

仕様や実装の異なる多数の端末を IPv6 only ネットワークに接続するための課題と考察

廣海緑里^{†1} 櫛山寛章^{†2} 尾上淳^{†3} 中村修^{†4}

^{†1}株式会社インテック ^{†2}奈良先端科学技術大学院大学 ^{†3}ソニー株式会社 ^{†4}慶応義塾大学

IPv4 アドレス在庫枯渇が現実となり、多くの事業者では IPv4 アドレスの移転手続きを通じたアドレス調達を行っている。一方で将来の IPv6 導入に向けた準備も本格化している。IPv4 から IPv6 への移行や2つのプロトコルバージョンの共存技術など多くの提案とその検証がされている。最近では、IPv4 sun-setting というキーワードで IPv4 をなくすための議論も行われている。コンシューマを対象としたネットワークにおいて、IPv6 だけで運用することを前提とした場合の問題として、接続する可能性のある端末における IPv6 技術の導入状況に違いが有る。本稿では端末の問題点を明らかにすると共に端末を設定変更なしで接続するためのネットワーク側での方策を紹介する。

A study report for various user devices into IPv6 only network

Ruri Hiromi^{†1} Hiroaki Hazeyama^{†2}
Atsushi Onoe^{†3} Osamu Nakamura^{†4}

^{†1}INTEC Inc. ^{†2}Nara Institute of Science and Technology ^{†3}SONY Corporation ^{†4}Keio University

Since IPv4 address stock depletion, lots of Internet Service Operators begun to use IPv4 trading scheme for their procurement. On the other hand, they are thinking to use and preparing for IPv6 adaptation. Co-existence, exclusive, and transitional scenarios are discussed then here are several way to operate IPv6. These days we are talking about how to fade IPv4 out from the Internet. In this paper, we introduce our experiments of IPv6 only network in WIDE Project Camp and show one example of how to connect various user devices without any troublesome settings.

1. IPv6 only ネットワーク

IPv4 アドレス在庫枯渇が現実となり、多くの事業者では IPv4 アドレスの移転手続きを通じたアドレス調達を行っている。一方で将来の IPv6 導入に向けた準備も本格化している。一口に「IPv6 の導入」といっても、IPv4 から IPv6 への移行を目指すものと2つのプロトコルバージョンの共存を目指すものがあり、その実現方法も千差万別である。現在も多くの技術提案とその検証がされている。

最近の新しい動きとしては、IPv4 sun-setting というキーワードで IPv4 をなくすための議論[1]も行われている。IPv4 sun-setting には、IPv4 ベースで IPv6 を展開した場合にどのように IPv4 をなくしていくか(あるいは依存を減らしていくか) という議論と、IPv6 だけで IP ネットワークを構築・運用していくための議論に二分される。これまでの人間の利用を主軸にしたインターネットから、M2M (Machine to Machine) Communication や IoT (Internet of Things) といったセンサー類や制御機器のためのネットワークとしてのインターネットも現実となってきており、こうしたネットワークでは IPv6 だけの構成というのも十分に考えられる。また、人間の利用を前提とする既存のインターネット環境においても、2つのプロトコルバージョンを導入、運用保守するコストや移行やセキュリティ対策

の煩雑さから共存・移行期間なしで一気に IPv6 化を進めるといった提案[2]や事例[3]が出始めている。

本稿では、IPv6 だけで構成したネットワークを「IPv6 only ネットワーク」と呼び、ネットワーク構成技術と利用者であるエンドユーザ端末における課題を明らかにする。また、WIDE プロジェクト[4]が定期的に開催している合宿で行った IPv6 only ネットワーク実験の結果についても紹介する。

特に、コンシューマを対象としたネットワークを IPv6 だけで運用することを前提とした場合に、IPv6 技術の対応の違いがある端末を接続するにはどうすべきか、という問題がある。本稿では各種端末の実装状況を設定変更なしで接続するためのネットワーク側での方策を紹介する。

2. 全3回の実験概要

WIDE プロジェクトでは、年2回春と秋に5日間の研究合宿を行っている。合宿では、毎回ネットワークを準備し、ネットワーク実験を行っている。2011年9月、2012年3月と9月の3度の合宿には、IPv6 only ネットワークを構築し、合宿参加者が持ち込むノート PC をはじめとする端末の接続状況を調査した。本実験の目的と目標は、IPv6 only ネットワーク全般に関する課題や問題点の発見とそ

の対策の検討である。特に、IPv6 only ネットワークに接続できる既存のユーザ端末と、利用可能なアプリケーションの抽出と整理を行う事を目標とした。問題のある端末やアプリケーションについては、その原因や傾向の把握も調査範囲とした。

各回における IPv6 only ネットワークの構成技術を以下に整理する。IPv6 only ネットワークは、基本的にユーザアクセス部やコアネットワーク部に IPv6 アドレスを配布・利用させ、IPv4 インターネットとの接続は NAT64/DNS64[6]による変換を行う事で確保する形態をベースとした (図 1)。実験機材の調達状況により、ベースの IPv6 only ネットワークに IPv4/IPv6 共存技術を取り込む形で実験を行った。

図 1 IPv6 only ネットワーク模式図

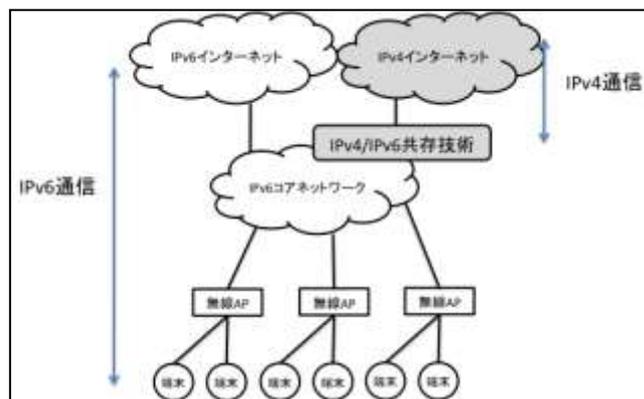


表 1 WIDE 合宿の IPv6 実験状況

2011/09	ユーザアクセス	WiFi/EAP-TLS 臨時有線 LAN
	IPv6 提供方法	RA (address prefix, router), DHCPv6 (DNS)
	IPv6 Internet との接続方法	IPv6 over L2TP
	IPv4 Internet との接続方法	4rd, SA46T, NAT64/DNS64
	参加者数	153
2012/03	ユーザアクセス	WiFi
	IPv6 提供方法	RA (address prefix, router), DHCPv6 (DNS)
	IPv6 Internet との接続方法	NGNv6 (IPoE), NGNv6 (PPPoE), IPv6 over L2TP
	IPv4 Internet との接続方法	4rd, 464XLAT, SA46T, NAT64/DNS64
	参加者数	171
2012/09	ユーザアクセス	WiFi
	IPv6 提供方法	RA (address prefix, router), DHCPv6 (DNS)
	IPv6 Internet との接続方法	NGNv6 (IPoE), IPv6 over L2TP
	IPv4 Internet との接続方法	SA46T-AT, NAT64/DNS64
	参加者数	136

(1) 2011 年秋合宿(2011 年 9 月)

1 回目の実験では、WIDE バックボーンと合宿会場の間で L2TP のトンネル接続による IPv6 接続を持ち、SA46T [5] や、NAT64 [6] / DNS64 [7] の変換技術を利用したネットワークを提供し、ユーザである合宿参加者には無線アクセスによる IPv6 接続という形で実験参加を呼び掛けた。無線アクセスでは証明書を用了 EAP/TLS が前提であったため、接続できない際に EAP/TLS の不具合と IPv6 の不具合の切り分けが難しいケースがあった。IPv6 の不具合や利用状況については、アンケート調査によって実態調査を行った。

(2) 2012 年春合宿(2012 年 3 月)

2 回目の実験では、NTT 西日本の NGNv6 サービスを利用した IPv6 アクセスネットワーク 2 種 (IPoE および PPPoE) と、従来行ってきた L2TP による WIDE バックボーンとの接続を利用した IPv6 の複数ネットワーク環境を構築した。また、IPv4 ネットワークへの接続も 464XLAT [7] を加え、4 種類の技術を時限で利用できるようにした。2012 年春の時点で実際に利用可能な IPv6 サービスと代表的な IPv4/IPv6 共存技術を揃えた実験環境を構築できた。それぞれの形態を均等に利用して評価を行う為、時間を決めて切り替えを行った。合宿参加者には組み合わせの詳細は伏せた上で無線情報 (SSID) を公開し、利用状況の推移を見た。また、従来のアンケート調査の他、ヒアリング調査も実施し、利用できないケースの問題を抽出し、その傾向を把握した。端末接続における問題と一部のアプリケーションで IPv6 対応における問題が存在する事が確認できた。

(3) 2012 年秋合宿(2012 年 9 月)

3 回目の実験では、2 回目の追調査として端末側の動向に絞った調査を実施した。合宿参加者に提供するネットワークは、IPv6 only のベース構成と、IPv4 への接続のための共存技術として SA46T を用いた構成の 2 種類を用意した。Windows OS、iOS、MacOSX、Android OS についての評価を掘り下げて行い、端末へのネットワーク接続情報の通知方法の検討とその対策についての効果測定や利用できないアプリケーションに関する具体的な種別の特定を行った。

3. 実験から得られた課題

3 回の実験で、多種類の端末を接続し、利用調査を行った。

ユーザ収容部では、接続するクライアント端末の OS における IPv6 の実装がそれぞれ違っており、IPv6 対応済みであったとしてもネットワーク側で使う IPv6 の機能の組み合わせによってはうまく接続できない事があることがわかった。また、ユーザインタフェースの実装の問題もあり、トラブルシュートが困難になるというケースが多々あった。

IPv6 only ネットワークにおける最大の問題は、ユーザ収容部にあると言える。3.1 で端末の問題点を詳述する。

一方、コアネットワークや IPv4 との境界部分については、安定運用という課題が存在するものの大きな問題はなかった。3.2 でネットワーク側の問題を詳述する。

アプリケーションについては、IPv6 環境で利用するためのソケット API [9] [10] が適切に利用されていれば問題は発生しないことが確認できた。3.3 で問題のあった例を取り上げる。

3.1 端末に見られる問題点

(1) 端末 OS 固有の IPv6 の実装上の問題

端末 OS における IPv6 の実装は、ベンダー毎に異なっている。利用される端末 OS の種類は数多くあり、古い OS が使われ続けている場合もある。そのため、新旧の OS をどれも同じように IPv6 only ネットワークに接続させるのは難しいことが予想される。

カタログ上は「IPv6 対応済み」となっていたとしても、ネットワークの IPv6 提供方法によってはうまく動作しないという事がわかった。例えば、IPv6 only ネットワークで利用するネームサーバ情報について、DHCPv6 により問題なく情報取得し、設定できる端末がある一方で、DHCPv6 クライアントが実装されておらず、手動でネームサーバ情報を設定しなければならない端末もあった。

また、IPv6 only ネットワークに接続させようとする IPv4 アドレスがないため自己解決アドレス [11] がつくが、IPv4 アドレスが自己解決アドレスの場合には IPv6 のアドレス自動設定の処理がされない端末 OS もあった。この端末は IPv4 のプライベートアドレスまたはグローバルアドレスが配布された状態であれば、IPv6 アドレスを自動設定で付ける事が確認できており、IPv6 単体での動作を想定していないものと思われる。同じく IPv4 の自己解決アドレスについては、ある種のスマートフォン端末において IPv6 のアドレスは使えている状況であるのに、IPv4 アドレスの取り直し時に IPv6 の設定もリセットされ、結果として定期的に接続断が発生してしまい、アプリケーションの利用に支障がでてしまうケースがあった。

2 回目の実験では、無線 AP を時限で切り替えて各種のネットワークを利用してもらった実験の性格上、ユーザは利用しやすいネットワークに移るため、アクセスポイントを選び変える事が多かった。この時、端末 OS によっては、事前に取得していたネームサーバ情報が残ってしまい、名前解決ができなくなるケースがあった。

表 2 は、端末 OS 毎の DHCPv6 の動作状況と、IPv4 の自己解決アドレスがついた場合に IPv6 のアドレス自動設定がうまくされなかったケースを抽出したものである。

表 2 端末 OS の IPv6 アドレス自動設定機構の稼働状況

OS	Ver.	RA	DHCPv6	169.254 の影響
Windows XP		0	X	
Windows Vista		0	0	
Windows 7		0	0	
Windows 8		0	0	
Mac OS X	SonwLeopard(10.6)	0	X	
Mac OS X	Lion(10.7)	0	0	
Mac OS X	Mountain Lion(10.8)	0	0	
Android	1.6	X	X	
Android	2.3.4	0	X	X
Android	2.3.5	0	X	X
Android	2.3.6	0	X	X
Android	4.0.3	0	X	X
Android	4.0.4	0	X	X
Android	4.1	0	X	X
iOS	4.3.2 JB	0	X	X
iOS	5	0	X	X
iPad iOS	5	0	0	
iPad iOS	6	0	0	
kindle	3.3	X	X	
NetBSD	5.1	0	X	
NetBSD	6.99.4	0	X	
Ubuntsu	12.0.4	0	0	

※RA, DHCPv6 は動作したものに 0、問題しないものに X

※「169.254 の影響」は、影響あったものに X

※未調査、不明なものは空欄

(2) ユーザインタフェースにおける問題

IPv6 のアドレス自動設定の機構が問題なく動作し、IPv6 only ネットワークに接続できているが、その状態をユーザインタフェースで確認できない OS がある。WiFi 設定情報や IP アドレス情報をみると IPv4 に関しては表示されるが、IPv6 に関しては設定状況が確認できないため、IPv6 アドレスを直接入力できるブラウザなどのアプリケーションを動作させ、問題なければ IPv6 は動く、という判断をせざるを得ないものもあった。

IPv6 アドレスを入力して静的に設定を行おうとした場合に、「無効な IP アドレス」といった判定でエラーになり、入力が受け付けられないものもあった。入力画面で IPv6

アドレスを入力すると「無効な IP アドレス」と認識される端末は多くあり、IPv6 スタックの実装はされているが、ユーザインタフェースは IPv4 依存のまま残されるという状況がある。また、WiFi などの稼働状況、通信状況を示すアイコンについても、デュアルスタックや IPv4 ネットワークでは正しい状況が表示されているが、IPv6 only ネットワークに接続している場合には「圏外」や「(データ取得中を示す) アイコンがグルグルと回る状況」となるものもあった。表 3 は、IPv6 の状況表示がされなかった OS と設定(入力)ができなかった OS である。

(1) で取り上げた IPv6 のアドレス自動設定機構が動作せず、ユーザインタフェースの IPv6 対応がされていない OS では静的な設定もできないため、IPv6 only ネットワークには接続できないことになる。

表 3 端末 OS のユーザインタフェースの対応状況

OS	Ver.	UI (表示)	UI (設定)
Android	2.3.4	X	X
Android	2.3.6	X	X
iOS	5	X	X
iPAD iOS	5	X	X

(3) その他の問題

PC ベンダーによっては、無線のアクセスコントローラなどをベンダー固有に製作し、提供している。一部のベンダーが提供する無線アクセスコントローラでは、IPv4 のプロパティを OFF にすることによって IPv6 only ネットワークに接続できるようになったという報告があった。IPv4 と IPv6 それぞれのスタックを選択して使うような、デュアルスタックではない仕様の無線コントローラである場合、IPv6 only ネットワークに接続するためには、上述のように IPv4 をオフにする必要がでてくる。今回、報告の有ったコントローラを表 4 にまとめた。

表 4 報告のあった無線アクセスコントローラ

コントローラ名	Ver.	端末 OS
Lenovo Access Connections	5.72	Windows XP
Lenovo Access Connections	5.85 83C7711WW	Windows 7
—	—	Ubuntu 11.10

3.2 ネットワーク側の問題点

IPv6 only ネットワークにおける問題は、Jari Arkko 等による先行研究[11]で明らかにされている「DNS64 における名前解決のエラー」などが顕著な問題として残っている。

全 3 回の実験では、DNS64 サーバが受け付けたクライアントからのクエリを、BIND9 と Unbound という異なる実装の名前解決機構にフォワードし、それぞれの応答の比較検証を実施した。これにより、A レコードが存在するにも関わらず A レコードの問い合わせにフォールバックしないケースについての具体的な発生要因を特定した。実験では、これらの発生要因である不正な NameError や lame delegation についての対応策を施すことで、暫定的な対処を行う事ができた。

オープンソースの DNS64 や Unbound のサーバプログラム運用時には、プロセスが停止してしまうのを防ぐために、定期的に監視し、起動させる必要があった。スケーラビリティと安定稼働も課題にあげられる。

3.3 アプリケーションにおける問題点

明らかに動作しないアプリケーションとして、音声系アプリケーション(skype など)や VPN アプリケーションがある。音声系アプリケーションには、それ自体にシグナリングや端末間のルーティングの仕組みなど管理プロトコルを備えている事があり、その一部は IPv4 アドレスと密接に関連しているようである[8]。また、単一のクライアントアプリケーションとしての対応の他に、スーパーノードなど含めたシステム全体としての対応が必要であると考えられる。IPv6 対応がされないアプリケーションでは、アドレスファミリに依存しないプログラムコードにする以前に、デュアルスタック化によるセキュリティ上の問題（例えば、意図しないプロトコル中継や踏み台化）への懸念から、何をどこまで対応させるべきかの検討が必要であり、簡単には対応できない事情もあると思われる。なお、一部の VPN クライアントには IPv6 対応版が存在するが、接続先のサーバ側に IPv6 アドレスが付与されていない場合、アプリケーションとしての動作が不安定になるなど運用面と仕様面での対応が必要なケースも発見できた。

昨今では、ブラウザソフトなどで本体のプログラムとは別に機能をプラグインという形で挿入できるものがある。ブラウザソフト本体は IPv6 対応されているが、プラグインが対応できていないケースもある。IPv6 未対応のプラグインによりブラウザ本体がクラッシュしてしまうケースもあった。

HTTP を使うアプリケーションの多くは IPv6 only ネットワークで問題なく利用できる。しかし、HTTP 技術をベースとしたアプリケーションであっても、dropbox(ファイルシェア)や tween (twitter クライアント)などでファイルの同期といった特定の処理だけができないものが発見された。追跡調査が必要であるが、考えられる要因としては、IPv4 アドレスのハードコーディング/認証/IP アドレスベースのクッキー等が考えられる。

なお、ここであげた問題は、問題点を探そうと努力した

上で出てきたものである。多くのアプリケーションでは、ネットワーク接続するには IP アドレスを直接扱うのではなく、ホスト名や URL を処理する形で実現されており、問題はなかった。IPv6 only ネットワークで接続できないケースのほとんどは名前解決に由来するものであり、アプリケーションの実装上の問題はないものと言える。

3.4 端末接続に対する対策と改善状況

2012 年の秋合宿では、IPv6 only ネットワークに接続できない端末の状況を、以下(a)~(d)であることを突き止めた。端末側の対応状況の違いに起因するこれら 4 つの問題について、ネットワーク側の対策による解消方法を検討した。

- (a) 端末に IPv6 のネームサーバ情報などが自動登録されないため、手動で設定する必要がある
- (b) ユーザインタフェースの IPv6 未対応により静的な設定が投入できない
- (c) IPv6 のネームサーバ情報が設定できないため名前解決ができない
- (d) IPv4 の自己解決アドレスの取得のため IPv6 通信が定期的リセットされてしまう

検討の結果、ネットワーク側に以下の(4)~(8)の追加設定を順次行ない、接続状況の変化を観測した。

- (1) NAT64/DNS64
- (2) DHCPv6 による DNS(v6)設定
- (3) DHCPv4 によるプライベート IPv4 設定
- (4) DHCPv4 で、DNS(v4)とデフォルトルータを返すようにした（このルータは実際にはフォワードしない）
- (5) DNS(v4) は IPv4 ローカルネット上に配置
- (6) DNS(v4, v6) で "A" フィルターを実装
- (7) すべての "A" に NODATA を返す (IPv4, IPv6 どちらのトランスポート利用に関わらず)
- (8) AAAA はそのまま上流の DNS64 に転送

新しい設定下では、MacOS、Andoroid、iOS といった端末 OS について IPv6 only ネットワークへの接続と利用ができるように改善された事を確認できた。

3.5 今後の課題

ここまで見てきたように、IPv6 only ネットワークで実験をしてみると、ネットワークやアプリケーションの IPv6 対応については概ね問題がない事がわかった。その一方で、端末の IPv6 に関する実装状況には大きな違いがある事がわかった。ユーザインタフェースについては端末ベンダー/OS ベンダーの開発に依存する部分であるため、対処は難しいが、ネットワーク情報をモニターするアプリケーショ

ンで、IPv6 対応しているものを利用することで確認できる。設定については、ネットワーク側で工夫することで対処できる面がある事もわかった。

但し、3 回目の実験での対処は、IPv6 only ネットワークにユーザ端末を接続するというシンプルな状況を想定したものであり、限定的な利用を強いる事のできるコンシューマネットワークなどの環境向けのものである。エンタープライズネットワークなど IPv4 も並行してユーザ接続に利用しなければいけないような混在環境においても、ゼロコンフィグで端末に提供できるかどうかは、状況分析や問題解析と検討が必要であり、今後の課題である。

今回の対処は、一つのワークアラウンドであり、有効性についてもっと多くの機種で検証するなど、さらなる調査が必要である。また、実装内容と方法についてもブラッシュアップが必要であると考ええる。さらに、セキュリティに関する考慮も必要である。

加えて、実際の商用サービスなどにおいては設計上、構成上の問題がないかを吟味する必要がある。

「IPv6 only ネットワーク・サービス」としてサービス定義するのであれば、そこに接続される端末は、本来 IPv4 の A レコードのクエリを出すべきではないかもしれない。その場合は、端末メーカーの協力の下、本来あるべき動作を定義し、それにむけた対処を行うべきである。

4. IPv6 ネットワークの本格導入に向けて

今回の実験では対象外とした点も幾つかある。例えば、マルチインターフェース・マルチプレフィックス時の最適な挙動の検討や、それに派生した携帯網との連動性への対応があげられる。主に、IETF (Internet Engineering Task Force) における MIF(マルチインタフェース)や 6man (maintenance)といったワーキンググループ[10]での問題意識に共通する部分があり、これらのワーキンググループでの問題設定を注視し、あるいは連携しつつ、国内外に役立つ研究と提案を推進していく予定である。

昨今のネットワーク・サービスにおいては「責任分界点」という言葉は聞かれなくなっている。ユーザからみた場合に、ネットワークとネットワーク、ネットワークとサービスの繋ぎ目にある「責任分界点」を意識する必要なく、一気通貫でサービスを利用できるのは良いことだ。しかし、現実には「サービスの購入」に関してワンストップ化されているばかりで、問題発生時の対応については事業者間でたらい回しになるという実情もある。IPv6 の導入により、ネットワークとサービスの設計について見直しがなされる際には、改めてインターネットというものが共通プロトコルを用いたネットワーク間の相互接続網ということを意識し、「問題発生時の対応まで含めた「あるべき姿」と「その際の責任分界点」や「切り分け手法」が明確化されている

べきであると考ええる。そのための監視手法、運用支援方法などについても踏み込んで実用化の支援をしていきたいと考えている。

5. 謝辞

最後に、本調査の場を提供して下さった WIDE プロジェクト、そして何よりも貴重な情報提供に協力して下さった合宿参加者に感謝する。とりわけ、調査方法の提案から鋭い問題指摘と対処の検討にご協力下さった以下の皆様には、多大なる感謝の意を表明いたします。

新 善文さん、新 麗さん、吉川 典史さん、石原 知洋さん、上野 幸杜さん、中村 遼さん、松平 直樹さん

参考文献

- [1] IPv4 sun-setting, http://transition.fcc.gov/oet/tac/tacdocs/meeting92711/Sun-Setting_the_PSTN_Paper_V03.docx, <http://datatracker.ietf.org/wg/sunset4/charter/>
- [2] Tore Anderson, The case for IPv6-only data centres, RIPE64, https://ripe64.ripe.net/presentations/67-20120417-RIPE64-The_Case_for_IPv6_Only_Data_Centres.pdf
- [3] Sony の IPv6 導入事例, http://www.soumu.go.jp/main_content/000124959.pdf, http://www.soumu.go.jp/menu_sosiki/kenkyu/02kiban04_03000054.html
- [4] WIDE プロジェクト, <http://www.wide.ad.jp/>
- [5] SA46T, N.Matsuhira, Stateless Automatic IPv4 over IPv6 Encapsulation / Decapsulation Technology: Specification, <https://datatracker.ietf.org/doc/draft-matsuhira-sa46t-spec/>
- [6] NAT64 [RFC6146], M. Bagnulo , P. Matthews , I. van Beijnum , Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers, <https://datatracker.ietf.org/doc/rfc6146/>
- [7] DNS64[RFC6147], M. Bagnulo , A. Sullivan , P. Matthews, I. van Beijnum, DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers <https://datatracker.ietf.org/doc/rfc6147/>
- [8] 464XLAT, M.Mawatari, M. Kawashima, C. Byrne, 464XLAT: Combination of Stateful and Stateless Translation , <https://datatracker.ietf.org/doc/draft-ietf-v6ops-464xlat/>
- [9] R. Gilligan, et al. Basic Socket Interface Extensions for IPv6(RFC3493)
- [10] W. Stevens, et al. Advanced Sockets Application Program Interface (API) for IPv6(RFC3542)
- [11] Fabrice DESCLAUX and Kostya KORTCHINSKY, Vanilla Skype part 2, RECON2006, <http://www.recon.cx/en/ff/vskype-part2.pdf>
- [12] S. Cheshire, B. Aboba and E. Guttman, Dynamic Configuration of IPv4 Link-Local Addresses(RFC3927), <http://tools.ietf.org/html/rfc3927>
- [13] IETF MIF,6MAN working group, <http://datatracker.ietf.org/wg/>.
- [14] J. Arkko, A. Keranen, Experiences from an IPv6-Only Network(RFC6586), <http://datatracker.ietf.org/doc/rfc6586/>

付録 1

WiMAX Router	
Nintendo DS / 3DS	

調査を実施した端末 OS

OS	Ver.
Windows XP	
Windows Vista	
Windows 7	
Windows 8	
Mac OS X	SonwLeopard(10.6)
Mac OS X	Lion(10.7)
Mac OS X	Mountain Lion(10.8)
Windows CE	
Windows Mobile	6.1
Windows Mobile	6.5
Android	
Android	1.6
Android	2.2
Android	2.3
Android	2.3.4
Android	2.3.5
Android	2.3.6
Android	4.0.2
Android	4.0.3
Android	4.1
Arch Linux	
blackberry	
CentOS	
Debian	
FreeBSD	
iOS	4.3.2 JB
iOS	5
iOS	6
iPad iOS	
iPod Touch iOS	
kindle	3.3
Linux Open WRT 2.7.39	
NetBSD	5.1
NetBSD	6.99.4
Nokia 5800	
PSP	
PS Vita	
3DS	
Ubuntsu	4.0.3
Ubuntsu	11.10
Ubuntsu	12.0.4