

明日からのインターネット利用指南 ～IPv6とは？ 対応の必要はあるの？
アプリケーションのIPv6対応

一般財団法人 インターネット協会
IPv6デプロイメント委員会
北口 善明

August 19, 2017

- IPv6に関する最近の環境
 - 端末OSやクラウドサービスにおけるIPv6対応状況
- アプリケーションのIPv6対応
 - IPv6対応アプリケーション作成時の基本方針とポイント

IPv6に関する最近の環境

● 端末OS

- Windows Vista以降, Mac OS X, macOS, Linux, FreeBSD, iOS, Android ...
- デフォルトでIPv6が利用可能

● インターネット接続回線

- フレッツ光ネクスト, auひかり, NURO光 ...
- 既存ユーザへの自動導入も進行中

● モバイルネットワーク

- docomo, au, SoftBank ...
- 2017年度モデルからデフォルトでIPv6利用可能

● クラウドサービス

- さくらインターネット, AWS, MS Azure, IJ ...
- 多くのクラウド事業者でIPv6サービスを利用可能

- 2016年6月1日からApp StoreアプリはIPv6-only対応が必須に

Supporting IPv6-only Networks

Starting June 1, 2016, all apps submitted to the App Store must support IPv6-only networking. A majority of apps will not require any changes as IPv6 is already supported by NSURLSession and CFNetwork APIs. However, if your app utilizes IPv4-specific APIs or hard-coded IP addresses, you will need to make changes. Be sure to test for IPv6 compatibility before submitting your app to the App Store for review.

For more information on supporting IPv6 networks, review [Supporting IPv6 DNS64/NAT64 Networks](#).

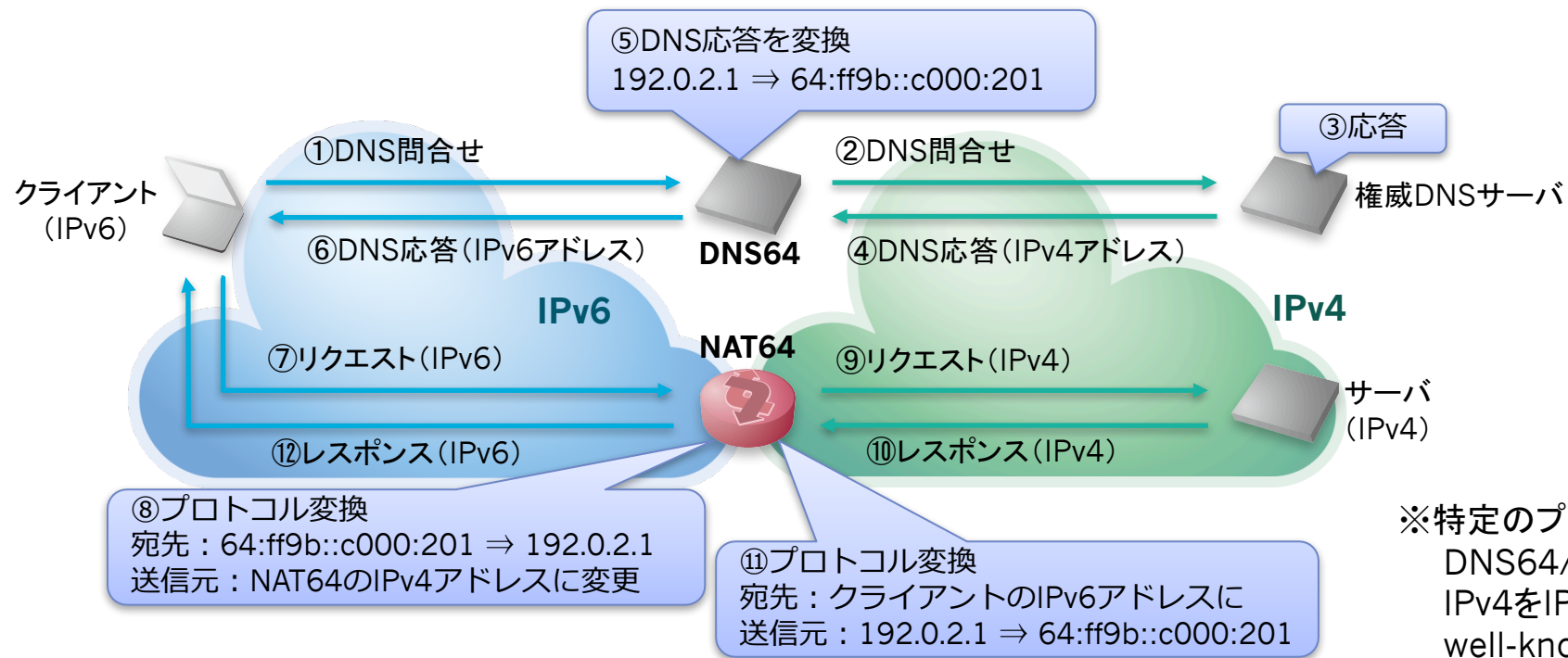
<https://developer.apple.com/support/ipv6/> より

- IPv4固有のAPIやIPアドレスのハードコーディングがあるとNG
- DNS64/NAT64環境下での動作確認が必要
- なぜIPv6なのか
 - IPv4枯渇, IPv6はIPv4より効率的, 4Gの普及, マルチメディアサービスとの互換性, コスト を理由に挙げている

“Supporting IPv6 DNS64/NAT64 Networks” より

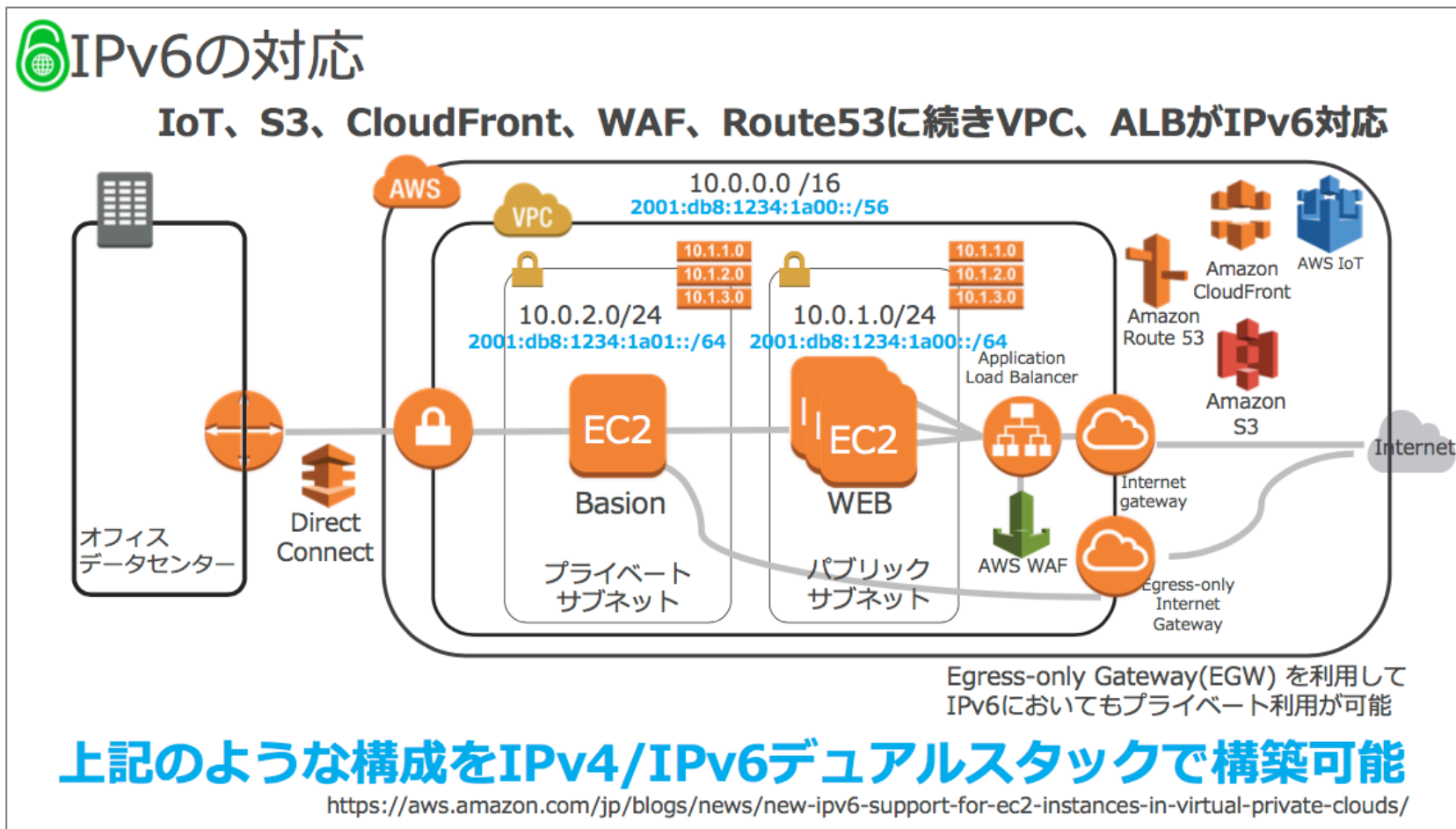
DNS64/NAT64

- IPv6をベースとしてIPv4へのアクセスをアドレス変換で提供
 - DNS64にてDNS応答におけるIPv4アドレスをIPv6アドレスに変換
 - NAT64にて特定のIPv6プレフィックス宛の通信をIPv4にアドレス変換
- DNS64/NAT64環境におけるIPv4サーバとの通信の流れ



クラウド事業者のIPv6対応 (Amazon社)

- 2016年8月のS3対応に続き各種サービスへの展開を実施
- 中国以外のDCにてIPv6対応が可能



“IPv6対応クラウドサービスワークショップ (https://www.iajapan.org/ipv6/2017/0419_ws.html)” 発表資料より

- 2016年9月にAzureのIPv6対応を発表
 - ネイティブでデュアルスタック利用が可能

IPv6 for Azure VMs available in most regions

2016年9月26日

At Ignite 2016, we announced general availability of IPv6 for Azure virtual machines.

Most public Azure regions can now host dual-stack (IPv4+IPv6) VMs:

- Brazil South
- Canada Central
- Canada East
- Central India
- Central US
- East Asia
- East US
- East US 2
- Japan East
- Japan West
- North Europe
- North Central US
- South India
- South Central US
- Southeast Asia

関連サービス



Load Balancer



Virtual Machines



Virtual Network

<https://azure.microsoft.com/ja-jp/updates/ipv6-for-azure-vms/> より

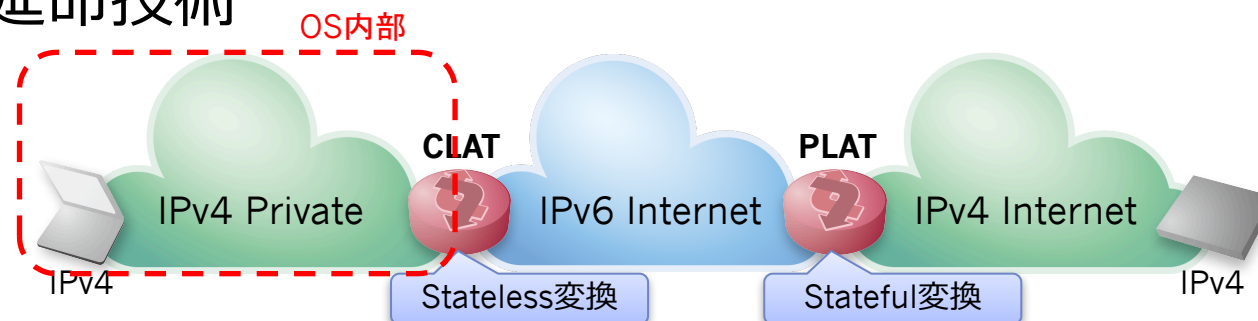
- Windows 10に464XLAT (CLAT) 実装 (Creators Updateにて)
 - IPv6非対応アプリケーションのIPv6 only環境で動作を可能に

Improved 464XLAT support

464XLAT was originally designed for cellular only scenarios since mobile operators are some of the first ISPs with IPv6 only networks. However, some apps are not IP-agnostic and still require IPv4 support. Since a major use case for mobile is tethering, 464XLAT should provide IPv4 connectivity to tethered clients as well as to apps running on the mobile device itself. Creators Update adds support for 464XLAT on desktops and tablets too. We also enabled support for TCP Large Send Offload (LSO) over 464XLAT improving throughput and reducing CPU usage.

<https://blogs.technet.microsoft.com/networking/2017/07/13/core-network-stack-features-in-the-creators-update-for-windows-10/> より

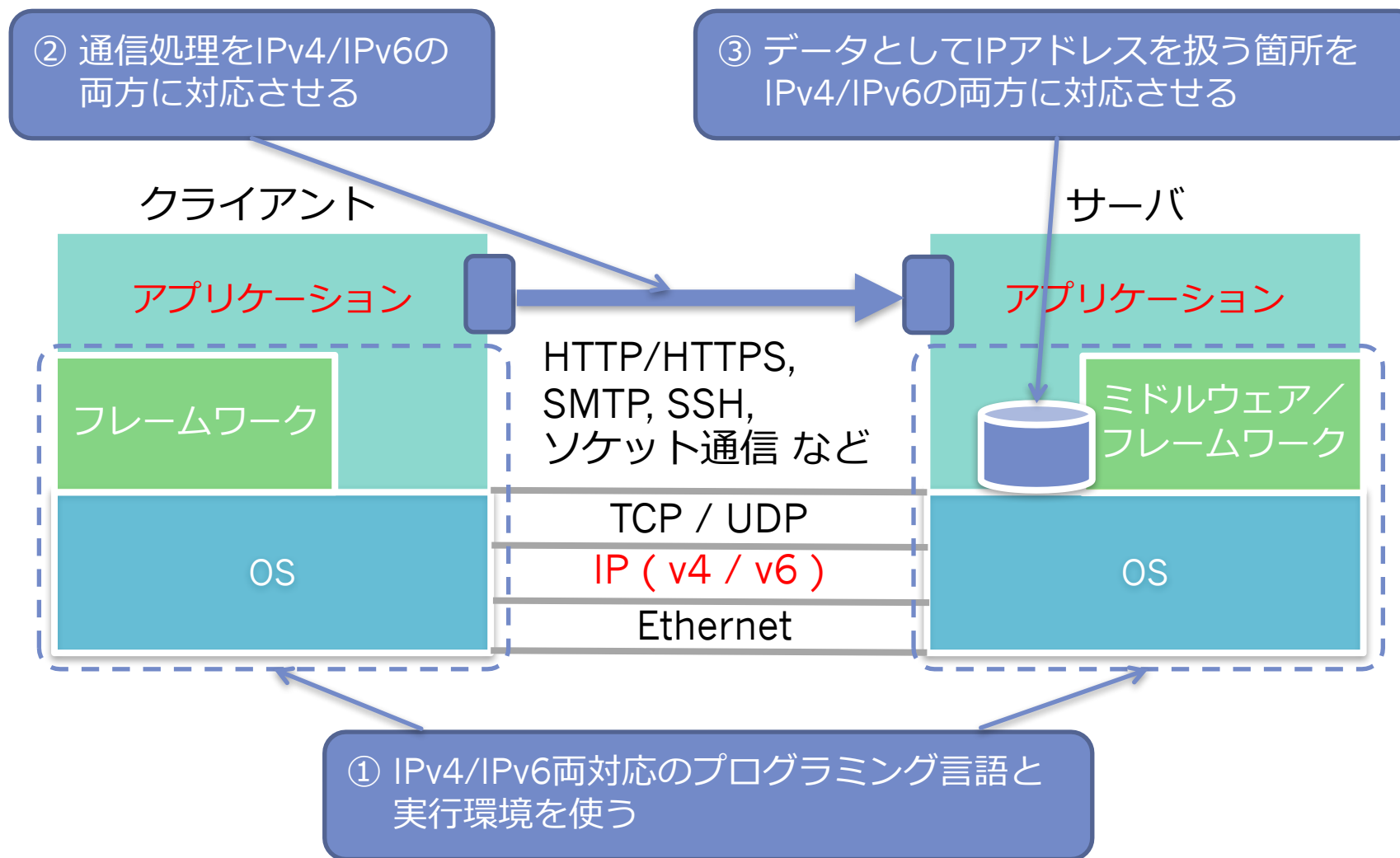
- AndroidはJelly Bean 4.3から対応していた機能
- 464XLAT (RFC6877) とは
 - アドレス変換技術によるIPv4の延命技術
 - CLAT: Customer-side translator
 - PLAT: Provider-side translator
 - CLATにてIPv4を仮想的に提供



アプリケーションのIPv6対応

- IPv6対応は「IPv4とIPv6の両方で動作する」 こと
 - IPv4とIPv6の共存期間への対応
 - これまでのサービスは今後も継続してIPv4にて動作する必要あり
- シングルソースコードで対応する
 - プログラミング言語のIPv6対応によりプロトコル毎の実装が不要に
 - アプリケーションのメンテナンス性を重視しソースコードを分けない
 - プロトコルによって機能差異が発生することを防ぐ

アプリケーションのIPv6対応のポイント



● プログラミング言語と実行環境におけるIPv4/IPv6両対応とは？

- 名前解決機構がIPv4/IPv6両方のアドレスを適切に扱える
- IPv4/IPv6両方で通信できる

これらを満たすプログラミング言語や実行環境を利用する

● 実装上の留意点

● プログラミング言語・実行環境における留意点

- 実際には各プロダクトでサポート状況に差異がある
- 開発するアプリケーションが提供する機能を考慮し個別に判断する必要あり

● プログラミングにおける留意点

- IPv4/IPv6双方に対応するライブラリ、オブジェクト、関数、データ型を使う
- 従来（IPv4のみ）のものとは別に用意されていることがある
 - C: addrinfo構造体, getaddrinfo()
 - Java: InetAddressクラス
 - Perl: IO::Socket::IP など
- アドレス検証、変換などはライブラリを有効活用する

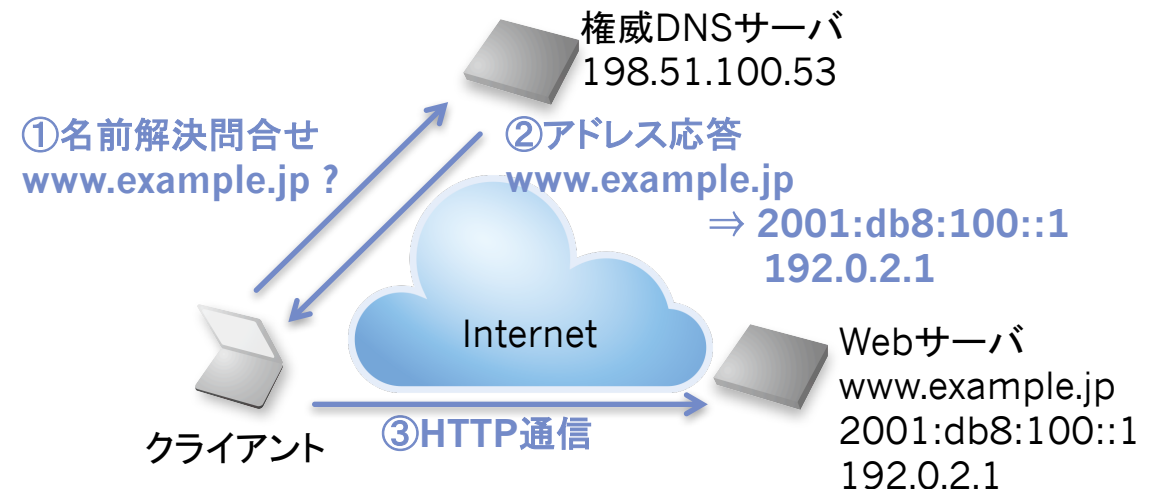
gethostbyname()はIPv6非対応

②通信処理のIPv6対応

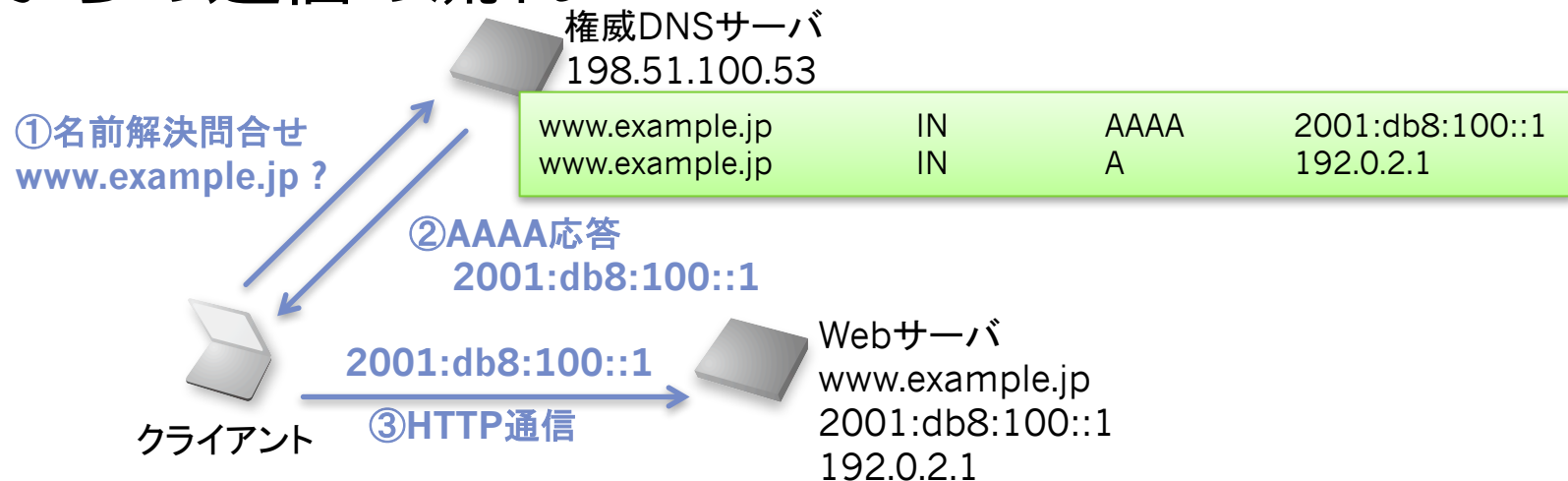
- IPv4とIPv6の両方で通信できることは？
 - クライアント：IPv4およびIPv6で意図するサーバへ接続できること
 - サーバ：IPv4およびIPv6で接続を受付けること
 - IPアドレスを**複数持つ**ことを考慮する！
 - クライアントがどのアドレスにアクセスするかはサーバ側では予測できない
 - **特定のアドレスに依存したシステムを構成すべきではない**

● ネットワークアクセスの作法「名前解決」を使う

- FQDNで接続先を指定し
DNSからアドレスを取得
- **IPアドレスの
ハードコーディングはNG**
- IPアドレスでユーザを
識別するべきではない

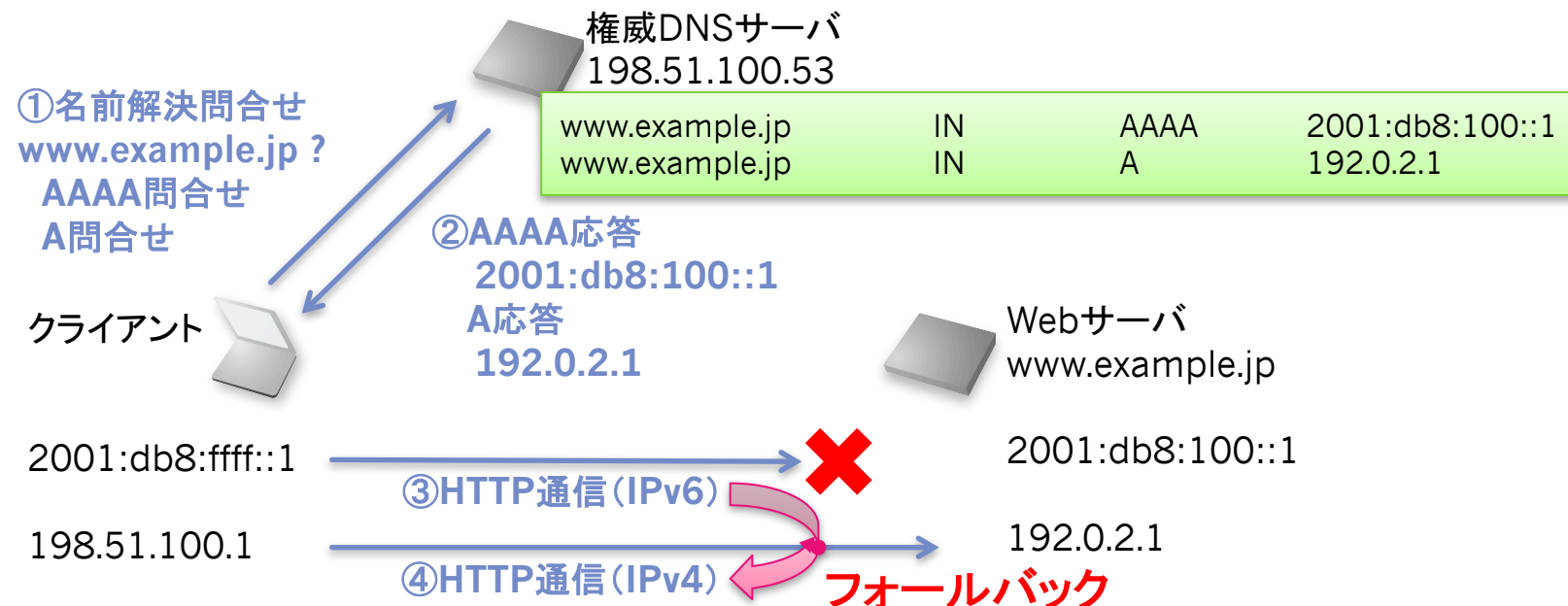


- FQDNで接続先を指定してIPv6通信を行うには
 - DNSにてFQDNからIPv6アドレスが名前解決できることが必要不可欠
- FQDNからIPv6アドレスを名前解決するには
 - 権威DNSサーバ上で接続先サーバのAAAAレコードにIPv6アドレスが登録されていること（AAAAレコードはIPv4におけるAレコードに相当）
 - デュアルスタック環境では権威DNSサーバのIPv6対応は必須ではない
 - クライアントから接続先サーバのAAAAレコードが引けること
- IPv6名前解決からの通信の流れ



- 通信の試行順序の理解：基本的にIPv6を優先
 - RFC6724: Default Address Selection for IPv6
 - RFC6724に準拠していない実装や設定により優先順位が変わる
- クライアントプログラム
 - IPv4/IPv6両方の宛先アドレスに接続できるようにする
 - 接続失敗時には別の宛先アドレスに切替えて接続する（**フォールバック**）
 - プラットフォームが提供するAPI利用することで意識せずにIPv6対応可能
 - iOSアプリ：WebKit, Cocoa URL, CFNetwork.Core Service
 - Androidアプリ：WebView (Android.webkit.WebView), HttpURLConnection (java.net.HttpURLConnection), Socket (java.net.Socket)
- サーバプログラム
 - IPv4/IPv6両プロトコルでの接続を処理する
 - 主要なWebサーバプログラムは対応済み
 - Apache HTTP Server, Microsoft Internet Information Server (IIS), nginx ...

● 接続できない場合に別の接続先への接続に切替える動作



● 想定されるフォールバックの主な原因

- サーバ側の問題 : サーバが当該サービスを提供していない
- 経路の問題 : ネットワークの接続性が失われている
- クライアント側の問題 : サーバへの到達性がないアドレスを選択

③データとしてIPアドレスを扱う箇所の対応

- IPv4/IPv6両対応するため文字列長を確認する
 - IPv6アドレスの文字列長はIPv6完全表記で39文字
 - プレフィックスを加味すると43文字
 - IPv6アドレスを扱えないケース
 - 15文字までの文字列（varchar(15)）として扱う場合
 - 1つの整数として扱う場合
 - 1オクテットずつの4つの整数として扱う場合
- IPv4/IPv6両対応：39文字以内の文字列 **varchar(39)**
- 既存システムの場合：格納領域にIPv6アドレスが収まるかチェック
- IPアドレス型定義がある場合：IPアドレス型を利用
 - PostgreSQLのネットワークアドレス型 など
- IPアドレス型定義がない場合：文字列型でIPv6完全表記を使う
 - IPv6完全表記）2001:0db8:0000:0000:0001:0000:0000:0001

IPアドレス . . .



● Webフォームからの入力値検証

● 入力された文字列がIPアドレスとして取りうる値であることを検証

- IPv4アドレス：0～9とピリオドで構成，オクテット内は0～255の10進数 など
- IPv6アドレス：0～fとコロンで構成，特殊アドレス（IPv4射影アドレス）への対応 など

● 2箇所では実施可能

- ブラウザ側（HTML5のForm Validation 等）
- サーバ側

● アドレス処理ライブラリを利用すると便利

● PHPの例

`filter_var($host, FILTER_VALIDATE_IP, FILTER_FLAG_IPV6)`

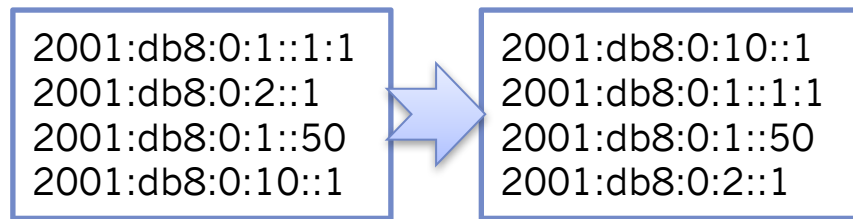
（参考）IPv4射影アドレス

IPv4アドレスをIPv6アドレスとして表現するためのIPv6アドレス。上位80ビットに0、81～96ビット目に1、下位32ビットにIPv4アドレスを埋め込む。機器内部での使用に限られ、パケットの始点/終点アドレスには使われない。

出典：IPv6関連用語集 IPv4-mapped (IPv6) address

● なぜIPv6完全表記を用いるのか

- 省略表示だと整列時にアドレス昇順にならない
- 整列は完全表記で行う必要あり



文字列で整列

● 文字列として扱う場合の注意点

- 省略表記と完全表記の変換にはライブラリを有効活用する
- 古いシステム・ツールではRFC5952に非準拠の省略表記が存在するので注意
 - 2001:DB8:CAFE::1 など
- ログファイル解析への影響を精査する
 - **ログ解析を自作している人は要注意！**
 - アドレス部分の文字列が長くなる、アドレスの区切り文字がコロンに変わる
 - OSSログ解析プログラムでは大抵問題なく処理できる
 - AWStats, Webalizer ...
 - アクセス元の国や地域の解析はできない場合があるので注意

- アプリケーションのIPv6対応の基本方針
 - IPv6対応 = IPv4/IPv6の両方で動作させる
 - シングルソースコードで対応する
- アプリケーションのIPv6対応のポイント
 - IPv4/IPv6両対応のプログラミング言語と実行環境を使う
 - IPv4固有のAPIを利用しない
 - OSによる高レベルなネットワークフレームワークを活用
 - 通信処理をIPv4/IPv6の両方に対応させる
 - IPアドレスのハードコーディングはNG
 - ホスト名・FQDNで接続先を指定する
 - データとしてIPアドレスを扱う箇所をIPv4/IPv6の両方に対応させる
 - 適正サイズのアドレスファミリストレージコンテナを使用

- 2017年はIPv6利用のターニングポイント
 - 米国を中心とした事業者のIPv6対応が加速
 - 大手コンテンツ事業者（Hyper Giants）は対応済み
 - 大手モバイル通信事業者でIPv6対応率80%を超える事業者が登場
 - NGN（フレッツ光ネクスト）利用者のIPv6利用率が50%に到達する伸び
 - 大手ISPにおけるIPv6のデフォルト提供化
 - モバイル事業者のIPv6デフォルト化が開始
 - 2017年度のモデルからIPv6デフォルト対応
 - SoftBankは2016年度夏モデルから一部対応済み
- IPv6がもたらすEnd-to-end通信プラットフォーム
 - NAT越えの壁がなくなり直接通信可能な世界の実現
 - 別途セキュリティの議論は存在
 - 新しいアプリケーションの可能性

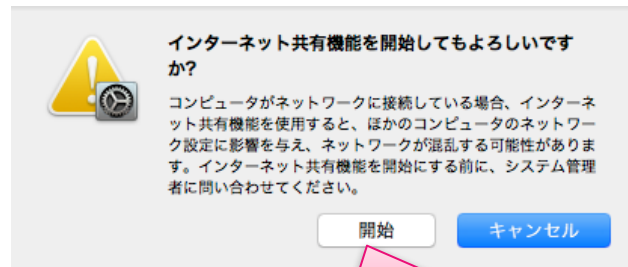
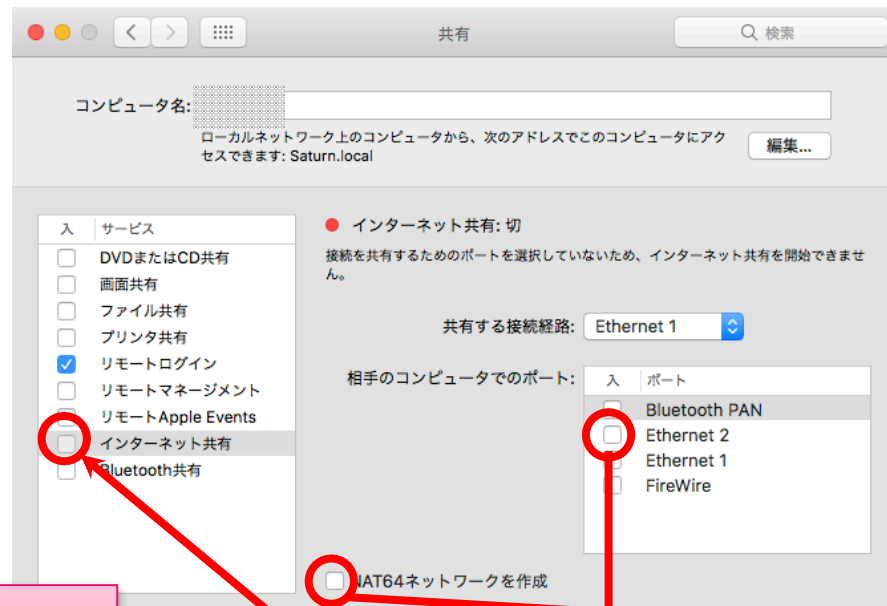
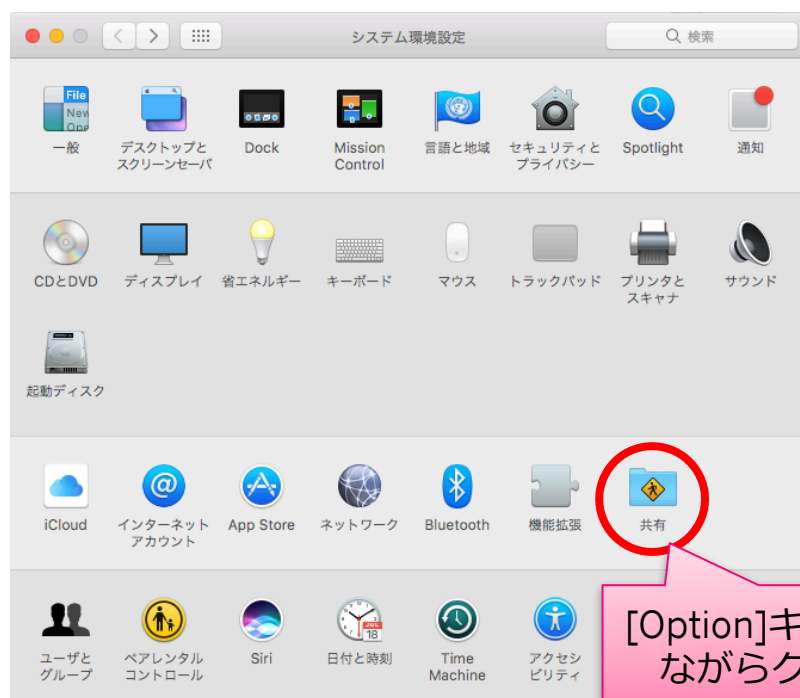
- Test your IPv6 (あなたのIPv6をテストしましょう)
 - <http://test-ipv6.com/> <http://test-ipv6.jp/>
 - World IPv6 Launchの際に用意されたサイト (com版のミラーサイトが存在)
- IPv6 test – IPv6/4 connectivity and speed test
 - <http://ipv6-test.com/> (IPv4とIPv6の到達性確認とスピードテスト)
- Google's IPv6 test
 - <http://ipv6test.google.com> (GoogleによるIPv6接続環境確認サイト)
- IPv6接続確認ツール | ぷらら
 - <https://www.plala.or.jp/ipv6/access/check/>
- IJmio IPv6スピードテスト
 - <http://speedtest6.ijmio.jp>

など

- 6lab – The place to monitor IPv6 adoption
 - <http://6lab.cisco.com/> (IPv6普及状況を示す総合サイト)
- ユーザのIPv6利用状況
 - Google IPv6 (GoogleへのIPv6アクセス状況)
 - <https://www.google.com/intl/ja/ipv6/statistics.html>
 - Measurement | World IPv6 Launch (各ISP毎のIPv6利用率)
 - <http://www.worldipv6launch.org/measurements/>
- ネットワークのIPv6対応率
 - IPv6 Enabled Networks (RIPE NCCによる国別対応状況)
 - <http://www.worldipv6launch.org/measurements/>
- コンテンツプロバイダのIPv6対応状況
 - IPv6普及度計測 | DNSのIPv6登録状況 (jpドメインでのIPv6普及状況)
 - <http://v6metric.jp/html/st04/>

(参考) macOSによるDNS64/NAT64環境の構築

- インターネット共有オプションでDNS64/NAT64を利用可能
 - システム環境設定で[Option]キーを押しながら「共有」を開く
 - 「NAT64ネットワークを作成」チェックボックスが表示される
 - 「NAT64ネットワークを作成」にチェックし「インターネット共有」を選択



● デフォルトのDNS64/NAT64環境の問題点

- 「NAT64ネットワークを作成」だけではIPv4への通信しかできない
- IPv4通信しかできない理由
 - クライアントに割り当てられるアドレスがベンチマークアドレス (2001:2::/64)
 - DNS64でIPv6サーバの名前解決も変換される (dns64-synthallが有効なため)

● IPv6通信も可能にする改造方法

- ① 上位ルータにてクライアントに割り当てるプレフィックスをルーティング
- ② /etc/com.apple.mis.rtadvd.confを変更
 - クライアントに割り当てるプレフィックスとrdnssの値を変更
- ③ /etc/com.apple.mis.unbound.confを変更
 - dns64-synthallの行を削除
 - "interface: ::0" を "interface: [rdnssサーバアドレス]" に変更
- ④ radvdおよびunboundをkillして再起動

※上記設定ファイルはNAT64有効時のみ出現 (NAT64を無効にした時点で設定が初期化され設定ファイルも消失)

お問い合わせ

一般社団法人インターネット協会 IPv6デプロイメント委員会
ipv6_dc@iajapan.org

参考資料：渡辺 露文（IPv6普及・高度化推進協議会），チュートリアル「IPv6対応アプリケーション開発」，
IPv6 Summit in KANAZAWA 2016，
https://www.iajapan.org/ipv6/summit/KANAZAWA2016/pdf/watanabe_kanazawa2016.pdf