中間まとめ

IPv6 とセキュリティ

2014 年 7 月 31 日

サイエンティフィック・システム研究会

移行期にあるネットワークサービスのセキュリティ WG

活動メンバー

			氏名	機関/所属(2014 年 7 月 31 日現在)
会員	担当幹事		西村　浩二	広島大学
	推進委員	まとめ役	只木　進一	佐賀大学
			吉田　和幸	大分大学
			笠原　義晃	九州大学
			武蔵　泰雄	熊本大学
			鈴木　聡	高エネルギー加速器研究機構
			長谷川　明生	中京大学
			鈴木　常彦	中京大学
			藤本　衡	東京電機大学
			西村　浩二	広島大学
			藤村　丞	福岡大学
			山守　一徳	三重大学
賛助会員 (富士通)	推進委員	まとめ役	吉田　真和	富士通(株)
			須永　知之	富士通(株)
			山下　眞一郎	富士通(株)
			山路　光昭	富士通(株)
			石川　武志	富士通(株)
			田口　雅晴	(株)富士通九州システムズ

目次

1　はじめに 1
2　IPv6 の基本と Windows での対応 3
3　IPv6 の疎通状況を端末上で確認する方法 4
4　トンネルの危険性とフィルタの方法 8
4.1　意図せず張られるトンネル 8
4.2　6to4 が有効になっているデバイスからの RA 送出 8
4.3　IPv4 がプライベートアドレスの場合もトンネルが張られるのか 9
4.4　ファイアウォール等の設定の留意点 9
5　IPv6 の今後の方向性 11
5.1　DNS 混ぜるな危険：IPv6 には移行できない 11
5.1.1 DNS 権威サーバ 11
5.1.2 DNS キャッシュサーバ 11
5.2　IPv6 と IPv4 は並行運用するしかない? 12
5.3　並行運用する意義はあるか 12
5.4　並行運用するとすればどういう形態が望ましいか 12
5.4.1 サーバ側 12
5.4.2 クライアント側 13
6　ファイアウォール、キャッシュ DNS サーバによる通信の制御について-IPv4 と IPv6 との比較- 15
6.1　はじめに 15
6.2　サンプルポリシー 15
6.3　キャッシュ DNS (bind9.5 以降) の設定 19
6.3.1 IPv6 over IPv4 トンネルの抑制 19
6.3.2 無駄な問い合わせの抑制 20
参考文献 22
添付資料 SA46T:IPv4 アドレス枯渇後の IPv6 移行と IPv4 継続利用を両立するカプセル化技術 24
松平直樹 (富士通株式会社)

IPv6とセキュリティ

1　はじめに

インターネットの標準として使われてきたIPv4のアドレスの枯渇が指摘され始めたのは1990年代前半でした。2011年には、アジア太平洋地域には、新規アドレス領域の割り当てが不可能となり、IPv4アドレスの枯渇が現実のものとなりました[1]。日本国内の大きな組織がクラスBやCのネットワークを有している一方、インターネットに後から参加することとなった国々では、大きな組織であっても利用できるIPv4アドレスがほとんどないというのも事実です。それら国々の組織のなかには、多段になったNAT配下のネットワークを利用している場合もあります。

IPv4アドレスの次のアドレス体系としてIPv6が提案されたのは1995年です[2]。日本ではKAME Projectなど、IPv6の実験、普及活動が活発に行われました[3]。その後、主要なOSベンダーがIPv6対応を進めた結果、現在使われているサーバやクライアントのOS、さらにrouterやfirewallなどのネットワーク機器の多くがIPv6に対応しています。インターネットを流れるデータも、IPv6対応のものが一定量あります。また、Internet Societyでは、2011年からIPv6普及の世界規模の催しを実施しています[4]。

しかし、周囲を見渡すとIPv6の影は薄いのが現状です。多くのインターネットサービスがIPv4環境対応のため、IPv6だけの環境は運用レベルでは存在しません。移行環境として想定されたIPv4/IPv6デュアルスタックは運用レベルで存在しますが、管理コストが大きく課題を抱えています。多くの組織ネットワーク、家庭などのプロバイダはIPv4サービスが主体です。さらに、IPv6ならではのサービスも見当たりません。つまり、IPv4アドレス枯渇を受けたIPv6への道筋が見えていません。

前述のように、日本の一定以上の規模の組織の多くはIPv4のアドレスを保有しています。また、プロバイダへの接続や、組織内の小グループであれば、IPv4プライベートアドレスを使うことができます。セキュリティの観点から、IPv4プライベートネットワークが推奨される場合もあります。IPv4アドレスの枯渇はそれほど深刻ではなく、IPv6はなくても大丈夫に思えます。

一方、前述のように、普段から使用しているOSはすでにIPv6に対応しており、知らぬ間にIPv6アドレスがついていることがあります。その結果、知らぬ間に、IPv6を介して内部のデータが流出する、あるいは外部から攻撃される恐れがあります。本WGでは、このような現状の下で、セキュリティの観点から、IPv6との付き合い方を考えることとしました。

現在使用されている主要OSは、IPv6対応となっています[4]。ネットワーク管理者がIPv6サービスを提供すればいつでも利用できます。セキュリティの観点から、管理者がIPv4しか提供していない環境の下で、デスクトップOSが自らIPv6用トンネルを作成して通信を行う可能性に着目しました。つまり、ネットワーク管理者がIPv6サービスを提供しなくても、IPv6通信が行われている可能性です。さらに、そのようなクライアントが、RA (Router Advertisement) を流すことを通じて、他のコンピュータも含め、組織外とのIPv6通信路を開放してしまう恐れがあります。

前述のように、組織としてはIPv4しかサービスしていないと考えていても、内部のクライアントがIPv6通信を、トンネルを介して行う可能性があります。この場合、IPv4ファイアウォールによるネットワーク防御機能を素通りしてしまう恐れがあります。そのため、Ipv4しか運用していないネットワークの管理者であっても、IPv6通信のトンネルに対する防御が必要となります。組織としてIPv6を提供していても、内部に潜むIPv4プライベートネットワークから同様のトンネル通信が発生する可能性もあります。予期しないIPv6通信への対策方法を検討しました。

しかし、IPv4アドレスの枯渇は事実であり、IPv6サービスは増えていくでしょう。実際、GoogleやFacebookなどクラウドサービス事業者はIPv6サービスを行っており、IPv6通信可能なクライアントはIPv6通信を行っているはずです。ではどうすればよいのでしょうか。すぐに、全てがIPv6化するということはあり得ませんが、IPv4に対応し続ける必要があるのは、サーバ類に限られるのではないでしょうか。これまでの、単純な「デュアルスタック化」ではなく、サーバ類とクライアント類を分けて考える必要があると思います。一つの方法として、クライアントネットワークのIPv6シングルスタックへの移行があり得ます。

2　IPv6 の基本と Windows での対応

現在使用されているインターネットのアドレス体系は IPv4 と呼ばれるものです。アドレスを 32bit で表し、8bit (octet) 毎にピリオドで区切った表記、通常は 10 進表記を用います。たとえば、`www.google.co.jp` の IP アドレスは `173.194.126.223` となっています。32bit アドレスであることから、アドレス数の上限は 2^{32} 個、つまり約 42 億個となります。地球上の全人口よりも少なく、一人一つのアドレスさえ持つことができません。インターネットの普及に伴って枯渇するのも道理です。

そこで、次世代のアドレス体系として提案されたのが IPv6 です。IPv6 では、アドレスを 128bit で表します。つまりアドレス数の上限は 2^{128} 個、約 340 澗個 (澗は 10^{36}) です。1 モルあたりの分子数を表す Avogadro 数 ($6.022 \times 10^{23}\mathrm{mol}^{-1}$) よりも圧倒的に大きな数であり、十分に大きな数であることが分かります。IPv6 アドレスは、16bit 毎 (field) にコロンで区切った 16 進表記で表します。たとえば、`www.google.co.jp` の IPv6 アドレスは

```
2404:6800:400a:802::101f
```

となっています。各区切りの先行する 0 は省略し、連続する複数の field が 0 の場合、`::` で表記します。アドレス表記方法については rfc5952 を参照してください [5]。

IPv6 アドレスは非常に長く、利用者が正しく個々のクライアントの IP アドレスやネットマスク、ルータのアドレス等を設定することは困難です。そこで、IPv6 では、特にクライアント PC に対してはアドレス自動設定が標準となっています。ルータは、RA (Router Advertisement) を送信して、クライアントの自動設定を補助します。

多くの利用者が用いている Windows では、Windows XP 以降で IPv6 が利用可能となっています。Windows Vista 以降では、初期設定で IPv6 が有効となっています。接続したネットワークに RA が流れている場合には、素直に IPv6 アドレスを自動設定します。問題は、RA が流れていない IPv4 ネットワークに接続した際にも、後述のように IPv6 通信を試みる点です。

3　IPv6 の疎通状況を端末上で確認する方法

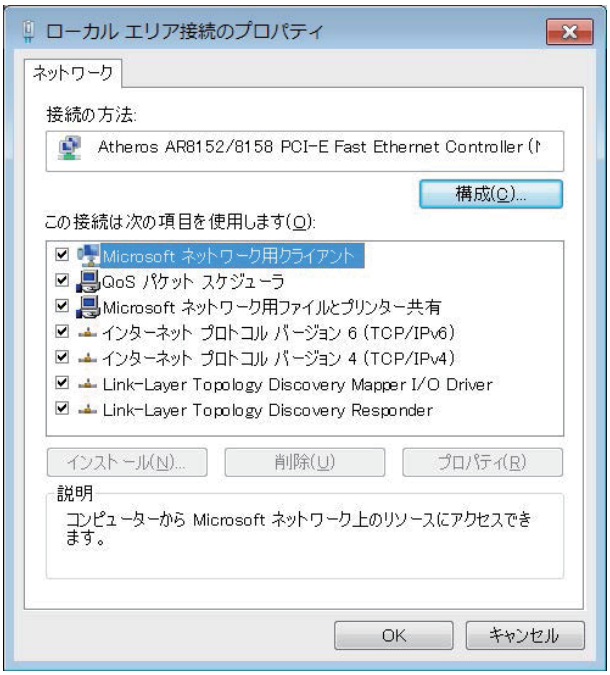

図 1: インターフェース毎に IPv6 が有効かどうかをコントロールパネルから確認する (Windows 7 の例)。

前述のように、IPv6 プロトコルは、Windows XP SP1 以降に搭載されています。Windows XP SP1 では、初期設定で無効、Windows Vista 以降では初期設定で有効です。この初期設定を確認する、あるいは変更するには、

コントロールパネル\ネットワークとインターネット\ネットワーク接続

を開きます。この中で、接続に用いているアダプター (例えば、ローカルエリア接続) を選び、プロパティの画面 (図 1) の中の、「インターネットプロトコルバージョン 6 (TCP/IPv6)」の左側のチェックボックスのチェックを操作します。チェックが入っていれば有効になっています。

IPv6 プロトコルでネットワーク接続しているかを直接に調べる最も簡単な方法は、適当な WEB ブラウザを用いて

`http://www.ipv6-test.com/`

にアクセスしてみることです。図 2 のように、接続性をチェックした結果が表示されます。この結果は、用いる WEB ブラウザによっても異なります。左側の IPv6 connectivity を見ると 6to4 を使っての接続可能性があることを示していますが、右側の Browser の欄を見ると、このブラウザはデフォルトでは IPv4 を使って接続しようとしていることを示しています。default の下の Fallback は、接続失敗時の次の手として何を使うかを示しています。IPv6 で全く通信できない場合は、図 3 のように表示されます。

このように接続性をチェックできるサイトは、他にもいろいろありますが、有名なサイトとして、「The KAME project」が運営するサイト

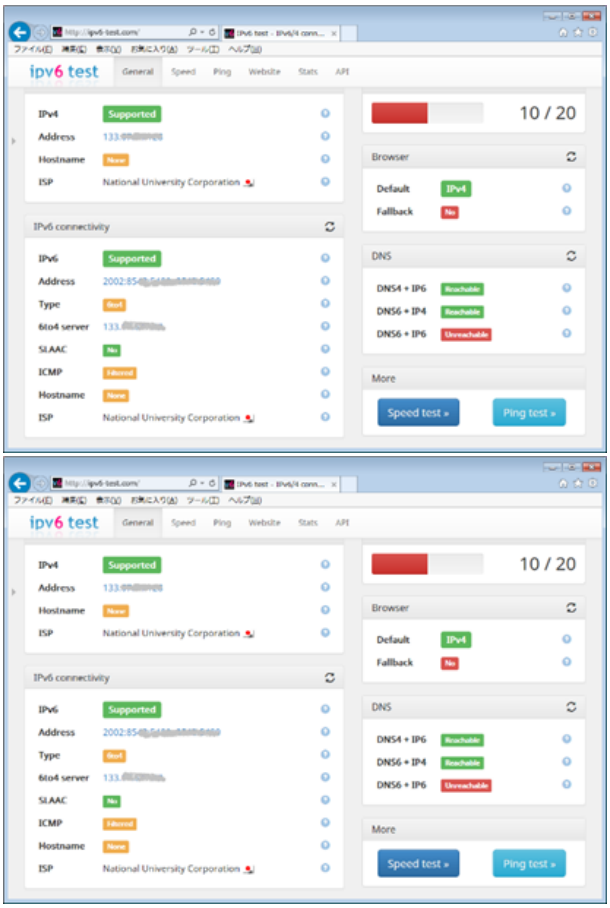

図 2: `http://www.ipv6-test.com/`に接続した結果、上は Internet Explorer、下は Firefox での接続結果。

`http://orange.kame.net/`

があります [3]。このサイトでは、IPv6 プロトコルでアクセスすると踊っている亀の絵がきれいな絵になって表示されます。しかし、IPv4 プロトコルでアクセスすると、踊っていない静止画の亀の絵が表示され、その右側にある「the dancing kame」をクリックした時に、モザイクの亀の絵が踊って表示されます。

ネットワーク管理者が注意しなければいけないのは、管理者として IPv6 グローバルアドレス配布を行っていない場合でも、クライアントが 6to4 や Teredo と呼ばれる方式で IPv6 の接続性を保持している可能性があることです。さらに、Windows 8 では、IPv6 を使って通信ができるかの確認を 30 日ごとに OS が自動で行っています。クライアントは、IPv6 通信可能であるとわかると、それ以降は IPv4 よりも IPv6 を優先して利用しようとします。ただし、WEB ブラウザは、OS の取り決めた優先順位とは独立して挙動しています。なお、IPv6 グローバルアドレスを提供している場合には、Teredo や 6to4 は起動しません。

コマンドを使って、OS レベルでの IPv6 に対する挙動を確認してみましょう。Windows のアプリのコマンドプロンプトを「管理者として実行」で動かします (コマンドプロンプトのアイコンを右クリックして「管理者として実行」を選択します)。ここで、

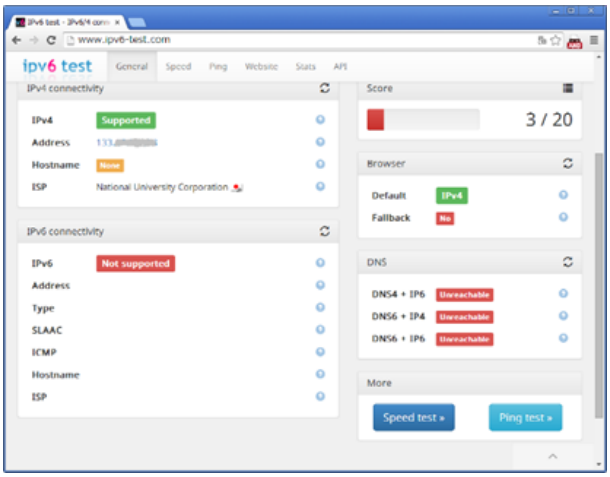

図 3: `http://www.ipv6-test.com/`に Chrome で IPv6 接続できなかった場合の例。

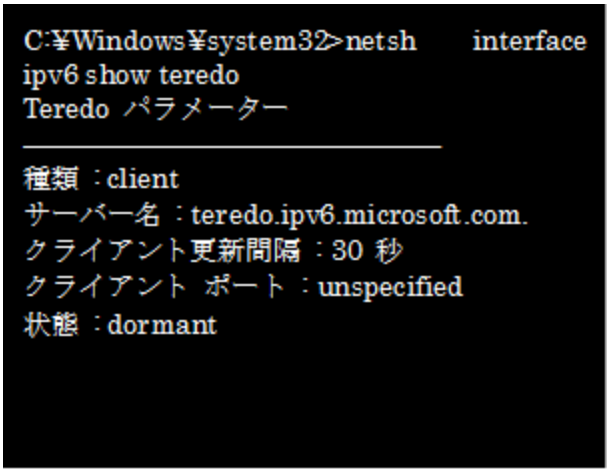

図 4: teredo の状態の確認

```
netsh interface ipv6 show teredo
```

コマンド入力すると、状態が表示されます。この時、`qualified` と表示されれば、Teredo による IPv6 トンネルが有効となっています。`qualified` でない場合の例としては、図 4 のように表示されます。

つぎに、IPv6 プロトコルがどのような優先順位で動いているのかを調べましょう。

```
netsh interface ipv6 show prefixpolicies
```

を実行すると、図 5 のように表示されます。これは、OS が判断している挙動の仕方を示しています。前述したように WEB ブラウザごとに独自の優先順位を持っていますので、この優先順位に従ってプロトコル選択されているとは限りません。

上の表示の最後の行の Teredo が、Teredo サーバ (デフォルトは Microsoft が運営しているサーバです) へ IPv4 パケットに作ったトンネル使ってアクセスし、自分のパソコンの IPv6 アドレスを

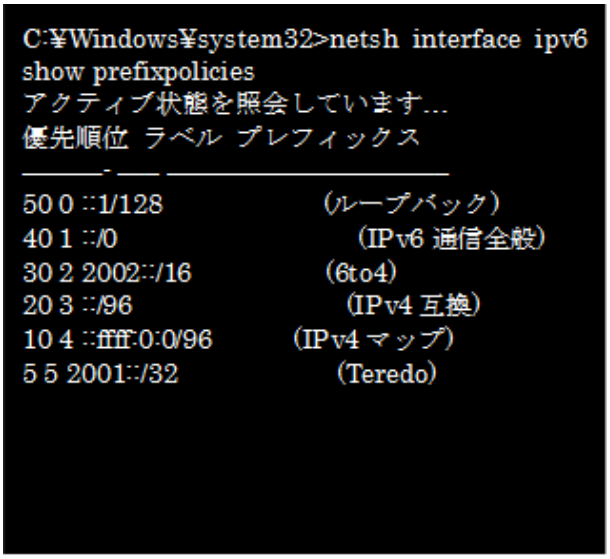

図 5: IPv6 プロトコルの優先順位の確認例

自動的に取得してくる働きを示しています。

接続相手の IPv6 アドレスを持つサーバへアクセスしに行く時には、IPv6 アドレスを包み隠して、IPv4 パケットを使って、Teredo サーバにアクセスに行き、Teredo サーバから最終到達先の接続相手までは、IPv6 プロトコルでアクセスに行く動きをします。自分のパソコンが接続するネットワークのルータが IPv6 パケットを遮断している場合でも Teredo サーバまでは IPv4 パケットで通過していくので、「踊る亀」を見ることができることになります。

実際のアクセスは、若干複雑です。

- DNS サーバから名前解決の IPv4 アドレスである A レコードを取得できる場合には IPv4 プロトコルを使ってアクセスするのが優先になる場合があります。
- プロトコル選択の挙動が OS や WEB ブラウザで異なるため、例えば、「The KAME Project」と呼ばれるサイトへアクセスする時に、

  `http://orange.kame.net/`

  の URL でアクセスするとモザイクの亀が見えるのに、

  `http://[2001:200:dff:fff1:216:3eff:feb1:44d7]/`

  という IPv6 の IP アドレスを直接指定してアクセスすると、モザイクでないきれいな亀が見えることがあります。(ブラケットで IPv6 アドレスを囲む表記はすべてのブラウザで使える訳ではありません。)

4　トンネルの危険性とフィルタの方法

4.1　意図せず張られるトンネル

前述のように、Windows 端末では IPv6 グローバルアドレスを配布していないネットワークに接続されると自動的に IPv6 のトンネルを開通させようとすることがあります。IPv4 グローバルアドレスが配布されている場合は 6to4 と Teredo が、IPv4 プライベートアドレスが配布されている場合は Teredo のみが試されます。こうしたトンネルは、WindowsXP では自動的には張られませんが、Windows Vista 以降では自動的に張られます。6to4 の場合、このアドレスは

```
2002:XXXX:YYYY::ZZZZ:ZZZZ:ZZZZ
```

のような prefix になります。ここで、`XXXX` と `YYYY` はその機器の IPv4 アドレスを 16 進表記したものになります。すなわち 6to4 のアドレスを見ると誰がトンネルを張っているのかわります。一方、Teredo の場合は IPv4 アドレスが埋め込まれた形にはなりません。

通信したいサーバが IPv4 および IPv6 のどちらでもアクセス可能な場合はこれらの自動トンネルを通った通信は、本来は行われません。しかしアプリケーション毎に実装が異なるため、いつでも IPv4 を通るわけではありません。また、IPv4 ネットワークにおいて、IPv6 でしか接続できないサービスに対しては、このトンネルを通った通信が発生します。

通常、ファイアウォールはこのトンネルの中まで解析して通信内容をフィルタすることはありません。そのため、この経路を使ってマルウェアが引き込まれた場合、IDS 等での検出は困難です [6]。最近では、6to4 は historic status への移行が議論されています [7]。

4.2　6to4 が有効になっているデバイスからの RA 送出

複数のグローバル IPv4 アドレスのサブネットに同時に接続している Windows 機器ではそれぞれのインターフェースについて 6to4 を張ります。複数のトンネルを持っている Windows 機器が、インターネット共有やルータ機能を有効にしていないにもかかわらず IPv6 の RA を IPv4 サブネット側に送出することがあります。無線・有線の両方に接続している場合、2 つの有線に同時接続している Windows 7 (SP なし) 場合で、まれにインストール直後にこのような事例が発生することがあります。このとき、その他の機器がこの RA により、グローバル IPv6 アドレスを意図せず取得してしまい、かつ IPv6 のデフォルトルートがその Windows 機器に向いてしまいます。

この現象によって不要な RA を受信した機器の IPv6 アドレスは

```
2002:XXXX:YYYY::ZZZZ:ZZZZ:ZZZZ
```

のようなものが設定されます。このうち、`XXXX:YYYY` の部分は 6to4 している機器の IPv4 アドレスなので誰が原因であるかを判断することが出来ます。

IPv6 対応機器のルータ探索パケット (RS, Router Solicitation) に反応して RA が出続けるわけではありません。しかし、いずれのアドレスも RA で有効期間が非常に長く設定されているため、配布した機器を停止させても配布された IPv6 アドレスが無効になるまではかなりの時間がかかります。また、マルウェアが意図的に RA を Wi-Fi 側にばらまいてトラフィックを吸い付けるという事例も発生しているようです。

Windows では RA の送出とルーティングの有無は別個に設定されます。自分のデバイスがどのように設定されているかは以下のように確認できます。

- コマンドプロンプトを開く
- 内部で `netsh` を実行する
- `netsh` 中で `interface ipv6` と入力し、プロンプトが `netsh interface ipv6>`になることを確認
- `show interface` を実行するとインターフェース一覧が表示される。左端のカラムにインターフェース番号が羅列されている。
- `show interface` インターフェース番号を実行すると、インターフェース毎の属性が表示される。
- 「転送」「アドバタイズ」が `disabled` になっていれば RA は出さない設定となっている

Firewall では、6to4 や Teredo などのトンネル通信の中身を見ることはできません。もしこれらのトンネルを経由してセキュリティ事象が発生すると何が起きているかわからなくなります。適切にフィルタするべきです。なお、前述のようにグローバル IPv6 アドレスが付与されるようなネットワークでは 6to4 や Teredo は自動的に接続されません。

Teredo および 6to4 をアプリケーションとして識別できるようなファイアウォールを使用している場合は、それによって不許可にするのが簡便です。そのような設定が出来ない場合は、6to4 は IP プロトコル番号、Teredo は UDP ポート番号の指定によってフィルタすることが可能です。6to4 は IP パケットですが、TCP でも UDP でもありません。しかし、プロトコル番号が 41 なのでフィルタも容易です。一方、Teredo は UDP でポート番号が 3544 ですが、いつでもこのポートであるとは限りません。

ポート以外に名前解決でトンネルの開通を止めることもある程度は可能です。Windows が自動的に Teredo もしくは 6to4 を有効にする際には DNS で名前解決を行います。現時点では teredo サーバである `teredo.ipv6.microsoft.com`、および 6to4 サーバである `6to4.ipv6.microsoft.com` の名前解決を拒否することでトンネルの自動開通を抑制できますが、将来にわたって十分とは言えません。

4.3 IPv4 がプライベートアドレスの場合もトンネルが張られるのか

6to4 はその仕組み上 IPv4 グローバルアドレスが必要なので、NAT 装置の配下でプライベートアドレスを使用している機器では有効になりません。一方 Teredo については NAT を通過できるように設計されているため、IPv4 プライベートネットワークであっても、マイクロソフト社のサーバあてのトンネルとして開通してしまいます。

4.4 ファイアウォール等の設定の留意点

昨今では Firewall を L2 デバイスとして組織の出入り口直後に挿入する運用も一般的になっています。一部の Firewall は IPv6 パケットを「未知の Ethernet フレーム」として取り扱い、フィルタしない機器が存在します。そこで、IPv6 パケットが意図通り Firewall でフィルタされているかを確認しておくことが必要です。

6to4 や Teredo をフィルタする場合は実際にユーザ端末で確認することが出来ます。前述の netsh コマンドを使用してトンネルインターフェースが `dormant` となっていれば使用されません。`dormant`

でない場合はフォールバック動作 (まずは IPv6 への接続性を試してタイムアウトしてから IPv4 で接続する) のためにアプリケーションの反応が遅いという問題を引き起こすことがあります。

5　IPv6の今後の方向性

5.1　DNS混ぜるな危険：IPv6には移行できない

以下に述べる理由から、現在の ICANN (The Internet Corporation for Assigned Names and Numbers)[8] の管理する名前空間において DNS は IPv6 に移行することができません。IPv6 普及のために AAAA レコードを現在の DNS に混ぜたことが IPv6 への移行の障害となっています。

5.1.1　DNS 権威サーバ

あるドメインの DNS 権威サーバが IPv4 アドレスを捨てるためには、そのドメイン名を問い合わせるすべての DNS キャッシュサーバが IPv6 に対応するのを待つか、それらを切り捨てる必要があります。そして、IPv4 から完全に独立したドメイン名は、他の IPv4 に依存しているドメイン名に依存しても依存されてもいけません。

IPv6 だけで名前解決したいドメイン名は、ルートからの名前空間ツリーの上位に位置するドメイン名の DNS 権威サーバがすべて IPv6 に対応し、再帰検索が IPv6 で完結することを必要とします。

さらに、*A Survey of DNS Security*[9] や力武、鈴木らの *Now and The Future* 3.2. Increasing level of indirection[10] にあるように、ドメイン名の依存関係は非常に複雑に入り組んでおり (平均 46 ドメイン〜最大 100 以上の依存関係)、IPv6 が独立して成り立つためにはこの依存関係を解消しなくてはいけません。

つまり、あるドメイン名の権威サーバの IPv6 化は、上位ゾーンがすべて IPv6 に対応したうえ、委譲を内部名の AAAA レコードで行うことで可能となります。しかし、下位に委譲を行っている、あるいは外部名への委譲によって依存関係のあるドメイン名が存在する場合には、それらすべてが IPv6 に対応する必要が生ずるということです。なお、現在は RFC3901 (BCP) : DNS IPv6 Transport Operational Guidelines により、DNS 権威サーバには以下のように IPv4 対応が要請されています [11]。

- すべての再帰ネームサーバは、IPv4 のみか、デュアルスタックであるべき
- すべてのゾーンには IPv4 で到達可能なネームサーバが一つは提供されるべき
- 子ゾーンへの委譲には少なくとも 1 つの IPv4 アドレスがあるべき

さらに、AAAA のないネームサーバが 1 つはないと (すべてのネームサーバに AAAA があると) 再帰検索に失敗する実装があることが報告されています [12, 13]。

5.1.2　DNS キャッシュサーバ

DNS キャッシュサーバにおいては、WAN 側のインターフェースは IPv4 に依存するドメイン名の再帰検索のために最後まで IPv4 アドレスを捨てることはできません。また、LAN 側インターフェースについてはすべてのクライアントが IPv6 トランスポートで　DNS 問合せが行えるようになるまでは IPv4 を捨てることはできません。実際、IPv6 に対応していても名前解決を IPv4 で行う必要のある機器はまだまだ多いのが現状です。

5.2 IPv6 と IPv4 は並行運用するしかない?

前述のように、DNS 一つをとっても IPv6 への完全移行は困難です。サーバ、たとえば Web やメールサーバは、すべてのクライアントが IPv6 に対応するまで IPv4 へのサービスを捨てることはできません。

クライアントは、必要なすべての周辺機器が IPv6 に対応するまで IPv4 を捨てらません。いまや、プリンタ・FAX やスキャナ、さらに実験器具などもネットワーク接続が標準です。それらがすべて IPv6 対応にならないと、クライアントであっても IPv4 接続を捨てることができません。

5.3 並行運用する意義はあるか

それでは、以上の 2 つの仮定が同時に正しいとしましょう。しかし、以下のように矛盾が生じます。つまり、IPv4 は捨てられず、IPv6 も必要だとすれば並行運用が必要ということになります。

- サーバ側は IPv6 しか対応しないクライアントのために IPv6 対応することが必要
- クライアントは IPv6 に対応していない大多数のサーバへの接続性を確保するために、NAT やプロキシを介してでもどこかに IPv4 アドレスを持つことが必要

NAT やプロキシを介さずにエンドツーエンドで通信する必要のあるアプリケーションがない限り並行運用の必要はないことになります。そういうアプリケーションが生まれてくる土壌を用意し、ジットレインのいう「生み出す力」[14] が失われない「インターネット」を維持するという意義は考えられるかもしれません。

5.4 並行運用するとすればどういう形態が望ましいか

以上から、安易に IPv6 を導入するとセキュリティ上の問題が生じかねません。ここでは、十分にセキュリティを考慮したうえで、すでに現在のインターネットでは幻想となっているオープンなインターネットを追い求めようという意思がある組織が IPv6 を導入する場合を考えてみましょう。IPv6 への移行を求めるのであれば、IPv4 の利用を限定していく方向を模索しないと意味はありません。完全なデュアルスタックではなく、サーバにのみ IPv4 を残し、クライアントは IPv6 化を進め、IPv4 ネットワークとの通信はゲートウェイ経由に限定していくのが望ましいと考えられます。

5.4.1 サーバ側

IPv4 時代のアプリケーションを提供するサーバにおいて、IPv4 アドレスが枯渇したクライアントへの接続性提供は特に考慮しなくともクライアント側にある LSNAT やプロキシが問題解決してくれると考えられます。しかし、それらのクライアント側ゲートウェイの負荷軽減や、IPv6 移行への道を残すために デュアルスタックあるいは IPv4 サーバ、IPv6 サーバの並行運用を行うことは意味があります。

DNS については、権威サーバおよびキャッシュサーバの WAN 側には最後まで IPv4 アドレスが必要であり、IPv6 トランスポートに無理に対応する必要はなく AAAA レコードの登録ができれば問題ありません。IPv6 が普及する可能性よりは DNS がその脆弱性により使いものにならなくなる時代が先にくると考えるのが現実的でしょう。

5.4.2 クライアント側

クライアントネットワークをデュアルスタックにする必要はあるでしょうか。本報告書の他の章にもあるように、デュアルスタックは名前解決を中心に多様な問題を抱えることになるため、コストをかけてまでデュアルスタック化する意義は少ないでしょう。

問題は「生み出す力」です。NAT や Proxy の下では新しい 時代の「何か」を生み出す力は大きく制限されるでしょう。生み出す力を提供するためにエンドツーエンドのネットワークが必要と考えるならばどうすればよいでしょうか。

効率的かつ導入に意味のある解はクライアントネットワークを IPv6 シングルスタック化してしまうことです。IPv4 ネットワークに対しては NAT や Proxy を介せば接続性を担保できます。IPv4 プライベートアドレスを使っているのであれば何も変化はなく、IPv4 グローバルアドレスを用いているとしても、現在のインターネットは多くのセキュリティの壁に阻まれています。エンドツーエンドの接続を取り戻すために IPv6 に移行するという選択はありえるでしょう。

クライアントネットワークを IPv6 シングルスタック化する際に必要なものは以下が考えられます。

1. ネットワークの IPv6 化
   - 全てのルータの IPv6 化
   - DHCPv6 サーバ (全ての機器で RA が DNS 対応していれば不要)
   - ファイアウォールの IPv6 化
   - IPv6 対応の DNS キャッシュサーバ (IPv4 ネットワークとゲートウェイするために DNS64 対応が必要)
   - NAT64 (IPv4 へのゲートウェイ) あるいはプロキシサーバ
2. IPv4 ゲートウェイへのルーティング
   以下の技術の組み合わせでクライアントと IPv4 ゲートウェイの間のシングルスタック化が可能
   - DNS64/NAT64 あるいはプロキシ (WWW など対応できるものだけでよければ)
   - (IPv6 完全対応ができない場合) 464XLAT など
3. イントラサーバの IPv6 化 (NAT64 の外側に置けば対応不要)
4. クライアント OS の IPv6 化
   - IPv6 スタック
   - IPv6 による名前解決
   - (完全対応できない場合) 464XLAT のような変換技術を利用
5. クライアントアプリケーションの IPv6 化
6. 周辺装置の IPv6 化
   対応が不十分であれば NAT64 を介する手もある
7. 運用管理の IPv6 化
   サーバ/ネットワークの管理・監視システムの IPv6 化が必要

8. 技術者の IPv6 化
   教育が必要 (これが一番困難か)

これらが揃わないと IPv6 シングルスタック化は困難ですが、技術は未熟であり時期をよく見極める必要があります。しかし IPv6 シングルスタック化ができないからといってデュアルスタック化を選択することに意義はあるでしょうか。一般には考えられないが、情報工学系の教育現場においては IPv6 の学習と「生み出す力」のために未熟な状態であっても導入する意義はあるでしょう。

以上述べた方策をとっても IPv6 への移行は困難です。名前解決を行わない (例えば P2P のような) システムを最初から IPv6 利用者だけで築いていくことが唯一 IPv4 を捨てる道でしょう。

6　ファイアウォール、キャッシュＤＮＳサーバによる通信の制御について – IPv4 と IPv6 との比較 –

6.1　はじめに

自組織のネットワークが IPv4/IPv6 のデュアルスタックである場合に、予期しない IPv6 トンネルを遮断する方法を考えます。自組織の IPv4 ネットワークアドレスを、`ppp.qqq.0.0/16`，IPv6 ネットワークアドレスを `rrrr:ssss:tttt::/48` として設定例を書きます (`ppp`、`qqq` は、それぞれ 0〜255 の十進数、`rrrr`、`ssss`、`tttt` は、0〜ffff の 16 進数)。図 6 のようなネットワークを想定して、Router と Firewall の設定例について述べます。IPv6 ノードの要求仕様は rfc 6434 に記述されています [15]。

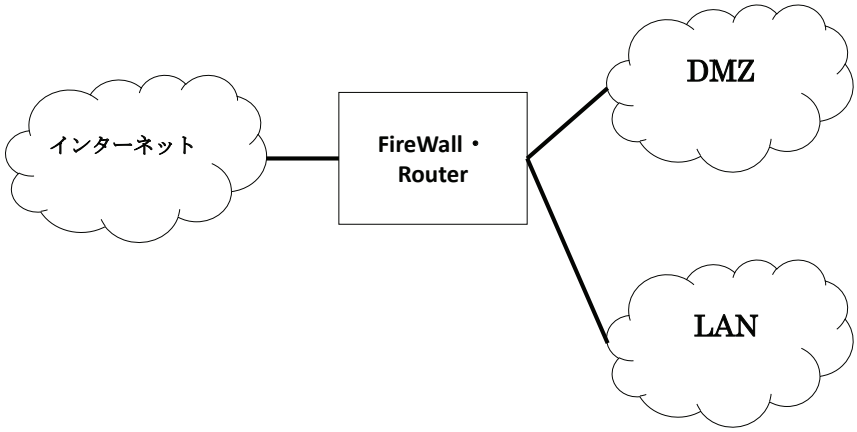

図 6: 対象となるネットワーク

6.2　サンプルポリシー

ファイアウォール/ルータで、以下の通信要件を想定します。

(a) 送信元アドレスを偽装しているパケットを遮断する。(外⇒内、内⇒外)

(b) ブロードキャストアドレス等宛のパケットを遮断する (外⇒内

(c) 未割当アドレス宛のパケットを吸収する　(外⇒内)

(d) プライベートアドレス宛のパケットを吸収する　(内⇒外)

(e) IPv6 over IPv4 のトンネルは、遮断する (内⇒外、外⇒内)

(f) 公開サーバのポートを開ける (外⇒内)

(g) それ以外のポートは閉じる (外⇒内)

(h) icmp echo request, icmp redirect は、遮断する (外⇒内)

(i) 他の icmp は、通す (Path MTU Discovery のために必須)[16](外⇒内)

(j) メールの送出はメールサーバのみに限定する (内⇒外、OP25B)

(k) icmp redirect は、遮断する (内⇒外)

(l) 他の icmp は、通す (Path MTU Discovery のために必須)[16](内⇒外)

(m) それ以外は、通す。(内⇒外)

ルータの ACL (Access Control List), static route, ファイアウォールのポリシーにより、実装します。

表 1: ACL, IPv4, 外→内

`ip access-list extended ingress-filter-ipv4`
`10 deny ip 0.0.0.0 0.255.255.255 any`
`20 deny ip 10.0.0.0 0.255.255.255 any`
`30 deny ip 127.0.0.0 0.255.255.255 any`
`40 deny ip 172.16.0.0 0.15.255.255 any`
`50 deny ip ppp.qqq.0.0 0.0.255.255 any`
`60 deny ip 100.64.0.0 0.63.255.255 any`
`70 deny ip 169.254.0.0 0.0.255.255 any`
`100 deny ip 192.168.0.0 0.0.255.255 any`
	ポリシー (a)

`220 deny ip any ppp.qqq.0.0 0.0.255.0` `230 deny ip any ppp.qqq.0.255 0.0.255.0`	ポリシー (b)

表 2: AACL IPv6, 外→内

`ipv6 access-list ingress-filter-ipv6`
`10 deny ipv6 rrrr:ssss:tttt::/48 any`
`100 permit ipv6 any any`
	ポリシー (a)

ファイアウォールポリシー

プロトコルが、icmp, ipv6 over ipv4 の場合には、ポート番号がないため、当該欄は、空欄となります。

表 3: ACL IPv4, 内→外

```ip access-list egress-filter-ipv4 10 permit ip ppp.qqq.0.0 0.0.255.255 any 100 deny ip any any```	ポリシー (a)

表 4: ACL IPv6, 内→外: (IPv6 では、リンクローカルアドレスを許可する必要あり。)

```ipv6 access-list egress-filter-ipv6 10 permit ipv6 rrrr:ssss:tttt::/48 any 20 permit ipv6 fe80::/10 any 100 deny ipv6 any any```	ポリシー (a)

表 5: static route

```ip route ppp.qqq.0.0/16 Null 0```	ポリシー (c)
```ip route 1.0.0.0/24 Null 0 ip route 1.1.1.0/24 Null 0 ip route 10.0.0.0/8 Null 0 ip route 100.64.0.0/10 Null 0 ip route 169.254.0.0/16 Null 0 ip route 172.16.0.0/12 Null 0 ip route 192.168.0.0/16 Null 0```	ポリシー (d)
```ipv6 route rrrr:ssss:tttt::/48 Null 0```	ポリシー (c)

表 6: (外⇒内)

送信元		宛先		protocol	許可/遮断	ポリシー
IP アドレス	ポート	IP アドレス	ポート			
any	any	メールサーバ	25	tcp	○	ポリシー (f)
any	any	any	25	tcp	×	ポリシー (g)
any	any	DNS 権威サーバ	53	tcp/udp	○	ポリシー (f)
any	any	any	53	tcp/udp	×	ポリシー (g)
any	any	Web サーバ	80	tcp	○	ポリシー (f)
any	any	any	80	tcp	×	ポリシー (g)
any	any	ntp サーバ	123	udp	○	ポリシー (f)
any	any	any	123	udp	×	ポリシー (g)
any	-	any	-	icmp echo request	×	ポリシー (h)
any	-	any	-	icmp redirect	×	ポリシー (h)
any	-	any	-	icmp その他	×	ポリシー (i)

表 7: (内⇒外)

送信元		宛先		protocol	許可/遮断	ポリシー
IP アドレス	ポート	IP アドレス	ポート			
メールサーバ	any	any	25	tcp	○	ポリシー (j)
any	any	any	25	tcp	×	ポリシー (j)
any	any	any	any	tcp	○	ポリシー (k)
any	any	any	any	udp	○	ポリシー (k)
any	-	any	-	ipv6 over ipv4	× ×	ポリシー (e)
any	-	any	-	icmp redirect	×	ポリシー (k)
any	-	any	-	icmp その他	×	ポリシー (l)

6.3 キャッシュDNS(bind 9.5 以降) の設定

6.3.1 ipv6 over ipv4 トンネルの抑制

ipv6 over ipv4 tunnel のゲートウェイへの接続を抑制するために、キャッシュDNS サーバに、bind の rpz(responsible policy zone) 機能を使用して、6to4 サーバ `6to4.ipv6.microsoft.com`、及び Terado サーバ `teredo.ipv6.microsoft.com` の問い合わせに `nxdomain`(登録なし) を返答するように設定します。

表 8: ファイル : named.conf (一部)

```
options{
   response-policy{
      zone “policy.example.jp”
   };
};
zone “policy.example.jp” {
   type master;
   file “data/policy.example.jp.zone”;
};
```

表 9: ファイル : policy.example.jp

```
$TTL      600
@     IN   SOA   localhost.   nobody.localhost.   (1  60  60  60  60)
      IN   NS    localhost.
6to4.ipv6.microsoft.com        IN  CNAME   .    ; NXDOMAIN
teredo.ipv6.microsoft.com      IN  CNAME   .    ; NXDOMAIN
;; 左辺は"."で終端しない
```

6.3.2 無駄な問い合わせの抑制

無駄な問い合わせを権威サーバに送らないために、6to4 で使用する IPv6 アドレスの空の逆引きゾーンを設定します。

表 10: ファイル : pnamed.conf (一部)

```
option {
   empty-zone-enable yes;
}
zone "2.0.0.2.IP6.ARPA" IN {
   type master;
   file "data/null.zone";
};
```

表 11: ファイル : null.zone

```
$TTL      86400
@     IN   SOA   localhost.   nobody.localhost.   (
                 1
                 10800
                 3600
                 3600000
                 86400 )
      IN   NS    localhost.
```

6to4 では、4 節で述べたように 2002(16 進数) のあとに IPv4 アドレス 32 ビット続けたアドレスを使用します。そのため、6to4 ゲートウェイに接続できなくても、LAN 内の閉じた 6to4 アドレスのネットワークを構成できます。そのため、上記の逆引きの抑制が必要となります。なお、Teredo で使用する IPv6 アドレスや、link local アドレス、IPv4 の private アドレス等の予約アドレス空間の逆引きゾーンは、bind9.5 以降には最初から組み込まれていますから、`empty-zone-enable yes;`(default) の設定のみで、ゾーン自体を改めて設定する必要はありません。

表 12:

```
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
D.F.IP6.ARPA
8.E.F.IP6.ARPA
9.E.F.IP6.ARPA
A.E.F.IP6.ARPA
B.E.F.IP6.ARPA
8.B.D.0.1.0.0.2.IP6.ARPA
0.in-addr.arpa
10.in-addr.arpa
64.100.in-addr.arpa
65.100.in-addr.arpa
  ∼
127.100.in-addr.arpa
127.in-addr.arpa
254.169.in-addr.arpa
16.172.in-addr.arpa
17.172.in-addr.arpa
  ∼
31.172.in-addr.arpa
2.0.192.in-addr.arpa
168.192.in-addr.arpa
100.51.198.in-addr.arpa
113.0.203.in-addr.arpa
255.255.255.255.in-addr.arpa
```

参考文献

[1] 日本ネットワークインフォメーションセンター、「IPv4 アドレスの在庫枯渇に関して」(2011) https://www.nic.ad.jp/ja/ip/ipv4pool/

[2] rfc 1883, http://tools.ietf.org/html/rfc1883

[3] The KAME project, http://www.kame.net/

[4] Internet Society, *The world IPv6 Launch*, http://www.worldipv6launch.org/

[5] rfc 5952, http://tools.ietf.org/html/rfc5952

[6] IPv6 技術検証協議会 セキュリティ評価・対策検証部会、http://ipv6tvc.jp/documents/20121023Report.pdf

[7] http://tools.ietf.org/html/draft-troan-v6ops-6to4-to-historic-01

[8] https://www.icann.org/

[9] http://www.cs.cornell.edu/people/egs/beehive/dnssurvey.html

[10] Rikitake, K., Suzuki, T. and Nakao, K., DNS Security: Now and The Future, IEICE Technical Report ICSS2007-01, pp.3-8 (2007).9.

[11] rfc 3901, http://tools.ietf.org/html/rfc3901

[12] http://dnsops.jp/bof/20081125/2008-11-25-dnsops.jp-BoF-dual-stack-01.pdf

[13] http://www.janog.gr.jp/meeting/janog33/program/dns.html

[14] ジョナサン・ジットレイン、「インターネットが死ぬ日」(井口耕二 (訳))(ハヤカワ新書 juice、2009)

[15] rfc 6434, http://tools.ietf.org/html/rfc6434

[16] rfc 1981, http://tools.ietf.org/html/rfc1981

添付資料

SA46T:IPv4 アドレス枯渇後の IPv6 移行と IPv4 継続利用を両立するカプセル化技術

松平直樹（富士通株式会社）

SA46T:IPv4アドレス枯渇後のIPv6移行とIPv4継続利用を両立するカプセル化技術

松平 直樹
富士通株式会社

APNIC など RIR に於ける IPv4 アドレス枯渇が現実となり、事業者の保有する IPv4 アドレス枯渇も時間の問題となった。IPv4 アドレス枯渇後は、IPv6 only な状況が自然発生する。このような IPv4 アドレス枯渇後の状況を踏まえた SA46T と呼ぶカプセル化技術を提案する。SA46T の本質は以下の 2 点である。(1) IPv4 アドレスのロケーターとアイデンティファイアの関係をそのまま IPv6 アドレス空間にマッピングすることにより、従来のアプセル化技術の課題であった設定数が膨大になる問題を解決した。(2) IPv6 アドレスのロケータ部に、IPv4 ネットワークの識別子を追加することにより、約 4.3 億個の IPv4 ネットワークを IPv6 バックボーンネットワークに多重化可能とした。実装の開発と、LAN 環境及び広域環境での実証実験を通じ、SA46T が実際に動作し、運用出来ることを証明した。さらに、独立に開発された実装間で相互接続可能であることも証明した。SA46T はシンプルな技術であり、十分実利用可能であることを示した。SA46T により、IPv6 移行と IPv4 継続利用を両立できる。特に、IPv4-IPv6 変換技術と相性の悪い IPv4 アプリケーションの継続利用に貢献できる可能性が高そうである。

SA46T: Encapsulation Technology which enable both transition to IPv6 and IPv4 continuous use

Naoki Matsuhira
Fujitsu Limited

Exhaustion of IPv4 addresses at RIRs such as APNIC became real, exhaustion of IPv4 addresses at provider will become real soon. After the exhaustion of IPv4 addresses, IPv6 only situation will be a natural thing. Based on such situation of IPv4 addresses exhaustion, encapsulation technology called SA46T is proposed. Following two points are the essences of SA46T. (1) By mapping the locator and identifier relationship of IPv4 address directory to the IPv6 address space, then solves the issue of number of configurations of existing encapsulation technology. (2) By adding the identifier of IPv4 network to locator part of IPv6 address, multiplexing is enabled approximately 430 million IPv4 networks to a single IPv6 backbone network. Through developing an implementation of SA46T, and operation as field demonstration both LAN and WAN environment, SA46T is proved to actually work and operate. And also, SA46T is proved that have interoperability between the implementations which are developed independently. SA46T is a simple technology, and shown that actually use. SA46T enable both transition to IPv6 and continuous use of IPv4. In particular, can contribute for continuous use of IPv4 applications which are unfriendly with IPv4-IPv6 translation technology.

1 はじめに

IPv4 アドレスの枯渇以後の状況を踏まえた移行技術として、SA46T と呼ぶ技術を提案する。SA46T は、IPv6 移行と IPv4 継続利用を両立するカプセル化技術である。本論では、まず、IPv4 アドレス枯渇による環境変化を述べた後、SA46T 技術について述べる。さらに、SA46T の実装と実証実験、相互接続試験についても述べる。最後に、本技術を評価する。

2 IPv4アドレス枯渇後の移行技術の課題

2.1 IPv6及び移行技術とIPv4アドレス枯渇

IPv4 アドレス枯渇が IETF(The Internet Engineering Task Force) で予測されたのは 1992 年である。この問題の抜本解として IPv6 が検討され、1996 年に第一段階の標準化 (Proposed Standard) が完了、RFC1883[1] として発行された。

IPv6 の標準化と並行して、IPv4 から IPv6 への移行技術についても検討が進められ、RFC1933[2] として発行された。これら移行技術は、(1) Dual Stack (IPv4/IPv6 双方に対応)、 (2) カプセル化、(3) IPv4-IPv6 変換技術 (トランスレータ) の 3 つに分類されている。

IETF での精力的な検討と、盛んな実装開発が行われ、2001 年にはマルチベンダ環境でのネットワーク構築が可能 [3] となったが、IPv6 の普及は順調とは言えず、ついに 2011 年 2 月に IANA(Internet Assigned Numbers Authority) で、2011 年 4 月に APNIC(Asia Pacific Network Information Centre) で、そして 2012 年 9 月には RIPE NCC(Reseaux IP Europeens Network Coordination Centre) で IPv4 アドレスが枯渇した。事業者の保有する IPv4 アドレスが枯渇するのは時間の問題とみられる。

2.2 IPv4 アドレス枯渇が移行技術に与える影響

IPv4 アドレスの枯渇後には、IPv6 only なノードが自然発生する。これは、Dual Stack の構成を取れない場合が生じること、従って、IPv6 only の状況を前提としなければならない状況に変化したことを意味する。この状況変化により、これからは、カプセル化及び IPv4-IPv6 変換技術の重要性が相対的に高まることとなる。

具体的には、ルータもしくはネットワークが IPv6 only となる場合、IPv4 を中継するには、IPv4 over IPv6 のカプセル化技術が必須になる。一方、ホストが IPv6 only となる場合、IPv4 ホストとの通信を行う際には、IPv4-IPv6 変換技術が必須になる。

本論は、このうち、IPv4 over IPv6 技術を対象にする。

2.3 既存のカプセル化の問題点

IPv4 over IPv6 の既存技術は、RFC2473[4] で記載されている、Generic Packet Tunneling in IPv6 である。また、カプセル化として古くから使われている技術は、RFC1853[5] で記載されている、IP in IP Tunneling である。

一般にカプセル化には、主に 2 つの問題がある。ひとつは、再帰的なカプセル化が起こりうるなど、管理が難しいこと、もうひとつは、設定数が多いことである。具体的に、N 個の拠点をフルメッシュで接続するための設定数は、$N(N-1) = N^2 - N$ となり、設定数がべき乗で増加する。

これらの理由により、既存のカプセル化技術は、スケーラビリティを有しているとは言い難いと言えよう。大規模に適用するのは容易でなく、部分的もしくは暫定的に使う技術として位置づけざるをえない。しかも、新たに拠点を追加する際に、既存の拠点での設定追加も必要になる。これは運用上、致命的な問題と言え、既存のカプセル化技術に依存することは難しいと考えられる。

なお、RFC2473 では、Tunnel Encapsulation Limit Option が設けられ、カプセル化段数を制限できる。

3 SA46T 技術

SA46T(Stateless Automatic IPv4 over IPv6 Encapsulation / Decapsulation Technology) は、上記したカプセル化の問題点を抜本的に解決した、全く新しい発想によるカプセル化技術である。以下、その技術について述べる。

3.1 SA46T の導入位置と処理対象

3.1.1 ネットワーク構成

一般に、ネットワークは、バックボーンネットワークと複数のスタブネットワークが接続された形態とモデル化できる。

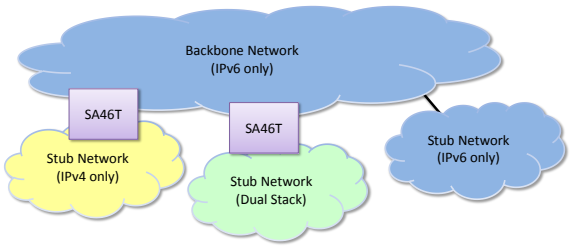

図 1: ネットワーク構成

図 1 は、SA46T の導入位置を示すためのネットワーク図である。スタブネットワークは IPv4 only、Dual、 IPv6 only の 3 種に分類できる。

SA46T はバックボーンネットワークと IPv4 ホストが存在するスタブネットワークの境界に導入する。スタブネットワークが IPv6 only であるなら、SA46T の導入は必要ない。SA46T により、IPv4 パケットが IPv6 にカプセル化されるため、バックボーンネットワークを IPv6 only 化できる。

なお、エッジに SA46T を導入するという導入ポリシにより、再帰的トンネルの発生を回避できる。SA46T の使い方を判り易くする効果もある。RFC2473 の Tunnel Encapsulation Limit Option 相当の機能も不要となるので、Tunnel MTU をこのオプション分短くせずに済む効果もある。

3.1.2 SA46T が処理対象とする範囲

表 1 は、エンドホスト間で、IPv4/IPv6 どちらのプロトコルスタックが用いられるかを示したものである。IPv4 と IPv6 双方が利用可能な場合は IPv6 が用いられる。

表 1: 使用されるプロトコルスタック

	IPv6 only	Dual Stack	IPv4 only
IPv6 only	IPv6	IPv6	-
Dual Stack	IPv6	IPv6	IPv4
IPv4 only	-	IPv4	IPv4

SA46T が機能提供するのは、IPv4 only ホストと IPv4 only ホスト間の通信、または IPv4 only ホストと Dual Stack ホスト間の通信で、表 1 に於いて使用されるプロトコルスタックが IPv4 と記載される範囲である。IPv6 と記載される通信に SA46T は関与しない。

IPv4 only ホストと IPv6 only ホスト間は、このままでは通信できない。このようなケースは、NAT-PT[6] や NAT-64[7] 等の、IPv4-IPv6 プロトコル変換技術を用いれば良い。SA46T は運用に於ける創意工夫を重視し、必要に応じ、自由に組み合わせ可能とすることを重視している。SA46T はこれら IPv4-IPv6 プロトコル変換技術と機能的に直交しており、自由に組み合わせ可能である。

3.2 SA46T プロトコル

3.2.1 カプセル化

SA46T は、IPv4 over IPv6 技術であり、IPv4 パケットを IPv6 でカプセル化することは一般のカプセル化技術と違いは無い。

3.2.2 IP アドレスの解決／生成

カプセル化の際、Outer header の IP アドレスの解決もしくは生成が必要となる。Outer header の IPv6 アドレスを、SA46T アドレスと呼ぶ。SA46T のアドレスの形式を図 2 に示す。

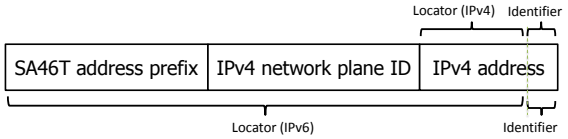

図 2: SA46T アドレス

SA46T アドレスは、SA46T prefix、 IPv4 network plane ID、 IPv4 アドレスから構成される。

SA46T prefix
SA46T アドレスの prefix

IPv4 network plane ID
IPv4 ネットワークの識別子

IPv4 address
Inner header の IPv4 アドレス

3.3 SA46T のアーキテクチャ

SA46T のアーキテクチャの本質は、アドレッシングとルーティングにあり、そのポイントは以下の 2 点である。

1. IPv4 アドレスをロケータとアイデンティファイアの関係を維持したまま IPv6 アドレス空間にマッピング
2. IPv4 network plane ID のロケータ部への埋め込みによる複数 IPv4 ネットワークの IPv6 ネットワークへの多重化

3.3.1 マッピングとルーティング

IP アドレスは、ロケータとアイデンティファイアから構成され、ロケータ部を経路として広告する。インタフェースは、ロケータとアイデンティファイア全体で識別される。この構造は IPv4 でも IPv6 でも変わらない。IP の基本的なアーキテクチャである。

SA46T は、IPv4 アドレスのロケータとアイデンティファイアの境界を、そのまま維持して IPv6 空間にマッピングする。IPv4 サブネットのアイデンティファイアが n bit のとき、ロケータが $32-n$ bit となる。これを、SA46T ではロケータを $128-n$ bit、アイデンティファイアを n bit として IPv6 空間にマッピングする。

経路は、IPv4 では、$32-n$ bit 長の prefix になるが、IPv6 では、$128-n$ bit 長の prefix となる。IP アドレス長は異なるが、その意味するところは同じである。

経路数は、Dual stack で運用する場合と同一になる。Dual stack 環境では、IPv6 の経路数と IPv4 の経路数の合計が総経路数になる。SA46T 環境では、IPv6 の経路数に加え、IPv4 の経路を SA46T のマッピングルールに則って変換された IPv6 の経路数が総経路数となる。従って、総経路数は変わらない。

3.3.2 IPv4 network plane ID

IPv4 では、グローバルユニークでないプライベートアドレスが定義されている。これを、IPv6 空間にて、グローバルユニークにするために付与する情報が、IPv4 network plane ID である。

IPv4 network plane ID が 32bit の場合で、約 4.3 億個の IPv4 ネットワークを多重化できる。事実上、上限を意識する必要がなくなると言えるであろう。

3.4 設定について

SA46T の設定に必要な情報は、SA46T address prefix の値、IPv4 network plane ID の値、そして、prefix 長の 3 つで、通常、1 行で記述する。

つまり、SA46T address prefix + IPv4 network plane ID/prefix length である。

SA46T 配下の IPv4 サブネットあたり 1 行なので、N 個の IPv4 サブネットを有するネットワーク全体で、N 個の設定が必要なだけである。

新たに IPv4 サブネットを接続する際に必要な設定は、新規に追加するインタフェースを有する SA46T にのみ 1 行の設定が必要になるだけである。既存の SA46T への設定変更は一切必要ない。

このように、あくまで、新たに接続するスタブネットワークに対する設定のみでよく、スケーラビリティを有する技術と言える。

3.5 SA46T と従来のカプセル化技術との違い

従来のカプセル化技術は、インタフェースとインタフェースを結ぶもので、仮想的なリンクを生成する技術である。点と点を結ぶ、従って、始点と終点の設定をそれぞれ行う必要が生じるので、N^2 もの設定数を必要とする。新規追加に際しても、それぞれ相手方の点の情報が必要になるので、既存の点に於いても設定が必要となる。

一方、SA46T は、あくまで、IPv4 アドレスをロケータとアイデンティファイアの構造を維持したまま IPv6 アドレス空間にマッピングした技術であり、仮想的なリンクを生成する技術ではない。マッピングルールは一意であり変換は直ちに可能となる。あとはマッピングした経路情報をルーティングプロトコルにより伝搬させることによりルーティング可能とする。SA46T に於ける設定情報は、実は、自身のサブネット情報そのものであり、それをルーティングプロトコルで伝えるのである。

IPv4 network plane ID により、IPv4 アドレスが同じであっても、異なるネットワークプレフィックスとして表現させることができる。あるスタブネットワークに接続される SA46T に於いて、IPv4 network plane ID にある値を設定するということは、同じ値が設定されている面に、そのスタブネットワークを参加させるということを意味する。IPv4 network plane ID は、IPv4 サブネットが属するグループの番号であり、顧客番号もしくは配線番号と読んでも差し支えない。同一の値を設定すれば済むので判り易く、間違えにくい。

以上から、SA46T は全く新しい発想によるカプセル化技術であると言えよう。

4 SA46T の特徴

以下に、SA46T の特徴を示す。

(1) Stateless

状態を持たないため、状態管理対象の上限など、数の制限が無く、かつ、状態不一致がないので信頼性も高い。

(2) IPv6 only による運用コストの削減が可能

プロトコルスタックを複数運用するより、シングルスタックのみの方が、運用コストは低い。IPv4 アドレスが枯渇した今日、選択肢となるのは IPv6 のみである。

(3) 少ない設定数で済む (N^2 でなく N)

既に述べたように、設定数が少ないのでスタブネットワークの数が多い大規模ネットワークにも導入しやすい。拠点数が増大すればするほど、その効果は大きい。

(4) 拠点追加に際し既設拠点の設定変更が不要

既に述べた通り、新たに拠点を追加する場合、その拠点での設定のみで十分であり、既設拠点の設定変更は必要ない。

(5) 特別なプロトコルを必要としない

スタブネットワーク、バックボーンネットワーク双方に於いて、OSPF(Open Shortest Path First) や IS-IS、BGP(Border Gateway Protocol) など、既存のルーティングプロトコルを拡張・変更することなしに、そのまま利用できる。

(6) 任意の Layer2 技術上で動作 (有線、無線)

Layer3 の技術であり、Ethernet や IEEE802.11 の無線 LAN でも動作する。

(7) 数億の IPv4 ネットワークの多重化が可能

plane ID により、IPv4 ネットワークの多重化が可能となる。plane ID に 32bit 割り当てた場合で、4.3 億の多重化が可能となる。

(8) IPv4 の運用を容易に停止可能

スタブネットワークに IPv4 ホストが無くなれば、SA46T の使命は完了する。この際、SA46T のみを取り除くだけで良い。

(9) マルチリンク構成による冗長構成が可能

スタブネットワークとバックボーンネットワークを複数の SA46T を経由して接続出来る。この際の SA46T の設定は全く同じものとなる。どちらのリンクが用いられるかは、ルーティングプロトコルが決める。ECMP(Equal Cost Multi Path) などによる負荷分散も可能である。なお、従来のカプセル化技術では冗長構成を取るのは不可能である。

(10) 既存ネットワークへの導入も容易

DHCP(Dynamic Host Configuration Protocol) などのブロードキャストトラフィックの収容が無ければ[1]、既存ネットワークを変更せず、SA46T を追加するのみで、バックボーンネットワークの IPv6 only 化が可能となる。

(11) トラブルシューティングの容易さ

SA46T でカプセル化された IPv6 パケットの IPv6 アドレスを見れば、そこに IPv4 アドレスがそのまま格納されているので、判り易く、トラブルシューティングしやすい。

5 SA46T 技術の標準化提案と実証

本技術を通じ、広くインターネットコミュニティに貢献すべく、IETF に標準化提案を行っている。

IETF での標準化の条件に、(1) 実装の存在、(2) 独立した実装間で相互接続できること、がある [8]。IETF での提案の反応を踏まえ、これら条件を満たすための、実装の開発や実証実験等を進めている。

5.1 標準化提案

IETF に、これまでに、基本仕様 [9]、SA46T グローバルアドレス [10]、適用の対象 [11]、提案の動機 [12]、マルチキャスト対応 [13] の合計 5 件の Internet Draft を提出している。

[1] スタブネットワーク内に DHCP Relay agent があるか、あるいは DHCP server があれば問題無い。

5.2 実装の開発

SA46T 技術が実装可能であることの証明と方式評価を目的に実装を開発した。

図 3 は、その実装構成を示したものである。本開発では、SA46T のカプセル化の機能を疑似ネットワークインタフェースとしてカーネル空間内に実装するアーキテクチャとした。ルーティングテーブルには、カプセル化すべき宛先 IPv4 アドレスの経路に一致した場合、カプセル化を行う疑似ネットワークインタフェースに経路選択されるようなエントリを設定する。また、デカプセル化すべき宛先 IPv6 アドレスの経路に一致した場合、デカプセル化を行う疑似ネットワークインタフェースに経路選択されるようなエントリを設定する。

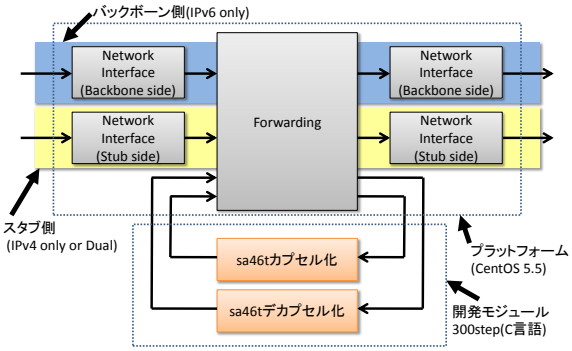

図 3: SA46T の実装構成

IPv4 パケットをカプセル化する SA46T に於いては、スタブネットワークから受信した IPv4 パケットは、まずカプセル化を行う疑似ネットワークインタフェースにルーティングされ、SA46T アドレスが設定された IPv6 パケットに変換され、その IPv6 パケットが再度ルーティングされてバックボーンネットワーク側にルーティングされる。一方、対向側の SA46T に於いては、バックボーンネットワーク側から受信した SA46T アドレスが設定された IPv6 パケットは、まずデカプセル化を行う疑似ネットワークインタフェースにルーティングされ IPv4 パケットが取り出される。この IPv4 パケットが再度ルーティングされ、スタブネットワーク側にルーティングされる。従って、カプセル化またはデカプセル化されるパケットは、SA46T 内部に於いて、それぞれ、2 度ルーティングされることになる。

スタブネットワークが Dual stack の場合は、IPv6 のルーティングエントリを、通常通り記載すれば良い。この場合、IPv6 パケットは、カプセル化またはデカプセル化を行う疑似ネットワークインタフェースにルーティングされることはない。

約 1ヶ月で実装が完成し、その規模は C 言語で 300 ステップとコンパクトなものとなった。富士通 FMV-BIBLO LOOX M/G30(Atom N450 1.66GHz、1GB メモリ、Fast Ethernet) に USB Fast Ethernet ドングルを用い 2 Ethernet 化したハードウェアに、CentOS5.5(Kernel 2.6.34) を搭載した環境で、90Mbps 以上の性能を確認した。

5.3 実証実験

SA46T 方式及び上記実装の実績獲得を目的に、3 ステップの実証実験を行った。最初は LAN 環境で確認を行い、次に、広域網での実証実験を行った。最後は、Plane を用いた、同一の IPv4 アドレスを用いた実証実験を行った。以下に、これら実証実験について述べる。

5.3.1 LAN 環境での実証実験

最初の実証実験の場として 2010 年 9 月に開催された WIDE 合宿にて実験を行った。

本実験で用いたハードウェアは、実装開発を行ったプラットフォームと同じもので、富士通 FMV-BIBLO LOOX M/G30(Atom N450 1.66GHz、1GB メモリ、 Fast Ethernet) に USB Fast Ethernet ドングルを用い 2 Ethernet 化したハードウェアに、CentOS5.5(Kernel 2.6.34) を搭載したものである。

図 4 は、この実証実験の構成と WAN 側のトラフィックを示したものである。トラフィックグラブは、プラス側がインターネットから合宿ネットワークへのトラフィック、マイナス側がその逆の合宿ネットワーク側からインターネット側のトラフィックを示している。

合宿ネットワーク全体に展開し、インターネット向けの上流 1 台と、ユーザ収容で 3 台の、合計 4 台の SA46T 搭載 PC を利用し、全ての合宿参加者が無線 LAN 経由で SA46T を介して通信する環境を構築した。DHCP を用いて IPv4 アドレス割り当てを行い、参加者は SA46T を全く意識せずに利用した。

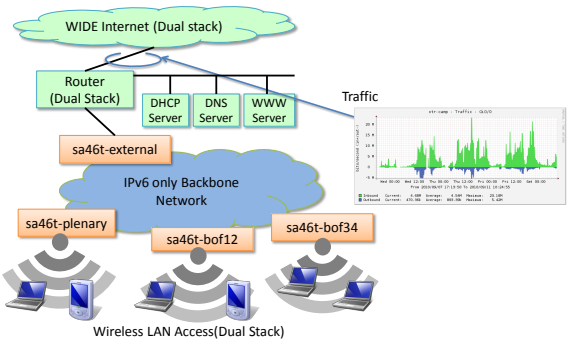

図 4: WIDE 合宿での SA46T 実証実験構成

4.5 日間の実験で、191 人が参加し、275 台のクライアントが接続し、約 2.3 億個の IPv4 パケットをカプセル化した。

5.3.2 広域網環境での実証実験

JGN2plus は、独立行政法人 情報通信研究機構 (NICT) の研究成果を実証するためのテストベッド基盤である。広域網での実績を得るため、2011 年 2 月に、JGN2plus 上で実験を行った。図 5 は、この実証実験の構成と、トラフィックを示したものである。

SA46T は、ハイビジョン映像、3D 映像及びデータの配信に用いられ、大量のトラフィックを長期に渡って運ぶ実験となった。また、はじめて OSPFv3(Open Shortest Path Fast version 3) によるルーティングを用い、さらに、BGP4+(Border Gateway Protocol 4 Plus) を用いて、所有者が異なるネットワーク間での SA46T 利用を実証した。

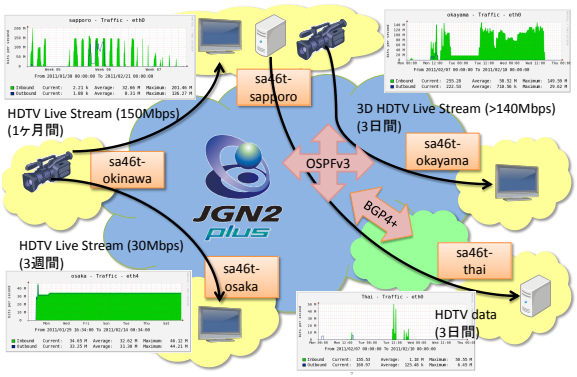

図 5: NICT JGN2plus での実証実験

Cisco UCS B5500 (Intel Xeon E5650 2.67GHz 6core 4socket、32GB メモリ、10GbE × 2) に CentOS5.5(Kernel 2.6.34) を搭載した環境で、800Mbps 以上の性能を確認した。これは、測定に用いた IA サーバの通信性能限界値である。

映像配信は 3 種、そしてデータ転送が 1 種行われた。映像配信では、(1) 沖縄から札幌に、約 150Mbps の HDTV Live Stream が 1ヶ月間にわたり配信、(2) 沖縄から大阪に、約 30Mbps の HDTV Live Stream が連続 3 週間配信、(3) 札幌から岡山に、最大 140Mbps の 3D HDTV Live Stream が 3 日間配信された。データ転送では、HTDV 映像のデータが札幌からタイ国に転送された。

5.3.3 Plane 機能の実証実験

2011 年 6 月開催の Interop Tokyo にて、SA46T の plane を用いた実験を行った。Interop ShowNet を中心に、(1) インターネットへのアクセス、(2) データセンター、(3) 映像伝送、の 3 つの適用分野を示すことを意識した実証実験兼デモンストレーションで、後者 2 つの plane で、意識的に同一のプライベートアドレスを利用した[2]。

この実証実験で、同一の IPv4 アドレスが、干渉することなく使用可能であることが実証された。

5.4 相互接続試験

上記実装はカーネル空間で実装したものであるが、これとは独立にアプリケーション空間での実装を開発した。これに加え、慶應義塾大学による実装 [16] が行われたことから、独立した実装が 3 種類存在することとなった。これらによる相互接続試験を、2012 年 3 月の WIDE 研究会にて実施した。

相互接続試験の結果は、フラグメント関連処理のサポート範囲の差が現れる結果となった [17]。フラグメント処理以外については、相互接続は成功した、なお、実用上は特に大きな問題は無かった。

[2] このデモにより、SA46T は、Best of Show Award のデモンストレーション部門に於いてグランプリを受賞した [14, 15]。

6　評価

ここでは、実装開発、実証実験、相互接続試験で得られた結果から、SA46T 技術の評価を行う。

6.1　実装可能であることの証明

SA46T の実装は現在 3 種あるが、それぞれ独立に実装されたものである。これは、SA46T が実装可能であることの証明であり、なおかつ、IETF に提出している Internet Draft を見て実装が可能な証明でもある。後者は、Internet Draft の品質が十分であることを示していると言えよう。これは、IETF 標準の、Proposed Standard レベルの品質を達成しているものと評価できる。

6.2　設定数に関する評価

本論では、既存のカプセル化技術では、設定数がべき乗で増加することを問題点であると指摘した。設定は、SA46T の実装モジュールを追加し、SA46T の疑似インタフェースを起動し、そこに SA46T アドレスの設定を行うものとなっている。

以下に、その設定例を示す。

```
insmod sa46t.ko
ifconfig sa46t0 up
ifconfig sa46t0 add 2001:3e8:0:2646:0:1:a00:100/120
```

この例は、IPv4 スタブネットワークが、10.0.1.0/24 で、SA46T プレフィックスが 2001:3e8:0:2646/64、IPv4 network plane ID が 1 の場合のものである。

これは、バックボーンネットワークに接続されている他の SA46T の台数に依存しない設定であり、設定数が少なく済むことが、机上ではなく、実際の実証実験等を通じて証明されたと言える。

6.3　相互接続性可能であることの証明

3 種ある SA46T の実装で相互接続を行ったが、実装によりフラグメンテーションのサポート範囲が異なり、その違いが試験結果として現れたが、フラグメンテーションが問題にならない範囲では、特に問題なく、相互接続が出来た。

これは、カプセル化処理そのものについては問題無く、この範囲に於いて、IETF 標準の Draft Standard レベルの品質を、ほぼ達成しているものと言えそうである。より相互接続制を高めるためにも、フラグメンテーションのサポートに関する記述を追加する予定である。

6.4　技術のシンプルさについて

評価目的の実装ではあるが、C 言語で 300 ステップで、それが、実証実験を通じて安定に動作し、なおかつ、遜色無い性能も達成できたことは、それだけ技術がシンプルであるからと言えよう。

7　まとめ

IPv4 アドレス枯渇による環境変化を述べ、この環境変化を踏まえた、IPv6 移行と IPv4 継続利用を両立する技術として SA46T を提案した。IPv6 移行を進めながら、IPv4 を IPv6 で巻き取っていくことにも利用可能であり、あるいは、IPv4 アドレス再利用を進める手段としても利用可能である。特に後者の実現に於いても、IPv6 が必要であることから、IPv6 移行に寄与するものと考えている。

SA46T によるソリューションは、NAT など IPv4-IPv6 変換技術との組み合わせも可能だが、用いない構成も可能となる。例えば VPN(Virtual Private Network) などの、IPv4-IPv6 変換技術と相性の悪い IPv4 アプリケーションの継続利用に、特に貢献できる可能性が高そうである。

SA46T はシンプルな技術であり、十分実利用可能であることを示した。様々なネットワークへの適用可能性があると考えている。広く、インターネットの健全な発展に寄与すべく、今後も標準化、実用化、そして普及に向け、取り組んでいく所存である。

謝辞

IETF への標準化提案を勧めてくださった慶應義塾大学の加藤朗教授に感謝します。また、JGN2plus での実証実験にお誘いくださった、情報通信研究機構／倉敷芸術科学大学の小林和真教授に感謝します。WIDE 合宿での実験でご支援くださった慶應

義塾大学の中村修教授及び NetPC の皆様、Interop Tokyo ShowNet NOC チームの皆様に感謝します。最後に、SA46T の実装を開発してくださった、慶應義塾大学 (当時) の中村遼氏に感謝します。

参考文献

[1] S. Deering, R. Hinden *Internet Protocol, Version 6 (IPv6) Specification* RFC1883, December 1995

[2] R. Gillogan, E. Nordmark *Transition Mechanisms for IPv6 Hosts and Routers* RFC1933, April 1996

[3] 松平直樹監修 「*IPv6* ネットワーク実践構築技法」オーム社, 2001

[4] A. Conta, S. Deering *Generic Packet Tunneling in IPv6 Specification* RFC2473, December 1998

[5] W. Sumpson *IP in IP Tunneling* RFC1853 October 1995

[6] G.Tsirtsis, P. Srisuresh *Network Address Translation - Protocol Translation (NAT-PT)* RFC2766, February 2000

[7] M. Bagnulo, P. Matthews, I. van Beijnum *Stateful NAT64: Network Address and Protocol Translation from IPv6 clients to IPv4 Servers* RFC6146, April 2011

[8] S. Bradner *The Internet Standard Process – Revison 3* RFC2016 October 1996

[9] N. Matsuhira *Stateless Automatic IPv4 over IPv6 Excapsulation / Decapsulation Technology: Specification* draft-matsuhira-sa46t-spec-05.txt Internet-Draft July, 2012

[10] N. Matsuhira *Stateless Automatic IPv4 over IPv6 Excapsulation / Decapsulation Technology: Global SA46T Address Format* draft-matsuhira-sa46t-gaddr-05.txt Internet-Draft July, 2012

[11] N. Matsuhira *Applicability of Stateless Automatic IPv4 over IPv6 Encapsulation / Decapsulation Technology (SA46T)* draft-matsuhira-sa46t-applicability-04.txt Internet Draft July, 2012

[12] N. Matsuhira *Motivation for developing Stateless Automatic IPv4 over IPv6 Encapsulation / Decapsulation Technology (SA46T)* draft-matsuhira-sa46t-motivation-02.txt Internet Draft July, 2012

[13] N. Matsuhira *SA46T Multicast Support* draft-matsuhira-sa46t-mcast-01.txt Internet Draft March, 2012

[14] *Interop Tokyo 2011* 出展報告NICT NEWS, 2011 年 6 月号, No.405 `http://www.nict.go.jp/publication/NICT-News/1106/03.html`

[15] *Interop Tokyo 2011* の「*Best of Show Award*」にてグランプリを受賞 `http://pr.fujitsu.com/jp/news/2011/06/9-1.html`

[16] `https://github.com/upa/sa46t/`

[17] 『「*IPv6* 時代の *IPv4* を考える」〜第二章〜』JANOG30 `http://www.janog.gr.jp/meeting/janog30/program/v64.html`