IPv4サーバ環境への IPv6導入ガイドライン

IPv6普及・高度化推進協議会

IPv4/IPv6共存ワーキンググループ

サービス移行サブワーキンググループ

目次

- 第1章：Introduction
- 第2章：IPv4アドレス枯渇の問題点
  - IPv4アドレス在庫枯渇が事業者に与える可能性のある影響やIPv6の早期対応のメリットについて説明します
- 第3章: IPv6対応 ~インフラ環境の構築~
  - IPv6環境構築に必要な構成やaddressing,DNSについて説明します
- 第4章：IPv4/IPv6変換
  - IPv4/IPv6変換の仕組みや実現する機器の特徴、導入の長短所について説明します
- 第5章：IPv6対応 ~サーバ編~
  - IPv4/IPv6それぞれもしくはDual stackでのサーバ構築のノウハウとそれぞれの注意点や現在の実装状況について説明します
- 第6章：IPv6対応 ~監視編~
  - IPv6ノードの監視方法やそのポイント、IPv4/IPv6共存環境での監視における注意点を説明します
- 第7章: 今後の課題
- 第8章：参考文献
- 第9章：検討メンバー
- 第10章：謝辞
- Appendix
  - 端末OSのIPv6対応状況
  - トランスレータの標準化動向
  - リバースプロキシの設定例とログ例
  - ロードバランサーの構成例
  - Nagiosでの設定例
  - 用語集
  - Appendix内における参考文献

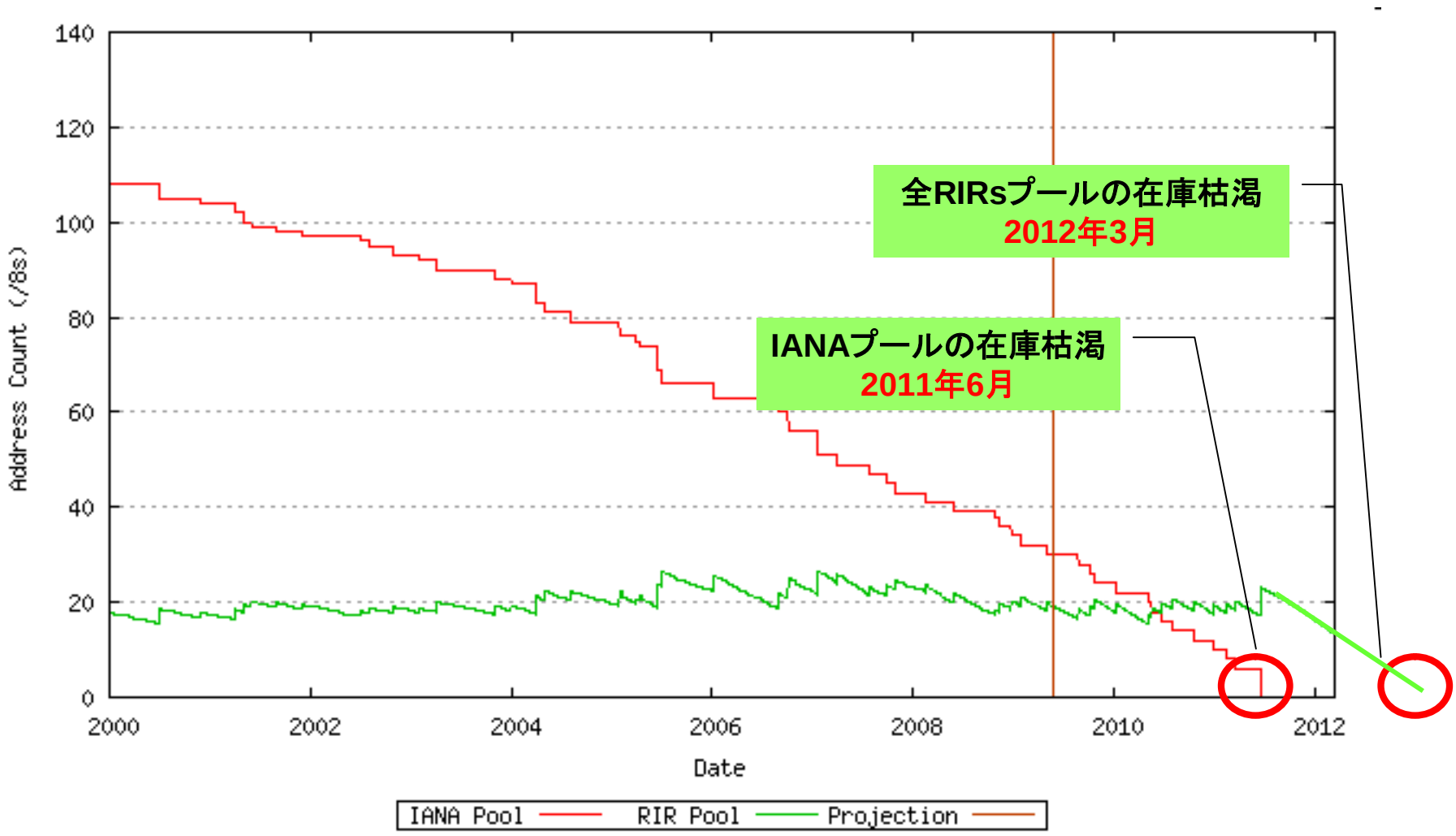
Chapter-1

Introduction

1-1.IPv4アドレスの枯渇状況と社会の動き

- 現在インターネットで利用されているIPv4アドレスの在庫枯渇が本格化しており、国際的な在庫枯渇は2011-12年頃と予測されている
  - (社)日本ネットワークインフォメーションセンターは2007年12月に『IPv4アドレス在庫枯渇問題に関する検討報告書』で下記の提言を行っている
    - IPv4アドレス在庫枯渇期が近い
    - IPv4アドレス枯渇時のIPv4アドレス利用者の分類とその対処法
  - また総務省は2008年6月にインターネットの継続的利用且つ発展を確保する為に『インターネットの円滑なIPv6移行に関する調査研究報告書』で新たなIPアドレス体系(IPv6)への移行を軸としたIPv4アドレス枯渇対応のアクションプランを策定し、発表を行った

1-2.IPv4アドレスの枯渇状況



Geoff Huston "IPv4 Address Report" (http://www.potaroo.net/tools/ipv4/), 2009年 5月28日現在

1-3.IPv4アドレス枯渇とはどういうことか

- 社会全体
  - 現在利用しているIPv4インターネットが無くなるわけではない
- 事業者
  - 新規顧客の獲得や既存サービスの拡張、新サービスの展開時にIPv4アドレスを取得することが困難である
  - 現在IPv4アドレスの移転ポリシーや市場取引等の疑問も行われているが需要に対する確実な供給は保障されない為、期待するのは危険である
  - 新規にIPv4アドレスを取得することが難しくなると新規事業者の参入の障壁となる
  - 新たにインターネットを利用したサービスの展開が不可能
- 利用者
  - 新規利用者がインターネットに接続できない
  - 接続しているISPのIPv4アドレスの取得困難に伴う費用負担を求められる可能性もある

1-4.リスクとチャンス

- IPv4アドレスの枯渇問題に対応しない事は大きなビジネスリスクになる可能性がある
  - 特にホスティング事業者はサーバに利用するグローバル・アドレスの調達は必須であり、IPv4アドレス枯渇に対応しない事はビジネスの発展性の危機となる可能性が極めて高い
- 枯渇対応はビジネスチャンスになる
  - 対応の優劣やサービス内容によっては新たなビジネスチャンスを生む可能性がある

1-5.本書のターゲット

- 本文書はIPv4アドレス枯渇期を迎えるにあたり、インターネットで「Web関連サービス」のビジネスを展開する事業者のIPv6対応について記載したものである

- 本書においてIPv4アドレス枯渇問題に対し、その対策としてLSNの採用等でなくIPv6対応を選択した理由は、LSN等の利用はIPv4枯渇問題においてIPv4アドレスの「延命処置」であり、未来永劫の解にならないという検討結果からIPv6対応を選択したものである

Chapter-2

IPv4アドレス枯渇の問題点

IPv4アドレス枯渇、あるいは新規割当て停止。その時。

2-1.本章について

- この章では、グローバルIPv4アドレスの枯渇（以下、枯渇）が訪れた場合にインターネット関連サービス事業者において、事業上どのような影響が起こりうるかを解説する

- 現在のインターネット関連サービスは、IPv4が前提技術のひとつとして成立していることから、ユーザとして実質的な枯渇である「グローバルIPv4アドレス新規割当ての停止（予告含む）」が訪れた時点から影響が出始めると予測している
  ※次ページ資料参照

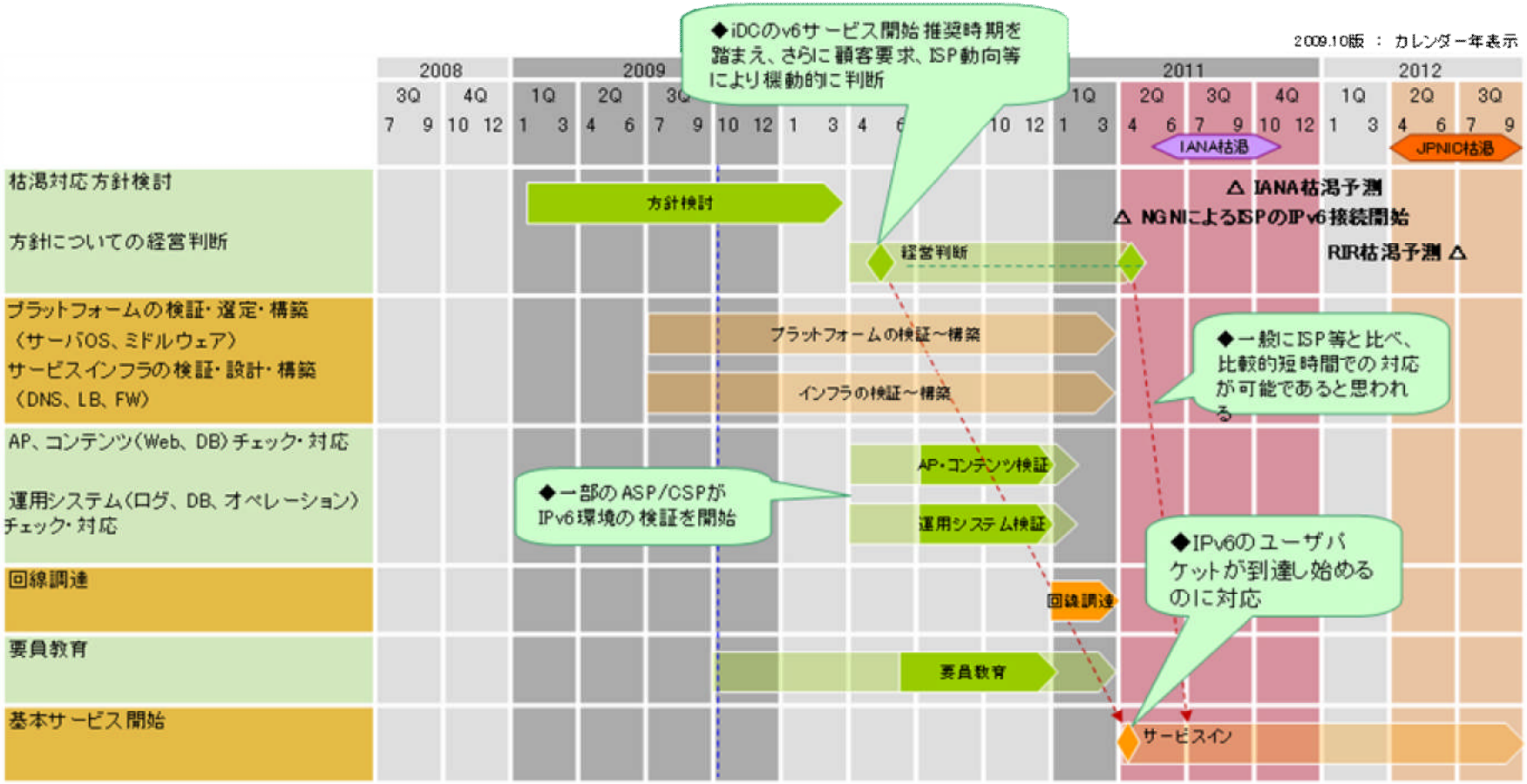
- また、本章でのサービス事業者とは「ホスティングサービス事業者」を指している

2-1.本章について

アクションプラン：サービス（ASP/CSP）関連

------▶ 推奨スケジュール

サービス（ASP/CSP）関連プレーヤーにおけるアクションプラン（基本形）



出典：IPv4アドレス枯渇対応タスクフォース作成の「IPv4アドレス枯渇対応アクションプラン 2009.10版」から引用。

2-2.「サービス事業者」のシステム形態について

- 現在のサービス事業者が顧客に提供している形態は、殆どが下記の区分のいずれかに当てはまる

サービスの種別	IPアドレスの提供形態
専用サーバサービスでの提供	顧客ごとに1つ以上
VPSサービスでの提供	顧客ごとに1つ以上
共用サーバサービスでの提供	収容されているサーバ（システム）で1つを共有。 もしくは、顧客ごとに1つ以上。 ※SSL用のみが個別に割り当てられるケースもある。

- なお、昨今のホスティングサービスでは、サービスの種別による差が小さくなってきており、IPアドレスの提供形態の差も同様になくなりつつある
  いまやIPアドレスもサービスの一要因として大きく存在しており、その結果、ホスティング事業全体でのIPアドレスニーズは増加する一方である

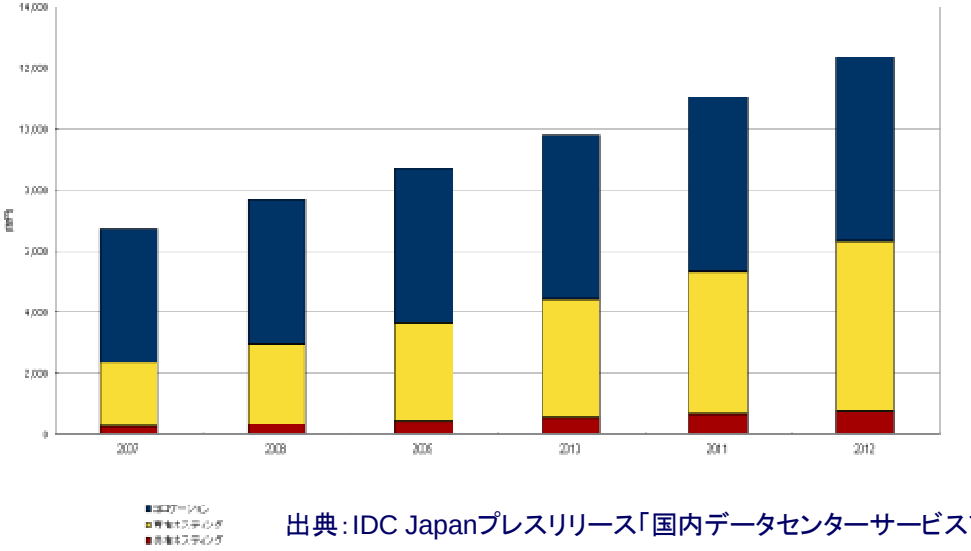
2-3.サービス事業者における問題点

- ホスティングサービス事業においては、サーバの増設・貸与、SSLの販売等の理由から、グローバルIPアドレスの需要は常に高い状態にあり、仮に枯渇してしまった場合、事業継続に支障が出る事態に陥る

- ホスティングを含むSaaS/PaaS/IDC市場としては年間20%程度の成長率が継続して維持・予測されていることから、グローバルIPアドレスニーズの鈍化は現実的に想像しにくい

2-4.国内データセンターサービス市場規模予測

- IDC Japanの予測では、2008年の同市場は前年比13.3％増の7,669億円
- 2008年から2012年まで年間平均成長率12.7％で市場規模が拡大
- 2012年には1兆2千億円を超えると予測
- 2008年10月のリーマンショックに端を発した世界経済危機の同市場への影響は小さいと見ている
- 経費抑制圧力が強まっているものの、企業がデータセンタに委託しているシステム運用規模を急激に減少させる可能性は少なく、むしろ既存システムの自社運用にかかるコストを抑制するために外部のデータセンタサービスの利用は堅調に拡大すると、IDCでは見ている

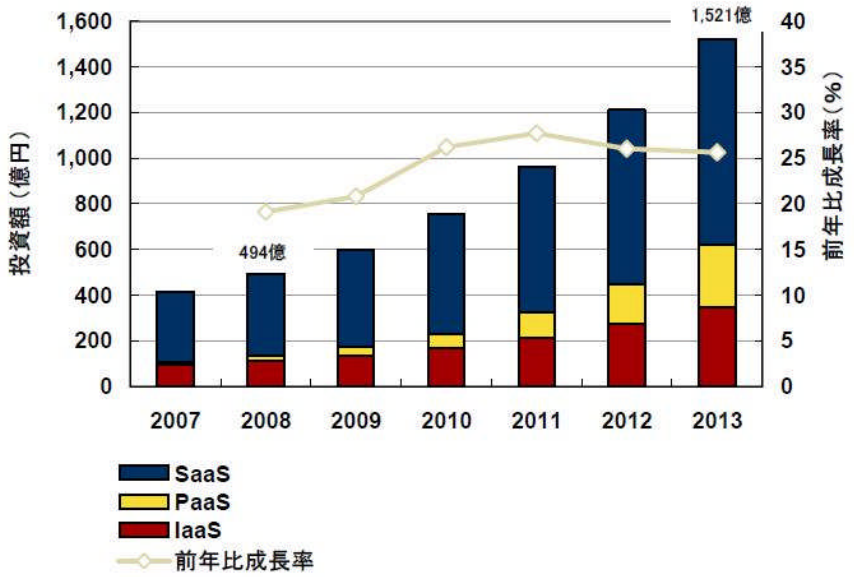


出典：IDC Japanプレスリリース「国内データセンターサービス市場予測を発表」2009年4月

2-4.国内SaaS/XaaS市場規模予測

- IDC Japanの予測では、2008年の国内SaaS/XaaS市場規模は、494億円
- 2009年以降も国内SaaS/XaaS市場は急速な拡大を継続し、2013年の同市場規模は2008年比3.1倍の1,521億円に達する見込み
- 現在の経済低迷は、短期的には国内SaaS市場の拡大を阻害する要因となっているが、長期的にはコスト削減、拡張性、迅速性といった利点から、同市場の拡大を加速する

国内SaaS/XaaS市場 投資額予測、2007年～2013年



出典：IDC Japanプレスリリース「国内SaaS/XaaS市場規模予測を発表」2009年7月

Source: IDC Japan, 7/2009

2-5.サービス事業者における問題点

- 枯渇となった際の代表的な問題点
  - SSLサービスの新規販売がほぼ不可能になる
  - 専用サーバサービス、VPSサービスの新規提供がほぼ不可能になる
  - SSLサービスの新規利用が困難になる（※都度割り当てる場合）

- 新規契約と解約のバランスで「すべての新規契約にグローバルIPアドレスを割り当てられなくなる」という事態は、容易には起こらないと想像できるが、ネットワークセグメントの利用実態や事業的な事情（ユーザへ貸与したグローバルIPアドレスを安易に変更できない）、ネットワーク設計上の理由により、グローバルIPアドレスの再利用が難しいケースは存在する

2-5.サービス事業者における問題点

- 枯渇にまつわる諸問題を解決するためには
  - IPv6への対応（サービスの共存）
  - 企業内でのIPv4ネットワークの再設計と再割当て
  - システムの再設計による技術的な解決

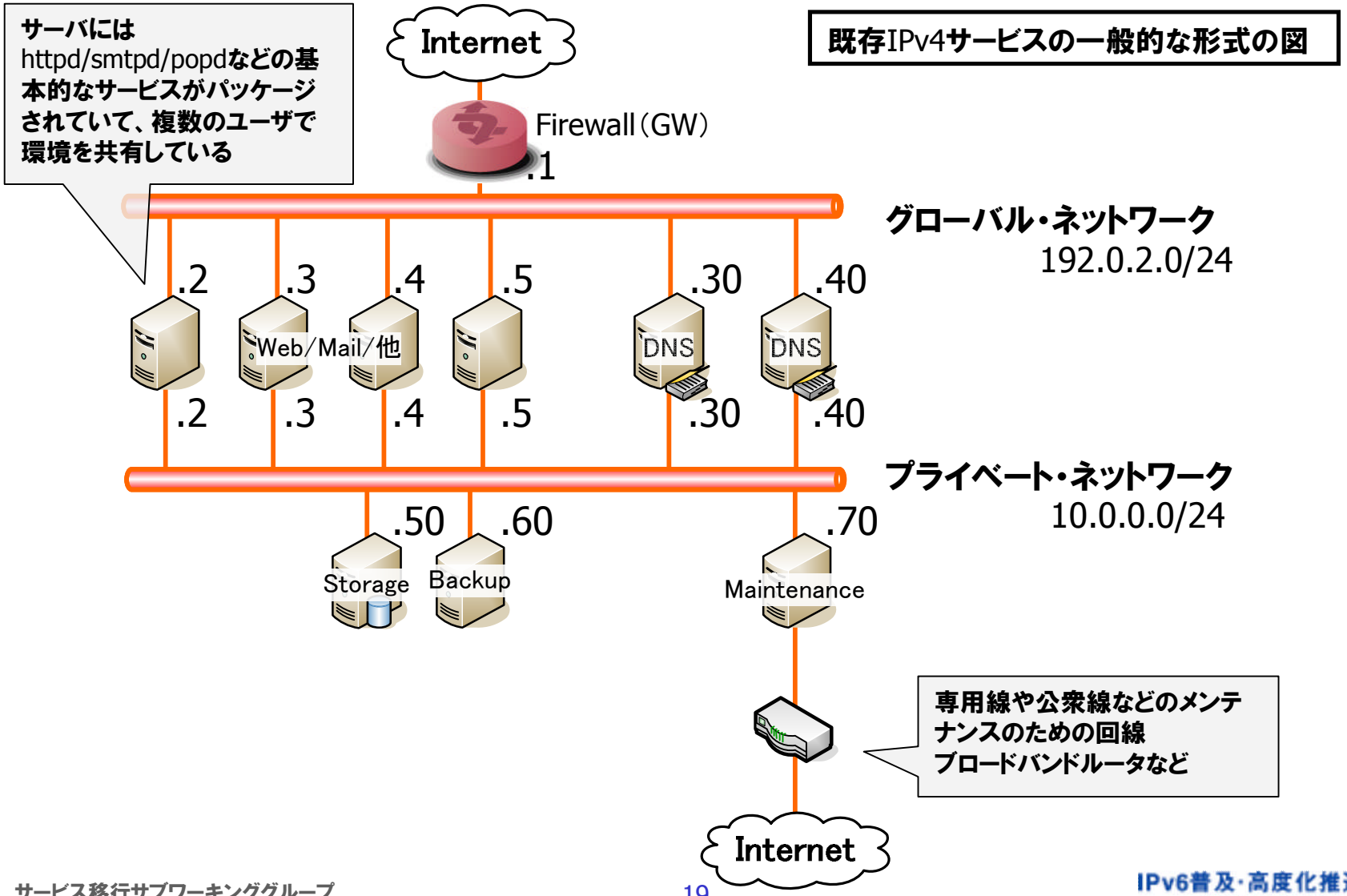
  といった施策のうち、少なくとも一つ以上を行う必要がある

- 無視できない可能性として、枯渇対策とIPv6の一般的な利用のタイミングが合致せず、枯渇と同義の事態が発生するがある
  如何にホスティングを含めたサービス提供側がIPv6に対応しようと、アクセス系からのエンドユーザの接続がシームレスに行われなければ、それをサービス化する意味が半減してしまい、IPv4ニーズも鈍化しない
- また、現在組織間でのグローバルIPv4アドレスの取引について検討が始まっている

2-6.早期のIPv6対応を勧める理由

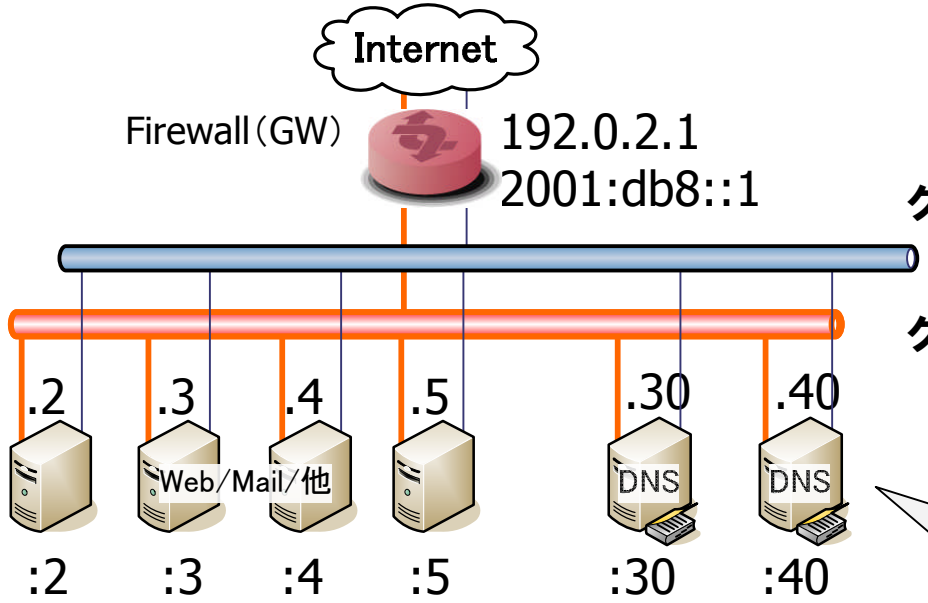
- この項までに説明してきた諸問題において、IPv6化による対応で解決することがもっとも効果的であるが、事業の水準で提供するためのノウハウが乏しいのが現実でもある
- 現段階から試験などを通じて対応を開始することで、その来る日にスムーズに共存体制へ移行できると想像している。その結果、利用者に高い価値を提供することができるのである

2-7.既存サービスのモデルとIPv6対応



2-7.既存サービスのモデルとIPv6対応

グローバルネットワークをデュアルスタックで構築する場合



IPv6 Network

IPv4 Network

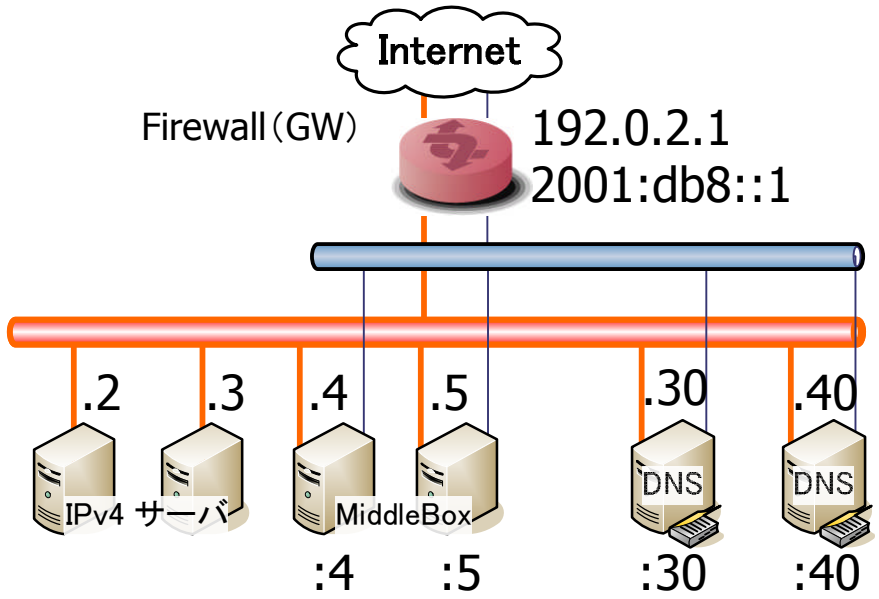
グローバル・ネットワーク（IPv6）
2001:db8::/64

グローバル・ネットワーク（IPv4）
192.0.2.0/24

グローバル側のネットワークのみをデュアルスタック化し、サービスを提供するFirewall(GW)は図ではデュアルスタックだが、任意で選択してよい

※プライベート・ネットワークは省略

2-7.既存サービスのモデルとIPv6対応



※プライベート・ネットワークは省略する

middle boxを用いてIPv6対応をする場合

IPv6 Network

IPv4 Network

グローバル・ネットワーク（IPv6）
2001:db8::/64

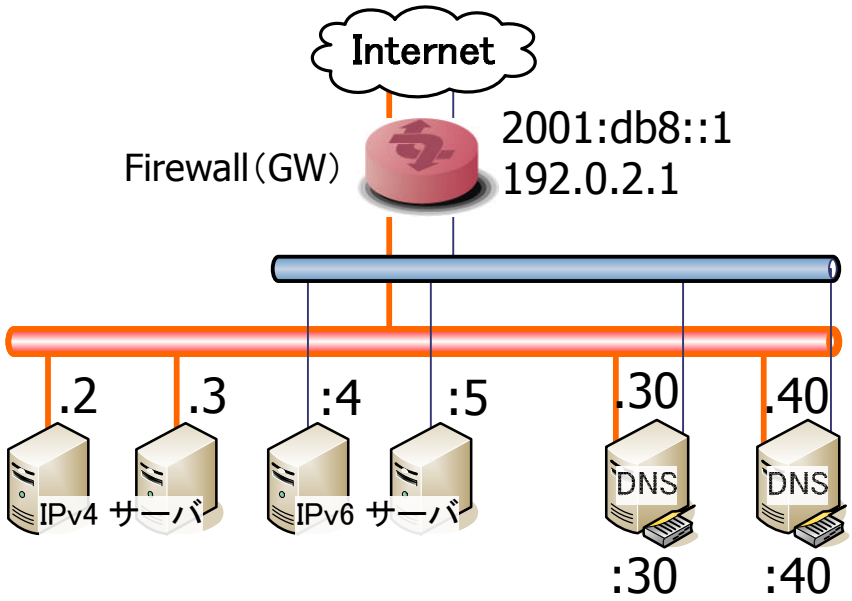
グローバル・ネットワーク（IPv4）
192.0.2.0/24

IPv6をmiddle boxで受け、IPv4サーバへ代理接続をする

ただし、DNSはIPv6からの問い合わせに対応するためにデュアルスタックである必要がある
Firewall(GW)は図ではデュアルスタックだが、任意に選択してよい

2-7.既存サービスのモデルとIPv6対応

IPv4とIPv6とでサーバを分けて対応をする場合



IPv6 Network

IPv4 Network

グローバル・ネットワーク（IPv6）
2001:db8::/64

グローバル・ネットワーク（IPv4）
192.0.2.0/24

※プライベート・ネットワークは省略

グローバル側のネットワークにIPv6ネットワークを新設する

ただし、DNSはIPv6からの問い合わせに対応するためにデュアルスタックである必要がある
Firewall(GW)**は図ではデュアルスタックだが、任意で選択してよい**

Chapter-3

IPv6対応 ~インフラ環境の構築~

3-1.IPv6環境を構築する準備

- IPv6のインフラ環境を構築する為には主に下記を調達する必要がある
  - IPv6アドレスの取得
  - IPv6 Internetと接続する
    - 足回り回線の調達
    - 接続性の調達
  - IPv6対応機器を用意する

3-2.IPv6アドレスの調達

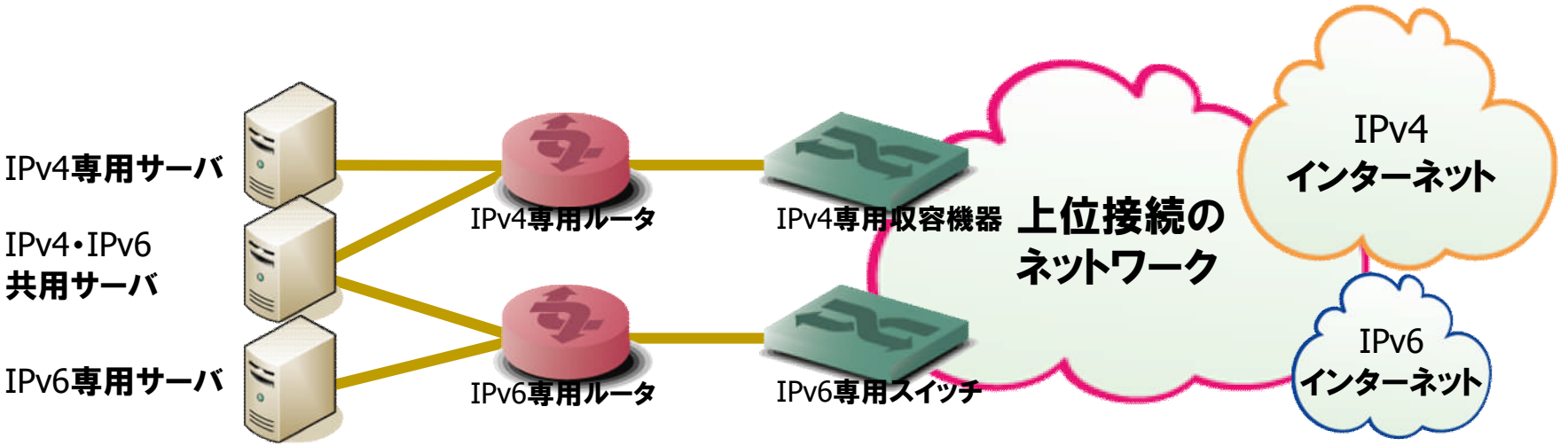
- IPv6アドレスは現在接続している(あるいはIPv6ネットワーク構築予定)のデータセンター等の接続事業者に割り当て申請を行う
- (社)日本ネットワークインフォメーションセンターの　『IPv6アドレス割り振りおよび割り当てポリシー』では事業者からの割り当てサイズについて下記のように明記されている。事業者は独自のアドレス割り当てポリシーをもち、それに従い顧客へのアドレス割り当てを行うが、そのポリシーは(社)日本ネットワークインフォメーションセンターのポリシーを順守している
  - エンドユーザはIP指定事業者またはIP指定事業者から割り振りを受けたISPから割り当てを受ける。実際の割り当てサイズは、/48以上の割り当てが正当化されるエンドサイトを除いて、最小/64(1サブネットのみがエンドサイトとして想定される場合)から原則として最大/48までの範囲でIP指定事業者または割り当てを行うISPが判断するものとする

3-3.接続性

- この章ではIPv6ネットワークへ接続するにあたりネイティブ・デュアル・トンネルの各手法とそれらの長所・短所について解説する

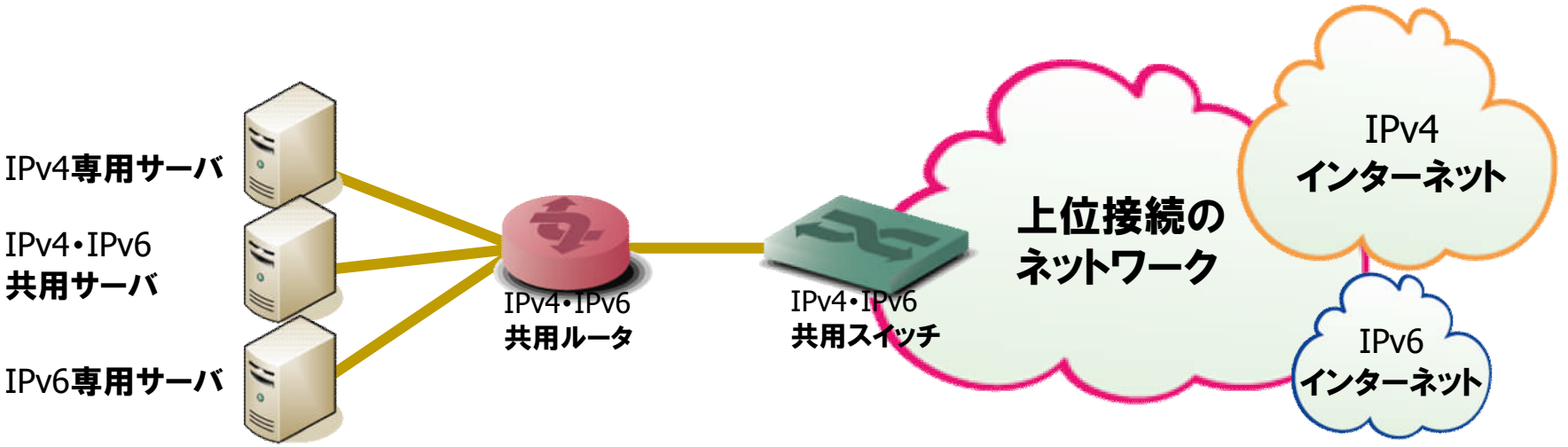
3-3-1.接続の方法 -ネイティブ接続-

- ネイティブ接続
  - 概要：
    - 上位接続とIPv4/IPv6が完全に別設備で接続する方法
  - メリット：
    - IPv4設備と完全に別設備を用意するためIPv6導入によるネットワークトラブルがIPv4設備側へ広がることを防げる
  - デメリット：
    - 多くの場合別途回線及び機器設備を用意する為のコストが必要



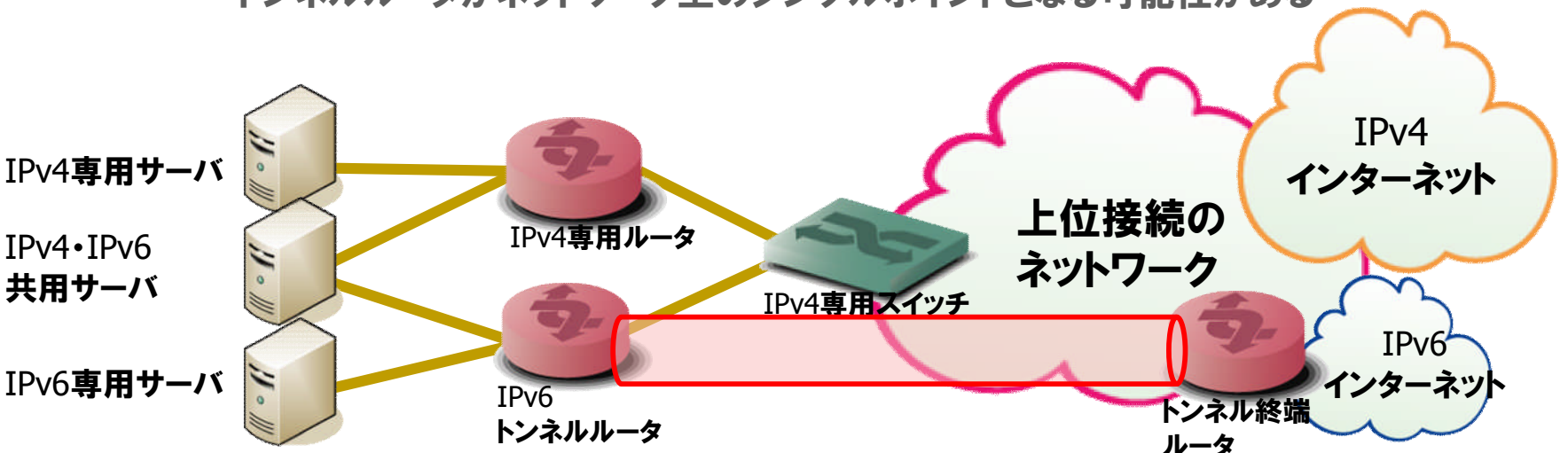
3-3-1.接続の方法 -デュアル接続-

- デュアル接続
  - 概要：
    - 上位接続とIPv4/IPv6を同じ設備で接続する方法
  - メリット：
    - IPv6の設定のみである場合、新たな機器や回線を用意する必要がなくコストインパクトを小さくすることが可能
  - デメリット：
    - IPv4接続で利用していた機器へIPv6設定を導入する場合、必要とする対応機能の有無によって機器のリプレイスが必要
    - IPv6、IPv4それぞれの障害により互いに影響が出る可能性もある



3-3-1.接続の方法 -トンネル接続-

- トンネル接続
  - 概要：
    - IPv6をIPv4パケットでカプセリングすることで接続する方法
  - メリット：
    - 既存IPv4ネットワークとは別にスモールスタートが可能
  - デメリット：
    - IPv4でのカプセリングを行うためトンネルルータの性能によってはネットワークのパフォーマンス低下を考慮する必要がある
    - 本構成の場合、カプセリングによるパケットサイズ制限によりIPv6ネットワーク上でのPath MTU Discovery問題が起きる可能性がある
    - トンネルルータがネットワーク上のシングルポイントとなる可能性がある

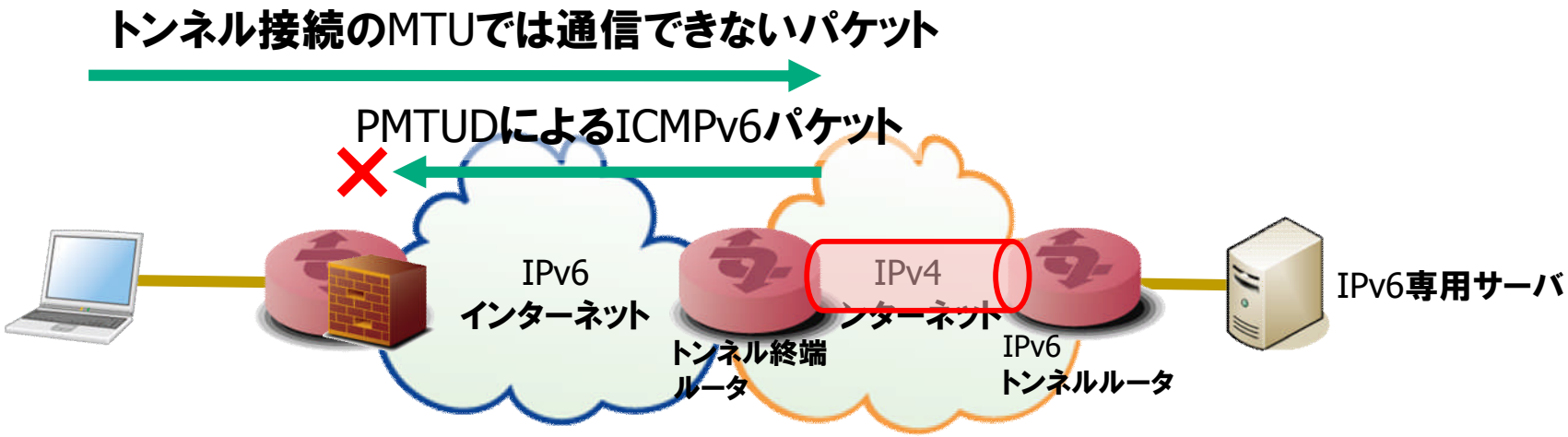


3-3-2.トンネル接続での Path MTU Discoveryの留意点 ①

- 通信経路上のフラグメント処理に関する問題はIPv4ネットワークでも存在していた
- IPv4ネットワークでは、ユーザ側でmssサイズを指定することで本問題を回避する方法や経路上のルータでフラグメント処理を行うことで回避することができた
- IPv6ネットワークでは、経路上のルータでフラグメント処理することは許可されていない為、Path MTU Discoveryによる送信元でのMTUサイズの調整が必要である

3-3-2.トンネル接続での Path MTU Discoveryの留意点 ②

- トンネル接続の場合、物理インターフェースよりもMTUサイズが小さくなる為、Path MTU Discoveryによる通信が不可能になる問題が起きやすくなる
- この場合において、ユーザ側ネットワークでICMPv6のフィルタが行われている場合に利用ユーザとの通信ができなくなることが考えられる



3-3-3.比較表

- 各手法の特徴について下記にまとめる

手法	長所	短所	導入の容易さ	将来性
トンネル	既存IPv4に影響を与えずに導入が可能	パケットのカプセリング処理が必要であるため、機器の性能低下やトラフィック増加を伴う	○	×
ネイティブ	IPv4への影響の最小限に抑えることが可能	回線・機器をIPv6専用に用意するため対応コストが必要	△	○
デュアル	IPv6設定のみで対応可能	IPv4ネットワークへの影響	△	◎

- 各手法に対して特徴を理解して、自ネットワークへ適切なIPv6対応を行うことが重要である

3-4.機器の調達 ①

非対応機器の有効活用

- 既存機器のIPv6アップグレードが可能であれば継続利用
- ネットワークスイッチでは、管理性を重要視しない部位で利用
- その他、共存期間にIPv4ネットワーク部位で継続利用

製品寿命(償却)を考慮した機器の有効活用

- 既存機器のIPv6アップグレードが可能であれば継続利用
- ネットワークスイッチの場合、管理性を重要視しない部位で利用
- その他、共存期間にIPv4ネットワーク部位で継続利用

新規IPv6対応機器導入

3-4.機器の調達 ②

- IPv6対応機器の現状調査
  - 機器メーカーによる実装状況に違いが存在する
  - IPv6対応予定とされている機器が存在する
    - ソフトウエアのみのアップデート対応（ハードウエア対応済み）
    - モジュール交換によるアップグレード対応

- IPv6 Ready Logo Programを参考
  - Phase-2ロゴ取得製品からの選択

    主に考慮すべき条件
    - IPv6 Core Protocols
      - http://www.ipv6ready.org/?page=documents&tag=ipv6-core-protocols
    - Management (SNMP-MIBs)
      - http://www.ipv6ready.org/?page=documents&tag=phase-2-snmp
  - 運用面と管理面を重視した選択要件の確認

3-4.機器の調達 ③

- IPv6新規導入時には機器評価を実施

  はじめてIPv6システム構築するにはカタログ仕様ではなく、実機による評価が重要となる

  - IPv6対応追加ライセンス入手状況の確認
    - 標準対応機器の場合には、ファームウエア・バージョンの確認
  - IPv6基本設定と、動作及び性能の確認
    - IPv6フォワーディング性能
      - 方式により性能が大きく異なる（ハードウエア、ソフトウエア、複合の各方式）
  - アクセスコントロール機能の動作確認
    - IPv6トラフィッククラス制御
    - IPv6フローラベル制御
  - 管理機能の動作確認（必要性の確認を含む）
    - SNMPv2c、SNMPv3
      - IPv6 MIBの実装状況
      - RFC 3416の実装状況 (将来サポートの場合あり)

3-5.アドレス設計

- 基本的な考え方
  - 現在、IPv6は立ち上がり時期であるため、現在の設計が将来にわたり通用する可能性は低い
  - アドレス設計の大枠については現在の事業計画に合わせて固めておくと良い
  - 詳細設計については設計変更をせざるを得ないことを前提としておくと良い

3-5-1. プレフィックスのアサイン

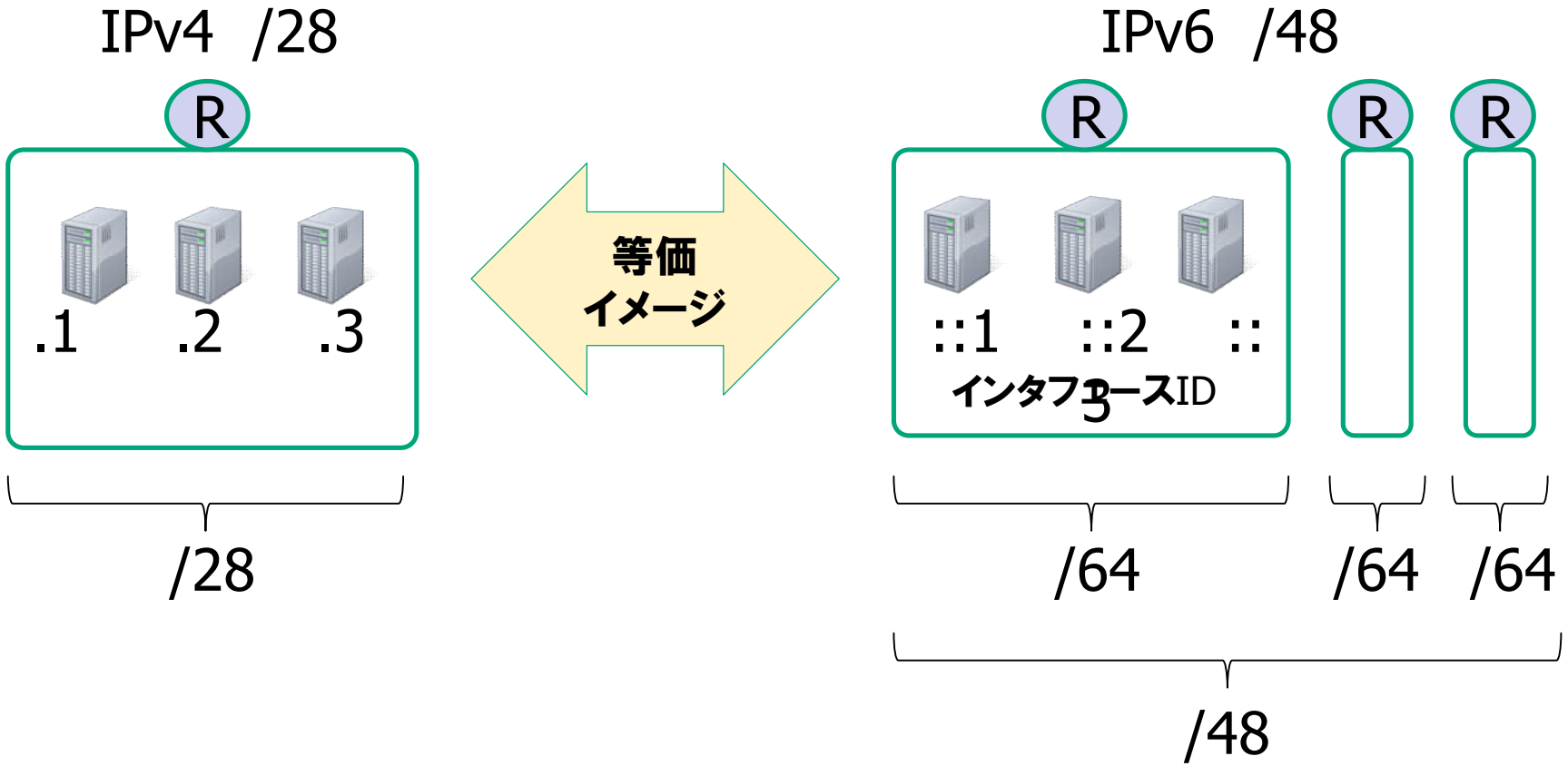
- オペレーターが解りやすく、管理DBがシンプルになるように、16進数表記で区切れるサイズでプレフィックスを区切るべきである
  - 大きな区切り
    - /32、/48、/64 で区切ると解りやすい
  - 中間のサイズの区切り
    - /32、/40、/48、/56、/64
  - 小さなサイズの区切り
    - /32、/36、/40、/44、/48、/52、/56、/60、/64

3-5-2.各セグメントのプレフィックス長

- /64を基本とする
  - /48を保有している場合でも/64を割り当てる
- /48の先頭の/64を使わないことを推奨する
  - 0::/64となり、紛らわしいため

3-5-3.IPv6とIPv4の違いのイメージ

IPv6はIPv4と比べて拡張性で優れている



3-5-4.128bitの使い方

IPアドレス

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX /128

プレフィックス　　インタフェースID

/32 （APNIC/JPNIC等から指定事業者に割り振られる基本サイズ）

XXXX:XXXX:0000:0000:0000:0000:0000:0000 /32

/48 （指定事業者からホスティング事業者に割り当てられるサイズ）

XXXX:XXXX:XXXX:0000:0000:0000:0000:0000 /48

/64 （データセンタ内で利用されるサイズ）

XXXX:XXXX:XXXX:XXXX:0000:0000:0000:0000 /64

3-5-5.ネットワークが二重化・三重化等の場合

- それぞれのネットワークにおいて同じルールで割り当てると良い

3-5-6.数字・文字列

- IPv6アドレスの一部にcafeなどの有意コード(意味を持つ文字)、AS番号、Sub-AS番号などを埋め込むと、解りやすい
- 将来ネットワークが大きくなる場合、設計が破綻する可能性があるため注意が必要である
- 例
  - 2001:db8::cafe:0001
    - IPv6アドレスにcafeを埋め込む例
  - 2001:db8::(ポート番号):0001
    - IPv6アドレスにポート番号を埋め込む例

3-5-7.機器への設定方法

- スタティックで設定すると良い
  - 詳細は5章で解説
- 手入力を避けると良い
  - IPv6アドレスは長く、誤入力を避けるため
- 入力したアドレスを十分目視確認すること
  - コピー＆ペーストをした場合、アドレス後半がペーストできないことなども考えられる

3-5-8.アドレス管理

- 省略表記について
  - 128ビット全てを明記して管理すると良い
    - 「::」により省略すると桁の区切りが動くため、人間が読み取りにくくなる

3-5-9.その他

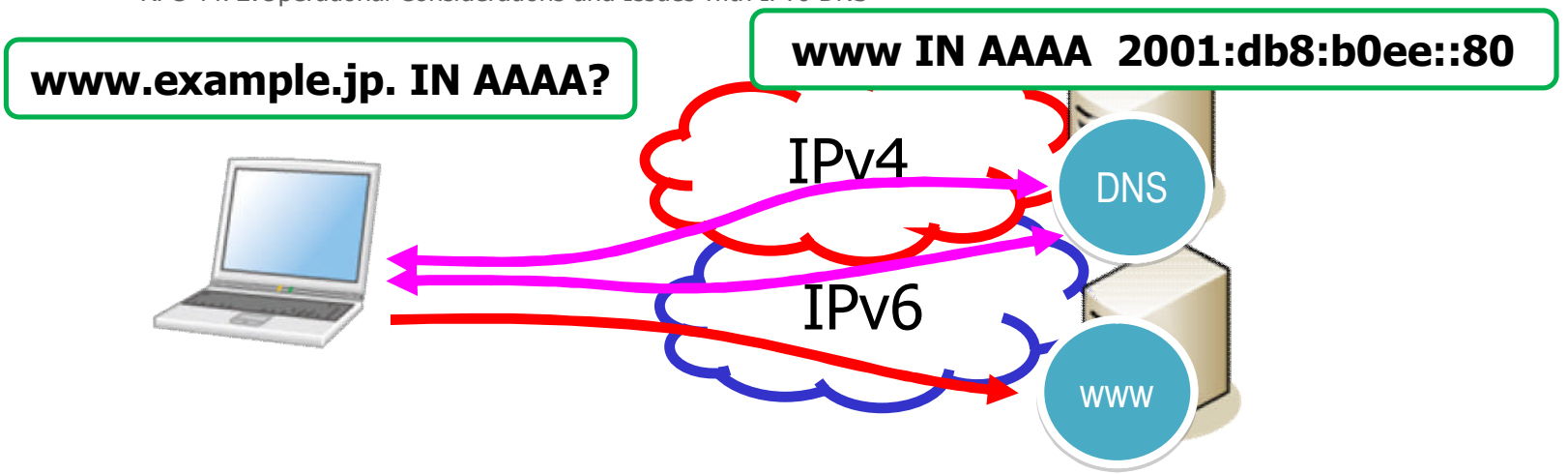
- 割り当てルール
  - JPNIC/APNIC等のレジストリーのポリシーを理解・遵守し、必要以上のサイズを確保するべきではない

3-6.DNS

- ホスティングサービス事業者(及びその利用者)にとってDNSのIPv6対応には2通りの意味が存在する
  - IPv6対応のサーバについて、コンテンツDNSに「AAAA」レコードを登録する
  - コンテンツDNSでIPv6のクエリを受け取れるようにする
- 以下、この2通りのDNSに関わるIPv6対応について説明する

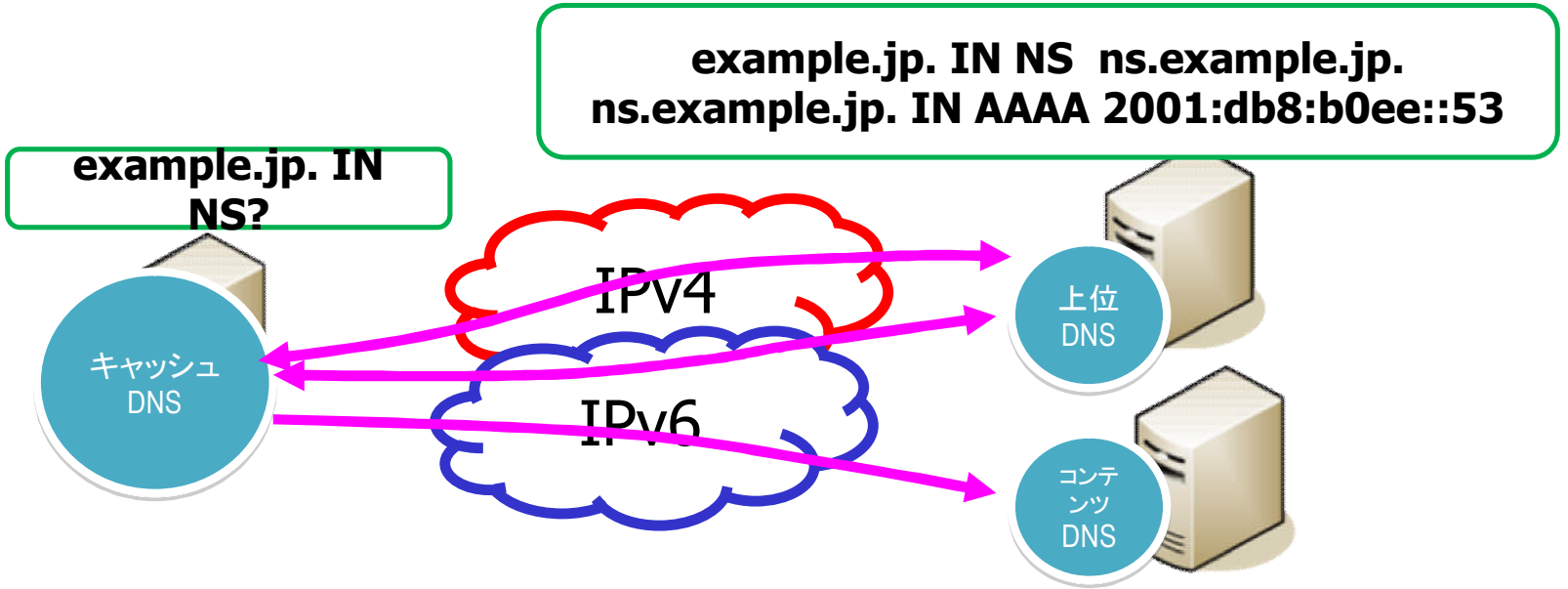
3-6-1.DNSのIPv6対応 ①

- AAAAレコード対応
  - IPv6ホストの名前解決
    - IPv6に対応したサーバについてIPv6でアクセスさせるためにAAAAレコードを登録する
- IPv6からのクエリ対応
  - ドメインに対するコンテンツDNSサーバはIPv4/IPv6双方に対して名前解決を行えるように運用する必要がある
    - RFC 3901：DNS IPv6 Transport Operational Guideline
    - RFC 4472:Operational Considerations and Issues with IPv6 DNS



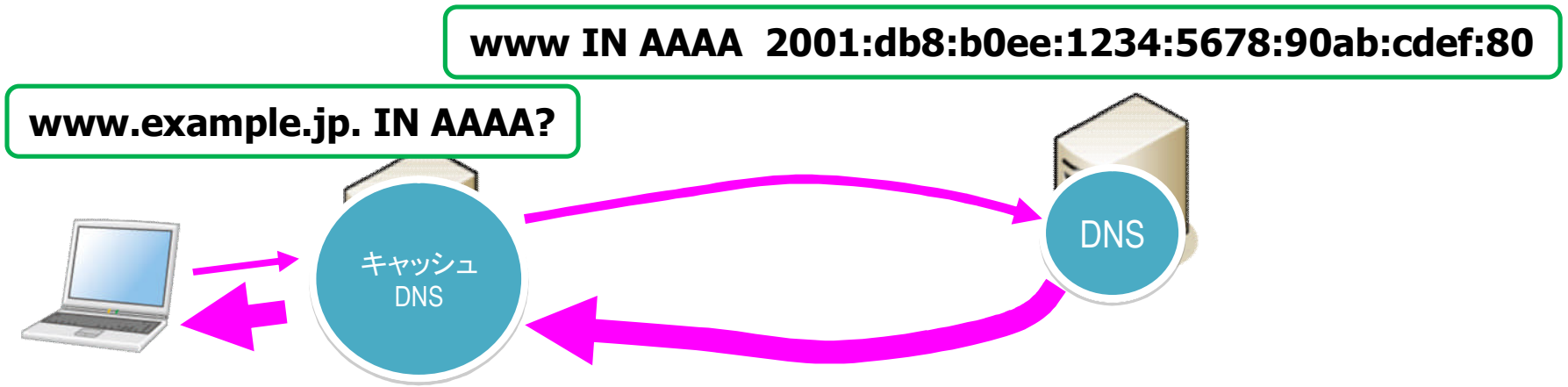
3-6-1.DNSのIPv6対応 ②

- 上位ゾーンに対してのNS登録
  - Glueレコードの登録
    - レジストラ業務を行っている場合、ドメインのNSにIPv6アドレスも登録できるように準備する必要がある(※Glueレコード)
    - GlueレコードにIPv6アドレスを登録できなければコンテンツDNSサーバへのIPv6を用いたクエリは発生しない



3-6-1.DNSのIPv6対応 ③

- EDNS0対応
  - RFC 2671 - Extension Mechanisms for DNS (EDNS0)
  - 512bytesに制限されていたDNSのUDPパケットサイズを拡張する
  - DNSをIPv6に対応させるためにはEDNS0対応が必須
    - RFC 3226-DNSSEC and IPv6 A6 aware server/resolver message size requirements
  - IPv6を記述したRR(※)の解決を試みた場合、返答のパケットサイズが大きくなってしまうことが考えられているため
    - ※AAAA、PTR等



3-6-2.留意点

- ホスティング事業者が提供するコンテンツDNSサーバについては、IPv4/IPv6双方の接続性を持ち、どちらからの名前解決要求に対しても同じ内容を返すことが重要
- レジストラサービスを行っている場合には、ドメインのNSに対してIPv6アドレスを登録できるようにする
- また、レジストラサービスを行っていない場合は、自社が利用しているレジストラがNSに対してIPv6アドレス登録を受付けているかを確認する必要がある
- 実際にコンテンツDNSサーバにIPv6を利用したクエリが発生するのは、エンドユーザーが利用するキャッシュDNSがIPv6に対応してからになるため、エンドユーザー環境に左右されずに安定した運用を行うためには、コンテンツDNSはIPv4/IPv6双方に対して同じ内容の返答を行うようにすべきである

Chapter-4

IPv4/IPv6変換

4-1.変換の必要性

- IPv4とIPv6との間には互換性がない
- そのため、それぞれのプロトコルに対応したサーバが必要となる
  - IPv4/IPv6 dual stackサーバを設置
  - IPv4サーバとIPv6サーバの双方を設置

IPv6サーバを用意できなければ、
IPv4トラフィックをIPv6へ変換する装置が必要となる

- 本書ではそれらの装置を「middle box」と呼称する
  - middle boxがIPv4/IPv6変換することによって、IPv6トラフィックをIPv4サーバへ到達させる

4-1-1.middle boxのメリット

- 段階的にIPv6への移行が可能となる
  - 初期コストの抑制が可能である
  - インフラのIPv6対応を段階的に行うことができる
  - 人員教育の実施も移行に併せて実施できる
- 最小限の変更でIPv6対応が可能となる
  - 既存環境への影響範囲を局所化できる
- 既存IPv4環境を維持しながらIPv6へ移行できる

4-1-2.middle boxの前提条件

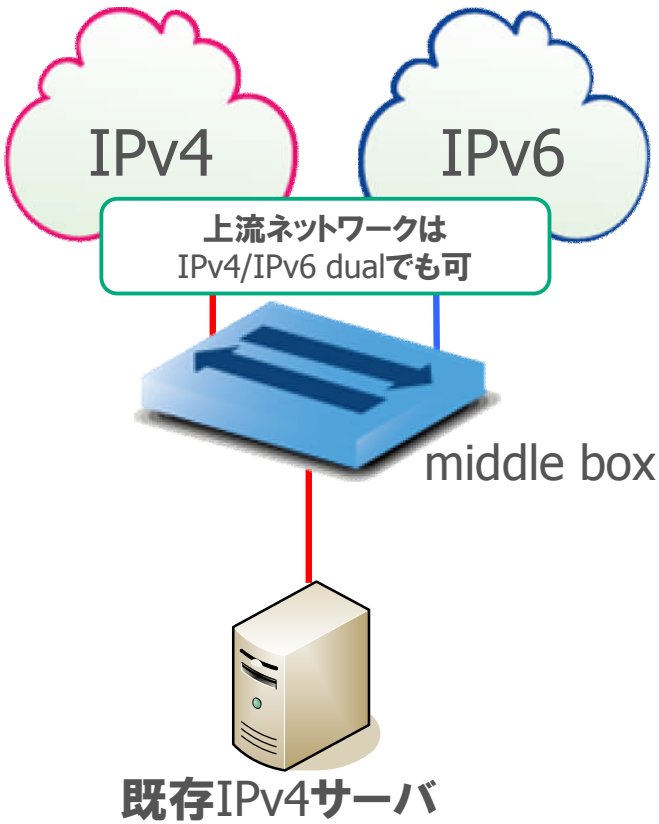
- あくまで移行するための技術である
  - middle boxを使用することによる問題がある
  - 対応できないアプリケーション・サービスも存在する
- middle boxを使用することによる影響を考慮した上で導入すべきである
  - アプリケーション
  - 機器の数
  - 構成の複雑さ
  - パフォーマンス

4-2.middle boxのトポロジ

- このガイドラインでは3種類のmiddle boxを用いたトポロジを提案する
  - トポロジA
    - サーバへのIPv4、IPv6全てのトラフィックをmiddle box経由で到達させる
  - トポロジB
    - 既存サーバにネットワークカードを追加し、IPv6からmiddle boxを経由して到達するトラフィックを収容する
  - トポロジC
    - 既存の構成を変更せず、middle boxをアドオンする

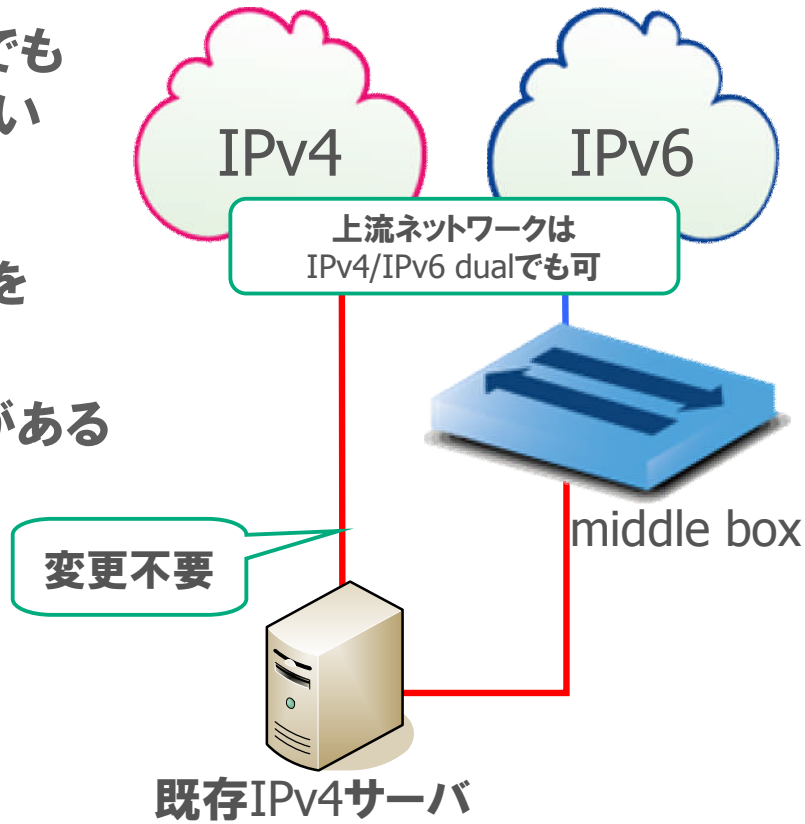
4-2-1.middle boxのトポロジ -トポロジA-

- メリット
  - IPv4/IPv6変換対応のmiddle boxを所有している場合、機材を追加する必要がない
- デメリット
  - middle boxがsingle point failureとなる
  - サービス監視ができない場合がある(middle box)に依存
  - サーバのメンテナンスが面倒になるケースが存在する(middle box に依存)



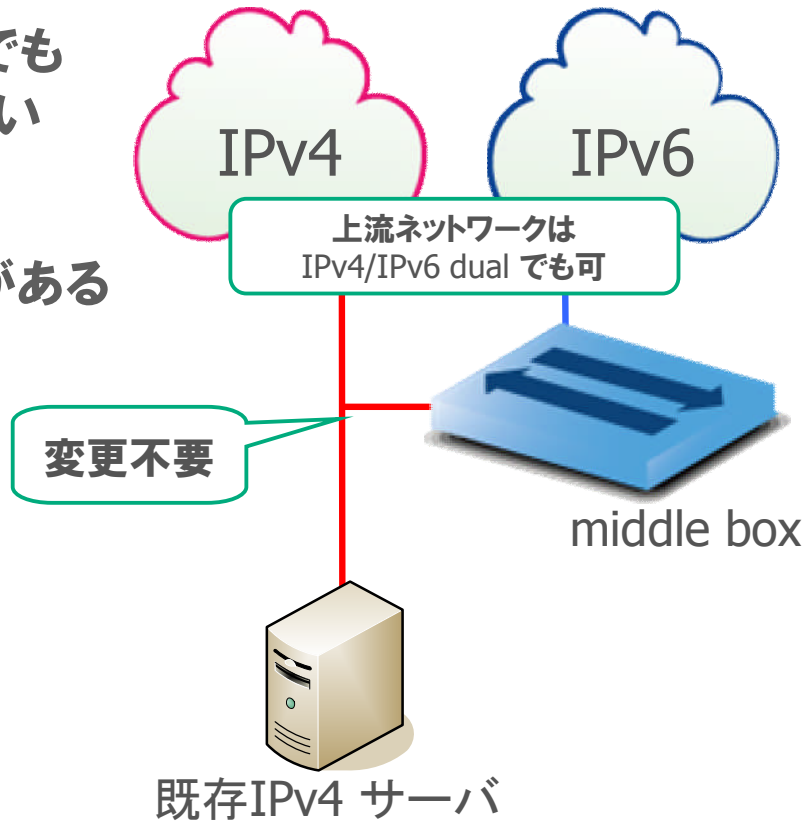
4-2-2.middle boxのトポロジ -トポロジB-

- メリット
  - 既存IPv4ネットワークへの変更が不要
  - middle boxに障害が発生した場合でも既存IPv4サービスには影響を与えない
- デメリット
  - 既存IPv4サーバにネットワークカードを追加する必要がある
  - 新規にmiddle boxを用意する必要がある



4-2-3.middle boxのトポロジ -トポロジC-

- メリット
  - 既存IPv4ネットワーク・サーバの変更が不要
  - middle boxに障害が発生した場合でも既存IPv4サービスには影響を与えない
- デメリット
  - 新規にmiddle boxを用意する必要がある



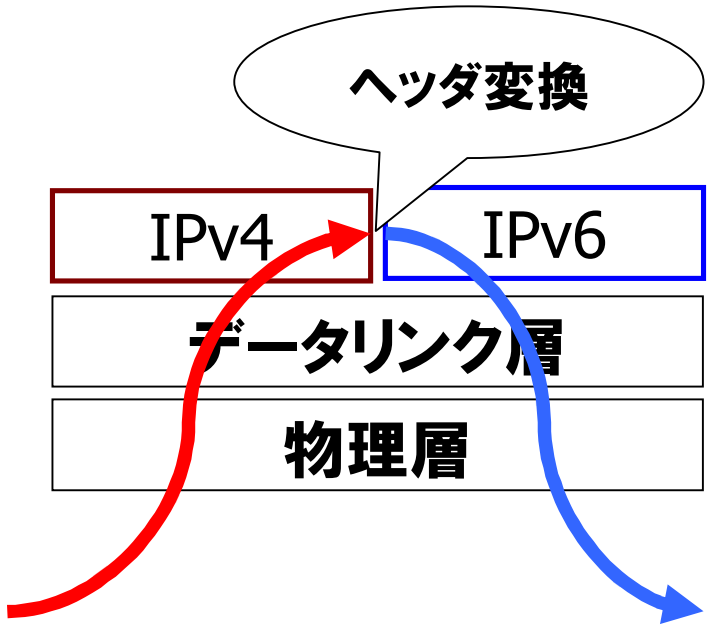
4-3 変換方式の解説

4-3.変換方式

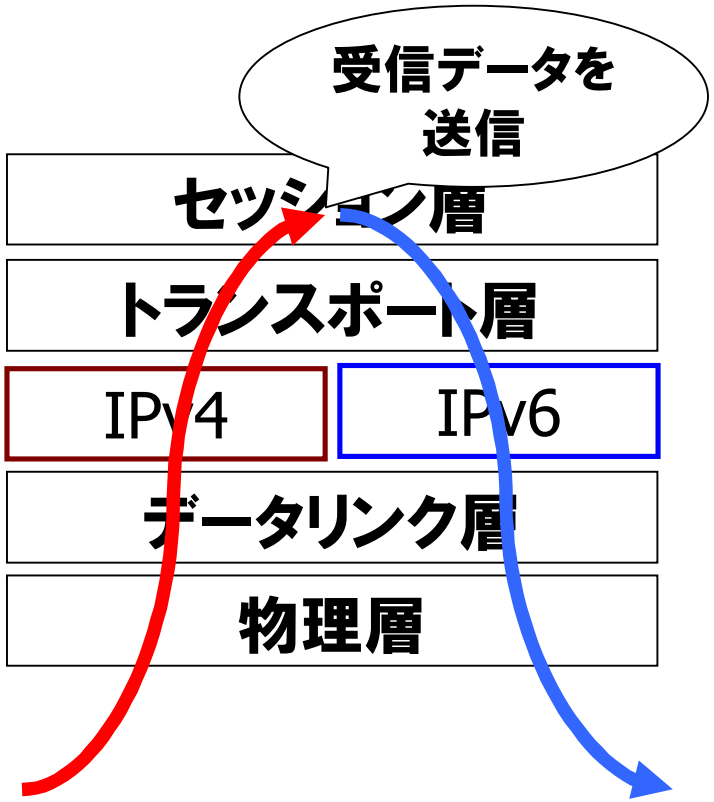
- NAT-PT
  - ネットワーク層での変換
    - IPヘッダの変換を実施
    - 様々なパケットの変換に対応
  - 変換負荷が少ないためパフォーマンスが出やすい
  - MTU処理が必要
- TRT/ALG
  - トランスポートリレー
    - IPヘッダの変換をしているわけではない
  - 変換負荷が高い
  - ICMPの変換ができない
  - 実装によっては、サービス毎に用意する必要がある

4-3-1.変換方式の図解

- NAT-PT



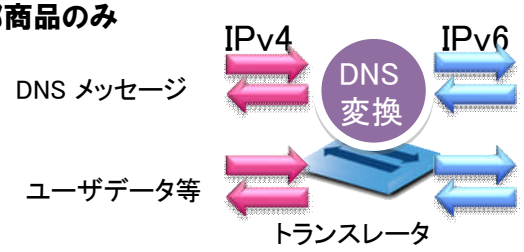
- TRT/ALG



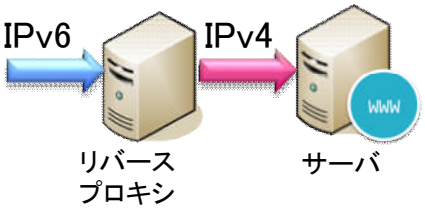
4-3-2.対応製品・アプリケーション

- トランスレータ
  - 多彩な変換方式をサポート
  - DNS変換も対応している(※)

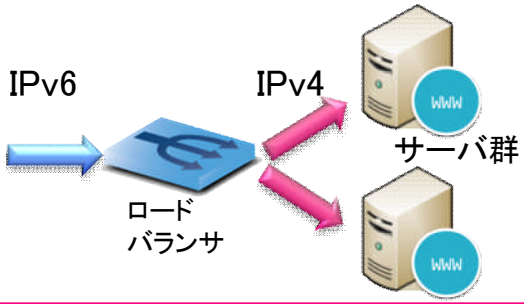
※ 一部商品のみ



- リバースプロキシ
  - IPv4サーバをIPv6へ公開
  - ALGを容易に実装可能



- ロードバランサ
  - 変換と同時に負荷分散も可能



- その他(一部機器が実装)
  - ルータ
  - ファイアウォール

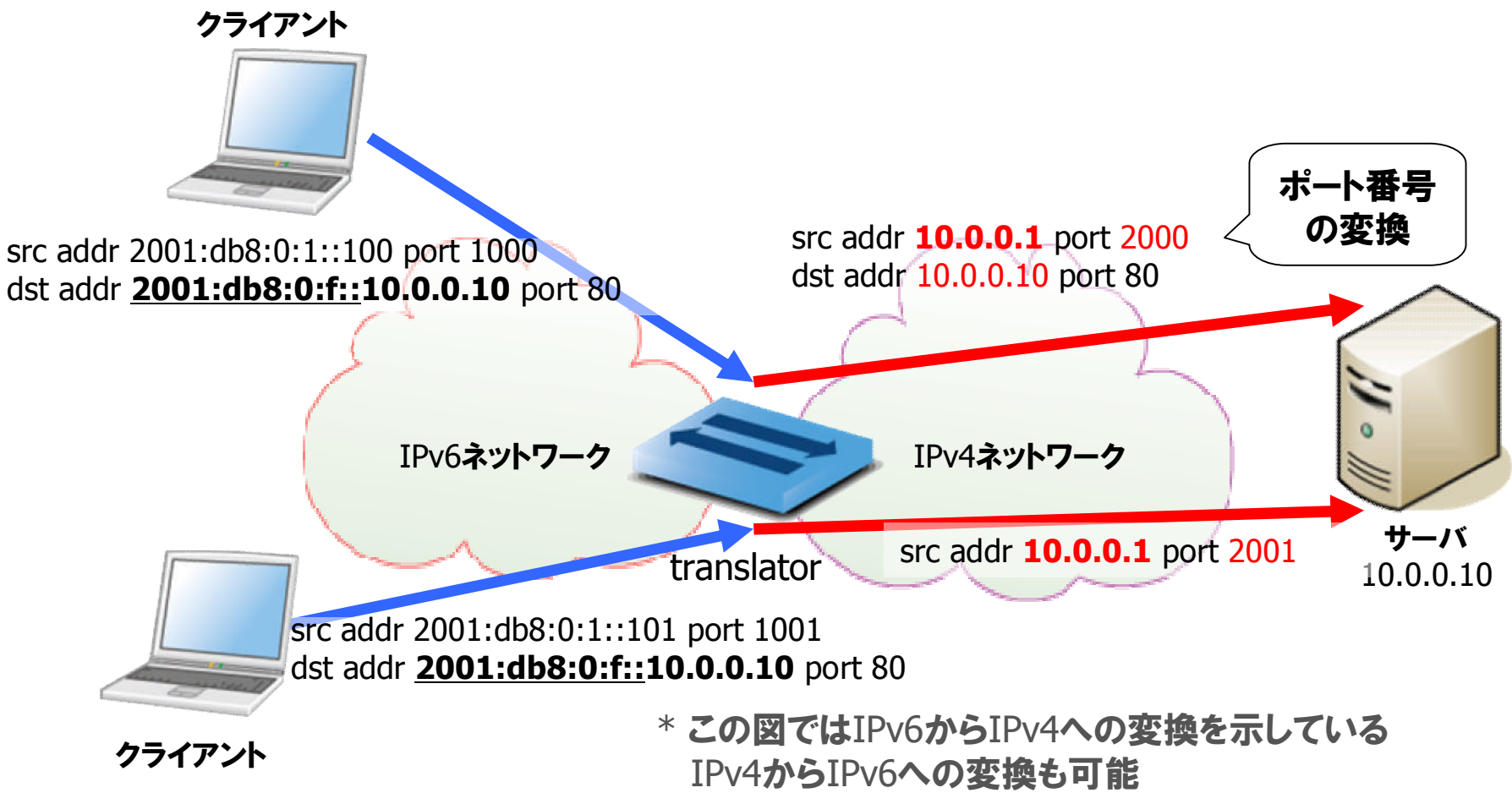
環境に合わせて最適なmiddle boxを選択することが重要

4-3-3.トランスレータの特徴

- メリット
  - 様々なパケットの変換に対応可能
    - ユニキャスト、マルチキャスト、ブロードキャスト
  - クライアントサイドへの設置も可能
  - 外部ネットワークからのサービス監視が可能
- デメリット
  - ペイロードにアドレスが含まれるパケットは変換できない場合がある
    - 一部のALGを搭載している製品も存在する

4-3-3.トランスレータでの変換

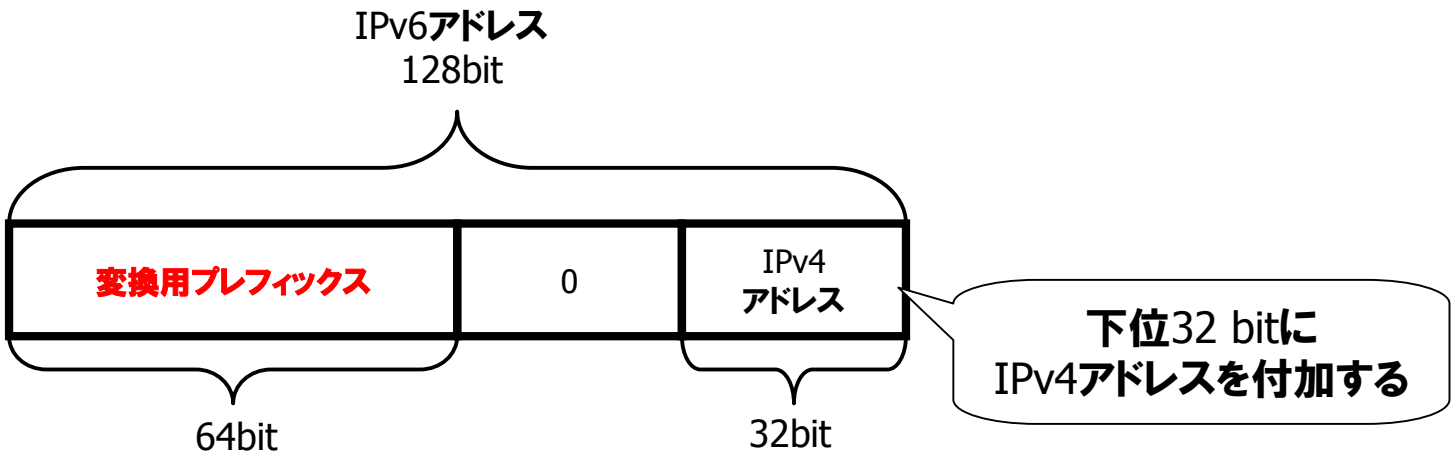
変換用プレフィックス宛のパケットを変換



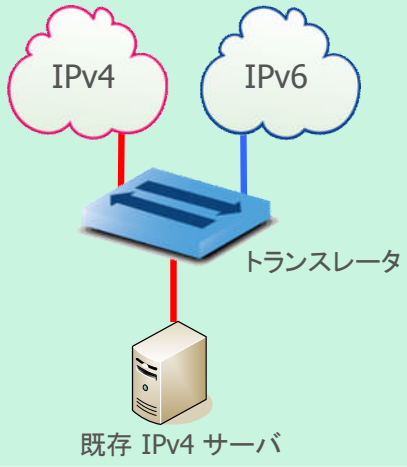
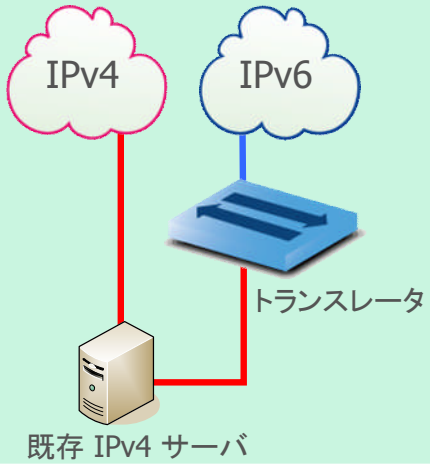
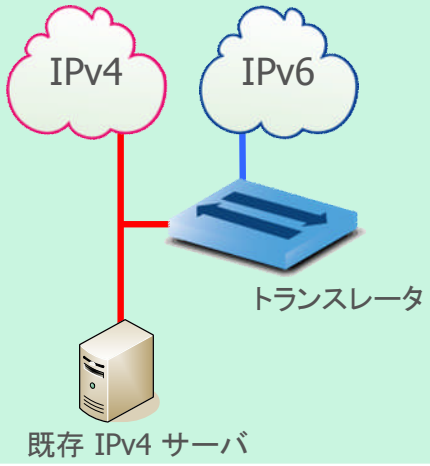
＊ この図ではIPv6からIPv4への変換を示している
IPv4からIPv6への変換も可能

4-3-3.変換用プレフィックス

- IPv6変換用プレフィックス (64ビット長)
  - トランスレータ1台に対して1プレフィックスをアサイン
  - IPv4アドレスをIPv6アドレスにマッピングするために使用



4-3-3.トポロジ適用可否

トポロジA	トポロジB	トポロジC
		
△	○	◎
IPv4 NAT機能を搭載している必要がある	**既存IPv4サーバにネットワークインターフェイスを追加する必要がある**	

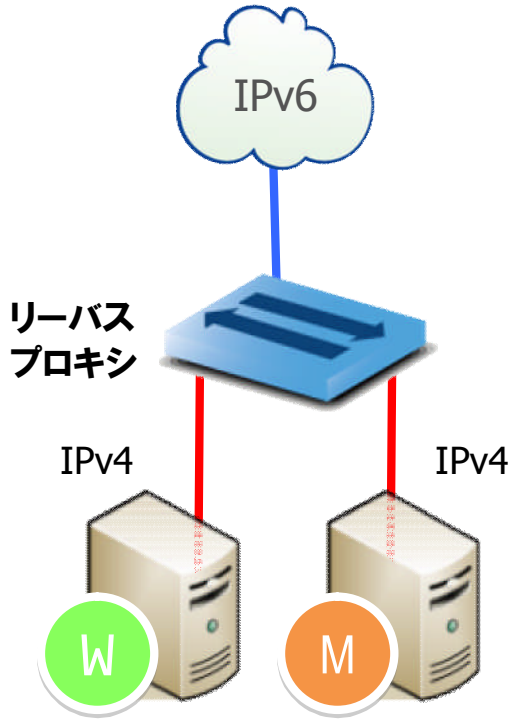
＊ ICMP**変換をサポートしていれば、**IPv6**ネットワークから直接サーバの監視も可能**

4-3-4.リバースプロキシの特徴

- メリット
  - ペイロード中の書き換えも可能
    (アプリケーション層のリバースプロキシの場合)
    - URLにIPv4アドレスが記述されている場合等
- デメリット
  - 性能と運用管理コストがかかる
    - 障害発生時の切り分けが必要
  - 利用するプロトコル毎にプロキシの設置が必要

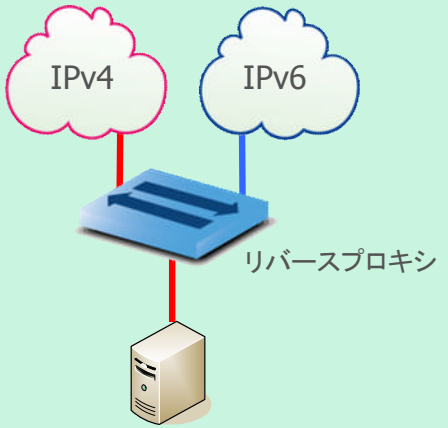
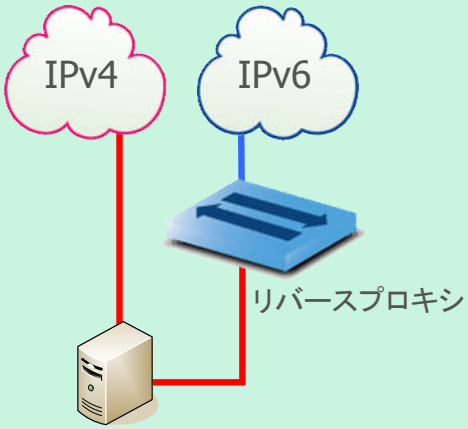
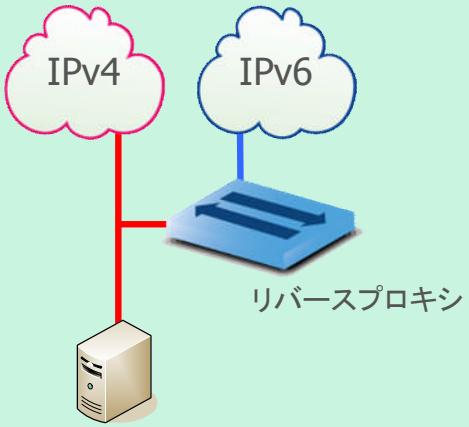
4-3-4.リバースプロキシによるアドレス変換

- IPv6クライアントからの接続を、IPv4に変換してサーバに中継する手法



- トランスポート層で通信を中継
  - TCP, UDPの通信を中継
- アプリケーション層で通信を中継
  - HTTP (リバースプロキシ)
  - SMTP(メールハブ)
- 既存のサーバはIPv4のまま

4-3-4.トポロジ適用可否

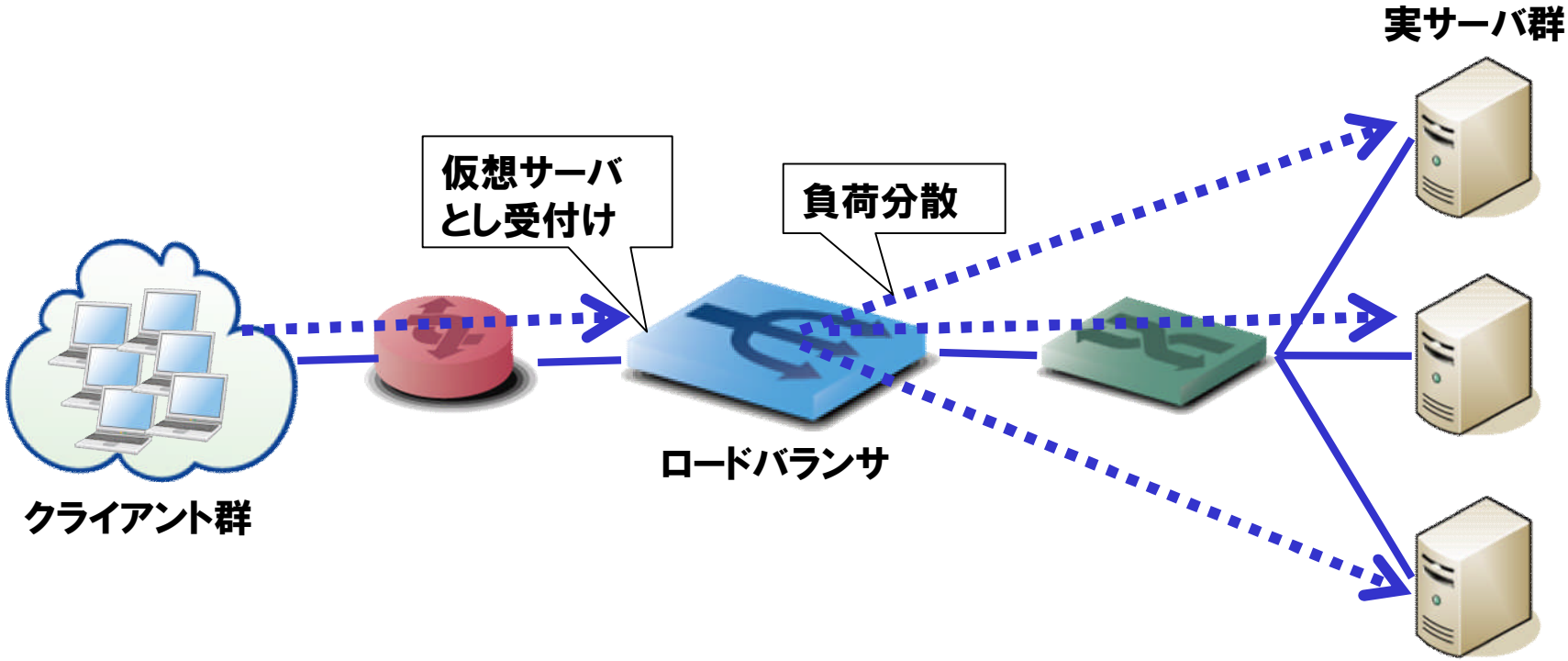
トポロジA	トポロジB	トポロジC
		
△	○	◎
IPv4のトラフィックもリバースプロキシを経由する必要がある	**既存IPv4サーバにネットワークインターフェイスを追加する必要がある**	

＊ **IPv6ネットワークからICMPを用いた疎通監視はできない**

4-3-5.ロードバランサの特徴 ①

ロードバランサの機能

- 多数のクライアントからのアクセスを仮想サーバとして受付け、複数の実サーバへ通信を転送 (負荷分散)させる



4-3-5.ロードバランサの特徴 ②

- ロードバランサの機能
  - サーバ負荷分散 (アドレス変換)
    - L2 MACアドレス
    - L3 IPバージョン(IPv6/IPv4変換) *
    - L3 IPアドレス
    - L4 TCP/UDPポート番号
    - L5～L7 URLなど **
  - セッション維持(同一セッションを同一サーバへ振り分け)
    - セッションの認識
  - 稼動監視
    - L3 ICMP監視
    - L4 ポート監視
    - L7 アプリケーション稼動監視 **

* IPv6/IPv4変換をサポートしている製品の場合
** L7(アプリケーション層)対応製品の場合

4-3-5.ロードバランサの特徴 ③

- メリット
  - 変換と同時にトラフィックの負荷分散が可能
  - ペイロード中の書き換えが可能 *
    - URLにIPv4アドレスが記述されている場合等
- デメリット
  - 運用管理コストがかかる
    - 障害発生時の切り分けが必要
  - 利用可能なプロトコルに対応したロードバランサが必要

* L7(アプリケーション層)対応製品の場合

4-3-5.ロードバランサのアドレス変換 ①

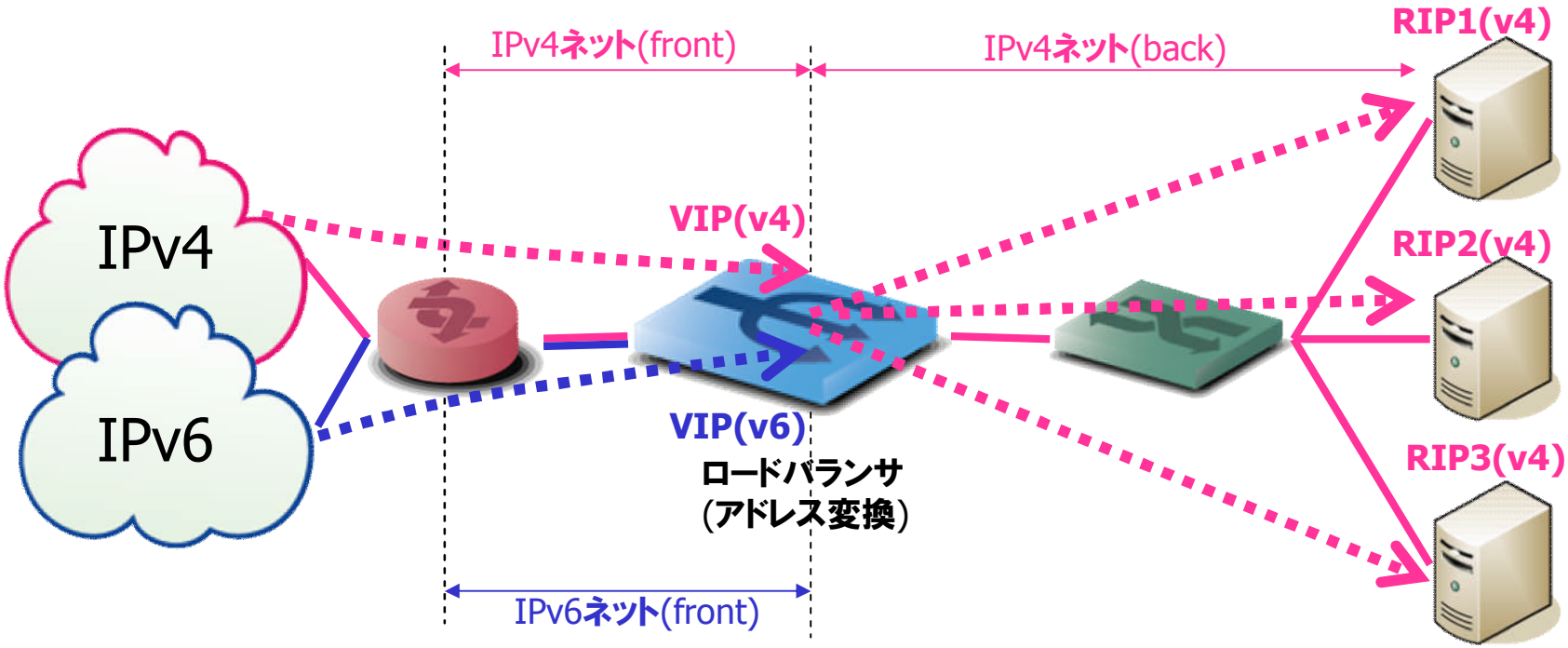
- 組み合わせ
  - IPv4クライアント→IPv4サーバ群
  - IPv6クライアント→IPv6サーバ群
  - IPv6クライアント→IPv4サーバ群 ＊
  - IPv4クライアント→IPv6サーバ群 ＊

- 製品により、負荷分散可能な組合せが異なるため注意が必要。
  - IPv6/IPv4変換を行う場合、対応製品を用意する必要がある

＊ IPv6/IPv4変換をサポートしている製品の場合

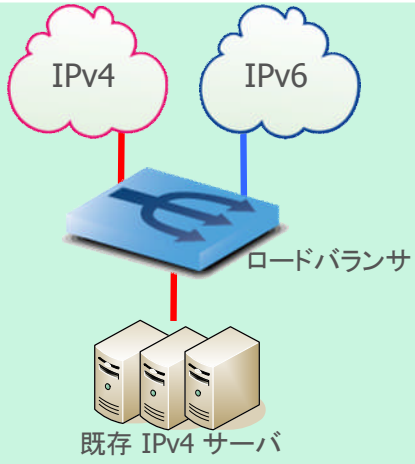
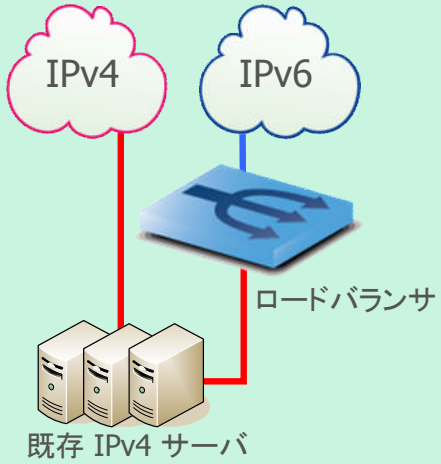
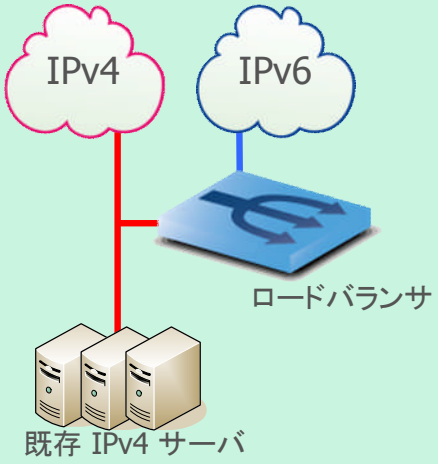
4-3-5.ロードバランサのアドレス変換 ②

- IPv6 Virtual IP (VIP) で受信したトラフィックを IPv4 実サーバのIP(RIP)へ送信



＊ この図ではIPv4をLayer3で終端(NAT)している
IPv4をNATしない設定も製品によっては可能

4-3-5.トポロジ適用可否

トポロジA	トポロジB	トポロジC
		
◎	○	○
IPv4/IPv6同時に負荷分散が可能	IPv4の負荷分散は別途装置が必要	IPv4の負荷分散は別途装置が必要 (既存IPv4ネットワーク・サーバの変更を行う場合はIPv4/IPv6同時に負荷分散も可能)

＊ IPv6ネットワークから直接疎通監視はできない
＊ ロードバランサが持つ監視機能を利用する必要がある

4-3-6.比較表

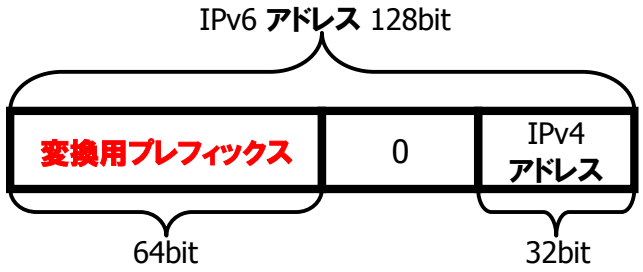
	トランスレータ	リバースプロキシ	ロードバランサー
メリット	様々なパケットの変換に対応可能 クライアントサイドへの設置も可能 外部ネットワークからのサービス監視が可能	ペイロード中の書き換えが容易に可能	変換と同時にトラフィックの負荷分散が可能 ペイロード中の書き換えが可能
デメリット	ペイロードにアドレスが含まれるパケットは変換できない場合がある	性能と運用管理コストがかかる 障害発生時の切り分けが必要 利用するプロトコル毎にプロキシの設置が必要	性能と運用管理コストがかかる 障害発生時の切り分けが必要 利用するプロトコル毎にロードバランサの設置が必要

4-4.middle boxとDNS

- ユーザはFQDNを指定してアクセスしてくるので middle boxの設置だけでは不十分
  - DNSにAレコードしか設定していない場合には、IPv4からのアクセスしかない
- ユーザからのDNS問い合わせに対してAAAAレコードを返答する必要がある
  - middle boxを通過するようなAAAAレコードを返答する
  - 設定方法
    - DNSサーバにAAAAレコードを追加する
    - DNS-ALGを設置する

4-4-1.DNSサーバにAAAAレコードを追加 ①

- middle boxによってマッピングされたIPv6アドレスを登録する
  - 登録するIPv6アドレスはmiddle boxのアドレスマッピングの仕組みによって異なる
- 例: サーバアドレス = 10.1.100.1
  - 1対1マッピングを使用する場合
    - 2001:db8:1000:2000::80をマッピング
    - AAAA IN 2001:db8:1000:2000::80
  - 変換用プレフィックスを使用する場合
    - 2001:db8:1000:ffff::/96を変換用プレフィックス
    - AAAA IN 2001:db8:1000:ffff::a01:6401

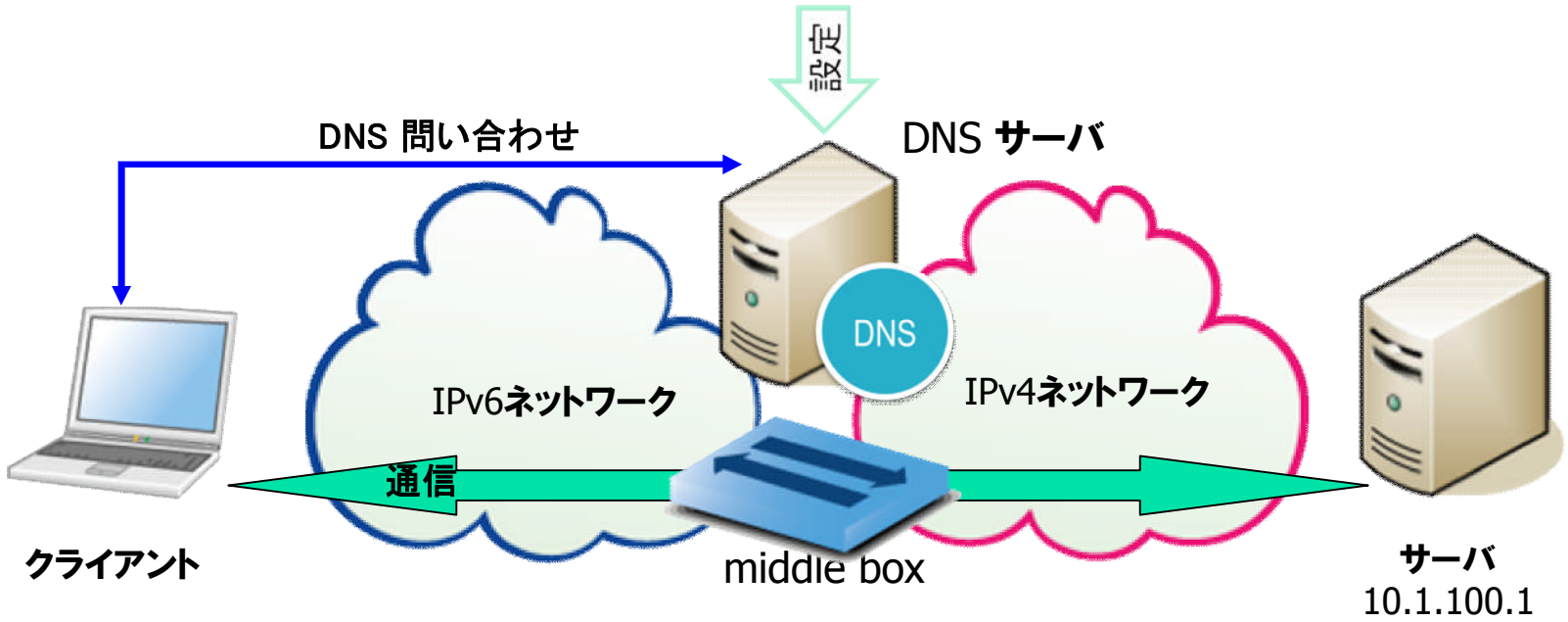


4-4-1.DNSサーバにAAAAレコードを追加 ②

- middle boxによってマッピングされたIPv6アドレスを登録

DNSレコード

サーバ	AAAA	2001:db8:1000:2000::80
サーバ	AAAA	2001:db8:1000:ffff::a01:6401

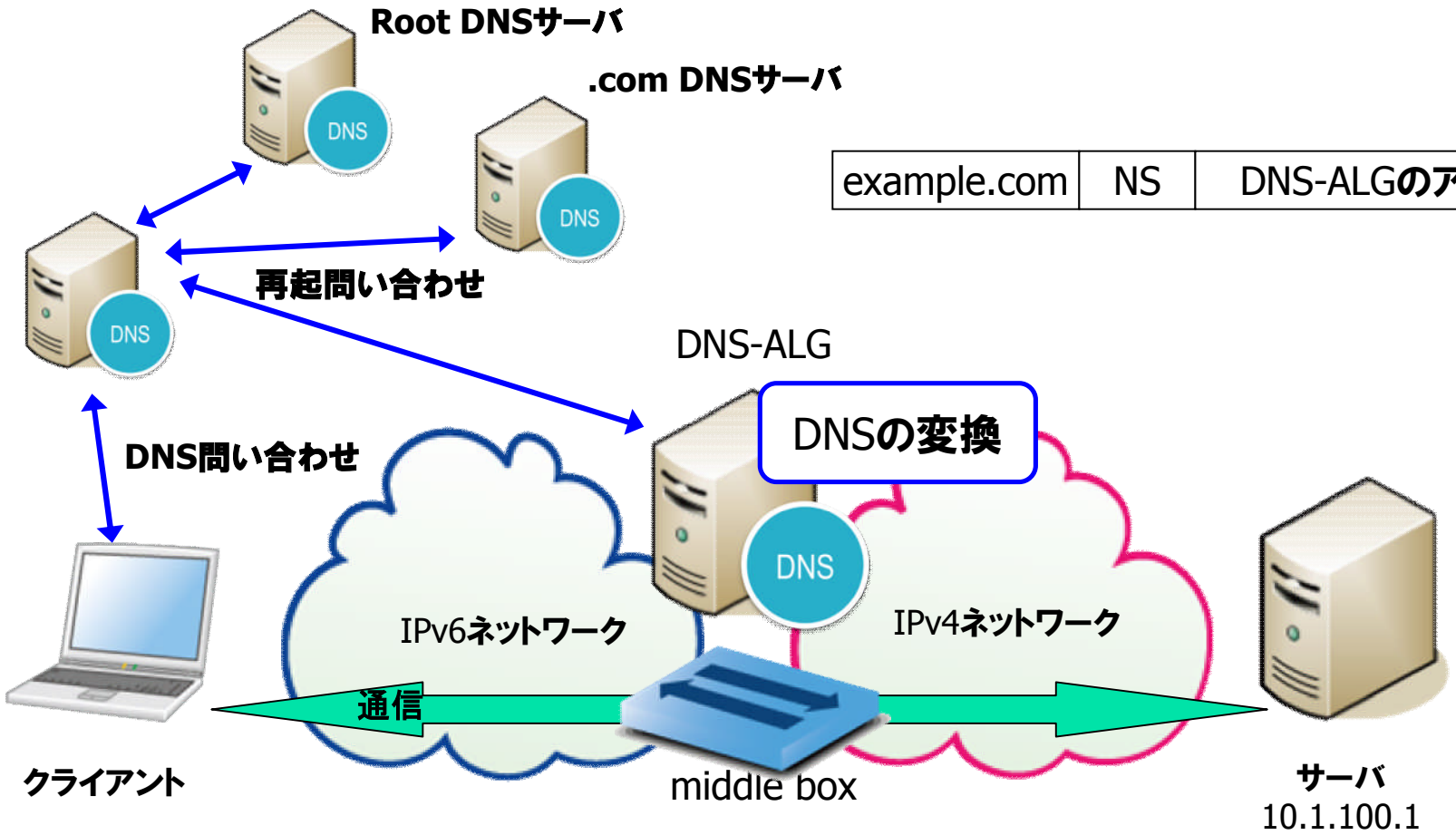


4-4-2.DNS-ALGを設置 ①

- DNS-ALGを設置し、既存AレコードをAAAAレコードに変換する
  - レコードの変換方法については、使用するmiddle boxのアドレスマッピングの仕組みに併せて設定しなければならない
    - マッピング方式の例は、前のページを参照
- 変換対象となるドメインのNSを設置するDNS-ALGにする必要がある
  - 受信したクエリを既存DNSサーバに転送し、レスポンスを変換する
  - 注意点: 全てのクエリをDNS-ALGが受けることになる
- トポロジへの配置については、middle boxの近くに置く必要はない

4-4-2.DNS-ALGを設置 ②

変換対象ドメインのNSレコードをDNS-ALGに設定



example.com	NS	DNS-ALGのアドレス

4-5.留意点 ①

- middle boxはトラフィック変換しか行わない
  - データ内にアドレス情報が含まれる場合は別途ALGが必要
    - データ内のアドレス情報を用いて別の通信を発生する場合は、そのアドレス情報も変換する必要がある
      - SIP
      - FTP
      - など
- 変換対象の制限を行う必要がある
  - 変換対象を限定しないと、middle boxを介してIPv4の世界へ到達してしまう
  - プロトコルは異なるものの、プロキシやNATと同義なので、踏み台にされないようにする

4-5.留意点 ②

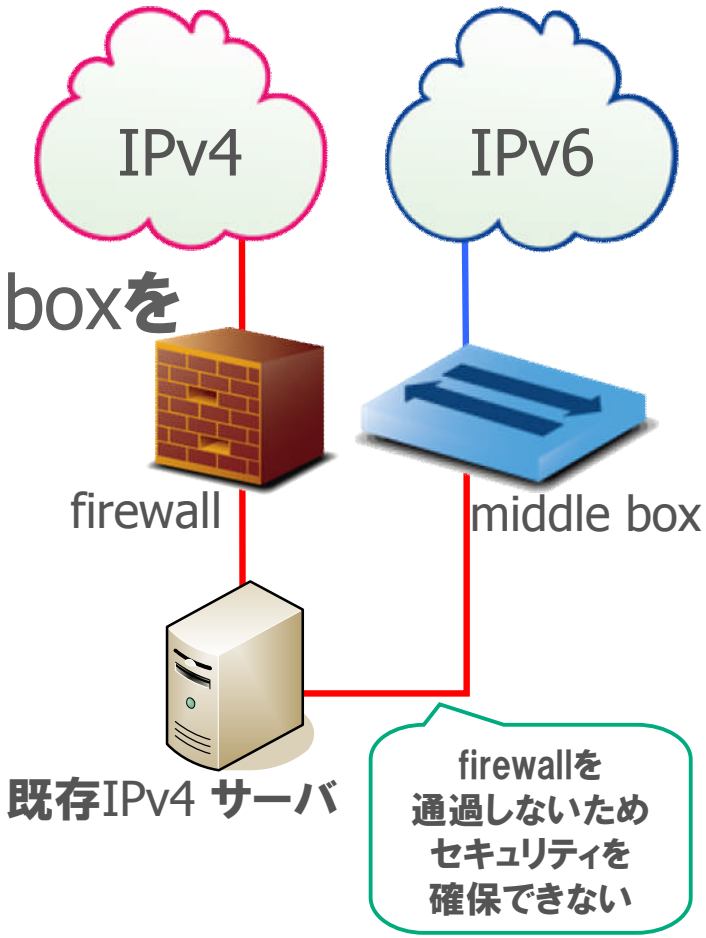
- 障害発生時の切り分けが複雑になる
  - middle boxを境に、IPv4とIPv6で別々に切り分けを行うことが重要
    - 例: middle box{から、まで}の疎通性の確認
  - middle boxの双方のインターフェイスでパケットキャプチャすると有効
- MTU
  - IPv6が1パケットで送信できるデータ量は、IPv4に比べて20バイト少ない
    - IPv6ヘッダがIPv4ヘッダより20バイト長いため
  - 1481バイト以上のIPv4パケットをIPv6へ変換する場合は、必ずフラグメントが発生する
    - middle boxの実装によっては、1261バイト以上のIPv4パケットでもフラグメントされる可能性がある

4-5.留意点 ③

- middle boxが出力するログ内容を事前に確認するとよい
  - middle boxを経由したトラフィックは、サーバから見たときに全て middle boxのアドレスからのアクセスとなる
  - ログ解析を行っている場合は、middle boxが出力する内容が十分であることを確認するとよい
- ログの出力や、snmp等の監視プロトコルがIPv6トランスポートに対応しているかを確認すべき
  - 製品によっては、IPv4のみ対応しているものも存在するため事前に確認しておくとよい

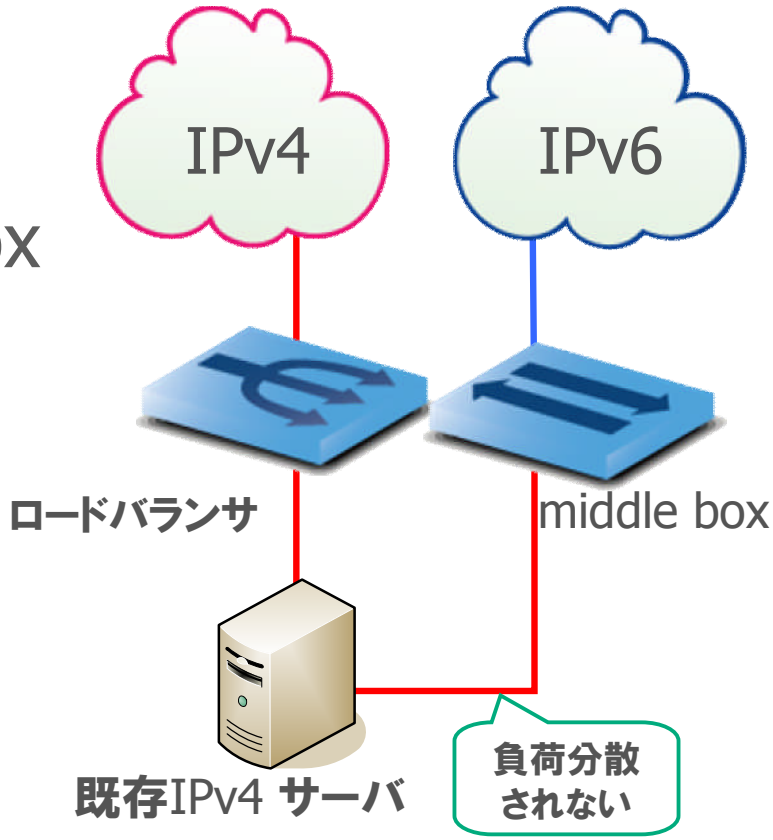
4-5.留意点 ④

- IPv4 firewall **と並列に**middle boxを**設置すると、セキュリティを保つことができない**
- **セキュリティを確保する場合は、**IPv4 firewall**の上流に**middle boxを**設置すべきである**



4-5.留意点 ⑤

- IPv4ロードバランサと併用する場合には、変換されたトラフィックを負荷分散すべきかどうかを考慮する
- 負荷分散する場合は、ロードバランサの上流にmiddle boxを設置すべきである



Chapter-5

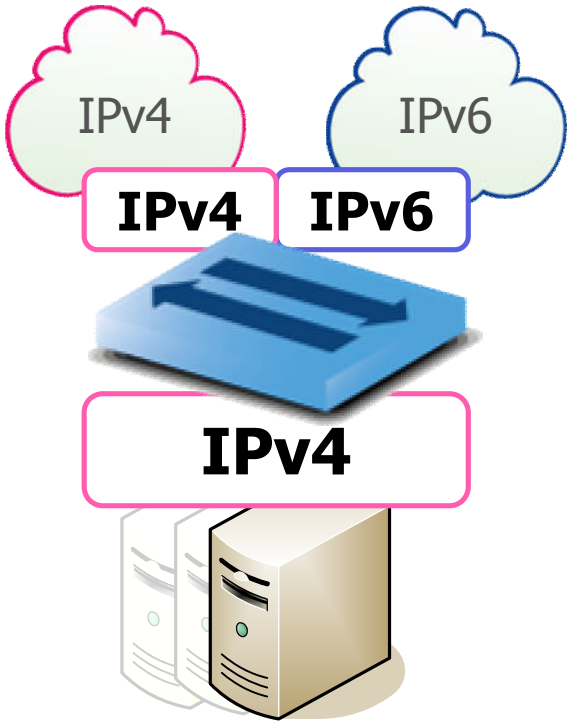
IPv6対応 ~サーバ編~

5-1.アドレスの設定方法

- **サーバのIPv6アドレスは静的に設定すべき**
- **自動構成による弊害**
  - EUI-64により下位64bitのインタフェースIDを自動生成
    - NICの交換時にMACアドレスが変更になるため、機器交換時にIPv6アドレスが変更されてしまう
    - OUI (Organizationally Unique Identifier)部分から機器のメーカーを知ることができるため、リモートからサーバのメーカーやOS名を推測される可能性がある
  - 匿名アドレス/一時アドレス (RFC 3041)
    - IPv6アドレスが一定時間で変更されるため、サーバとしての利用には向かない
  - **サーバでは、RAや匿名アドレスによるIPv6アドレス設定機能は無効化した方がよい**

5-2.IPv4サーバ + middle box

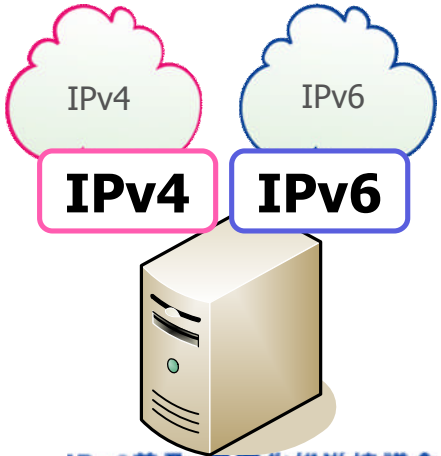
- middle boxを用いてIPv6アクセスをIPv4に変換
  - サーバ側はIPv4シングルスタックのまま



- メリット
  - サーバ自体をIPv4シングルスタックで運用できる
    - IPv6導入コストが低い
  - スケーラビリティを確保
    - サーバを容易に増やせる
- デメリット
  - サーバ以外の機器が必要に
    - 新たに導入する場合にはコスト増要因
    - ログ取得が複雑に(アクセス元のIPアドレスがmiddle boxになる場合も)
  - 今後、新たに登場するであろうプロトコルやアプリケーションの対応に制約

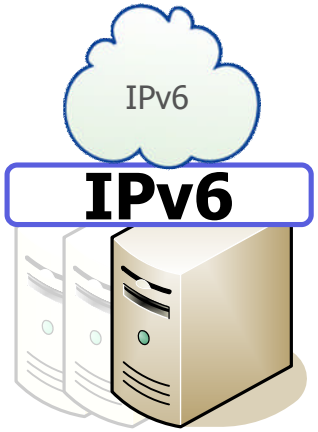
5-3.サーバのIPv4/IPv6デュアルスタック化

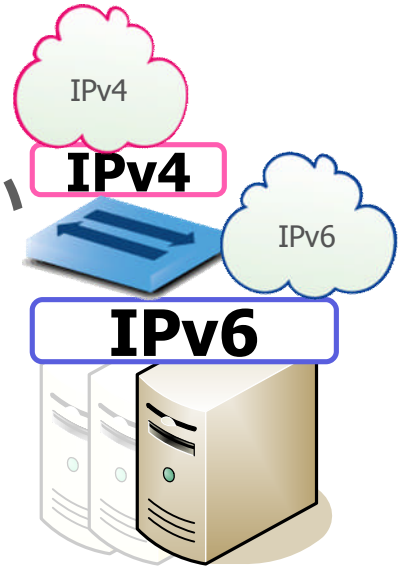
- サーバはIPv4/IPv6の両方でサービス提供
- メリット
  - End-to-End通信
    - 新規アプリケーションへの対応が容易
  - middle boxを導入する必要がないため、構成がシンプル
    - アクセス元のIPアドレスを確実に取得出来るため、アクセス制御をシンプルにできる
- デメリット
  - サーバアプリケーションの設計を慎重に行う必要がある
    - ソケットの実装方式、IPv4射影アドレスの取り扱いなどがOS、アプリケーションや設定により異なる
- サービス設計上の課題
  - IPv4/IPv6の分解点が設定できないため、サービスの全面的なデュアルスタック化が必要
  - IPv4に加え、IPv6に関する検証工数・監視項目が増加



5-4.IPv6シングルスタックサービス

- サーバはIPv6でのみサービス提供
- メリット
  - 設計・運用がシンプル
    - デュアルスタック運用に伴う問題を回避
      (障害切り分け、IPv4射影アドレスなど)
- デメリット
  - 新規開発が必要
  - 単独ではIPv4クライアントのサポートができない
    - IPv4クライアント向けにサービスを提供するには、
      IPv4用サーバの用意、middle boxの導入など、
      工夫が必要





5-5.各方式の比較

	IPv4シングルスタック+middle box	サーバのIPv4/IPv6デュアルスタック	サーバのIPv6シングルスタック
IPv4クライアント対応	○ (対応)	○ (対応)	× (未対応) 対応にはmiddle boxが必要
IPv6対応に係るソフトウェア改修	△ (少ない)	× (大きい)	× (大きい)
ネットワークのIPv6対応範囲	一部区間 (middle boxまで)	全区間 (サーバまで)	全区間 (サーバまで)
ソフトウェア実装の成熟度	○ (影響なし)	△ (未成熟)	△ (未成熟)
IPv4/IPv6共存に伴う問題 (IPv4射影アドレス、障害範囲の切り分けなど)	○ (影響なし)	× (影響あり)	○ (影響なし)
End-to-End通信 (新規アプリケーションへの対応,アクセス元IPアドレスの取得など)	×	○	○

5-6.サーバ自体をIPv6対応にする際の課題: 個々のサービス構成要素

- IPv6対応により影響を受ける要素の例
  - ネットワーク
    - ルータ、ファイヤウォール、ロードバランサ
  - ソフトウェア
    - OS、サーバソフトウェア、Webアプリケーション
  - 運用
    - サービス監視、パッチ適用
- どのIPv6対応方式を採るかにより、改修が必要な要素が大きく異なる
  - middle boxモデルを採ると、IPv6→IPv4変換により、IPv6対応が必要な要素を最小限に抑えることが出来る
  - DNSと監視についてはIPv4/IPv6デュアルスタック対応が必要
- IPv6対応の設計検討時には、各対応方式において影響を受ける範囲の分析が重要

5-7.サーバ向けOSのIPv6対応

- サーバ向けOS
  - 主要OS全てがIPv6プロトコルスタックを実装済み
    - Linux kernel 2.2.1以降
    - FreeBSD 4.0以降
    - Solaris 8以降
    - Windows Server 2003以降
- できる限り新しい実装を利用すべき
  - OS自体のバージョンアップとともに、IPv6プロトコルスタックの実装も改良されている
  - 古い実装では最新のIPv6関連仕様に追従していないケースも見受けられる
  - 例: IPv6 Type0ルーティングヘッダ処理の変更 (RFC 5095)

5-8.サーバアプリケーション側の実装

- 主要なサーバアプリケーションはほぼ全てIPv6に対応済み
  - 設定や動作はソフトウェアにより異なるため、十分な調査検討が必要
- ソケットの実装
  - IPv6用ソケットのみ(IPv4クライアントはIPv4射影アドレス利用)
  - IPv4用ソケット+IPv6用ソケット
  - IPv4射影アドレスの利用可否の決定
  - IPv4射影アドレスの実装がOSやアプリケーションにより異なる
- ソフトウェアとIPv6設定
  - 表記方法: アドレスやプレフィックスの記述方法に差異がある
  - デフォルト設定
    - 有効なソフトウェアの例: Apache
    - 無効なソフトウェアの例: sendmail, postfix, dovecot, net-snmp
- 課題: ソフトウェアの成熟度 - "枯れていない"コードの利用
  - IPv6プロトコルスタックやIPv6対応コード内のバグやセキュリティホールの影響により、メンテナンス回数増の可能性

5-9.障害発生時の切り分け

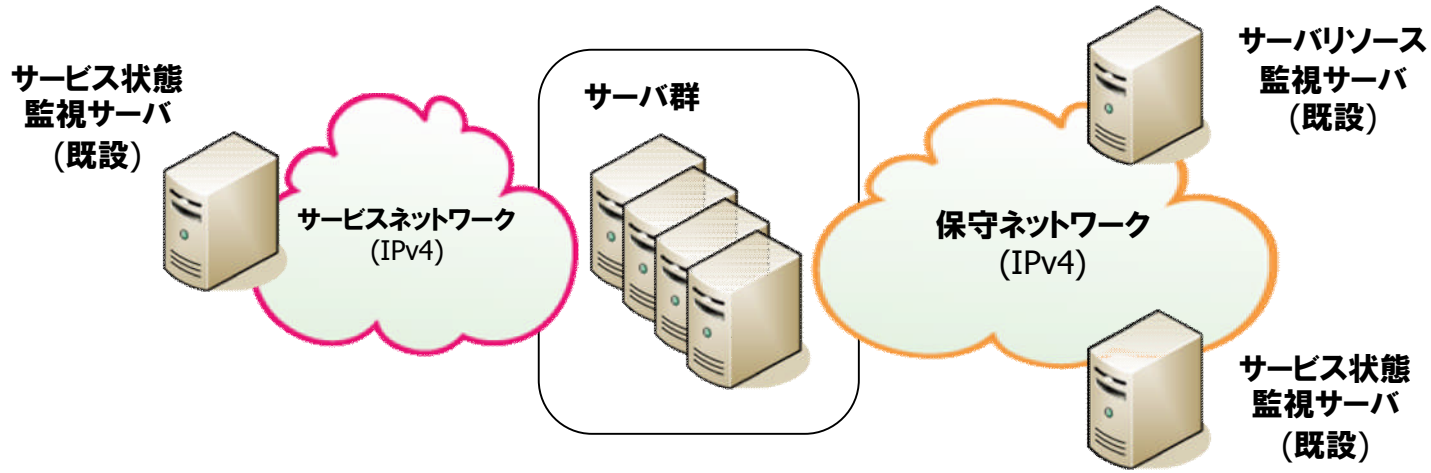
- 切り分けの手順
  - IPv4,IPv6のどちらで障害が起きているのかを切り分ける
    - IPv4サービスのみの障害
    - IPv6サービスのみの障害
    - デュアルスタック環境特有の障害
  - 障害が起きているレイヤの特定
    - ネットワーク層だけではなくアプリケーション層についても要調査
      - 例: WebサーバまでのIPv6到達性はあるものの、IPv6のソケットでは接続できない
    - middle box導入時には、直接とmiddle box経由の両方を監視
- 留意点
  - IPv6のサービスのみの障害が発生した場合には、フォールバックが行われるため、ホスト名ではなくアドレスを対象にする
  - 既存サーバをIPv6対応に変更した場合、IPv4アドレスの表記形式がIPv4射影アドレスに変更されるため、文字列処理に注意が必要

Chapter-6

IPv6対応 ~監視編~

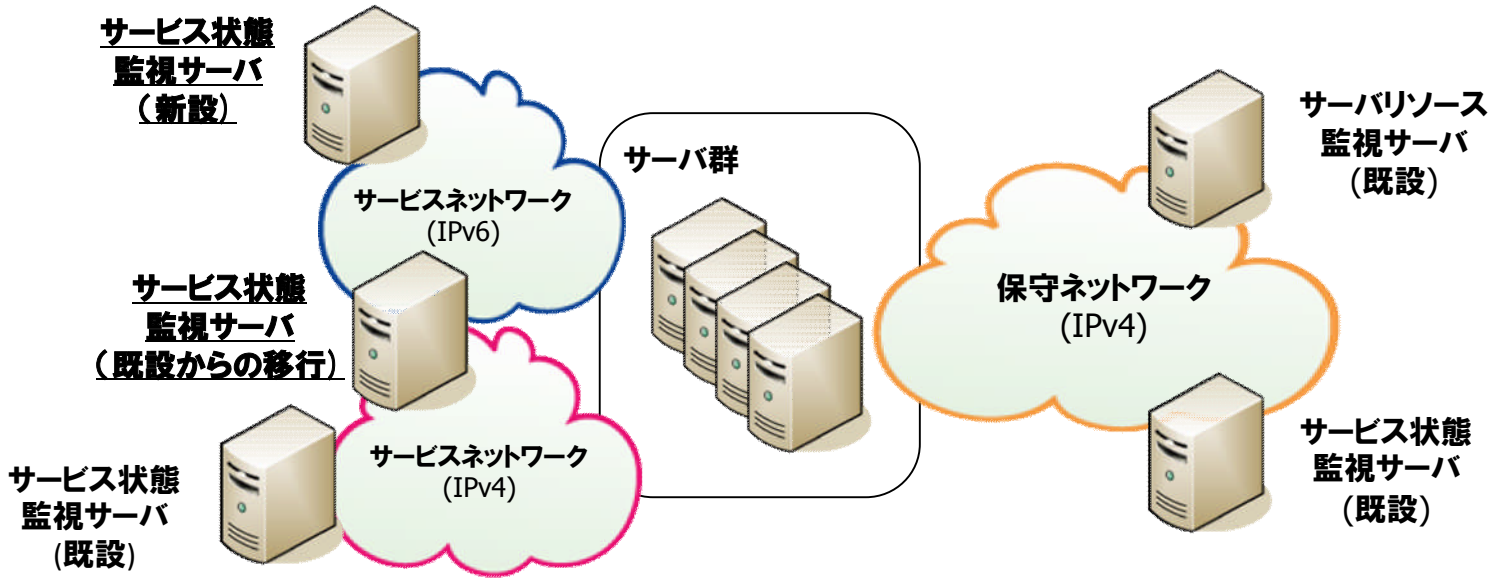
6-1.監視の現状

- 監視の種別
  - サーバリソース監視
    - CPU、メモリの使用状況、Webサーバプロセスの状態を監視する
    - 監視にはSNMPを利用する場合が多い
  - サービス状態監視
    - ユーザの利用環境に近いネットワークからサービスの状態について監視を行う
    - ICMPやHTTPによるWebコンテンツ取得で監視を行う場合が多い
- 監視サーバの設置場所
  - サーバリソース監視は、SNMPを利用するため、セキュリティ上の理由からIPv4保守ネットワーク上に置かれる場合が多い
  - サービス状態監視は、ユーザの利用環境に近いネットワークからの監視が必要なため、サービスネットワーク側に置かれる場合が多い



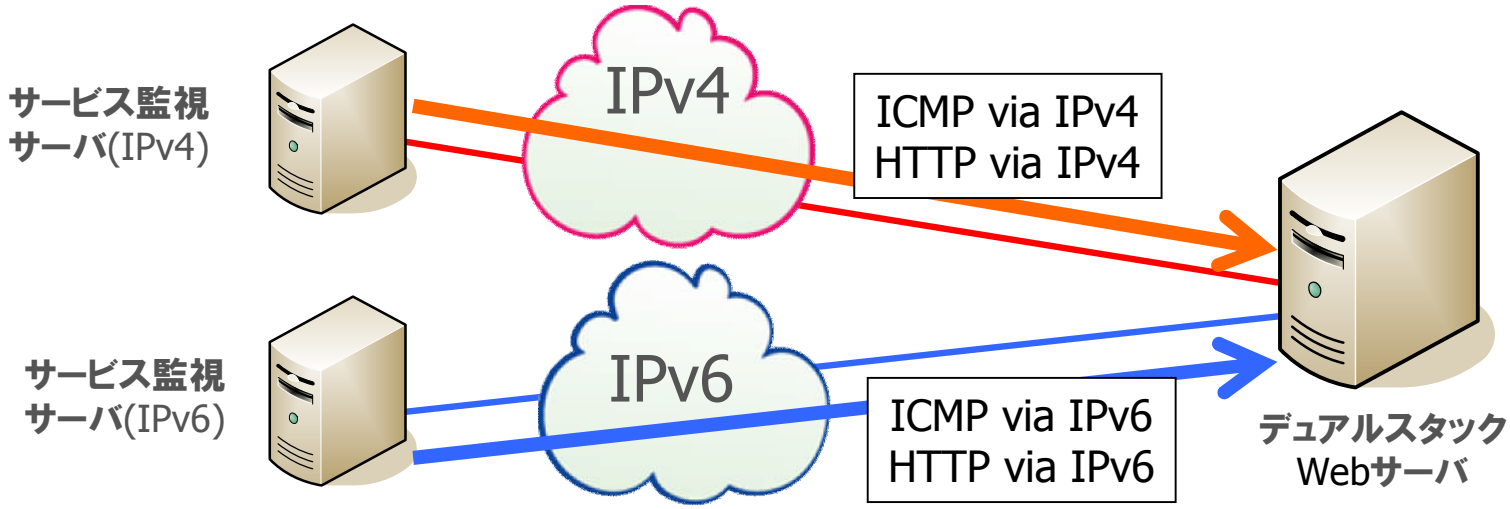
6-2.IPv6サービスにおける監視の例

- ホスティングサービスをIPv6対応にした場合、監視ポリシーは以下が考えられる
  - 全ての監視をIPv6に移行する必要はなく、既存のシステムでは監視できない部分のみをIPv6対応させる
  - サーバのリソース監視(CPUやメモリ、ディスクの状態)は、既存のリソース監視サーバを利用することで対応が可能
  - IPv6側からサービスの状態を監視する必要があることから、IPv6サービスネットワーク側にサービス状態監視サーバを設置する
  - この場合、IPv6側からの監視のみを行うサーバを新設する方法と既設のIPv4サービス状態監視サーバをデュアルスタック化する方法がある



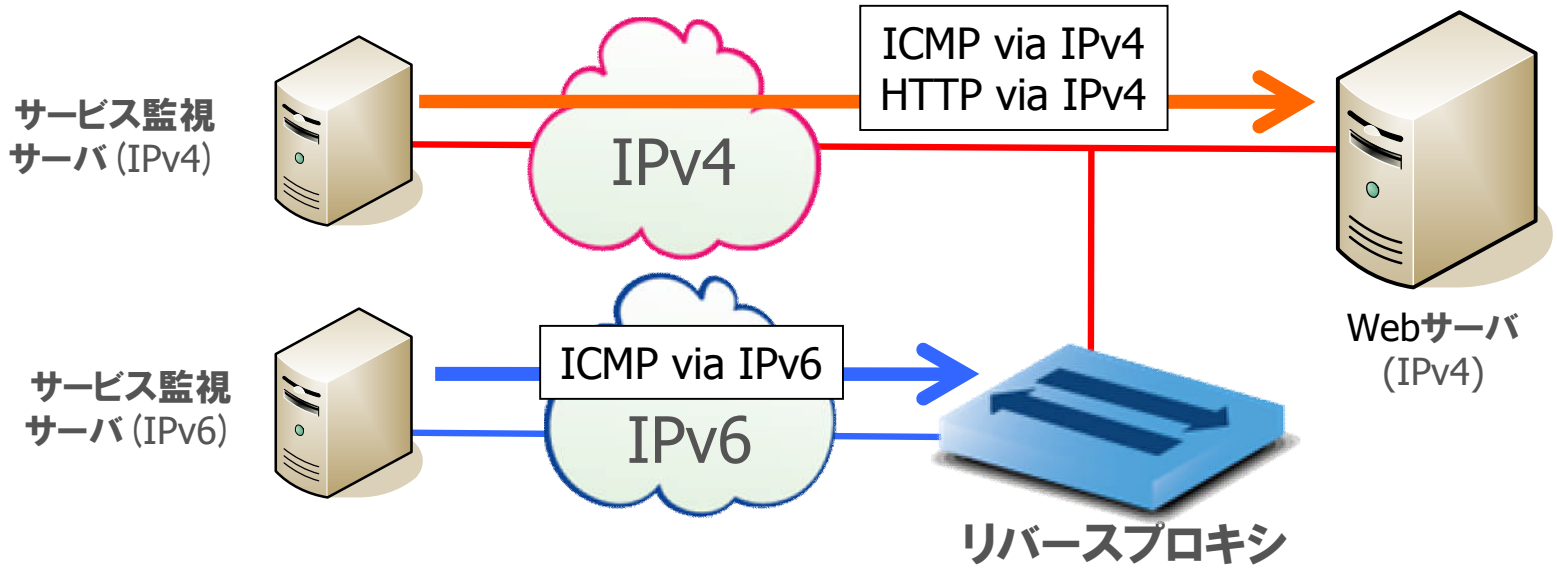
6-3.トポロジー別サービス監視方法 デュアルスタック

- 監視対象のデュアルスタックサーバに対してIPv4でのサービス監視と同等の監視が提供可能
- 図のような構成の場合、IPv4でのICMP監視やHTTP監視と同様にIPv6でのICMP監視やHTTP監視が可能
- この場合の監視対象はデュアルスタックサーバそのものになる



6-3.トポロジー別サービス監視方法
middle box

- デュアルスタックの場合と同様の対応が可能だが、middle boxの変換方式の実装によっては、IPv4と同様の監視ができない場合がある
- 下図の例でリバースプロキシを使用しているが、IPv6でのICMP監視はリバースプロキシに対してのみ可能



6-4.留意点 ①

- 監視対象のシステム構成によっては、IPv4経由で行っている監視項目をIPv6経由で同様に行うことができない場合がある。また、IPv6対応にすることでシステムが複雑化し、監視箇所の増加や監視方法が複雑化する可能性がある
  - トポロジー別サービス監視方法を参照
- この結果、故障箇所が分かりにくくなり故障回復までの時間が長くなる可能性がある

6-4.留意点 ②

- 想定されるミスとしては以下がある
  - デュアルスタック監視サーバにおいて、IPv4経由で監視を行っているつもりがIPv6経由での監視になっていた
  - IPv6経由で監視を行っているつもりが、監視コマンドがフォールバックしIPv4経由での監視になっていた
- このようなミスを防ぐため、監視設定(監視コマンドのオプションやアドレス指定等)でIPv4とIPv6のどちらを使って監視するのか明示的に指定しておくとよい

Chapter-7

今後の課題

今後の課題

- 各章での課題は各章末に『留意点』として掲載していますのでここでは再掲を控えます
- 本章ではIPv6対応における今後の課題や社会的な関わりについて記載します

引き続き検討すべき課題

- 運用
  - 運用者の技術力向上
    - マニュアルの整備
    - IPv6対応のトレーニングの充実（社外/社内）
    - 各種運用ツールのIPv6対応
  - 顧客サポート
    - ユーザー向けマニュアル類の改修
    - サポート要員の教育
    - サポート部門向けマニュアル・ツールのIPv6対応
  - 各種ログ管理
    - アクセスログ、middle boxでの変換ログの保管

引き続き検討すべき課題

- サービス
  - IPv6対応による差別化
  - ユーザーとのインターフェース(申請書類等)の改修
    - アドレス入力箇所のIPv6対応
  - ユーザー管理手法の見直し
    - ユーザー割当アドレスの管理
    - IPv4アドレスとIPv6アドレスの管理
    - 顧客管理データベースの改修
  - 他サービスのIPv4枯渇対応
    - ISPのIPv4枯渇対応(IPv6対応)状況
    - NTT-NGNのIPv6対応
    - エンドユーザーのIPv6利用方法
  - 他社IPv6サービスの動向

これからの課題～IPv6を運用というすること～

- 現在のインターネットは社会インフラの一部と言われるまでに発展し、それに伴い品質も急速によくなってきた
- また、その品質の高まりを受け、クラウンドコンピューティングや仮想化といった新技術の基盤として利用されてきている
- その社会インフラとまで成長したインターネットがIPv4枯渇という問題に直面していることは極めて憂慮すべき状況であり、何も対応を行わなければ社会的な問題になる可能性が高い
- 今後、IPv4枯渇という現実を受けIPv6インターネットが確実に広まっていくことが予想されるが、そこに求められる品質はIPv4インターネットが十数年の歳月を得て高めてきたレベルとなることは明白である
- インターネットの「サービス品質」は、そこに携わる全ての人々が協力して維持しているものであり、IPv6インターネットにおいても同様である
- インターネットに携わる全ての人々とはISP/xSPのようなサービス事業者のみにあらず、SIer、利用者含めIPv4によるインターネットを利用する全ての人々である
- この全ての人々が、IPv4枯渇が叫ばれている今この時点から、IPv6対応を真剣に考え、実際に利用し、経験することによって、運用のノウハウを蓄積することが重要である

Chapter-8

参考文献

参考文献 / 参考RFC について

- 本ガイドラインを纏めるにあたり参照した文書を以下に、参考としたRFC及びI-D.を次頁に記します
  - URL等は提供側の都合により変更となり閲覧不能となる場合もありますので予めご了承下さい

Chapter	文献名	著者名	URL
1	IPv4アドレス在庫枯渇問題に関する検討報告書	JPNIC	http://www.nic.ad.jp/ja/ip/ipv4pool/ipv4exh-report-071207.pdf
	インターネットの円滑なIPv6移行に関する調査研究報告書	総務省	http://www.soumu.go.jp/menu_news/s-news/2008/pdf/080617_2_bt1.pdf
2	IPv4アドレス枯渇対応 アクションプラン 2009.10版	IPv4アドレス枯渇対応タスクフォース	http://www.kokatsu.jp/blog/ipv4/event/ipv4kokatsujp-2010-actionplan_20091005.pdf
3	DNSのv6対応	(株)インターネット総合研究所 伊藤高一	http://www.janog.gr.jp/doc/v6ops/v6ops-f-dns-ito.pdf
	IPv6時代のDNS設定を考える	(株)クララオンライン 白畑真	http://www.janog.gr.jp/doc/v6ops/v6ops-f-dns-shin.pdf
	IPv4枯渇TFのInterop資料	(株)JPRS 高嶋 隆一	http://www.kokatsu.jp/blog/ipv4/data/interop2009/11_JPRS_TAKASHIMA.pdf

参考RFC / Internet Draft

- Chapter3
  - RFC 3901 DNS IPv6 Transport Operational Guideline
  - RFC 4472:Operational Considerations and Issues with IPv6 DNS
  - RFC 2671 Extension Mechanisms for DNS (EDNS0)
  - RFC 3226 DNSSEC and IPv6 A6 aware server/resolver message size requirements
- Chapter5
  - RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6
  - RFC 5095 Deprecation of IPv6 Type 0 Routing Headers in IPv6

Chapter-9

検討メンバー

本sub-WGは下記のメンバーで検討を進めました

Co-chiars
- 工藤 真吾 (ソフトバンクテレコム株式会社)
- 仲西 亮子 (三井情報株式会社)

検討メンバー
- 新 善文　(アラクサラネットワークス株式会社)
- 石川 武志 (富士通株式会社)
- 内山 巧　(株式会社電算)
- 遠藤 正仁 (横河電機株式会社)
- 大澤 貴行 (GMOホスティング&セキュリティ株式会社)
- 岡本 裕子 (NTTスマートコネクト株式会社)
- 上根 義昭 (ソネットエンタテインメント株式会社)
- 北口 善明 (株式会社インテック・ネットコア)
- 菅沼 真　(株式会社電算)
- 杉崎 透　(三井情報株式会社)
- 白畑 真　(株式会社クララオンライン)
- 高橋 弘哲 (日商エレクトロニクス株式会社)
- 竹内 友哉 (MKIネットワーク・ソリューションズ株式会社)
- 谷崎 文義 (西日本電信電話株式会社)
- 田村 友彦 (富士通株式会社)
- 土居 昭夫 (GMOホスティング&セキュリティ株式会社)
- 中川あきら(KDDI株式会社)
- 花山 寛　(ネットワンシステムズ株式会社)
- 本橋 篤　(富士通株式会社)
- 山田 英弘 (富士通株式会社)
- 脇谷 康宏 (株式会社インテック・ネットコア)
- 渡辺 信秀 (株式会社クララオンライン)

共存WG主査:
- 荒野 高志 (株式会社インテック・ネットコア)

共存WG副査:
- 津国 剛　(株式会社三菱総合研究所)

50音順/敬称略

Chapter-10

謝辞

- 本ガイドラインを執筆するにあたり、検討段階からIPv6協議会IPv4/IPv6共存WGのメンバ及びサービス移行sub-WGのメンバの協力を頂きました。多忙の中、度重なるミーティングへの参加、検証実験の実施、執筆に時間を割いて頂いた事に感謝致します。
- 本ガイドライン執筆にあたっては前項で紹介したsub-WGメンバー以外にも多くの方にご協力頂いています。
  - 東京大学 教授 江崎博士には本sub-WGの検討段階から多くのご支援を賜りました。
  - 慶応大学 教授 加藤博士には多忙の中、検証実験にお立ち会い頂き多くのコメントを頂きました。

- 本ガイドライン執筆にあたり事前検証として行いました実験の際には下記の企業に多大なるご協力を頂きましたのでこの場を借りて御礼申し上げます
  - アラクサラネットワークス株式会社
  - 富士通株式会社
  - 横河電機株式会社
  - GMOホスティング&セキュリティ株式会社
  - ソフトバンクテレコム株式会社
  - 株式会社電算
  - 三井情報株式会社
  - 日商エレクトロニクス株式会社
  - ネットワンシステムズ株式会社
  - 株式会社クララオンライン
- この他、多くの方に本ガイドライン執筆にご協力を頂きました。ここで感謝の意を表したいと思います。ありがとうございました。

Appendix

Appendix 目次

- Appendix A: 端末OSのIPv6対応状況
- Appendix B: トランスレータの標準化動向
- Appendix C: リバースプロキシの設定例とログ例
- Appendix D: ロードバランサーの構成例
- Appendix E: Nagiosでの設定例
- Appendix F: 用語集
- Appendix G: Appendix内における参考文献

Appendix A

端末OSのIPv6対応状況

A-1.対象機能と対象端末OS

- 対象機能
  - DNSリゾルバ
  - アドレス選択機構
  - ルータ優先度拡張
  - 自動トンネリング（6to4, Teredo）
  - IPv4射影アドレス
  - セキュリティ関連（RH0, onlink assumptionなど）
  - その他（SEND, MIPv6など）
- 対象端末OS
  - Windows XP, Windows Vista, Windows 7 RC
  - Mac OS X, Linux(kernel-2.6), FreeBSDなど

A-2-1.DNSリゾルバ

- 問い合わせリソースレコード(RR)の順序
  - 通信開始時にIPv4とIPv6双方の名前解決を実施
  - 問い合わせ順序は実装により異なる
    - Aクエリを優先する実装はAAAAクエリ時にAクエリの結果を利用するものが多い
- トランスポートの対応
  - 名前解決時に利用するプロトコルのIPv6対応
  - IPv4とIPv6のDNSサーバが登録されている場合の問い合わせ順序は実装により異なる

A-2-2.OS毎のDNSリゾルバ実装

OS	検索順序	DNSサーバの優先順序
Windows XP	AAAAクエリを優先	IPv6非対応
Windows Vista	Aクエリを優先	IPv6 DNSを優先
Windows 7 RC	Aクエリを優先	IPv6 DNSを優先
Mac OS X	Aクエリを優先	IPv6 DNSを優先
Linux	AAAAクエリを優先	/etc/resolv.confに従う
FreeBSD	Aクエリを優先	/etc/resolv.confに従う

A-3-1.アドレス選択機構(RFC 3484)

- ポリシーテーブル
  - アドレス選択時に利用するラベルや優先度を定義
    - 優先度:終点アドレス選択時に利用され高い値ほど優先
    - ラベル:始点/終点アドレス選択時に利用され一致するものを優先
  - デフォルトの設定

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

※IPv4-mappedアドレスがIPv4アドレスを表す

  - 仕様の変化で現在新しい仕様がIETFにおいて議論中
    - TeredoアドレスやULAの追加など

A-3-2.ポリシーテーブルの実装(1)

- Windows Vistaのデフォルト設定
  - netsh interface ipv6 show prefixpoliciesで確認可能

```
C:>netsh interface ipv6 show prefixpolicies
アクティブ状態を照会しています...

優先順位    ラベル   プレフィックス
----------  -----  --------------------------------
        50      0  ::1/128
        40      1  ::/0
        30      2  2002::/16
        20      3  ::/96
        10      4  ::ffff:0:0/96
         5      5  2001::/32


C:¥Windows¥system32>
```

  - RFC 3484における標準設定＋Teredoアドレス

A-3-3.ポリシーテーブルの実装(2)

- Linux(kernel 2.6.25以上)
  - ip addrlabel showで確認可能(iproute2-2.6.25以上)

```
$ ip addrlabel show
prefix ::1/128 label 0
prefix ::/96 label 3
prefix ::ffff:0.0.0.0/96 label 4
prefix 2001::/32 label 6
prefix 2002::/16 label 2
prefix fc00::/7 label 5
prefix ::/0 label 1
```

ULA

  - LinuxではULAのラベルも独自追加されている
  - 優先度は/etc/gai.confで設定

```
$ cat /etc/gai.conf
...
#precedence ::1/128 50
#precedence ::/0 40
#precedence 2002::/16 30
#precedence ::/96 20
#precedence ::ffff:0:0/96 10
...
```

※デフォルトは/usr/share/doc/glibc-common-2.*/gai.conf

A-3-4.ポリシーテーブルの実装(3)

- FreeBSD
  - ip6addrctl showで確認可能

```
% ip6addrctl show
Prefix                                Prec Label      Use
::1/128                                 50     0        0
::/0                                    40     1       10
2002::/16                               30     2        0
::/96                                   20     3        0
::ffff:0.0.0.0/96                       10     4        0
```

※6.2 RELEASE, 7.0 RELEASE までは /etc/rc.conf に ip6addrctl_enable="YES“ の設定が必要

  - RFC 3484の標準設定のみ

A-3-5.OS毎のアドレス選択機構実装

OS	RFC 3484の実装	独自実装
Windows XP	対応済	Teredoアドレスを追加
Windows Vista	対応済	Teredoアドレスを追加
Windows 7 RC	対応済	Teredoアドレスを追加
Mac OS X	未対応	※再長一致の動作
Linux	対応済	TeredoアドレスとULAを追加
FreeBSD	対応済	特になし

A-4-1.ルータ優先度拡張(RFC 4191)

- ルータ優先度
  - 複数のルータ広告が存在していた場合に利用するルータの優先度を指定できる
    - low, medium, highの三段階
  - 未対応ルータ広告はmediumとなる
- 特定経路(More Specific Route)の受信
  - ルータ広告において経路設定を実現する

A-4-2.OS毎のルータ優先度拡張実装

OS	ルータ優先度	同優先度の動作	特定経路の受信
Windows XP	対応	後のRAを優先	SP3から対応
Windows Vista	対応	先のRAを優先	対応
Windows 7 RC	対応	先のRAを優先	対応
Mac OS X	未対応	先のRAを優先	未対応
Linux	2.6.17から対応	先のRAを優先	2.6.17から対応
FreeBSD	6.1Rから対応	先のRAを優先	未対応

A-5-1.自動トンネリング

- 6to4（RFC 3056）
  - トンネル接続とIPv6アドレス割り当てを同時に実現
  - IPv4グローバルアドレスを利用したIPv6アドレス

6to4アドレス

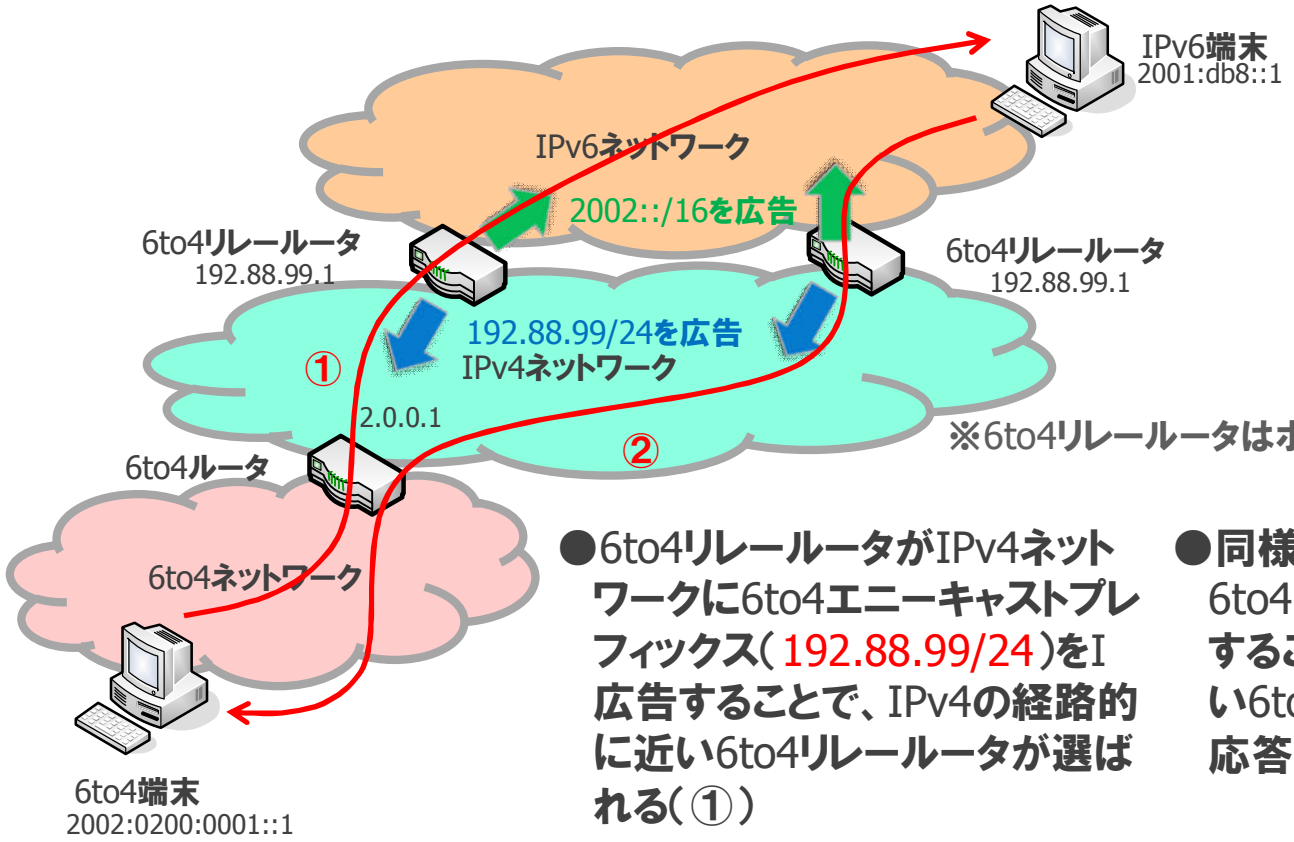
6to4 TLA 2002	6to4端末のIPv4アドレス	サブネットID	インターフェイスID
16ビット	32ビット	16ビット	64ビット

- Teredo（RFC 4380）
  - NATトラバーサルをIPv6で実現する技術
  - NATの内側からIPv6トンネル接続が可能

Teredoアドレス

Teredoプレフィックス 2001:0000	TeredoサーバのIPv4アドレス	フラグ	隠蔽したポート番号	隠蔽したNAT IPv4アドレス
32ビット	32ビット	16ビット	16ビット	32ビット

A-5-2.6to4端末⇔IPv6端末通信の挙動



※6to4リレールータはボランティア運用が一般的

●6to4リレールータがIPv4ネットワークに6to4エニーキャストプレフィックス(192.88.99/24)をI広告することで、IPv4の経路的に近い6to4リレールータが選ばれる(①)

●同様にIPv6ネットワークに6to4空間(2002::/16)を広告することでIPv6の経路的に近い6to4リレールータが選ばれ応答が帰る(②)

往復の経路は基本的に異なるためトラブルシュートが困難

A-5-3.6to4の実装と挙動

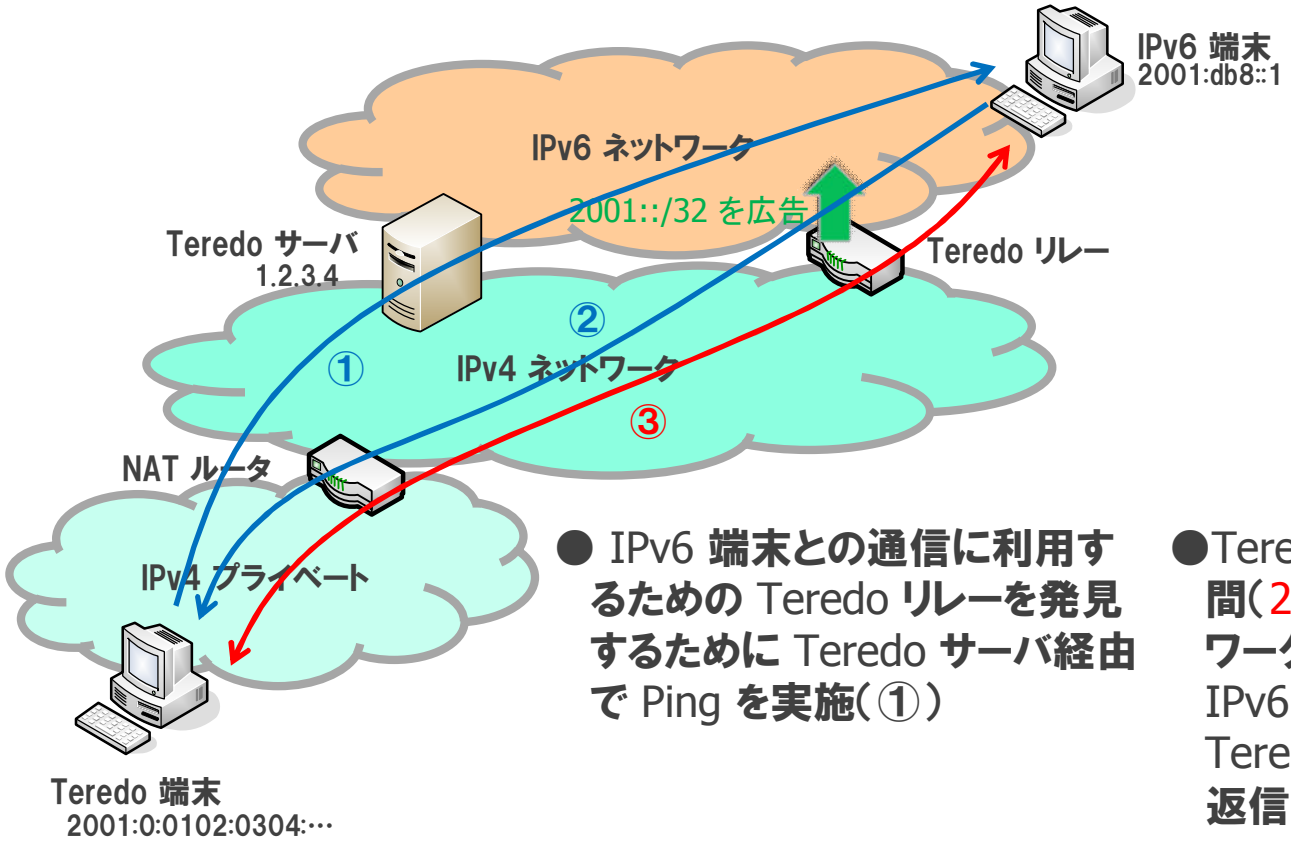
- Windows Vista では
  - IPv4 グローバルアドレスが設定されると 6to4 トンネルインターフェイスもデフォルトで設定

```
C:>ipconfig
...
Tunnel adapter ローカル エリア接続* 20:

   接続固有の DNS サフィックス . . . :
   IPv6 アドレス . . . . . . . . . . . : 2002:dxxx:xxx2::dxxx:xxx2
   デフォルト ゲートウェイ . . . . . : 2002:c058:6301::c058:6301
...
```

※6to4 リレールータとして 6to4.ipv6.microsoft.com（192.99.88.1）が設定されている

- 利用されるのは IPv6 オンリーサーバのみ
  - ポリシーテーブルのラベルが IPv6 アドレスと異なるため
  - デュアルスタックサーバに対しては
    - ラベルが一致する IPv4 アドレスが選ばれ 6to4 は使われない

A-5-4.Teredo端末⇔IPv6端末通信の挙動



● IPv6 端末との通信に利用するための Teredo リレーを発見するために Teredo サーバ経由で Ping を実施(①)

●Teredo リレーは Teredo 空間(2001::/32)を IPv6 ネットワークに広告しているため IPv6 の経路的に近い Teredo リレー経由で Ping が返信(②)

IPv6 端末に近い Teredo リレーを経由して通信(③)

A-5-5.Teredoの実装と挙動

- Windows Vista では
  - NAT ルータ配下の IPv4 プライベートアドレスが設定されると Teredo トンネルインターフェイスをデフォルトで設定
    （Windows Firewall が有効である場合）

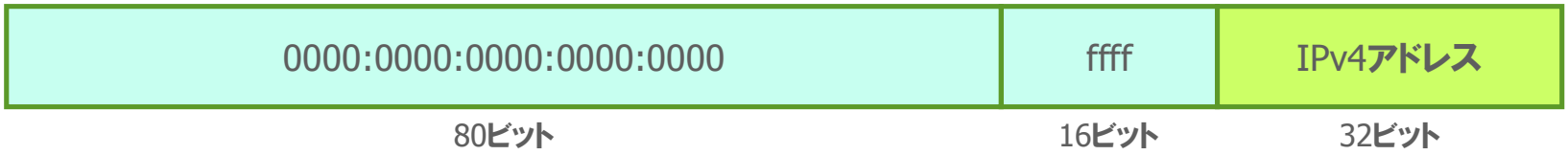
```
C:>ipconfig
...
Tunnel adapter ローカル エリア接続* 6:

   接続固有の DNS サフィックス . . . :
   IPv6 アドレス . . . . . . . . . . . : 2001:0:cf2e:3096:247d:XXXX:XXXX:YYYY
   リンクローカル IPv6 アドレス. . . . : fe80::247d:XXXX:86fd:XXXX%10
   デフォルト ゲートウェイ . . . . . :
```

※Teredo サーバとして teredo.ipv6.microsoft.com が設定されている

- Teredo アドレスのみでは IPv6 は利用されない
  - AAAA クエリを実施しないので実質利用されない
    - IPv6 アドレスを指定すると通信を行う
  - 中から一度発信しないと外から受け付けない

A-6-1.IPv4射影アドレス

- IPv4射影アドレス（IPv4-mapped address）
  - IPv4アドレスをIPv6アドレスとして表す特殊なアドレス



  - 例: 192.0.2.128の場合

    ::ffff:192.0.2.128もしくは::ffff:c000:280
  - IPv6対応アプリケーションがIPv4/IPv6対応のソケット"::"でIPv4のみを持つノードと通信する際に利用
    - ノード内部での利用に限定
    - 送信元・宛先アドレスとしては利用されない
  - IPv4射影アドレスの無効化が好ましい
    - IPv4で利用できなくなるアプリケーションもあるので注意

A-6-2.IPv4射影アドレスを巡る議論

- **問題点を指摘する**Internet Draft（RFC**には至らず**）
  - IPv4-Mapped Addresses on the Wire Considered Harmful
    http://tools.ietf.org/html/draft-itojun-v6ops-v4mapped-harmful-02
  - IPv4-Mapped Address API Considered Harmful
    http://tools.ietf.org/html/draft-cmetz-v6ops-v4mapped-api-harmful-01
- IPv4-Mapped Address API Considered Harmful
  - **コードの移植性が低下**
  - **実装の複雑化**
  - **アクセス制御が複雑化**
    - **同じノードと通信する場合でも、**"192.0.2.128 (AF_INET)" **と** "::ffff:192.0.2.128 (AF_INET6)" **のように見え方が異なる**

A-6-3.IPv4射影アドレスを利用しない方法

- IPv4射影アドレスを利用しないsysctl変数の設定
  - Linux（kernel 2.4.21以降および2.6以降）
    - net.ipv6.bindv6only変数を1に設定する
    - デフォルトではIPv4射影アドレスが有効
    - 一部のディストリビューションのkernelの挙動
  - FreeBSD
    - net.inet6.ip6.v6only変数を1に設定する
    - 最近のバージョンではデフォルトでIPv4射影アドレスが無効
- アプリケーション側での対応
  - IPV6_V6ONLYソケットオプションを利用

IPv4射影アドレスを利用しない場合、IPv6のソケットとは別に
別途IPv4のソケットを開く必要がある

A-6-4.ソケットの開き方: OpenSSHの場合

- IPv4/IPv6で別々にソケットをbind(2)する場合
  - "netstat –an"コマンドの結果:

sshd_configファイル:
ListenAddress 0.0.0.0
ListenAddress ::

```
tcp      0 0 0.0.0.0:22              0.0.0.0:*                   LISTEN
tcp      0 0 :::22                   :::*                        LISTEN
tcp      0 0 192.0.2.2:22            10.0.0.1:34634              ESTABLISHED
```

- IPv6でのみソケットをbind(2)する場合
  - "netstat –an"コマンドの結果:

sshd_configファイル:
ListenAddress ::

```
tcp      0 0 :::22                   :::*                        LISTEN
tcp      0 0 ::ffff:192.0.2.2:22     ::ffff:10.0.0.1:34644       ESTABLISHED
```

ソケットの開き方は、TCP/IPスタックの設定、
個々のアプリケーションの実装とListen設定方法により異なる
アプリケーション毎に確認したほうがよい

A-6-5.IPv4射影アドレスとソケット

- IPv6プロトコルスタックを有効にした場合の影響
  - IPv6に対応しているサーバアプリケーション側で、"::"をbind(2)しただけのアプリケーションでは、IPv4に関する挙動が変わる
    - AF_INETソケットのIPv4アドレスではなく、AF_INET6ソケットのIPv4 Mapped Addressが利用される（"192.0.2.1"ではなく"::ffff:192.0.2.1"となる）

bind(2) の対象		IPv6 クライアントからの接続	IPv4 クライアントからの接続
	IPV6_V6ONLY ソケットオプション		
0.0.0.0 IN_ADDR_ANY		×	○ 192.0.2.1
:: IN6_ADDR_ANY	Yes	○	×
	No	○	○ :ffff:192.0.2.1 IPv4 射影アドレス

A-7.セキュリティ関連

- ルーティングヘッダType 0（RH0）
  - ソースルーティングによる問題が指摘
    - http://jvn.jp/cert/JVNVU267289/index.html
    - http://www.securityfocus.com/bid/23615
    - RFC 5095において使用禁止に
- On-link Assumption
  - デフォルトルートがない場合全てのノードがオンリンクである前提仕様が存在していたが現在は廃止
    - 古い端末OSにおいて実装が残っている場合がある

A-8-1.その他(1)

- ルータ広告とDHCPv6の連携
  - ルータ広告のOフラグとMフラグによりDHCPv6に移行
    - ただしRFC 4862では削除された仕様
  - Windows系の実装では上記動作がデフォルトで機能する
    - Oフラグ設定でステートレスDHCPv6が動作
      - DNSサーバなどのネットワークパラメータのみ取得
    - Mフラグ設定でステートフルDHCPv6が動作
      - IPv6アドレスもDHCPv6で設定

A-8-2.その他(2)

- SEND (RFC 3971)
  - 近隣探索プロトコル(NDP)のセキュリティ対応版
  - 特許(NTT DoCoMo America)関連などで未普及
  - 端末OSにおける実装はほぼなし
    - Easy-SEND: JAVAによるSENDの実装利用が可能

    http://sourceforge.net/project/screenshots.php?group_id=257017&ssid=102894
- MIPv6 (RFC 3775)
  - 移動ノード(MN)としての実装と通信相手(CN)としての双方の実装がある
    - MN対応:ホームアドレス利用やルーティングヘッダType 2の対応
    - CN対応:経路最適化のための宛先オプションヘッダの対応

A-9-1.Windows XPのIPv6実装概要

- 主な特徴
  - IPv6に対応させるには下記のコマンドを実行する必要あり
    - C:¥ >netsh interface ipv6 install
  - DNSリゾルバはIPv6トランスポートに未対応
    - nameproxyという外部アプリケーションで対応可能
  - 名前解決はAAAAクエリを優先
  - アドレス選択機構には独自実装あり
    - 一時アドレス(プライバシ拡張アドレス)を優先利用
    - ポリシーテーブルにTeredoアドレスを追記

      2001::/32　5　5　　※SP2の途中で旧プレフィックス(3ffe:831f::/32)から変更された
  - RFC 4191対応はSP2とSP3で異なる
    - SP3から特定経路の通知に対応

A-9-2.Windows XPのIPv6実装一覧

機能		実装	挙動
DNS	クエリー順序	ー	AAAAクエリを実施後Aクエリを実施
	トランスポート	未対応	nameproxyという外部アプリケーションで対応可能
アドレス選択機構（RFC 3484）		対応	パブリックアドレスよりも一時アドレスが優先される Teredoアドレスがポリシーテーブルに追加
RFC 4191	ルータ優先度	対応	同じ優先度の場合、後で受信したものを利用
	特定の経路	対応	SP3から利用可能
自動トンネル	6to4	対応	C:¥ >netsh interface ipv6 6to4 set state state=enableで有効化が可能（デフォルト無効）
	Teredo	対応	C:¥ >netsh interface ipv6 set teredo enterpriseclient ¥ teredo.ipv6.microsoft.comで有効化が可能（デフォルト無効）
IPv4射影アドレス		無効	機能を有効化する方法はなし
ルーティングヘッダType 0		無効	受信した場合に破棄する
on-link assumption		無効	SP2には実装は残っているがデフォルトルートがない場合にはIPv4を利用する
ルータ広告とDHCPv6の連携		対応	Oフラグでステートレス、Mフラグでステートフル DHCPv6に移行

A-9-3.Windows VistaのIPv6実装概要

- 主な特徴
  - デフォルトでIPv6 Ready
    - IPv4機能の無効化はコマンドで可能だがIPv6機能は不可能
    - インターフェースで機能を利用しない設定は可能
  - Windowsアプリケーションが IPv6で利用可能
    - DNSリゾルバもIPv6トランスポートに対応
  - 名前解決はAクエリを優先
    - Aクエリの応答時間を基にAAAAクエリの待ち時間を決定
    - AクエリでNXDOMAINならAAAAクエリを抑制
    - グローバルIPv6アドレスが付与されない限りAAAAクエリを抑制

      ※Teredoアドレスを除くグローバルIPv6アドレス
  - RFC 4191対応はWindows XPと異なる
    - ルータ優先度が同じ場合に先に受信したものを優先

A-9-4.Windows VistaのIPv6実装一覧

機能		実装	挙動
DNS	クエリー順序	ー	Aクエリを実施後AAAAクエリを実施 Aクエリの応答時間を基にAAAAクエリの待ち時間を決定 グローバルIPv6アドレスが付与されない限りAAAAクエリを抑制
	トランスポート	対応	IPv6のDNSサーバを優先利用
アドレス選択機構（RFC 3484）		対応	パブリックアドレスよりも一時アドレスが優先される Teredoアドレスがポリシーテーブルに追加
RFC 4191	ルータ優先度	対応	同じ優先度の場合、先に受信したものを利用
	特定の経路	対応	設定不要で利用可能
自動トンネル	6to4	対応	デフォルトでグローバルIPv4アドレスが設定されると利用可能に
	Teredo	対応	NATルータ配下のプライベートIPv4アドレスが設定されると利用可能に（Windows Firewallが有効な場合）
IPv4射影アドレス		無効	機能を無効化する方法はなし
ルーティングヘッダType 0		有効	受信した場合に破棄する
on-link assumption		無効	
ルータ広告とDHCPv6の連携		対応	Oフラグでステートレス、Mフラグでステートフル DHCPv6に移行

A-9-5.Windows 7 RCのIPv6実装概要

- 主な特徴
  - Windows Vistaと同様の実装
    - デフォルトで IPv6 Ready
    - WindowsアプリケーションがIPv6で利用可能
    - 名前解決はAクエリを優先
    - 自動トンネル機能がデフォルトで有効

A-9-6.Windows 7 RCのIPv6実装一覧

機能		実装	挙動
DNS	クエリー順序	ー	Aクエリを実施後AAAAクエリを実施 Aクエリの応答時間を基にAAAAクエリの待ち時間を決定 グローバルIPv6アドレスが付与されない限りAAAAクエリを抑制
	トランスポート	対応	IPv6のDNSサーバを優先利用
アドレス選択機構（RFC 3484）		対応	パブリックアドレスよりも一時アドレスが優先される Teredoアドレスがポリシーテーブルに追加
RFC 4191	ルータ優先度	対応	同じ優先度の場合、先に受信したものを利用
	特定の経路	対応	設定不要で利用可能
自動トンネル	6to4	対応	デフォルトでグローバルIPv4アドレスが設定されると利用可能に
	Teredo	対応	NATルータ配下のプライベートIPv4アドレスが設定されると利用可能に（Windows Firewallが有効な場合）
IPv4射影アドレス		無効	機能を無効化する方法はなし
ルーティングヘッダType 0		有効	受信した場合に破棄する
on-link assumption		無効	
ルータ広告とDHCPv6の連携		対応	Oフラグでステートレス、Mフラグでステートフル DHCPv6に移行

A-9-7.Mac OS XのIPv6実装概要

- 主な特徴
  - FreeBSD 5系におけるKAMEの実装がベース
    - RFC 3484に未対応など実装が古い
  - デュアルスタックに対してはIPv6を優先利用
    - RFC 3484をサポートしていないため、IPv4を優先利用するためにはIPv6をオフにする必要がある
    - DNSもIPv6設定を優先

A-9-8.Mac OS XのIPv6実装一覧

機能		実装	挙動
DNS	クエリー順序	ー	Aクエリを実施後AAAAクエリを実施
	トランスポート	対応	IPv6のDNSサーバを優先利用
アドレス選択機構（RFC 3484）		未対応	再長一致の挙動
RFC 4191	ルータ優先度	未対応	先に受信したものを利用
	特定の経路	未対応	
自動トンネル	6to4	対応	トンネルインタフェースstf（10.2 以降）にて利用可能
	Teredo	外部対応	miredoにより利用可能
IPv4 射影アドレス		有効	net.inet6.ip6.v6only = 1で無効化可能
ルーティングヘッダType 0		無効	10.4.10より以前のバージョンでは処理するため更新が必要
on-link assumption		無効	
ルータ広告とDHCPv6の連携		未対応	

A-9-9.Linux（2.6系）のIPv6実装概要

- 主な特徴
  - アドレス選択機構は他の実装と異なる(2.6.25以上)
    - 終点アドレス選択設定と始点アドレス選択設定が別々
      - 終点：/etc/gai.confファイル　　始点：ip addrlabelコマンド
    - ポリシーテーブルにULAやTeredoアドレスを追記

      fc00::/7 label 5, 2001::/32 label 6
  - 名前解決はAAAAクエリを優先
  - 2.6系のカーネルバージョンにより機能が異なる
    - 2.6.9　　on-link assumptionを無効化
    - 2.6.20.9　　ルーティングヘッダType 0の全面禁止
    - 2.6.17　　RFC 4191を実装
    - 2.6.25　　設定可能なポリシーテーブルを実装

A-9-10.Linux（2.6系）のIPv6実装一覧

機能		実装	挙動
DNS	クエリー順序	ー	AAAAクエリを実施後Aクエリを実施
	トランスポート	対応	/etc/resolv.confにおける設定順序に依存
アドレス選択機構（RFC 3484）		対応	2.6.25からポリシーテーブルの設定が可能
RFC 4191	ルータ優先度	対応	2.6.17から対応（CONFIG_IPV6_ROUTER_PREF） 同じ優先度の場合、先に受信したものを利用
	特定の経路	対応	2.6.17から対応（CONFIG_IPV6_ROUTE_INFO） accept_ra_rt_info_max_plen = <num> で受け入れを許容する プレフィックス長を設定（48 だと/0～/48までの範囲で許可）
自動トンネル	6to4	対応	トンネルインタフェースsitにて利用可能
	Teredo	外部対応	miredoにより利用可能
IPv4射影アドレス		有効	net.ipv6.bindv6only = 1で無効化可能
ルーティングヘッダType 0		無効	ルータとして動作させる場合にのみ有効となる 2.6.20.9からはルータ利用においても禁止実装に
on-link assumption		無効	2.6.9以前には実装が残っていて問題（2.4系も問題）
ルータ広告とDHCPv6の連携		未対応	

A-9-11.FreeBSDのIPv6実装概要

- 主な特徴
  - 最初にKAMEのコードがマージされたOS（2000年3月）
  - RFCに準拠した実装
    - RFC 3484のポリシーテーブルに独自追加なし
  - 名前解決はAクエリを優先
    - Aクエリの応答時間を基にAAAAクエリの待ち時間を決定
  - RFC 4191の特定の経路の受信は実装されていない
    - ルータ広告デーモン（rtadvd）における機能実装は完了

A-9-12.FreeBSDのIPv6実装一覧

機能		実装	挙動
DNS	クエリー順序	ー	Aクエリを実施後AAAAクエリを実施 Aクエリの応答時間を基にAAAAクエリの待ち時間を決定
	トランスポート	対応	/etc/resolv.confにおける設定順序に依存
アドレス選択機構（RFC 3484）		対応	6.2R以前では/etc/rc.confにおけるデフォルト設定が異なるため ip6addrctl_enable="YES"が必要
RFC 4191	ルータ優先度	対応	6.1Rから対応　同じ優先度の場合、先に受信したものを利用
	特定の経路	未対応	
自動トンネル	6to4	対応	トンネルインタフェースsitにて利用可能
	Teredo	外部対応	miredoにより利用可能
IPv4 射影アドレス		無効	net.inet6.ip6.v6only = 0で有効化可能
ルーティングヘッダType 0		無効	6.3Rからsysctlで制御可能に net.inet6.rthdr0_allowed = 1で利用可能（デフォルト無効） 7.0R からは net.inet6.rthdr0_allowedはなく完全に無効
on-link assumption		無効	
ルータ広告とDHCPv6の連携		未対応	

Appendix B

トランスレータの標準化動向

B-1.標準化されたトランスレータ技術

- 三つのRFCが存在していた
  - RFC 2765
  - RFC 2766
  - RFC 3142

B-2.RFC 2765

- Stateless IP/ICMP Translator Algorithm (SIIT)
  - Layer 3での変換方法を定義したもの
    - IPv4とIPv6ヘッダの変換方法
      - 各フィールドの変換
      - 存在しないヘッダ/フィールドの変換
    - ICMPv4とICMPv6の変換方法
      - 各ヘッダの変換
      - エラーメッセージの変換
  - 変換セッションの管理は行わないシンプルなもの

B-3.RFC 2766

- Network Address Translation – Protocol Translation (NAT-PT)
  - Layer 3で変換処理を行うトランスレータを定義
    - TCP/UDPのポートマッピング方法の定義
      - NAT-PT
      - NAPT-PT
    - ALGの定義
      - DNS ALG
      - FTP ALG
    - アドレスマッピング方法の定義
    - ヘッダ変換については、SIITを参照している
  - ステートフルトランスレータの仕様である

これらの ALG はトランスレータに内蔵されている

B-4.RFC 3142

- Transport Relay Translator
  - トランスポート層で行うトランスレータの定義
    - トランスポートリレーを応用したもの
    - IPヘッダの変換は行わない
      - パケットを一旦受信してから送信する
      - MTUを気にしなくてよいというメリットがある
  - TCP/UDPの変換のみをサポートしている

B-5.RFC 2766がヒストリカル状態へ

- RFC 4966の登場
  - Reason of Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic State
  - 大分すると二つの問題が報告されている
    - DNS-ALGに起因する問題
      - single point failure
      - DNS SEC　　など
    - DNS-ALGに起因しない問題
      - Fragment
      - IPsec　　など
- これにより、Layer 3で処理をする標準化されたステートフルトランスレータが存在しない状態となった

B-6.今後の動向

- 現在IETF behave wgにてIPv6移行技術に関する議論が行われている。
  - http://www.ietf.org/html.charters/behave-charter.html
    - behave wgは、もともとNATに関する議論を行ってきた
  - IPv6移行期のシナリオを定義して、それぞれに適応した技術の議論を行っている。
    - トランスレータ
    - トンネル技術
    - Large Scale NAT (LSN)
- 現時点*ではwg draftは存在しない
  - wg draftへ向けての作業を実施中

＊ 2009/05 現在

B-7.トランスレータ関連のドラフト

- 2009/05時点で発行されているトランスレータ関連ドラフト

項目	draft 名
Translation Framework	draft-baker-behave-v4v6-framework
Stateless Translation	draft-baker-behave-v4v6-translation
Stateful Translation	draft-bagnulo-behave-nat64
DNS Translation	draft-bagnulo-behave-dns64
FTP Application Layer Gateway	draft-beijnum-behave-ftp64

Appendix C

リバースプロキシの設定例とログ例

C-1.設定例: stone

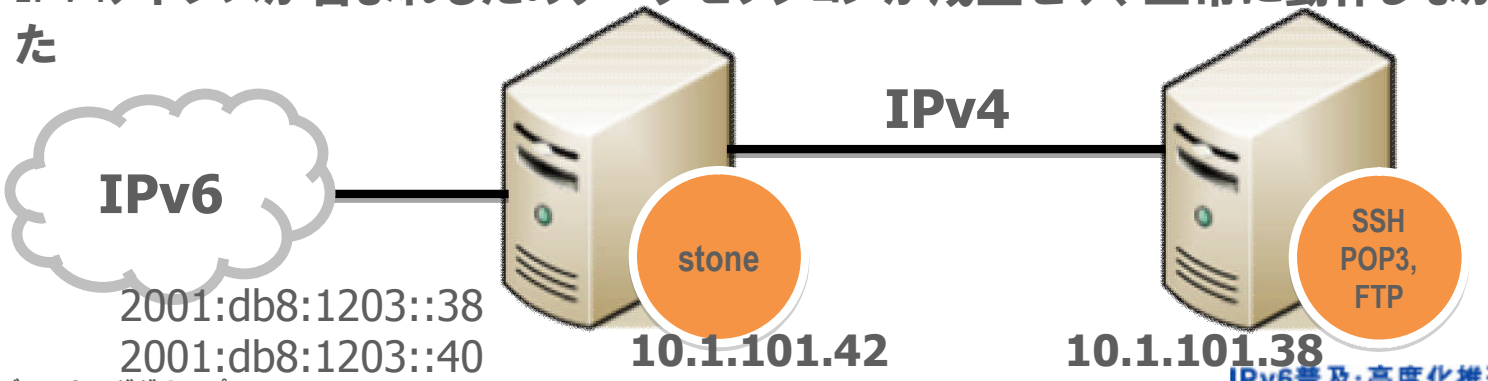
IPv4/IPv6デュアルスタックのソケットリピータ

例: 中継ノードにて、全IPv6アドレスの110番ポート、2001:db8:1203::38:22の22番ポート、全IPv6アドレスの21番ポートを10.1.101.38に転送する設定

```
•/etc/stone.conf に以下の設定を追加:
-a /var/log/stone.log
10.1.101.38:110/tcp 110/tcp,v6 --
10.1.101.38:22/tcp 2001:db8:1203::38:22/tcp,v6 --
10.1.101.38:21/tcp 21/tcp,v6
```

```
•/var/log/stone.log のログ
•Jan 16 14:05:40 110[1] 2001:db8:1203::40:60821/v6[2001:db8:1203::40]-4715
•Jan 16 15:54:16 22[2] 2001:db8:1203::38:53092/v6[2001:db8:1203::38]-1244
```

結果: SSH、POP3のポート転送は正常に動作したが、FTPについてはペイロードにIPv4アドレスが含まれるためデータセッションが成立せず、正常に動作しなかった



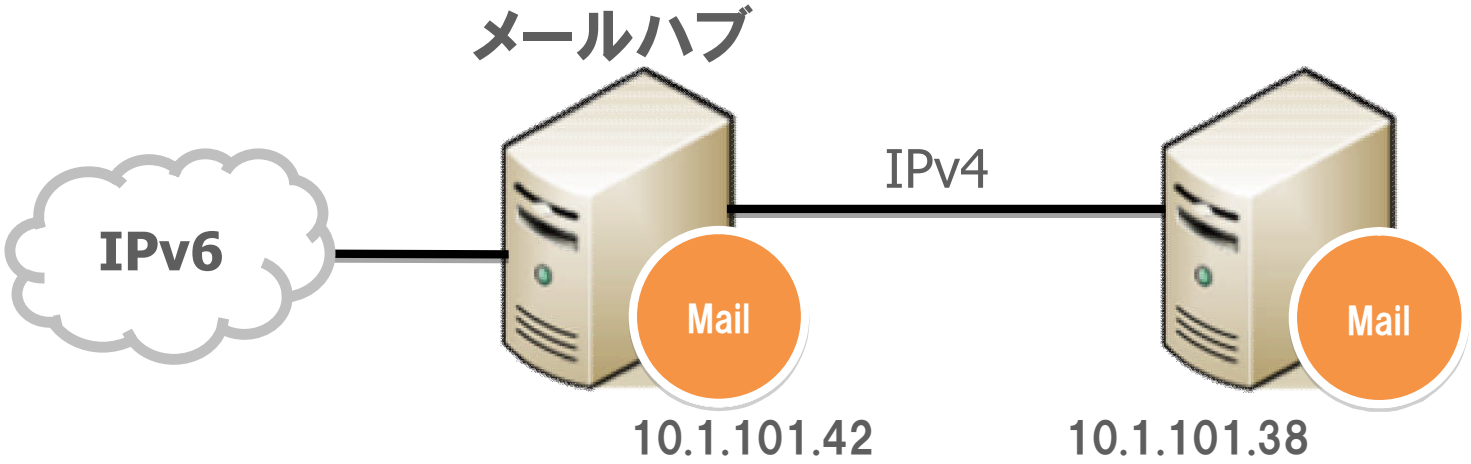
C-2-1.設定例: sendmail（1）

IPv4/IPv6デュアルスタックのメールハブ

例: メールハブにおいてcamp.ipv6lab.jpドメイン宛てのメールを、IPv4シングルスタックのメールサーバ(10.1.101.38)へメールを中継する設定

```
•/etc/mail/sendmail.mc
dnl DAEMON_OPTIONS(`port=smtp,Addr=::1, Name=MTA-v6, Family=inet6')dnl
                                        ↓
DAEMON_OPTIONS(`port=smtp,Addr=::, Name=MTA-v6, Family=inet6')dnl
```

```
•/etc/mail/mailertable
amp.ipv6lab.jp smtp:[10.1.101.38]
```



C-2-2.設定例: sendmail（2）

メールログ

外部のメールサーバからのメールログ

```
Jan 16 16:04:27 sv07 sendmail[17924]: n0G74RVF017924: from=<true@v6pc.example.com>,
size=667, class=0, nrcpts=1, msgid=<20090116160421.E4AC.94DBBD5E@v6pc.example.com>,
proto=ESMTP, daemon=MTA-v6, relay=[IPv6:2001:db8:0:8803:20d:56ff:feb8:1f63]
Jan 16 16:04:27 sv07 sendmail[17926]: n0G74RVF017924: to=<shin@camp.example.com>,
delay=00:00:00, xdelay=00:00:00, mailer=smtp, pri=120667, relay=[10.1.101.38]
[10.1.101.38], dsn=2.0.0, stat=Sent (n0G74K4F006753 Message accepted for delivery)
```

受信したメールのヘッダ:

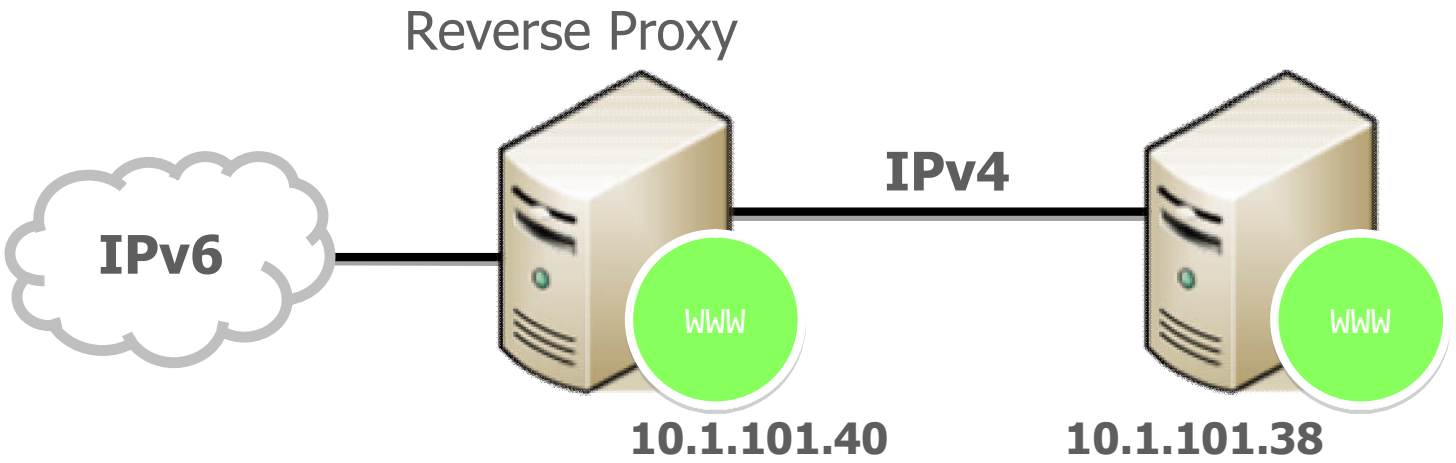
```
Return-Path: <true@v6pc.example.com>
Received: from sv07.camp.example.com ([10.1.101.42])
        by sv05.camp.example.com (8.13.8/8.13.8) with ESMTP id n0G74K4F006753
        for <shin@camp.example.com>; Fri, 16 Jan 2009 16:04:20 +0900
Received: from mail.v6pc.example.com ([IPv6:2001:db8:0:8803:20d:56ff:feb8:1f63])
        by sv07.camp.example.com (8.13.8/8.13.8) with ESMTP id n0G74RVF017924
        for <shin@camp.example.com>; Fri, 16 Jan 2009 16:04:27 +0900
Received: from [10.1.1.102] (unknown [IPv6:2001:db8:e02:fd:88fb:9801:c8d0:fc86])
        by mail.v6pc.example.com (Postfix) with ESMTP id 049674C13B;
        Fri, 16 Jan 2009 16:04:26 +0900 (JST)
Date: Fri, 16 Jan 2009 16:04:26 +0900
From: Shin SHIRAHATA <true@v6pc.example.com>
To: shin@camp.example.com
```

C-3-1.設定例: Apache（1）

IPv4/IPv6デュアルスタックのリバースプロキシ

例: リバースプロキシにおいてsv05-01.camp.example.comドメイン宛てのアクセスを、IPv4シングルスタックのWebサーバ(10.1.101.38)へ中継する設定

```
•/etc/hosts
# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1         sv07.camp.example.com sv07 localhost.localdomain localhost
::1               localhost6.localdomain6 localhost6
10.1.101.38       sv05-01.camp.example.com
```



C-3-2.設定例: Apache（2）

リバースプロキシの/etc/httpd/conf/httpd.conf

```
<VirtualHost 120.88.27.38:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /var/www/html
    ProxyPass / http://sv05-01.camp.example.com/
    ProxyPassReverse / http://sv05-01.camp.example.com/
    ServerName sv05-01.camp.example.com
    ErrorLog logs/sv05-01.error_log
    CustomLog logs/sv05-01.access_log common
</VirtualHost>

<VirtualHost [2001:db8:1203::38]:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /var/www/html
    ProxyPass / http://sv05-01.camp.example.com/
    ProxyPassReverse / http://sv05-01.camp.example.com/
    ServerName sv05-01.camp.example.com
    ErrorLog logs/sv05-01.error_log
    CustomLog logs/sv05-01.access_log common
</VirtualHost>
```

C-3-3.設定例: Apache（3）

アクセスログ

•リバースプロキシへのアクセスログ（抜粋）
•**sv05-01.access_log**

```
2001:db8:1203::40 - - [16/Jan/2009:17:25:22 +0900] "GET / HTTP/1.0" 200 385
10.1.63.245 - - [16/Jan/2009:17:25:37 +0900] "GET / HTTP/1.0" 200 385
```

•**sv05-01.error_log**
特になし

•Webサーバのログ（抜粋）

```
10.1.101.42 - - [16/Jan/2009:17:25:51 +0900] "GET / HTTP/1.1" 200 385 "-" "Mo
zilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CL
R 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)
Sleipnir/2.8.4"
10.1.101.42 - - [16/Jan/2009:17:26:30 +0900] "GET / HTTP/1.1" 200 385 "-" "Mo
zilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CL
R 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)
Sleipnir/2.8.4"
```

備考

- Webサーバへのアクセス元IPアドレスはリバースプロキシとなる
- リバースプロキシでキャッシュ機能が有効な場合、アクセス件数とWebサーバへのアクセス数は一致しない

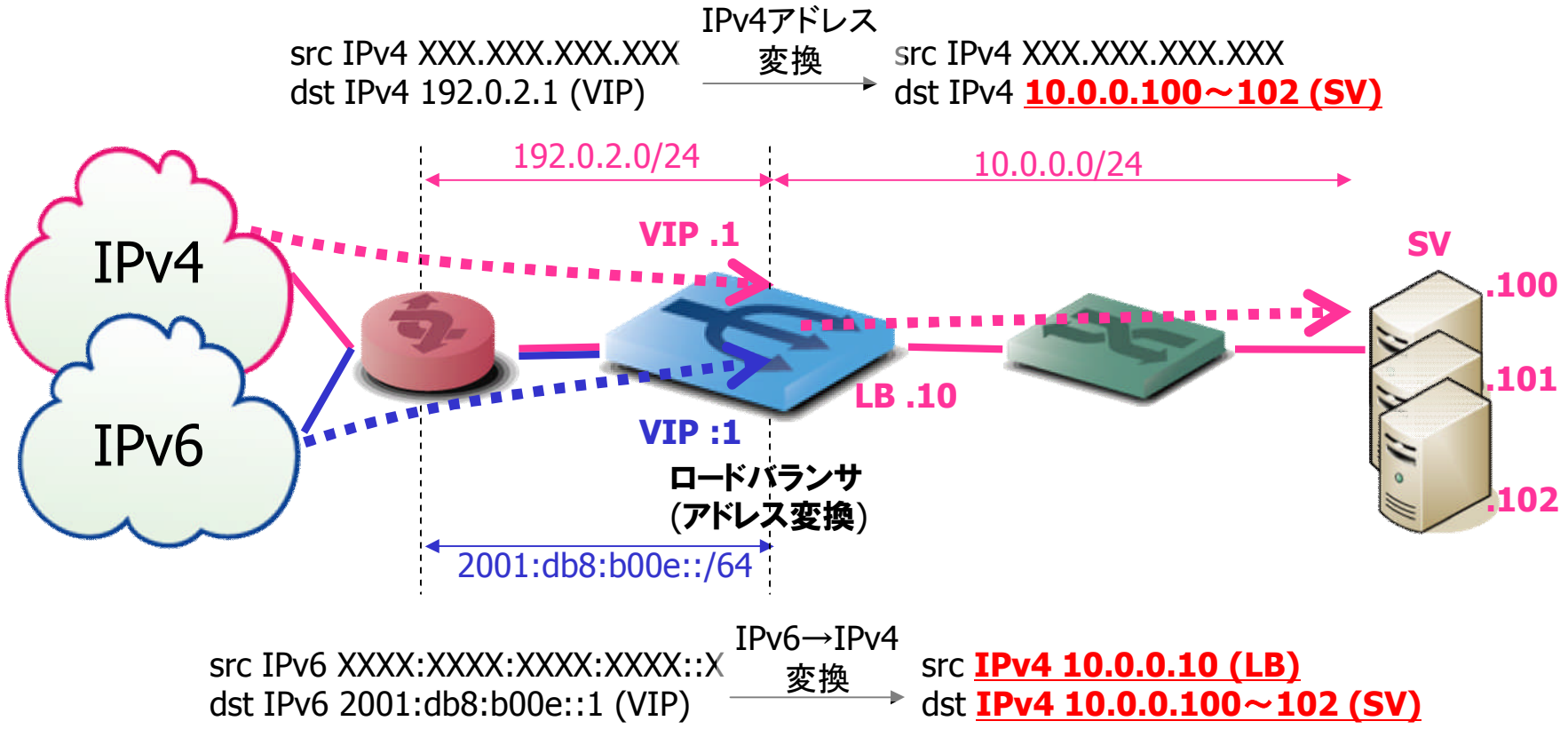
Appendix D

ロードバランサ構成例

D-1.IPv6適用構成例(1)

通過型配置(a)

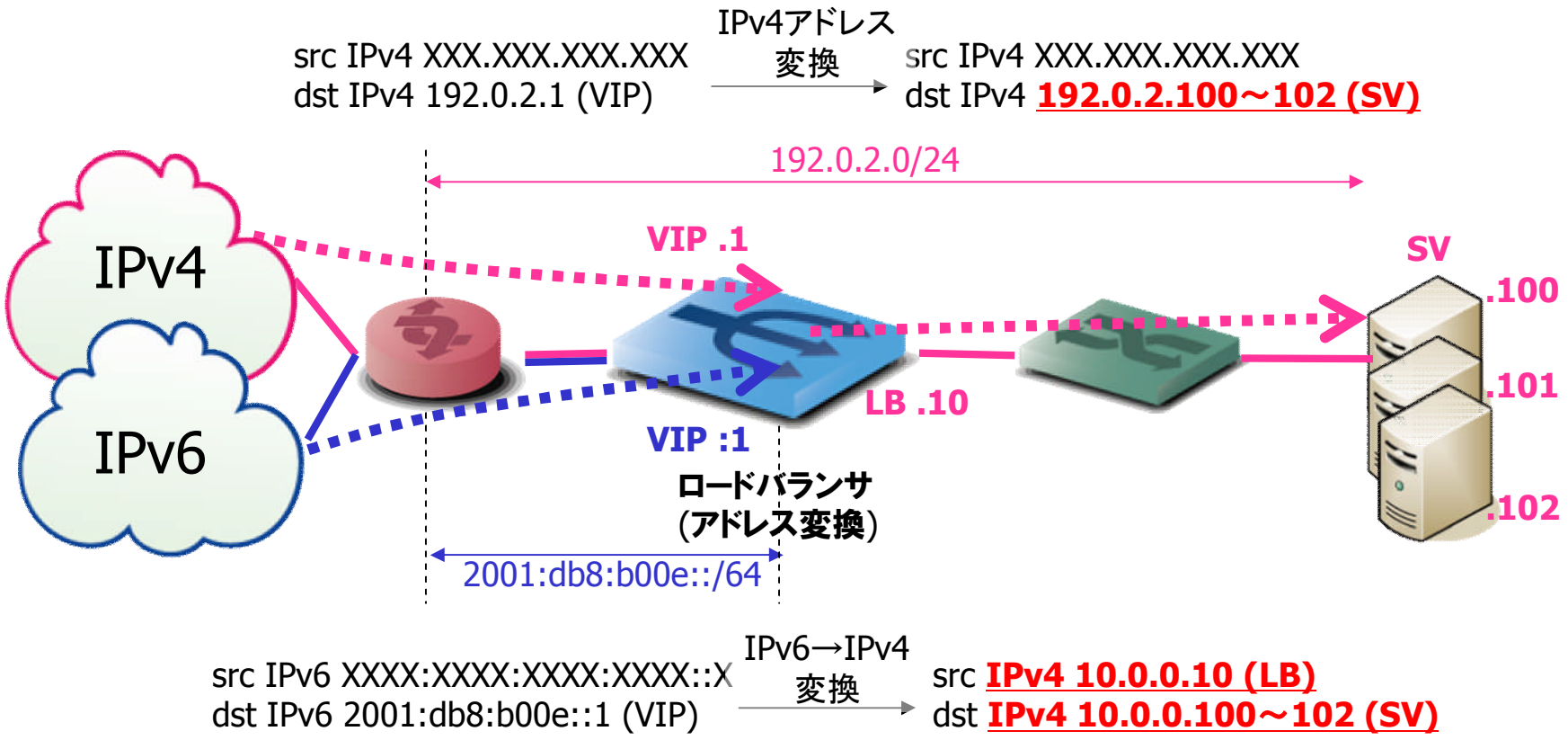
IPv4はL3中継の場合にIPv6→IPv4変換を追加



D-2.IPv6適用構成例(2)

通過型配置(b)

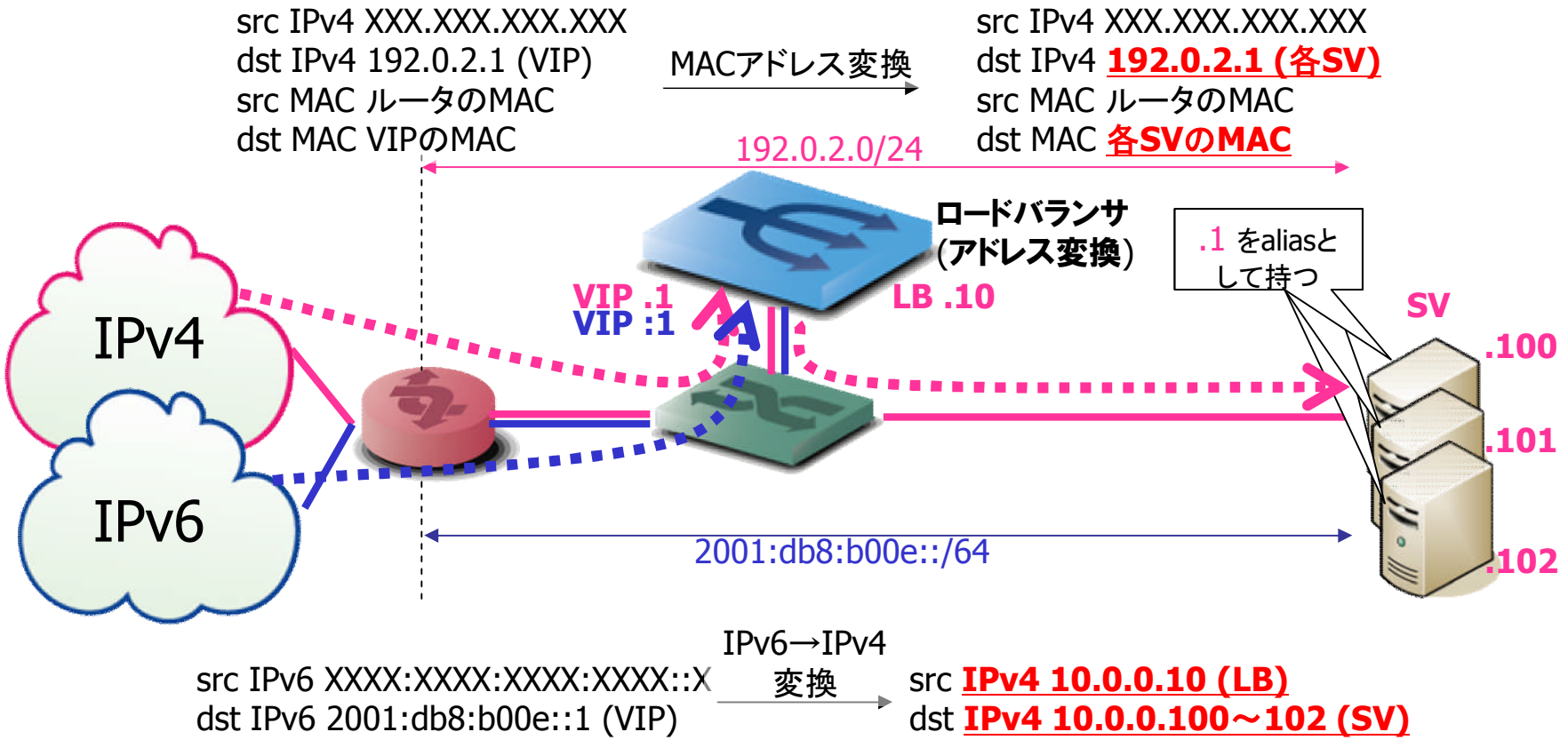
IPv4はL2中継の場合にIPv6→IPv4変換を追加



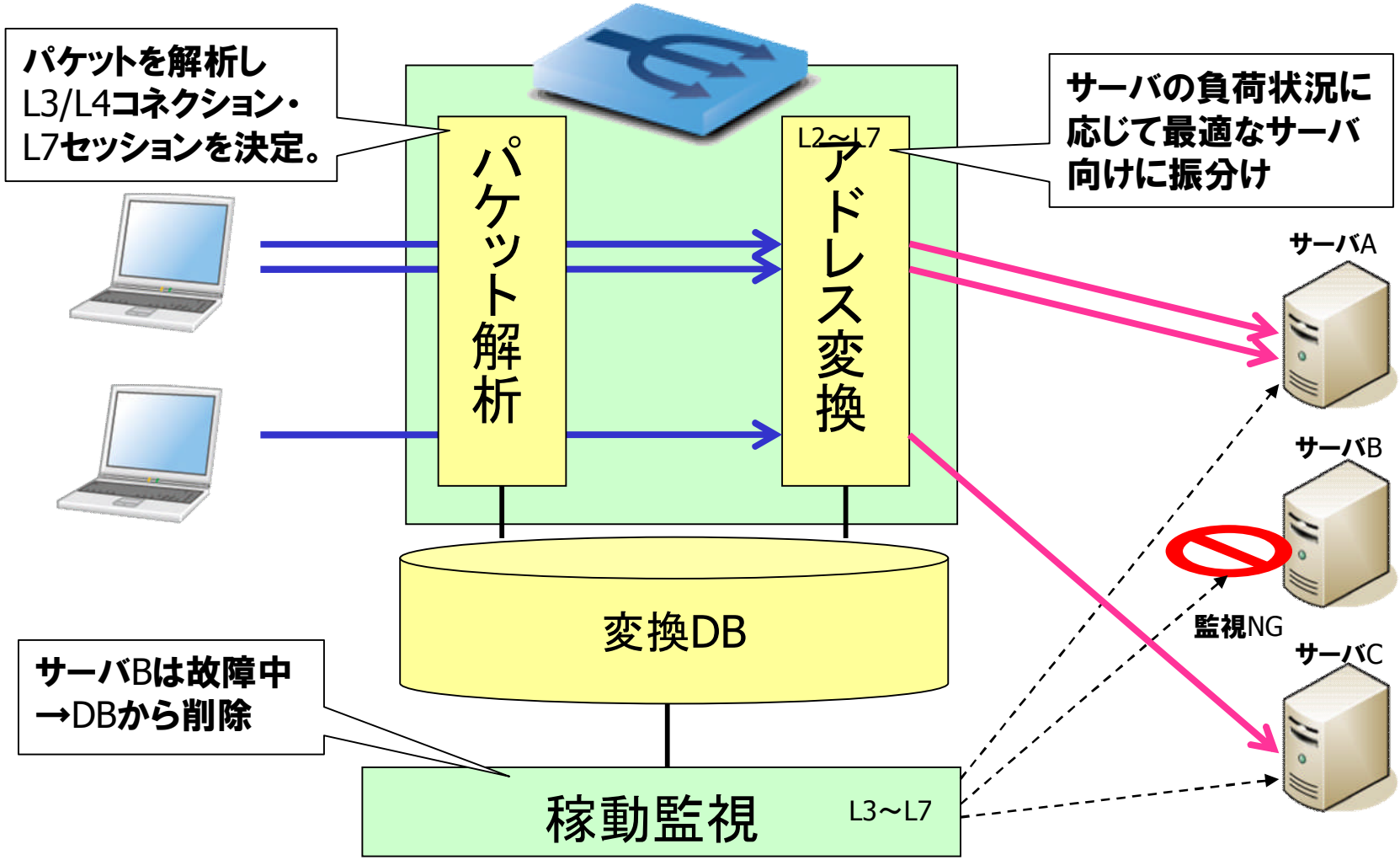
D-3.IPv6適用構成例(3)

並列型配置

IPv4並列型構成にIPv6→IPv4変換を追加



D-4.[参考]ロードバランサの構造



Appendix E

Nagiosでの設定例

E-1.Nagiosについて

- Nagios（http://www.nagios.org/）はIPv6監視に対応しており、IPv6対応OS上で動作させることでIPv6監視システムを構築することが可能
- 使用したバージョン
  - nagios 3.0.6
  - nagios-plugin-1.4.13
  - http://www.nagios.org/download/download.php
  - 実験では上記URLから取得したtarballを元にRPMを作成しインストールを行った

E-2.設定例での環境

- 監視対象ホスト名
  - www.example.jp
  - DNSには上記ホスト名をAとAAAAで登録する。
- 監視対象IPアドレス
  - 10.10.10.1
  - 2001:db8::1
- 監視対象サービス
  - web（IPv6とIPv4で同じコンテンツを利用）
- 監視項目は以下をIPv4とIPv6で設定
  - ホスト名指定でのICMP監視
    - ping,ping6
  - ホスト名指定でのHTTPサービス監視
    - check_http4, check_http6
  - IPアドレス指定でのICMP監視
    - check_int, check_int6
  - 明示的にコマンド名をIPv4/IPv6を分けて定義する事によって設定や管理、運用が容易になる

E-3.設定ファイル

- IPv6監視に関わる設定ファイル
  - commands.cfg
    - 主にサービスチェックで実際に使用するコマンドを定義
  - hosts.cfg
    - Hostオブジェクトを定義し、ホストチェックに関する設定を行う

E-4-1.コマンド設定例：commands.cfg（1）

Ping6（ホスト名指定でのIPv6 ICMP監視）

```
define command{
  command_name    check_ping
  command_line    $USER1$/check_ping -6 -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p
  }
```

check_http4（ホスト名指定でのIPv4 HTTPサービスチェック）

```
define command{
  command_name    check_http4
  command_line    $USER1$/check_http -4 -H $ARG1$
  }
```

check_http6（ホスト名指定でのIPv6 HTTPサービスチェック）

```
define command{
  command_name    check_http6
  command_line    $USER1$/check_http -6 -H $ARG1$
  }
```

E-4-2.コマンド設定例：commands.cfg（2）

check_int（IPv4アドレス指定でのICMP監視）

```
define command{
  command_name    check_int
  command_line    $USER1$/check_ping -H $ARG3$ -w $ARG1$ -c $ARG2$ -p 5
  }
```

check_int6（IPv6アドレス指定でのICMP監視）

```
define command{
  command_name    check_int6
  command_line    $USER1$/check_ping -6 -H $ARG3$ -w $ARG1$ -c $ARG2$ -p 5
  }
```

E-4-1.ホスト設定例：hosts.cfg（1）

ホスト名設定

```
define host{
        use  default
        host_name     www.example.jp
        alias         www
        address       10.10.10.1
        }
```

ホスト名指定でのICMP監視（IPv4）

```
define service{
        use                     local-service
        host_name               www.example.jp
        service_description     PING
        check_command           check_ping!100.0,20%!500.0,60%
        }
```

ホスト名指定でのICMP監視（IPv6）

```
define service{
        use                     local-service
        host_name               www.example.jp
        service_description     PING6
        check_command           check_ping6!100.0,20%!500.0,60%
        }
```

E-4-2.ホスト設定例：hosts.cfg（2）

ホスト名指定でのIPv4 HTTPサービスチェック

```
define service{
        use                     local-service
        host_name               www.example.jp
        service_description     HTTP
        check_command           check_http4!www.example.jp
        }
```

ホスト名指定でのIPv6 HTTPサービスチェック

```
define service{
        use             local-service
        host_name               www.example.jp
        service_description     HTTP6
        check_command           check_http6!www.example.jp
        }
```

E-4-3.ホスト設定例：hosts.cfg（3）

IPv4アドレス指定でのICMP監視

```
define service{
        use                     local-service
        host_name               www.example.jp
        service_description     check_int
        check_command           check_int!100.0,20%!500.0,60%!192.168.1.1
        }
```

IPv6アドレス指定でのICMP監視

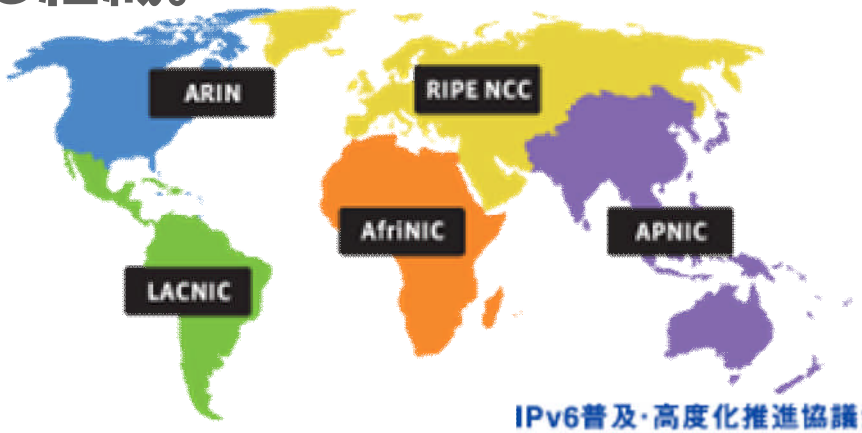
```
define service{
        use                      local-service
        host_name               www.example.jp
        service_description     check_int6
        check_command           check_int6!100.0,20%!500.0,60%!2001:db8::1
        }
```

Appendix F

用語集

F-1.用語集①

- IANA
  - Internet Assigned Numbers Authority
    http://www.iana.org/
- RIR
  - Regional Internet Registry （地域インターネットレジストリ）
    IANA より委託されたインターネットリソースを、管轄する地域において、管理・配分する組織。
    以下の組織が存在する。



F-2.用語集②

RIR(Cont.)

名称	略称	管轄地域	URL
African Network Information Centre	AfriNIC	Africa	http://www.afrinic.net/
Asia-Pacific Network Information Centre	APNIC	Asia-Pacific Region	http://www.apnic.net/
American Registry for Internet Numbers	ARIN	North America	https://www.arin.net/
Latin American and Caribbean Internet Address Registry	LACNIC	South America	http://www.lacnic.net/
Réseaux IP Européens Network Coordination Centre	RIPE NCC	Europe, Central Asia and the Middle East	http://www.ripe.net/

F-3.用語集③

- LSN Large Scale NAT : **大規模NAT**
  - **従来インターネットサービス事業者は1件のユーザに対して1つのIPv4グローバルアドレスを割り当てていたが、IPv4グローバルアドレスの枯渇後はこの構成を拡張することができなくなる。そこで複数のユーザが1つのIPv4グローバルアドレスを共用するネットワークモデルがIETFで提案されている（NAT444モデル）。このネットワークにおいて事業者側に設置する複数ユーザ収容可能のNATのことをLSNという。当初、CGN(Carrier Grade NAT)と呼ばれていたが、本来キャリア専用に考案されたものではないことから2008年11月のIETF73においてCGNの提案者からLSNという用語を使うよう提案があった。**

F-4.用語集④

- PMTUD : Path MTU Discovery
  - IPv6ノードが通信を行う際、より効率的に通信を行う為に パケットサイズを大きくしようと試みる。IPv6では1280 octet以上でパケットサイズを自由に決めることができるが、そのパケットサイズは通信先への途中経路に設定された最小サイズ(PMTU)よりも小さくなければならない。途中経路で転送できる最小パケットサイズが変化 している場合、IPv6ではICMPv6によって送信元へ最小パケットサイズを　知らせる。その最小パケットサイズを送信元へ知らせる仕組みをPMTUDと言う。
  - RFC1981 Path MTU Discovery for IP Version 6

F-5.用語集⑤

IPv4アドレス枯渇対応タスクフォース

- IPv4アドレス枯渇対応タスクフォースは、IPv4アドレスの在庫が早ければ2011年にも枯渇するとの危機感を共有するとともに、既に社会基盤として重要な役割りを果たしているインターネットやその上で行われているビジネスに多大な影響を及ぼす可能性があることを認識し、その対策について、インターネットに関わる各プレーヤーが連携・協力して推進するために、総務省を含むテレコム／インターネット関連団体によって発足。
- タスクフォースの活動は、IPv4アドレス枯渇の影響を、ネットワークシステムに係わる様々な立場から検証すると同時に、それらの立場の人たちが対策の必要性を認識して、対策のための情報共有が円滑に行われることを目的としている。
- URL http://www.kokatsu.jp

F-6.用語集⑥

デュアルスタック

- 単一のノードにIPv4とIPv6という仕様の異なるプロトコルスタックを共存させる仕組み。

※ http://www.nic.ad.jp/ja/basics/terms/IPv4_IPv6-Dual-Stack.html

F-7.用語集⑦

リバースプロキシ

- 本来のサーバに代わり、リクエストに応答するプロキシサーバの一種。通常のプロキシサーバは、特定のクライアントに対して代理でコンテンツを取得するのに対して、リバースプロキシは不特定多数のクライアントに対してコンテンツを提供する。
- IPv4/IPv6変換では、IPv6によるクライアントの要求を受付け、本来のサーバにはIPv4で要求する。また、本来のサーバからの応答をIPv4で受けつけ、IPv6でクライアントに転送するという動作を行う。

F-8.用語集⑧

- フォールバック
  - 利用したい機能がなんらかの理由で使えなかった場合に、代替となる機能に切り替えること。
- コンテンツDNSサーバ
  - 自分が管理するドメインの情報(ゾーン情報)について、クライアント若しくはキャッシュサーバからの問い合わせに回答するDNSサーバ。コンテンツDNSサーバだけの機能を提供している場合、自分が管理していない情報について他のネームサーバへの問い合わせを行わない。

F-9.用語集⑨

- キャッシュDNSサーバ
  - クライアントからの問い合わせ要求を受け、該当のドメインを管理するコンテンツサーバに対して再帰的に問い合わせを行い、結果をクライアントに返すDNSサーバ。PC等のクライアントがネットワークに接続する際に設定するDNSサーバにはキャッシュDNSサーバとして機能しているDNSサーバを指定することになる。

F-10.用語集⑩

- レジストリ
  - 「.COM」「.NET」「.ORG」などのgTLD登録ドメイン名のデータベースを維持管理する機関。
- レジストラ
  - 登録者（ユーザ）からドメイン名の登録申請を受け付け、その登録データをレジストリのデータベースに登録する機関。

※ http://www.nic.ad.jp/ja/basics/terms/registry-registrar.html

F-11.用語集⑪

- glueレコード
  - コンテンツサーバが権限の異なるDNSサーバに委任したサブドメインについてのNSレコードを返答する際に追加で必要となる「ネームサーバの名前に対応するIPアドレス」情報。
  - サブドメインを異なるDNSサーバへ委任させている場合、元のドメインのコンテンツDNSサーバに委任先サブドメインの問い合わせがあると、委任先のDNSサーバの名前を返すが、問い合わせ元はこの名前に対しても再度名前解決を行う必要がある。この際に、委任先のDNSサーバの名前が委任先のドメイン名で設定されていた場合に、DNSサーバの名前解決を行うことができなくなってしまう。これを防ぎ、あるドメインとそのサブドメインの情報をつなぐために使われるのがglue(接着剤)レコードである。

Appendix G

Appendix **内における参考**
RFC/Internet Draft

参考RFC / Internet Draft

- Appendix :A
  - RFC 3484 Default Address Selection for Internet Protocol version 6 (IPv6)
  - RFC 4191 Default Router Preferences and More-Specific Routes
  - RFC 3056 Connection of IPv6 Domain via IPv4 Clouds
  - RFC 4380 Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)
  - I-D. IPv4-Mapped Addresses on the Wire Considered Harmful
  - I-D. IPv4-Mapped Address API Considered Harmful
  - RFC 5095 Deprecation of IPv6 Type 0 Routing Headers in IPv6
  - RFC 4862 IPv6 Stateless Address Autoconfiguration
  - RFC 3971 SEcure Neighbor Discovery (SEND)
  - RFC 3775 Mobility Support in IPv6

参考RFC / Internet Draft

- Appendix :B
  - -B(**標準化動向**) RFC 2765 Stateless IP/ICMP Translation Algorithm (SIIT)
  - RFC 2766 Network Address Translation - Protocol Translation (NAT-PT)
  - RFC 3142 An IPv6-to-IPv4 Transport Relay Translator
  - RFC 4966 Reasons to Move the Network Address Translatior - Protocol Translator (NAT-PT) to Historic Status
  - I-D. Framework for IPv4/IPv6 Translation
  - I-D. IP/ICMP Translation Algorithm
  - I-D. NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers
  - I-D. DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers
  - I-D. IPv6-to-IPv4 translation FTP considerations