

mboned WG
Internet-Draft
Intended status: Standards Track
Expires: January 02, 2014

Y. Cao
C. Wang
W. Meng
ZTE Corporation
B. Khasnabish
ZTE USA, Inc
July 01, 2013

IPv4-IPv6 Multicast Address Dynamic Conversion
draft-cao-sunset4-v4v6-mcast-addr-conversion-02

Abstract

This draft describes a mechanism for stateless conversion of IPv4 multicast address to IPv6 multicast address and vice versa, using different rules. These rules can be used in both IPv4-IPv6 translation or encapsulation. This solution can be used in any scenarios describe in [RFC6144].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 02, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Convention and Terminology	2
3. Architecture	3
4. IPv4/IPv6 Multicast Address Conversion	4
4.1. Rule Design	4
4.2. IPv4 Multicast Address Suffix-embedded IPv6 Multicast Address	5
4.3. Full IPv4 Multicast Address-embedded IPv6 Multicast Address	5
5. Forwarding	6
5.1. From IPv4 Multicast System to IPv6 Multicast System . . .	6
5.2. From IPv6 Multicast System to IPv6 Multicast System . . .	6
6. Backwards compatibility	7
7. Security Considerations	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Authors' Addresses	7

1. Introduction

This draft describes a mechanism for stateless translation between IPv4 multicast address and IPv6 multicast address using different rules. These rules can be used in both IPv4-IPv6 translation or encapsulation. This solution can be used in any scenarios describe in [RFC6144].

The approach described in this draft is fully compatible with [I-D.ietf-mboned-64-multicast-address-format].

2. Convention and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Rule_IPv6_M_Prefix/Length:

Define an IPv6 Prefix assigned by a Service Provider for a IPv4/IPv6 Multicast Address Conversion rule.

Rule_IPv4_M_Prefix/Length:

Define an IPv4 Prefix assigned by a Service Provider for a IPv4/IPv6 Multicast Address Conversion rule.

Rule_IPv4_Offset:

Define an offset where IPv4 Multicast Address should embedded in the IPv6 Multicast Address.

Rule_IPv4_Type:

Defined whether an IPv4 Multicast Address Suffix or a full IPv4 Multicast Address is embedded in the IPv6 Multicast Address. Value 0 is default and means IPv4 Multicast Address Suffix is embedded in the IPv6 Multicast Address. Value 1 means a full IPv4 Multicast Address is embedded in the IPv6 Multicast Address.

3. Architecture

All of the scenarios that are describe in [RFC6144] can be easily illustrate using the diagram show in Figure 1 below:

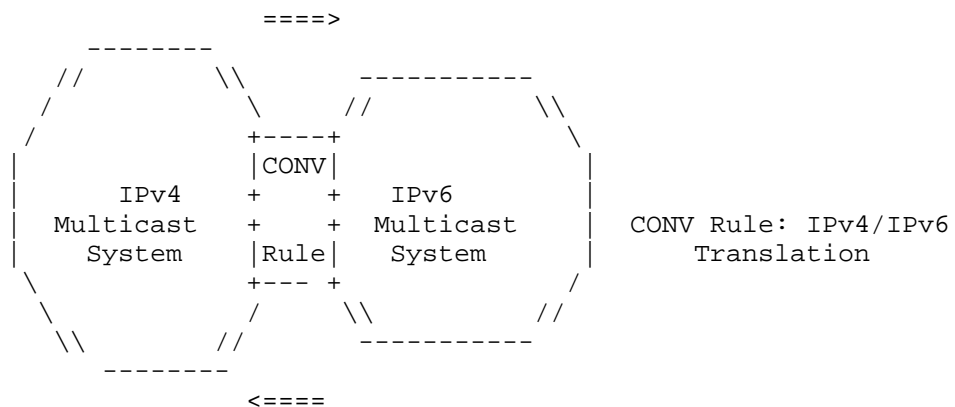


Figure 1: IPv4-IPv6 Address Conversion

As shown in this diagram(Fig.1), there is a conversion node between an IPv4 Multicast System and IPv6 Multicast System. Every conversion node must be provisioned with at least one rule defined in the document used for IPv4/IPv6 Multicast Address Conversion. There are also two arrows: an arrow from IPv4 Multicast system to IPv6 Multicast System means IPv4 Multicast system initiates the multicast flow. Another arrow from IPv6 Multicast system to IPv4 Multicast System means IPv6 Multicast system initiates the multicast flow. And

this also means that the algorithmic described in this document support both IPv4-initiated communication and IPv6-initiated communication.

4. IPv4/IPv6 Multicast Address Conversion

This section specifies the rule(s) for IPv4/IPv6 multicast address conversion.

4.1. Rule Design

Every CONV node must be provisioned with at least one rule. When there are several rules for IPv4/IPv6 Conversion assigned for a CONV node, this node should choose the rule which is longest match prefix for the destination IP address in multicast flow.

Each rule includes the following:

Rule_IPv6_M_Prefix (including prefix length)

Rule_IPv4_M_Prefix (including prefix length, optional)

Rule_IPv4_Offset (optional)

Rule_IPv4_Type (optional)

Rule_IPv6_M_Prefix/Length is according to section 2.7 of [ADDRARCH][RFC3513], or based on [RFC3306]. This parameter is mandatory.

Rule_IPv4_M_Prefix/Length is in IPv4 multicast group address scope. By default, this parameter is empty, which means match any IPv4 group address in the destination address field in the receiving packet. This parameter is optional.

Rule_IPv4_Offset defines the offset where IPv4 multicast address is embedded in the IPv6 multicast address. By default, the value is 96, which means embedded the IPv4 multicast address in the last 32 bits of the IPv6 multicast address. This parameter is optional.

Rule_IPv4_Type defines two kinds of IPv6 Multicast Address format: one format is IPv4 Multicast Address Suffix is embedded in the IPv6 Multicast Address, and corresponding Rule_IPv4_Type value is 0; another format is Full IPv4 Multicast Address is embedded in the IPv6 Multicast Address, and corresponding Rule_IPv4_Type value is 1. By default, Rule_IPv4_Type value is 0. This parameter is optional.

When Rule_IPv6_M_Prefix is SSM mode, the corresponding Rule_IPv4_M_Prefix in the same rule should be SSM mode. When Rule_IPv6_M_Prefix is ASM mode, the corresponding Rule_IPv4_M_Prefix in the same rule should be ASM mode.

If Rule_IPv6_M_Prefix is ASM mode but the corresponding Rule_IPv4_M_Prefix is SSM mode, the CONV node should process this rule as invalid. Also, if Rule_IPv6_M_Prefix is SSM mode but the corresponding Rule_IPv4_M_Prefix is ASM mode, the CONV node should process this rule as invalid.

4.2. IPv4 Multicast Address Suffix-embedded IPv6 Multicast Address

When Rule_IPv4_Type value is 0, the concentrated IPv6 Multicast Address format is as follow:

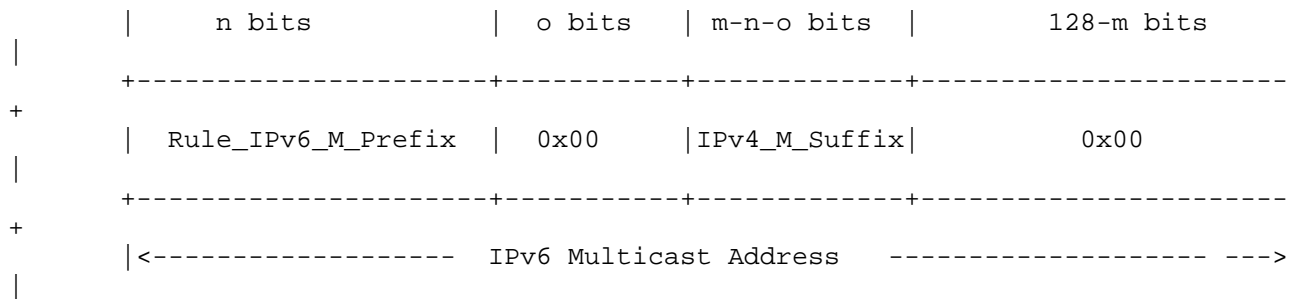


Figure 2: IPv6 Multicast Address Format for Rule_IPv4_Type=0

The IPv6 Multicast Address is created by combining the Rule_IPv6_M_Prefix and IPv4_M_Suffix and all zeros. Where the IPv4_M_Suffix is embedded is dependent with the Rule_IPv4_Offset(m). From the above format, with the Rule_IPv4_Offset(m), can induce the embedded position of the IPv4_M_Suffix. Then can concentrate the IPv6 Multicast Address as above. The IPv4_M_Suffix illustrates as follow:

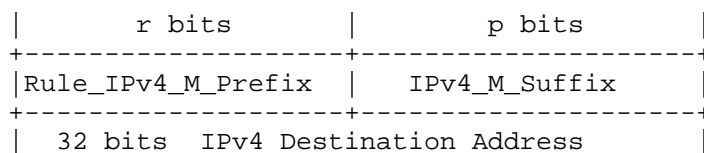


Figure 3

If Rule_IPv4_Offset value is 0, puts the IPv4_M_Suffix in the last (32-r) bits in the 128-bits IPv6 Multicast Address.

4.3. Full IPv4 Multicast Address-embedded IPv6 Multicast Address

When Rule_IPv4_Type value is 1, the concentrated IPv6 Multicast Address format is as follow:

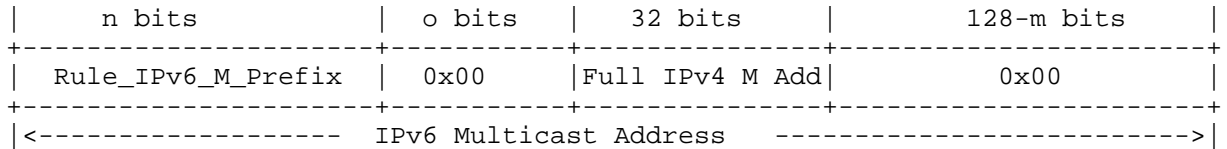


Figure 4: IPv6 Multicast Address Format for Rule_IPv4_Type=1

The IPv6 Multicast Address is created by combining the Rule_IPv6_M_Prefix and Full IPv4 Destination Address and all zeros. Where the Full IPv4 Destination Address is embedded is dependent with the Rule_IPv4_Offset(m). From the above format, with the Rule_IPv4_Offset(m), can induce the embedded position of the Full IPv4 Destination Address. Then can concentrate the IPv6 Multicast Address as above. The Full IPv4 Destination Address is the destination IPv4 address in the multicast flow.

5. Forwarding

5.1. From IPv4 Multicast System to IPv6 Multicast System

When a CONV node receives IPv4 multicast flow from IPv4 Multicast System, the CONV node should check whether there is a Rule_IPv4_M_Prefix longest match with the destination IPv4 multicast address. If there is no such rule which has a longest match prefix, the CONV node should drop these IPv4 multicast flow. If there is a rule which has a longest match prefix with the destination IPv4 multicast address, then do the IPv4-IPv6 conversion according to this rule. And then derive the IPv6 multicast address. The CONV node then checks the IPv6 multicast routing table, finds the outgoing interface and forwards the IPv6 multicast flow into the IPv6 Multicast System.

5.2. From IPv6 Multicast System to IPv6 Multicast System

When a CONV node receives IPv6 multicast flow from IPv6 Multicast System, the CONV node should check whether there is a Rule_IPv6_M_Prefix longest match with the destination IPv6 multicast address. If there is no such rule which has a longest match prefix, the CONV node should drop these IPv6 multicast flow. If there is a rule which has a longest match prefix with the destination IPv6 multicast address, then do the IPv4-IPv6 conversion according to this rule. If the Rule_IPv4_Type value is 0, then derives the IPv4_M_Suffix from the destination IPv6 address at the Rule_IPv4_Offset, concentrates the Rule_IPv4_M_Prefix with the

IPv4_M_Suffix as the destination IPv4 multicast address. If the Rule_IPv4_Type value is 1, then derives the destination IPv4 address from the destination IPv6 address at the Rule_IPv4_Offset. The CONV node then checks the IPv4 multicast routing table, finds the outgoing interface and forwards the IPv4 multicast flow into the IPv4 Multicast System.

6. Backwards compatibility

This solution is fully compatible with the multicast address format in the "draft-ietf-mboned-64-multicast-address-format".

7. Security Considerations

To be added later on as-needed basis.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, August 2002.
- [RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.

8.2. Informative References

- [I-D.ietf-mboned-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv6 Multicast Address With Embedded IPv4 Multicast Address", draft-ietf-mboned-64-multicast-address-format-05 (work in progress), April 2013.

Authors' Addresses

Yalin Cao
ZTE Corporation
No.68 Zijinghua Road, Yuhuatai District
Nanjing
China

Email: cao.yalin1@zte.com.cn

Cui Wang
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: wang.cuil@zte.com.cn

Wei Meng
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: meng.wei2@zte.com.cn, vally.meng@gmail.com

Bhumip Khasnabish
ZTE USA, Inc
55 Madison Avenue, Suite 160
Morristown, NJ 07960
USA

Email: bhumip.khasnabish@zteusa.com, vumipl@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 29, 2013

G. Chen
China Mobile
February 25, 2013

Analysis of NAT64 Port Allocation Method
draft-chen-sunset4-cgn-port-allocation-01

Abstract

The document enumerated methods of port assignment in CGN contexts, more focused on NAT64 environments. The analysis categorized the different methods with several key features. Corresponding to those features, the uses of existing protocols are also described. The potential concerns and workaround have been discussed. It's expected the document could provide a informative base line to help operators choosing a proper method.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Port Allocation Management	3
2.1. NAT vs NAPT	3
2.2. Dynamic vs Static	4
2.3. Centralized vs Distributed	5
3. Discussions	6
4. Security Considerations	6
5. IANA Considerations	7
6. References	7
6.1. Normative References	7
6.2. Informative References	7
Author's Address	8

1. Introduction

With the depletion of IPv4 address, CGN has been adopted by ISPs to expand IPv4 spaces. Relying upon the mechanism of multiplexing multiple subscribers' connections over a smaller number of shared IPv4 addresses, CGN mapped IP addresses from one address realm to another, providing transparent routing to end hosts.

[I-D.ietf-behave-lsn-requirements] defined the term of CGN. Several proposals including DS-Lite[RFC6333], NAT64[RFC6145], [RFC6146], NAT444 would likely fall into the scope. Focusing on the topic of IPv6 migration, the memo elaborate the considerations in NAT64 environment, where there IPv6-only nodes are connected.

[RFC6269] has provided a thoughtful analysis on the issues of IP sharing. It was point out that IP sharing may bring the impacts to law enforcement since the information of source address would be lost during the translation. Network administrators have to log the mapping status for each connection in order to identify a specific user associated with an IP address. It would post a challenge to operators, since it requires additional storage resource and data inspection process for indentifying the real users. It's desirable to compact the logging information by a rational port allocation. Those allocation policies should consider the tradeoff between port utilization and log storage compression. The document is trying to enumerate the several dimensions for assigning the port information. It's expected administrator could use those factors to determine their own properties.

2. Port Allocation Management

This section lists several factors to allocate the port information in NAT64 equipments. It's likely that each allocation model would have an exemplified case. The relevant issues and potential workarounds have also been described for each aspect.

2.1. NAT vs NAPT

NAT64 may not do Network Address Port Translation (NAPT), but only Network Address Translation (NAT). In those cases, there is no concern about port assignment. Those translation methods would relieve the demands of log information storage, since NAT does not have to administer address management with session flows. Furthermore, there is no requirement to maintain log when NAT64 performing stateless translations. Some existing practices are listed below from two aspects.

- o Stateful NAT

The stateful NAT can be implemented either by static address translation or dynamic address translation.

In the case of static address assignment, one-to-one address mapping for hosts between a IPv6 network address and an IPv4 network address would be pre-configured on the NAT operation. Those cases normally occurred when a server deployed in a IPv6 domain. The static configuration ensure the stable inbound connectivity. The static method is also easier for Lawful interception system to derive the mapped address, since the mapping didn't change with time.

Dynamic address assignment would periodically free the binding so that the global address could be recycled for later uses. Addresses could be more efficiently used by time-division manner. It only requires systems maintaining mappings for per-customer, other than per-session flow. This method is usually adopted to reduce the log burden in some protocols.

- o Stateless NAT

The stateless NAT is performed in compliant with [RFC6145]. Public IPv4 address is required to be inserted in IPv6 address. Therefore, NAT64 could directly extract the address and no need to record mapping states. The lawful interception could likely identify the IPv4 address through received IPv6 address. It's a protocol to eliminate the log information storage. There are two potential concerns for those technologies. First off, the static one-to-one mapping may didn't address the issue of IPv4 depletion. Secondly, it introduced the dependency of IPv4/IPv6. That would create new limitations since the change of IPv4 address would cause renumbering of IPv6 addresses. Whereas, that is useful for the IDC migration where there is IPv6 servers pools to receive inbound connections from IPv4 users externally[I-D.anderson-siit-dc].

2.2. Dynamic vs Static

When the case comes to port assignment, there are two methods for port allocations.

- o Dynamic assignment

NAT64 normally do the dynamic assignment. In respect to the received connections, ports can be allocated to each sessions. NAT64 would do the dynamic approach by default, since it achieves maximum port utilization. One downside for this approach is the gateway has to record log information for each session. That would potentially

increase the log volume. There is a statistic from field trials that the average number of connections per customer per day at approximately 10,000 connections. If log system is required to store information for 180 days, the testing shown that the amount of data records would achieve 20T.

- o Static assignment

The static assignment make a bulk of port reservation for a specific address. The bulk of port could be either a contiguous or non-contiguous port range for sake of attacks defense. [I-D.donley-behave-deterministic-cgn] has described a deterministic NAT to reserve a port range for each specific IP address. That is a significant improvement for lightening log volume. However, a trade-off should be made when administrator has to consider the port utilization. For the administrator who prioritize the port utilization, dynamic assignment maybe a suitable solution for them. Another consideration is using Address-Dependent Mapping or Address and Port-Dependent Mapping[RFC4787] to increase the port utilization. This feature has already been implemented as vendor-specific features. Whereas, it should be noted that REQ-7, REQ-12 in [I-D.ietf-behave-lsn-requirements] may reduce the incentives.

2.3. Centralized vs Distributed

There are increasing needs to connect NAT64 with downstream NAT46-capable CE devices to support IPv4 hosts/applications in a IPv6-only access. Several solutions have been proposed in this area, e.g. 464xlat[I-D.ietf-v6ops-464xlat], MAP-T[I-D.ietf-softwire-map-t] and 4rd[I-D.ietf-softwire-4rd]. With the feature of double-translation, the port allocation can be managed as a centralized way on NAT64 or distributed to downstream devices(e.g, CPE connected with NAT64) .

- o Centralized Assignment

A centralized method would make port assignments when traffic come to NAT64. The allocation policy is enforced on a centralized gateway. Either a dynamic or static port assignment is made for received sessions.

- o Distributed Assignment

NAT64 could also delegate the pre-allocated port range to customer edge devices. That can be achieved through additional out-band provisioning signals(e.g.[I-D.ietf-pcp-base], [I-D.tsou-pcp-natcoord][I-D.ietf-softwire-map-dhcp]). The distributed model normally performed A+P style for static port assignment. NAT64 should hold the corresponding mapping in

accordance with assigned ports. Those methods could shift NAT64 port computation/states into downstream devices. The detailed benefits was documented in [I-D.ietf-softwire-stateless-4v6-motivation].

3. Discussions

With demands of reducing log volume, there are several approaches of port assignment described in the aforementioned sections. It could be found that a trade-off between maximum port utilization and log volume always exist to justify the use of different solutions. In respect to difference of port assignment, the granularity of log could be ranked as per-session, per-port-bulk, per-customer and None. With the reduction of log volume, port utilization ratio is likely decreased. Therefore, the decision should be made if there is a quantitative statistic to evaluate what is gain from reducing log volume and loss from decreasing port utilization. Those data analysis is planned to be added after further lab testing. Operators could choose the proper method considering following:

- o Average connectivities per customer per day
- o Peak connectivities per day
- o The amount of public IPv4 address in NAT64
- o Application demands for specific ports
- o The parallel processing capabilities of NAT64
- o The tolerance of Log volume

Apart from above, the port allocation can be tuned corresponding to the phase of IPv6 migration. The use of NAT64 would advance IPv6, because it provides everyone incentives to use IPv6, and eventually the result is an end-to-end IPv6-only networks with no needs for IPv4. As more content providers and service are available over IPv6, the utilization on NAT64 goes down since fewer destinations require translation progressing. In the trend of decreased IPv4 connections, NAT64 could relax the multiplexing ratio of shared IPv4 address by either a delivered message or a centralized control. A load for log system can also be relieved due to simplified mapping states.

4. Security Considerations

TBD

5. IANA Considerations

This document makes no request of IANA.

6. References

6.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.
- [I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Bao, C., Dec, W., Yeh, L., and X. Deng, "DHCPv6 Options for Mapping of Address and Port", draft-ietf-softwire-map-dhcp-02 (work in progress), February 2013.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

6.2. Informative References

- [I-D.anderson-siit-dc]
Anderson, T., "Stateless IP/ICMP Translation in IPv6 Data Centre Environments", draft-anderson-siit-dc-00 (work in progress), November 2012.
- [I-D.donley-behave-deterministic-cgn]
Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier Grade NAT Deployments", draft-donley-behave-deterministic-cgn-05 (work in progress), January 2013.

- [I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A.,
and H. Ashida, "Common requirements for Carrier Grade NATs
(CGNs)", draft-ietf-behave-lsn-requirements-10 (work in
progress), December 2012.
- [I-D.ietf-softwire-4rd]
Jiang, S., Despres, R., Penno, R., Lee, Y., Chen, G., and
M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless
Solution (4rd)", draft-ietf-softwire-4rd-04 (work in
progress), October 2012.
- [I-D.ietf-softwire-map-t]
Li, X., Bao, C., Dec, W., Troan, O., Matsushima, S., and
T. Murakami, "Mapping of Address and Port using
Translation (MAP-T)", draft-ietf-softwire-map-t-01 (work
in progress), February 2013.
- [I-D.ietf-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O.,
Borges, I., and G. Chen, "Motivations for Carrier-side
Stateless IPv4 over IPv6 Migration Solutions",
draft-ietf-softwire-stateless-4v6-motivation-05 (work in
progress), November 2012.
- [I-D.ietf-v6ops-464xlat]
Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT:
Combination of Stateful and Stateless Translation",
draft-ietf-v6ops-464xlat-10 (work in progress),
February 2013.
- [I-D.tsou-pcp-natcoord]
Sun, Q., Boucadair, M., Deng, X., Zhou, C., Tsou, T., and
S. Perreault, "Using PCP To Coordinate Between the CGN and
Home Gateway", draft-tsou-pcp-natcoord-09 (work in
progress), November 2012.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P.
Roberts, "Issues with IP Address Sharing", RFC 6269,
June 2011.

Author's Address

Gang Chen
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China

Email: phdgang@gmail.com

INTERNET-DRAFT
Intended Status: Informational
Expires: April 18, 2013

R.Hiromi
Intec, Inc.
Hazeyama
NAIST
A.Onoe
Sony Corporation
O.Nakamura
Keio University
October 15, 2012

A workaround for termination of IPv4 network services
draft-hiromi-sunset4-termination-ipv4-00

Abstract

After sun-setting of IPv4, many devices are connected to IPv6 single stack network. In this document we describe a workaround of IPv6 enabled network configuration. At this moment, the condition of IPv6 adoption on the consumer devices such as PC, tablet, mobile terminal and entertainment devices are various. For example, some devices are fully support IPv6 client but some are not. It is very hard to provide IPv6 network service with these various conditioned devices. To solve this problem, we tried to verify some configurations to connect these devices into IPv6 enabled consumer network for termination of IPv4 network services.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
1.1	Terminology	3
1.2	Test Network	3
2.	Problems Statement	4
2.1	Device classification	4
2.2	Observation on personal device	6
3	Workaround	7
3.1	trial and result	7
Experiment #1:	7
Experiment #2:	7
Experiment #3:	7
3.2	remaining issue	8
4	Security Considerations	8
5	IANA Considerations	8
6	References	8
6.1	Normative References	8
6.2	Informative References	9
7	Acknowledgement	9
Authors' Addresses	9

1 Introduction

After sun-setting of IPv4, many devices are connected to IPv6 only network. In this document we describe a workaround of IPv6 enabled network configuration. At this moment, the condition of IPv6 adoption on the consumer devices such as PC, tablet, smart phones and entertainment devices are various. For example, some devices are fully support DHCPv6 client function but some are not. In IETF, additional function for connecting clients are still discussed. Implementation of client function will be changing for a while. It must be very hard to provide IPv6 network service with these various conditioned devices. To solve this issue, we tried to verify some configurations to connect these devices into IPv6 enabled consumer network.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2 Test Network

In this document we call "IPv6-only network" as a user network which connects IPv6 clients and carries IPv6 packets. We made an IPv6-only test network as following configuration. We brought this IPv6-only network to seasonal WIDE Camp from 2011 to 2012. WIDE Camp was hold 3 times. Through DNS64/NAT64 translation, the clients can access IPv4 Internet services and with this technique the users can turn off IPv4 at the edge network. We can observe simply IPv6 client behavior and solve the specified points.

Here is the basic configuration;

```
User-Access:   WiFi(IEEE802.11a,b,g,n)
IPv6 Address Configuration: SLAAC(address prefix,router),DHCPv6(DNS)
ISP(IPv6):     DHCP-PD or static setting of prefix
IPv4 Internet: using coexistence
mechanism(4rd,464XLAT,SA46T,DNS64/NAT64) over IPv6, base technology
is DNS64/NAT64
DNS64/NAT64:   Map all IPv4 on Internet into IPv6
RA:           Enable Other Config (for DHCP6)
DHCPv6:       Distribute DNS64 address
DHCP4:        No DHCPv4 running!
```

```
+-----+
| IPv6 router |
```

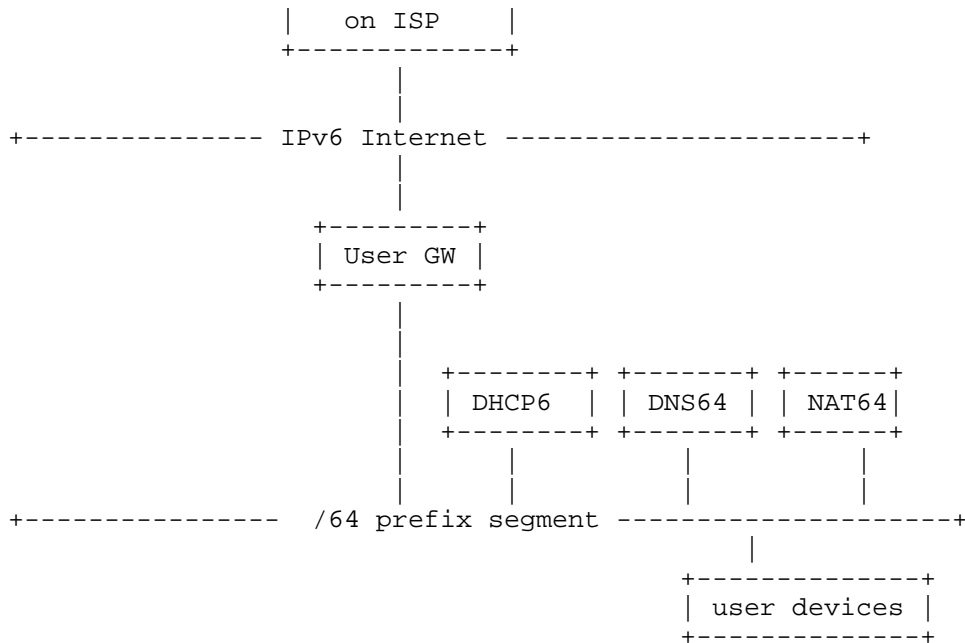


Fig.1 Basic Network Topology

2. Problems Statement

"IPv6 support" on consumer devices are inconsistent in implementation. Especially some devices are designed with IPv6 limitation if no IPv4 information provided on the device. The IPv6 network services has to absorb these differences in some way.

2.1 Device classification

Over 300 devices were brought into every WIDE Camp. We classified their Device Type, OS, OS version. We determined deference of DHCP6 client behavior and possible precondition per OS. Table 1 and 2 shows the result of them. Popular OS on PC was both Windows and MacOS X series. Various versions were observed. Popular OS on Mobile Phone was both Android and iOS. Feature phone were also popular but there were no WiFi function on such feature phone so that they were out of our target. In Table 1, we listed up auto-address configuration function on each OS. The last column means the failure occurred case with checked mark if they got IPv4 local address within 169.254.0.0/16. In Table 2, we picked up the OS which has no IPv6 friendly User Interface.

OS	Version	RA	DHCPv6	169.254/16
Windows	XP	o	x	
	Vista	o	o	
	7	o	o	
	8	o	o	
MacOS X	10.6	o	x	
	10.7	o	o	
	10.8	o	o	
Android	1.6	x	x	
	2.3.4	o	x	x
	2.3.5	o	x	x
	2.3.6	o	x	x
	4.0.3	o	x	x
	4.0.4	o	x	x
	4.1	o	x	x
iOS	4.3.2 JB	o	x	x
	5	o	x	x
iPad iOS	5	o	o	
	6	o	o	
kindle	3.1	x	x	
NetBSD	5.1	o	x	
	6.99.4	o	x	
Ubuntsu	12.0.4	o	o	

Table 1. Result of the client behavior per OS

OS	Version	display	configuration
Android	2.3.4	x	x
	2.3.6	x	x
iOS	5	x	x
iPad iOS	5	x	x

Table 2. List of OS without IPv6 support on User Interface

2.2 Observation on personal device

5 problematic behaviors are observed.

- (a). failed to set name server(v6) information(WinXP, MacOS SL, Android)
- (b). failed to input name server(v6) by manual configuration(Android)
- (c). failed to resolve resource record under (a) or (c) condition(WinXP, MacOS SL, Android)
- (d). "network setting" won't be completed before getting set of IPv4 information(DNS,router,IPv4 Address)(iOS5,Android)
- (e). waiting for IPv4 connection timeout(MacOS)

The causes of problems on consumer devices are sorted by 3 types. First one is IPv6 implementation issue, which we listed on Table 1. The other issue is User Interface issue, which we listed on Table 2. The last one is coming from other layer's function, typical issue is a WiFi controller which provided by vendor(Lenovo Access Connections) and turn off IPv4 with the controller setting then the client was able to connect to IPv6-only network.

3 Workaround

In this document, we focused on the problem which occurred by IPv6 implementation on the clients.

How do we solve the problematic behavior? It might be a distant idea that waiting for all devices fully support IPv6. We considered to put additional configuration parameters to comply with improvements.

3.1 trial and result

To solve (a)to(e) in Section 2.2, we reconsidered network setting and putting into testbed network step by step.

- (1) DNS64/NAT64 Map all IPv4 on Internet into IPv6
- (2) DHCPv6 for DNS(IPv6) configuration
- (3) DHCPv4 for IPv4 private address configuration
- (4) DHCPv4 for DNS(IPv4) and Default Router
(actual packet transfer is prohibited on this router)
- (5) DNS(IPv4) is located local segment
- (6) DNS(IPv4, IPv6) set 'A' filter
- (7) DNS(IPv4, IPv6) always returns 'NODATA' with 'A' query
(Over both IPv4 and IPv6 transport)
- (8) AAAA queries forward to DNS64 server

Experiment #1:

With "IPv4 private address assignment via DHCP4 without Default Router nor DNS", no issues were solved.

- Timeout problem exist on MacOSX.
- iOS applications are sometimes working, but periodically fails due to retrying Wi-Fi connection.

Experiment #2:

Put BIND9 forwarder on-link and configure DHCP4/6 to use this DNS. Configure BIND9 forwarder with: deny-answer-addresses { 0.0.0.0/0; }; Which direct no IPv4 address answer should be trusted. It returns SERVFAIL to resolver.

- Android is now working: Browser, Twitter, Facebook are OK.
- iOS is working: but periodically fails due to retrying.
- MacOS is working: but still encountered fallback timeout.
- Windows is NOT WORKING: all DNS queries failed due to SERVFAIL.

Experiment #3:

Hack AAAA filtering code on BIND9 to filter 'A' instead of 'AAAA' both on IPv4/IPv6 transport. Put BIND9 above to local link, which is configured to forward all queries to DNS64. Configure DHCP4/DHCP6 to use the DNS proxy.

- Windows, MacOS X, iOS, Android is now working.

- Some of applications still failed on IPv6 only, but many are OK: IE/Safari/Chrome/Firefox, Twitter, Facebook, Instagram, APNS, ...

After bringing all new additional network configuration, most of all clients were able to connect IPv6-only network with zero-configuration on the client side.

This is the technique for IPv6-only network but also can be useful for terminating IPv4 network environment at the user network.

3.2 remaining issue

"Waiting for IPv4 connection timeout on MacOS" is still issued. A possible reason for this connection failure during 1-2 minutes after WiFi connection established is timing of sending RS(Router Solicitation). RS is sent from kernel before Wi-Fi link is established. No IPv6 address is obtained until periodical RA(Router Advertisement) is received.

Workaround for this is considered as follows but we were not unable to examine it on the camp at this time.

- shorten RA interval to 5-10 seconds (though it disturb Wi-Fi...)
- Detect association through AP log and kick RS or RA.

4 Security Considerations

Possible security threats are same as what pointed out in original protocols and technologies.

5 IANA Considerations

This document has no IANA implications.

6 References

6.1 Normative References

- [NAT64] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [DNS64] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

6.2 Informative References

TBD

7 Acknowledgement

Here, we thank to all the participants of WIDE camp(s) on the experiments. We also say thank you to whom serving implementations and services in the Matsuhira Royal Hotel.

Y. Ueno of Keio Univ. for IPv6 L2TP implementation
NTT EAST and IIJ for the commercial IPv6 service

R. Nakamura of Univ. of Tokyo, Y. Ueno of Keio Univ. and R. Shouhara of Univ. of Tokyo for helping us on the base settings of the IPv6 only experiments and merging into the camp-net.

T. Jimei of Internet Systems Consortium for his quick hack on A filter of Bind 9.

T. Ishihara of Univ. of Tokyo for his DNS operating advisory

Y. Atarashi of Alaxala Networks and R. Atarashi of IIJ Innovation Institute for designing the items of face to face interview and analyzing user survey data.

Authors' Addresses

R.Hiromi
INTEC Inc.
1-3-3, Shinsuna,
Koto-ku,
Tokyo, Japan
EMail: hiromi@inetcore.com

Hiroaki Hazeyama
NAIST
Takayama 8916-5
Nara, Japan
Phone: +81 743 72 5216
Email: hiroa-ha@is.naist.jp

Atsushi Onoe
SONY Corporation
EMail: onoe@wide.ad.jp

INTERNET DRAFT<A workaround for termination of IPv4 network services>October 15,
2012

Osamu Nakamura
WIDE Project
5322 Endo
Kanagawa, Japan
Email: osamu@wide.ad.jp

i»¿

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 28, 2018

W. Liu
W. Xu
C. Zhou
Huawei Technologies
T. Tsou
Philips Lighting
S. Perreault
Jive Communications
P. Fan

R. Gu
China Mobile
C. Xie
China Telecom
Y. Cheng
China Unicom
July 29, 2017

Gap Analysis for IPv4 Sunset
draft-ietf-sunset4-gapanalysis-09

Abstract

Sunsetting IPv4 refers to the process of turning off IPv4 definitively. It can be seen as the final phase of the transition to IPv6. This memo enumerates difficulties arising when sunseting IPv4, and identifies the gaps requiring additional work.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 28, 2018.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Related Work	3
3. Remotely Disabling IPv4	4
3.1. Indicating that IPv4 connectivity is unavailable	4
3.2. Disabling IPv4 in the LAN	4
4. Client Connection Establishment Behavior	5
5. Disabling IPv4 in Operating System and Applications	5
6. On-Demand Provisioning of IPv4 Addresses	6
7. IPv4 Address Literals	6
8. Managing Router Identifiers	7
9. IANA Considerations	7
10. Security Considerations	7
11. Acknowledgements	7
12. Informative References	7
Annex A. Solution Ideas	9
A.1. Remotely Disabling IPv4	9
A.1.1. Indicating that IPv4 connectivity is unavailable	9
A.1.2. Disabling IPv4 in the LAN	9
A.2. Client Connection Establishment Behavior	10
A.3. Disabling IPv4 in Operating System and Applications	10
A.4. On-Demand Provisioning of IPv4 Address.	10
A.5. Managing Router Identifiers	10
Authors' Addresses	11

1. Introduction

The final phase of the transition to IPv6 is the sunset of IPv4, that is turning off IPv4 definitively on the attached networks and on the upstream networks.

Some current implementation behavior makes it hard to sunset IPv4. Additionally, some new features could be added to IPv4 to make its sunsetting easier. This document analyzes the current situation and proposes new work in this area.

The decision about when to turn off IPv4 is out of scope. This document merely attempts to enumerate the issues one might encounter if that decision is made.

2. Related Work

[RFC3789], [RFC3790],[RFC3791], [RFC3792], [RFC3793], [RFC3794], [RFC3795] and [RFC3796] contain surveys of IETF protocols with their IPv4 dependencies.

Additionally, although reviews in RFCs 3789-3796 ensured that IETF standards then in use could support IPv6, no IETF-wide effort has been undertaken to ensure that the issues identified in those drafts are all addressed, nor to ensure that standards written after RFC3100 (where the previous review efforts stopped) function properly on IPv6-only networks.

The IETF needs to ensure that existing standards and protocols have been actively reviewed, and any parity gaps either identified so that they can be fixed, or documented as unnecessary to address because it is unused or superseded by other features.

First, the IETF must review RFCs 3789-3796 to ensure that any gaps in specifications identified in these documents and still in active use have been updated as necessary to enable operation in IPv6-only environments (or if no longer in use, are declared historic).

Second, the IETF must review documents written after the existing review stopped (according to RFC 3790, this review stopped with approximately RFC 3100) to identify specifications where IPv6-only operation is not possible, and update them as necessary and appropriate, or document why an identified gap is not an issue i.e. not necessary for functional parity with IPv4.

This document does not recommend excluding Informational and BCP RFCs as the previous effort did, due to changes in the way that these documents are used and their relative importance in the RFC Series. Instead, any documents that are still active (i.e. not declared historic or obsolete) and the product of IETF consensus (i.e. not a product of the ISE Series) should be included. In addition, the reviews undertaken by RFCs 3789-3796 were looking for "IPv4 dependency" or "usage of IPv4 addresses in standards". This document recommends a slightly more specific set of criteria for review. Reviews should include:

- o Consideration of whether the specification can operate in an environment without IPv4.
- o Guidance on the use of 32-bit identifiers that are commonly populated by IPv4 addresses.

- o Consideration of protocols on which specifications depend or interact, to identify indirect dependencies on IPv4.
- o Consideration of how to transit from an IPv4 environment to an IPv6 environment.

3. Remotely Disabling IPv4

3.1. Indicating that IPv4 connectivity is unavailable

- PROBLEM 1: When an IPv4 node boots and requests an IPv4 address (e.g., using DHCP), it typically interprets the absence of a response as a failure condition even when it is not.
- PROBLEM 2: Home router devices often identify themselves as default routers in DHCP responses that they send to requests coming from the LAN, even in the absence of IPv4 connectivity on the WAN.

3.2. Disabling IPv4 in the LAN

- PROBLEM 3: IPv4-enabled hosts inside an IPv6-only LAN can auto-configure IPv4 addresses [RFC3927] and enable various protocols over IPv4 such as mDNS [RFC6762] and LLNMR [RFC4795]. This can be undesirable for operational or security reasons, since in the absence of IPv4, no monitoring or logging of IPv4 will be in place.
- PROBLEM 4: IPv4 can be completely disabled on a link by filtering it on the L2 switching device. However, this may not be possible in all cases or may be too complex to deploy. For example, an ISP is often not able to control the L2 switching device in the subscriber home network.
- PROBLEM 5: A host with only Link-Local IPv4 addresses will "ARP for everything", as described in Section 2.6.2 of [RFC3927]. Applications running on such a host connected to an IPv6-only network will believe that IPv4 connectivity is available, resulting in various bad or sub-optimal behavior patterns. See [I-D.yourtchenko-ipv6-disable-ipv4-proxyarp] for further analysis.

Some of these problems were described in [RFC2563], which standardized a DHCP option to disable IPv4 address auto-configuration. However, using this option requires running an IPv4 DHCP server, which is contrary to the goal of IPv4 sunsetting.

4. Client Connection Establishment Behavior

PROBLEM 6: Happy Eyeballs [RFC6555] refers to multiple approaches to dual-stack client implementations that try to reduce connection setup delays by trying both IPv4 and IPv6 paths simultaneously. Some implementations introduce delays which provide an advantage to IPv6, while others do not [Huston2012]. The latter will pick the fastest path, no matter whether it is over IPv4 or IPv6, directing more traffic over IPv4 than the other kind of implementations. This can prove problematic in the context of IPv4 sunsetting, especially for Carrier-Grade NAT phasing out because CGN does not add significant latency that would make the IPv6 path more preferable. Traffic will therefore continue using the CGN path unless other network conditions change.

PROBLEM 7: `getaddrinfo()` [RFC3493] sends DNS queries for both A and AAAA records regardless of the state of IPv4 or IPv6 availability. The `AI_ADDRCONFIG` flag can be used to change this behavior, but it relies on programmers using the `getaddrinfo()` function to always pass this flag to the function. The current situation is that in an IPv6-only environment, many useless A queries are made.

5. Disabling IPv4 in Operating System and Applications

It is possible to completely remove IPv4 support from an operating system as has been shown by the work of Bjoern Zeeb on FreeBSD. [Zeeb] Removing IPv4 support in the kernel revealed many IPv4 dependencies in libraries and applications.

PROBLEM 8: Completely disabling IPv4 at runtime often reveals implementation bugs. Hard-coded dependencies on IPv4 abound, such as on the 127.0.0.1 address assigned to the loopback interface, and legacy IPv4-only APIs are widely used by applications. It is hard for the administrators and users to know what applications running on the operating system have implementation problems of IPv4 dependency. It is therefore often operationally impossible to completely disable IPv4 on individual nodes.

PROBLEM 9: In an IPv6-only world, legacy IPv4 code in operating systems and applications incurs a maintenance overhead and can present security risks.

6. On-Demand Provisioning of IPv4 Addresses

As IPv6 usage climbs, the usefulness of IPv4 addresses to subscribers will become smaller. This could be exploited by an ISP to save IPv4 addresses by provisioning them on-demand to subscribers and reclaiming them when they are no longer used. This idea is described in [I-D.fleischhauer-ipv4-addr-saving] and [BBF.TR242] for the context of PPP sessions. In these scenarios, the home router is responsible for requesting and releasing IPv4 addresses, based on snooping the traffic generated by the hosts in the LAN, which are still dual-stack and unaware that their traffic is being snooped.

As described in TR-092 and TR-187, NAS(e.g., BRAS, BNG) stores pools of IPv4 and IPv6 addresses, which are used for DHCP distribution to the hosts in home network. IPv4 and IPv6 addresses of hosts can be dynamic assignment from a pool of IPv4 and IPv6 prefixes in NAS.

As the IPv4 sunsets, the number of IPv4 hosts is reduced, therefore the IPv4 address resource in NAS needs to be reduced too. These reduced IPv4 addresses will be reclaimed by the address management system (NMS, controller, IPAM, etc.). At the same time, as the number of IPv6 hosts increases, NAS need incrementally increase the number of IPv6 address resource. The increased IPv6 address resource can be assigned by the address management system, which makes the transition more smoothly by dynamically adding / releasing IP address resources in NAS. In modern network systems, protocols such as NETCONF / RESTCONF / RADIUS can be used for this process. With NETCONF, NAS acts as NETCONF server with the opening port to listen for the client connection, while the address management system as a netconf client that connects and processes IP address request from NAS.

PROBLEM 10: Dual-stack hosts that implement Happy-Eyeballs [RFC6555] will generate both IPv4 and IPv6 traffic even if the algorithm end up choosing IPv6. This means that an IPv4 address will always be requested by the home router, which defeats the purpose of on-demand provisioning.

PROBLEM 11: Many operating systems periodically perform some kind of network connectivity check as long as an interface is up. Similarly, applications often send keep-alive traffic continuously. This permanent "background noise" will prevent an IPv4 address from being released by the home router.

PROBLEM 12: Hosts in the LAN have no knowledge that IPv4 is available to them on-demand only. If they had explicit knowledge of this fact, they could tune their behaviour so as to be more conservative in their use of IPv4.

PROBLEM 13: This mechanism is only being proposed for PPP even though it could apply to other provisioning protocols (e.g., DHCP).

PROBLEM 14: When the number of IPv4 hosts connected to NAS is reduced, the NAS releases the IPv4 address resource and the NAS requests more IPv6 address resource for it to serve hosts transitting from IPv4 to IPv6.

7. IPv4 Address Literals

IPv4 addresses are often used as resource locators. For example, it is common to encounter URLs containing IPv4 address literals on web

sites [I-D.wing-behave-http-ip-address-literals]. IPv4 address literals may be published on media other than web sites, and may appear in various forms other than URLs. For the operating systems which exhibit the behavior described in [I-D.yourtchenko-ipv6-disable-ipv4-proxyarp], this also means an increase in the broadcast ARP traffic, which may be undesirable.

PROBLEM 15: IPv6-only hosts are unable to access resources identified by IPv4 address literals.

8. Managing Router Identifiers

IPv4 addresses are often conventionally chosen to number a router ID, which is used to identify a system running a specific protocol. The common practice of tying an ID to an IPv4 address gives much operational convenience. A human-readable ID is easy for network operators to deal with, and it can be auto-configured, saving the work of planning and assignment. It is also helpful to quickly perform diagnosis and troubleshooting, and easy to identify the availability and location of the identified router.

PROBLEM 16: In an IPv6 only network, there is no IP address that can be directly used to number a router ID. IDs have to be planned individually to meet the uniqueness requirement. Tying the ID directly to an IP address which yields human-friendly, auto-configured ID that helps with troubleshooting is not possible.

9. IANA Considerations

None.

10. Security Considerations

It is believed that none of the problems identified in this draft are security issues.

11. Acknowledgements

Thanks in particular to Andrew Yourtchenko, Jordi Palet Martinez, Lee Howard, Nejc Skoberne, and Wes George for their thorough reviews and comments.

Special thanks to Marc Blanchet who was the driving force behind this work and to Jean-Philippe Dionne who helped with the initial version of this document.

12. Informative References

[BBF.TR242]

Broadband Forum, "TR-242: IPv6 Transition Mechanisms for Broadband Networks", August 2012.

[Huston2012]

Huston, G. and G. Michaelson, "RIPE 64: Analysing Dual Stack Behaviour and IPv6 Quality", April 2012.

- [I-D.fleischhauer-ipv4-addr-saving]
Fleischhauer, K. and O. Bonness, "On demand IPv4 address provisioning in Dual-Stack PPP deployment scenarios", draft-fleischhauer-ipv4-addr-saving-05 (work in progress), September 2013.
- [I-D.wing-behave-http-ip-address-literals]
Wing, D., "Coping with IP Address Literals in HTTP URIs with IPv6/IPv4 Translators", draft-wing-behave-http-ip-address-literals-02 (work in progress), March 2010.
- [I-D.yourtchenko-ipv6-disable-ipv4-proxyarp]
Yourtchenko, A. and O. Owen, "Disable "Proxy ARP for Everything" on IPv4 link-local in the presence of IPv6 global address", draft-yourtchenko-ipv6-disable-ipv4-proxyarp-00 (work in progress), May 2013.
- [RFC2563] Troll, R., "DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients", RFC 2563, May 1999.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.
- [RFC3789] Nesser, P. and A. Bergstrom, "Introduction to the Survey of IPv4 Addresses in Currently Deployed IETF Standards Track and Experimental Documents", RFC 3789, June 2004.
- [RFC3790] Mickles, C. and P. Nesser, "Survey of IPv4 Addresses in Currently Deployed IETF Internet Area Standards Track and Experimental Documents", RFC 3790, June 2004.
- [RFC3791] Olvera, C. and P. Nesser, "Survey of IPv4 Addresses in Currently Deployed IETF Routing Area Standards Track and Experimental Documents", RFC 3791, June 2004.
- [RFC3792] Nesser, P. and A. Bergstrom, "Survey of IPv4 Addresses in Currently Deployed IETF Security Area Standards Track and Experimental Documents", RFC 3792, June 2004.
- [RFC3793] Nesser, P. and A. Bergstrom, "Survey of IPv4 Addresses in Currently Deployed IETF Sub-IP Area Standards Track and Experimental Documents", RFC 3793, June 2004.
- [RFC3794] Nesser, P. and A. Bergstrom, "Survey of IPv4 Addresses in Currently Deployed IETF Transport Area Standards Track and Experimental Documents", RFC 3794, June 2004.

- [RFC3795] Sofia, R. and P. Nesser, "Survey of IPv4 Addresses in Currently Deployed IETF Application Area Standards Track and Experimental Documents", RFC 3795, June 2004.
- [RFC3796] Nesser, P. and A. Bergstrom, "Survey of IPv4 Addresses in Currently Deployed IETF Operations & Management Area Standards Track and Experimental Documents", RFC 3796, June 2004.
- [RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.
- [RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, January 2007.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [Zeeb] "FreeBSD Snapshots without IPv4 support", <<http://wiki.freebsd.org/IPv6Only>>.

Annex A. Solution Ideas

A.1. Remotely Disabling IPv4

A.1.1. Indicating that IPv4 connectivity is unavailable

One way to address these issues is to send a signal to a dual-stack node that IPv4 connectivity is unavailable. Given that IPv4 shall be off, the message must be delivered through IPv6.

A.1.2. Disabling IPv4 in the LAN

One way to address these issues is to send a signal to a dual-stack node that auto-configuration of IPv4 addresses is undesirable, or that direct IPv4 communication between nodes on the same link should not take place.

A signalling protocol equivalent to the one from [RFC2563] but over IPv6 is necessary, using either Router Advertisements or DHCPv6.

Furthermore, it could be useful to have L2 switches snoop this signalling and automatically start filtering IPv4 traffic as a consequence.

Finally, it could be useful to publish guidelines on how to safely block IPv4 on an L2 switch.

A.2. Client Connection Establishment Behavior

Recommendations on client connection establishment behavior that would facilitate IPv4 sunsetting would be appropriate.

Happy Eyeballs timers and related parameters should get gradually increased, so even if IPv6 is "slower" than IPv4, IPv6 gains preference anyway.

A.3. Disabling IPv4 in Operating System and Applications

It would be useful for the IETF to provide guidelines to programmers on how to avoid creating dependencies on IPv4, how to discover existing dependencies, and how to eliminate them. It would be useful if operating systems provide functions for users to see what applications uses legacy IPv4-only APIs, so they can know it better whether they can turn off IPv4 completely. Having programs and operating systems that behave well in an IPv6-only environment is a prerequisite for IPv4 sunsetting.

A.4. On-Demand Provisioning of IPv4 Address

As the sunset of IPv4 in NAS, parts of hosts no longer need IPv4 address. IPv4 address resources in NAS appears surplus, NAS should obtain the unoccupied IPv4 address, generate a request and send it to the address management system to release those IPv4 address resource. Meanwhile, NAS needs more IPv6 address resources for the host transiting from IPv4 to IPv6. NAS judges whether the usage status of the IPv6 address resource satisfies certain condition, and the condition can be IPv6 address utilization ratio. If the IPv6 address utilization ratio is too high, the NAS generates a resource request containing IPv6 addresses information that needs to be applied and sends it to the address management system. When the address management system receives the IPv6 address resource request, it allocates IPv6 address pool from its assignable IPv6 address resource according to the information of the resource request, then it sends a response message with the information of allocated IPv6 address pool for this NAS to the NAS. Then the NAS receives the response and gets the information of allocated IPv6 address pool.

A.5. Managing Router Identifiers

Router IDs can be manually planned, possibly with some hierarchy or design rule, or can be created automatically. A simple way of automatic creation is to generate pseudo-random numbers, and one can use another source of data such as the clock time at boot or configuration time to provide additional entropy during the generation of unique IDs. Another way is to hash an IPv6 address down to a value as ID. The hash algorithm is supposed to be known and the same across the domain. Since typically the number of routers in a domain is far smaller than the value range of IDs, the hashed IDs are hardly likely to conflict with each other, as long as the hash algorithm is not designed too badly. It is necessary to be able to override the automatically created value, and desirable if the mechanism is provided by the system implementation.

If the ID is created from IPv6 address, e.g. by hashing from an IPv6 address, then naturally it has relationship with the address. If the ID is created regardless of IP address, one way to build association with IPv6 address is to embed the ID into an IPv6 address that is to be configured on the router, e.g. use a /96 IPv6 prefix and append it with a 32-bit long ID. One can also use some record keeping mechanisms, e.g. text file, DNS or other provisioning system like network management system to manage the IDs and mapping relations

with IPv6 addresses, though extra record keeping does introduce additional work.

Authors' Addresses

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
China

Email: liushucheng@huawei.com

Weiping Xu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
China

Email: xuweiping@huawei.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
China

Email: cathy.zhou@huawei.com

Tina Tsou
Philips Lighting
United States of America

Email: tina.tsou@philips.com

Simon Perreault
Jive Communications
Quebec, QC
Canada

Email: sperreault@jive.com

Peng Fan
Beijing
China

Email: fanp08@gmail.com

Rong Gu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing 100053
China

Email: gurong_cmcc@outlook.com

Chongfeng Xie
China Telecom
China Telecom Beijing Information Science&Technology Innovation Park
Beiqijia Town Changping District, Beijing 102209,
China

Email: xiechf.bri@chinatelecom.cn

Ying Cheng
China Unicom
No.21 Financial Street, XiCheng District
Beijing 100033
China

Email: chengying10@chinaunicom.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

S. Perreault
Viagenie
W. George
Time Warner Cable
T. Tsou
Huawei Technologies (USA)
T. Yang
L. Li
China Mobile
July 15, 2013

Turning off IPv4 Using DHCPv6 or Router Advertisements
draft-perreault-sunset4-noipv4-03

Abstract

This memo defines a new DHCPv6 option and a new Router Advertisement option for indicating to a dual-stack host or router that IPv4 is to be turned off.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. The Problems We're Trying to Fix	4
3.1. Load on DHCPv4 Server	4
3.2. Bandwidth Consumption	4
3.3. Power Inefficiency	4
3.4. IPv4 only Applications	4
4. Design Considerations	4
4.1. DHCPv6 vs DHCPv4	4
4.2. DHCPv6 vs RA	5
5. The No-IPv4 Option	6
5.1. DHCPv6 Wire Format	6
5.2. RA Wire Format	6
5.3. Semantics	7
5.4. Example	9
6. Security Considerations	10
7. IANA Considerations	10
8. Acknowledgements	10
9. References	10
9.1. Normative References	10
9.2. Informative References	11
Appendix A. Test Results of Terminals Behavior	11
Authors' Addresses	12

1. Introduction

When a dual-stack host makes a DHCPv4 request, it typically interprets the absence of a response as a failure condition. This makes it difficult to deploy such nodes in an IPv6-only network.

Take for example a home router that is dual-stack capable but provisioned with an IPv6-only WAN connection. When the router boots, it typically assigns an IPv4 address to its LAN interface, starts services on that interface, and starts handing out IPv4 addresses to clients on the LAN by answering DHCPv4 requests. This is done unconditionally, without taking the status of the IPv4 connectivity on the WAN interface into account. Hosts on the LAN, in turn, install a default route pointing to the router and start behaving as if IPv4 connectivity was available. IPv4 packets destined to the Internet get dropped at the router and timeouts happen. The end result is that IPv4 remains fully active on the LAN and on the router itself even when it is desired that it be turned off.

The other example is about DHCPv4 server. In Dual-Stack LAN/WLAN network or intranet, the core router or AC often plays the role of DHCP server, and the clients are server thousands PC or mobile phones. If the server is configured in IPv6-only, the dual-stack or IPv4-only clients will broadcast DHCPDISCOVER messages endlessly in the LAN or WLAN. The thousands clients will cause a DDOS-like attack to all the servers in the network. Since there are not specific descriptions in any RFCs for client's behavior when it does not receive the DHCPOFFER in response to its DHCPDISCOVER message, various OS deploy different backoff algorithms. We tested server popular OS(es), the test results is listed in the appendix.

A new mechanism is needed to indicate the absence of IPv4 connectivity or service that the goal is turning off IPv4, this new signaling mechanism shall be transported over IPv6. Therefore, we introduce a new DHCPv6 [RFC3315] option and a new Router Advertisement (RA) [RFC4861] option for the purpose of explicitly indicating to the host that IPv4 connectivity is unavailable.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms are also used in this document:

Upstream Interface: An interface on which the No-IPv4 option is received over either DHCPv6 or RA.

3. The Problems We're Trying to Fix

3.1. Load on DHCPv4 Server

When a DHCPv4 server is present but intentionally does not respond to a dual-stack node, the aggregated traffic generated by multiple such dual-stack nodes can represent a significant useless load. This scenario can be encountered for example with an ISP serving multiple types of subscribers where some will get IPv4 addresses and others not. It might not be feasible for operational reasons to block the useless requests before they reach the DHCPv4 server, e.g. if the DHCPv4 server itself is the one that has knowledge about which node should or should not get an IPv4 address.

3.2. Bandwidth Consumption

In addition to useless load on the DHCPv4 server, the above scenario could also consume a significant amount of bandwidth, particularly if the aggregated traffic from many clients goes through a low-bandwidth link.

3.3. Power Inefficiency

A dual-stack node that does not get a DHCPv4 response will usually continue retransmitting forever. Therefore, only providing IPv6 on a link will cause the node to needlessly wake up periodically and transmit a few packets. For example, the popular DHCPv4 client implementation by ISC wakes up every 5 minutes by default and tries to contact a DHCPv4 server for 60 seconds. With this configuration, a node will not be able to sleep 20% of the time.

3.4. IPv4 only Applications

In many cases, IPv4-only applications such as Skype use IPv4 LLA to bombard the LAN with IPv4 packets. In an IPv6-only environment, it can get quite annoying and waste a lot of bandwidth.

4. Design Considerations

4.1. DHCPv6 vs DHCPv4

NOTE: This section will be removed before publication as an RFC.

This document describes a new DHCPv6 option for turning off IPv4. An equivalent option could conceivably be created for DHCPv4. Here is a discussion of the pros and cons. Arguments with a + sign argue for a DHCPv4 option, arguments with a - sign argue against.

- + Devices that don't speak IPv6 won't be listening for a "turn off IPv4" code, and therefore won't stop trying to establish IPv4 connectivity.
- Devices that haven't been updated to speak IPv6 likely won't recognize a new DHCPv4 code telling them that IPv4 isn't supported.
 - + However, it's easier to implement something that turns off the IP stack than implement IPv6.
- Devices that don't speak IPv6 that are still active on the network mean that either IPv4 can't/shouldn't be turned off yet, or IPv4 local connectivity should be maintained to retain local services, even if global IPv4 connectivity is not necessary (think local LAN DLNA streaming, etc).
- When the goal is to turn off IPv4, having to maintain and operate an IPv4 infrastructure (routing, ACLs, etc.) just to be able to send negative responses to DHCPv4 requests is not productive. Having the option transported in IPv6 allows the ISP to focus on operating an IPv6-only network.
 - + However, a full IPv4 infrastructure would not be necessary in many cases. The local router could contain a very restricted DHCPv4 server function whose only purpose would be to reply with the No-IPv4 option. No IPv4 traffic would have to be carried to a distant DHCPv4 server. Note however that this may not be operationally feasible in some situations.
- Turning IPv4 off using an IPv4-transported signal means that there is no way to go back. Once the DHCPv4 option has been accepted by the DHCPv4 client, IPv4 can no longer be turned on remotely (rebooting the client still works). Configurations change, mistakes happen, and so it is necessary to have a way to turn IPv4 back on. With a DHCPv6 option, IPv4 can be turned back on as soon as the client makes a new DHCPv6 request, which can be the next scheduled one or can be triggered immediately with a Reconfigure message.

The authors conclude that a DHCPv6 option is clearly necessary, whereas it is not as clear for a DHCPv4 option. More feedback on this topic would be appreciated.

4.2. DHCPv6 vs RA

Both DHCPv6- and RA-based solutions are presented in this draft. It is expected that the working group will decide whether both solutions, only one, or none are desirable.

5. The No-IPv4 Option

The No-IPv4 DHCPv6 option is used to signal the unavailability of IPv4 connectivity.

5.1. DHCPv6 Wire Format

The format of the DHCPv6 No-IPv4 option is:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                   OPTION_NO_IPV4                   | option-len |
+-----+-----+-----+-----+-----+-----+-----+-----+
|    v4-level    |
+-----+-----+-----+-----+-----+-----+-----+

```

option-code OPTION_NO_IPV4 (TBD).

option-len 1.

v4-level Level of IPv4 functionality.

The DHCPv6 client MUST place the OPTION_NO_IPV4 option code in the Option Request Option ([RFC3315] section 22.7). Servers MAY include the option in responses (if they have been so configured). Servers MAY also place the OPTION_NO_IPV4 option code in an Option Request Option contained in a Reconfigure message.

5.2. RA Wire Format

The format of the RA No-IPv4 option is:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|    Type    |    Length    |    v4-level    |    Reserved    |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Reserved                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type TBD

Length	1.
v4-level	Level of IPv4 functionality.
Reserved	These fields are unused. They MUST be initialized to zero by the sender and MUST be ignored by the receiver.

5.3. Semantics

The option applies to the link on which it is received. It is used to indicate to the client that it should disable some or all of its IPv4 functionality. What should be disabled depends on the value of v4-level.

v4-level can take the following values:

- 0 - IPv4 fully enabled: This is equivalent to the absence of the No-IPv4 option. It is included here so that a DHCPv6 server can explicitly re-enable IPv4 access by including it in a Reply message following a Reconfigure, or similarly by a router in a spontaneous Router Advertisement.
- 1 - No IPv4 upstream: Any kind of IPv4 connectivity is unavailable on the link on which the option is received. Therefore, any attempts to provision IPv4 by the host or to use IPv4 in any fashion, on that link, will be useless. IPv4 MAY be dropped, blocked, or otherwise ignored on that link.

Upon reception of the No-IPv4 option with value 1, the following IPv4 functionality MUST be disabled on the Upstream Interface:

- a. IPv4 addresses MUST NOT be assigned.
 - b. Currently-assigned IPv4 addresses MUST be unassigned.
 - c. Dynamic configuration of link-local IPv4 addresses [RFC3927] MUST be disabled.
 - d. IPv4, ICMPv4, or ARP packets MUST NOT be sent.
 - e. IPv4, ICMPv4, or ARP packets received MUST be ignored.
 - f. DNS A queries MUST NOT be sent, even transported over IPv6.
- 2 - No IPv4 upstream, local IPv4 restricted: Same semantics as value 1, with the following additions:

If all DHCPv6- or RA-configured interfaces receive the No-IPv4 option with a mix of values 1, 2, and 3 (but not exclusively 3), and no other interface provides IPv4 connectivity to the Internet, IPv4 is partially shut down, leaving only local connectivity active. On the Upstream Interface, IPv4 MUST be shut down as listed above. On other interfaces, IPv4 addresses MUST NOT be assigned except for the following:

- * Loopback (127.0.0.0/8)
- * Link Local (169.254.0.0/16) [RFC3927]
- * Private-Use (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) [RFC1918]

- 3 - No IPv4 at all: This is intended to be a stricter version of the above.

The host or router receiving this option MUST disable IPv4 functionality on the Upstream Interface in the same way as for value 1 or 2.

If all DHCPv6- or RA-configured interfaces received the No-IPv4 option with exclusively value 3, and no other interface provides IPv4 connectivity to the Internet, IPv4 is completely shut down. In particular:

- a. IPv4 address MUST NOT be assigned to any interface.
- b. Currently-assigned IPv4 addresses MUST be unassigned.
- c. Dynamic configuration of link-local IPv4 addresses [RFC3927] MUST be disabled.
- d. IPv4, ICMPv4, or ARP packets MUST NOT be sent on any interface.
- e. IPv4, ICMPv4, or ARP packets received on any interface MUST be ignored.
- f. In the above, "any interface" includes loopback interfaces. In particular, the 127.0.0.1 special address MUST be removed.
- g. Server programs listening on IPv4 addresses (e.g., a DHCPv4 server) MAY be shut down.
- h. DNS A queries MUST NOT be sent, even transported over IPv6.

- i. If the host or router also runs a DHCPv6 server, it SHOULD include the No-IPv4 option with value 2 in DHCPv6 responses it sends to clients that request it, unless prohibited by local policy. If it currently has active clients, it SHOULD send a Reconfigure to each of them with the OPTION_NO_IPV4 included in the Option Request Option.
- j. If the router sends Router Advertisement, it SHOULD include the No-IPv4 option with value 2 in RA messages it sends, unless prohibited by local policy. It SHOULD also send RAs immediately so that the changes take effect for all current hosts.

The intent is to remove all traces of IPv4 activity. Once the No-IPv4 option with value 3 is activated, the network stack should behave as if IPv4 functionality had never been present. For example, a modular kernel implementation could accomplish the above by unloading the IPv4 kernel module at run time.

5.4. Example

A dual-stack home gateway is set up with a single WAN uplink and is configured to use DHCPv4 and DHCPv6 to automatically obtain IPv4 and IPv6 connectivity. On the LAN side, it has one link with multiple hosts.

When it boots, the router assigns 192.168.1.1/24 to its LAN interfaces and starts a DHCPv4 server listening on it. It hands out addresses 191.168.1.100-199 to clients. It also starts an IPv6 Router Advertisement daemon as well as a stateless DHCPv6 server, also listening on the LAN interfaces.

On the WAN side, it starts two provisioning procedures in parallel: one for IPv4 and one for IPv6.

At this point, the ISP does not know if the router supports IPv6-only operation. Therefore, by default, the ISP responds to DHCPv4 requests as usual.

As part of the IPv6 provisioning procedure, the router sends a DHCPv6 request containing OPTION_NO_IPV4 in an Option Request Option. The ISP's DHCPv6 server's reply includes the No-IPv4 option with value 3. When this procedure finishes, the ISP has determined that this customer will run in IPv6-only mode and starts dropping all IPv4 packets at the first hop. If an IPv4 address was assigned, it is reclaimed, and possibly reassigned to another subscriber.

The home router aborts the IPv4 provisioning procedure (if it is still running) and deactivates all IPv4 functionality. It shuts down its DHCPv4 server. It also configures its own stateless DHCPv6 server to send the No-IPv4 option to clients that request it.

As an optimization, the router could delay setting up IPv4 by a few seconds (10 seconds seems reasonable). If the IPv6 procedure completes with the No-IPv4 option during that time, IPv4 will never have been set up and the router will operate in pure IPv6-only mode from the start.

6. Security Considerations

One security concern is that an attacker could use the No-IPv4 option to deny IPv4 access to a victim. However, unprotected vanilla DHCP can already be exploited to cause such a denial of service ([RFC2131] section 7).

TO BE COMPLETED

7. IANA Considerations

IANA is requested to assign value TBD with description OPTION_NO_IPV4 in the "DHCP Option Codes" table which is part of the dhcpv6-parameters registry [1].

IANA is requested to assign value TBD with description "No-IPv4 Option" in the IPv6 Neighbor Discovery Option Formats table which is part of the icmpv6-parameters registry.

8. Acknowledgements

Thanks in particular to Marc Blanchet who was the driving force behind this work.

Rajiv Asati contributed section Section 3.4.

9. References

9.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

9.2. Informative References

- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

Appendix A. Test Results of Terminals Behavior

In RFC3315 [RFC3315, DHCPv6], SOL_MAX_RT is defined in DHCPv6 to prevent the frequently requesting of clients, which reduces the aggregated traffic. But in RFC2131 [RFC2131, DHCPv4], there are not corresponding IPv4 definitions or options for client's behavior if the server does not respond for the Discover messages.

In fact, most of the terminals creat backoff algorithms to help them retransmit DHCPDISCOVER message in different frequency according to their state machine. The same point of almost all the verious Operating Systems is that they could not stop DHCPDISCOVER requests to the server. And that will cause DDoS-Like attack to the server and bandwidth consumption in the link.

We test some of the most popular terminals' OS in WLAN, the results are illuminated as below.

DHCP Discovery Packages Time Table

No	Windows7		Windows XP		IOS_5.0.1		Android_2.3.7		Symbian_S60	
	Time	Time offset	Time	Time offset	Time	Time offset	Time	Time offset	Time	Time offset
1	0		0		0.1		7.8		0	
2	3.9	3.9	0.1	0.1	1.4	1.3	10.3	2.5	2	2
3	13.3	9.4	4.1	4	3.8	2.4	17.9	7.6	6	4
4	30.5	17.2	12.1	8	7.9	4.1	33.9	16	8	2
5	62.8	32.3	29.1	17	16.3	8.4	36.5	2.6	12	4

6	65.9	3.1	64.9	35.8	24.9	8.6	reconnect		14	2
7	74.9	9	68.9	4	33.4	8.5	56.6	20.1	18	4
8	92.1	17.2	77.9	9	42.2	8.8	60.2	3.6	20	2
9	395.2	303.1	93.9	16	50.8	8.6	68.4	8.2	24	4
10	399.1	3.9	433.9	340	59.1	8.3	84.8	16.4	26	2
11	407.1	8	438.9	5	127.3	68.2	86.7	1.9	30.1	4.1
12	423.4	16.3	447.9	9	128.9	1.6	reconnect		32.1	2
13	455.4	32	464.9	17	131.1	2.2	106.7	20	36.1	4
14	460.4	5	794.9	330	135.1	4	111.4	4.7	38.1	2
15	467.4	7	799.9	5	143.4	8.3	120.6	9.2	42.1	4
16	483.4	16	808.9	9	151.7	8.3	134.9	14.3	44.1	2
17	842.9	359.5	824.9	16	160.4	8.7	136.8	1.9	48.2	4.1
18	846.9	4	1141.9	317	168.8	8.4	reconnect		50.2	2

Figure:Terminals DHCPDISCOVER requests when Server's DHCPv4 module is down

In this figure:

For Windows7, it seems to initiate 8 times DHCPDISCOVER requests in about 300s interval.

For WindowsXP, firstly it launches 9 times DHCPDISCOVER messages, but after that it cannot get any response from the server, then it initiates 5 times requests in one cycle in around 330s intervals, and never stop.

For IOS5.0.1, it seems like WindowsXP. There are 10 times attempts in one cycle, and the interval is about 68s.

Symbian_S60 uses the simplest backoff method, it launches DISCOVER in every 2 or 4 seconds.

Android2.3.7 is the only Operating System which can stop DISCOVER request by disconnect its wireless connection. It reboot wireless and dhcp connection every 20 seconds.

Authors' Addresses

Simon Perreault
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Phone: +1 418 656 9254
Email: simon.perreault@viagenie.ca
URI: <http://viagenie.ca>

Wes George
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA 20171
USA

Email: wesley.george@twcable.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4424
Email: tina.tsou.zouting@huawei.com

Tianle Yang
China Mobile
32, Xuanwumenxi Ave.
Xicheng District, Beijing 100053
China

Email: yangtianle@chinamobile.com

Li Lianyuan
China Mobile
32, Xuanwumenxi Ave.
Xicheng District, Beijing 100053
China

Email: lilianyuan@chinamobile.com