東京工業大学
Tokyo Institute of Technology

IPv6クライアントOSの実装検証報告

September 13, 2019

北口 善明（東京工業大学）

本日の内容

- IPv6における自動アドレス設定
- IPv6クライアントOSの実装検証
- ネットワーク運用における課題

東京工業大学
Tokyo Institute of Technology

IPv6における自動アドレス設定

IPv6対応とは

- IPv4とIPv6のネットワークを二重運用
  - 単純に運用コストが**二倍**
    - トラブル発生時には切り分けが必要でコスト増の可能性あり
    - 監視系統も双方のプロトコルで実施する必要あり
  - プロトコル毎にサーバを分けると整合性確認が必要に
    - DNSレコードやセキュリティポリシの統一が重要
- IPv6オンリーネットワーク
  - IPv4をサービスの一つとして運用
    - NAT64+DNS64
  - ネットワーク部分はIPv6だけにしてシンプルに
    - 米国モバイルキャリアでこの形態が利用されている
  - イベントネットワークでも提供例あり

自動アドレス設定手法の違い

- 二種類のIPv6アドレス設定手法
  - SLAAC：ステートレスなIPv6アドレス設定　※SLAAC（StateLess Address Auto Configuration）
  - DHCPv6：ステートフルなIPv6アドレス設定
- 自動アドレス設定で設定される項目と手法

	SLAAC	DHCPv6	（参考）DHCP
デフォルト経路	○	× (1)	○
アドレス	○ (2)	○	○
プレフィックス長	○	× (1)	○
サーバ情報（RDNSSなど）	○ (3)	○	○
ルータ優先度（RFC 4191）	○	× (1)	ー

(1) IETFにて過去に議論があったが標準化の見通しなし（draft-ietf-mif-dhcpv6-route-option (expired)）
(2) プレフィックス情報からアドレスを生成　(3) RDNSSオプション（RFC 6106 -> 8106）

- RAのRDNSS（Recursive DNS Server）オプションの必須化（**RFC 8106**）

SLAACとDHCPv6の関係

- ルータ広告中のフラグにより挙動制御（**RFC 4861**）

	A flag	O flag	M flag	備考
SLAAC	1	0	0	RDNSSオプションでDNSサーバ（**Windows 10 Creators Updateで対応**）
SLAAC+ステートレスDHCPv6	1	1	0	ほとんどのOSで利用可能（**Androidは非対応**）
ステートフルDHCPv6	0	N/A	1	割当アドレス管理を実施する形態
SLAAC+ステートフルDHCPv6	1	N/A	1	SLAACによるアドレスとDHCPv6による双方のアドレスが付く（冗長？）

  - A (autonomous address-configuration) flag：
    - プレフィックス情報オプションのフラグ
    - =1 でプレフィックス情報を利用したSLAACによるアドレス設定を促す
  - O (other configuration) flag：
    - アドレス以外の設定をDHCPv6で実施するためのフラグ
    - =1 でステートレスDHCPv6処理を促す
  - M (managed address configuration) flag：
    - SLAAC以外でのアドレス設定をDHCPv6で実施するためのフラグ
    - =1 でステートフルDHCPv6処理を促す（O flagの値は無視される）

デュアルスタックサービスの提供形態

- 自動アドレス設定による提供形態

Type	IPv4	IPv6			備考
		RA flags	Address	DNS	
0	DHCP	A	RA	IPv4 only	IPv4依存型（導入初期の主流形態）
1		A, RDNSS	RA	RA	RAのみによる独立型
2		A, O	RA	DHCPv6	アドレス非管理の独立型
3		M	DHCPv6	DHCPv6	アドレス管理型
4	DNS64/ NAT64	A, RDNSS	RA	RA	RAのみによる独立型（IPv6 only）
5		A, O	RA	DHCPv6	アドレス非管理の独立型（IPv6 only）
6		M	DHCPv6	DHCPv6	アドレス管理型（IPv6 only）

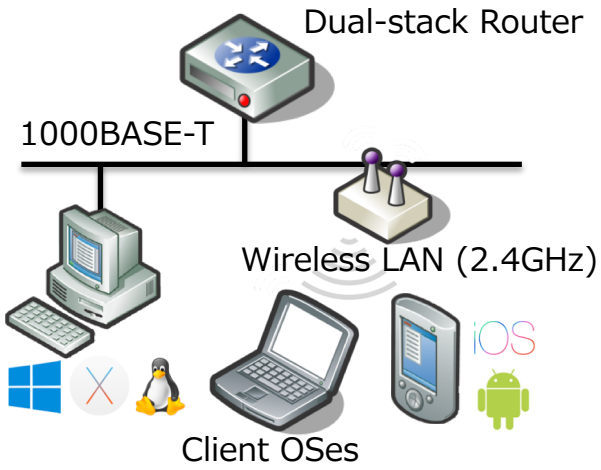
- DNSサーバの配布手法で実装の差異がある
  - Windows系：RDNSS非対応（2017年以前）、Android系：DHCPv6非対応
  - 双方に対応するためには複合形態を取る必要あり

東京工業大学
Tokyo Institute of Technology

IPv6クライアントOSの実装検証

クライアントOSのIPv6実装検証

- デュアルスタックとIPv6オンリー環境での動作
  - SLAACとDHCPv6の実装確認
  - 複数のIPv6アドレスやDNSサーバがある場合の動作確認
    - 区別のためRDNSSとDHCPv6で別のIPv6アドレスを配布
- 検証時の手順
  - 無線LANインタフェースの停止
  - DNSキャッシュデータの削除
  - パケットキャプチャ開始
  - 無線LANインタフェースの起動
  - IPv4およびIPv6ウェブサーバへの接続確認
- 調査結果詳細



検証実験のネットワーク構成

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
情報処理学会デジタルプラクティス, Vol.9, No.4, pp.902-922, October 2018.
https://www.ipsj.or.jp/dp/contents/publication/36/S0904-R1701.html

検証に用いたクライアントOS一覧

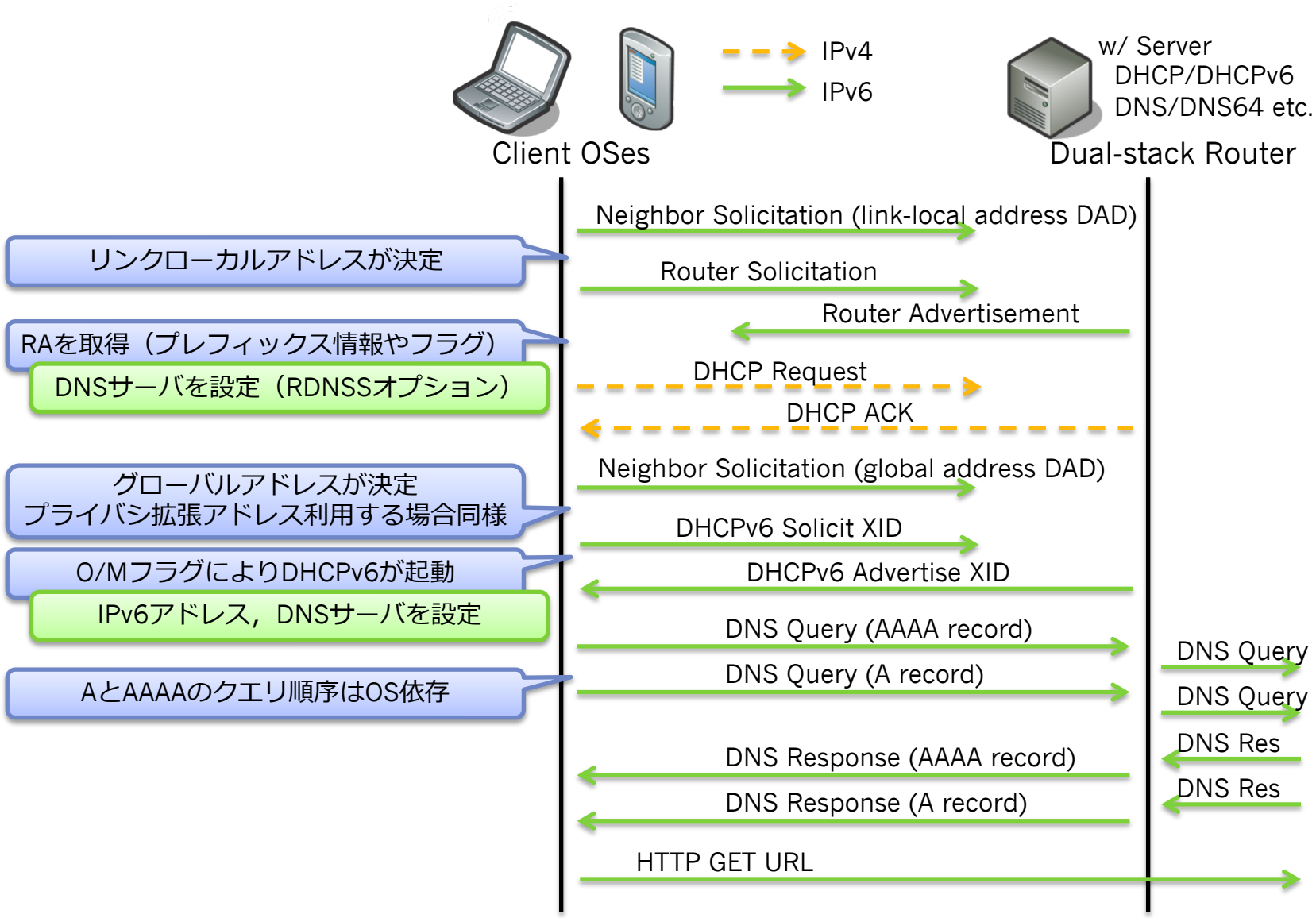
東京工業大学 Tokyo Institute of Technology

OS	Version	Release	Hardware
Windows 7	6.1 (7601, SP1)	2011/02/22	Virtual Machine
Windows 8.1	6.3 (9600)	2013/10/18	Think Pad Tablet2
Windows 10	1703 (15063.726)	2017/04/11	Surface Pro 4
Windows 10'	1903 (18362.295)	2019/05/23	Surface Pro 4
OS X 10.10	10.10.5	2015/08/13	Virtual Machine
OS X 10.11	10.11.6	2016/07/18	Virtual Machine
macOS 10.12	10.12.2	2017/07/19	MacBook Pro
macOS 10.13	10.13.1	2017/10/31	MacBook Pro
macOS 10.14	10.14.6	2019/07/22	MacBook Pro
iOS 9	9.3.5	2016/08/26	iPad2
iOS 10	10.2	2017/07/19	iPad Air
iOS 11	11.1.2	2017/11/18	iPhone 7
iOS 12	12.4	2019/07/23	iPad Air

OS	Version	Release	Hardware
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
Android 9	9.0	2019/08/01	Pixel 3a
Chrome OS	60.0.3112.114	2017/10/31	MacBook Pro
CentOS 7	3.10.0-693.2.2.el7	2017/10/01	Virtual Machine
Fedora 26	4.12.14-300.fc26	2017/09/07	Virtual Machine
Ubuntu 16.04	4.4.0-96 (16.04.3)	2017/10/01	Virtual Machine
Ubuntu 18.04	5.0.0-25 (18.04.2)	2019/08/26	Virtual Machine
Raspbian 9.1	4.9.41-v7 (Sep.2017)	2017/10/01	Raspberry Pi 3
Raspbian 10	4.19.66-v7 (Jul.2019)	2019/08/26	Raspberry Pi 3+

今回追加検証したクライアントOS

典型的なクライアントOS起動時のフロー



クライアントOS毎の挙動調査結果（1）

東京工業大学 Tokyo Institute of Technology

- インタフェースIDとRDNSサーバの配布、IPv6オンリー環境耐性

OS	SLAAC	RA DNS	DHCPv6	v6only
Windows 7	Microsoft	×	○	○
Windows 8.1	Microsoft	×	○	○
Windows 10	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	○	○	○
macOS 10.14	RFC 7217	○	○	○
iOS 9	RFC 7217	○	○	○
iOS 10	RFC 7217	○	○	○
iOS 11	RFC 7217	○	○	○
iOS 12	RFC 7217	○	○	○

OS	SLAAC	RA DNS	DHCPv6	v6only
Android 4	RFC 4291	×	×	×
Android 5	RFC 4291	○	×	○
Android 6	RFC 4291	○	×	○
Android 7	RFC 4291	○	×	○
Android 8	RFC 4291	○	×	○
Android 9	RFC 4291	○	×	○
Chrome OS	RFC 4291	○	×	○
CentOS 7	RFC 7217	○	○	○
Fedora 26	RFC 7217	○	○	○
Ubuntu 16.04	RFC 7217	○	○	○
Ubuntu 18.04	RFC 7217	○	○	○
Raspbian 9.1	RFC 7217	○	○	○
Raspbian 10	RFC 7217	○	○	○

- インタフェースID生成は多くのOSでRFC 7217に移行しつつある
- AndroidでのDHCPv6実装がない状況は変わらない
- Windows 10ではDHCPv6クライアントが常に起動

インタフェースIDの議論

- IPv6アドレス構造

- RFC 4291での仕様はModified EUI-64 format
  - MACアドレスを元に生成し常に一意
    - プレフィックスが変化しても一意に特定可能（プライバシ）
    - MACアドレスによる特定機器を狙った攻撃（セキュリティ）
- プライバシ拡張アドレス（RFC 4941）
  - ランダム生成で定期的に変更する仕様
    - 同じネットワークにて定期的に変化するため管理が難しい
    - 追加仕様なのでModified EUI-64 formatのアドレスと併用
- プライバシを保護した固定ID（RFC 7217）
  - プレフィックス情報が変化した場合に再生成　Semantically Opaque Interface Identifiers
    - 同一ネットワーク内では変化しないため管理可能
    - MACアドレスを推測できないので特定機器攻撃を回避

自動アドレス設定の実装に関する議論

- AndroidにおけるDHCPv6非実装の問題
  - IPv6におけるステートフルアドレス設定が統一的にできない
  - IPv4と同様の管理手法が利用できない点に注意が必要
- Googleの主張（**RFC 7934**）
  - DHCPv6利用はインタフェースに1つのIPv6アドレスと限定することに
    - 1インタフェースに複数のアドレスを持つIPv6の拡張性を害する
    - 1つにするとNAPT利用を助長する

    以上の理由からAndroidでDHCPv6を実装しない
  - 複数アドレスのメリット
    - プライバシ拡張アドレスでトレース回避
    - アプリケーション毎にアドレスを使い分けることが可能
    - テザリングや仮想マシンに対して独立したアドレスを提供可能

クライアントOS毎の挙動調査結果（2）

東京工業大学 Tokyo Institute of Technology

- 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	○	○	○	–
Windows 10'	○	○	–	○
OS X 10.10	○	○	○	○
OS X 10.11	○	○	○	○
macOS 10.12	○	○	○	○
macOS 10.13	○	○	○	○
macOS 10.14	○	○	○	○
iOS 9	○	○	○	○
iOS 10	○	○	○	○
iOS 11	○	○	○	○
iOS 12	○	○	○	○

OS	SLAAC	Timing		
		Reboot	Pref. Chg.	IF Down
Android 4	○	○	–	○
Android 5	○	○	–	–
Android 6	○	○	○	○
Android 7	○	○	○	○
Android 8	○	○	○	○
Android 9	○	○	–	–
Chrome OS	○	○	–	–
CentOS 7	×	N/A	N/A	N/A
Fedora 26	×	N/A	N/A	N/A
Ubuntu 16.04	○	○	–	–
Ubuntu 18.04	○	○	–	–
Raspbian 9.1	×	N/A	N/A	N/A
Raspbian 10	×	N/A	N/A	N/A

- Linux系はsysctlにてON/OFFが可能
- IF downで再生成する実装が多い ⇒ 無線LANなど不安定な環境で課題

端末へのプレフィックス割り当て

- Unique IPv6 Prefix per Host（**RFC 8273**）
  - クライアント毎に異なるプレフィックス情報をユニキャストRAで配布
    - L flagを0にセットすることでシェアードメディアでの通信を制御
  - L (on-link) flag
    - プレフィックス情報オプション中のフラグ
    - =1で当該プレフィックス情報をオンリンク決定に利用可能
      =0で自身以外のアドレスがルータの先にあると判断（NDPでの解決をしない）
- Lフラグoff時の挙動差異（追加検証OSのみ）
  - macOS 10.14, iOS 12
    - 同一プレフィックス内通信のNDPリンクレイヤアドレス解決を実施しない
    - 全てルータ経由の通信になるがルータからのリダイレクトを受ければ直接通信可能
  - Windows 10, Ubuntu 18.04, Raspbian 10
    - NDPリンクレイヤアドレス解決を実施し直接通信

クライアントOS毎の挙動調査結果（3）

東京工業大学 Tokyo Institute of Technology

- RDNSサーバおよびDNSクエリ順序 ※(A) の表記はIPv6オンリー環境で問い合わせしないことを示す

OS	DNS Order		DNS query Order
	Dual-stack	IPv6 Only	
Windows 7	DHCPv6	DHCPv6	(A), AAAA
Windows 8.1	DHCPv6	DHCPv6	(A), AAAA
Windows 10	DHCPv6	DHCPv6	(A), AAAA
Windows 10'	Random	Random	(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)
macOS 10.14	Random	Random	AAAA, (A)
iOS 9	DHCPv6	DHCPv6	AAAA, (A)
iOS 10	DHCPv6	DHCPv6	AAAA, (A)
iOS 11	DHCPv6	DHCPv6	AAAA, (A)
iOS 12	Random	DHCPv6	AAAA, (A)

OS	DNS Order		DNS query Order
	Dual-stack	IPv6 Only	
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
Android 9	RA	RA	A, AAAA
Chrome OS	IPv4	RA	A, AAAA
CentOS 7	IPv4	RA	A, AAAA
Fedora 26	IPv4	RA	A, AAAA
Ubuntu 16.04	IPv4	RA	A, AAAA
Ubuntu 18.04	Random	Random	A, AAAA
Raspbian 9.1	IPv4	RA	A, AAAA
Raspbian 10	IPv4	DHCPv6	A, AAAA

- RDNSサーバの利用順序はOSにより異なる
- DNSクエリはほぼ同じタイミングで実施
  - IPv6オンリーではAの問い合わせを実施しない実装も

※3分程度のパケットダンプ評価のため
精度は高くない

ネットワーク運用における課題

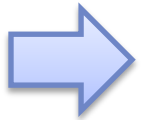
ネットワーク運用における課題（1）

- ネットワーク運用管理コスト
  - 割り当てアドレス管理の必要性
    - セキュリティインシデント発生時における利用者の特定
    - 未登録ユーザの利用禁止と利用中アドレスの把握
  - DHCPv6利用の差異
    - Android系が非対応、Windows系はクライアントが常に起動
    - DUIDで固定アドレス設定（IPv4ではMACアドレス）
- DHCPv6以外の管理手法
  - MACアドレス認証＋NDP近隣キャッシュの定期収集 or NDPモニタリング
    - MACアドレス認証のスイッチが必要
    - NDPモニタリングはtcpdumpやndpmon

ネットワーク運用における課題（2）

東京工業大学 Tokyo Institute of Technology

- プライバシ拡張アドレス
  - 定期的に更新されるためアドレスがどんどん増える
    - ルータ機器におけるNDPキャッシュが肥大化する課題
      - リンクローカルアドレス、グローバルアドレス（固定）に加えグローバルアドレス（一時アドレス）が定期的に増加
      - 機器選定時に注意が必要
    - 一週間の運用で
      - macOS 10.12（4アドレス）、iOS 10（10アドレス）
  - アドレス選択ルールにて優先利用される
    - 端末のトレースが難しい（NDPモニタの対応が必要）



RFC 7217利用が進むと不要になりそうだがRFC 4941bisの議論も進行中

- DNSの挙動など相違点の把握が必要
  - トラブル時の切り分けが困難になる可能性
  - Happy Eyeballsまで含めた検証が必要

まとめ

- クライアント端末のIPv6実装は枯れていない
  - 未だに新しい仕様が登場 or 仕様変更の議論
  - 複雑な自動アドレス設定の仕様
  - デュアルスタックでの挙動の差異
- ネットワーク運用における課題
  - IPv6における利用アドレス管理に問題
  - ネットワーク機器の資源確保・確認が重要
  - IPv6オンリーネットワーク運用の課題整理が必要
- これからも仕様変更・追加の議論に注意
  - アドレスアーキテクチャの標準化議論中
  - IPv6ネットワーク運用におけるセキュリティ対策（**I-D ietf-opsec-v6**）