V6OPS
Internet-Draft
Intended status: Informational
Expires: February 18, 2017

B. Liu
S. Jiang
Huawei Technologies
X. Gong
W. Wang
BUPT University
E. Rey
ERNW GmbH

August 17, 2016

DHCPv6/SLAAC Interaction Problems on Address and DNS Configuration
draft-ietf-v6ops-dhcpv6-slaac-problem-07

Abstract

The IPv6 Neighbor Discovery (ND) Protocol includes an ICMPv6 Router Advertisement (RA) message. The RA message contains three flags, indicating the availability of address auto-configuration mechanisms and other configuration such as DNS-related configuration. These are the M, O, and A flags, which by definition are advisory, not prescriptive.

This document describes divergent host behaviors observed in popular operating systems. It also discusses operational problems that the divergent behaviors might cause.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IPv6 [RFC2460] hosts could invoke Neighbor Discovery (ND) [RFC4861] to to discover which auto-configuration mechanisms are available to them. There are two auto-configuration mechanisms in IPv6:

- DHCPv6 [RFC3315]

- Stateless Address Autoconfiguration (SLAAC) [RFC4862]

ND specifies an ICMPv6-based [RFC4443] Router Advertisement (RA) message. Routers periodically multicast the RA messages to all on-link nodes. They also unicast RA messages in response to solicitations. The RA message contains (but not limited to):

- an M (Managed) flag, indicating that addresses are available from DHCPv6 or not

- an O (OtherConfig) flag, indicating that other configuration information (e.g., DNS-related information) is available from DHCPv6 or not

- zero or more Prefix Information (PI) Options

  an A (Autonomous) flag is included, indicating that the prefix can be used for SLAAC or not

The M and O flags are advisory, not prescriptive. For example, the M flag indicates that addresses are available from DHCPv6, but It does not indicate that hosts are required to acquire addresses from DHCPv6. Similar statements can be made about the O flag. (A flag is also advisory by definition in standard, but it is quite prescriptive in implementations according to the test results in the appendix.)

Because of the advisory definition of the flags, in some cases different operating systems appear divergent behaviors. This document analyzes possible divergent host behaviors might happen (most of the possible divergent behaviors are already observed in popular operating systems) and the operational problems might caused by divergent behaviors.

2. The M, O and A Flags

This section briefly reviews how the M, O and A flags are defined in ND[RFC4861] and SLAAC[RFC4862].

2.1. Flags Definition

- M (Managed) Flag

  As decribed in [RFC4861], "When set, it indicates that addresses are available via Dynamic Host Configuration Protocol".

- O (Otherconfig) Flag

  "When set, it indicates that other configuration information is available via DHCPv6. Examples of such information are DNS-related information or information on other servers within the network." [RFC4861]

  "If neither M nor O flags are set, this indicates that no information is available via DHCPv6" . [RFC4861]

- A (Autonomous) Flag

  A flag is defined in the PIO, "When set indicates that this prefix can be used for stateless address configuration as specified in [RFC4862].".

2.2. Flags Relationship

Per [RFC4861], "If the M flag is set, the O flag is redundant and can be ignored because DHCPv6 will return all available configuration information.".

There is no explicit description of the relationship between A flag and the M/O flags.

3. Behavior Ambiguity Analysis

The ambiguity of the flags definition means that when interpreting the same messages, different hosts might behave differently. The ambiguity space is analyzed as the following aspects.

1) Dependency between DHCPv6 and RA

   In standards, behavior of DHCPv6 and Neighbor Discovery protocols is specified respectively. But it is not clear that whether there should be any dependency between them. More specifically, it is unclear whether RA (with M=1) is required to trigger DHCPv6; in other words, It is unclear whether hosts should initiate DHCPv6 by themselves if there are no RAs at all.

2) Overlapping configuration between DHCPv6 and RA

When address and DNS configuration are both available from DHCPv6 and RA, it is not clear how to deal with the overlapping information. Should the hosts accept all the information? If the information conflicts, which one should take higher priority?

For DNS configuration, [RFC6106] clearly specifies "In the case where the DNS options of RDNSS and DNSSL can be obtained from multiple sources, such as RA and DHCP, the IPv6 host SHOULD keep some DNS options from all sources" and "the DNS information from DHCP takes precedence over that from RA for DNS queries" (Section 5.3.1 of [RFC6106]). But for address configuration, there's no such guidance.

3) Interpretation on Flags Transition

- Impact on SLAAC/DHCPv6 on and off

  When flags are in transition, e.g. the host is already SLAAC-configured, then M flag changes from FALSE to TRUE, it is not clear whether the host should start DHCPv6 or not; or vise versa, the host is already configured by both SLAAC and DHCPv6, then M flag change from TRUE to FALSE, it is also not clear whether the host should turn DHCPv6 off or not.

- Impact on address lifetime

  When one address configuration method is off, that is, the A flag or M flag changes from TRUE to FALSE, it is not clear whether one host should immediately release the corresponding address or just retain it until the lifetime expires.

4) Relationship between the Flags

As described above, the relationship between A flag and M/O flags is unspecified.

It could be reasonably deduced that M flag should be independent from A flag. In other words, the M flag only cares DHCPv6 address configuration, while the A flag only cares SLAAC.

But for A flag and O flag, ambiguity could possibly happen. For example, when A is FALSE (when M is also FALSE) and O is TRUE, it is not clear whether the host should initiate a stand-alone stateless DHCPv6 session.

Divergent behaviors on all these aspects have been observed among some popular operating systems as described in Section 4 below.

4. Observed Divergent Host Behaviors

The authors tested several popular operating systems in order to determine what behaviors the M, O and A flag elicit. In some cases, the M, O and A flags elicit divergent behaviors. The table below characterizes those cases. For test details, please refer to Appendix A.

Operation diverges in two ways: one is regarding to address auto-configuration; the other is regarding to DNS configuration.

4.1. Divergent Behavior on Address Auto-Configuration

Divergence 1-1

- Host state: has not acquired any addresses.

- Input: no RA.

- Divergent Behavior

  1) Acquiring addresses from DHCPv6.

  2) No DHCPv6 action.

Divergence 1-2

- Host state: has acquired addresses from DHCPv6 only (M = 1).

- Input: RA with M =0.

- Divergent Behavior

  1) Releasing DHCPv6 addresses immediately.

  2) Releasing DHCPv6 addresses when they expire.

Divergence 1-3

- Host state: has acquired addresses from SLAAC only (A=1).

- Input: RA with M =1.

- Divergent Behavior

1) Acquiring DHCPv6 addresses immediately.

2) Acquiring DHCPv6 addresses only if their SLAAC addresses expire and cannot be refreshed.

4.2. Divergent Behavior on DNS Configuration

Divergence 2-1

- Host state: has not acquired any addresses or information.

- Input: RA with M=0, O=1, no RDNSS; and a DHCPv6 server on the same link providing RDNSS (regardless of address provisioning).

- Divergent Behavior

  1) Acquiring RDNNS from DHCPv6, regardless of the A flag setting.

  2) Acquiring RDNNS from DHCPv6 only if A=1.

Divergence 2-2

(This divergence is only for those operations systems which support[RFC6106].)

- Host state: has not acquired any addresses or information.

- Input: RA with M=0/1, A=1, O=1 and an RDNSS is advertised; and a DHCPv6 server on the same link providing IPv6 addresses and RDNSS.

- Divergent Behavior

  1) Getting RDNSS from both the RAs and the DHCPv6 server, and the RDNSS obtained from the router has a higher priority.

  2) Getting RDNSS from both the RAs and the DHCPv6 server, but the RDNSS obtained from the DHCPv6 server has a higher priority.

  3) Getting RDNSS from the router, and a "domain search list" information only from the DHCPv6 server(no RDNSS).

Divergence 2-3

(This divergence is only for those operations systems which support[RFC6106].)

- Host state: has acquired address and RDNSS from the first router's RAs (M=0, O=0, PIO with A=1, and RDNSS advertised).

- Input: another router advertising M=1, O=1, no prefix information; and a DHCPv6 server on the same link providing IPv6 addresses and RDNSS.

- Divergent Behavior

  1) Never getting any information (neither IPv6 address nor RDNSS) from the DHCPv6 server.

  2) Getting an IPv6 address and RDNSS from the DHCPv6 server while retaining the address and RDNSS obtained from the RAs of the first router.

     (More details: the RDNSS obtained from the first router has a higher priority; when they receive again RAs from the first router, they lose/forget the information (IPv6 address and RDNSS) obtained from the DHCPv6 server.)

Divergence 2-4

(This divergence is only for those operations systems which support[RFC6106].)

- Host state: has acquired address and RDNSS from the DHCPv6 server indicated by the first router (M=1, O=1, no PIO or RDNSS advertised).

- Input: another router advertising M=0, O=0, PIO with A=1, and RNDSS.

- Divergent Behavior

  1) Getting address and RDNSS from the second router's RAs, and releasing the IPv6 address and the RDNSS obtained from the DHCPv6 server.

     (More details: when receiving RAs from the first router again, it performs the DHCPv6 Confirm/Reply procedure and gets an IPv6 address and RDNSS from the DHCPv6 server while retaining the ones obtained from the RAs of the second router. Moreover, the RDNSS from router 1 has higher priority than the one from DHCPv6.)

  2) Getting address and RDNSS from the second router's RAs, and retaining the IPv6 address and the "Domain Search list"

obtained from the DHCPv6 server. (It did not get the RDNSS from the DHCPv6 server, as described in Divergence 2-2.)

(More details: when receiving RAs from the first router again, there is no change; all the obtained information is retained.)

3) Getting address but no RDNSS from the second router's RAs, and also retaining the IPv6 address and the RDNSS obtained from the DHCPv6 server.

(More details: when receiving RAs from the first router again, there is no change; all the obtained information is retained.)

5. Operational Problems

This section is not a full collection of the potential problems. It is some operational issues that the authors could see at current stage.

5.1. Standalone Stateless DHCPv6 Configuration not available

It is impossible for some hosts to acquire stateless DHCPv6 configuration unless addresses are acquired from either DHCPv6 or SLAAC (Which requires M flag or A flag is TURE).

5.2. Renumbering Issues

According to [RFC6879] a renumbering exercise can include the following steps:

- Causing a host to

  release the SLAAC address and acquire a new address from DHCPv6; or vice-versa.

  release the current SLAAC address and acquire another new SLAAC address (might comes from different source).

  retain current SLAAC or DHCPv6 address and acquire another new address from DHCPv6 or SLAAC.

Ideally, these steps could be initiated by multicasting RA messages onto the link that is being renumbered. Sadly, this is not possible, because the RA messages may elicit a different behavior from each host.

6. Security Considerations

An attacker, without having to install a rogue router, can install a rogue DHCPv6 server and provide IPv6 addresses to Windows 8.1 systems. This can allow her to interact with these systems in a different scope, which, for instance, is not monitored by an IDPS system.

If an attacker wants to perform MiTM (Man in The Middle) using a rogue DNS while legitimates RAs with the O flag set are sent to enforce the use of a DHCPv6 server, the attacker can spoof RAs with the same settings with the legitimate prefix (in order to remain undetectable) but advertising the attacker's DNS using RDNSS. In this case, Fedora 21, Centos 7 and Ubuntu 14.04 will use the rogue RDNSS (advertised by the RAs) as a first option.

Fedora 21 and Centos 7 behaviour cannot be explored for a MiTM attack using a rogue DNS information either, since the one obtained by the RAs of the first router has a higher priority.

The behaviour of Fedora 21, Centos 7 and Windows 7 can be exploited for DoS purposes. A rogue IPv6 router not only provides its own information to the clients, but it also removes the previous obtained (legitimate) information. The Fedora and Centos behaviour can also be exploited for MiTM purposes by advertising rogue RDNSS by RAs which include RDNSS information.

(Note: the security considerations for specific operating systems are based on the detailed test results as described in Appendix A.)

7. IANA Considerations

This draft does not request any IANA action.

8. Acknowledgements

The authors wish to acknowledge BNRC-BUPT (Broad Network Research Centre in Beijing University of Posts and Telecommunications) for their testing efforts. Special thanks to Xudong Shi, Longyun Yuan and Xiaojian Xue for their extraordinary effort.

Special thanks to Ron Bonica who made a lot of significant contribution to this draft, including draft editing and presentations which dramatically improved this work.

The authors also wish to acknowledge Brian E Carpenter, Ran Atkinson, Mikael Abrahamsson, Tatuya Jinmei, Mark Andrews and Mark Smith for their helpful comments.

9. References

9.1. Normative References

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <http://www.rfc-editor.org/info/rfc4443>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, DOI 10.17487/RFC6106, November 2010, <http://www.rfc-editor.org/info/rfc6106>.

9.2. Informative References

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, DOI 10.17487/RFC3736, April 2004, <http://www.rfc-editor.org/info/rfc3736>.

[RFC6879] Jiang, S., Liu, B., and B. Carpenter, "IPv6 Enterprise Network Renumbering Scenarios, Considerations, and Methods", RFC 6879, DOI 10.17487/RFC6879, February 2013, <http://www.rfc-editor.org/info/rfc6879>.

Appendix A. Test Results

The authors from two orgnizations tested different scenarios independent of each other. The following text decribes the two test sets respectively.

A.1. Test Set 1

A.1.1. Test Environment

The test environment was replicated on a single server using VMware. For simplicity of operation, only one host was run at a time. Network elements were as follows:

- Router: Quagga 0.99-19 soft router installed on Ubuntu 11.04 virtual host

- DHCPv6 Server: Dibbler-server installed on Ubuntu 11.04 virtual host

- Host 1: Window 7 / Window 8.1 Virtual Host

- Host 2: Ubuntu 14.04 (Linux Kernel 3.12.0) Virtual Host

- Host 3: Mac OS X v10.9 Virtual Host

- Host 4: IOS 8.0 (model: Apple iPhone 5S, connected via wifi)

A.1.2. Address Auto-configuration Behavior in the Initial State

The bullet list below describes host behavior in the initial state, when the host has not yet acquired any auto-configuration information. Each bullet item represents an input and the behavior elicited by that input.

- A=0, M=0, O=0

  * Windows 8.1 acquired addresses and other information from DHCPv6.

  * All other hosts acquired no configuration information.

- A=0, M=0, O=1

  * Windows 8.1 acquired addresses and other information from DHCPv6.

  - Windows 7, OSX 10.9 and IOS 8.0 acquired other information from DHCPv6.

  - Ubuntu 14.04 acquired no configuration information.

- A=0, M=1, O=0

  - All hosts acquired addresses and other information from DHCPv6.

- A=0, M=1, O=1

  - All hosts acquired addresses and other information from DHCPv6.

- A=1, M=0, O=0

  - Windows 8.1 acquired addresses from SLAAC and DHCPv6. It also acquired non-address information from DHCPv6.

  - All the other host acquired addresses from SLAAC

- A=1, M=0, O=1

  - Windows 8.1 acquired addresses from SLAAC and DHCPv6. It also acquired other information from DHCPv6.

  - All the other hosts acquired addresses from SLAAC and other information from DHCPv6.

- A=1, M=1, O=0

  - All hosts acquired addresses from SLAAC and DHCPv6. They also acquired other information from DHCPv6.

- A=1, M=1, O=1

  - All hosts acquired addresses from SLAAC and DHCPv6. They also acquired other information from DHCPv6.

As showed above, four inputs result in divergent behaviors.

A.1.3. Address Auto-configuration Behavior in State Transitions

The bullet list below describes behavior elicited during state transitions. The value x can represents both 0 and 1.

- Old state (M = x, O = x, A = 1) , New state (M = x, O = x, A = 0) (This means a SLAAC-configured host, which is regardless of DHCPv6 configured or not, receiving A in transition from 1 to 0.)

  - All the hosts retain SLAAC addresses until they expire

- Old state (M = 0, O = x, A = 1), New state (M = 1, O = x, A = 1) (This means a SLAAC-only host receiving M in transition from 0 to 1.)

  - Windows 7 acquires addresses from DHCPv6, immediately.

  - Ubuntu 14.04/OSX 10.9/IOS 8.0 acquires addresses from DHCPv6 only if the SLAAC addresses are allowed to expire

  - Windows 8.1 was not tested because it always acquire addresses from DHCPv6 regardless of the M flag setting.

- Old state (M = 1, O = x, A = x), New state (M = 0, O = x, A = x) (This means a DHCPv6-configured host receiving M in transition from 1 to 0.)

  - Windows 7 immediately released the DHCPv6 address

  - Windows 8.1/Ubuntu 14.04/OSX 10.9/IOS 8.0 keep the DHCPv6 addresses until they expire

- Old state (M = 1, O = x, A = 0), New state (M = 1, O = x, A = 1) (This means a DHCPv6-only host receiving A in transition from 0 to 1.)

  - All host acquire addresses from SLAAC

- Old state (M = 0, O = 1, A = x), New state (M = 1, O = 1, A = x) (This means a Stateless DHCPv6-configured host [RFC3736], which is regardless of SLAAC configured or not, receiving M in transition from 0 to 1 with keeping O=1)

  - Windows 7 acquires addresses and refreshes other information from DHCPv6

  - Ubuntu 14.04/OSX 10.9/IOS 8.0 does nothing

  - Windows 8.1 was not tested because it always acquire addresses from DHCPv6 regardless of the M flag setting.

- Old state (M = 1, O = 1, A = x), New state (M = 0, O = 1, A = x) (This means a Stateful DHCPv6-configured host, which is regardless of SLAAC configured or not, receiving M in transition from 0 to 1 with keeping O=1)

  - Windows 7 released all DHCPv6 addresses and refreshes all DHCPv6 other information.

  - Windows 8.1/Ubuntu 14.04/OSX 10.9/IOS 8.0 does nothing

A.2. Test Set 2

A.2.1. Test Environment

This test was built on real devices. All the devices are located on the same link.

- A DHCPv6 Server and specifically, a DHCP ISC Version 4.3.1 installed in CentOs 6.6. The DHCPv6 server is configured to provide both IPv6 addresses and RDNSS information.

- Two routers Cisco 4321 using Cisco IOS Software version 15.5(1)S.

- The following OS as clients:

  - Fedora 21, kernel version 3.18.3-201 x64

  - Ubuntu 14.04.1 LTS, kernel version 3.13.0-44-generic (rdnssd packet installed)

  - CentOS 7, kernel version 3.10.0-123.13.2.el7

  - Mac OS-X 10.10.2 Yosemite 14.0.0 Darwin

  - Windows 7

  - Windows 8.1

A.2.2. **Address/DNS Auto-configuration Behavior of Using Only One IPv6** Router and a DHCPv6 Server

In these scenarios there is two one router and, unless otherwise specified, one DHCPv6 server on the same link. The behaviour of the router and of the DHCPv6 server remain unchanged during the tests.

Case 1: One Router with the Management Flag not Set and a DHCPv6 Server

- Set up

  - One IPv6 Router with M=0, A=1, O=0 and an RDNSS is advertised

* A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS

o Results

* Fedora 21, MAC OS-X, CentOS 7 and Ubuntu 14.04 get an IPv6 address and an RDNSS from the IPv6 router only.

* Windows 7 get an IPv6 address from the router only, but they do not get any DNS information, neither from the router nor from the DHCPv6 server. They also do not get IPv6 address from the DHCPv6 server.

* Windows 8.1 get an IPv6 address from both the IPv6 router and the DHCPv6 server, despite the fact that the Management flag (M) is not set. They get RDNSS information from the DHCPv6 only.

Case 2: One Router with Conflicting Parameters and a DHCPv6 Server

o Set up

* One IPv6 Router with M=0, A=1, O=1 and an RDNSS is advertised

* A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS

o Results

* Fedora 21, Centos 7 and Ubuntu 14.04 get IPv6 address using SLAAC only (no address from the DHCPv6 server).

  + Fedora 21, Centos 7 get RDNSS from both the RAs and the DHCPv6 server. The RDNSS obtained from the router has a higher priority though.

  + Ubuntu 14.04 gets an RDNSS from the router, and a "domain search list" information from the DHCPv6 server - but not RDNSS information.

* MAC OS-X also gets RDNSS from both, IPv6 address using SLAAC (no IPv6 address from the DHCPv6 server) but the RDNSS obtained from the DHCPv6 server is first (it has a higher priority). However, the other obtained from the RAs is also present.

* Windows 7 and Windows 8.1 obtain IPv6 addresses using SLAAC and RDNSS from the DHCPv6 server. They do not get IPv6 address

from the DHCPv6 server. Compare the Windows 8.1 behaviour with the previous case.

Case 3: Same as Case 2 but Without a DHCPv6 Server

- o Set up

  - One IPv6 Router with M=0, A=1, O=1 and an RDNSS is advertised

  - no DHCPv6 present

- o Results

  - Windows 7 and Windows 8.1 get an IPv6 address using SLAAC but they do not get RDNSS information.

  - MAC OS-X, Fedora 21, Centos 7 and Ubuntu 14.04 get an IPv6 address using SLAAC and RDNSS from the RAs.

Case 4: All Flags are Set and a DHCPv6 Server is Present

- o Set up

  - One IPv6 Router with M=1, A=1, O=1 and an RDNSS is advertised

  - A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS

- o Results

  - Fedora 21 and Centos 7:

    - They get IPv6 address both from SLAAC and DHCPv6 server.

    - They get RDNSS both from RAs and DHCPv6 server.

    - The DNS of the RAs has higher priority.

  - Ubuntu 14.04:

    - It gets IPv6 address both using SLAAC and from the DHCPv6 server.

    - It gets RDNSS from RAs only.

    - From the DHCPv6 server it only gets "Domain Search List" information, no RDNSS.

- MAC OS-X:

  - It gets IPv6 addresses both using SLAAC and from the DHCPv6 server.

  - It also gets RDNSS both from RAs and the DHCPv6 server.

  - The DNS server of the DHCPv6 has higher priority.

- Windows 7 and Windows 8.1:

  - They get IPv6 address both from SLAAC and DHCPv6 server.

  - They get RDNSS only from the DHCPv6 server.

Case 5: All Flags are Set and There is No DHCPv6 Server is Present

- Set up

  - One IPv6 Router with M=1, A=1, O=1 and an RDNSS is advertised

  - no DHCPv6 is present

- Results

  - Windows 7 and Windows 8.1 get an IPv6 address using SLAAC but no RDNSS information.

  - MAC OS-X, Fedora 21, Centos 7, Ubuntu 14.04 get an IPv6 address using SLAAC and RDNSS from the RAs.

Case 6: A Prefix is Advertised by RAs but the 'A' flag is not Set

- Set up

  - An IPv6 Router with M=0, A=0 (while a prefix information is advertised), O=0 and an RDNSS is advertised.

  - DHCPv6 is present

- Results

  - Fedora 21, Centos 7, Ubuntu 14.04 and MAC OS-X:

    - They do not get any IPv6 address (neither from the RAs, nor from the DHCPv6).

    - They get a RDNSS from the router only (not from DHCPv6).

  - Windows 8.1

    - They get IPv6 address and RDNSS from the DHCPv6 server ("last resort" behaviour).

    - They do not get any information (neither IPv6 address not RDNSS) from the router.

  - Windows 7:

    - They get nothing (neither IPv6 address nor RDNSS) from any source (RA or DHCPv6).

A.2.3. Address/DNS Auto-configuration Behavior of Using Two IPv6 Router and a DHCPv6 Server

these scenarios there are two routers on the same link. At first, only one router is present (resembling the "legitimate router)", while the second one joins the link after the clients first configured by the RAs of the first router. Our goal is to examine the behaviour of the clients during the interchange of the RAs from the two different routers.

Case 7: Router 1 Advertising M=0, O=0 and RDNSS, and then Router 2 advertising M=1, O=1 while DHCPv6 is Present

- Set up

  - Initially:

    - One IPv6 router with M=0, O=0, A=1 and RDNSS advertised and 15 seconds time interval of the RAs

  - After a while (when clients are configured by the RAs of the above router):

    - Another IPv6 router with M=1, O=1, no advertised prefix information, and 30 seconds time interval of the RAs.

    - A DHCPv6 server on the same link providing IPv6 addresses and RDNSS.

- Results

  - MAC OS-X and Ubuntu 14.04:

    - Initially they get address and RDNSS from the first router.

    - When they receive RAs from the second router, they never get any information (IPv6 address or RDNSS) from the DHCPv6 server.

- Windows 7:

    - Initially they get address from the first router - no RDNSS.

    - When they receive RAs from the second router, they never get any information (IPv6 address or RDNSS) from the DHCPv6 server.

- Fedora 21 and Centos 7:

    - Initially they get IPv6 address and RDNSS from the RAs of the first router. o

    - When they receive an RA from router 2, they also get an IPv6 address and RDNSS from the DHCPv6 server while retaining the ones (IPv6 address and RDNSS) obtained from the RAs of the first router. The RDNSS obtained from the first router has a higher priority than the one obtained from the DHCPv6 server (probably because it was received first). o

    - When they receive again RAs from the first router, they lose/forget the information (IPv6 address and RDNSS) obtained from the DHCPv6 server.

- Windows 8.1:

    - Initially, they get just an IPv6 address from the first router 1 - no RDNSS information (since they do not implement RFC 6106).

    - When they receive RAs from the second router, then they also get an IPv6 address from the DHCPv6 server, as well as RDNSS from it. They do not lose the IPv6 address obtained by the first router using SLAAC.

    - When they receive RA from the first router, they retain all the obtained so far information (there isn't any change).

Case 8: (Router 2) Initially M=1, O=1 and DHCPv6, then 2nd Router (Router 1) Rogue RAs Using M=0, O=0 and RDNSS Provided

o Set up

- Initially:

    + One IPv6 router with M=1, O=1, no advertised prefix information, and 30 seconds time interval of the RAs.

    + A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.

  * After a while (when clients are configured by the RAs of the above router):

    + Another IPv6 router with M=0, O=0, A=1, RDNSS advertised and 15 seconds time interval of the RAs.

- Results

  * Fedora 21 and Centos 7:

    + At first, they get information (IPv6 address and RDNSS) from the DHCPv6 server.

    + When they receive RAs from the second router, they get address(es) and RDNSS from these RAs. At the same time, the IPv6 address and the RDNSS obtained from the DHCPv6 server are gone.

    + When they receives again an RA from the first router, they perform the DHCPv6 Confirm/Reply procedure and they get an IPv6 address and RDNSS from the DHCPv6 server while retaining the ones obtained from the RAs of the second router. Moreover, the RDNSS from router 1 has higher priority than the one from DHCPv6.

  * Ubuntu 14.04:

    + At first, it gets information (IPv6 address and RDNSS) from the DHCPv6 server.

    + When it receives RAs from the second router, it also gets information from it, but it does not lose the information obtained from the DHCPv6 server. It retains both. It only gets "Domain Search list" from the DHCPv6 server-no RDNSS information.

    + When it receives RAs from the first router, there is no change; it retains all the obtained information.

  * Windows 7:

    - Initially they get IPv6 address and RDNSS from the DHCPv6 server.

    - When they get RAs from the second router, they lose this information (IPv6 address and RDNSS obtained from the DHCPv6 server) and they get only SLAAC addresses using the RAs of the second router-no RDNSS.

    - When they receive RAs from the first router again, they get RDNSS and IPv6 address from the DHCPv6 server, but they also keep the SLAAC addresses.

- Windows 8.1:

    - Initially they get information (IPv6 address and RDNSS) from the DHCPv6 server.

    - When they receive RAs from the second router, they never get any information from them.

- MAC OS-X:

    - Initially it gets information (IPv6 address and RDNSS) from the DHCPv6 server.

    - When it gets RAs from the second router, it also gets a SLAAC IPv6 address but no RDNSS information from the RAs of this router. It also does not lose any information obtained from DHCPv6.

    - When it gets RAs from the first router again, the situation does not change (IPv6 addresses from both the DHCPv6 and SLAAC process are retained, but RDNSS information only from the DHCPv6 server).

Authors' Addresses

Bing Liu
Huawei Technologies
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: leo.liubing@huawei.com

Sheng Jiang
Huawei Technologies
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

Xiangyang Gong
BUPT University
No.3 Teaching Building
Beijing University of Posts and Telecommunications (BUPT)
No.10 Xi-Tu-Cheng Rd.
Hai-Dian District, Beijing
P.R. China

Email: xygong@bupt.edu.cn

Wendong Wang
BUPT University
No.3 Teaching Building
Beijing University of Posts and Telecommunications (BUPT)
No.10 Xi-Tu-Cheng Rd.
Hai-Dian District, Beijing
P.R. China

Email: wdwang@bupt.edu.cn

Enno Rey
ERNW GmbH

Email: erey@ernw.de