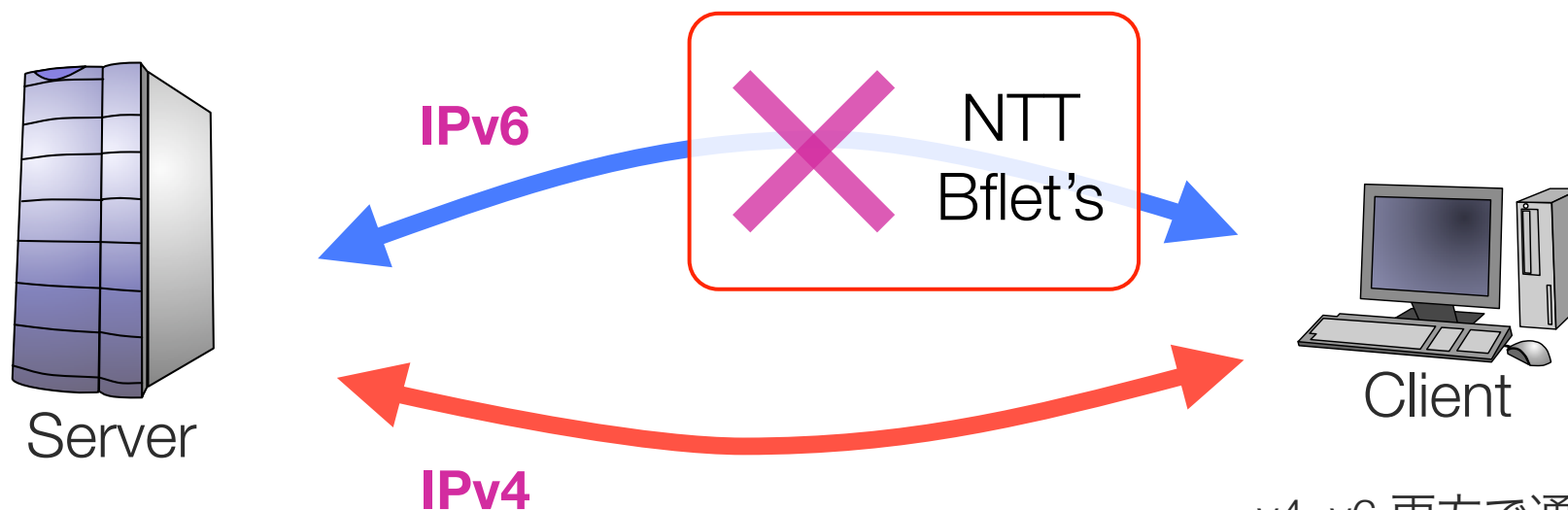


Client

v4, v6 両方で通信で
きるOS (ほぼ全部)

1. www.google.com はv6 でもサービスされていた
2. DNS で名前からアドレスを引く
3. v6 のアドレスがあるので、まず v6 でアクセス
4. しかし失敗する
5. v4 で再トライ

Workaround



v4, v6 両方で通信できるOS (ほぼ全部)

Google

じゃあ NGN ユーザには v6 アドレスを答えないようにするよ

NTT

じゃあ DNS の返事から v6 アドレスを抜いてユーザに知らせようか

1. www.google.com はv6 でもサービスされていた
2. DNS で名前からアドレスを引く
3. v6 のアドレスがあるので、まず v6 でアクセス
4. しかし失敗する
5. v4 で再トライ

まとめ

- パケット通信のしくみ
- インターネットのしくみ
- DNS のしくみ
- アドレスという概念
 - v4 アドレスの枯渇
 - v6 への移行
- そのほころびと対策

- 技術とどう向き合うか

常にまっすぐ飛ぶわけではない

それでも歪め過ぎず、未来に向けて進まねばならない

- そのための視点を育てる

技術の理解と周囲の状況とのバランスが取れた舵切りを