Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 15, 2014

G. Chen
China Mobile

July 14, 2013

Analysis of NAT64 Port Allocation Method
draft-chen-sunset4-cgn-port-allocation-02

Abstract

The document enumerates methods of port assignment in CGN contexts, more focusing on a NAT64 environment. The analysis categorizes the different methods with several key features. The uses of existing protocols are further described corresponding to those features. This memo also states the port-usage experiences, relevant findings, evaluations and workarounds. It's expected the document could provide an 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 January 15, 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. Port Consumption on NAT64 3
3. Port Allocation Category and Usages 3
3.1. NAT vs NAPT 3
3.2. Dynamic vs Static 4
3.3. Centralized vs Distributed 5
4. Log Volume Optimization 6
5. Connectivity State Optimization 7
6. Port Randomization 8
7. Security Considerations 8
8. IANA Considerations 8
9. Acknowledgements 8
10. References 8
10.1. Normative References 8
10.2. Informative References 9
Author's Address 10

1. Introduction

With the depletion of IPv4 address, CGN(Carrier Grade NAT) has been adopted by operators to expand IPv4 spaces. Relying upon the mechanism of multiplexing 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. [RFC6888] defined the term of CGN. Several proposals including DS-Lite[RFC6333], NAT64[RFC6145], [RFC6146], NAT444 would likely fall into the scope. Focusing on the IPv6 migration, the memo elaborates the port allocation methods in a NAT64 environment, where there are IPv6-only nodes connected.

There are several aspects may have to consider in order to deploy a suitable method. The below enumerates the potential aspects.

- Specific features of port-usages in a NAT64 environment
- The category of different port-allocations methods
- The port allocation method to improve connectivity
- The port allocation method to optimize log volume
- The port allocation method to enhance security

The document is trying to detail the analysis and relevant experiences.

2. Port Consumption on NAT64

With the merits of simplicity and efficiency, NAT64 will be likely deployed. In those cases, NAT64 would enable internal IPv6 hosts to connect to external dual-stack networks. Compared with NAT44, fewer ports consumed on NAT64. The reason for the fewer port consumption is NAT64 are deployed to provide connectivities from one address family to another. Only flows between different address families require ports to be assigned. That is, a NAT44 might be deployed in an IPv4-only environment. Since all traffic will have to traverse the NAT, all flows will need ports. Conversely, NAT64 only requires a port when one end is IPv4-only. Therefore, the more hosts support IPv6, the fewer ports are needed on the NAT64.

There was a testing on the comparison of port consumption on NAT64 and NAT44. Top100 websites (referring to Alexa statistics) were being assessed to evaluate status of port usage on NAT44 and NAT64 respectively. It's observed that the port consumptions on NAT64 is roughly only half on NAT44. 43 percent of top100 websites have AAAA records, therefore the NAT64 didn't have to assign ports to the traffic going to those websites. The results may be different if more services (e.g. game, web-mail, etc) are considered. Whereas, it's obvious that the port saving on NAT64 could be amplified by increasing native IPv6 supports.

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 port allocations. 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 IPv6 migration, NAT64 may relax the multiplexing ratio of shared IPv4 address by either a distributed port delegation or a centralized control.

3. Port Allocation Category and Usages

This section lists several methods to allocate the port information in NAT64 equipments. It also describes exemplified cases for each allocation model.

3.1. 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. Some existing practices are listed below.

o Stateful

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

In the case of static address assignments, 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.

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 a time-division manner.

o Stateless

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. A promising usage of stateless could be appeared in IDC(Internet Data Center) environments where there are IPv6 servers farms to receive inbound connections from external IPv4 users [I-D.anderson-siit-dc]. The other uses may consider two issues. First off, the static one-to-one mapping may didn't resolve IPv4 depletions. Secondly, it introduced the dependency between IPv4 and IPv6. It causes new limitations since the changes of IPv4 address lead renumbering of IPv6 addresses.

3.2. Dynamic vs Static

There are two methods on port allocations.

o Dynamic assignments

NAT64 normally do the dynamic assignments, since maximum port utilizations could be achieved. In respect to port allocations, it could be allocated with the granularity of per-session or per-customer. Per-session assignments are basically configured on NAT64 by default for efficient port utilizations. However, a heavy log volume may have to be recorded for lawful interception uses. In order to mitigate the concerns, bulk port allocation is enabled .When a subscriber creates the first session, a number of ports are pre-

allocated. It would significantly reduce log volumes. It's desirable to configure a proper port-range for each subscriber. Two aspects are listed below.

a. The number of session initiations for each subscriber. A subscriber normally uses multiple applications simultaneously, e.g. map, online video or games. The number of concurrent sessions are essential to determine the size of port range. It has been learned from subscriber's behaviors that the average session number of a standalone device is around 200~300 ports. Several devices maybe appeared behind a CPE. Operators may configure a range with 1000 ports to each CPE(Customer Premises Equipment) in a residential network.

b. Impacts to NAT64 capacity. The pre-assigned port ranges occupy the memory even there are unused ports. NAT64 CGN served a centralized point for numerous subscribes. Therefore, it should be cautious the impacts of port-range reservation to the capacity of attempted concurrent sessions.

o Static assignments

The static assignment makes a bulk of port reservations for each internal address before subscriber's connection. The bulk of ports could be either a contiguous or non-contiguous port range for the sake of attack defense. [I-D.donley-behave-deterministic-cgn]has described a deterministic NAT to assign a port range for internal IP address pool in a sequence. The difference of the static method with dynamic port-range is address/ports mappings have been established before subscriber's connection attempts. Log recording may not be necessary due to the stable mapping relations. The considerations of port-range allocation and capacity impacts could also be applied to the case of static assignments.

3.3. Centralized vs Distributed

There are increasing needs to connect NAT64 with downstream NAT46-capable devices to support IPv4 users/applications on a IPv6-only path. Several solutions have been proposed in this area, e.g. 464xlat[RFC6877], 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 categorized as a centralized assignment on NAT64 or a port delegation distributed to downstream devices (e.g, customer edges connected with NAT64) .

o Centralized Assignment

A centralized method would make port assignments once IP flows come to NAT64. The allocation policy is enforced on a centralized point. 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-port-set][I-D.ietf-softwire-map-dhcp]). The distributed model normally performs A+P style for static port assignment. NAT64 should hold the consistent mapping in accordance with assigned ports. Those methods could shift NAT64 port computations/states into downstream devices. The detailed benefits was documented in [I-D.ietf-softwire-stateless-4v6-motivation].

4. Log Volume Optimization

[RFC6269] has provided a thoughtful analysis on the issues of IP sharing. It pointed out that IP sharing may bring the impacts to law enforcements 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 in a particular time slot. The storage of log information may post a challenge to operators, since it requires additional transport, storage resources and data inspection process to indentify users. It's desirable to compact the logging information. Referring the categories of port allocation, the assignment could be managed on either per-session or per-customer. The bigger granularity would lead fewer log volume storage. A testing was made to record the log information from 200,000 subscribers for 60 days. The volume of recorded information reach up to 42.5 terabytes with per-session log. Conversely, it only occupy 40.6 gigabytes with per-customer log. There is even a method, which doesn't have to log any information.

Whereas, high compression would cause lower efficiency of port utilization. A port allocation based on per-customer granularity have to retain vacant ports in order to avoid traffic overflow. The efficiency could be evaluated by port utilization rate. The below table is trying to make a composite analysis.

Type of log records	Method of port allocations	Log Volume(e.g. 200,000 users for 60 days)	Port utilization ratio

per-session	Dynamic NAPT	43.5 Terabytes	100%
per-customer	Dynamic port-range	40.6 Gigabytes	75%(e.g.400 ports)
None Log	Deterministic NAT, MAP,4rd	0	60% *75%= 45%

Note:75% is evaluated for port utilization ratio.
60% is evaluated for the ratio of active subscribers.

The data shared in the table may roughly demonstrates the tradeoff between port utilization and log volume compression. The efficiency could be even lower if static bulk-port allocation is used since the ports were pre-allocated to customers regardless online or offline status. Administrator may consider below factors to determine their own solution

- Average connectivity per customer per day
- Peak connectivity per day
- The amount of public IPv4 address in NAT64
- Application demands for specific ports
- The processing capabilities of NAT64
- The tolerance of log volume

5. Connectivity State Optimization

It's observed that port consumption would be significantly increased when subscribers stick to a web page for VoD (Video On Demand), online games or map services. In those cases, multiple TCP connections may be initiated to optimize the performance of data transmissions for video download and message exchange. With the trends of the video traffic growth, it likely presents a challenge for network operators that need to optimize connectivity states so as to avoid port depletions. Those optimizations may even be significant to the method of port-range allocation, because a subscriber is only allowed to use a pre-configured port resource.

The optimization could be considered from two aspects.

- Reducing the TIME-WAIT state. Acceleration of TIME-WAIT state transitions could increase the efficiency of port utilization. [RFC6191] defines the mechanism of reducing TIME-WAIT state by

proposing TCP timestamps and sequence numbers. [I-D.penno-behave-rfc4787-5382-5508-bis] recommended applying [RFC6191] and PAWS (Protect Against Wrapped Sequence numbers described in [RFC1323]) to NAT. It might a way to improve port utilizations.

- o 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 this use didn't recommended by [RFC6888] (e.g. REQ-7) that may reduce the incentives.

6. Port Randomization

Port randomization is a feature to enhance the defense of hijack flows. [RFC6056] specified that NAPTs should consider obfuscating the selection of ephemeral ports. A NAPT based on per-session may be less difficult to implement by allocating non-contiguous port to each connection. The methods of port-range allocation have to correlate the algorithm configuration and input parameters between NAT64 and log system to identify the subscribers. In general, a simply algorithm on port assignment is mostly desirable to simplify the log process. It could be considerable to enlarge the size of port range to alleviate security issues, if the port resource is allowed.

7. Security Considerations

The non-contiguous port allocations could be considered to enhance the security of port allocations. This document shares the considerations in [RFC6056].

8. IANA Considerations

This document makes no request of IANA.

9. Acknowledgements

The author would like to thank Lee Howard and Simon Perreault for their helpful comments.

Many thanks to Wesley George and Marc Blanchet encourage the author to continue this work.

10. References

10.1. Normative References

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for Mapping of Address and Port", draft-ietf-softwire-map-dhcp-03 (work in progress), February 2013.

[RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[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.

[RFC6191] Gont, F., "Reducing the TIME-WAIT State Using TCP Timestamps", BCP 159, RFC 6191, April 2011.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

10.2. 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-06 (work in progress), July 2013.

[I-D.ietf-pcp-port-set]
Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", draft-ietf-pcp-port-set-01 (work in progress), May 2013.

[I-D.ietf-softwire-4rd]

Despres, R., Jiang, S., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd)", draft-ietf-softwire-4rd-06 (work in progress), July 2013.

[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-03 (work in progress), July 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.penno-behave-rfc4787-5382-5508-bis]
Penno, R., Perreault, S., Kamiset, S., Boucadair, M., and K. Naito, "Network Address Translation (NAT) Behavioral Requirements Updates", draft-penno-behave-rfc4787-5382-5508-bis-04 (work in progress), January 2013.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

[RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.

Author's Address

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

Email: phdgang@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 11, 2014

H. Hazeyama
NAIST / WIDE Project
T. Ishihara
Univ. of Tokyo / WIDE Project
O. Nakamura
Keio Univ. / WIDE Project

July 10, 2013

DNS A Record Filtering for the migration from dual stack networks to IPv6 only networks. draft-hazeyama-sunset4-dns-a-filter-00

Abstract

Filtering out of A records of a DNS response on a DNS proxy, we call it ``DNS A record filtering'', is an effective and efficient solution as a smooth migration to IPv6 only networks. DNS A record filtering can mitigate fallback problems of dual stack nodes on IPv6 only environment. This memo mentions the components of the DNS A record filter solution, procedure of DNS queries and refers current issues.

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 11, 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 3
1.1. Requirements Language 3
2. Technology and Terminology 3
3. The mechanism of DNS A Record Filtering 4
3.1. Assumptions 4
3.2. Components 4
3.3. Procedure 6
3.3.1. IPv6-only hosts 6
3.3.2. IPv6-full-capable dual stack host 6
3.3.3. IPv6-partial-capable dual stack host 7
4. Discussions 9
4.1. Limitation for IPv4 only applications 9
4.2. CNAME of the reply to an type A query 9
5. Security Considerations 9
6. IANA Considerations 9
7. References 10
7.1. Normative References 10
7.2. Informative References 10
Appendix A. Acknowledgments 11
Authors' Addresses 11

1. Introduction

In an IPv6 only network [RFC6586], that is composed of DHCP6 and DNS64/NAT64, IPv4/IPv6 dual stack hosts have fallback problems due to the partial IPv6 capability, happy eyeball functions [RFC6555], default route on IPv4 link local address due to the link local assumption[RFC3927], arrival timings of DNS responses, and so on.

As well as so-called DNS AAAA record filtering in IPv4 only networks, filtering out of A records of a DNS response on a DNS proxy, we call it ‘‘DNS A record filter proxy’’, is an effective and efficient solution to mitigate fallback problems of dual stack nodes on IPv6 only environment.

DNS A record filtering solution allows dual stack nodes to resolve names both by IPv4 and IPv6 by notifying the IPv4 address of an DNS A record filter proxy through DHCP4 and the IPv6 address of the DNS A record filter proxy through DHCP6. On the other hand, a DNS A record filter proxy forces dual stack nodes to conduct actual communications after the name query procedure through IPv6, by telling only AAAA or NAT64 mapped AAAA records to dual stack nodes. In this solution, no special action is required on dual stack nodes. A network operator can choose DNS64 and NAT64 location along with their design policy.

1.1. Requirements Language

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].

2. Technology and Terminology

In this document, the following terms are used. "Dual Stack" refers to a technique for providing complete support for both Internet protocols -- IPv4 and IPv6 -- in hosts and routers [RFC4213].

"NAT64" refers to a Network Address Translator - Protocol Translator defined in [RFC6052], [RFC6144], [RFC6145], [RFC6146], [RFC6384].

"DNS64" refers DNS extensions to use NAT64 translation from IPv6 clients to IPv4 servers with name resolution mechanisms that is defined in [RFC6147].

"DHCP4" refers Dynamic Host Configuration Protocol for IPv4 that is defined in [RFC2131].

"DHCP6" refers Dynamic Host Configuration Protocol for IPv6. So

called "Stateful DHCP6" is defined in [RFC3315] and "Stateless DHCP6" is defined in [RFC3736]. "DHCP-PD" or "DHCPv6 Prefix Delegation" refers IPv6 Prefix Options for DHCP6 that is initially defined in [RFC3633] and updated in [RFC6603].

"ND" refers Neighbor Discovery for IP version 6 (IPv6) that is defined in [RFC4861] and updated in [RFC5942].

3. The mechanism of DNS A Record Filtering

3.1. Assumptions

The DNS A record filtering simply filters out ''A record'' entry of a DNS reply on a DNS proxy. As our assumption, the DNS A record filtering solution is mainly used in an IPv6 only network by combining DNS64/NAT64, DHCP4 and DHCP6.

We also assume that hosts are dual stack capable, that is, hosts have ND function and the IPv6 address assignment function by RA at least. Stateful DHCP6, Stateless DHCP6 and IPv6 DNS query functions are preferable to be equipped by hosts.

3.2. Components

The components of the network, where this DNS A record filtering is employed, are as follows;

- DHCP4 server: this DHCP4 server offers a private IPv4 address to mitigate the long fall back problem due to the IPv4 link local assumption. The DHCP4 server also offers the IPv4 address of the DNS A record filter proxy. To avoid the selection of the IPv4 by Happy Eyeball [RFC6555] in a dual stack host, this DHCP4 server MUST NOT provide the IPv4 default route.

- DHCP6 server: this DHCP6 server MUST provide the IPv6 address of the DNS A record filter proxy. Both stateful DHCP6 and stateless DHCP6 can be employed. If the subnet is composed of a security switch and/or security wi-fi controllers, stateful DHCP6 is prefer to avoid the blocking due to the multiple temporary IPv6 address on a host.

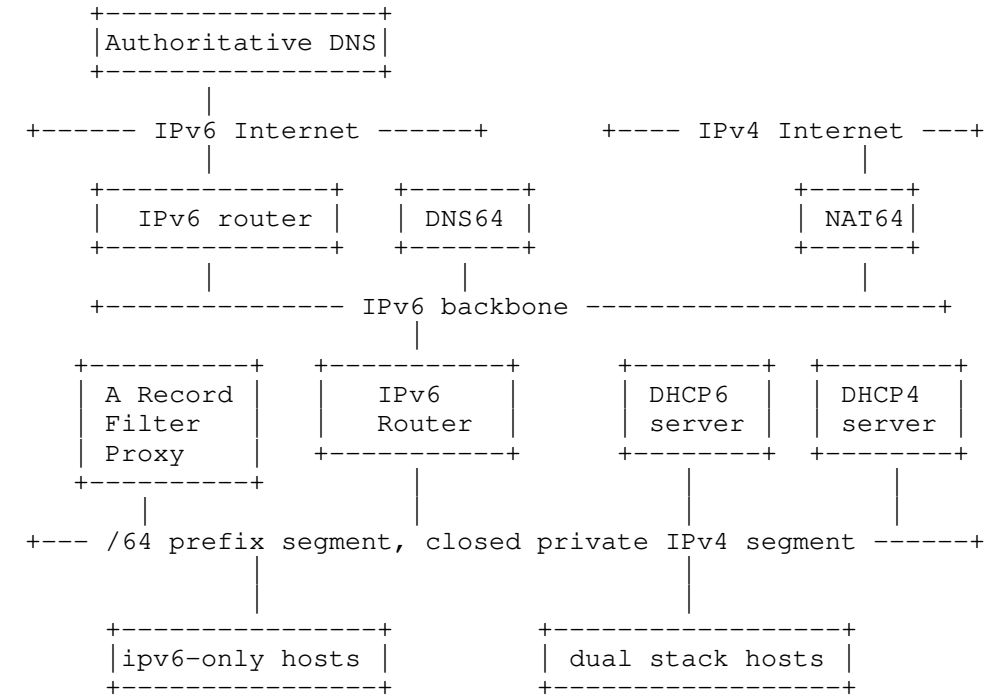
- DNS A record filter proxy : this DNS proxy is the key component of this solution. The DNS proxy SHOULD be located on the leaf subnet. The DNS proxy has a private IPv4 address that SHOULD be the same subnet address provided by the DHCP4. The DNS proxy has an IPv6 address that is announced to hosts through DHCP6.

- DNS64 server : at least, one DNS64 server is required. The DNS A record filter proxy forwards all queries to this DNS64 server directly, or several DNS forwarder can be placed for load balancing of DNS64 servers

- Authoritative DNS servers : these authoritative DNS servers would be queried by DNS64 servers. These authoritative DNS servers MUST NOT return inappropriate replies mentioned in [RFC4074] to kick the fallback function of DNS64 servers

- NAT64 translators : at least, one NAT64 translator is placed that can be reached by hosts through IPv6. A NAT64 translator can be settled as the gateway of the leaf subnet, or an aggregated translator of the intra network, or a global reachable open translator. Several NAT64 translators can be registered in DNS64 servers for the load balancing or for handling different IPv4 prefixes by each NAT64 translrator.

Figure 1 shows a sample network topology of this solution.



A sample network topology of DNS A record filtering

Figure 1

3.3. Procedure

3.3.1. IPv6-only hosts

The procedure on IPv6-only hosts is as follows;

- The host connects to the leaf subnet in layer 2 level.
- The host gets global IPv6 address through RA or stateful DHCP6, and also learns the IPv6 address of the DNS A record filter proxy.
- When the host connects to an URL, the hosts queries by type ANY or by type AAAA to the DNS A record filter proxy through IPv6.
- The DNS A record filter proxy forwards the received the type ANY query to the upper DNS forwarder or DNS64 server.
- When the DNS64 server receives the query, the DNS64 server forwards the issued FQDN to the upper authoritative DNS.
  - If the FQDN has AAAA record, the DNS64 returns AAAA record to the DNS A record filter proxy.
  - If the FQDN has only A record, the DNS64 returns NAT64 prefix mapped AAAA record to the DNS A record filter proxy.
  - The DNS64 server or the upper DNS forwarder may return A record to the DNS A record filter proxy with AAAA record.
- When the DNS A record filter proxy receives the reply, the DNS A record filter proxy filters out A record if the reply contains A record.
- The DNS A record filter proxy returns only AAAA records to the host.
- The host access to the issued URL through the IPv6 address of the destination or the NAT64 prefix mapped address.

3.3.2. IPv6-full-capable dual stack host

An IPv6-full-capable dual stack node equips DHCP6 function and IPv6 DNS query function, therefore, IPv6-full-capable dual stack node can send DNS queries through both IPv4 and IPv6.

The procedure on IPv6-full-capable dual stack hosts (like Windows 7,

etc.) is as follows;

- The host connects to the leaf subnet in layer 2 level.
- The host gets a global IPv6 address through RA or stateful DHCP6, and also learns the IPv6 address of the DNS A record filter proxy.
- The host also gets a private IPv4 address through DHCP4, and also learns the IPv4 address of the DNS A record filter proxy.
- The network connectivity check sequence of the Operating System may run, then, IPv6 will be selected on the host because the IPv4 is not global reachable.
- When the host connects to an URL, the host queries by type ANY to the DNS A record filter proxy through IPv6.
- The DNS A record filter proxy forwards the received type ANY query to the upper DNS forwarder or DNS64 server.
- When the DNS64 server receives the query, the DNS64 server forwards the issued FQDN to the upper authoritative DNS.
  - If the FQDN has AAAA record, the DNS64 returns AAAA record to the DNS A record filter proxy.
  - If the FQDN has only A record, the DNS64 returns NAT64 prefix mapped AAAA record to the DNS A record filter proxy.
  - The DNS64 server or the upper DNS forwarder may return A record to the DNS A record filter proxy with AAAA record.
- When the DNS A record filter proxy receives the reply, the DNS A record filter proxy filters out A record if the reply contains A record.
- The DNS A record filter proxy returns only AAAA records to the host.
- The host access to the issued URL by using the IPv6 address of the destination or the NAT64 prefix mapped address.

3.3.3. IPv6-partial-capable dual stack host

An IPv6-partial-capable dual stack node does not equip either DHCP6 function or IPv6 DNS query function, therefore, IPv6-partial-capable dual stack node will send DNS queries through only IPv4. However, IPv6-partial-capable dual stack can recognize AAAA record or IPv6

address.

The procedure on IPv6-partial-capable dual stack host is as follows;

- The host connects to the leaf subnet in layer 2 level.
- The host gets a global IPv6 address through RA.
- The host also gets a private IPv4 address through DHCP4, and also learns the IPv4 address of the DNS A record filter proxy.
- The network connectivity check sequence of the Operating System may run, then, IPv6 may be selected on the IPv6-preferred host because the IPv4 is not global reachable. In some case, the network connectivity check may pass by name resolution to the anchor server, or the network connectivity may run again with certain interval.
- When the host connects to an URL, the host queries by type ANY to the DNS A record filter proxy through IPv4.
- The DNS A record filter proxy forwards the received type ANY query to the upper DNS forwarder or DNS64 server through IPv6.
- When the DNS64 server receives the query, the DNS64 server forwards the issued FQDN to the upper authoritative DNS.
  - If the FQDN has AAAA record, the DNS64 returns AAAA record to the DNS A record filter proxy.
  - If the FQDN has only A record, the DNS64 returns NAT64 prefix mapped AAAA record to the DNS A record filter proxy.
  - The DNS64 server or the upper DNS forwarder may return A record to the DNS A record filter proxy with AAAA record.
- When the DNS A record filter proxy receives the reply, the DNS A record filter proxy filter out A record if the reply contains A record.
- The DNS A record filter proxy returns only AAAA records to the host.
- The host access to the issued URL by using the IPv6 address of the destination or the NAT64 prefix mapped address.

4. Discussions

4.1. Limitation for IPv4 only applications

As mentioned in [RFC6586], IPv4-only (or IPv6-incapable) applications exist. Such IPv4-only applications will not work on this DNS A record filtering environment. It is preferable that such IPv4-only applications become dual stack applications if possible.

4.2. CNAME of the reply to an type A query

We conducted a field trial of this DNS A record filter solution in Interop Tokyo 2013. We provided an IPv6-only Wi-Fi access with this DNS A record filter solution. We used the current Debian release and bind 9.9.2-p1 patch provided from WIDE project as the DNS A Record filter proxy.

In the hot stage of Interop Tokyo 2013, we met a trouble case of the current DNS A record filter. In the trouble case, a host used Firefox browser and crawled several web pages for test. In some web page, several contents were lost. We inspected by packet captures, the reply of the A query to the host arrived faster than the arrival of the reply of AAAA record. The reply of A query contained as type CNAME, that was not filtered in the current bind A filter patch. (The A filter patch removed type A record of the CNAME.) On the other hand, in a successful case, the reply of AAAA record, that contained type CNAME and type AAAA of the CNAME, arrived faster than the type A reply.

Of course, we have to conduct further investigation and tests, the CNAME on a type A reply would be removed by DNS A filter proxy as well as type A record on the type A reply.

5. Security Considerations

As well as mentioned in [RFC6586], the use of IPv6 instead of IPv4 by itself does not make a big security difference.

6. IANA Considerations

This document has no IANA implications.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

7.2. Informative References

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

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.

[RFC4074] Morishita, Y. and T. Jinmei, "Common Misbehavior Against DNS Queries for IPv6 Addresses", RFC 4074, May 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, July 2010.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.

[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.

[RFC6147] 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.

[RFC6384] van Beijnum, I., "An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation", RFC 6384, October 2011.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.

[RFC6586] Arkko, J. and A. Keranen, "Experiences from an IPv6-Only Network", RFC 6586, April 2012.

[RFC6603] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, May 2012.

Appendix A. Acknowledgments

T. Jinmei of Internet Systems Consortium for providing DNS A filter patch of Bind 9.

O. Onoe of Sony Corporation for his deep inspection and testing of end node devices in WIDE project meeting in September 2012.

Interop Tokyo 2013 NOC members and Nano Opt Media for providing a field trial of this DNS A Record Filter solution as a service network. Especially, K. Mano of A10 Networks and STM members for their deep inspection and testing.

Authors' Addresses

Hiroaki Hazeyama
NAIST / WIDE Project
Takayama 8916-5
Nara,
Japan

Phone: +81 743 72 5216
Email: hiroa-ha@is.naist.jp

Tomohiro Ishihara
Univ. of Tokyo / WIDE Project
3-8-1 Komaba, Meguro
Tokyo,
Japan

Email: sho@c.u-tokyo.ac.jp

Osamu Nakamura
Keio Univ. / WIDE Project
5322 Endo
Kanagawa,
Japan

Email: osamu@wide.ad.jp

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 13, 2014

Q. Sun
Y. Cui
Tsinghua University
M. Siodelski
ISC
S. Krishnan
Ericsson
I. Farrer
Deutsche Telekom AG

June 11, 2014

DHCPv4 over DHCPv6 Transport
draft-ietf-dhc-dhcpv4-over-dhcpv6-09

Abstract

IPv4 connectivity is still needed as networks migrate towards IPv6. Users require IPv4 configuration even if the uplink to their service provider supports IPv6 only. This document describes a mechanism for obtaining IPv4 configuration information dynamically in IPv6 networks by carrying DHCPv4 messages over DHCPv6 transport. Two new DHCPv6 messages and two new DHCPv6 options are defined for this purpose.

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 December 13, 2014.

Copyright Notice

Copyright (c) 2014 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. Requirements Language 3
3. Terminology 3
4. Applicability 4
5. Architecture Overview 4
6. New DHCPv6 Messages 5
6.1. Message Types 6
6.2. Message Formats 6
6.3. DHCPv4-query Message Flags 7
6.4. DHCPv4-response Message Flags 7
7. New DHCPv6 Options 7
7.1. DHCPv4 Message Option Format 7
7.2. 4o6 Server Address Option Format 8
8. Use of the DHCPv4-query Unicast Flag 9
9. DHCP 4o6 Client Behavior 10
10. Relay Agent Behavior 12
11. DHCP 4o6 Server Behavior 12
12. Security Considerations 13
13. IANA Considerations 14
14. Contributors List 14
15. References 14
15.1. Normative References 14
15.2. Informative References 15
Authors' Addresses 15

1. Introduction

As the migration towards IPv6 continues, IPv6-only networks will become more prevalent. In such networks, IPv4 connectivity will continue to be provided as a service over IPv6-only networks. In addition to provisioning IPv4 addresses for clients of this service, other IPv4 configuration parameters may also be needed (e.g. addresses of IPv4-only services).

This document describes a transport mechanism to carry DHCPv4 messages using the DHCPv6 protocol for the dynamic provisioning of IPv4 addresses and other DHCPv4 specific configuration parameters across IPv6-only networks. It leverages the existing DHCPv4

infrastructure, e.g. failover, DNS updates, DHCP Leasequery, etc.

When IPv6 multicast is used to transport 4o6 messages, another benefit is that the operator can gain information about the underlying IPv6 network the 4o6 client is connected to from the the DHCPv6 relay agents the request has passed through.

2. Requirements Language

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].

3. Terminology

This document makes use of the following terms:

CPE:	Customer Premises Equipment (also known as Customer Provided Equipment), which provides access for devices connected to a Local Area Network (typically at the customer's site/home) to the Internet Service Provider's network.
DHCP 4o6 client (or client):	A DHCP client supporting both the DHCPv6 protocol [RFC3315] as well as the DHCPv4 over DHCPv6 protocol described in this document. Such a client is capable of requesting IPv6 configuration using DHCPv6 and IPv4 configuration using DHCPv4 over DHCPv6.
DHCP 4o6 server (or server):	A DHCP server that is capable of processing DHCPv4 packets encapsulated in the DHCPv4 Message option (defined below).
DHCPv4 over DHCPv6:	A protocol described in this document, used to carry DHCPv4 messages in the payload of DHCPv6 messages.

4. Applicability

The mechanism described in this document is not universally applicable. This is intended as a special-purpose mechanism that will be implemented on nodes that must obtain IPv4 configuration information using DHCPv4 in specific environments where native DHCPv4 is not available. Such nodes are expected to follow the advice in the "client behavior" section; nodes that do not require this functionality are expected not to implement it, or not to enable it by default. This mechanism may be enabled using an administrative control, or may be enabled automatically in accordance with the needs of some dual-stack transition mechanism such as [I-D.ietf-softwire-lw4over6]. Such mechanisms are beyond the scope of this document.

5. Architecture Overview

The architecture described here addresses a typical use case, where a DHCP client's uplink supports IPv6 only and the Service Provider's network supports IPv6 and limited IPv4 services. In this scenario, the client can only use the IPv6 network to access IPv4 services, so IPv4 services must be configured using IPv6 as the underlying network protocol.

Although the purpose of this document is to address the problem of communication between the DHCPv4 client and the DHCPv4 server, the mechanism that it describes does not restrict the transported messages types to DHCPv4 only. As the DHCPv4 message is a special type of BOOTP message, BOOTP messages [RFC0951] MAY also be transported using the same mechanism.

DHCP clients may be running on CPE devices, end hosts or any other device that supports the DHCP client function. This document uses the CPE as an example for describing the mechanism. This does not preclude any end-host, or other device requiring IPv4 configuration, from implementing DHCPv4 over DHCPv6 in the future.

This mechanism works by carrying DHCPv4 messages encapsulated within the newly defined DHCPv6 messages. The DHCPv6 relay encapsulation is used solely to deliver DHCPv4 packets to a DHCPv4-capable server, and do not allocate any IPv6 addresses nor provide IPv6 configuration information to the client. Figure 1, below, illustrates one possible deployment architecture of this mechanism.

The DHCP 4o6 client implements a new DHCPv6 message called DHCPv4-query, which contains a new option called the DHCPv4 Message option encapsulating a DHCPv4 message sent by the client. The format of this option is described in Section 7.1.

The DHCPv6 message can be transmitted either via DHCPv6 Relay Agents or directly to the DHCP 4o6 server. The server replies with a DHCPv4-response message, which is a new DHCPv6 message carrying the DHCPv4 response encapsulated in the DHCPv4 Message option.

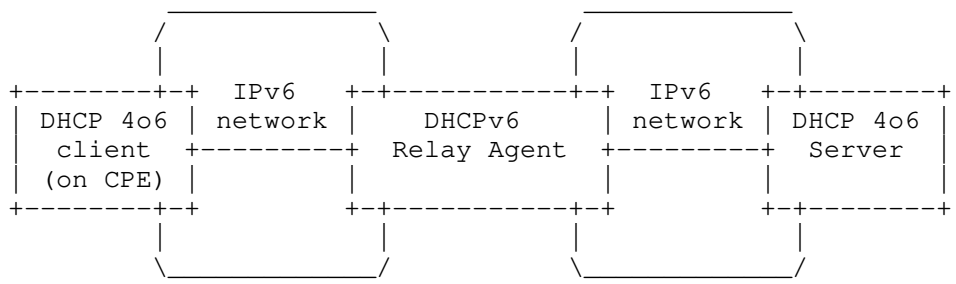


Figure 1: Architecture Overview

Before the client can use DHCPv4 over DHCPv6, it MUST obtain the necessary IPv6 configuration. The client requests the 4o6 Server Address option from the server by sending the option code in Option Request option as described in [RFC3315]. If the server responds with the 4o6 Server Address option, it is an indication to the client to attempt using DHCPv4 over DHCPv6 to obtain IPv4 configuration. Otherwise, the client MUST NOT use DHCPv4 over DHCPv6 to request IPv4 configuration.

The client obtains the address(es) of the DHCP 4o6 server(s) from the 4o6 Server Address option and uses them to communicate with the DHCP 4o6 servers as described in Section 9. If the 4o6 Server Address option contains no addresses (is empty), the client uses the well-known All_DHCP_Relay_Agents_and_Servers multicast address to communicate with the DHCP 4o6 server(s).

Before applying for an IPv4 address via a DHCPv4-query message, the client must identify a suitable network interface for the address. Once the request is acknowledged by the server, the client can configure the address and other relevant parameters on this interface. The mechanism for determining a suitable interface is out of the scope of the document.

6. New DHCPv6 Messages

Two new DHCPv6 messages carry DHCPv4 messages between the client and the server using the DHCPv6 protocol: DHCPv4-query and DHCPv4-response. This section describes the structures of these messages.

6.1. Message Types

DHCPV4-QUERY (TBD): The DHCP 4o6 client sends a DHCPv4-query message to a DHCP 4o6 server. The DHCPv4 Message option carried by this message contains a DHCPv4 message that the DHCP 4o6 client uses to request IPv4 configuration parameters from the server.

DHCPv4-RESPONSE (TBD): A DHCP 4o6 server sends a DHCPv4-response message to a DHCP 4o6 client. It contains a DHCPv4 Message option carrying a DHCPv4 message in response to a DHCPv4 message received by the server in the DHCPv4 Message option of the DHCPv4-query message.

6.2. Message Formats

Both DHCPv6 messages defined in this document share the following format:

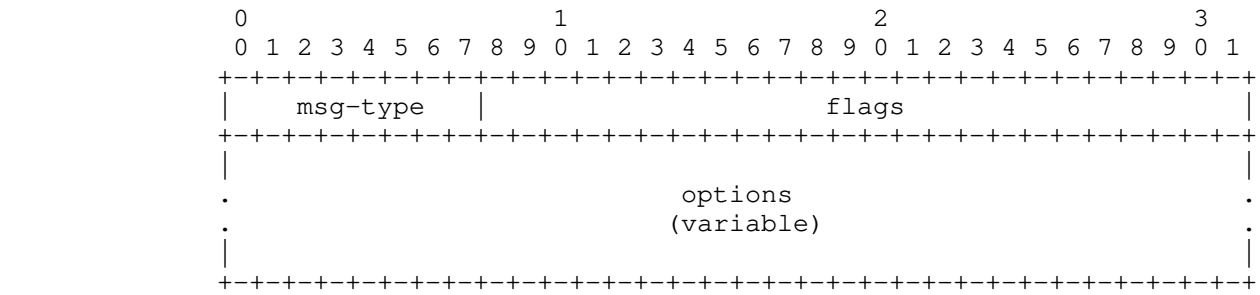


Figure 2: The format of DHCPv4-query and DHCPv4-response messages

msg-type Identifies the message type. It can be either DHCPV4-QUERY (TBD) or DHCPV4-RESPONSE (TBD) corresponding to the contained DHCPv4-query or DHCPv4-response, respectively.

flags Specifies flags providing additional information required by the server to process the DHCPv4 message encapsulated in the DHCPv4-query message, or required by the client to process a DHCPv4 message encapsulated in the DHCPv4-response message.

options Options carried by the message. The DHCPv4 Message Option (described in Section 7.1) MUST be carried by the message. Only DHCPv6 options for IPv4

configuration may be included in this field. It MUST NOT contain DHCPv6 options related solely to IPv6, or IPv6-only service configuration.

6.3. DHCPv4-query Message Flags

The "flags" field of the DHCPv4-query is used to carry additional information that may be used by the server to process the encapsulated DHCPv4 message. Currently only one bit of this field is used. Remaining bits are reserved for the future use. The "flags" field has the following format:

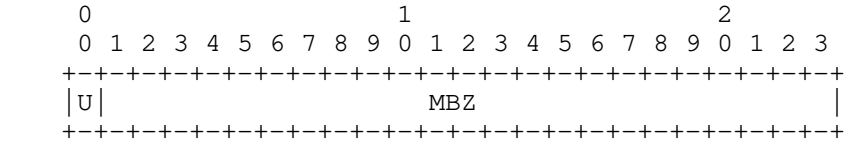


Figure 3: DHCPv4-query flags format

U Unicast Flag. If set to 1, it indicates that the DHCPv4 message encapsulated within the DHCPv4-query message would be sent to a unicast address if it was sent using IPv4. If this flag is set to 0, it indicates that the DHCPv4 message would be sent to the broadcast address if it was sent using IPv4. The usage of the flag is described in detail in Section 8.

MBZ Bits MUST be set to zero when sending and MUST be ignored when receiving.

6.4. DHCPv4-response Message Flags

This document introduces no flags to be carried in the "flags" field of the DHCPv4-response message. They are all reserved for the future use. The DHCP 4o6 server MUST set all bits of this field to 0 and the DHCP 4o6 client MUST ignore the content in this field.

7. New DHCPv6 Options

7.1. DHCPv4 Message Option Format

The DHCPv4 Message option carries a DHCPv4 message that is sent by the client or the server. Such messages exclude any IP or UDP headers.

The format of the DHCPv4 Message option is:

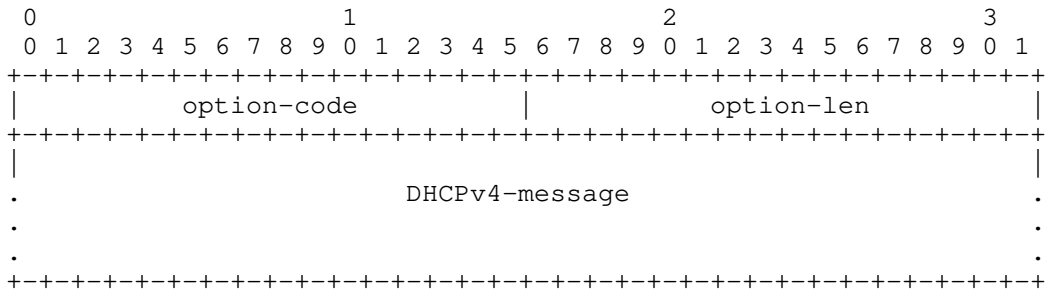


Figure 4: DHCPv4 Message option Format

option-code OPTION_DHCPV4_MSG (TBD).

option-len Length of the DHCPv4 message.

DHCPv4-message The DHCPv4 message sent by the client or the server. In a DHCPv4-query message it contains a DHCPv4 message sent by a client. In a DHCPv4-response message it contains a DHCPv4 message sent by a server in response to a client.

7.2. 4o6 Server Address Option Format

The 4o6 Server Address option is sent by a server to a client requesting IPv6 configuration using DHCPv6 [RFC3315]. It carries a list of DHCP 4o6 servers' IPv6 addresses that the client should contact to obtain IPv4 configuration. This list may include multicast and unicast addresses. The client sends its requests to all unique addresses carried in this option.

This option may also carry no IPv6 addresses, which instructs the client to use the All_DHCP_Relay_Agents_and_Servers multicast address as the destination address.

The presence of this option in the server's response indicates to the client that it should use DHCPv4 over DHCPv6 to obtain IPv4 configuration. If the option is absent, the client MUST NOT enable DHCPv4-over-DHCPv6 function.

The format of the 4o6 Server Address option is:

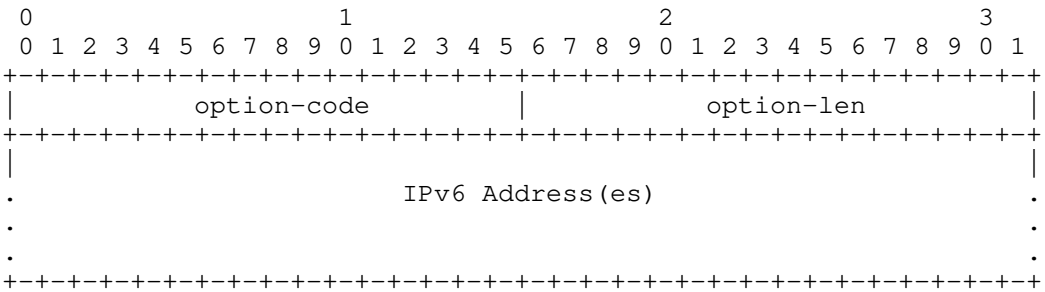


Figure 5: 4o6 Servers Address Option Format

option-code OPTION_DHCP4_O_DHCP6_SERVER (TBD).

option-len Length of the IPv6 address(es) carried by the option, i.e. multiple of 16 octets. Minimal length of this option is 0.

IPv6 Address Zero or more IPv6 addresses of the DHCP 4o6 Server(s).

8. Use of the DHCPv4-query Unicast Flag

A DHCPv4 client conforming to [RFC2131] may send its DHCPREQUEST message to either a broadcast or unicast address depending on its state. For example, a client in the RENEWING state uses a unicast address to contact the DHCPv4 server to renew its lease. A client in the REBINDING state uses a broadcast address.

In DHCPv4 over DHCPv6, IPv6 is used to deliver DHCPv4 messages to the DHCP 4o6 server. There is no relation between the outer IPv6 address and the inner DHCPv4 message. As a result, the server is unable to determine whether the received DHCPv4 messages should have been sent using broadcast or unicast in IPv4 by checking the IPv6 address.

In order to allow the server to determine the client's state, the "Unicast" flag is carried in the DHCPv4-query message. The client MUST set this flag to 1 when the DHCPv4 message would have been sent to the unicast address if using DHCPv4 over IPv4. This flag MUST be set to 0 if the DHCPv4 client would have sent the message to the broadcast address in IPv4. The choice whether a given message should be sent to a broadcast or unicast address is made based on the [RFC2131] and its extensions.

Note: The "Unicast" flag reflects how the DHCPv4 packet would have been sent; not how the DHCPv6 packet itself is sent.

9. DHCP 4o6 Client Behavior

The client MUST obtain necessary IPv6 configuration from a DHCPv6 server before using DHCPv4 over DHCPv6. The client requests the 4o6 Server Address option using Option Request option (ORO) in every Solicit, Request, Renew, Rebind and Information-request message. If the DHCPv6 server includes the 4o6 Server Address option in its response, it is an indication that the client can use DHCPv4 over DHCPv6 to obtain the IPv4 configuration (by sending DHCPv4 messages encapsulated in DHCPv4-query messages).

The client MUST NOT use DHCPv4 over DHCPv6 to request IPv4 configuration if the DHCPv6 server does not include the 4o6 Server Address option. If the IPv6 configuration that contained the 4o6 Server Address option subsequently expires, or if the renewed IPv6 configuration does not contain the 4o6 Server Address option, the client MUST stop using DHCPv4 over DHCPv6 to request or renew IPv4 configuration. However, the client continues to request 4o6 Server Address option in the messages sent to the DHCPv6 server as long as it desires to use DHCPv4 over DHCPv6.

It is possible in a multi-homed configuration for there to be more than one DHCPv6 configuration active at the same time that contains a 4o6 Server Address option. In this case, the configurations are treated as being independent, so that when any such configuration is active, a DHCPv4-over-DHCPv6 function may be enabled for that configuration.

An implementation may also treat such configurations as being exclusive, such that only one is kept active at a time. In this case, the client keeps the same configuration active continuously as long as it is valid. If that configuration becomes invalid but one or more other configurations remain valid, the client activates one of the remaining valid configurations.

Which strategy to follow is dependent on the implementation: keeping multiple configurations active at the same time may provide useful redundancy in some applications, but may be needlessly complex in other cases.

If the client receives the 4o6 Server Address option and DHCPv4 [RFC2131] is used on the interface over which the DHCPv6 option was received, the client MUST stop using the IPv4 configuration received using DHCPv4 on this interface. The client MAY send a DHCPRELEASE to the DHCPv4 server to relinquish an existing lease as described in [RFC2131] in section 4.4.6. The client MUST NOT use DHCPv4 on this interface as long as it receives 4o6 Server Address option in the messages received from the DHCPv6 server.

If the client receives a 4o6 Server Address option that contains no IP addresses, i.e. the option is empty, the client MUST send its requests to the All_DHCP_Relay_Agents_and_Servers multicast address. If there is a list of IP addresses in the option, the client SHOULD send requests to each unique address carried by the option.

If the client obtained stateless IPv6 configuration by sending Information-request message to the server, the client MUST follow the rules in [RFC4242] to periodically refresh the DHCPv4-over-DHCPv6 configuration (i.e. list of DHCP 4o6 servers) as well as other configuration data. The client which obtained stateful IPv6 configuration will refresh the status of DHCPv4-over-DHCPv6 function when extending a lifetime of acquired IPv6 address (Renew and Rebind messages).

The client MUST employ an IPv6 address of an appropriate scope to source the DHCPv4-query message from. When the client sends a DHCPv4-query message to the multicast address, it MUST use a link-local address as the source address as described in [RFC3315]. When the client sends a DHCPv4-query message using unicast, the source address MUST be an address of appropriate scope, acquired in advance.

The client generates a DHCPv4 message and stores it verbatim in the DHCPv4 Message option carried by the DHCPv4-query message. The client MUST put exactly one DHCPv4 Message option into a single DHCPv4-query message. The client MUST NOT request the 4o6 Server Address option in the DHCPv4-query message.

The client MUST follow rules defined in Section 8 when setting the Unicast flag based on the DHCPv4 destination.

On receiving a DHCPv4-response message, the client MUST look for the DHCPv4 Message option within this message. If this option is not found, the DHCPv4-response message is discarded. If the DHCPv4 Message option is present, the client extracts the DHCPv4 message it contains and processes it as described in section 4.4 of [RFC2131].

When dealing with IPv4 configuration, the client MUST follow the normal DHCPv4 retransmission requirements and strategy as specified in section 4.1 of [RFC2131]. There are no explicit transmission parameters associated with a DHCPv4-query message, as this is governed by the DHCPv4 [RFC2131] "state machine".

The client MUST implement [RFC4361] to ensure that the device correctly identifies itself. It MUST send a 'client identifier' option when using DHCPv4 over DHCPv6.

10. Relay Agent Behavior

When a DHCPv6 relay agent receives a DHCPv4-query message, it may not recognize this message. The unknown message MUST be forwarded as described in [I-D.ietf-dhc-dhcpv6-unknown-msg].

If it recognises the message, the DHCPv6 relay agent MAY allow the configuration of a dedicated DHCPv4 over DHCPv6 specific destination address(es), differing from the address(es) of the DHCPv6-only server(s). To implement this function, the relay checks the received DHCPv6 message type and forwards according to the following logic:

1. If the message type is DHCPV4-QUERY, the packet is relayed to the configured DHCP 4o6 Server's address(es) in the form of normal DHCPv6 packet (i.e. DHCPv6/UDP/IPv6).

2. For any other DHCPv6 message type, forward according to section 20 of [RFC3315].

The above logic only allows for separate relay destinations configured on the relay agent closest to the client (single relay hop). Multiple relaying hops are not considered in the case of separate relay destinations.

11. DHCP 4o6 Server Behavior

When the server receives a DHCPv4-query message from a client, it searches for the DHCPv4 Message option. The server discards a packet without this option. In addition, the server MAY notify an administrator about the receipt of this malformed packet. The mechanism for this notification is out of scope for this document.

If the server finds a valid DHCPv4 Message option, it extracts the original DHCPv4 message. Since the DHCPv4 message is encapsulated in the DHCPv6 message, it lacks the information which is typically used by the DHCPv4 server, implementing [RFC2131], to make address allocation decisions, e.g. giaddr for relayed messages and IPv4 address of the interface which the server is using to communicate with directly connected client. Therefore, the DHCP 4o6 server allocates addresses according to the local address assignment policies determined by the server administrator. For example, if the DHCPv4-query message has been sent via a relay, the server MAY use the link-address field of the Relay-forward message as a lookup for the IPv4 subnet to assign DHCPv4 address from. If the DHCPv4-query message has been sent from a directly connected client, the server MAY use IPv6 source address of the message to determine the appropriate IPv4 subnet to use for DHCPv4 address assignment.

Alternatively, the server may act as a DHCPv4 relay agent and forward the DHCPv4 packet to a "normal" DHCPv4 server. The details of such a solution have not been considered by the working group; describing that solution is out of scope of this document and is left as future work should the need for it arise.

The server SHOULD use the "flags" field of the DHCPv4-query message to create a response (server to client DHCPv4 message). The use of this field is described in detail in Section 8.

When an appropriate DHCPv4 response is created, the server places it in the payload of a DHCPv4 Message option, which it puts into the DHCPv4-response message.

If the DHCPv4-query message was received directly by the server, the DHCPv4-response message MUST be unicast from the interface on which the original message was received.

If the DHCPv4-query message was received in a Relay-forward message, the server creates a Relay-reply message with the DHCPv4-response message in the payload of a Relay Message option, and responds as described in section 20.3 of [RFC3315].

12. Security Considerations

In this specification, DHCPv4 messages are encapsulated in the newly defined option and messages. This is similar to the handling of the current relay agent messages. In order to bypass firewalls or network authentication gateways, a malicious attacker may leverage this feature to convey other messages using DHCPv6, i.e. use DHCPv6 as a form of encapsulation. However, the potential risk from this is no more severe than that with the current DHCPv4 and DHCPv6 practice.

It is possible for a rogue server to reply with a 4o6 Server Address Option containing duplicated IPv6 addresses, which could cause an amplification attack. To avoid this, the client MUST check if there are duplicate IPv6 addresses in a 4o6 Server Address Option when receiving one. The client MUST ignore any but the first instance of each address.

When considering whether to enable DHCPv4-over-DHCPv6, one important consideration is that when it is enabled, this gives the DHCPv6 server the ability to shut off DHCPv4 traffic, and, consequently, IPv4 traffic, on the interface that is configured to do DHCPv4-over-DHCPv6. For this reason, DHCPv4-over-DHCPv6 should only be enabled in situations where there is a clear trust relationship that eliminates this concern. For instance, a CPE device can safely enable this on its WAN interface, because it is reasonable to assume

that an ISP will not accidentally configure DHCPv4 over DHCPv6 service on that link, and that it will be impractical for an attacker to set up a rogue DHCPv6 server in the ISP's network.

13. IANA Considerations

IANA is requested to allocate two DHCPv6 option codes for use by OPTION_DHCPV4_MSG and OPTION_DHCP4_O_DHCP6_SERVER from the "Option Codes" table, and two DHCPv6 message type codes for the DHCPV4-QUERY and DHCPV4-RESPONSE from the "Message Types" table of the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Registry. Both tables can be found at http://www.iana.org/assignments/dhcpv6-parameters/.

14. Contributors List

Many thanks to Ted Lemon, Bernie Volz, Tomek Mrugalski, Cong Liu and Yuchi Chen, for their great contributions to the specification.

15. References

15.1. Normative References

[I-D.ietf-dhc-dhcpv6-unknown-msg]
Cui, Y., Sun, Q., and T. Lemon, "Handling Unknown DHCPv6 Messages", draft-ietf-dhc-dhcpv6-unknown-msg-08 (work in progress), March 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.

[RFC4361] Lemon, T. and B. Sommerfeld, "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)", RFC 4361, February 2006.

15.2. Informative References

[I-D.ietf-softwire-lw4over6]
Cui, Y., Qiong, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-ietf-softwire-lw4over6-10 (work in progress), June 2014.

[RFC0951] Croft, B. and J. Gilmore, "Bootstrap Protocol", RFC 951, September 1985.

Authors' Addresses

Qi Sun
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: sunqi@csnet1.cs.tsinghua.edu.cn

Yong Cui
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Marcin Siodelski
950 Charter Street
Redwood City, CA 94063
USA

Phone: +1 650 423 1431
Email: msiodelski@gmail.com

Suresh Krishnan
Ericsson

Email: suresh.krishnan@ericsson.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4,Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de

DHC WG
Internet-Draft
Intended status: Informational
Expires: January 2, 2015

B. Rajtar
Hrvatski Telekom
I. Farrer
Deutsche Telekom AG
July 01, 2014

Provisioning IPv4 Configuration Over IPv6 Only Networks
draft-ietf-dhc-v4configuration-06

Abstract

As IPv6 becomes more widely adopted, some service providers are choosing to deploy IPv6 only networks without dual-stack functionality for IPv4. However, as access to IPv4 based services will continue to be a requirement for the foreseeable future, IPv4 over IPv6 mechanisms, such as softwire tunnels are being developed.

In order to provision end-user's hosts with the IPv4 configuration necessary for such mechanisms, a number of different approaches have been proposed. This memo discusses each of the proposals, identifies the benefits and drawbacks and recommends approaches to be used as the basis for future deployment and development.

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 2, 2015.

Copyright Notice

Copyright (c) 2014 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. Overview of IPv4 Parameter Configuration Approaches . . . 4
1.2. DHCPv4o6 Based Provisioning – Functional Overview 4
1.3. DHCPv6 Based Provisioning – Functional Overview 6
1.4. DHCPv6 + Stateless DHCPv4oSW Based Provisioning – Functional Overview 6
1.5. DHCPv4oDHCPv6 Based Provisioning – Functional Overview . 7
2. Requirements for the Solution Evaluation 8
3. Comparison of the Four Approaches 9
3.1. DHCPv4o6 Based Provisioning 9
3.1.1. Pros 9
3.1.2. Cons 10
3.2. DHCPv6 Based Provisioning 10
3.2.1. Pros 10
3.2.2. Cons 10
3.3. DHCPv6 + Stateless DHCPv4oSW Based Provisioning 11
3.3.1. Pros 11
3.3.2. Cons 11
3.4. DHCPv4oDHCPv6 Based Provisioning 12
3.4.1. Pros 12
3.4.2. Cons 12
4. Conclusion 13
5. Transporting Unmodified DHCPv4 Messages over an IPv6 Link Layer 13
5.1. Combined Hub and DHCPv4 Relay Required Functionality . . 14
6. IANA Considerations 14
7. Security Considerations 15
7.1. DHCPv4oIPv6 15
7.2. DHCPv6 15
7.3. DHCPv6+DHCPv4oSW 15
7.4. DHCPv4oDHCPv6 15
8. Acknowledgements 15
9. Informative References 15
Authors' Addresses 17

1. Introduction

A service provider with an IPv6-only network must also be able to provide customers with access to the IPv4 Internet and other IPv4-only services. IPv4 over IPv6 tunneling / translation mechanisms are an obvious example of this, such as the ones described in:

- [I-D.ietf-softwire-lw4over6]
- [I-D.ietf-softwire-map]
- [I-D.ietf-softwire-map-t]

In today's home networks, each residential user is allocated a single global IPv4 address which is used for NAT44. Decentralizing NAT44 allows for much better scaling and, when combined with stateless network functions, can simplify redundancy and logging when compared to centralized Carrier Grade NAT architectures. This results in the need to provision a number of configuration parameters to the CPE, such as the external public IPv4 address and a restricted port-range to use for NAT. Other parameters may also be necessary, depending on the underlying transport technology that is in use. In IPv4 only networks, DHCPv4 has often been used to provide IPv4 configuration, but in an IPv6 only network, DHCPv4 messages cannot be transported natively without either IPv6 encapsulation or translation.

DHCPv4 messages can be transported, unmodified, over a broadcast capable link-layer, depending on the underlying IPv4 in IPv6 technology, network topology and DHCPv4 client capabilities. A functional description of how unmodified DHCPv4 can be used is provided in Section 5. This approach is recommended for service providers whose network and clients can support this DHCPv4 architecture.

For the most simple IPv4 provisioning case, where the client only needs to receive a static IPv4 address assignment (with no dynamic address leasing or additional IPv4 configuration), a DHCPv6 based approach (e.g. [I-D.ietf-softwire-map-dhcp]) may provide a suitable solution.

This document is concerned with more complex IPv4 configuration scenarios, to bring IPv4 configuration over IPv6-only networks in line with the functionality offered by DHCPv4 in IPv4 native networks. DHCPv4 options may also need to be conveyed to clients for configuring IPv4 based services, e.g., SIP server addresses.

Although IPv4-in-IPv6 softwire tunnel and translation clients are currently the only use-case for DHCP based configuration of IPv4 parameters in IPv6 only networks, a suitable IPv4 provisioning solution should not be limited to only supporting the configuration of softwires, or be bound to specific IPv4 over IPv6 architectures or mechanisms. The solution needs to be flexible enough to support new IPv4 over IPv6 technologies as they are developed.

This document describes and compares four different methods which have been proposed as solutions to this problem.

1.1. Overview of IPv4 Parameter Configuration Approaches

The following approaches for transporting IPv4 configuration parameters over IPv6 only networks have been suggested:

1. Adapt DHCPv4 format messages to be transported over IPv6 as described in [I-D.ietf-dhc-dhcpv4-over-ipv6]. For brevity, this is referred to as DHCPv4o6.

2. Extend DHCPv6 to support IPv4 address leasing and other DHCPv4 options.

3. Use DHCPv6 for external IPv4 address and source port configuration (e.g. [I-D.ietf-softwire-map-dhcp]. Use DHCPv4 over IPv4 messages within an IPv6 softwire for configuring additional parameters. This is referred to as DHCPv6 + Stateless DHCPv4oSW.

4. Use DHCPv4 format messages, transporting them within a new DHCPv6 message type as described in [I-D.ietf-dhc-dhcpv4-over-dhcpv6]. This is referred to as DHCPv4oDHCPv6.

At the time of writing, working examples of all but the third method have been developed and successfully tested in several different operators networks.

The following sections describe each of the approaches in more detail.

1.2. DHCPv4o6 Based Provisioning - Functional Overview

In order to receive IPv4 configuration parameters, IPv4-only clients initiate and exchange DHCPv4 messages with the DHCPv4 server. To adapt this for an IPv6-only network, an existing DHCPv4 client implements a Host Client Relay Agent (HCRA) function, which takes DHCPv4 messages and puts them into UDP and IPv6.

As the mechanism involves unicast IPv6 based communications, the IPv6 address of the server must be provisioned to the client. A DHCPv6 option for provisioning clients with this address is described in [I-D.mrugalski-softwire-dhcpv4-over-v6-option].

The IPv6 Transport Server (TSV) provides an IPv6 interface to the client. This interface may be implemented directly on the server and/or via an intermediary 'Transport Relay Agent' (TRA) device which acts as the gateway between the IPv4 and IPv6 domains.

For the dynamic allocation of IPv4 addresses, the DHCPv4 server function needs to be extended to add DHCPv4o6 TSV capabilities, such as the storing the IPv6 address of DHCPv4o6 clients and implementing the CRA6ADDR option.

This approach currently uses functional elements for ingress and egress of the IPv6-only transport domain - the HCRA on the host and the TRA or TSV on the server. As a result, this has sometimes been referred to as a tunneling approach. However, relay agent encapsulation is not a tunnel, since it carries only DHCP traffic; it would be more accurate to describe it as an encapsulation based transport.

[I-D.ietf-dhc-dhcpv4-over-ipv6] also defines an On-Link Client Relay Agent (LCRA), which is a Client Relay Agent located on the same link as an unmodified DHCPv4 client. It is worth noting that there is no technical reason for using relay encapsulation for DHCPv4o6; this approach was taken because the authors of the draft originally imagined that it might be used to provide configuration information for an unmodified DHCPv4 client. However, this turns out not to be a viable approach: in order for this to work, there would have to be IPv4 routing on the local link to which the client is connected. In that case, there's no need for DHCPv4o6.

Given that this is the case, there is no technical reason why DHCPv4o6 can't simply use the IPv6 transport directly, without any relay encapsulation. This would greatly simplify the specification and the implementation, and would still address the requirements stated in this document.

[I-D.ietf-dhc-dhcpv4-over-ipv6] describes this solution in detail.

The protocol stack for provisioning IPv4/IPv6 tunneling and translation mechanisms is as follows:

DHCPv4/UDP/IPv6

1.3. DHCPv6 Based Provisioning – Functional Overview

In this approach, DHCPv6 [RFC3315] would be extended with new DHCPv6 options for configuring all IPv4 based services and functions (i.e. IPv4 address assignment and any necessary DHCPv4 options). DHCPv4 options needed by IPv4 clients connected to the IPv6 network are updated as new DHCPv6 native options carrying IPv4 configuration parameters. IPv4 address leasing would also need to be managed by the DHCPv6 server.

At the time of writing, it is not known which or how many such options would need to be ported from DHCPv4 to DHCPv6.

The protocol stack for provisioning IPv4/IPv6 tunneling and translation mechanisms is as follows:

DHCPv6/UDP/IPv6

1.4. DHCPv6 + Stateless DHCPv4oSW Based Provisioning – Functional Overview

In this approach, configuration of the IPv4 address and source ports (if required) is carried out using DHCPv6, e.g. using [I-D.ietf-softwire-map-dhcp]. Any additional IPv4 configuration parameters that are required are then provisioned using DHCPv4 messages transported, within IPv6, through the configured softwire in the same manner as any other IPv4 based traffic. Broadcast based DHCPv4 DHCPDISCOVER messages (necessary for IPv4 address assignment) can not be transported as some softwire mechanisms implement NBMA links, where broadcast isn't supported. Additionally, there is a more general issue with the use of fixed L4 ports in A+P [RFC6346] based approaches. Here, a single IPv4 address is shared among multiple users, each using a unique set of ports for differentiation meaning that it is not possible for every client to be allocated a fixed L4 within its unique port set.

On receipt by the tunnel concentrator (e.g. MAP Border Router or a Lightweight 4over6 lwAFTR), the DHCPv4 message is extracted from the IPv6 packet and forwarded to the DHCPv4 server in the same way as any other IPv4 forwarding plane packet is handled.

As the client is already configured with its external IPv4 address and source ports (using DHCPv6 or a well-known IPv4 address for DS-Lite clients), the messages exchanged between the DHCPv4 client and server would be strictly DHCPINFORM/DHCPACK messages. These can be used for conveying additional DHCPv4 based options.

For this approach to function, a mechanism for the DHCPv4 client to learn the IPv4 address of the DHCPv4 server is also required. This could be via a well-known IPv4 address for the DHCPv4 server, a DHCPv4 relay function within the tunnel concentrator or other methods.

From a transport perspective, the key difference between this method and DHCPv4o6 (described above) is the protocol stack. Here the DHCPv4 message is first put into UDP and IPv4 and then into the IPv6 softwire, instead of placing the DHCPv4 message directly into UDP and IPv6.

Currently, this approach is only theoretical and does not have a corresponding Internet Draft providing more detail.

For IPv4/IPv6 tunneling and translation mechanism, the protocol stack used for obtaining an IPv4 address and source ports (if required) is as follows:

DHCPv6/UDP/IPv6

For provisioning IPv4/IPv6 tunneling mechanisms, the protocol stack for obtaining additional IPv4 configuration is:

DHCPv4/UDP/IPv4

NB: The encapsulating IPv6 tunneling header is not shown as it is functionally a layer 2 header.

And for provisioning IPv4/IPv6 translation mechanisms:

DHCPv4/UDP/IPv6

1.5. DHCPv4oDHCPv6 Based Provisioning - Functional Overview

[I-D.ietf-dhc-dhcpv4-over-dhcpv6] describes transporting DHCPv4 messages within two new DHCPv6 messages types: DHCPV4-QUERY and DHCPV4-RESPONSE. These new messages types must be implemented in both the DHCPv4oDHCPv6 client and server.

In this approach, dynamic IPv4 addressing, and/or any additional IPv4 configuration, is provided using DHCPv4 messages carried (without IPv4/UDP headers) within a new OPTION_DHCPV4_MSG DHCPv6 option.

OPTION_DHCPV4_MSG enables the client and server to send BOOTP/DHCPv4 messages verbatim across the IPv6 network. When a DHCPv4oDHCPv6 server receives a DHCPv6 request containing OPTION_DHCPV4_MSG within a DHCPV4-QUERY message, it passes it to the DHCPv4 server engine.

Likewise, the DHCPv4 server place its DHCPv4 response in the payload of OPTION_DHCPV4_MSG and puts this into a DHCPV4-RESPONSE message.

DHCPv4 messages can be carried within DHCPv6 multicast messages, using the All_DHCP_Relay_Agents_and_Servers multicast address. These can be relayed in exactly the same way as any other DHCPv6 multicasted messages.

Optionally, DHCPv6 relays could be updated so that they forward the DHCPV4-QUERY message to a different destination address, allowing for the separation of DHCPv4 and DHCPv6 provisioning infrastructure.

If the DHCPv4oDHCPv6 client is provisioned with a unicast IPv6 address(es) for the server(s), then an entirely unicast message flow between the client and server is also possible without the need for relaying.

For provisioning IPv4/IPv6 tunneling and translation mechanisms, the protocol stack used for obtaining dynamic v4 addressing and/or additional IPv4 configuration is as follows:

DHCPv4/DHCPv6/UDP/IPv6

2. Requirements for the Solution Evaluation

The following requirements have been defined to evaluate the different approaches:

1. Minimize the amount of work necessary to implement the solution through re-use of existing standards and implementations as much as possible.

2. Provide a method of supporting all DHCPv4 options so that they can be utilized without the need for further standardization.

3. Allow for the dynamic leasing of IPv4 addresses to clients. This allows for more efficient use of limited IPv4 resources.

4. Enable the separation of IPv4 and IPv6 host configuration infrastructure, i.e. independent DHCPv4 and DHCPv6 server functions to restrict provisioning domains to the relevant protocol and allow the removal of IPv4 infrastructure in the future.

5. Avoid leaving legacy IPv4 options in DHCPv6.

6. Provide a flexible architecture to give operators the option of only deploying the functional elements necessary for their specific requirements.

7. Not be restricted to specific underlying IPv4 over IPv6 transport mechanisms or architectures. The solution needs to be flexible enough to support new IPv4 over IPv6 technologies as they are developed.

3. Comparison of the Four Approaches

The table below provides a comparative evaluation showing how the different approaches meet the solution requirements described above.

Req. No.	DHCPv4o6	DHCPv6	DHCPv6 + Stateless DHCPv4oSW	DHCPv4oDHCPv6
1	No	Yes	No	Yes
2	Yes	No	Yes	Yes
3	Yes	No	No	Yes
4	Yes	No	Yes	Yes
5	Yes	No	Yes	Yes
6	No	No	Yes	Yes
7	Yes	Yes	No	Yes

Table 1: Approach Comparison

The following sections of the document provide more detail on the pros and cons of each of the approaches.

3.1. DHCPv4o6 Based Provisioning

3.1.1. Pros

1. Implementation makes all existing DHCPv4 options available with no further ongoing development work necessary.

2. IPv4 and IPv6 based provisioning can be separated from each other if required, allowing flexibility in network design.

3. Easy to implement through minor adaptation of existing DHCPv4 client, relay and server code.

4. Suitable for dynamic IPv4 address leases where the IPv4 address lifetime is not linked to the lifetime of a DHCPv6 lease.

5. Implementations already exist, proving that the approach works.

3.1.2. Cons

1. More new functional elements required within the architecture (CRA, DHCPv4o6 server and optionally TRA) than are necessary in DHCPv6 based provisioning.

2. A new DHCPv6 option is necessary in order to provision the IPv6 address of the DHCPv4 server to the end device.

3. The DHCPv4 client host needs to be updated to implement the IPv6 encapsulation and decapsulation function (i.e., an HCRA). Otherwise a separate On-Link CRA (LCRA) functional element must be deployed.

4. A DHCPv4 server must be deployed and maintained.

5. The DHCPv4 server needs to be updated to implement new DHCPv4o6 functionality.

3.2. DHCPv6 Based Provisioning

3.2.1. Pros

1. No additional functional elements are required except the DHCPv6 client and server.

2. A single protocol is used to deliver configuration information for IPv4 and IPv6.

3. Single provisioning point for all configuration parameters.

3.2.2. Cons

1. Any required DHCPv4 options must be ported to DHCPv6, which will require re-development work for each option.

2. Means that DHCPv4 'legacy' options (which will be of decreasing relevance in the future) will remain in DHCPv6 for the lifetime of the protocol.

3. Each time that a DHCPv4 option is ported to DHCPv6, all clients, servers and possibly relays would need to be updated to implement the new option.

4. Architecture does not allow for the separation of IPv4 and IPv6 domains.

5. Does not provide a mechanism for dynamic IPv4 address leasing. The lifetime of the IPv4 address is linked to the lifetime of a DHCPv6 address lease (i.e. the IPv4 address can only be changed when a DHCPv6 RENEW/REBIND message is sent). To remove this interdependency, a new DHCPv4 lease management mechanism would need to be added to DHCPv6 (e.g. a new Identity Association solely for IPv4 address leasing).

3.3. DHCPv6 + Stateless DHCPv4oSW Based Provisioning

3.3.1. Pros

1. Once implemented, all existing DHCPv4 options will be available with no ongoing development work required.

2. Uses existing DHCPv4 and DHCPv6 architectures in order to provide IPv4 configuration in an IPv6 only environment.

3. If required, DHCPv4 and DHCPv6 based provisioning can be separated from each other, allowing flexibility in network design.

3.3.2. Cons

1. More new functional elements required than are necessary with DHCPv6 based provisioning.

2. IPv4 over IPv6 softwire approaches that distribute the NAT44 function to the CPE and allow for IP address sharing (MAP-E & LW4o6) forbid the use of reserved TCP/UDP ports (e.g. 0-1024). Every DHCPv4 client sharing the same address needs to have a UDP listener running on UDP port 68. To resolve this would require significant rework to either the softwire mechanisms and/or the DHCPv4 client implementation.

3. From the current specification, DHCPINFORM is not suitable for use over a softwire. Additional work, such as the development of 'shims' would be necessary.

4. The current DHCPINFORM specification has a number of unclear points, such as those described in [I-D.ietf-dhc-dhcpinform-clarify]. Substantial work would be required to resolve this.

5. Links the deployment of IPv4 configuration over IPv6 to a softwire implementation (e.g. requiring a softwire concentrator to act as a DHCPv4 relay). Whilst softwires are the only

application for this functionality at the moment, this may not be the case in the future, meaning another solution may be required.

6. A new mechanism must be defined in order to provide the DHCPv4 client with the IPv4 address of the DHCPv4 server so that unicast DHCPINFORM messages can be sent.

7. As only the DHCPINFORM/DHCPACK DHCPv4 message types are supported, dynamic IPv4 address leasing (using DHCPDISCOVER messages) cannot be used.

8. Restricted to underlying hub-and-spoke IPv4 over IPv6 architectures. The hub is necessary to locate the DHCPv4 relay function, as all traffic must pass through it. An underlying mesh architecture does not have such a location to deploy the relay function.

9. The approach is currently unproven. Although existing implementations may currently exist, the approach has not been demonstrated.

3.4. DHCPv4oDHCPv6 Based Provisioning

3.4.1. Pros

1. Once implemented, all existing DHCPv4 options will be available with no ongoing development work necessary.

2. Uses existing DHCPv4 and DHCPv6 architectures in order to provide IPv4 configuration in an IPv6 only environment.

3. If required, DHCPv4 and DHCPv6 based provisioning can be separated from each other, allowing flexibility in network design.

4. Suitable for the provisioning of dynamic IPv4 configuration as the existing DHCPv4 leasing mechanism can be used.

3.4.2. Cons

1. More new functional elements within the architecture than are necessary in DHCPv6 based provisioning.

2. DHCPv6 clients need to be updated to implement the new DHCPv6 message types (BOOTPREQUESTv6 and BOOTPREPLYv6).

3. The DHCPv6 server needs to be updated to implement the new DHCPv4oDHCPv6 message types and functionality.

4. The approach is currently unproven as no existing implementations exist.

4. Conclusion

Whilst all of the approaches described here will require some development work to realize, it is clear from the above analysis that the most sustainable approach capitalizes on existing DHCPv4 implementations and include them as new DHCPv6 message types. The main rationale for this is that it enables all of DHCPv4's existing options to be migrated for use over IPv6 in a single step.

Porting of all necessary DHCPv4 options to DHCPv6 would require ongoing development work, re-implementing existing DHCPv4 functionality in DHCPv6. This will result in having legacy DHCPv4 options in DHCPv6, which will no longer be useful once IPv4 is completely abandoned.

Therefore, the DHCPv6 approach is not appropriate for delivering IPv4 configuration parameters.

The dynamic leasing of IPv4 addresses is fundamental to the efficient use of remaining IPv4 resources. This will become increasingly important in the future, so a mechanism which supports this is necessary. DHCPv6 + Stateless DHCPv4oSW does not provide this function and so is not recommended.

The DHCPv4o6 approach requires a DHCPv4 server (with DHCPv4o6 functionality) for all deployment scenarios, even when DHCPv4 specific functionality (e.g. sending DHCPv4 options) is not required by the operator.

Therefore, this memo recommends DHCPv4oDHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6] as the best underlying approach for provisioning IPv4 parameters over an IPv6 only network.

5. Transporting Unmodified DHCPv4 Messages over an IPv6 Link Layer

DHCPv4 can be transported across a broadcast capable link layer, such as a softwire. Functionally, a DHCPv4 client operates on the link layer interface (e.g. the softwire tunnel interface). As the link layer must support broadcasts, DHCPDISCOVER and other broadcast DHCPv4 messages can be transported. The DHCPv4 message flow is then the same as described in section 3.1 of [RFC2131].

For an unmodified DHCPv4 client to function over an IPv6 native network, the underlying IPv4 over IPv6 architecture must be based on a point-to-point link between the client and a central point (i.e. a

hub or tunnel concentrator) which all client DHCPv4 broadcast messages will pass through. This hub must function as either the DHCPv4 server or a DHCPv4 relay. The relay forwards broadcast DHCPv4 DHCPDISCOVER/DHCPREQUEST messages to a separate DHCPv4 server.

5.1. Combined Hub and DHCPv4 Relay Required Functionality

When the DHCPv4 relay function is co-located with the IPv4 in IPv6 hub function, there are some implementation considerations and requirements that must be fulfilled. The following list describes these.

1. Depending on the underlying IPv4 over IPv6 mechanism that the hub is based upon, it may be necessary to modify the encapsulation/ decapsulation or IPv6/IPv4 translation packet validation policy so that IPv4 payload packets sourced from the unspecified address (0.0.0.0) are not dropped for broadcast DHCPv4 payload packets.

2. The DHCPv4 relay must use the DHCPv4 Relay Information Option (option 82) Relay-ID sub-option (2) to convey the client's source IPv6 address. This is used by the relay to route DHCPv4 response packets sent by the DHCPv4 server to the correct client.

3. For some IPv4 in IPv6 transition technologies, a client may be configured with an IPv4 address which is shared by other clients. In these cases, clients using a single IPv4 address are differentiated using the combination of the IPv4 address and a range of restricted layer 4 source ports unique to each client (used for NAPT). The DHCPv4 client L4 port (68) must not be provisioned to any client for NAPT use.

4. The DHCPv4 relay must implement the Server Identifier Override Sub-option described in [RFC5107] to direct all DHCPv4 messages through the DHCPv4 relay. As option 82 is being used to identify the destination IPv6 address for messages from the DHCPv4 server to the client, the L4 destination port is not required for the return path lookup process and is left unchanged as port 68.

6. IANA Considerations

This document does not make any request from IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

This document analyzes various solutions and doesn't introduce any new capabilities necessitating additional security considerations. The following sub-sections provide pointers to the documented security considerations associated with each approach.

7.1. DHCPv4oIPv6

Security considerations associated with this approach are described in Section 8 of [I-D.ietf-dhc-dhcpv4-over-ipv6].

7.2. DHCPv6

Security considerations associated with this approach are described in Section 23 of [RFC3315].

7.3. DHCPv6+DHCPv4oSW

There is currently no document describing this mechanism, so no security considerations have been documented.

7.4. DHCPv4oDHCPv6

Security considerations associated with this approach are described in [I-D.ietf-dhc-dhcpv4-over-dhcpv6].

8. Acknowledgements

Thanks to Ted Lemon, Tomek Mrugalski, Ole Troan, Bernie Volz and Francis Dupont for their input and reviews.

9. Informative References

[I-D.ietf-dhc-dhcpinform-clarify]
Hankins, D., "Dynamic Host Configuration Protocol DHCPINFORM Message Clarifications", draft-ietf-dhc-dhcpinform-clarify-06 (work in progress), October 2011.

[I-D.ietf-dhc-dhcpv4-over-dhcpv6]
Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4 over DHCPv6 Transport", draft-ietf-dhc-dhcpv4-over-dhcpv6-09 (work in progress), June 2014.

[I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., Lemon, T., and Q. Sun, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-09 (work in progress), April 2014.

[I-D.ietf-softwire-lw4over6]
Cui, Y., Qiong, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-ietf-softwire-lw4over6-10 (work in progress), June 2014.

[I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-10 (work in progress), January 2014.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-07 (work in progress), March 2014.

[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-05 (work in progress), February 2014.

[I-D.mrugalski-softwire-dhcpv4-over-v6-option]
Mrugalski, T. and P. Wu, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for DHCPv4 over IPv6 Endpoint", draft-mrugalski-softwire-dhcpv4-over-v6-option-01 (work in progress), September 2012.

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

[RFC5107] Johnson, R., Kumarasamy, J., Kinnear, K., and M. Stapp, "DHCP Server Identifier Override Suboption", RFC 5107, February 2008.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

Authors' Addresses

Branimir Rajtar
Hrvatski Telekom
Zagreb
Croatia

Email: branimir.rajtar@t.ht.hr

Ian Farrer
Deutsche Telekom AG
Bonn
Germany

Email: ian.farrer@telekom.de

ï»¿

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 sunsetting 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:

- Consideration of whether the specification can operate in an environment without IPv4.

- 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 LLMNR [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 choooosing 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 ritio. 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

DISPATCH
Internet-Draft
Intended status: Informational
Expires: January 10, 2014

C. Klatsky, Ed.
Comcast
O. Johansson
Edvina
R. Shekh-Yusef
Avaya
A. Hutton
Siemens Enterprise Communications
G. Salgueiro
Cisco Systems
July 09, 2013

Interoperability Impacts of IPv6 Interworking with Existing IPv4 SIP Implementations
draft-klatsky-dispatch-ipv6-impact-ipv4-01

Abstract

This document captures potential impacts to IPv4 SIP implementations when interworking with IPv6 SIP implementations. Although some amount of interworking translation will occur at the network and application layers, an IPv4 SIP application may still encounter a SIP message with some IPv6 values in it, resulting in unforeseen error conditions. Such potential scenarios will be identified in this document so that SIP application developers can define solutions to handle these cases. Note, this document is not intended to be an exhaustive list, rather to provide an overview of some of the more commonly encountered potential scenarios.

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 10, 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 3
2. Terminology and Conventions Used in This Document 3
3. Potential IPv4/IPv6 Interoperability Failure Scenarios . . . 4
3.1. IPv6 Address Handling in Via Headers 4
3.2. IPv6 Address Handling in Record-Route and Route Headers . 4
3.3. IPv6 Address Handling in From / To / Contact Headers . . 4
3.4. IPv6 Address Handling in SDP Body 4
3.5. IPv6 Address Handling in 'reginfo' XML Registration Information Document 5
3.6. IPv6 Address Handling in 30x Redirect 5
3.7. IPv6 Address Handling in REFER-based Transfer 5
3.8. DNS Resolution of IPv4/IPv6 in SRV Records 6
3.9. IPv6 Address Handling in Multiple Contact Registrations . 6
3.10. Unsupported Address 6
4. Security Considerations 6
5. IANA Considerations 6
6. Acknowledgements 6
7. References 7
7.1. Normative References 7
7.2. Informative References 7
Appendix A. Additional Guidelines 7
A.1. IPv6 Implementation Guidelines 8
A.2. IPv6/IPv4 Interworking Function: Avoid IPv6 address Leakage? 8
Authors' Addresses 9

1. Introduction

The continued proliferation of IPv6 infrastructure deployments has resulted in more IPv6 Session Initiation Protocol (SIP) User Agents (UAs) being turned up on the network. Considering the large deployed install base of IPv4 SIP UAs developed prior to the widespread deployment of IPv6, it is a well known fact that not all IPv4 SIP UAs have taken into account all possible IPv4 SIP-to-IPv6 SIP interoperability considerations at the time of their development. The scenarios outlined in this document are intended as guidance for application developers to help identify solutions to resolve the identified interoperability challenges.

2. Terminology and Conventions Used in This Document

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].

RFC 3261 [RFC3261] defines additional terms used in this document that are specific to the SIP domain such as "proxy"; "registrar"; "redirect server"; "user agent server" or "UAS"; "user agent client" or "UAC"; "back-to-back user agent" or "B2BUA"; "dialog"; "transaction"; "server transaction".

This document uses the term "SIP Server" that is defined to include the following SIP entities: user agent server, registrar, redirect server, a SIP proxy in the role of user agent server, and a B2BUA in the role of a user agent server.

This document also uses the following terminology to make clear distinction between SIP entities supporting only IPv4, only IPv6 or supporting both IPv4 and IPv6.

IPv4-only UA/UAC/UAS: An IPv4-only UA/UAC/UAS supports SIP signaling and media only on the IPv4 network. It does not understand IPv6 addresses.

IPv6-only UA/UAC/UAS: An IPv6-only UA/UAC/UAS supports SIP signaling and media only on the IPv6 network. It does not understand IPv4 addresses.

IPv4/IPv6 UA/UAC/UAS: A UA/UAC/UAS that supports SIP signaling and media on both IPv4 and IPv6 networks; such a UA/UAC/UAS is known (and will be referred to in this document) as a "dual-stack" [RFC4213] UA/UAC/UAS.

3. Potential IPv4/IPv6 Interoperability Failure Scenarios

3.1. IPv6 Address Handling in Via Headers

As an IPv6 SIP message makes its way through the network, the Via header is updated and includes specific IPv6 addresses of IPv6 nodes that it has traversed. If the message arrives at an IPv4-only UAS it may still contain those IPv6 addresses in the Via header. Presumably the topmost Via header references an IPv4 address or a Fully Qualified Domain Name (FQDN) resolvable to any IPv4 address. In this case the IPv4-only UAS is able to send its response on to its next hop, otherwise the message would not have made it to the IPv4-only UA at all. The challenge for the IPv4-only UA then becomes to not generate an error even if the other Via headers that it does not need to act upon contain IPv6 addresses.

3.2. IPv6 Address Handling in Record-Route and Route Headers

Similar to the concerns of having IPv6 addresses in the Via headers, IPv4 SIP UAS may also encounter Record-Route headers that contain IPv6 addresses of IPv6 nodes the SIP message has traversed. It is again assumed that if the SIP message arrives at an IPv4-only UA that the topmost Record-Route header references an IPv4 address or a FQDN resolvable to any IPv4 address, such that the response may be routed back to a node reachable by the IPv4-only UAS. In this instance the IPv4-only UA should not generate an error when parsing the IPv6 addresses. Additionally, the IPv4-only UA may also need to populate the Route header in the response that includes the IPv6 addresses learned from previously received Record-Route header, and again do so without generating an error.

3.3. IPv6 Address Handling in From / To / Contact Headers

Another scenario with possible IPv6-to-IPv4 interoperability implications is the case where the IPv4-only UAS receives an IPv6 address in the Contact header and no Record-Route header. Since this represents the peer's reachable contact IP, it may not have been modified by any interworking element in the communications path. The IPv4-only UAS will have to send its requests through its outbound SIP server, and not generate an error upon receipt of a message with this IPv6 information.

In addition, using an IP address instead of domain in To and/or From headers may impact communication, as the From header is used for other communication sessions or added to a phone book.

3.4. IPv6 Address Handling in SDP Body

IPv4-only UASs may also receive SDP offers with IPv6 addresses in the Session Description Protocol (SDP) [RFC4566] portion of the message. An IPv6 address can appear in multiple places in the SDP, such as the o= line, c= line or a= lines (for Interactive Connectivity Establishment (ICE) [RFC5245] attributes). A working assumption is that minimally the c= line will reference an IPv4 address of a media interworking element to allow the media communications being established by this session to work. Nonetheless the IPv4-only UAS needs be aware and properly handle any IPv6 addresses that may be within the received SDP.

3.5. IPv6 Address Handling in 'reginfo' XML Registration Information Document

There may be instances where an IPv4-only UAC subscribes to the registration event package [RFC3680] as a "watcher" for a specific entity, to be informed of registration state changes for that entity. The "watcher" may have no knowledge of the IP address family in use on the "watched" entity, and it is possible that a NOTIFY indicating an IPv6 address in the Extensible Markup Language (XML) [XML] body is received. The "watcher" needs to properly parse such a NOTIFY and provide the status update of the "watched" entity to the user or system that requested the information.

3.6. IPv6 Address Handling in 30x Redirect

There may be scenarios where an IPv4-only UAC receives a 30x redirect message in response to a request it has sent. This 30x message may contain a Contact header with an IPv6 address. This is the case where the call is being redirected to an IPv6-only UAS. Since this represents the peer's reachable contact IP, it may not have been modified by any interworking element in the communications path.

If the UAC has a configured outbound proxy the new call will be setup to that proxy. If that proxy is not dual stack, the call will fail. If there's no outbound proxy configured, the call will fail. If the UAC is a soft phone or hard phone, an error message should be displayed.

3.7. IPv6 Address Handling in REFER-based Transfer

After establishing a call between two IPv4-only UAs, one of the parties in the call may attempt to transfer the other party to a 3rd party using the REFER method [RFC5589]. This transfer may be to an IPv6-only UAS. The implication is that both IPv4-only UASs involved in the call transfer need to be able to handle a REFER with an IPv6 address in the Refer-To header. The transferor needs to be able to form the proper REFER message with the IPv6 Contact and the

transferee needs to be able to process the REFER message and attempt to establish a call with the transfer target.

3.8. DNS Resolution of IPv4/IPv6 in SRV Records

A dual-stack UA may use the Domain Name System (DNS) SRV mechanism to resolve addresses of proxies that it needs to communicate with. In such a case it needs to be able to locate both IPv4 proxies and IPv6 proxies. This implies that the DNS server has been updated with both A and AAAA records for the SIP server, and that the dual-stack UA requests for both IPv4 and IPv6 SIP server addresses.

3.9. IPv6 Address Handling in Multiple Contact Registrations

A 200 OK to a REGISTER request might include multiple Contact headers because the user has registered his or her Address of Record (AOR) on multiple clients. Some of these Contact headers might have IPv6 addresses. An IPv4-only UAC must be able to handle the IPv6 information properly.

3.10. Unsupported Address

If the endpoint is an IPv4-only client and it receives a request with an SDP offer that has IPv6 address(es) only, the IPv4-only client should decline the request by returning 488 "Not Acceptable Here" (as defined in section 13.3.1.2 of RFC3261) with Warning header that has warning code of 301 "Incompatible Network Address Formats" (as defined in section 20.43 of RFC3261). If the ipv4-only client receives a request with an SDP offer that has a mixed set of IPv4 and IPv6 addresses, then the IPv4-only client should accept the IPv4 address(es) and decline the IPv6 address(es) by setting the port number in the m-line to zero.

4. Security Considerations

This document merely describes the potential impacts of IPv6 on IPv4 SIP implementations. The scenarios discussed in this informational document do not introduce any new security threats. The specific security vulnerabilities, attacks, threat models of the various protocols discussed in this document (SIP, SDP, ICE, etc.) are well documented in their respective documents.

5. IANA Considerations

This document does not require actions by IANA.

6. Acknowledgements

The authors would like to acknowledge the support and contribution of the SIP Forum IPv6 Working Group. Mohamed Boucadair has contributed significant ideas and text. Dan Wing, Hadriel Kaplan, Paul Kyzivat, Dale Worley, and Neel Neelakantan have all provided a detailed review of the document and thoughtful comments.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3680] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations", RFC 3680, March 2004.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5589] Sparks, R., Johnston, A., and D. Petrie, "Session Initiation Protocol (SIP) Call Control - Transfer", BCP 149, RFC 5589, June 2009.

[XML] Sperberg-McQueen, C., Yergeau, F., Bray, T., Maler, E., and J. Paoli, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

Appendix A. Additional Guidelines

Some additional interoperability guidelines are presented in this section.

A.1. IPv6 Implementation Guidelines

This section lists basic IPv6 recommendations for SIP implementations:

To avoid parsing errors, IPv6 address MUST be delimited by "[" and "]" in the following cases:

* If an IPv6 address is included in a SIP Request URI
* If an IPv6 address is included in a SIP "Via" header
* If an IPv6 address is included in a SIP "Contact" header

No delimiters are needed for other SIP tags such as "received" or even at the SDP level.

The SIP ABNF for IPv6 reference defined in [RFC3261] MUST NOT be used. Instead, rules defined in [RFC3986] MUST be supported.

To compare SIP URIs, [RFC5954] MUST be used instead of [RFC3261].

[RFC5952] MUST be supported for IPv6 textual representation purposes.

An IPv6-enabled SIP MUST NOT include any loopback address (::1) or link local address (fe80) in SIP headers and SDP body.

A.2. IPv6/IPv4 Interworking Function: Avoid IPv6 address Leakage?

The introduction of IPv6-enabled SIP UAs may lead to some failure issues of the legacy (IPv4-only) UA are unable to parse IPv6 addresses. To prevent those failure cases, an IPv6/IPv4 Interworking Function may be deployed in the SIP infrastructure to adapt SIP messages. In particular, this interworking function may be configured to avoid leaking any IPv6 address to a legacy IPv4-only SIP UA (and vice versa). An IPv6-only SIP UA will be seen by a remote IPv4-only SIP UA as any legacy IPv4-only SIP UA. Leaking IPv6 addresses in headers is a concern only for headers used for session routing purposes (e.g., topmost via, contact, etc.).

Within managed SIP networks, the impact of leaking addresses of distinct address family should be assessed through testing campaigns. If no failures are experienced, enabling the function which prevents leaking addresses of distinct address family may be avoided.

In order to promote the use of IPv6 transfer capabilities and avoid extensive usage of IPv4/IPv6 interworking resources, leaking IPv6 addresses in a backward compatible manner should be encouraged. For instance, the SDP offer can include both IPv4 and IPv6 addresses (e.g., [RFC6947]). The address family to be used to place the session will be decided by the remote peer.

When both IPv4 and IPv6 SIP UA are deployed in a network, the SIP Proxy Server will need a trigger to decide whether invoking IPv4/IPv6 Interworking function is required; otherwise IPv4/IPv6 IWF resources won't be optimized. A potential solution for this problem is discussed in [I-D.boucadair-dispatch- ipv6-atypes]. Relying on the address of contact is not deterministic since a dual-stack SIP UA may be registered with its IPv4 address while it supports also IPv6.

Authors' Addresses

Carl Klatsky (editor)
Comcast
1717 Arch St.
Philadelphia, PA 19103
US

Email: carl_klatsky@cable.comcast.com

Olle E. Johansson
Edvina
Runbovaegen 10
Sollentuna SE-192 48
SE

Email: oej@edvina.net

Rifaat Shekh-Yusef
Avaya
250 Sidney Street
Belleville, Ontario
Canada

Email: rifatyu@avaya.com

Andrew Hutton
Siemens Enterprise Communications
Technology Drive
Nottingham NG9 1LA
UK

Email: andrew.hutton@siemens-enterprise.com

Gonzalo Salgueiro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: gsalguei@cisco.com

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 exmaple 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 popuplar 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

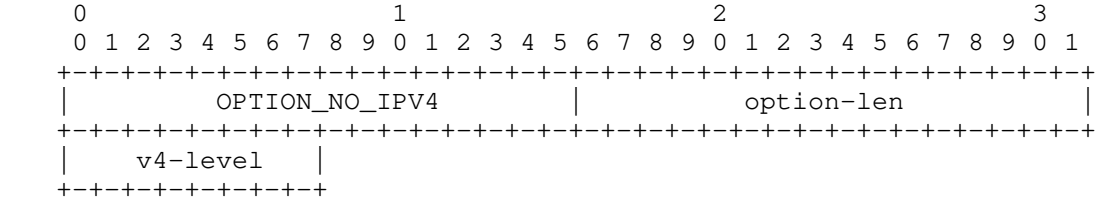
Both DHCPv6- and RA-based solutions are presented in this draft. It is expected that the working group will decide whether both solutions, only one, or none are desirable.

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:



option-code OPTION_NO_IPV4 (TBD).

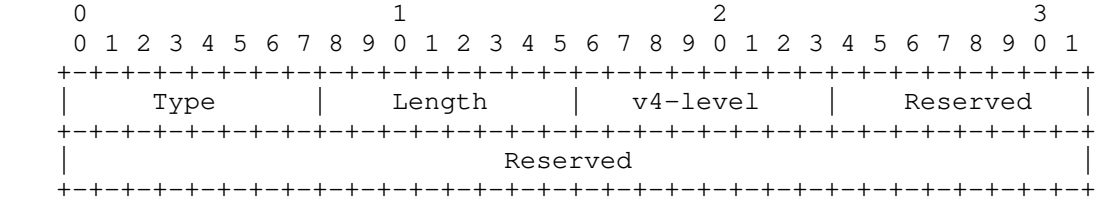
option-len 1.

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:



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

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 17, 2014

P. Patil
Cisco
M. Boucadair
France Telecom
D. Wing
T. Reddy
Cisco

September 13, 2013

DHCPv6 Dynamic Reconfiguration
draft-wing-dhc-dns-reconfigure-02

Abstract

This specification extends DHCPv6 so that a DHCPv6 Relay Agent can dynamically indicate end host connectivity to a DHCPv6 Server. This information is also triggered by any change in connectivity type provided to the host. The DHCPv6 server uses this information as an input to its decision-making about configuration parameters to be conveyed to that host.

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 March 17, 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. Problem Statement: Focus on DNS Reconfiguration 3
4. Host Connectivity Status Option 4
5. DHCPv6 Relay Agent Behavior 5
5.1. Relay Forward 5
5.2. Reconfigure Request 6
6. DHCPv6 Server Behavior 6
6.1. Relay Forward 6
6.2. Reconfigure Request 6
7. Host Tracking 7
8. Security Considerations 7
9. IANA Considerations 7
10. References 7
10.1. Normative References 8
10.2. Informative References 8
Authors' Addresses 9

1. Introduction

Some networks are expected to support IPv4-only, dual-stack, and IPv6-only hosts at the same time. Due to devices capabilities and available connectivity types, providing generic configuration from a DHCP server to connected hosts is sub-optimal in most cases, and may even break functionality in some cases. Network infrastructure is usually well equipped to be aware of single/dual-stack nature of hosts. The network can also track and detect transitions from single to dual-stack or vice-versa.

This specification describes a DHCPv6 extension for relay agents to indicate host characteristics pertaining to host connectivity to DHCPv6 servers. The information passed by a relay is generic and a DHCPv6 server can interpret and process this information to make a more informed decision on the configuration parameters that a client is to receive.

The DHCPv6 server can either be configured or have built-in logic to use this information as desired, which is outside the scope of this document.

Section 3 describes a typical problem that can be addressed using the mechanism described in this specification. A DHCPv6 server makes a decision on priority of DNS servers to be sent back to the client based on host connectivity characteristics provided by the relay agent.

While the host stack can be upgraded to send this information to the DHCPv6 server on its own, a generalized upgrade of all DHCPv6 client implementations on all operating systems is extremely difficult.

> [DISCUSSION NOTE: A companion solution could be to define a container that can be used to return per-AF specific configuration parameters to the client. In such a scheme, the server blindly returns all pieces of configuration and it is up to the client to make use of appropriate set of parameters according to its available connectivity. This alternative assumes an update of dhcp client. This approach can be seen as complimentary to the one defined in this specification. The document will be updated to reflect consensus of the WG on whether the additional option is to be specified.]

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].

Dual-Stack host: Denotes a host that is configured with both an IPv4 address and IPv6 prefix and is reachable using both IPv4 and IPv6 connectivity.

3. Problem Statement: Focus on DNS Reconfiguration

Default address selection rules specified in [RFC6724] prefers IPv6 over IPv4. If a dual-stack host is configured to use a DNS64 server [RFC6147], it will send its DNS queries to that DNS64 server which will synthesize a AAAA response if no A record is found. Thus, the dual-stack host will always use IPv6 if a DNS lookup was involved, even if IPv4 could have been used more optimally.

In some deployments, if NAT44 [RFC3022] and NAT64 [RFC6146] are deployed on the same network, it is preferable to use NAT44 over NAT64 because of scale, performance and application incompatibility issues (e.g., FTP) [RFC6384]. At the same time, native IPv6 can still be preferred over IPv4.

A DHCPv6 Relay Agent can observe host characteristics on a network to determine if a host is IPv4-only, dual-stack or IPv6-only and also

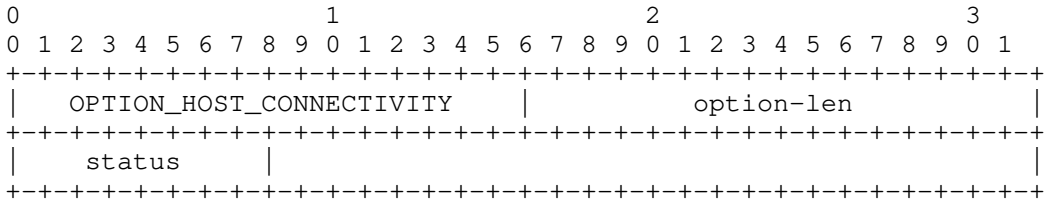
detect transitions from single to dual-stack or vice-versa. This information can be used by the DHCPv6 Relay Agent to influence the DHCPv6 Server to send appropriately prioritized DNS Servers to the client. The DHCPv6 server can implement the following based on connectivity information received from the relay agent.

- o IPv6-only transition to Dual-Stack: In case a host is IPv6-only, it is provided with a DNS64 server. When transitioning to dual-stack, an IPv4 DNS server is assigned as a consequence of obtaining an IPv4 Address. The DHCPv6 Relay Agent can detect this and send a RECONFIGURE_REQUEST message [RFC6977] to the DHCPv6 Server indicating that the host needs to be provided with a regular DNS server. In lieu of this mechanism, the host would continue to use the DNS64 server until the host stack reinitializes.

- o Dual-Stack to IPv6-only: In case a host is dual-stack, it is provided with a regular DNS server followed by DNS64 server. When transitioning to IPv6-only, the DHCPv6 Relay Agent can detect this change and send a RECONFIGURE_REQUEST message to the DHCPv6 server indicating that the host needs to be assigned a DNS64 server only. In lieu of this mechanism, the host would continue to use the regular DNS Server which is inaccessible and eventually time out to fail over to the DNS64 Server. The host will take additional time to fully initialize causing delays in connection.

4. Host Connectivity Status Option

The option (Figure 1) includes an 8-bit status code that indicates specific host connectivity characteristics. The option can be included by a DHCPv6 Relay Agent in RELAY-FORW and RECONFIGURE-REQUEST.



option-code OPTION_HOST_CONNECTIVITY (TBA).
option-len 1.
status 8-bit integer value carrying the connectivity status of a host. The following codes are defined:

Value	Name

1	IPv4_TO_DUAL_STACK
2	IPv6_TO_DUAL_STACK
3	DUAL_STACK_TO_IPv4
4	DUAL_STACK_TO_IPv6

Figure 1: Relay Agent Host Connectivity Option message format

- o IPv4_TO_DUAL_STACK: Host is transitioning from IPv4-Only to Dual-Stack mode.
- o IPv6_TO_DUAL_STACK: Host is transitioning from IPv6-Only to Dual-Stack mode.
- o DUAL_STACK_TO_IPv4: Host is transitioning from Dual-Stack to IPv4-Only mode.
- o DUAL_STACK_TO_IPv6: Host is transitioning from Dual-Stack to IPv6-Only mode.

5. DHCPv6 Relay Agent Behavior

DHCPv6 relay agents that implement this specification MUST be configurable for tracking host connectivity and inserting the OPTION_HOST_CONNECTIVITY option in RELAY-FORW and RECONFIGURE-REQUEST messages.

To be able to notify details of hosts' connectivity, a relay agent must be able to track host connectivity. A Relay Agent can detect host connectivity type using mechanisms discussed in Section 7. The Relay Agent then includes this information in the appropriate DHCPv6 message.

Relay agents need to maintain connectivity state of each host it can track. This ensures that notifications to the DHCPv6 server, especially DHCPv6 RECONFIGURE_REQUEST, are accurately sent when there is a change in status. If a relay agent loses state due to some reason (e.g., during restart events), it will build state again using the mechanisms described in Section 7 and then send appropriate notifications to the server. Such notifications are redundant and a DHCPv6 Server can choose to ignore such redundant notifications from the relay agent. Redundant notifications are also possible when relay agents are deployed in fault tolerant mode.

5.1. Relay Forward

DHCPv6 relay agents that implement this specification MAY include the option OPTION_HOST_CONNECTIVITY in the RELAY_FORW to indicate status of host connectivity.

5.2. Reconfigure Request

DHCPv6 relay agents that implement this specification MUST be configurable for sending the RECONFIGURE_REQUEST message. The relay agent generates a Reconfigure-Request [RFC6977] anytime status of host connectivity changes by including OPTION_HOST_CONNECTIVITY in the request.

6. DHCPv6 Server Behavior

A DHCPv6 Server that supports OPTION_HOST_CONNECTIVITY may either have specific configuration or built-in logic to process information available in the option and send configuration parameters in DHCPv6 responses. How the server consumes and acts on the information obtained in the option are outside the scope of this document.

The DHCPv6 server may use this connectivity information, if available, in addition to other relay agent option data, other options included in the DHCPv6 client messages, server configuration, and physical network topology information in order to assign appropriate configuration to the client.

The server MUST ignore the option if it doesn't recognize the status in the OPTION_HOST_CONNECTIVITY option. The server SHOULD maintain the latest status received from the relay agent. The server can use this state to match against subsequent notifications and only further process if there is change in status. A relay agent could, for reasons such as restart, fault-tolerant mode etc, send redundant notifications and matching of status at the server will avoid unnecessary processing and message exchanges.

6.1. Relay Forward

Upon receiving a RELAY-FORW message containing OPTION_HOST_CONNECTIVITY, the server can send appropriate configuration in the RELAY-REPLY response. The server MUST NOT return this option in a RELAY-REPLY message.

6.2. Reconfigure Request

Upon receiving a RECONIFURE-REQUEST message containing an OPTION_HOST_CONNECTIVITY option, the server MUST follow the mechanism described in [RFC6977] to create and send Reconfigure message. The server MUST NOT return this option in a RECONFIGURE-REPLY message.

7. Host Tracking

Relay Agents can actively keep track of all IPv4/IPv6 addresses and associated lease times assigned to hosts via the respective DHCP servers. Relay Agents can therefore detect transitions from single to dual-stack and vice-versa efficiently. In addition to this technique, relay agents closest to the client can detect transitions using snooping mechanisms. Network devices today use mechanisms such as ARP and NDP snooping (bindings learnt by snooping all NDP traffic, NS, NA, RS, RA) to determine host characteristics such as IPv4/IPv6 - MAC - DUID bindings. IPv4/IPv6 and MAC counters are also used to determine host liveliness.

First hop devices that implement first hop security also track IP address bindings and determine binding updates such as temporary addresses, deprecated addresses, etc. Existing work such as [I-D.ietf-savi-dhcp] and [I-D.levy-abegnoli-savi-plbt] also aim to active current host bindings, all of which can be leveraged to track host addresses.

These mechanisms help determine if a particular IP address family is inactive, has reverted to using a single stack even though it initially had dual-stack capabilities and detect active dual-stack usage after long periods of single-stack activity.

Other techniques to track host connectivity can be envisaged. It is out of scope of this document to provide an exhaustive list of host tracking techniques.

8. Security Considerations

This document describes an application of the mechanism specified in [RFC6977]. Host tracking mechanisms MUST be reliable. If a relay is compromised, it may be used to force an uncompromised server abuse clients by triggering repetitive reconfigurations. Security considerations described in [RFC6977] are applicable to this mechanism.

9. IANA Considerations

IANA is requested to assign the following new DHCPv6 Option Code in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

- OPTION_HOST_CONNECTIVITY

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6384] van Beijnum, I., "An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation", RFC 6384, October 2011.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

[RFC6977] Boucadair, M. and X. Pougnard, "Triggering DHCPv6 Reconfiguration from Relay Agents", RFC 6977, July 2013.

10.2. Informative References

[I-D.ietf-savi-dhcp] Bi, J., Wu, J., Yao, G., and F. Baker, "SAVI Solution for DHCP", draft-ietf-savi-dhcp-18 (work in progress), June 2013.

[I-D.levy-abegnoli-savi-plbt] Levy-Abegnoli, E., "Preference Level based Binding Table", draft-levy-abegnoli-savi-plbt-02 (work in progress), March 2010.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[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.

[RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

[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.

[RFC6147] 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.

Authors' Addresses

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

Email: praspati@cisco.com

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com