

最近のIPv6標準化に思うこと

神明達哉

Infoblox, Inc.

(Slide URL: <https://goo.gl/kT6Xfg>)

“ここまできたIPv6の歴史を振り返り、IPv6
技術の動向...等をお伝えします”

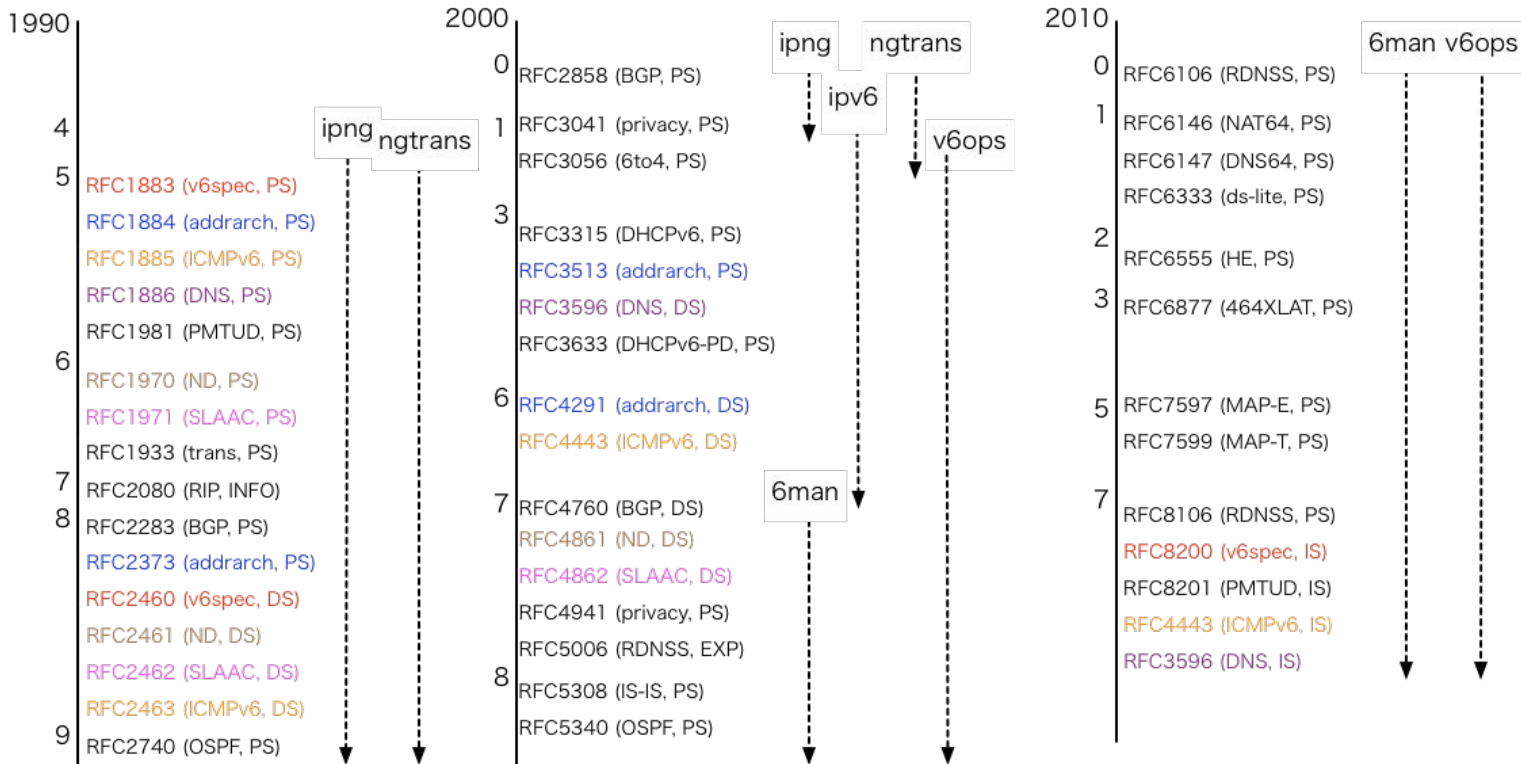
(開催概要より)

内容

- IPv6標準化のここまでと現在
- 最近のIPv6標準化の論点
 - とくに主な議論の対立点について
- 「思うこと」
 - 貢献のススメ

IPv6標準化のここまでと現在

IPv6標準化の歴史



IPv6標準化との関わり(または自己紹介)

- 1995
 - (株)東芝入社、独自のIPv6プロトコルスタック開発
 - 仕様を追いかけるためにIETFの議論に参加
- 1998-2006
 - BSD用IPv6参照実証開発(WIDE/KAMEプロジェクト)
 - 開発者として標準化の議論にフィードバック
 - 4つのRFC発行: 3542(API)、4007(スコープ付きアドレス)、4074(DNS)、4862(アドレス自動設定)
- 2008-2013
 - Internet Systems ConsortiumでDNS開発、おもにDNS関連でIPv6標準化にも緩く参加
- 2013-
 - Infoblox勤務、本業の傍ら「IETF担当」としてオブザーバー的に参加
 - MLの議論、ドキュメントレビュー、たまにf2fミーティング参加

IETFのIPv6関連WG

- IPv6 Maintenance (6man)
 - <https://datatracker.ietf.org/wg/6man/about/>
 - IPv6プロトコルの「保守」
 - 運用上見つかった問題点の修正など
 - 根本的なプロトコルの改変はしない
- IPv6 Operations (v6ops)
 - <https://datatracker.ietf.org/wg/v6ops/about/>
 - IPv6の運用、展開に関するガイドラインを提供
 - 新しいプロトコルの策定は(原則)しない

最近のIPv6標準化のトレンド

- 基本仕様の“Internet Standard”化
- 実運用からのフィードバック
 - Happy Eyeballs v2, Unique Prefix per host
- IPv4枯渇とIPv6-only化を見据えた議論
- 明るい点: 仕様の成熟、実運用に基づく地に足のついた議論
- 暗い点: 意見の対立で議論が滞りがち
 - IETFでは日常茶飯事ではあるが ...
 - Stake holderが増えたことによる副作用か

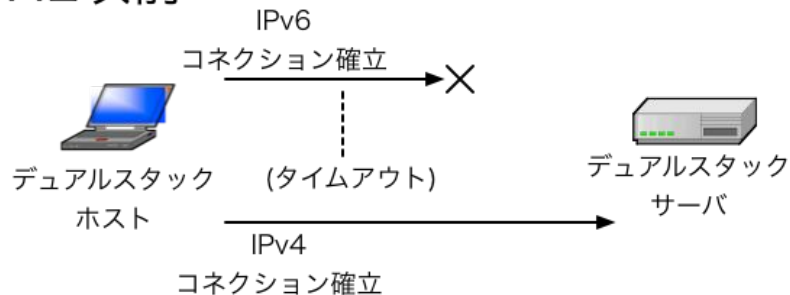
基本仕様のInternet Standard化

- 基本プロトコル
 - RFC2460→draft-ietf-6man-rfc2460bis→RFC8200 (STD86)
- Path MTU探索
 - RFC1981→draft-ietf-6man-rfc1981bis→RFC8201 (STD87)
- ICMPv6
 - RFC4443 (DSからISに格上げ, STD89)
- DNS
 - RFC3596(DSからISに格上げ, STD88)
- アドレスアーキテクチャ
 - RFC4291→draft-ietf-6man-rfc4291bis

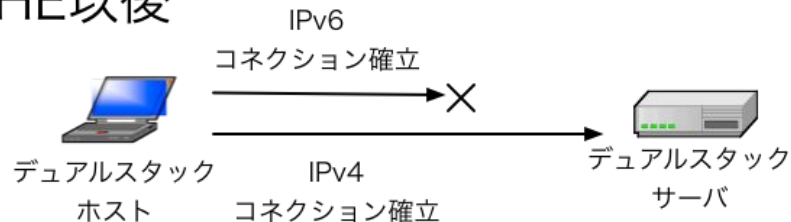
Happy Eyeballs

- 初期バージョン: RFC6555 (2012)
 - IPv6/v4混在環境での接続確立フォールバック問題を改善
 - 複数の接続を並行して確立することで、一方の品質が悪くても引きずられなくなる

HE以前

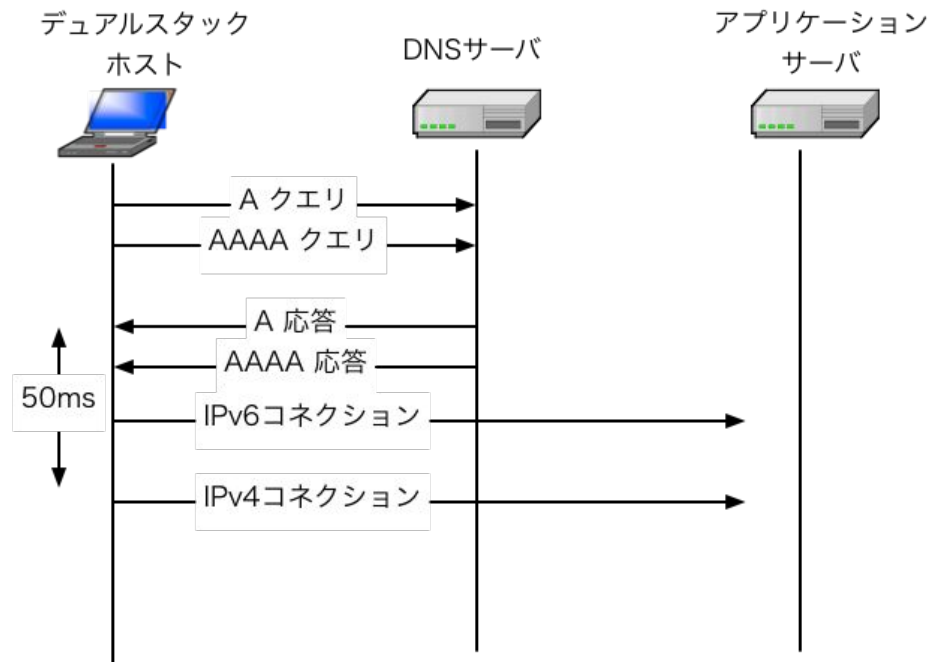


HE以後



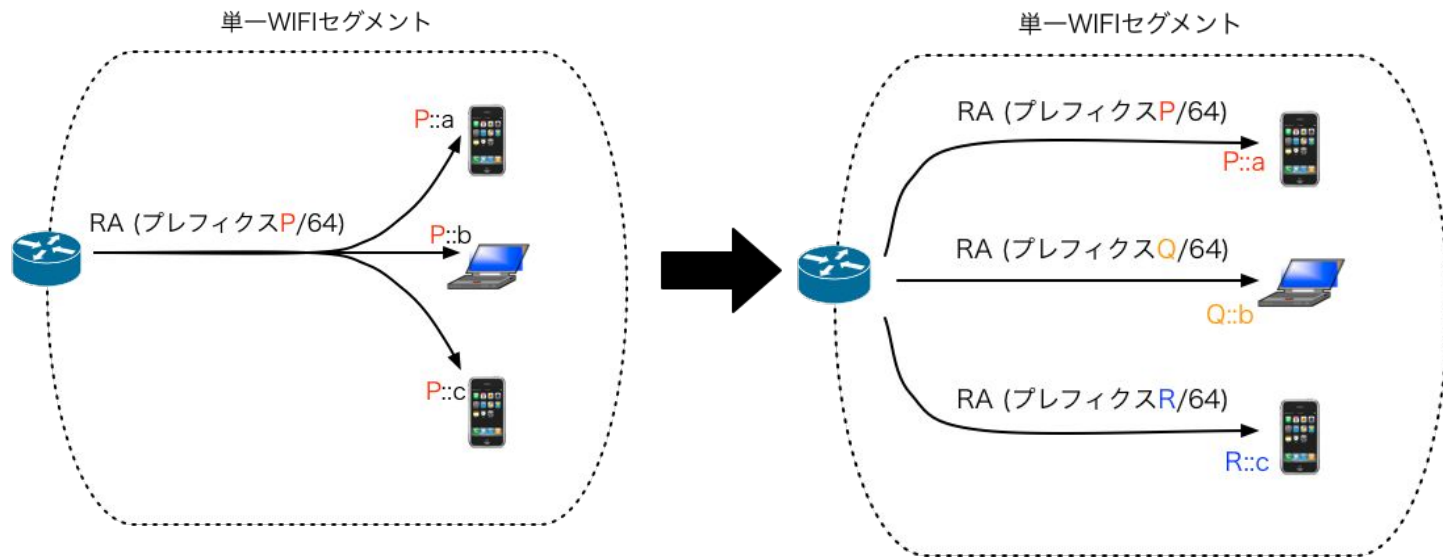
Happy Eyeballs v2

- バージョン2: draft-ietf-v6ops-rfc6555bis (2017)
 - Apple iOSデバイスで取得した統計情報をもとに最適化
 - さらにDNSのAAAA/Aレコード問い合わせにおけるフォールバックも改善
 - AAAAとAを同時に問い合わせ、応答が得られ次第コネクション確立へ
 - Aレコードの応答後には50msの猶予 →AAAA(IPv6)を優先するポリシー



Unique Prefix per Host

- Comcastの公共WiFiネットワークにおける知見を共有
- 各端末に固有のIPv6プレフィクスを与える
 - 異なるプレフィクスを含む Router Advertisement(RA)を実質的にユニキャストで配布
 - On-linkビットを落とすことですべてのトラフィックをルータ経由にする



Unique Prefix per Host

- メリット
 - ルータ側で広大な空間の近隣キャッシュを管理しなくて済む (DoS耐性)
 - ホスト間の直接通信に伴なう種々の問題を回避しやすい
 - 各ホストに事実上無制限のアドレスを与えるという推奨 (RFC7934)にも合致
- 対象端末の広さでRAを採用
 - 一部実装はDHCPv6-PDには対応していない

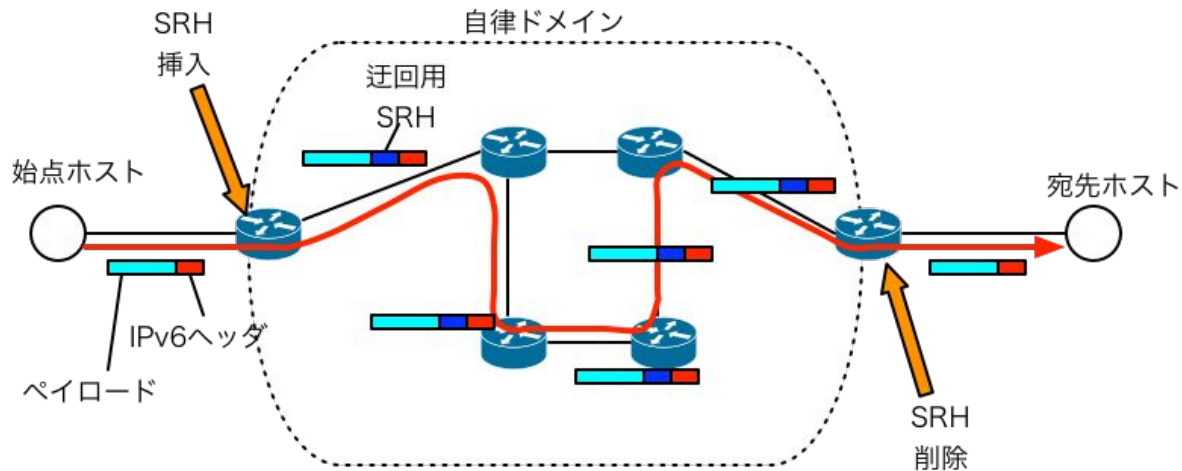
最近のIPv6標準化の論点

ヘッダ挿入問題

- RFC2460
 - “extension headers are not examined or processed by any node along a packet's delivery path”(パケットの配送路上のノードは拡張ヘッダの検査も処理もしない)
- ヘッダの挿入や削除は“処理(process)”に含まれるか？
 - RFCの著者も含めて多くの人は含まれると思っていたが、意外に誤解されていた
 - RFC4782 (hop-by-hop optionの削除を許している、ただし experimentalで普及もしていない)
 - Segment routingの一部実装

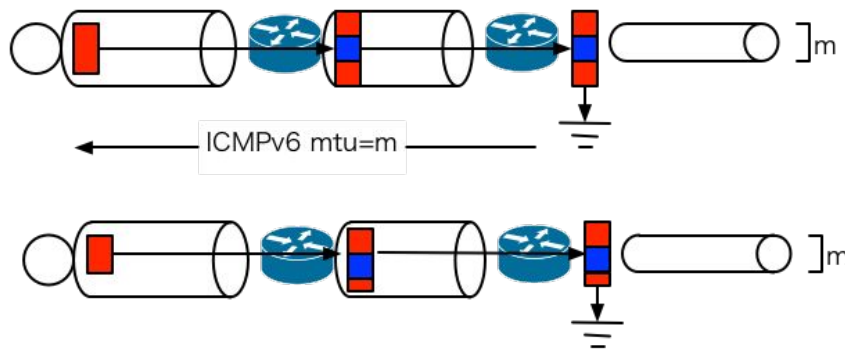
ヘッダ挿入問題(続き)

- Segment Routing (for IPv6)
 - 自律ドメイン内でIPv6ルーティングヘッダ (SRH)を用いた経路カスタマイズ
 - トラフィックエンジニアリング、SLA、ロードバランスなど
 - 公式仕様: draft-ietf-6man-segment-routing-header カプセル化ベース(挿入なし)
 - 実際には途中ノードでSRHを挿入・削除する実装が先行



ヘッダ挿入問題: rfc2460bisの議論

- 当初の提案
 - 本来の意図が明確になるように、挿入・削除が起きないことを明記
 - Path MTU探索(右図)や認証ヘッダで問題が起こるのは明らか
 - 標準仕様は一般のグローバルインターネットを想定すべき
 - 「制御されたドメイン」からでもパケットは漏れるもの
- 挿入支持派からの反論
 - 自律ドメイン内でのオペレーションは柔軟であってよい
 - 挿入・削除をするSRv6は実装も運用実績もあって問題も起きていない
 - 時代遅れの設計思想で新しいイノベーションを阻害すべきでない

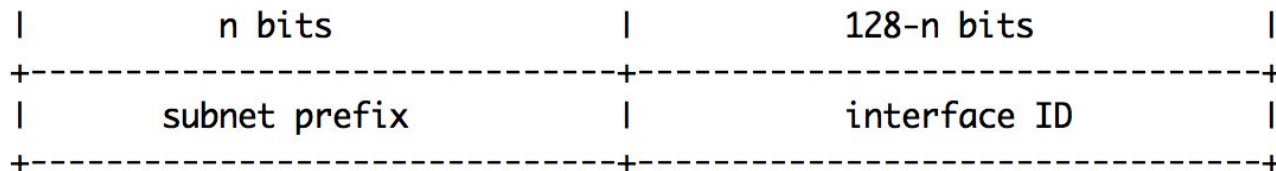


ヘッダ挿入問題: 現状

- RFC8200での決着
“Extension headers ...are not processed, **inserted**, or **deleted** by any node along a packet's delivery path” (パケットの配送路上のノードは拡張ヘッダの処理も**挿入**も**削除**もしない)
- SRv6のような特殊ケースについては別途updateを議論
 - draft-voyer-6man-extension-header-insertion

サブネット/インタフェースID長問題

- サブネットプレフィクスとインタフェースID(IFID)



- RFC4291
“For all unicast addresses, except those that start with the binary value 000, Interface IDs are required to be 64 bits long” (ビット000ではじまるものを除き、すべてのユニキャストアドレスのインタフェースIDは64ビット長)

サブネット/IFID長問題: rfc4291bisでの議論

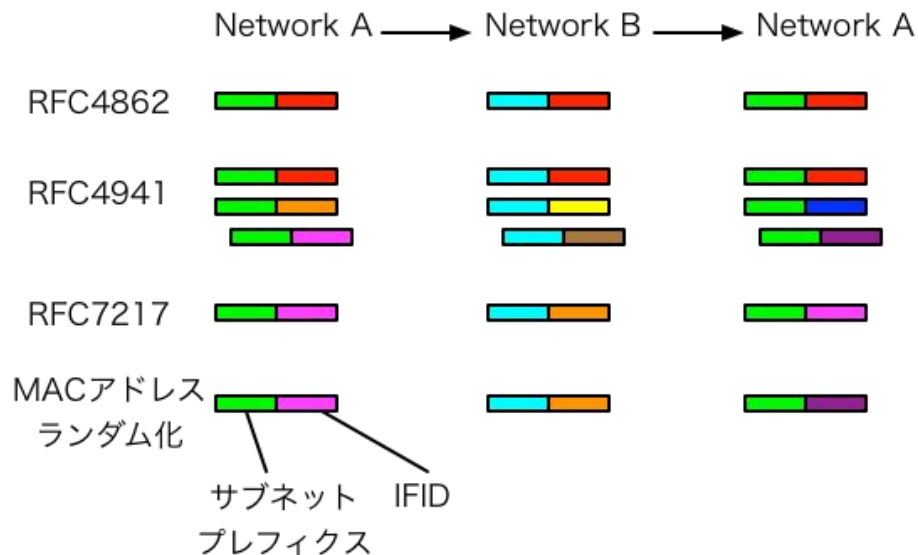
- おもに手動/DHCPv6でアドレス設定するオペレーターからの不満
 - draft-bourbaki-6man-classless-ipv6
 - サブネット長のような概念はアーキテクチャの一部ではなく運用者が決められるべき
 - 現に例外がある: RFC6164(p2pリンクでの/127)、`ifconfig`や`ip`コマンドの実装
 - 広すぎる空間はムダだトラブルの原因にもなる
- 固定長支持派による反論
 - ホスト実装が簡単になる/すでに64ビットを仮定した応用がある
 - プライバシー確保などのために大きな IFID空間が必要
 - 一度可変長にすればなし崩し的にサブネット長が伸びて最後は NATすることになる
 - IPv4時代の運用常識に囚われていては IPv6の革新を享受できない
- 議論膠着、rfc4291bisは実質棚上げ状態

サブネット/IFID長問題: 混乱しがちな詳細

- On-link判定のプレフィクスは初期からずっと可変長(RFC4861)
 - しばしば誤解されていて、混乱のもとになっている
 - `ifconfig addr/96` は128ビットのアドレスと96ビットのon-linkプレフィクスの手動設定と考えればRFCの記述とは矛盾しない
 - それでもモヤモヤは残る: 下位32ビットはIFID?
 - YESなら現在のRFCの記述と矛盾
 - NOなら下位32ビットは何なのかという話になる
- 複数の標準がIFID長に言及、辛うじて整合性が保たれている
 - アドレスアーキテクチャ: 「アドレス空間の一部を除いて64ビット」
 - ステートレス自動アドレス設定: 「リンク層ごとに規定」
 - リンク層の仕様(RFC2464など): 現状すべて64ビット、だが違う値の仕様が出ない保証はない

アドレスプライバシー拡張方式の変遷

- 初期のアドレス設定方式におけるプライバシー問題
 - IFIDが固定のMACアドレス由来のため容易に追跡できる
- 一時アドレスの導入 (RFC4941)
 - 固定IDから生成したランダムIDを定期的に変更して利用
 - 追跡は防止できるが永続的アプリに不向き
- Stable Privacyアドレス(RFC7217)
 - サブネット固有のランダムID
 - 移動による追跡を防止しつつ移動しない間の安定性を提供
- MACアドレスランダム化の利用
 - 移動後に戻ってきた場合の追跡も防止
 - レイヤ境界違反だとして反対する声もあり、揉めている



ホスト自動設定問題

- DNSサーバアドレスを配布する二つの方式
 - DHCPv6(RFC3315,3736)とRouter Advertisement(RDNSS, RFC8106)
 - 一方しかサポートしない実装がある：AndroidはRDNSSのみ、ほとんどのWindowsはDHCPv6のみ
 - 現状では実質的に両方を提供する必要がある
- RDNSSの実装を“MUST”とする提案@v6ops
 - 2017年3月IETF会合の議論を受けてMLで協議
 - 典型的な“DHCPv6 vs RA”の議論になって収集つかず
 - ただしWindows 10のRDNSSサポートによりデファクト的に解決するかも

DHCPv6 vs RA

- RA派の主張
 - RAによるアドレス設定は広く普及済み、それを応用するのが現実的
 - アドレスやプレフィクスは経路情報を持つルータが配布するのが自然
 - DHCPv6では設定の急な変更に対応できない (Reconfigureはジョーク)
 - RAによる自動化はIPv6の主要なイノベーション。IPv4時代の発想から脱却すべき
- DHCPv6派の主張
 - マルチキャストを多用するRAはネットワークに負荷をかけやすい
 - DHCPv4の運用には何も問題はなかった。新しいプロトコルは運用の重荷なだけ
 - ホストが自律的にアドレスを設定して勝手に通信されるのは嫌
 - RA推しは一部ベンダの陰謀である
- この対立で多くの時間が浪費されている
 - DNSサーバ問題(前述)
 - 出ては消えるDHCPv6デフォルトルートオプションの提案

最近のIPv6標準化に思うこと

IPv6標準化の今後: 大胆予測(?)

- 基本仕様のInternet Standard化
 - 象徴程度の意味しかないが、仕様の整理の機会でもある
 - アドレスアーキテクチャの議論の決着
 - Neighbor Discovery、SLAAC、経路制御など
- 運用を通じたフィードバック
- v6-only network関連
 - IPv4-as-a-service
 - 完全な“sunset IPv4”に向けたプロトコル・運用上の標準化
 - NAT64・DNS64の適用可能性と是非

紛糾する議論についての見解

- 立場の違いによる対立で議論が滞りがち
 - オペレータ:「IPv6流」の押し付けを嫌う
 - ホスト実装者: エンドホストに与えられる自由度を重視
 - プロトコルアーキテクト: 安易な例外を嫌がる一方、IPv6流の方法には寛容
 - それぞれ一理あり、やってみないとわからないので決定打もない
 - 多くの真剣な利害関係者が出てきたという意味では歓迎すべきことかも
- “Internet Standard”化の頓挫はある意味瑣末
- IETFではよくある話でもある
 - “Rough consensus”ベースなので強引な解決を取れない
 - 個人の意見が尊重される代わりに誰でも blockerになれる
 - Big player間の思惑も無視できず

貢献しましょう！

- 問題に気がついたり解を見つけたりしたら共有
 - 本格的な普及がはじまったいまがチャンス (unique prefixやhappy eyeballsの例)
- ドラフトのレビューやラストコールへの賛否も立派な貢献
 - 組織の後ろ盾がない個人の意見でも採用され得る (ヘッダ挿入の例)
 - 投票がないとは言っても “rough consensus” 作りには多数決の要素もある → 意見を言うことが重要
 - 不毛な対立の解消につながられれば大きな成果
- 老婆心からのtips
 - 賛成できなくても尊重はする → 多くの場合、絶対的に正しい/間違いということはない
 - 膠着したらこだわらずに前に進む → 議論に勝つことが目的ではない
 - 歴史的経緯にも学ぶ → 何度も持ち出された話題を蒸し返しても多くの場合時間のムダ
 - 設計思想を尊重しつつ現実的になる