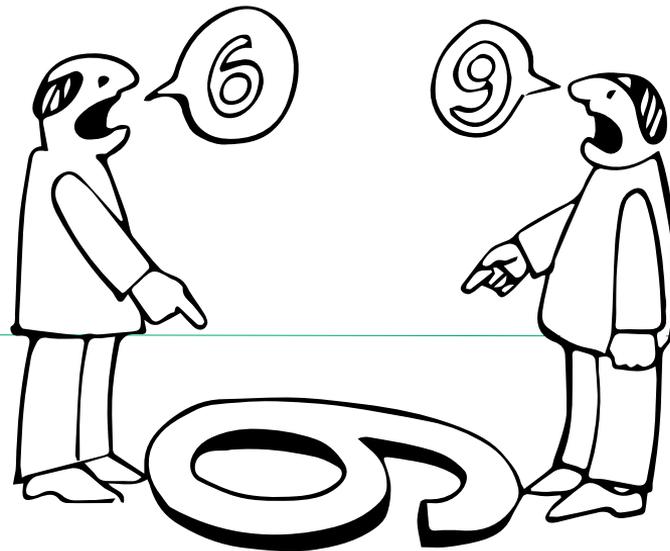


OS IPv6 Behavior in Conflicting Environments

Enno Rey, erey@ernw.de
[@Enno_Insinuator](https://twitter.com/Enno_Insinuator)



Who I Am



- Founder and managing director of vendor independent network consulting & security assessment company ERNW.
- Old-school network guy with some background in large scale operations.
- Involved with IPv6 since 1999 and regularly blogging at www.insinator.net.

Agenda



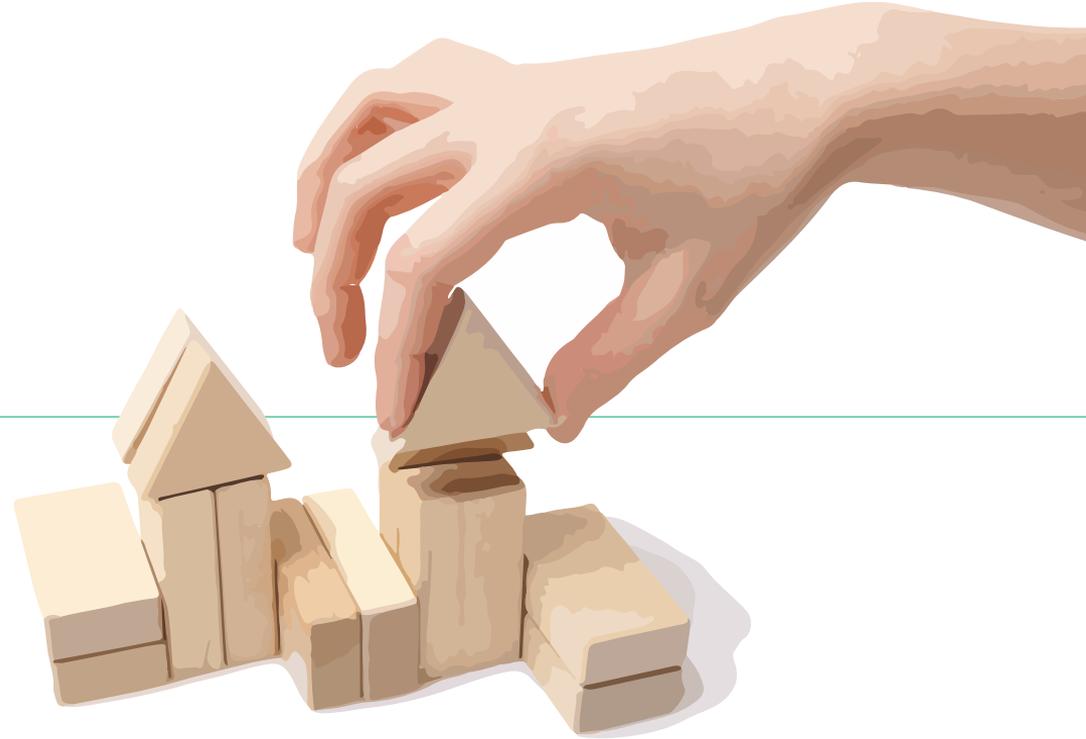
- Fundamentals
 - Quick Refresher of Basics & Specifications

- Results from the Lab
 - Some Surprises (?)

- Conclusions
 - What All this Means from an Operations Perspective

Fundamentals

What the textbook tells you



Address Autoconfig Overview



- IPv6 interfaces are meant to configure themselves automatically, in terms of "basic IP parameters".
 - Even without DHCPv6.
 - In particular without DHCPv6!
 - Remember: IPv6 = consumer technology.
- Link-local addresses are always configured, for each interface.
- Using the *router discovery* process, other addresses, router addresses and other configuration parameters are selected.

Types of Autoconfiguration

- Stateless
 - Via *Router Advertisement Messages* (with one or more prefix)
 - Can (in theory) also distribute "other parameters", see RFC 6106.
 - SLAAC: "stateless address autoconfiguration"
- Stateful
 - Usage of a *Stateful Address Protocol* (e.g. DHCPv6).
- Stateless with DHCP
 - Use of Router Advertisement messages for allocation of prefixes
 - In addition, DHCP for "other parameters" (e.g. DNS Server, Domain Search List).

(In all cases there is always at least one link-local address anyway!)



Neighbor Discovery Protocol RFC 4861



- *Neighbor Discovery* (ND) provides mechanisms for the following tasks:
 1. Neighbor Discovery / Address Resolution
 2. Router Discovery
 3. Prefix Discovery
 4. Parameter Discovery
 5. Address Autoconfiguration
 6. Next-Hop Determination
 7. Neighbor Unreachability Detection
 8. Duplicate Address Detection
 9. Redirects

Router Discovery

- Used to detect routers that are connected to the local network.
- IPv6 router discovery can also help to provide the following information:
 - Default value for the "Hop Limit" field
 - Whether any "stateful address protocol" (DHCPv6) should be used.
 - Settings for the "Retransmission Timer"
 - The network prefix for the local network
 - The MTU of the network
 - Mobile IPv6 Information
 - Routing Information



Multicast Router Solicitation Message

Ethernet Header

- Dest.-MAC: 33-33-00-00-00-02

IPv6 Header

- Source-IP:::
- Dest.-IP: FF02::2
- Hop limit: 255

Router Solicitation



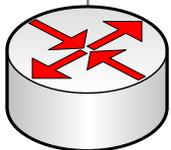
Alice

MAC: 00-01-02-03-04-05

IP: none

1. Multicast Router Solicitation

Router Solicitation



MAC: 00-11-22-33-44-55

IP: FE80::211:22FF:FE33:4455

Router



Router Advertisement Message

Ethernet Header

- Dest.-MAC: 33-33-00-00-00-01

IPv6 Header

- Source-IP: FE80::211:22FF:FE33:4455
- Dest.-IP: FF02::1
- Hop limit: 255

Router Advertisement Header

- Current Hop Limit, Flags, Router Lifetime, Reachable and Retransmission Timers

Neighbor Discovery Options

- Source Link-Layer Address
- MTU
- Prefix Information

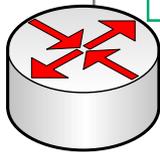


MAC: 00-01-02-03-04-05
IP: none

Router Advertisement

2. Multicast Router Advertisement

Router



MAC: 00-11-22-33-44-55
IP: FE80::211:22FF:FE33:4455



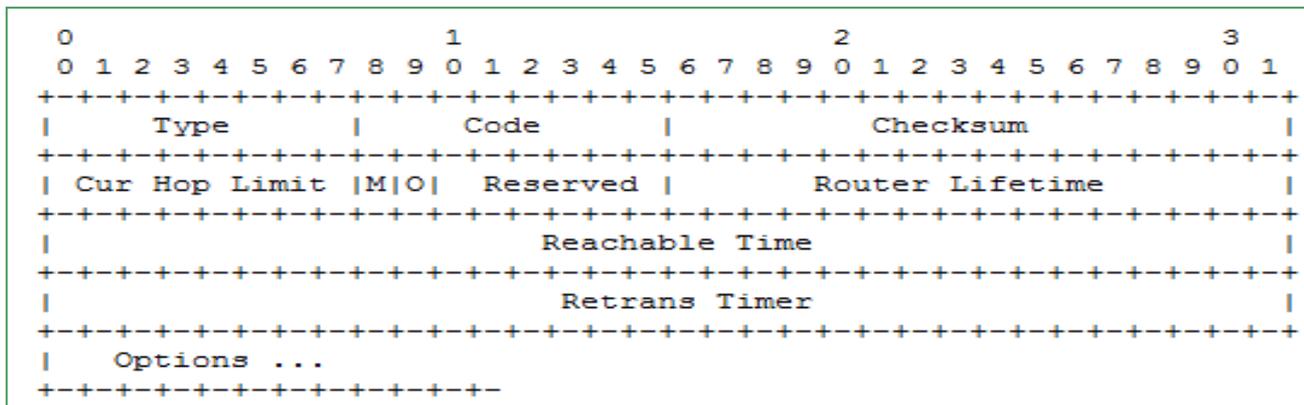
Router Advertisements, Flags (I)



- The “*Autonomous address configuration*” (A) flag. When set, this flag indicates that this prefix can be used for stateless address autoconfiguration, as specified in [RFC4862].

Router Advertisements, Flags (II)

- Routers can inform adjacent hosts (neighbors on the local link) that additional configuration parameters (like a DNS server) are available over a stateful configuration protocol (DHCPv6).
- In the router advertisement header two flags (M and O) can be included which can be set to inform the clients that additional configuration parameters are available.



O-Flag



- 1-bit "other configuration" flag
- When set, it indicates that other configuration information is available via DHCPv6.
- Examples of such information are DNS-related information (DNS Server, DNS Suffix).
- Both flags are defined in RFC 4861 (Section 4.2).

M-Flag



- 1-bit "Managed address configuration" flag.
- When set, it indicates that addresses are available through DHCPv6.
- If the M flag is set, the O flag is redundant and can be ignored because DHCPv6 will return all available configuration information.
- If neither M nor O flags are set, this indicates that no information is available via DHCPv6.
 - Rly? See below...

And Finally There's RFC 6106

[\[Docs\]](#) [\[txt\]](#) [\[pdf\]](#) [\[draft-ietf-6man-d...\]](#) [\[Diff1\]](#) [\[Diff2\]](#)

PROPOSED STANDARD

J. Jeong
Brocade/ETRI
S. Park
SAMSUNG Electronics
L. Beloel
France Telecom R&D
S. Madanapalli
iRam Technologies
November 2010

Internet Engineering Task Force (IETF)
Request for Comments: 6106
Obsoletes: [5006](#)
Category: Standards Track
ISSN: 2070-1721

IPv6 Router Advertisement Options for DNS Configuration

Abstract

This document specifies IPv6 Router Advertisement options to allow IPv6 routers to advertise a list of DNS recursive server addresses and a DNS Search List to IPv6 hosts.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6106>.

Copyright Notice

DHCPv6

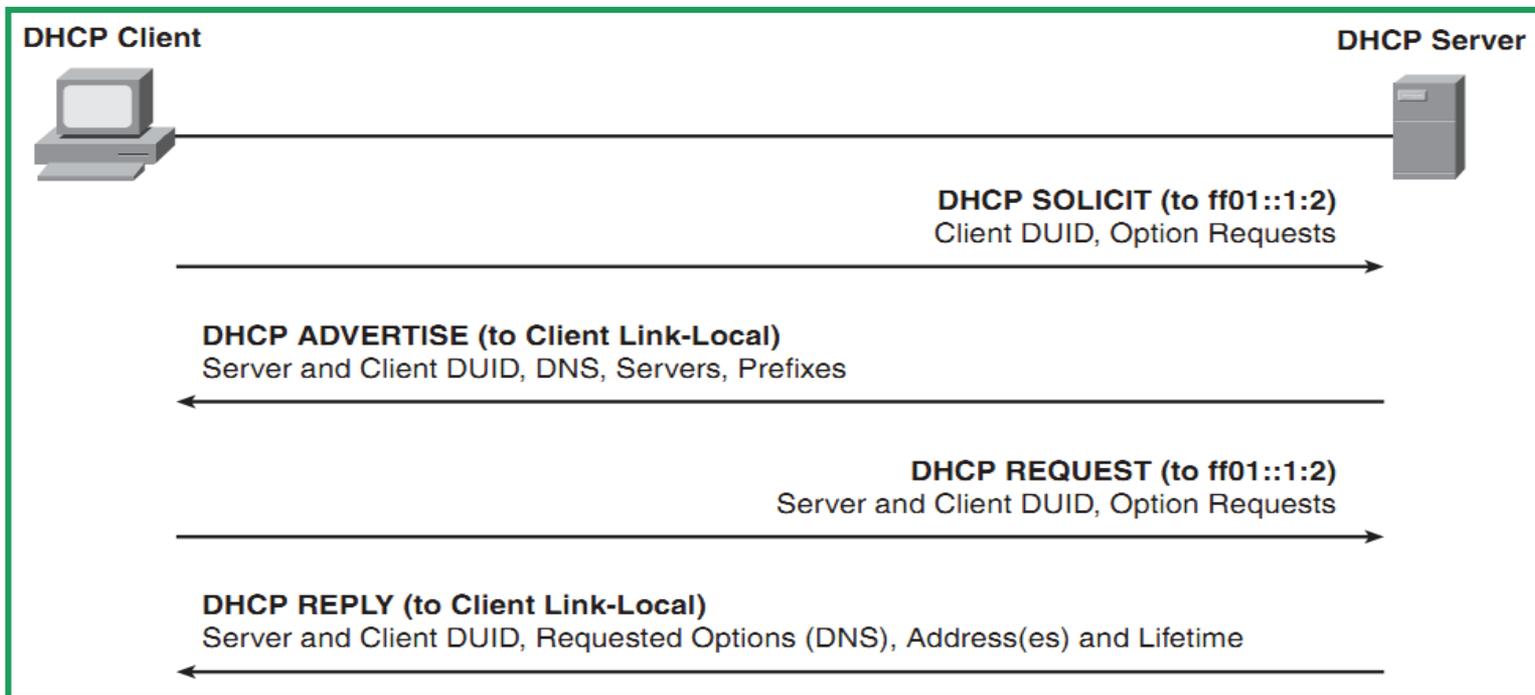


- Specified (initially|mainly) in RFC 3315.
- Uses UDP Ports 546 (Clients) and 547 (Server/Relays).
- DHCPv6 uses multicast packets in IPv6.
- Clients/Server will be identified by:
 - DUID + IAID(s)
- Components of a DHCPv6 Infrastructure
 - DHCPv6 Clients
 - DHCPv6 Server
 - DHCPv6 Relay Agents

DHCPv6 Message Types

DHCPv6 Message Type	DHCPv4 Message Type
SOLICIT (1)	DHCPDISCOVER
ADVERTISE (2)	DHCPOFFER
REQUEST (3), RENEW (5), REBIND (6)	DHCPREQUEST
REPLY (7)	DHCPACK/DHCPNAK
RELEASE (8)	DHCPRELEASE
INFORMATION-REQUEST (11)	DHCPINFORM
DECLINE(9)	DHCPDECLINE
CONFIRM (4)	- No equivalent -
RECONFIGURE (10)	DHCPFORCERENEW
RELAY-FORW (12), RELAY REPLY (13)	- No equivalent -

DHCP Message Exchange ["M-Flag Variant"]



Main Differences

On the Protocol Level



- There is no “route option“ in DHCPv6
- Concept of DUID
- The (Non-) Role of DHCPv6 in IPv6’s “Subnet Model“ (RFC 5942)

Differences

Here's another one not to strictly blame on the protocol itself.



- (Informational) RFC 6434 IPv6 Node Requirements, sect. 5.9.5:
 - “[A]ll hosts SHOULD implement address configuration via DHCPv6.”

- For the record, RFC 2119 states:
 - “SHOULD This word[...] mean[s] that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.”

DHCPv6 Support by OSs

What could possibly go wrong? Who could possibly deviate?



<https://code.google.com/p/android/issues/detail?id=32621>

- “Marking [Support for DHCPv6] declined until there is a compelling use case.
 - -- Lorenzo Colitti (Google) on Dec 07 2014
- → No DHCPv6 on Android
 - Except for the *Fairphone*.
- There are people who expect that Android is going to be one of the major OS for #IoT...

Once upon a Time

When our ancestors did the initial specs of IPv6



- They had a certain place for DHCPv6 in mind, within the IPv6 universe.
- This happened to be a very different role from the (at the time developing) role of DHCP in IPv4.
- Tell you what: this is going to haunt you.

What Do You Mean?

Can you please elude?



- DHCPv4 was meant to be *exclusive*.
 - Either configure basic IPv4 properties manually *or* get the stuff from DHCPv4.
 - Once DHCPv4 is used, it's used for everything.

- DHCPv6 is meant to be *complementary*.
 - It can (and must) be mixed with other spicy stuff.
 - Add some #RFCambiguity to the mix.

- To fully understand what this means, let's step back one step and look at...

Relevant Specifications



RFC 4861

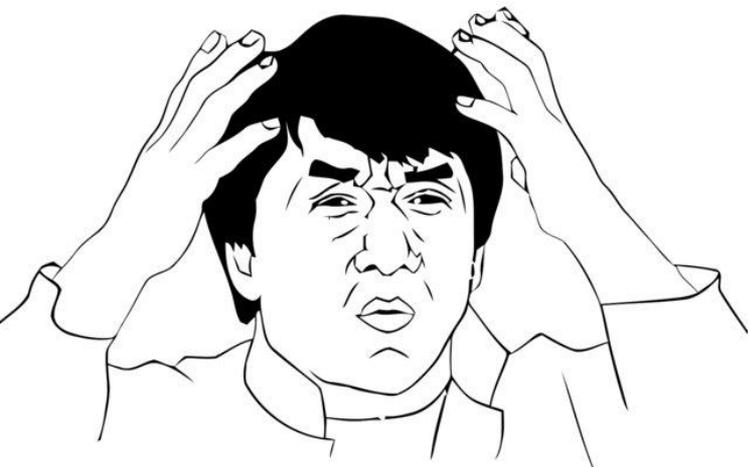


- Sect. 4.2
 - “If neither M nor O flags are set, this indicates that no information is available via DHCPv6.”

- If the M flag is set, the O flag is redundant and it can be ignored.

Some More Quotes

Not much RFC 2119 in there, is it?



- RFC 4862, 5.5.2 *Absence of Router Advertisements*
 - “Even if a link has no routers, the DHCPv6 service to obtain addresses may still be available, and hosts may want to use the service.”
- RFC 4862, 5.6 *Configuration Consistency*
 - “If the same configuration information is provided by multiple sources, the value of this information should be consistent.”
 - “In any case, if there is no security difference, the most recently obtained values SHOULD have precedence over information learned earlier.”

RFC 6106

“1.2 Coexistence of RA Options and DHCP Options for DNS Configuration



Two protocols exist to configure the DNS information on a host, the Router Advertisement options described in this document and the DHCPv6 options described in [RFC3646]. They can be used together.

The rules governing the decision to use stateful configuration mechanisms are specified in [RFC4861]. **Hosts conforming to this specification MUST extract DNS information from Router Advertisement messages**, unless static DNS configuration has been specified by the user.

If there is DNS information available from multiple Router Advertisements and/or from DHCP, the host **MUST** maintain an ordered list of this information as specified in Section 5.3.1.

RFC 6106

Section 5.3.1



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.

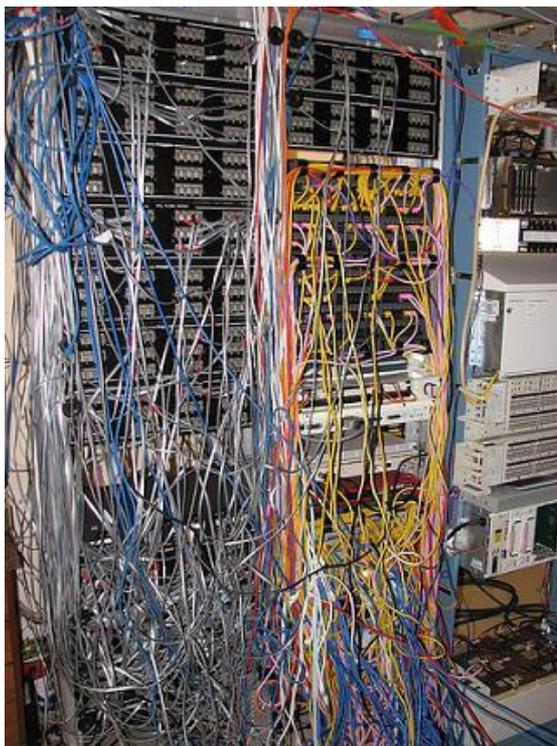
Unless explicitly specified for the discovery mechanism, the exact number of addresses and domain names to keep is a matter of local policy and implementation choice.

However, the ability to store at least three RDNSS addresses (or DNSSL domain names) from at least two different sources is RECOMMENDED.

The DNS options from Router Advertisements and DHCP SHOULD be stored into the DNS Repository and Resolver Repository so that information from DHCP appears there first and therefore takes precedence.

Thus, the DNS information from DHCP takes precedence over that from RA for DNS queries.

In Short



- It's a mess!
At least on the specs level.

Problem Statement

From a High-Level Perspective



Problem Statement (I)



- IPv6 provides two mechanisms for one task, that is provisioning of IP parameters to nodes.

Problem Statement (II)

There's two mechanisms

- They are independent.
 - Well, mostly.
- In many environments both of them are needed, in some combination.
 - In particular this applies in (wrt OSs, devices) heterogeneous environments.
Read: probably in pretty much all of your environments.
- In some environments different groups might be responsible for operating them.
 - Why did you add this to the "problem statement"? Well...
- There's differences as for the degree of vendor support & their strategies.



Problem Statement (III)

Let's look at the specs...



- Some properties and elements have been enhanced over time, e.g. RFC 6106.
 - Evolution is a good thing. Seriously!
- Still, there's a certain (alas, when it comes to IPv6: usual) amount of ambiguity and vagueness in the main RFCs.

Problem Statement (IV)



- The “cooperation” of those two mechanisms has been discussed quite a bit, both in the specs and in “the relevant fora”.
- However, not so much as for scenarios where the information provided by them is conflicting.
- This is what this talk is about.

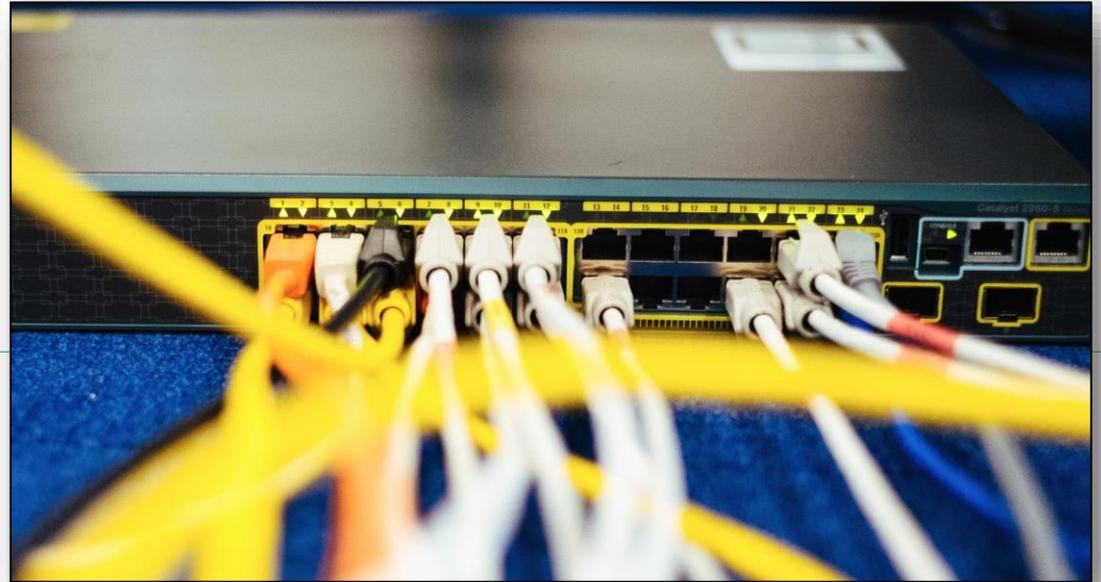
Problem Statement (IV)

Can this (“contradiction scenario”) happen?

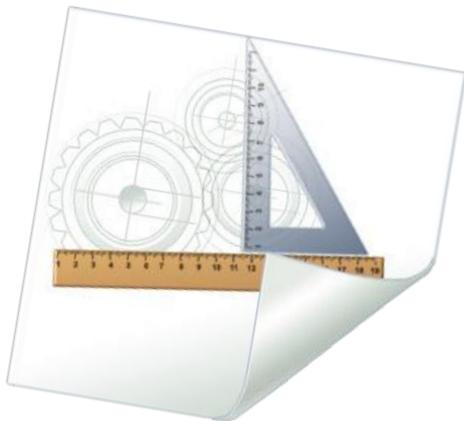


- Human error
 - Both on the *active failure* and *latent failure* level.
- Conflicting/differing vendor default settings
 - Network devices
 - CPEs!
 - Keep in mind: there might be any OS in customers' networks.
- Attacker injecting nasty packets
 - Boils down to “standard local-link sec” discussion → I will only briefly cover this.

From the Lab



Lab Setup



See also:

https://www.ernw.de/download/ERNW_White_paper_IPv6_RAs_RDNSS_DHCPv6_Conflicting_Parameters.pdf

- A DHCPv6 Server (DHCP ISC Version 4.3.1) installed on CentOS 6.6 . The DHCPv6 server is configured to provide both IPv6 addresses and RDNSS information.
- Two (2) 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
 - CentOS 7, kernel version 3.10.0-123.13.2.el7
 - Mac OS X 10.10.2 Yosemite
 - Windows 7
 - Windows 8.1

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

Router: M=0, A=1, O=0 and an RDNSS is advertised.

DHCPv6 server on the same link offering IPv6 addresses & RDNSS

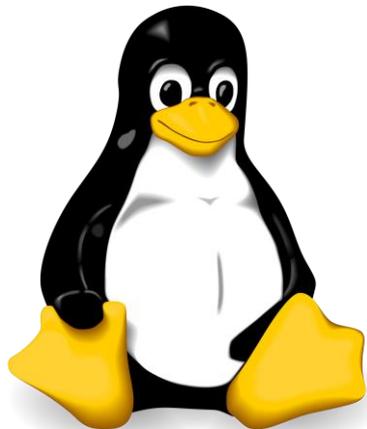


- Fedora 21, MAC OS X, CentOS 7 and Ubuntu 14.04
 - Get an IPv6 address and an RDNSS from the IPv6 router only.
- Windows 7
 - Get an IPv6 address from the router only, but they do not get any DNS information, neither from the router nor from the DHCPv6 server. They also do not get IPv6 address from 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

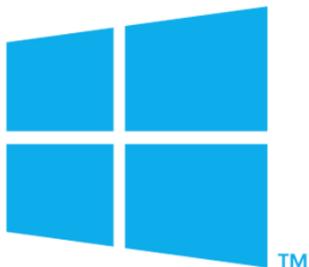
Router: M=0, A=1, O=1, and an RDNSS is advertised.

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



- Fedora 21, Centos 7 and Ubuntu 14.04
 - get IPv6 address using SLAAC only.
 - 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.

Case 2 Results cont'd



- MAC OS X
 - 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 an IPv6 address from the DHCPv6 server.
 - ⇔ compare the Windows 8.1 behaviour with the previous case.

Additional Observations

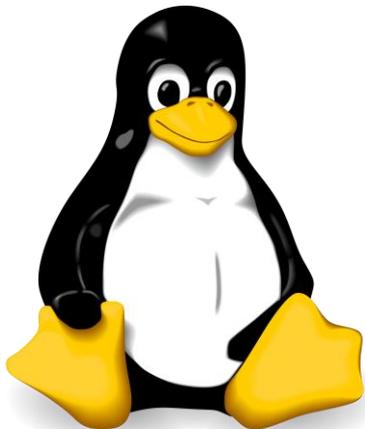
- [draft-ietf-v6ops-dhcpv6-slaac-problem-03] explicitly discusses the role of *state transitions*.
- We can confirm that these lead to particularly interesting effects.
 - → Pay special attention in times when you perform those deliberately.
Be prepared for surprises...
- In general the *order of events* seems to play a role, too.
 - See also test cases with two routers.



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

Router: M=1, A=1, O=1, and an RDNSS is advertised.

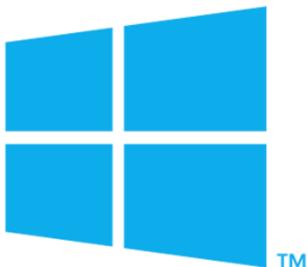
A DHCPv6 server on the same link advertising IPv6 addresses and RDNSS.



- Fedora 21 and Centos 7:
 - They get IPv6 addresses 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 addresses 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.

Case 4 Results cont'd



→ 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 from DHCPv6 has higher priority.

→ Windows 7 and Windows 8.1:

- They get IPv6 addresses both from SLAAC and DHCPv6 server.
- They get RDNSS only from the DHCPv6 server.

Summary

	Scenario	Collected Information	Windows 7	Windows 8.1	Ubuntu 14	Centos 7	Fedora 21	MAC OS-X
1	A=1, M=0, O=0 DHCPv6 present	IPv6 address	router	both	router	router	router	router
		RDNSS	-	DHCPv6	router	router	router	router
2	A=1, M=0, O=1 DHCPv6 present	IPv6 address	router	router	router	router	router	router
		RDNSS	DHCPv6	DHCPv6	router	router/DHCPv6	router/DHCPv6	DHCPv6/router
3	A=1, M=0, O=1 no DHCPv6 present	IPv6 address	router	router	router	router	router	router
		RDNSS	-	-	router	router	router	router
4	A=1, M=1, O=1 DHCPv6 present	IPv6 address	both	both	both	both	both	both
		RDNSS	DHCPv6	DHCPv6	router	router/DHCPv6	router/DHCPv6	DHCPv6/router
5	A=1, M=1, O=1 no DHCPv6 present	IPv6 address	router	router	router	router	router	router
		RDNSS	-	-	router	router	router	router
6	A=0, M=0, O=0 DHCPv6 present	IPv6 address	-	DHCPv6	-	-	-	-
		RDNSS	-	DHCPv6	router	router	router	Router

https://www.ernw.de/download/ERNW_Whitepaper_IPv6_RAs_RDNSS_DHCPv6_Conflicting_Parameters.pdf
<https://tools.ietf.org/html/draft-ietf-v6ops-dhcpv6-slaac-problem-03>

More Stuff from the Lab



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

Initially:

One IPv6 router with the following settings:

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 the following:

M=1, O=1, no advertised prefix information, and 30 seconds time interval of the RAs.

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.



Case 7 Results cont'd



- Fedora 21 and Centos 7:
 - Initially they get IPv6 address and RDNSS from the RAs of the first router.
 - 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).
 - When they receive again RAs from the first router, they lose/forget the information (IPv6 address and RDNSS) obtained from the DHCPv6 server.
 - Troubleshooting nightmare...

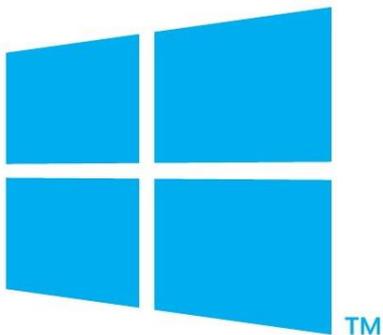
Case 7 Results cont'd

– 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.



Case 7 Results cont'd



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

Summary

Scenario		Collected Information	Windows 7	Windows 8.1	Ubuntu 14	Centos 7	Fedora 21	MAC OS-X	
7	Initial Situation	Router 1: A=1, M=0, O=0, RDNSS, 15 sec. RA interval	IPv6 address	router	router	router	router	router	router
			RDNSS	-	-	router	router	router	router
	Later addition	Router 2: M=1, O=1, no advertised prefix, 30 sec. RA interval DHCPv6 server.	IPv6 address	router	both	router	Both	both	router
			RDNSS	-	DHCPv6	router	Router/DHCPv6	Router/DHCPv6	router
	Router 1 RAs received again		IPv6 address	router	both	router	router	router	router
			RDNSS	-	DHCPv6	router	router	router	router
8	Initial Situation	Router 2: M=1, O=1, no advertised prefix, 30 sec. RA interval DHCPv6 server	IPv6 address	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6
			RDNSS	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6	DHCPv6
	Later addition	Router 1: A=1, M=0, O=0, RDNSS, 15 sec. RA interval	IPv6 address	Router 1	DHCPv6	both	Router 1	Router 1	both
			RDNSS	-	DHCPv6	Router 1	Router 1	Router 1	DHCPv6
	Router 2 RAs received again		IPv6 address	Both		both	Both	both	both
			RDNSS	DHCPv6		Router 1	Router1/DHCPv6	Router1/DHCPv6	DHCPv6

Conclusions





- Don't assume a certain OS' IPv6 behavior just because:
 - "the specs say so"
 - "another OS does it that way"
 - you have a good understanding of IPv4.
- Sorry guys ;-)
- Test, test, test!
 - Helps to gain (even more) IPv6 knowledge anyway.
 - Yes, pls allocate budget for test lab.

Keep RFC 1925 in Mind



- "(4) Some things in life can never be fully appreciated nor understood unless experienced firsthand. Some things in networking can never be fully understood by someone who neither builds commercial networking equipment nor runs an operational network."

- Deploying IPv6 is not a paper exercise. You need hands-on experience!

- Did you note Jeff Carrell gives his cool workshops at the *IPv6 Business Conference*?
 - Mark June 17-19 2015 in your calendar!

**IPv6 Business
Conference**

Organized by
SWISS
IPv6
COUNCIL

**2015
June, 18**

Operations Perspective

- Keep configs/properties related to IPv6 parameter provisioning in sync, at all times
 - IPv6 transition might be an opportunity to re-think your ops model.
 - Yes, we understand you'll be happy to survive that one mostly unscathed, hence concentrate on one task at a time. Still #justsayin ;-)



Planning Perspective

Considerations how to set up the whole SLAAC/DHCPv6 thing

- In short: it depends 😊
- **Seriously: it depends heavily on the client base you want to support. Here's some thoughts:**
 - in case there's Android devices, your routers should advertise RDNSS info (RFC 6106), else the Android clients will have to rely on their IPv4 DNS or manual kludges. RFC 6106 is supported since Lollipop.
 - in case you don't have Android devices, you might go `_without_` advertising RDNSS in RAs, for the simple reason of reducing potential for errors/"unexpected behavior".
 - once you go with m-flag DHCPv6 clearing the A-flag in prefix information, but leaving the L-flag set (to "circumvent RFC 5942") is usually a good idea.
 - Ofc, you can't do this once there's Android devices as those won't generate any (non LL) address then.
 - we observe that most of our customers strive to go with m-flag DHCPv6. that's just an observation...

Troubleshooting

For the poor sod responsible...



A helpful resource:

<https://wikispaces.psu.edu/display/ipv6/IPv6+Rosetta+Stone>

- You should know how to diagnose a node's exact properties on the OS level
 - incl. types of addresses and order of name resolution
 - "netsh int ipv6" commands on Win
 - "ip -6 add show" on Linux
 - btw: /etc/resolv.conf not relevant on Mac

- The truth is in the packets...

Troubleshooting

In such scenarios



- Being familiar with the following certainly helps
 - Flags in router advertisements
 - Main DHCPv6 messages
 - IPv6 Subnet Model (RFC 5942) and its (non-) relationship with DHCPv6

Summary



- There's a certain learning curve when it comes to IPv6.
- Just looking at the specs might not be sufficient.
- Start now ;-)

There's never enough time...

THANK YOU...



...for yours!

Slides:

<https://www.insinator.net>



Save the Dates

IPv6 Business Conference

Organized by
SWISS
IPv6
COUNCIL

| 2015
June, 18

- **Pre-Conference Day** – Wednesday, 17. June 2015
IPv6 Workshop: Build it, Use it
with Jeff Carrell
 - Hands-On
- **IPv6 Business Conference** – Thursday, 18. June 2015
- **Post-Conference Day** – Friday, 19. June 2015
IPv6 Interactive Addressing Workshop with Practical Hands-on Labs with Jeff Carrell
 - Hands-On, Build your own lab and take it home!
- **Do you want to be a sponsor?**



Guys, we would love to see you in Heidelberg!
March, 14-18 2016
Heidelberg, Germany
TROOPERS - Make the world a safer



More info & extensive archives @ www.troopers.de

Questions?



- You can reach us at: 
 - erey@ernw.de, www.ernw.de
- Our blog: 
 - www.insinuator.net
- Follow me at: 
 - [@Enno_Insinuator](https://twitter.com/Enno_Insinuator)