デジタルプラクティス　Vol.9 No.4 (Oct. 2018)

推薦投稿論文

クライアントOSのIPv6実装検証から見たネットワーク運用における課題の考察

北口 善明[1]　近堂 徹[2]　鈴田 伊知郎[3]　小林 貴之[4]　前野 譲二[5]

[1]東京工業大学　[2]広島大学　[3]アラクサラネットワークス（株）　[4]日本大学　[5]早稲田大学

各オペレーティングシステム（OS）のIPv6実装が進んだことで，IPv6ネットワークが提供されればクライアント端末はIPv6通信が優先となるケースが増えてきている．一方で，現在IPアドレス自動設定の手法としてはRAやDHCPv6を用いたステートレス・ステートフル設定がRFCで策定され，その組合せによってはクライアントの意図しない挙動を誘発するだけでなく，ネットワーク環境へ与える影響も考慮しなければならない．
本稿では，各種クライアントOSにおけるIPv6実装状況の検証結果を報告するとともに，これらがネットワーク運用管理に与える影響について考察する．

1. はじめに

IPv6は，現行のインターネットプロトコルであるIPv4におけるアドレス枯渇問題を解消することを目的として，1995年に最初の仕様がRFCにて策定されて以降，20年以上経た現在においても継続的な仕様更新・追加がなされている．ほとんどのネットワーク機器やクライアントおよびサーバ用OSでは，IPv6のプロトコルスタックがすでに実装され，多くのネットワークサービスもIPv6に対応している．特にGoogleやFacebook等のハイパージャイアントを中心にIPv6化が完了しており，日本においても約20%以上のトラフィックがIPv6通信となる状況となっている[1]．

その一方でインターネットにおける多くの通信は未だIPv4であり，IPv6に対応していないネットワークやサービスのほうが優勢な状況のままである．このIPv6対応が進まない原因としては，IPv6がIPv4機器との直接通信するための互換性を持っていないことに起因する「IPv6対応のコスト増」が考えられる．IPv6に対応することは，IPv4と別のIPv6ネットワークを二重に運用することを意味しており，ネットワーク運用コストが単純に増加する．また，利用者がIPv6を積極的に利用するメリットがないことから，追加コストをかけてまでIPv6を利用する動機が弱く，ネットワーク運用コストの増加分を回収する術がない点が大きいと考えられる．しかし，インフラストラクチャとして位置付けられるようになっている現在のインターネットにおいて，IPv6導入に向けた課題を正しく把握し，効率的に運用するための知識・経験を養っていくことが必要となっている．

そこで，本稿では，クライアントOSのIPv6実装（2017年11月時点）に焦点を当て，IPv6アドレス自動設定やIPv6オンリーネットワーク（ネットワークをIPv6のみで構成し，トランスレータ機能などを用いてIPv4をサービスの1つとして提供するネットワーク環境）への対応などの実装状況を検証した結果を報告する．また，この検証結果を元に，現時点におけるネットワーク運用管理における課題を考察し，今後のIPv6対応ネットワークの普及促進に向けて提言を行う．

本稿の構成は以下の通りである．まず第2章において，評価対象とするIPv6アドレス自動設定およびアドレス選択手法に関して，RFCにおける仕様の整理を行う．次に，IPv6時代におけるネットワーク形態の分類を第3章にて定義し，第4章にて，クライアントOSにおける実装検証実験の結果をまとめる．実装検証実験を受け，第5章にて，今後のネットワーク運用管理への影響に関して考察し，IPv6導入に際して必要となる知見と問題点について述べる．

2. RFCにおけるIPv6アドレス自動設定とアドレス選択関連の仕様

IPv6では，クライアント端末（以下，端末）がネットワークに接続した際にIPアドレスを自動設定する手法が二種類存在する．一方がルータ広告（RA：Router Advertisement）に含まれるプレフィックス情報を利用するSLAAC（StateLess Address AutoConfiguration）[2]で，もう1つがDHCPのIPv6版であるDHCPv6[3]を用いる手法である．さらにDHCPv6には，IPv4におけるDHCPと同様にIPアドレスの割り当てまでを実施するステートフルモードと，IPアドレス以外のネットワーク情報（DNSサーバ情報など）を配布するだけのステートレスモードが定義されている．これらのアドレス自動設定は，ルータから送信されるRAに含まれる3つのフラグにより制御される仕様[4]となっており，それぞれ**表1**に示す動作を制御している．

表1 RAのフラグとアドレス自動設定の関係

Flag	Function	Description
A	SLAAC	Autonomous address configuration flag: プレフィックス情報オプションに含まれるフラグで，このフラグが 1 の場合に端末は当該プレフィックスを利用した IPv6 アドレス設定（SLAAC）が可能となり端末にてアドレスが生成される．
O	ステートレス DHCPv6	Other configuration flag: このフラグが 1 の場合，当該ネットワーク環境にて IPv6 アドレス設定以外のネットワーク情報を DHCPv6 が配布する用意があることを示しており，端末はステートレス DHCPv6 のプロセスを起動する．
M	ステートフル DHCPv6	Managed address configuration flag: このフラグが 1 の場合，当該ネットワークでは DHCPv6 にて IPv6 アドレスを払い出す用意があることを示しており，端末はステートフル DHCPv6 のプロセスを起動して IPv6 アドレス設定を実施する．なお，ステートフル DHCPv6 では，IPv6 アドレス以外の情報も配布されるため，M フラグが利用される場合には O フラグの値は無視可能とされている．

SLAACは，サブネットにおけるアドレス空間を潤沢に利用できるIPv6であるが故に定義されたもので，DHCPv6と異なりサーバレスでのアドレス自動設定が可能となる利点を持っている．その反面，DHCPv6におけるリースファイルのようなアドレスの利用歴を保持する機能はないため，アドレスのトレーサビリティを確保することが困難となる．本節では，IPv6運用の要となるIPv6アドレス自動設定（SLAAC, DHCPv6，DNSサーバ情報配布）とアドレス選択に関連する仕様についてRFCを元に整理する．

2.1 SLAAC

IPv6アドレスは128ビットからなり，サブネットプレフィックスとインタフェースID（以下，IID）に分かれている[5]．SLAACで扱うプレフィックス長はRAにより設定されるが，現時点では64ビットが標準的な値であるため，IIDも64ビット（128 - 64）であることが一般的である．

図1に典型的なクライアントOS起動時のパケットフローを示す．SLAACでは，まず端末は64ビットのIIDを生成し，fe80::で始まるリンクローカルアドレスを設定する．このリンクローカルアドレスを用いて，端末はRAを促すルータ要請（RS：Router Solicitation）をルータにマルチキャストで送信し，ルータからのRAを受信する（図1 ①）．端末は，RAに含まれるプレフィックス情報からプレフィックス長を取得し，Aフラグが0でない限りそのプレフィックス情報をサブネットプレフィックスとしてIIDと組み合わせることでIPv6アドレスを得ることができる．この時端末は，生成したアドレスの重複検知（DAD：Duplicate Address Detection）を実施し，ネットワーク内でアドレスが重複していないか確認する（図1 ②）．また，RAの送信元アドレスが端末のデフォルトルートとして設定され，これにより端末はIPv6通信が可能となる．

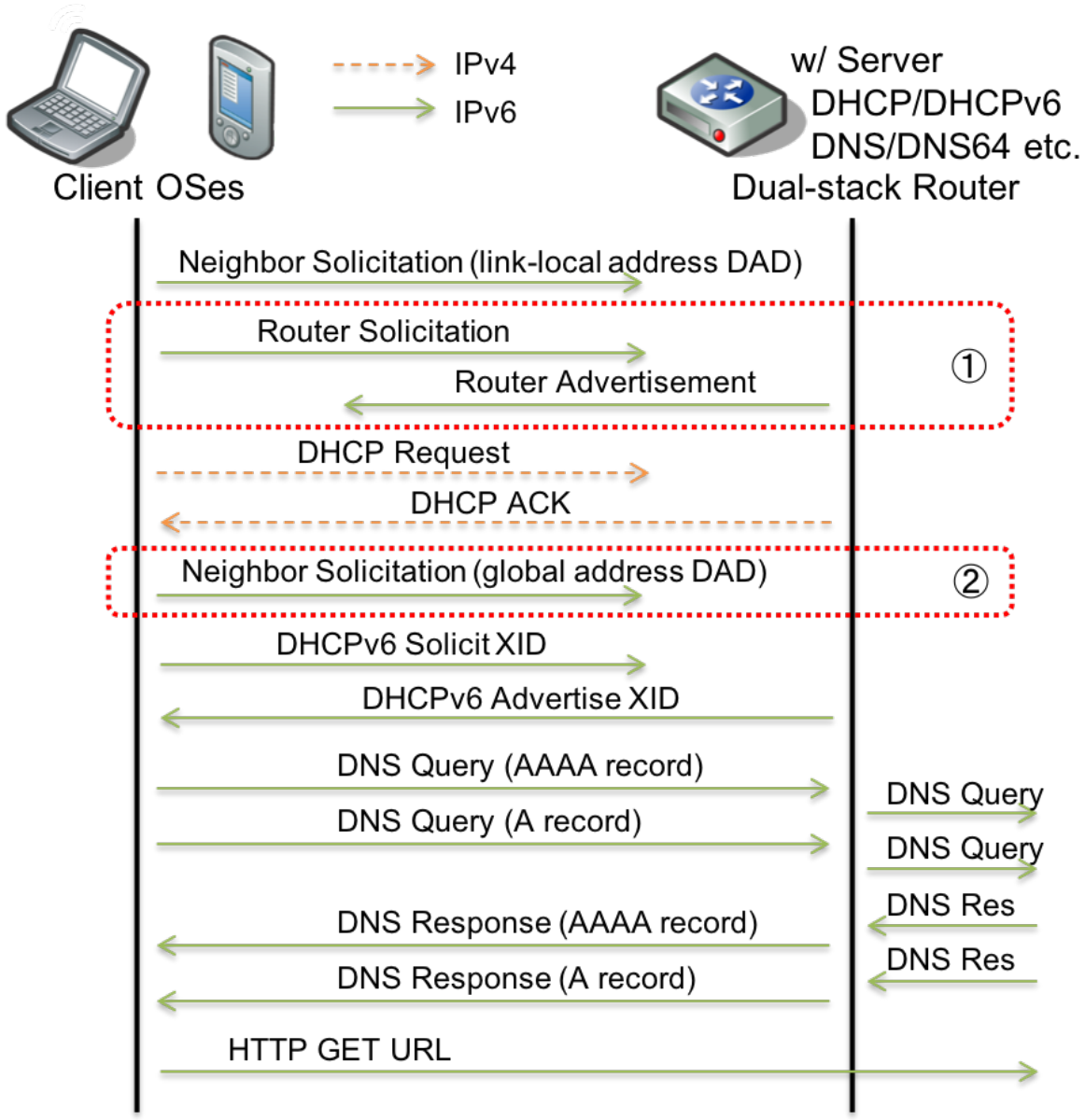

図1　典型的なクライアントOS起動時のパケットフロー
*矢印が途中で止まっているものはマルチキャスト通信を意味している

IIDの生成方法として，初期の仕様ではMACアドレスを元にしたModified EUI-64形式[5]が利用される．Modified EUI-64形式は，重複しないという利点があるが，常に同じ値であることからセキュリティ（ハードウェアを特定できるためハードウェアの脆弱性を突かれる恐れがあ

る点）とプライバシ（端末の行動履歴の推定が可能になる点）に関する懸念が指摘されていた[6].

そこでプライバシ対策として，IIDをランダムに生成し定期的に更新するプライバシ拡張アドレス[7]が2007年に策定され，Modified EUI-64形式によるアドレスに追加する形で利用されることとなった．このプライバシ拡張アドレスは，ほとんどのクライアントOSにて実装されており，デフォルトで利用されている.

また，Modified EUI-64形式によるアドレスに対するセキュリティ上の問題に対しては，MACアドレスの推測を不可能にし，プレフィックス情報が変化しない限り固定となるIIDの生成方法（Semantically Opaque IID）[8]が2014年に規定されている.

2.2 DHCPv6

DHCPv6は，前述したようにRAのフラグにより端末上での動作が2種類規定されている.

ステートレスDHCPv6は，RAのOフラグが設定されたRAを端末が受信することで起動され，IPv6アドレス以外のネットワーク情報をDHCPv6により取得する．このモードではステートレスと示されているように，DHCPv6サーバは管理情報を保持していないため導入が容易という利点がある.

ステートフルDHCPv6は，IPv4のDHCPのようにIPv6アドレス設定とネットワーク情報の配布をDHCPv6で実施するモードで，Mフラグが設定されたRAを端末が受信することで利用される．ステートフルDHCPv6で設定されるIPv6アドレスにはプレフィックス情報がなく，加えて，デフォルトルートを設定する機能もDHCPv6には規定されていないため，インターネット接続のためにはRAとの併用が必須となり，DHCPv6単独での利用は事実上不可能である．この点がIPv4のDHCPと異なる点である.

2.3 DNSサーバ情報の配布手法

IPv6アドレスの自動設定において，IPv6仕様策定の初期から議論が繰り返されているものがDNSサーバ情報の配布手法である．一般的に，インターネットにおいて通信を行う場合，アプリケーションではIPアドレスではなくドメイン名を利用するため，ドメイン名からIPアドレスを取得するDNSサーバの設定が必要不可欠である.

初期のIPv6仕様では，DNSサーバ情報の配布はIPレイヤの範疇ではないため，基本仕様には含めない設計であった．そのため，DNSサーバのIPv6アドレスを端末に自動設定する手法として，DHCPv6しか規定されていなかったが，RAにてDNSサーバアドレスを通知するオプション機能（RDNSS[☆1]オプション）[9]が2010年[☆2]に追加規定されたため，ステートレスDHCPv6が必須ではなくなった．さらに，RAのRDNSSオプションは2017年に必須機能に格上げされている[9].

このRAの仕様と並行してDHCPv6単独でのIPv6自動アドレス設定の仕様化議論も展開されたが，最終的にはプレフィックス長とデフォルトルートの設定機能追加は見送られた経緯がある.

2.4 デフォルトアドレス選択

IPv6では，同一セグメント内通信に用いられるリンクローカルアドレスとグローバル通信のためのグローバルIPv6アドレスが利用されるため，1つのインタフェースに複数のアドレスが設定される．また，プライバシ拡張アドレスなどの追加により，通信に利用するアドレスを選択するルールが必要となった．そこで，IPv4アドレスも含めた形で通信に用いるアドレス選択の仕組みが2003年に規定され，2012年には仕様の更新が行われた[10]．

文献[10]におけるルールに則ると，SLAACとDHCPv6にてIPv6アドレスを設定する場合，Rule 7による一時アドレス優先によりプライバシ拡張アドレスが選ばれ，DHCPv6によるIPv6アドレスなどは基本的に通信に利用されないことになる．宛先アドレス選択においても同様のルールが規定されており，DNSでの名前解決を行うリゾルバにて複数のIPv6アドレスが解決された場合の順位付けに利用される．このポリシーテーブルの値は手動で変更することができるだけでなく，DHCPv6の拡張オプション[11]を用いることでネットワーク側からの制御も可能となるが，後者に関しては現時点においてサーバおよびクライアントアプリケーションにおける実装を確認できていない．

3. IPv6導入におけるネットワーク形態の整理

第2章で述べたように，IPv6の自動アドレス設定手法として，SLAAC＋RDNSSオプション，SLAAC＋ステートフルDHCPv6などの複数のパターンが考えられる．また，IPv6を導入する際に用いられる一般的なデュアルスタックでは，IPv4とIPv6双方のネットワーク運用が必要となり，ネットワーク障害が発生した際の問題切り分けが複雑になること[12]や，ネットワーク運用に際しても双方のプロトコル監視が必要となるため，運用コストやセキュリティリスクの増加も課題となる．そのため，IPv6オンリーネットワークの利用も現実的なものとなっている．

以上のことから，IPv4/IPv6通信環境を提供するネットワーク形態を整理し，**表2**にまとめた．IPv4とIPv6とをそれぞれ提供する構成と，IPv4をトランスレーションで提供する構成に大別できる．IPv6のアドレス設定に関しては，RAにおける冗長なフラグ構成を排除し，IPv6アドレスやネットワーク情報をRAもしくはDHCPv6のどちらかで提供する形態のみとしている．また，クライアントOSがデュアルスタック環境に接続した際（表2におけるType2のケース），Web通信を開始するまでの典型的なパケットフローを図1に示す．このように，自動アドレス設定などの必要最低限の通信フローであっても，IPv4のみ提供する場合と比較して複雑な動作となっていることが分かる．

表2 IPv4/IPv6 通信環境を提供するネットワーク形態

Type	IPv4	IPv6		
		RA flag, opt.	Address	DNS Server
0	DHCP	A	RA	IPv4 only
1		A, RDNSS	RA	RA
2		A, O	RA	DHCPv6
3		M	DHCPv6	DHCPv6
4	DNS64/ NAT64	A, RDNSS	RA	RA
5		A, O	RA	DHCPv6
6		M	DHCPv6	DHCPv6

IPv6オンリーネットワークではトランスレータ機能が必要と述べたが，このトランスレータ機能としてはDNS64[13]とNAT64[14]を組み合わせた仕組みがデファクトスタンダートとなっている．これは，Apple社がiOS用アプリケーションの審査基準に，このDNS64/NAT64を用いたIPv6オンリーネットワークでの動作確認を2016年6月より追加している[15]ことからも見て取れる．DNS64/NAT64では，DNS64がDNSの名前解決時にIPv4アドレスを特別なIPv6プレフィックス（デフォルトで64:ff9b::/96が利用される[16]）で合成してIPv6アドレスをDNS64が回答し，この特別なアドレス宛のパケットをNAT64にてアドレス変換することでトランスレーションを実現している．したがって，IPv4アドレスしかないFQDN（この確認用に“ipv4only.arpa”が定義されている）の名前解決においてIPv6アドレスを得ることができれば，DNS64/NAT64環境であることと，利用されている特別なIPv6プレフィックスを確認できる．

4. クライアントOSの実装検証実験

IPv6における自動アドレス設定は複雑な仕様であることに加え，少しずつ追加・改良されてきたこともあり，すべてのクライアントOSにおいて仕様が均質に実装されていない．そのため，RAに設定されるフラグによる挙動がクライアントOSごとに異なることが指摘された[17]．これらを受け，2017年11月時点における主要なクライアントOSのIPv6自動アドレス設定の挙動と，IPv6オンリーネットワークにおける動作検証を行った．**図2**に検証環境におけるネットワーク構成を示す．

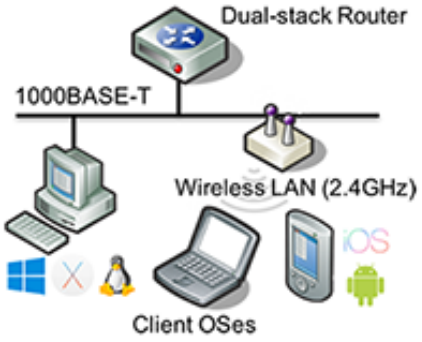

図2　検証環境のネットワーク構成

4.1　検証環境

今回の検証を進めるために，さまざまなIPv6自動アドレス設定環境を提供するためのルータはLinux OSを利用して構築した．今回は，デバイスとしてIntel® NUC（D34010WYK）を利用して構築しており，**表3**に利用したアプリケーションとバージョン情報をまとめる．また，スマートフォン端末の検証も可能とするため，無線LANアクセスポイントとしてAruba IAP-205を別途準備した．

クライアントOS検証に用いたクライアントOSのバージョン情報を**表4**にそれぞれ記す．参考までに，各OSのリリース日とハードウェア情報も記載している．検証を実施する際には，ルータにおける設定を変更した後，下記の手順にてクライアントOSの挙動データを取得した．

表3 利用したルータ用アプリケーションのバージョン

Application	Version	Purpose
Ubuntu 16.04	16.04.3	OS
linux-image	4.4.0-93.116	kernel
radvd	2.11-1	RA
isc-dhcp-server	4.3.3-5ubuntu12.7	DHCP/DHCPv6
unbound	1.5.8-1ubuntu1	DNS/DNS64
jool	3.5.4	NAT64

表4 検証に用いたクライアントOS のバージョン
*CentOS, Fedora, Ubuntu, Raspbianはkernelバージョンを記している

OS	Version	Release	Hardware
Windows 7	6.1(build7601 SP1)	2011/02/22	VM
Windows 8.1	6.3(build9600)	2013/10/18	ThinkPad Tablet2
Windows 10	1703(15063.726)	2017/11/14	Surface Pro 4
OS X 10.10	10.10.5	2015/08/13	VM
OS X 10.11	10.11.6	2016/07/18	VM
macOS 10.12	10.12.6	2017/07/19	MacBook Pro
macOS 10.13	10.13.1	2017/10/31	MacBook Pro
iOS 9	9.3.5	2016/08/26	iPad2
iOS 10	10.3.3	2017/07/19	iPad Air
iOS 11	11.1.2	2017/11/18	iPhone 7
Android 4	4.4.4	2014/06/19	Xperia Z Ultra
Android 5	5.1.1	2015/04/21	Nexus 7
Android 6	6.0.1	2015/12/07	Nexus 792013
Android 7	7.1.1	2016/12/05	Xperia
Android 8	8.0.0	2017/08/21	Nexus 5X
ChromeOS	60.0.3112.114	2017/09/14	Chrome Book
CentOS 7	3.10.0-693.2.2.el7	2017/10/01	VM
Fedora 26	4.12.14-300.fc26	2017/09/07	VM
Ubuntu 16.04	4.4.0-96 (16.04.3)	2017/10/01	VM
Raspbian	4.9.41-v7 (Sep.2017)	2017/10/01	Raspberry Pi 3

1．無線LAN／有線LANインタフェースの停止
2．DNSキャッシュデータの削除
　・Windows：ipconfig /flushdns or 再起動
　・OS X/macOS：sudo dscacheutil －flushcache
　・iOS, Android, ChromeOS：再起動
　・Linux：キャッシュしないため対処不要
3．ルータにおけるパケットキャプチャの開始
4．無線LAN／有線LANインタフェースの有効化
5．IPv4およびIPv6 Webサーバへの接続

4.2　検証結果と考察

表5に，各クライアントOSにおけるSLAACで用いるIIDの生成手法，RAにおけるRDNSSオプションの実装，DHCPv6の実装およびIPv6オンリーネットワーク環境での検証結果をまとめる．IIDの生成手法の記載は，

- RFC 4291：Modified EUI-64 IID
- RFC 7217：Semantically Opaque IID
- Microsoft：Fixed IID（Microsoft独自実装）

をそれぞれ意味している．Microsoft社のFixed IIDは，Modified EUI-64 IIDとは異なるが，インタフェースごとに固定のIIDとなっており，生成方法は確認できていない．

表5 クライアントOSごとのアドレス設定における実装
*動作確認ができたものを”○”，できなかったものを”×”としている

OS	SLAAC	RA DNS	DHCPv6	v6only
Windows 7	Microsoft	×	○	○
Windows 8.1	Microsoft	×	○	○
Windows 10	Microsoft	○	○	○
OS X 10.10	RFC 4291	○	○	○
OS X 10.11	RFC 4291	○	○	○
macOS 10.12	RFC 7217	○	○	○
macOS 10.13	RFC 7217	○	○	○
iOS 9	RFC 7217	○	○	○
iOS 10	RFC 7217	○	○	○
iOS 11	RFC 7217	○	○	○
Android 4	RFC 4291	×	×	×
Android 5	RFC 4291	○	×	○
Android 6	RFC 4291	○	×	○
Android 7	RFC 4291	○	×	○
Android 8	RFC 4291	○	×	○
ChromeOS	RFC 4291	○	×	○
CentOS 7	RFC 7217	○	○	○
Fedora 26	RFC 7217	○	○	○
Ubuntu 16.04	RFC 7217	○	○	○
Raspbian	RFC 7217	○	○	○

表6には，デュアルスタック環境とIPv6オンリーネットワーク環境で優先的に利用されるDNSサーバ（Preferred DNS Server）とDNSクエリ順（DNS query order）を記している．優先利用されるDNSサーバにおける”Random”表記は，設定されているDNSサーバからどれか1つを，複数記載されているものはそれらをランダムに選んで利用していることを示している．DNSクエリ順では，左側に記載したものが先に利用されていることを示している．

表6 クライアントOSごとのDNSサーバ利用とDNSクエリ順序
*(A)はIPv6オンリーネットワーク環境で実施されないことを表している

OS	Preferred DNS Server		DNS query order
	Dual-stack	v6only	
Windows 7	DHCPv6, IPv4	DHCPv6	(A), AAAA
Windows 8.1	DHCPv6, IPv4	DHCPv6	(A), AAAA
Windows 10	DHCPv6, IPv4	DHCPv6	(A), AAAA
OS X 10.10	Random	Random	AAAA, (A)
OS X 10.11	Random	Random	AAAA, (A)
macOS 10.12	Random	Random	AAAA, (A)
macOS 10.13	Random	Random	AAAA, (A)
iOS 9	DHCPv6	DHCPv6	AAAA, (A)
iOS 10	DHCPv6	DHCPv6	AAAA, (A)
iOS 11	DHCPv6	DHCPv6	AAAA, (A)
Android 4	IPv4	N/A	AAAA, A
Android 5	RA	RA	AAAA, A
Android 6	RA	RA	AAAA, A
Android 7	RA	RA	AAAA, A
Android 8	RA	RA	AAAA, A
ChromeOS	IPv4	RA	A, AAAA
CentOS 7	IPv4	RA	A, AAAA
Fedora 26	IPv4	RA	A, AAAA
Ubuntu 16.04	IPv4	RA	A, AAAA
Raspbian	IPv4	DHCPv6	A, AAAA

以下にクライアントOSごとに検証結果をまとめ，最後にプライバシ拡張アドレスの実装について議論する．

4.2.1 Windows OS

Windows OSの実装では，DHCPv6はステートフル，ステートレス双方への対応を確認できたが，Windows7とそれ以外の実装において差異が見られた．Windows7はRAにおけるOフラグおよびMフラグの指定がない限りDHCPv6クライアントが起動しないが，Windows8.1とWindows10ではインタフェース起動時にDHCPv6クライアントが起動される実装となっている．この実装のため，2017年3月時点におけるWindows10の実装では，起動時にDHCPv6でのアドレス設定やDNSサーバ取得ができない現象が発生していた．この問題は，コマンドプロンプトにて“ipconfig /renew6”を実行することで解決できるが，記事[18]に記載されているよう

に，Windows8以降の実装に含まれるBUGである可能性が高い．このWindows10における問題は今回検証に用いたバージョン（1703（15063.726））にて解決されていることを確認した．

Windows OSではこれまでRAにおけるRDNSSオプションを実装していなかったが，このオプションが必須機能に格上げされた[9]ことに伴い，2017年10月に配布が開始されたWindows10 Creators Updateにより実装されることになった[19]．ただし，Windows8.1以前のOSへの実装は行われていないことを確認している．Creators Updateにおいては，Androidにしか実装されていなかった464XLAT（後述）のサポートも行われている[19]．

DNSサーバの利用は，Windows OSのいずれのバージョンにおいても取得したDNSサーバをランダムに利用していた．ただし，Windows10 Creators Updateからは実装されたRAによるDNSサーバは設定されるがDHCPv6のものを優先的に利用している．また，名前解決の順番はデュアルスタックではA, AAAAクエリの順で同時に実施しており，IPv6オンリーの場合にはAクエリを省略していることを確認した．

Windows OSの実装で設定されるSLAACによるIPv6アドレスには，Modified EUI-64によるIIDが用いられておらず，この値はSemantically Opaque形式の実装とも異なり，ネットワークアドレスが変更されても同じものが利用されている．

以上のようにWindows OSは，Windows7ではRFCによる仕様に忠実な実装になっていたといえるが，Windows8.1およびWindows10ではDHCPv6がRAのフラグとは関係なく起動する実装となっている．また，RAによるRDNSSオプションがWindows10 Creators Update以降しかサポートされておらず，IPv6オンリーネットワークではDHCPv6によるDNSサーバ情報配布が必要となる．

4.2.2　OS X, macOS, iOS

Apple社によるOS X/macOSおよびiOSの実装では，今回検証に用いたいずれのバージョンにおいても，RAのRDNSSオプションとDHCPv6の実装を確認できた．ただし，設定されたDNSサーバの利用順がOS X/macOSとiOSで異なっており，前者がランダムに，後者はDHCPv6によるものを優先的に利用することが確認できた．

DNSのクエリ順はAAAA, Aクエリの順となっており，IPv6オンリーネットワークではWindows OSと同様にAクエリを省略していることを確認している．macOS 10.12以降とiOSでは，SLAACにおけるIIDの生成がModified EUI-64からSemantically Opaqueとなっている．

また，IPv6オンリーネットワークでの動作を推進しているApple社による独自の実装も確認している．先に述べたように，DNS64/NAT64ではDNS名前解決時にIPv6アドレスを合成することでトランスレーションを実現するため，IPv4リテラルアドレス（IPv4アドレスの生表記）で通信相手を指定された場合には対処できない．この問題を解決するため，Apple社は提供するAPIにてIPv4アドレスをIPv6アドレスに変換する機能を実装しており[20]，今回の検証においてもOS X 10.10を除いて検証したすべてのOSにてIPv4リテラルアドレスによるブラウジングがSafariにて可能であることを確認した．

4.2.3　Android

Androidの実装では，いずれのバージョンにおいてもDHCPv6の実装が確認できず，SLAACでのIIDはModified EUI-64形式であった．RAのRDNSSオプションに関しては，バージョン5以降で動作を確認できたが，バージョン4では確認できなかった．また，バージョン4では，インタフェースアップ後にDHCPでIPv4アドレスが取得できないとアドレス取得中状態のままインタフェースが有効にならない挙動となっており，IPv6オンリーネットワークでの利用ができない状態であった．

なお，記事[21]に記載されているように，NAT64（PLAT：Provider-side translator）と連携してIPv4通信を可能にするCLAT（Customer-side translator）がCLATdとしてAndroid 4.3以降で実装されている．NAT64（PLAT）とCLATによりIPv6ネットワークを介してIPv4通信を可能にする仕組みは464XLAT[22]として標準化されているものである．このCLATdは，NAT64配下であることを確認したのちに起動され，clatインタフェースが作成されることを確認した．ただし，今回評価に用いたバージョン4では，IPv6オンリーネットワークにてインタフェースが有効にならず，CLATdの動作を確認できなかった．

以上のように，Androidの実装ではDHCPv6が利用できない状況であり，IPv6オンリーネットワーク環境で動作させるためにはRAのRDNSSオプションが必須となる．また，Windows10 Creators Updateが登場するまではCLATの実装がされている唯一のOSで，現時点でIPv4にしか対応していないアプリケーションをIPv6オンリーネットワーク環境で利用可能となっている．

4.2.4　ChromeOS

ChromeOSはAndroidと同様にDHCPv6の実装がなされておらず，RAのRDNSSオプションでのみIPv6 DNSサーバ情報を設定できる．設定されるDNSサーバの優先順やDNSのクエリ順は，Androidとは異なりIPv4 DNSサーバ優先であったりA, AAAAクエリの順であったりと，IPv4を優先利用する状況となっていることが分かる．

4.2.5　Linux

今回検証に用いたLinuxはすべてクライアント利用に用意されたもので，GUIが提供されている版を選択している．いずれのディストリビューションにおいてもRAとDHCPv6いずれのDNSサーバ情報も利用可能であったが，ChromeOS同様，DNS利用においてはIPv4を優先している実装であることを確認した．

4.2.6　プライバシ拡張アドレスの実装

SLAACによるアドレス設定では，プライバシ拡張アドレスがどのクライアントOSにおいても利用することが可能であり，Linuxを除いてデフォルトで利用されていた（LinuxにおいてUbuntu 16.04クライアントのみデフォルト有効となっていた）．そのため，インタフェースに複数のIPv6アドレスが設定されていたとしても，IPv6通信に利用されるアドレスは，ポリシーテーブルのルールに示されるようにプライバシ拡張アドレスのみが利用されていた．ただし，Apple社のOSでは，最新版（macOS 10.13, iOS 11）を除きローカルセグメント内での通信において固定のSLAACアドレスやDHCPv6アドレスが利用されていることが確認された．

表7に各クライアントOSにおけるプライバシ拡張アドレスの実装をまとめる．デフォルト利用の状況と，アドレスが再設定されるタイミング（再起動：reboot，RAによるプラフィックス情報の変化：pref chg.，インタフェースのDown/Up：if down）を調査した結果を記している．“○”となっているものが再生成が行われたことを表している．ここから分かるように，

Apple社のOSや最新のAndroidではインタフェースのDown/Upであってもプライバシ拡張アドレスを再生成しており，これは無線LANなどの不安定なインタフェースでは頻繁に新しいIPv6アドレスが設定されることを意味している．

表7 クライアントOSごとのプライバシ拡張アドレスの実装

OS	Default	Timing		
		reboot	pref chg.	if down
Windows 7	○	○	○	–
Windows 8.1	○	○	○	–
Windows 10	○	○	○	–
OS X 10.10	○	○	○	○
OS X 10.11	○	○	○	○
macOS 10.12	○	○	○	○
macOS 10.13	○	○	○	○
iOS 9	○	○	○	○
iOS 10	○	○	○	○
iOS 11	○	○	○	○
Android 4	○	○	–	○
Android 5	○	○	–	–
Android 6	○	○	○	○
Android 7	○	○	○	○
Android 8	○	○	○	○
ChromeOS	○	○	–	–
CentOS 7	×	N/A	N/A	N/A
Fedora 26	×	N/A	N/A	N/A
Ubuntu 16.04	○	○	–	–
Raspbian	×	N/A	N/A	N/A

5. ネットワーク運用に与える影響

第4章ではOSごとのアドレス自動設定実装の差異について検証実験により示した．本章において，IPv6導入時に検討が必要なネットワーク運用に与える影響について考察する．

5.1 アドレス自動設定と運用管理コスト

5.1.1 デュアルスタック運用における課題

IPv4とIPv6をそれぞれ提供する場合では，各端末ともにインターネットに接続できない状況は発生しない．しかしながら第3章でも述べた通り，IPv4とIPv6双方のネットワーク運用が必要となり，ネットワーク障害が発生した際の問題切り分けが複雑になることなどから，恒久的なネットワーク運用やセキュリティリスクへの対応によるコスト増は避けられない．デュアルスタックでの運用は，複数のプロトコルを利用することによる冗長性向上の反面，正しい通信状態管理を行っていないと中途半端に利用できてしまうため，障害発生の検知が遅れる場合があり，また，障害率が高まることにも繋がる．

5.1.2　IPv6オンリーネットワークでの課題

一方で，端末にIPv6アドレスのみを割り当て，IPv4接続をトランスレーションで提供する場合，OS実装の差異がネットワーク運用者側へ影響を与える可能性がある．本検証の結果からも分かるように，Windows8.1以前のWindows OSやAndroidも含めたすべてのOSを接続可能にするには，RDNSSオプション付きのRAとステートレスDHCPv6を併用する必要がある．次節で取り上げるIPアドレス管理を考えると，現行のIPv4におけるDHCPと同等の管理ができない点で運用手法の検討が求められる．

5.2　IPアドレス管理手法の整理と課題

昨今セキュリティインシデント対応の迅速化が求められる中，端末に割り当てるIPアドレス管理の必要性が高まっている．静的な割り当て管理は確実性を考えると高コストとなるため，IPv4ではDHCPを用いたステートフル管理が一般的に用いられている．DHCPでは，割り当てたIPアドレスと端末情報（IPv4の場合はMACアドレス）をリースファイルにて管理しているため，この情報を参照することで時系列にIPアドレス割り当てを管理できる．また，DHCP snooping[☆3]と組み合わせることで，管理の完全性を高めることも可能である．

5.2.1　DHCPとDHCPv6の差異

IPv6にて同様のステートフル管理を実現する場合，DHCPとDHCPv6の仕様の差異を認識し，運用に際して若干の工夫が必要になる．DHCPv6では，IPv4の場合のMACアドレスと異なり，クライアント端末の識別にDUID（DHCP Uniqe ID）を用いる．これはNICの変更によりIDが変わることを防ぐためであり，3種類のDUIDフォーマットが文献[3]に定義されている．さらに，UUIDを利用する形式[23] も2011年に追加されており，どの形式を利用するかはクライアントの実装に依存している．

このDUIDはMACアドレスよりも長く，一意性保証の面から優れてはいるが，利用者からの伝達方法やIPv4のDHCPと関連付けを考慮すると運用上問題がある．そこで，IPv6においてもMACアドレスを利用した管理を実現するためには，DHCPv6リレーエージェントに追加されたClient Link-Layer Addressオプション[24]を活用する必要がある．ただし，直接DHCPv6サーバを管理セグメントに設置すると，このオプションを利用することができないため注意が必要となる．

5.2.2 DHCPv6非対応端末への対策

DHCPv6を利用した管理においては，Android端末におけるDHCPv6非実装の問題も残っている．

Google社は，ステートフルDHCPv6が一般化することで，IPv6においてもインタフェースに設定されるグローバルIPアドレスが1つに制限され，IPv6の拡張性が阻害されると主張している[25]．1つに制限することで，テザリング環境などにおいてIPv4と同様なNAPT利用が助長され

ることも理由に挙げ，そのためAndroidへのDHCPv6実装を行わないとしている．これは，現時点においてIPアドレス管理をすべてのOSに対して実現する手段としてDHCPv6を選択できないことを意味している．

そこで，2017年時点ですべてのクライアント端末のIPv6アドレス管理を実現する手法としては，ルータやL3スイッチにて管理しているNDP近隣キャッシュ（以下，NDPキャッシュ）を定期的に収集する手法や，セグメント内に流れるNDPパケットを収集・監視することも有効であると考えられる．

5.3　ネットワーク機器への影響

本節では，ネットワーク機器仕様の面からIPv6ネットワークが与える影響について考察する．

5.3.1　複数アドレス利用によるリンクレイヤへの影響

各端末においては，なんらかの手法でIPアドレスを取得しDNSサーバの情報が得られれば通信が可能となるが，ルータおよびL3スイッチにおいては各端末のMACアドレスとIPアドレスの対応テーブルを持つ必要がある．IPv4の場合はARPテーブルとしてMACアドレスに対して基本的には1つのIPv4アドレスが対応しているのみであったが，IPv6アドレスについてはMACアドレスに対してリンクローカルアドレスと通信用のグローバルアドレス，さらにはプライバシ拡張アドレスが割り当てられる．それらについてもOSの実装によりタイマーあるいはインタフェースのUp/Down等でアドレスが変化することも確認できている．

その上，OSの実装によってはRAとDHCPv6の双方からアドレスを取得し保持することが今回の検証でも明らかになった．そのため，IPv6通信に利用されるアドレスはポリシーテーブルのルールにしたがうものの，プライバシ拡張によるアドレス変更やリンクローカルアドレスによる通信などにより，ARPより多くのエントリを占めることが想定される．

5.3.2　広島大学における実例

ここでは，組織内へのIPv6ネットワーク導入例として，広島大学のキャンパスで運用されている無線LANネットワークを例に示す．広島大学の無線LANネットワークはエリアに応じてVLANが別れており，すべてのVLANでIPv4/IPv6のデュアルスタック運用を行っている．本稿では，IPv4が/20，IPv6が/64のアドレス空間を持つVLANを対象に調査を行った．接続する端末は大学の構成員や学外者の持ち込みのパソコン，スマートフォンなどで，使用用途は大学生活で日常的に使うWeb検索やメール処理などが多い．**図3**にこのVLANにおける2017年12月13日のARPテーブルおよびNDPキャッシュを30分ごとに集計した結果を表している☆4．

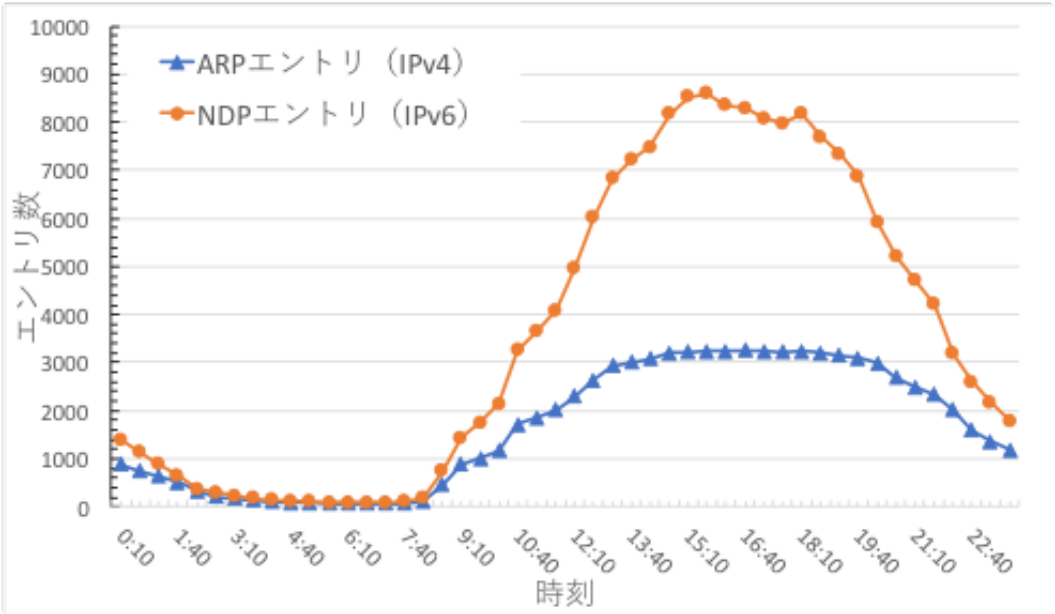

図3　ARPおよびNDPエントリ数の時間変化

ARPテーブルと比較するとNDPキャッシュのエントリ数が大幅に増えていることが見て取れる．ここから読み取れるのはピーク時においてNDPはARPの約3倍のエントリ数になることを想定する必要があるということである．また，評価日のピーク時刻（15:40）におけるMACアドレス（端末）ごとのエントリ数の分布を比較すると，IPv6では1つの端末に対して2つ以上のエントリを持つものが大勢を占め，19エントリとなる端末も存在していることが分かる（**図4**）．先に述べたように，プライバシ拡張アドレス利用とその再設定による影響の表れであると考えられる．なお，観測したルータにおけるARPテーブルおよびNDPキャッシュのエージングタイムは，双方とも4時間（機器におけるデフォルト値）の設定としている．

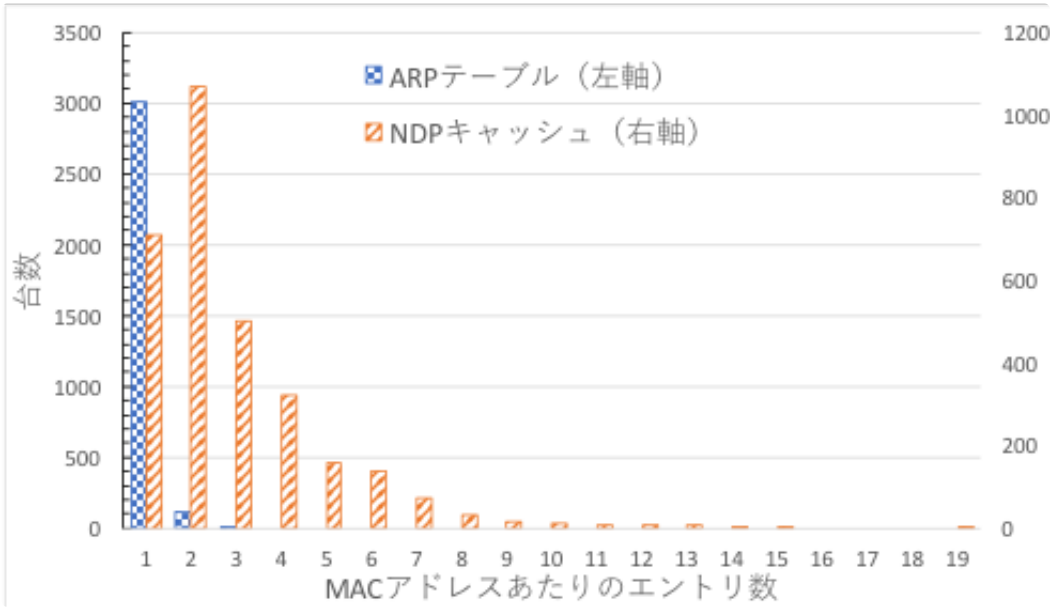

図4　MACアドレスあたりのARPおよびNDPエントリ数分布

5.3.3　ネットワーク機器の対応状況

NDPキャッシュを受ける側のL3スイッチの収容数は現在過渡期にあるといえる．アラクサラネットワークス社の1Uボックス型タイプのL3スイッチについて調査した結果，2010年に販売開始されたAX3650S[26]では，IPv4のみの運用であればARPは11,000強収容できるにもかかわらず，デュアルスタックにするとNDPは2,000しか入らず，ARPも2,000と減ってしまう．またIPv6のみの端末収容を優先する仕様も用意されていない．

一方2016年に販売開始されたAX3660S[27]では，IPv4のみであればARPは30,000強，デュアルスタックでARPが15,000とNDPが15,000，IPv6ユニキャスト優先とすればARPが7,680に対しNDPが23,000強と，かなりの改善が見られる．これはARPとNDPのエントリ数の割合についても前述の3倍という値に即しているといえる．

5.3.4　NDPテーブル肥大化に対する運用面での対策

以上のように，NDPエントリの肥大化に対してはネットワーク機器の性能を向上させることで対応できる可能性もある．一方で，本質的な課題解決となるエントリ数を削除する手法としては以下の方法が考えられる．

1つ目の手法としては，クライアント端末におけるプライバシ拡張アドレスの利用を制限することである．表5に示したようにプライバシに配慮したSemantically Opaque IIDが普及しつつあり，定期的に生成されるプライバシ拡張アドレスがなくともプライバシを確保できる環境になりつつある．ただし本手法においては，クライアント端末に対して設定を実施する必要があり，制御手法について別途検討する必要がある．

2つ目の手法として，文献[28]に示されているように，端末に/64のプレフィックスをすべて割り当てる運用方法を活用するものである．この手法を用いると，ルータは端末ごとに保持するNDPキャッシュのエントリをIPv6プレフィックスに集約することが可能で，NDPキャッシュ用のメモリを大きく保持する必要がなくなる可能性がある．ただし，このような集約が実装されている機器は存在しない．今後ネットワーク機器を導入するにあたって，有用性を評価するべきポイントであると考えている．

6. おわりに

本稿では，クライアントOSのIPv6実装における動作検証を，IPv6アドレス自動設定およびIPv6オンリーネットワーク環境下での挙動を中心に評価した結果を報告した．2017年時点のクライアントOS実装では，いずれもデュアルスタック環境での動作が可能であり，IPv6通信も問題なく実現できていることが確認できた．また，IPv6オンリーネットワーク環境では，一部のクライアントOSにおいて課題があることが分かり，クライアントOSごとの実装差異による問題も明らかとなった．今回検証対象としたすべてのクライアントOSをIPv6オンリーネットワーク環境で動作させるためには，RDNSSオプションとOフラグを設定したRAを利用し，ステートレスDHCPv6を用いる運用が最低限必要となることを確認した．

検証実験により，IPv6対応時にクライアントOSに設定されるIPv6アドレスはIPv4のみの時と大きく異なることが明らかになったことを受け，ネットワーク管理における課題を運用面と実装面から整理して考察した．運用面では，一様にステートフルでのアドレス管理ができない状況であることから，セキュリティインシデント対応時のトレーサビリティに関して課題があり，

IPv4とは異なる管理手法の導入が必要と考えられる．実装面では，デュアルスタック運用で機器のメモリ資源が切迫することが予想でき，機器導入時において考慮が必要であることを指摘した．

謝辞　本研究は，科学研究費補助金（15K00118）の助成を一部受けたものである．

参考文献

1）CISCO：6lab - The place to monitor IPv6 adoption（Statistics per country: Japan）, http://6lab.cisco.com/stats/search.php（参照 2017-12-27）
2）Thomson,S., Narten, T. and Jinmeim, T.：IPv6 Stateless Address Autoconfiguration, RFC 4862（2007）.
3) Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, and Carney, M.：Dynamic Host Configuration Protocol for IPv6（DHCPv6）, RFC 3315（2003）.
4）Narten, T., Nordmark, E., Simpson, W. and Soliman, H.：Neighbor Discovery for IP version 6（IPv6）, RFC 4861（2007）.
5）Hinden,R. and Deering, S.：IP Version 6 Addressing Architecture, RFC 4291（2006）.
6）Cooper, A., Gont, F. and Thaler, D.：Security and Privacy Considerations for IPv6 Address Generation Mechanisms, RFC 7721（2016）.
7）Narten, T., Draves, R. and Krishnan, S.：Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC 4941（2007）.
8）F. Gont：A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration（SLAAC）, RFC 7217（2014）.
9）Jeong, J., Park, S., Beloeil, L. and Madanapalli, S.：IPv6 Router Advertisement Options for DNS Configuration, RFC 8106（2017）[Obsoletes RFC 6106（2010）].
10) Thaler,D., Draves, R., Matsumoto, A. and Chown, T.：Default Address Selection for Internet Protocol Version 6（IPv6）, RFC 6724（2012）[Obsoletes RFC 3484（2003）].
11) Matsumoto,A., Fujisaki, T. and Chown, T.：Distributing Address Selection Policy Using DHCPv6, RFC 7078（2014）.
12) IPv6普及・高度化推進協議会　IPv4/IPv6共存WG IPv6導入に起因する問題検討SWG: IPv6導入時に注意すべき課題,
http://www.v6pc.jp/jp/pdf/20111124_v6fix.pdf（2001）,（参照 2017-12-27）
13）Bagnulo,M., Sullivan, A., Matthews, P. and van Beijnum, I.：DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers, RFC 6147（2011）.
14）Bagnulo, M., Matthews, P. and van Beijnum, v I.：Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers, RFC 6146（2011）.
15）Apple Inc.：Supporting Ipv6-only Networks,
https://developer.apple.com/support/ipv6/（参照 2017-12-27）
16）Savolainen, T. Korhonen, J. and Wing, D.：Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis, RFC 7050（2013）.
17）Liu, B., Jiang, S., Gong, X., Wang, W. and Ray, E.：DHCPv6/SLAAC Interaction Problems on Address and DNS Configuration,
https://tools.ietf.org/html/draft-ietf-v6ops-dhcpv6-slaac-problem-07（2016）.
18）Jackson, M.：UPDATE Microsoft Windows DHCP Bug Causing Internet Problems for Users,
https://www.ispreview.co.uk/index.php/2016/12/microsoft-windows-dhcp-bug-causing-internet-problems-users.html（2016）,（参照 2017-12-27）
19）Havey, D. M.：Core Network Stack Features in the Creators Update for

Windows 10,
https://blogs.technet.microsoft.com/networking/2017/07/13/core-network-stack-features-in-the-creators-update-for-windows-10/ （2017），（参照 2017-12-27）．
20）Apple Inc. : Supporing IPv6 DNS64/NAT64 Networks,
https://developer.apple.com/library/content/documentation/NetworkingInternetWeb/Conceptual/NetworkingOverview/UnderstandingandPreparingfortheIPv6Transition/UnderstandingandPreparingfortheIPv6Transition.html （参照 2017-12-27）．
21） Zorz. J.: Skype On Android Works Over IPv6 On Mobile Networks Using 464XLAT,
http://www.internetsociety.org/ deploy360/blog/2013/11/skype-on-android-works-over-ipv6-on-mobile-networks-using-464xlat/ （2013），（参照 2017-12-27）．
22）Mawatari, M., Kawashima, M. and Byrne, C. : 464XLAT : Combination of Stateful and Stateless Translation, RFC 6877 （2013）．
23） Narten, T. and Johnson, J. : Definition of the UUID-Based DHCPv6 Unique Identifier （DUID-UUID）, RFC 6355 （2011）．
24） Halwasia, G., Bhandari, S. and Dec, W. : Client Link-Layer Address Option in DHCPv6, RFC 6939 （2013）．
25） Colitti, L., Cerf, V., Cheshire, S. and Schinazi, D. : Host Address Availability Recommendations, RFC 7934 （2016）．
26）ALAXALA Networks Corp. : AX3650Sのテーブルエントリ数.
https://www.alaxala.com/jp/techinfo/archive/manual/AX3650S/HTML/11_14/CFGUIDE/0021.HTM#ID00076 （2015），（参照 2017-12-27）．
27）ALAXALA Networks Corp. : AX3660Sのテーブルエントリ数.
https://www.alaxala.com/jp/techinfo/archive/manual/AX3660S/HTML/12_1_B/CFGUIDE/0018.HTM （2017），（参照 2017-12-27）．
28）Brzozowski,J. and Van de Velde, G. : Unique IPv6 Prefix per Host, RFC 8273 （2017）．

脚注

☆1　RDNSS（Recursive DNSサーバ）：再帰的にDNS問い合わせを代理実行するDNSサーバ．フルサービスリゾルバ．キャッシュDNSサーバ．
☆2　Experimental Stateでは2007年にRFC化されている．
☆3　DHCP通信を監視してDHCPにてアドレス設定されていない管理対象外端末を接続させない機能．
☆4　なおネットワーク利用状況の目安として，調査当日のDHCPによるアドレス割当数のピークが2,078（16:20時点）であった．

北口 善明（正会員）kitaguchi@gsic.titech.ac.jp
1997年新潟大学自然科学研究科修士課程修了．同年（株）インテックに入社．2004年電気通信大学大学院情報システム学研究科博士課程満期退学．2005年同大学で博士号を取得．金沢大学総合メディア基盤センター助教を経て現在，東京工業大学学術国際情報センター准教授．博士（工学）．ネットワークの運用管理およびIPv6の研究に従事．電子情報通信学会，IEEE各会員．

近堂 徹（正会員）tkondo@hiroshima-u.ac.jp

2001年広島大学工学部第二類（電気系）卒業．2003年同大大学院工学研究科博士課程前期修了．2006年同大大学院工学研究科博士課程後期修了．現在，広島大学情報メディア教育研究センター准教授．博士（工学）．ネットワーク管理・運用，リアルタイムマルチメディア通信，仮想化技術の研究に従事．電子情報通信学会，IEEE各会員．

鈴田 伊知郎（正会員）ichiroh@suzuta.jp

1993年東京都立科学技術大学工学部管理工学科卒業．1995年同大大学院工学研究科修士課程修了．1998年同大大学院工学研究科博士課程満期退学．湘北短期大学電子情報学科助手，豊橋創造大学経営情報学部助教授を経て現在，アラクサラネットワークス（株）ネットワークシステム部所属．

小林 貴之（非会員）tkoba@chs.nihon-u.ac.jp

1992年東京都立大学大学院理学研究科化学専攻博士課程単位修得満期退学後，北里大学助手を経て現在，日本大学文理学部情報科学研究所准教授．専門は核地球宇宙科学と教育情報基盤構築．最近は学内ネットワークやe-Learningなどの構築および学内外でのICT活用に従事．

前野 譲二（正会員）joji@decode.waseda.ac.jp

1995年早稲田大学商学部卒業．1997年同大学大学院商学研究科修士課程修了，2000年同博士課程単位取得退学．1997年から同大学メディアネットワークセンター助手，同客員講師を経て現在，教育学部非常勤講師，情報教育研究所招聘研究所員．教育の情報化，情報倫理の研究，情報基盤構築などに従事．

投稿受付：2018年1月15日
採録決定：2018年6月15日
編集担当：寺田真敏（日立製作所）