

Host address availability recommendations
draft-ietf-v6ops-host-addr-availability-0607

IETF Internet Draft読み込みworkshop

2016/06/22 (水) 18:30 - 20:30

西塚要 / @__kaname__

Host address availability recommendations

- WG: v6ops (IPv6 Operations)
- 著者:
 - Google V. Cerf/ L. Colitti
 - Apple S. Cheshire/ D. Schinazi
- Intended Status: BCP (Best Current Practice)
- <https://tools.ietf.org/html/draft-ietf-v6ops-host-addr-availability-06>
- WG draftへのadoptがとても早かった

アブストラクト

This document recommends that networks provide general-purpose end hosts with multiple global IPv6 addresses when they attach, and describes the benefits of and the options for doing so.

- general-purpose end hosts
 - 汎用的な(=様々なアプリケーションが載せられる)端末等に対し、
- multiple global IPv6 addresses
 - 複数のグローバルIPv6アドレスを、
- networks provide
 - ネットワークが付与することを推奨(recommend)する。

⇒その利点と方法を記載する。

1. イントロダクション

- IPv4のベストプラクティスとIPv6のベストプラクティスは若干異なる。
- その違いの一つが”アドレッシング”
 - IPv4: ホストあたり1IPアドレス
 - IPv6: 同一リンク上のアドレスの数が/64空間(2^{64} 個)
- 複数アドレスを持つことのメリット
 1. アプリケーションの機能性向上
 2. シンプル
 3. プライバシーの問題
 4. 将来の拡張性
 5. NATが無いこと

2. 一般的なIPv6のアドレス

- IPv6はそもそも複数アドレスをサポートするように設計されている[RFC4291]

IFにSLAACまたはDHCPv6で付与されるアドレス

Link Local Address		
IPv6 unicast addresses from IID	EUI-64 address[RFC4291] Opaque Interface Identifiers[RFC7217]	SLAAC
Privacy Address	[RFC4941]	SLAAC
Temporary or Non-temporary Address	[RFC3315]	DHCPv6

- 一般的なNW: 3GPP, Ethernet, Wi-Fi
 - SLAACを使っていれば、ネットワークに明示的にリクエストすることなく、IPv6アドレスの追加が可能

3. 複数アドレスのメリット

- 現状で2個目以上のアドレスを使う例：
 1. プライバシーアドレス
 2. 同一デバイス内での複数プロセッサがそれぞれ通信する
 - アプリケーションプロセッサ
 - ベースバンドプロセッサ
 3. テザリング
 4. ホスト上のVM
 5. 464XLATのようなIPv4枯渇対策技術
 - 変換されたIPv6prefixか、nativeかを区別するため(in方向)
 6. Identifier-locator addressing[I-D.herbert-nvo3-ila]
 7. Per-application IPv6 addresses[TARP]

3. 複数アドレスのメリット

- ホストからアドレスを追加する例:

1. 464XLAT[RFC6877]

- PLATをリモートで提供する場合:

<http://www.jpix.ad.jp/en/service/ipv6v4.html>

- Source IPv6 アドレスを、ISPから提供された空間から生成する

2. /64 sharing[RFC7278]

- テザリング配下のLANに3GPPから得られた/64 ネットワークを拡張する
- DHCPv6 PDが**必要**ない

4. アドレス数を制限することのデメリット

- ホストにアドレス数個しか与えない場合：
 - アドレスの追加に人力のオペレーションが入り、時間がかかる
 - 運用上の複雑さが増える
- そのようにする理由：
 - IPアドレス単位で課金したい場合が考えられる
- アドレスにチャージすると、結局人々は回避するためにNATを導入するので、収入が増えないだけでなく運用が複雑になるだけ

5. NATによる制限

- NATは欠点のある方法：
 - NAT越えのためにアプリケーションが複雑になる
 - モバイルバッテリーのもちが悪くなる
 - 少なくとも30秒ごとにキープアライブが必要
 - <http://www.ietf.org/proceedings/88/slides/slides-88-tsvarea-10.pdf>
 - ちなみにQUICは15秒おきにキープアライブする
- IPv6ではNATをすべきではない
 - IAB Thoughts on IPv6 Network Address Translation [RFC5902]

5. NATによる制限

- NAT手法
 - NAT66 = Stateful (Address+Port)
 - NPTv6 = Stateless (Address only)
- 現実にはNAT66を実装してしまっている
 - Linux:
 - http://kernelnewbies.org/Linux_3.7#head-103e14959eeb974bbd4e862df8afe7c118ba2beb
 - VMware:
 - <https://communities.vmware.com/docs/DOC-29954>

Indeed, it could be argued that the main reason for deploying IPv6, instead of continuing to scale the Internet using only IPv4 and large-scale NAT44, is because doing so can provide all the hosts on the planet with end-to-end connectivity that is constrained not by accidental technical limitations, but only by intentional security policies.

6. 複数アドレスの提供方法

- SLAAC: Stateless Address Autoconfiguration [RFC4862]
 1. 複数ホストが/64セグメントを共用した状況で使う方法
 2. /64をホストごとに付与する方法
 - 3GPPのようなPoint-to-point link で可能
- DHCPv6: [RFC3315]
 1. 多くのOSでは非一時アドレスの要求は1つだけ
 - プロトコル的には複数要求してもよい
 2. 異なるDUIDを使う方法
- DHCPv6-PD: DHCPv6 prefix delegation [RFC3633]
 1. /64より大きければ、分割して配る方法
 2. /64であれば、L2 bridging, ND proxyingまたは/64 sharingを使う方法

7. 必要なアドレス数

- デバイスに対して20個くらい必要と見積もる
 - プライバシーアドレス: 8個くらい/1日
 - VM用アドレス: 数個
 - 464XLAT用アドレス: それぞれ別に必要
 - モバイル端末: ベースバンドチップ用1個

8. 推奨対応

1. IPv6アドレスの個数に上限を設けるべきではない
2. とくに、ホストあたり1つというのは推奨できない
3. 新しいアドレスを、ステートレスに増やせるべき
 - SLAACによる方法
 - /64 prefixをホストごとに割り当てる方法
 - DHCPv6-PD
 - SLAAC with per-device link
 - ステートフルな方法(DHCPv6 IA_NA or IA_TA)はお勧めできない

9. 運用上の考慮事項

1. ホストのトラッキングについて:

- ルータのネイバーテーブルを見ればよい
- /64単位のSLAACや DHCPv6-PDの場合は、容易にトラックが可能
- DHCPv6を使って、DHCPv6サーバのログでトラックすることはお勧めできない

9. 運用上の考慮事項

2. アドレス空間の管理について:

- 現状のRIRポリシーであれば、ホストに/64単位を割り振ったとしても問題ない
 - /48あれば、192.168.0.0/16 に相当
 - /40あれば、10.0.0.0/8 に相当

まとめ

- -00 draftよりもかなり論点が纏まってきた
 - -00 draft当初は、「なんとなく重要なことが書いてあるっぽいけど」ということで、ML上で沢山議論された
- プロトコルやホストの挙動のスペックを変えるものではないので、運用上のベストプラクティスをまとめる、というところに落ち着いた
 - DHCPv6-PDが良い、というように読めたがニュートラルなものに書き換えられた
- 結論としては、以下の2点：
 - NATするな
 - /128割り当てするな