

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 12, 2014

G. Chen
China Mobile
T. Tsou
Huawei Technologies
C. Donley
CableLabs
T. Taylor
PT Taylor Consulting
April 10, 2014

Analysis of NAT64 Port Allocation Methods for Shared IPv4 Addresses
draft-chen-sunset4-cgn-port-allocation-04

Abstract

This document enumerates methods of port assignment in Carrier Grade NATs (CGNs), focused particularly on NAT64 environments. A theoretical framework of different NAT port allocation methods is described. The memo is intended to clarify and focus the port allocation discussion and propose an integrated view of the considerations for selection of the port allocation mechanism in a given deployment.

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 October 12, 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	3
2. Considerations For the Choice of Port Allocation Methods . .	3
2.1. Port Consumption on NAT64	4
2.2. Classification of Port Allocation Models	5
2.2.1. Stateful vs. Stateless	5
2.2.2. Dynamic vs. Static	5
2.2.3. Centralized vs. Distributed	6
2.3. Port Allocation Solutions	7
2.3.1. Older Transition Technologies	7
2.3.2. Current Work On Stateless Transition Technologies . .	7
2.3.3. Port Control Protocol (PCP)	8
2.4. Specific Considerations	8
2.4.1. Log Volume Optimization	8
2.4.2. Connectivity State Optimization	10
2.4.3. Port Randomization	11
3. Considerations For the Dynamic Assignment of Port-Ranges . .	11
3.1. Motivation	11
3.2. Implementation Issues -- Port Randomization and Port- Range Deallocation	12
3.3. Issues Of Traceability	13
3.4. Other Considerations	14
4. Deterministic Port Allocation	14
4.1. Motivation	14
4.2. Deterministic Port Ranges	16
4.2.1. IPv4 Port Utilization Efficiency	19
4.2.2. Planning and Dimensioning	19
4.2.3. Deterministic CGN Example	20
4.2.4. Additional Management Considerations	21
4.3. Failover Considerations	22
4.4. Impact On IPv6 Transition	22
5. Security Considerations	22
6. IANA Considerations	24
7. Contributors	24
8. Acknowledgements	24
9. References	24
9.1. Normative References	24
9.2. Informative References	25
Appendix A. Configuration of Server Software to Log Source	

Port	28
A.1. Apache	28
A.2. Postfix	28
A.3. Sendmail	28
A.4. sshd	29
A.5. Cyrus IMAP and UW IMAP	30
Authors' Addresses	30

1. Introduction

With the depletion of IPv4 addresses, Carrier Grade NAT (CGN) has been adopted by ISPs to expand IPv4 spaces. CGN maps IP addresses from one address realm to another, relying upon the mechanism of multiplexing multiple subscribers' connections over a smaller number of shared IPv4 addresses to provide transparent routing to end hosts. [RFC6888] specifies a number of CGN requirements. A network-based NAT is implied by several approaches to IPv6 transition including DS-Lite [RFC6333], NAT64 ([RFC6145] and [RFC6146]), and NAT444. All of these would likely fall within the scope of the CGN requirements document [RFC6888].

The first part of this memo (Section 2) focusses on the topic of IPv6 migration. The CGN may not do Network Address Port Translation (NAPT), but only Network Address Translation (NAT) [RFC3022]. In this scenario, there is no concern about port assignment. When NAPT is involved, Section 2 elaborates on the considerations for address sharing and particularly port assignment in the NAT64 environment, where IPv6-only nodes are connected to external dual-stack or IPv4 networks.

Section 3 looks more closely at dynamic bulk assignment of ports to individual subscriber sites, particularly as a means of log volume reduction. The proposals made in this section are applicable to the CGN environment in general, independently of the particular flavour of translation being used.

Finally, Section 4 looks at a scheme for assignment of ports using a deterministic algorithm that has the potential to simplify operations.

2. Considerations For the Choice of Port Allocation Methods

For port allocations on NAT64, several aspects may have to be considered when selecting a suitable method. Here is a list of the potential considerations, which are covered in more detail below.

- o specific features of port usage in a NAT64 environment;

- o classification of different port allocation methods;
- o port allocation to improve connectivity;
- o port allocation to optimize log volume;
- o port allocation to enhance security.

Both analysis and relevant experimental results are presented in the sub-sections that follow.

2.1. Port Consumption on NAT64

Thanks to its simplicity and efficiency, NAT64 will likely be deployed widely. In a typical scenario, NAT64 will enable internal IPv6-only hosts to connect to external dual-stack or IPv4 networks. Compared with NAT44, fewer ports per subscriber are consumed on NAT64, because only flows between different address families require ports to be assigned. That is, a NAT44 will 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.

One of the authors did a test comparison of port consumption on NAT64 and NAT44. Top100 websites (referring to Alexa statistics) were assessed to evaluate status of port usage on NAT44 and NAT64 respectively. It was observed that the port consumption per session on NAT64 is roughly only half that 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. But it is apparent that the effects of port saving on NAT64 will be amplified by increasing native IPv6 support.

Apart from the above observation, port allocation can be tuned according to the phase of IPv6 migration. The use of NAT64 will advance IPv6 deployment, because it provides everyone with incentives to use IPv6, and eventually the result is an end-to-end IPv6-only network with no need for port allocations. As more content providers and services become available over IPv6, the utilization of NAT64 goes down since fewer destinations require translation progressing. Thus as IPv6 migration proceeds, it will be possible to relax the multiplexing ratio of IPv4 address sharing.

2.2. Classification of Port Allocation Models

This section lists several models to allocate the port information in NAT64 equipment. It also describes example cases for each allocation model.

2.2.1. Stateful vs. Stateless

o Stateful

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

In the case of static address assignment, a one-to-one address mapping for hosts between a IPv6 network address and an IPv4 network address is pre-configured on the NAT operation. This case normally occurs when a server is deployed in a IPv6 domain. The static configuration ensures stable inbound connectivity.

Dynamic address assignment would periodically free the binding so that the global address could be recycled for later use. This increases the efficiency of usage of IPv4 addresses.

o Stateless

Stateless NAT is performed in compliance with [RFC6145]. The public IPv4 address is required to be embedded in the IPv6 address. Thus the NAT64 can directly extract the address and has no need to record mapping states.

A promising usage of stateless NAT may appear in the data centre environment where IPv6 server pools receive inbound connections from IPv4 users externally [I-D.anderson-siit-dc]. NAT usage in other cases may be controversial. First off, the static one-to-one mapping does not address the issue of IPv4 depletion. Secondly, it introduces a dependency between IPv4 and IPv6 addressing. That creates new limitations since a change of IPv4 address will cause renumbering of IPv6 addresses.

2.2.2. Dynamic vs. Static

Port assignment can be dynamic (ports allocated on demand) or static (ports allocated as part of the configuration process).

o Dynamic assignment

NAT64 normally uses dynamic assignment, since this achieves higher port utilization. Port allocations can be made with per-session

or per-customer granularity. Per-session assignment is configured on the NAT64 by default since it maximizes port utilization. However, this can result in a heavy log volume that may have to be recorded for lawful interception systems. To mitigate that concern, the NAT64 may dynamically allocate a port range for each connected subscriber. This will significantly reduce log volume.

A proper port-range configuration may have to take into account two considerations:

- A. The number of session initiations for each subscriber. A subscriber normally uses multiple applications simultaneously, e.g. map, online video or game. The number of concurrent sessions is essential to determine the number of ports the subscriber needs. It has been learned from subscribers' behaviors that the average number of sessions consumed by one user's device is around 200 to 300 ports. Several devices may appear behind a CPE. Administrators may configure a range with 1000 ports to each CPE in fixed networks.
- B. Impacts on NAT64 capacity. Preassigned port ranges occupy memory even when there are unused ports. Therefore, the operator should be cautious about the impact of port-range reservation on the capacity for attempted concurrent sessions, especially in the case of a centralized NAT64 CGN serving numerous subscribers.

- o Static assignment

Static assignment makes port reservations in bulk for each internal address before subscriber connection. The assigned ports can be in either a contiguous or non-contiguous port range for the sake of attack defense. Log recording may not be necessary due to the stable mapping relations. Considerations of the interaction between port-range allocation and capacity impact are also applicable in the case of static assignment. Section 4 describes a deterministic algorithm to assign a port range for an internal IP address pool in a sequence.

2.2.3. Centralized vs. Distributed

There is an increasing need to connect NAT64 with downstream NAT46-capable devices to support IPv4 users/applications on an IPv6-only path. Several solutions have been proposed in this area, e.g., 464xlat [RFC6877], MAP-T [I-D.ietf-software-map-t] and 4rd [I-D.ietf-software-4rd]. Port allocation can be categorized as a centralized assignment on NAT64 or as a port delegation distributed to downstream devices (e.g, Customer Edge connected with NAT64).

- o Centralized Assignment

A centralized method makes port assignments once IP flows come to the 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 can also delegate the pre-allocated port range to customer edge devices. That can be achieved through additional out-of-band provisioning signals (e.g., [I-D.ietf-pcp-port-set], [I-D.ietf-softwire-map-dhcp]). The distributed model normally is performed A+P style for static port assignment. The NAT64 should also hold the corresponding mapping in order to validate port usage in the outgoing direction and route inbound packets. Delegated port ranges shift NAT64 port computations/states into downstream devices. The detailed benefits of this approach are documented in [I-D.ietf-softwire-stateless-4v6-motivation].

2.3. Port Allocation Solutions

2.3.1. Older Transition Technologies

In older work, stateful NAT64 [RFC6146] uses bindings between IPv4 and IPv6 addresses that may be either static or dynamic. [RFC6146] describes a process where the dynamic binding is created by an outgoing packet, but it may also be created by other means such as a Port Control Protocol request (see Section 2.3.3). Stepping outside the boundaries of NAT64 for the moment, DS-Lite [RFC6333] refers to the cautions in [RFC6269] but does not specify any port allocation method. Both technologies assume a centralized model.

The specifications for both transition methods thus allow implementations to use the proposals made in Section 3 and Section 4.

2.3.2. Current Work On Stateless Transition Technologies

The port allocation solutions that are being specified at the time of writing of this document are all variations on the static distributed model, to minimize the amount of state that has to be held in the network. The proposals made in Section 3 and Section 4 do not apply to the current work in progress because that work has gone in another direction. That work includes:

- o Light-weight 4over6 (LW4o6 [I-D.ietf-softwire-lw4over6]), which requires the CPE to be configured explicitly with the shared IPv4 address and port set it will use on the WAN side of its NAT44

function. The border router is configured with the same information, reducing the state it must hold from per-session to per-subscriber amounts.

- o Mapping of Address and Port with Encapsulation (MAP-E [I-D.ietf-softwire-map]) and the experimental specifications Mapping of Address and Port with Translation (MAP-T [I-D.ietf-softwire-map-t]) and 4rd [I-D.ietf-softwire-4rd], already mentioned. These rely on an algorithmic embedding of WAN-side IPv4 address and assigned port set within the IPv6 prefix assigned to each CPE. Both the CPE and the border router must be configured with this information. However, the algorithm is designed to aggregate routing information such that the amount of state carried by the border router is of a lower order of magnitude than even the per-subscriber level.

MAP-E also supports a 1-1 mapping mode, where the IPv4 and IPv6 addresses assigned to a CPE are independent. This can be helpful in transition, but, as with LW4o6, raises the amount of state in the network back to the per-subscriber level.

For a packet destined to a host outside the MAP domain from which the packet originated: MAP-E and 4rd treat the packet as an IPv4 over IPv6 tunnel via the border router.

MAP-T uses stateless mapping in the sense of Section 2.2.1 by embedding the destination IPv4 address within the IPv6 address of the packet sent to the border router.

2.3.3. Port Control Protocol (PCP)

The Port Control Protocol (PCP, [RFC6887]) can be used to reserve a single port or a port set [I-D.ietf-pcp-port-set] for applications. It requires that the NAT be collocated with a PCP server function. PCP provides an out-of-band signalling mechanism for coordinating dynamic allocation of ports between hosts and the border router.

2.4. Specific Considerations

2.4.1. Log Volume Optimization

[RFC6269] has provided a thoughtful analysis on the issues of IP sharing. It points out that IP sharing may impact law enforcement since source address information will 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 pose a challenge to operators, since it requires additional

resources and data inspection processes to identify users. For concrete details of what should be logged, see Section 3.1 of [I-D.ietf-behave-syslog-nat-logging]. The actual logging may use either IPFIX [RFC7011] or Syslog [RFC5424] depending on the operator's requirements.

It is desirable to reduce the volume of the logged information. Referring to the classification of port allocation methods given above, dynamic assignments can be managed on either a per-session or per-customer granularity. The coarser granularity will lead to lower log volume storage. A test was made by recording the log information from 200,000 subscribers in the Chinese network for 60 days. The volume of recorded information reached up to 42.5 terabytes with per-session logging in the raw format. The volume could be reduced to 10.6 terabytes with gzip format. Compared with that, it only occupied 40.6 gigabytes, three orders of magnitude smaller volume, with per-customer logging in the raw format. With static allocation, of course, no logs at all are required.

On the other hand, the lower logging volumes are associated with lower efficiency of port utilization. A port allocation based on per-customer granularity has to retain vacant ports in order to avoid traffic overflow. The efficiency can be evaluated by port utilization rate, and will be even lower if the static port allocation method is used. Inactive users may also impact the efficiency.

Table 1 summarizes the test results using Syslog. The ports were pre-allocated to customers regardless of online or offline status.

Port Allocation Method	Log Granularity	Estimated Log Volume	Port Utilization
Dynamic NAPT	Per-session	42.5 terabytes	100%
Dynamic port-range	Per-customer	40.6 Gigabytes	75%
Deterministic NAT, MAP-T, 4rd	None	None	(60% * 75%) = 45%

Table 1: Estimated Log Volumes For 200,000 Users Over 60 Days

Note: 75% is the estimated port utilization ratio per active subscriber. 60% is the estimated ratio of active subscribers to the total number of subscribers.

The data shown in Table 1 roughly demonstrates the tradeoff between port utilization and log volume reduction. Administrators may consider the following factors to determine their own solution:

- o average connectivity per customer per day;
- o peak connectivity per day;
- o the number of public IPv4 addresses available to the NAT64;
- o application demands for specific ports;
- o processing capabilities of the NAT64;
- o tolerable log volume.

2.4.2. Connectivity State Optimization

It has been observed that port consumption is significantly increased once subscribers land on a web page for video on demand, an online game, 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. Given the video traffic growth trend, this likely presents a challenge for network operators who need to optimize connectivity states and avoid port depletion. Those optimizations may even affect the method of port-range allocation, because a subscriber is only allowed to use a pre-configured port resource.

Two optimizations may be considered:

- o Reducing the TIME-WAIT state. The user's behavior normally correlates with system performance. It is rather common that users change video channels often. Investigations have shown that 60% of videos are watched for less than 20% of their duration. The user's access patterns may leave a number of the TIME-WAIT states. Therefore, acceleration of TIME-WAIT state transitions could increase the efficiency of port utilization. [RFC6191] defines a mechanism for 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. This may also be a way to improve port utilization.

- o Another possibility is to use Address-Dependent Mapping or Address and Port-Dependent Mapping [RFC4787] to increase port utilization.

This feature has already been implemented on a vendor-specific basis. However, it should be noted that REQ-7 and REQ-12 in [RFC6888] may reduce the incentive to use anything but the Address-Independent Mapping behaviour recommended by [RFC4787].

2.4.3. Port Randomization

Port randomization is a feature to enhance the defense against hijacking of flows. [RFC6056] specifies that:

"A NAT that does not implement port preservation ([RFC4787], [RFC5382]) should obfuscate selection of the ephemeral port of a packet when it is changed during translation of that packet."

A NAT based on per-session allocation normally follows this recommendation. However, a simple algorithm for port assignment is generally desirable for a deterministic NAT even if it increases hijack vulnerability.

See Section 5 for a fuller discussion of port randomization.

3. Considerations For the Dynamic Assignment of Port-Ranges

3.1. Motivation

During the IPv6 transition period, large-scale NAT devices may be introduced, e.g. DS-Lite AFTR, NAT64. When a NAT device needs to set up a new connection for a given internal address behind the NAT, it needs to create a new mapping entry for the new connection, which will contain source IP address, source port or ICMP identifier, converted source IP address, converted source port, protocol (TCP/UDP), etc.

For various reasons it is necessary to log these mappings. Some high performance NAT devices may need to create a large amount of new sessions per second. As seen in Section 2.4.1, if the logs are generated for each mapping entry, the log traffic could reach tens of megabytes per second or more, which would be a problem for log generation, transmission and storage. (The per-session volumes in Table 1 amount to 42 bytes per served subscriber per second. The volumes reported in Section 2.4.2 for US users are even higher, around 58 bytes per second per subscriber served.)

[RFC6888], REQ-13, REQ-14, and REQ-15 deal explicitly with port allocation schemes and logging. However, it is recognized that these are conflicting requirements, requiring a tradeoff between the efficiency with which ports are used and the rate of generation of log records.

Allocating a range of N ports at once reduces the log volume by a factor of N, while also reducing port utilization by a factor which varies with the address sharing ratio and other configuration parameters. This provides a clear motivation to use dynamic allocation of port-ranges rather than individual ports when it is possible to do so while maintaining a satisfactory level of port utilization (and by implication, shared global IPv4 address utilization).

Dynamic allocation of port ranges may be used either as the sole strategy for port allocation on the NAT, or as a supplement to an initial static allocation.

3.2. Implementation Issues -- Port Randomization and Port-Range Deallocation

Here is how dynamic allocation of port-ranges would work in greater detail. When the user sends out the first packet, a port resource pool is allocated for the user, e.g., assigning ports 2001~2300 of a public IP address to the user's resource pool. Only one log should be generated for this port block. When the NAT needs to set up a new mapping entry for the user, it can use a port in the user's resource pool and the corresponding public IP address. If the user needs more port resources, the NAT can allocate another port block, e.g., ports 3501~3800, to the user's resource pool. Again, just one log needs to be generated for this port block.

[I-D.bajko-pripaddressign] takes this idea further by allocating non-contiguous sets of ports using a pseudorandom function. Scattering the allocated ports in this way provides a modest barrier to port guessing attacks. The use of randomization is discussed further in Section 5.

Suppose now that a given internal address has been assigned more than one block of ports. The individual sessions using ports within a port block will start and end at different times. If no ports in some port block are used for some configurable time, the NAT can remove the port block from the resource pool allocated to a given internal address, and make it available for other users. In theory, it is unnecessary to log deallocations of blocks of ports, because the ports in deallocated blocks will not be used again until the blocks are reallocated. However, the deallocation may be logged when it occurs to add robustness to troubleshooting or other procedures.

The deallocation procedure presents a number of difficulties in practice. The first problem is the choice of timeout value for the block. If idle timers are applied for the individual mappings (sessions) within the block, and these conform to the recommendations

for NAT behaviour for the protocol concerned, then the additional time that might be configured as a guard for the block as a whole need not be more than a few minutes. The block timer in this case serves only as a slightly more conservative extension of the individual session idle timers. If, instead, a single idle timer is used for the whole block, it must itself conform to the recommendations for the protocol with which that block of ports is associated. For example, REQ-5 of [RFC5382] requires an idle timer expiry duration of at least 2 hours and 4 minutes for TCP. The suggestions made in Section 2.4.2 may be considered for reducing this time.

The next issue with port block deallocation is the conflict between the desire to randomize port allocation and the desire to make unused resources available to other internal addresses. As mentioned above, ideally port selection will take place over the entire set of blocks allocated to the internal address. However, taken to its fullest extent, such a policy will minimize the probability that all ports in any given block are idle long enough for it to be released.

As an alternative, it is suggested that when choosing which block to select a port from, the NAT should omit from its range of choice the block that has been idle the longest, unless no ports are available in any of the other blocks. The expression "block that has been idle the longest" designates the block in which the time since the last packet was observed in any of its sessions, in either direction, is earlier than the corresponding time in any of the other blocks assigned to that internal address. As [RFC6269] points out, port randomization is just one security measure of several, and the loss of randomness incurred by the suggested procedure is justified by the increased utilization of port resources it allows.

3.3. Issues Of Traceability

Section 11 of [RFC6269] provides a good discussion of the traceability issue. Complete traceability given the NAT logging practices proposed in this draft requires that the remote destination record the source port of a request along with the source address (and presumably protocol, if not implicit). In addition, the logs at each end must be timestamped, and the clocks must be synchronized within a certain degree of accuracy. Here is one reason for the guard timing on block release, to increase the tolerable level of clock skew between the two ends.

The ability to configure various server applications to record source ports has been investigated, with the following results:

- o Source port recording can be configured in Apache, Postfix, sendmail and sshd. Please refer to the appendix for a configuration guide.
- o Source port recording is not supported by IIS, Cyrus IMAP and UW IMAP. But it should not be too difficult to get Cyrus IMAP and UW IMAP to support it by modifying the source code.

Where source port logging can be enabled, this memo strongly urges the operators to do so. Similarly, intrusion detection systems should capture source port as well as source address of suspect packets.

In some cases [RFC6269], a server may not record the source port of a connection. To allow traceability, the NAT device needs to record the destination IP address of a connection. As [RFC6269] points out, this will provide an incomplete solution to the issue of traceability because multiple users of the same shared public IP address may access the service at the same time. From the point of view of this draft, in such situations the game is lost, so to speak, and port allocation at the NAT might as well be completely dynamic.

The final possibility to consider is where the NAT does not do per-session logging even given the possibility that the remote end is failing to capture source ports. In that case, the port allocation strategy proposed in this section can be used. The impact on traceability is that analysis of the logs would yield only the list of all internal addresses mapped to a given public address during the period of time concerned. This has an impact on privacy as well as traceability, depending on the follow-up actions taken.

3.4. Other Considerations

[RFC6269] notes several issues introduced by the use of dynamic as opposed to static port assignment. For example, Section 12.2 of that document notes the effect on authentication procedures. These issues must be resolved, but are not specific to the dynamic port-range allocation strategy.

4. Deterministic Port Allocation

4.1. Motivation

CGN connection logging satisfies the need to identify attackers and respond to abuse/public safety requests, but it imposes significant operational challenges to operators. In lab testing, CGN log messages were observed to be approximately 150 bytes long for NAT444 [I-D.shirasaki-nat444], and 175 bytes for DS-Lite [RFC6333]

(individual log messages vary somewhat in size). Although the authors are not aware of definitive studies of connection rates per subscriber, reports from several operators in the US set the average number of connections per household at approximately 33,000 connections per day. If each connection is individually logged, this translates to a data volume of approximately 5 MB per subscriber per day, or about 150 MB per subscriber per month; however, specific data volumes may vary across different operators based on myriad factors. Based on available data, a 1-million subscriber service provider will generate approximately 150 terabytes of log data per month, or 1.8 petabytes per year.

The volume of log data poses a problem for both operators and the public safety community. On the operator side, it requires a significant infrastructure investment by operators implementing CGN. It also requires updated operational practices to maintain the logging infrastructure, and requires approximately 23 Mbps of bandwidth between the CGN devices and the logging infrastructure per 50,000 users. On the public safety side, it increases the time required for an operator to search the logs in response to an abuse report, and could delay investigations. Accordingly, an international group of operators and public safety officials approached some of the authors and contributors to this document to identify a way to reduce this impact while improving abuse response.

As noted in Section 3.1, the volume of CGN logging can be reduced by assigning port ranges instead of individual ports. Using this method, only the assignment of a new port range is logged. This may massively reduce logging volume. The log reduction may vary depending on the length of the assigned port range, whether the port range is static or dynamic, etc. This has been acknowledged in [RFC6269] and Section 5.6.10 of [I-D.ietf-behave-ipfix-nat-logging]. Per [RFC6269]:

"Address sharing solutions may mitigate these issues to some extent by pre-allocating groups of ports. Then only the allocation of the group needs to be recorded, and not the creation of every session binding within that group. There are trade-offs to be made between the sizes of these port groups, the ratio of public addresses to subscribers, whether or not these groups timeout, and the impact on logging requirements and port randomization security ([RFC6056])."

However, the existing solution still poses an impact on operators and public safety officials for logging and searching. Instead, CGNs could be designed and/or configured to deterministically map internal addresses to {external address + port range} in such a way as to be able to algorithmically calculate the mapping. Only inputs and

configuration of the algorithm need to be logged. This approach reduces both logging volume and subscriber identification times. In some cases, when full deterministic allocation is used, this approach can eliminate the need for translation logging.

This section describes a method for such CGN address mapping, combined with block port reservations, that significantly reduces the burden on operators while offering the ability to map a subscriber's inside IP address with an outside address and external port number observed on the Internet.

The activation of the proposed port range allocation scheme is compliant with BEHAVE requirements such as the support of APP. [What is APP? Reference for the complied-with requirements? Or can this para be removed?]

4.2. Deterministic Port Ranges

While a subscriber uses thousands of connections per day, most subscribers use far fewer resources at any given time.

Appendix B of [RFC6269] introduces the term "address space multiplicative factor" to denote the number of subscribers sharing the same public IPv4 address, and goes on to qualify how this value should be calculated. When the address space multiplicative factor is low (e.g., the ratio of the number of subscribers to the number of public IPv4 addresses allocated to a CGN is closer to 10:1 than 1000:1), each subscriber could have access to thousands of TCP/UDP ports at any given time. Thus, as an alternative to logging each connection, CGNs can deterministically map customer private addresses (received on the customer-facing interface of the CGN, a.k.a., internal side) to public addresses extended with port ranges (used on the Internet-facing interface of the CGN, a.k.a., external side).

The mapping algorithm allows an operator to identify a subscriber internal IP address when provided the public side IP and port number without having to examine the CGN translation logs, and avoids having to transport and store massive amounts of session data from the CGN and then process it to identify a subscriber. It can be classified as a static centralized port allocation strategy.

The algorithmic mapping can be expressed as:

(External IP Address, Port Range) = function 1 (Internal IP Address)

Internal IP Address = function 2 (External IP Address, Port Number)

Deterministic Port Range allocation requires configuration of the following variables:

- o The set of inside IPv4/IPv6 addresses <I>;
- o the set of outside IPv4 addresses <O>;
- o the address space multiplicative factor (F), i.e., ratio of number of inside IP addresses to outside IP addresses;
- o dynamic address pool factor (D), to be added to the compression ratio in order to create an overflow address pool;
- o maximum ports per user (M);
- o address assignment algorithm (A) (see below); and
- o number of reserved TCP/UDP ports (R).

Note: The inside address set <I> will consist of IPv4 addresses in NAT444 operation (NAT444 [I-D.shirasaki-nat444]) and of IPv6 addresses in DS-Lite [RFC6333] operation.

A subscriber may be identified by an internal IPv4 address (e.g., NAT44) or an IPv6 prefix (e.g., DS-Lite or NAT64). For a fuller discussion of subscriber identification, see Section 2.4 of [I-D.ietf-behave-syslog-nat-logging].

The algorithm is not designed to retrieve an internal host among those sharing the same internal IP address (e.g., in a DS-Lite context, only an IPv6 address/prefix can be retrieved using the algorithm while the internal IPv4 address used for the encapsulated IPv4 datagram is lost).

Several address assignment algorithms are possible. Using predefined algorithms, such as those that follow, simplifies the process of reversing the algorithm when needed. However, the CGN may support additional algorithms, and may not support all algorithms described below. Subscribers could be restricted to ports from a single IPv4 address, or could be allocated ports across all addresses in a pool, for example. The following algorithms and corresponding values of A are suggested as a starting set:

A = 0: Sequential (e.g. the first block goes to address 1, the second block to address 2, etc.)

A = 1: Staggered (e.g. for every n between 0 and $((65536-R)/(F+D))-1$, address 1 receives ports $n \cdot F + R$, address 2 receives ports $(1+n) \cdot F + R$, etc.)

A = 2: Round robin (e.g. the subscriber receives the same port number across a pool of external IP addresses. If the subscriber is to be assigned more ports than there are in the external IP pool, the subscriber receives the next highest port across the IP pool, and so on. Thus, if there are 10 IP addresses in a pool and a subscriber is assigned 1000 ports, the subscriber would receive a range such as ports 2000–2099 across all 10 external IP addresses).

A = 3: Interlaced horizontally (e.g. each address receives every C th port spread across a pool of external IP addresses).

A = 4: Cryptographically random port assignment (Section 2.2 of [RFC6431]). If this algorithm is used, the Service Provider needs to retain the keying material and specific cryptographic function to support reversibility.

The assigned range of ports can also be used when translating ICMP requests (when re-writing the Identifier field).

The CGN then reserves ports as follows:

1. The CGN removes reserved ports (R) from the port candidate list (e.g., 0–1023 for TCP and UDP). At a minimum, it is likely that the operator will prefer the CGN to remove system ports [RFC6335] from the port candidate list reserved for deterministic assignment.
2. The CGN calculates the total address space multiplicative factor $(F+D)$, and allocates $1/(F+D)$ of the available ports to each internal IP address. Specific port allocation is determined by the algorithm (A) configured on the CGN. Any remaining ports are allocated to the dynamic pool.

Note: Setting D to 0 disables the dynamic pool. This option eliminates the need for per-subscriber logging at the expense of limiting the number of concurrent connections that 'power users' can initiate.

3. When a subscriber initiates a connection, the CGN creates a translation mapping between the subscriber's inside local IP address/port and the CGN outside global IP address/port. The CGN uses one of the ports allocated in step 2 for the translation as long as such ports are available. The CGN allocates ports

randomly within the port range assigned by the deterministic algorithm. This is to increase subscriber privacy. The CGN must also use the preallocated port range from step 2 for Port Control Protocol (PCP, [RFC6887]) reservations as long as such ports are available. While the CGN maintains its mapping table, it need not generate a log entry for translation mappings created in this step.

4. If $D > 0$, the CGN will have a pool of ports left for dynamic assignment. If a subscriber uses more than the range of ports allocated in step 2 (but fewer than the configured maximum ports M), the CGN assigns a block of ports from the dynamic assignment range for such a connection or for PCP reservations. The CGN logs dynamically assigned port blocks to facilitate subscriber-to-address mapping. The CGN should manage dynamic ports as described in Section 3.
5. Configuration of reserved ports (e.g., system ports) is left to the operator.

Thus, the CGN will maintain translation mapping information for all connections within its internal translation tables; however, it only needs to externally log translations for dynamically-assigned ports.

4.2.1. IPv4 Port Utilization Efficiency

For Service Providers requiring an aggressive address space multiplicative factor, the use of the algorithmic mapping may impact the efficiency of the address sharing. A dynamic port range allocation assignment is more suitable in those cases.

4.2.2. Planning and Dimensioning

Unlike dynamic approaches, the use of the algorithmic mapping requires more effort from operational teams to tweak the algorithm (e.g., size of the port range, address space multiplicative factor, etc.). Operators should configure dedicated alarms triggered by port utilization threshold crossings so that the configuration can be refined.

The use of algorithmic mapping also affects geolocation. Changes to the inside and outside address ranges (e.g. due to growth, address allocation planning, etc.) will require external geolocation providers to recalibrate their mappings.

4.2.3. Deterministic CGN Example

To illustrate the use of deterministic NAT, let us consider a simple example. The operator configures an inside address range (I) of 100.64.0.0/28 [RFC6598] and outside address (O) of 203.0.113.1. The dynamic address pool factor (D) is set to '2'. Thus, the total compression ratio is $1:(14+2) = 1:16$. Only the system ports (e.g. ports < 1024) are reserved (R). This configuration causes the CGN to preallocate $(65536-1024)/16 = 4032$ TCP and 4032 UDP ports per inside IPv4 address. For the purposes of this example, let's assume that they are allocated sequentially, where 100.64.0.1 maps to 203.0.113.1 ports 1024-5055, 100.64.0.2 maps to 203.0.113.1 ports 5056-9087, etc. The dynamic port range thus contains ports 57472-65535 (port allocation illustrated in Table 2). Finally, the maximum ports/subscriber is set to 5040.

Inside Address / Pool	Outside Address & Port
Reserved	203.0.113.1:0-1023
100.64.0.1	203.0.113.1:1024-5055
100.64.0.2	203.0.113.1:5056-9087
100.64.0.3	203.0.113.1:9088-13119
100.64.0.4	203.0.113.1:13120-17151
100.64.0.5	203.0.113.1:17152-21183
100.64.0.6	203.0.113.1:21184-25215
100.64.0.7	203.0.113.1:25216-29247
100.64.0.8	203.0.113.1:29248-33279
100.64.0.9	203.0.113.1:33280-37311
100.64.0.10	203.0.113.1:37312-41343
100.64.0.11	203.0.113.1:41344-45375
100.64.0.12	203.0.113.1:45376-49407
100.64.0.13	203.0.113.1:49408-53439
100.64.0.14	203.0.113.1:53440-57471
Dynamic	203.0.113.1:57472-65535

Table 2: Port Allocation For Deterministic NAT Example

When subscriber 1 using 100.64.0.1 initiates a low volume of connections (e.g. < 4032 concurrent connections), the CGN maps the outgoing source address/port to the preallocated range. These translation mappings are not logged.

Subscriber 2 concurrently uses more than the allocated 4032 ports (e.g. for peer-to-peer, mapping, video streaming, or other connection-intensive traffic types), the CGN allocates up to an additional 1008 ports using bulk port reservations. In this example,

subscriber 2 uses outside ports 5056-9087, and then 100-port blocks between 58000- 58999. Connections using ports 5056-9087 are not logged, while 10 log entries are created for ports 58000-58099, 58100-58199, 58200-58299, ..., 58900-58999.

In order to identify a subscriber behind a CGN (regardless of port allocation method), public safety agencies need to collect source address and port information from content provider log files. Thus, content providers are advised to log source address, source port, and timestamp for all log entries, per [RFC6302]. If a public safety agency collects such information from a content provider and reports abuse from 203.0.113.1, port 2001, the operator can reverse the mapping algorithm to determine that the internal IP address subscriber 1 has been assigned generated the traffic without consulting CGN logs (by correlating the internal IP address with DHCP /PPP lease connection records). If a second abuse report comes in for 203.0.113.1, port 58204, the operator will determine that port correlate with connection records, and determine that subscriber 2 generated the traffic (assuming that the public safety timestamp matches the operator timestamp. As noted in [RFC6269], accurate time-keeping (e.g., use of NTP or Simple NTP) is vital).

In this example, there are no log entries for the majority of subscribers, who only use pre-allocated ports. Only minimal logging would be needed for those few subscribers who exceed their pre-allocated ports and obtain extra bulk port assignments from the dynamic pool. Logging data for those users will include inside address, outside address, outside port range, and timestamp. See [I-D.ietf-behave-syslog-nat-logging] Section 3.1.4 for a detailed specification of the information required.

4.2.4. Additional Management Considerations

The CGN should provide a method for administrators to test the mapping function in both directions, i.e., enter an External IP Address + Port Number and receive the corresponding Internal IP Address and vice versa.

In order to be able to identify a subscriber based on observed external IPv4 address, port, and timestamp, an operator needs to know how the CGN was configured with regards to internal and external IP addresses, dynamic address pool factor, maximum ports per user, and reserved port range at any given time. Therefore, the operator needs to keep a record of the current configuration and changes to it. The record itself may be generated by the CGN, or may be retrieved from a router configuration management system. For auditing purposes, such records should be generated on a daily basis and checked for unauthorized or unintended changes.

4.3. Failover Considerations

Due to the deterministic nature of algorithmically-assigned translations, no additional logging is required during failover conditions provided that inside address ranges are unique within a given failover domain. Even when directed to a different CGN server, translations within the deterministic port range on either the primary or secondary server can be algorithmically reversed, provided the algorithm is known. Thus, if 100.64.0.1 port 3456 maps to 203.0.113.1 port 1000 on CGN 1 and 198.51.100.1 port 1000 on Failover CGN 2, an operator can identify the subscriber based on outside source address and port information.

Similarly, assignments made from the dynamic overflow pool need to be logged as described above, whether translations are performed on the primary or failover CGN.

4.4. Impact On IPv6 Transition

The solution described in this section is applicable to Carrier Grade NAT transition technologies (e.g. NAT444, DS-Lite, and NAT64). Native IPv6 will offer subscribers a better experience than CGN. However, many CPE devices only support IPv4. Likewise, as of July 2012, only approximately 4% of the top 1 million websites were available using IPv6. Accordingly, deterministic CGN should in no way be understood as making CGN a replacement for IPv6 service. The authors encourage [RFC6540] device manufacturers to consider and include IPv6 support. In the interim, however, CGN has already been deployed in some operator networks. Deterministic CGN will provide operators with the ability to quickly respond to public safety requests without requiring excessive infrastructure, operations, and bandwidth to support per-connection logging.

5. Security Considerations

The discussion which follows addresses an issue that is particularly relevant to the strategies described in Section 3 and Section 4 of this document. The security considerations applicable to NAT operation for various protocols as documented in, for example, [RFC4787] and [RFC5382] also apply to this proposal.

[RFC6056] summarizes the TCP port-guessing attack, by means of which an attacker can hijack one end of a TCP connection. One mitigating measure is to make the source port number used for a TCP connection less predictable. [RFC6056] provides various algorithms for this purpose.

As Section 3.1 of that RFC notes: "...provided adequate algorithms are in use, the larger the range from which ephemeral ports are selected, the smaller the chances of an attacker are to guess the selected port number." Conversely, the reduced range sizes proposed by the present document increase the attacker's chances of guessing correctly. This result cannot be totally avoided. However, mitigating measures to improve this situation can be taken both at port block assignment time and when selecting individual ports from the blocks that have been allocated to a given user.

At assignment time, one possibility is to assign ports as non-contiguous sets of values as proposed in [I-D.bajko-pripaddressign]. However, this approach creates a lot of complexity for operations, and the pseudo randomization can create uncertainty when the accuracy of logs is important to protect someone's life or liberty.

Alternatively, the NAT can assign blocks of contiguous ports. However, at assignment time the NAT could attempt to randomize its choice of which of the available idle blocks it would assign to a given user. This strategy has to be traded off against the desirability of minimizing the chance of conflict between what [RFC6056] calls "transport protocol instances" by assigning the most-idle block, as suggested in Section 3. A compromise policy might be to assign blocks only if they have been idle for a certain amount of time whenever possible, and select pseudorandomly between the blocks available according to this criterion. In this case it is suggested that the time value used be greater than the guard timing mentioned in Section 3, and that no block should ever be reassigned until it has been idle at least for the duration given by the guard timer.

Note that with the possible exception of cryptographically-based port allocations, attackers could reverse-engineer algorithmically-derived port allocations to either target a specific subscriber or to spoof traffic to make it appear to have been generated by a specific subscriber. However, this is exactly the same level of security that the subscriber would experience in the absence of CGN. CGN is not intended to provide additional security by obscurity.

While the block assignment strategy can provide some mitigation of the port guessing attack, the largest contribution will come from pseudo-randomization at port selection time. [RFC6056] provides a number of algorithms for achieving this pseudo-randomization. When the available ports are contained in blocks which are not in general consecutive, the algorithms clearly need some adaptation. The task is complicated by the fact that the number of blocks allocated to the user may vary over time. Adaptation is left as an exercise for the implementor.

6. IANA Considerations

This document makes no request of IANA.

7. Contributors

This document is the result of merging three separate Internet Drafts: the original Chen document (version -03), draft-tsou-behave-natx4-log-reduction-04, and draft-donley-behave-deterministic-cgn-07. Aside from the authors listed on the front of the present document, the following co-authors of the other two original drafts deserve credit for their contributions:

- o Weibo Li (China Telecom) and James Huang (Huawei) for their work on draft-tsou-behave-natx4-log-reduction, and
- o Chris Grundemann (Internet Society), Vikas Sarawat and Karthik Sundaresan (CableLabs), and Olivier Vautrin (Juniper) for their work on draft-donley-behave-deterministic-cgn.

8. Acknowledgements

The authors of draft-donley-behave-deterministic-cgn would like to thank the following people for their suggestions and feedback: Bobby Flaim, Lee Howard, Wes George, Jean-Francois Tremblay, Mohammed Boucadair, Alain Durand, David Miles, Andy Anchev, Victor Kuarsingh, Miguel Cros Cecilia, and Reinaldo Penno.

The authors of draft-tsou-behave-natx4-log-reduction have their own thanks to give. Mohamed Boucadair reviewed the initial document and provided useful comments to improve it. Reinaldo Penno, Joel Jaeggli, and Dan Wing provided comments on the subsequent version that resulted in major revisions. Serafim Petsis provided encouragement to publication after a hiatus of two years.

9. References

9.1. Normative References

- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 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.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.
- [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.

9.2. Informative References

- [APACHE_LOG_CONFIG]
The Apache Software Foundation, "http://httpd.apache.org/docs/2.4/mod/mod_log_config.html", 2013.
- [I-D.anderson-siit-dc]
Anderson, T., "Stateless IP/ICMP Translation in IPv6 Data Centre Environments (expired work in progress)", November 2012.
- [I-D.bajko-pripaddrassign]
Bajko, G., Savolainen, T., Boucadair, M., and P. Levis, "Port Restricted IP Address Assignment (expired Work in Progress)", March 2012.
- [I-D.ietf-behave-ipfix-nat-logging]
Sivakumar, S. and R. Penno, "IPFIX Information Elements for Logging NAT Events (Work in Progress)", February 2014.
- [I-D.ietf-behave-syslog-nat-logging]
Chen, Z., Zhou, C., Tsou, T., and T. Taylor, "Syslog Format for NAT Logging (Work in Progress)", January 2014.
- [I-D.ietf-pcp-port-set]
Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perrault, "Port Control Protocol (PCP) Extension for Port Set Allocation (Work in Progress)", November 2013.
- [I-D.ietf-software-4rd]
Despres, R., Jiang, S., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd) (Work in Progress)", October 2013.

- [I-D.ietf-software-map-dhcp]
Mrugalski, T., Troan, O., Dec, W., Farrer, I., Perrault, S., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for configuration of Software Address and Port Mapped Clients (Work in Progress)", March 2014.
- [I-D.ietf-software-map-t]
Li, X., Bao, C., Dec, W., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T) (Work in progress)", February 2014.
- [I-D.ietf-software-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP) (Work in Progress)", January 2014.
- [I-D.ietf-software-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 (Work in Progress)", November 2012.
- [I-D.penno-behave-rfc4787-5382-5508-bis]
Penno, R., Perrault, S., Kamiset, S., Boucadair, M., and K. Naito, "Network Address Translation (NAT) Behavioral Requirements Updates (expired Work in Progress)", January 2013.
- [I-D.shirasaki-nat444]
Yamagata, I., Shirasaki, Y., Nakagawa, A., Yamaguchi, J., and H. Ashida, "NAT444 (expired Work in Progress)", July 2012.
- [I-D.ietf-software-lw4over6]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture (Work in Progress)", March 2014.
- [POSTFIX_LOG_CONFIG]
"<http://www.postfix.org/postconf.5.html>", 2013.
- [RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.
- [RFC6191] Gont, F., "Reducing the TIME-WAIT State Using TCP Timestamps", BCP 159, RFC 6191, April 2011.
- [RFC6269.e46ua] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.
- [RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.
- [RFC6431] Boucadair, M., Levis, P., Bajko, G., Savolainen, T., and T. Tsou, "Huawei Port Range Configuration Options for PPP IP Control Protocol (IPCP)", RFC 6431, November 2011.
- [RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, April 2012.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.
- [RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.
- [SENDMAIL_LOG_CONFIG] O'Reilly, "Sendmail, 3rd Edition, Page 798", December 2002.
- [SSHD_LOG_CONFIG] "http://www.openbsd.org/cgi-bin/man.cgi?query=sshd_config&sektion=5", April 2013.

Appendix A. Configuration of Server Software to Log Source Port

A.1. Apache

The user can use LogFormat command to define a customized log format and use CustomLog command to apply that log format. "%a" and "%{remote}p" can be used in the format string to require logging the client's IP address and source port respectively. This feature is available since Apache version 2.1.

A detailed configuration guide can be found at [APACHE_LOG_CONFIG].

A.2. Postfix

In order to log the client source port, macro smtpd_client_port_logging should be set to "yes" in the configuration file. See [POSTFIX_LOG_CONFIG].

This feature has been available since Postfix version 2.5.

A.3. Sendmail

Sendmail has a macro \${client_port} storing the client port. To log the source port, the user can define some check rules. Here is an example which should be in the .mc configuration macro [SENDMAIL_LOG_CONFIG]:

```
LOCAL_CONFIG
Klog syslog
```

```
LOCAL_RULESETS
Slocal_check_mail
R $* $@ $(log Port_Stat ${client_addr} ${client_port} $)
```

This feature has been available since version 8.10.

A.4. sshd

SSHD_CONFIG(5) OpenBSD Programmer's Manual SSHD_CONFIG(5) NAME
 sshd_config - OpenSSH SSH daemon configuration file LogLevel Gives the verbosity level that is used when logging messages from sshd(8). The possible values are: QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2, and DEBUG3. The default is INFO. DEBUG and DEBUG1 are equivalent. DEBUG2 and DEBUG3 each specify higher levels of debugging output. Logging with a DEBUG level violates the privacy of users and is not recommended. SyslogFacility Gives the facility code that is used when logging messages from sshd(8). The possible values are: DAEMON, USER, AUTH, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7. The default is AUTH.

sshd supports logging the client IP address and client port when a client starts connection since version 1.2.2, here is the source code in sshd.c:

```
...
verbose("Connection from %.500s port %d", remote_ip, remote_port);
...
```

sshd supports logging the client IP address when a client disconnects, from version 1.2.2 to version 5.0. Since version 5.1 sshd supports logging the client IP address and source port. Here is the source code in sshd.c:

```
...
/* from version 1.2.2 to 5.0*/
verbose("Closing connection to %.100s", remote_ip);
...

/* since version 5.1*/
verbose("Closing connection to %.500s port %d",
remote_ip, remote_port);
```

In order to log the source port, the LogLevel should be set to VERBOSE [SSHD_LOG_CONFIG] in the configuration file:

LogLevel VERBOSE

A.5. Cyrus IMAP and UW IMAP

Cyrus IMAP and UW IMAP do not support logging the source port for the time being. Both software packages use syslog to create logs; it should not be too difficult to get source port logging supported by adding some new code.

Authors' Addresses

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

Email: phdgang@gmail.com

Tina Tsou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: tina.tsou.zouting@huawei.com

Chris Donley
CableLabs
858 Coal Creek Cir
Louisville, CO 80027
USA

Email: c.donley@cablelabs.com

Tom Taylor
PT Taylor Consulting
Ottawa, Ontario
Canada

Email: tom.taylor.stds@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 22, 2014

P. Fan
China Mobile
June 20, 2014

Managing Router Identifiers during IPv4 Sunset
draft-fan-sunset4-router-id-00

Abstract

This document describes problems of managing protocol identifiers when turning off IPv4 and migrating to IPv6 only network, with some potential solutions discussed.

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 22, 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. Problem Statement	2
3. Solution Ideas	2
4. Security Considerations	3
5. IANA Considerations	3
6. Acknowledgements	3
Author's Address	3

1. Introduction

There are many places in IETF protocols where a unique identifier is needed. An identifier is typically referred to as a router ID or system ID identifying a router/system running the protocol, and is traditionally designed to be a 32-bit number. Usually the IDs are required to be unique across some domain, but the actual value is not relevant. The value of IDs is often conventionally chosen to be an IPv4 address on the router, and in many implementations the IDs are even expressed in dotted decimal notation. There is some operational convenience of the common practice of tying the IDs to IP addresses:

1. A human-readable set of information is easy for network operators to deal with.
2. IDs can be auto-configured, saving the work of planning and assignment.
3. It is helpful to quickly perform diagnosis and troubleshooting, and easy to identify the availability and location of the identified router.

2. Problem Statement

In an IPv6 only network, there are no IP addresses that can be directly used to number an ID. IDs have to be planned individually to meet the uniqueness requirement, and the advantages of tying to IP addresses indicated in section 1 are lost.

3. Solution Ideas

If the ID is required to correspond to some information on the router or system, e.g. an IP address, the ID should be extended to meet the requirement; if the value is irrelevant and only needs to be unique, there has been suggestion about avoiding protocol change.

One can use some record keeping mechanisms, e.g. DNS or even text file, to associate IDs and IPv6 addresses to retain some of the

operational convenience, though extra record keeping does introduce additional work. Record keeping should be reliable enough so as to be reachable when a network problem occurs. Another option is to use some external provisioning system, e.g. network management system, to manage and allocate the IDs.

Another possible solution is to embed the ID into an IPv6 address, e.g. use a /96 IPv6 prefix and append it with a 32-bit long ID, then an ID is naturally tied to an IP address.

The above ideas require IDs be planned and generated in advance and meet the uniqueness requirement. IDs can be manually planned, possibly with some hierarchy or design rule, or can be created automatically. A simple way of automatic ID 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.

One can also hash an IPv6 address down to a value as ID. It is necessary to be able to override the hashed value, and desirable if hash is provided by the router implementation. 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.

4. Security Considerations

TBD.

5. IANA Considerations

None.

6. Acknowledgements

Thanks to Fred Baker, Shane Amante, David Farmer, Wes George for their valuable ideas in forming this document.

Author's Address

Peng Fan
China Mobile
32 Xuanwumen West Street, Xicheng District
Beijing 100053
P.R. China

Email: fanp08@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Best Current Practice
Expires: July 10, 2014

W. George
L. Howard
Time Warner Cable
January 6, 2014

IPv6 Support Within IETF work
draft-george-ipv6-support-02

Abstract

This document recommends that IETF formally require its standards work to be IP version agnostic or to explicitly include support for IPv6, with some exceptions. It further recommends that IETF revisit and update the previous attempts to review existing standards for IPv6 compliance. It makes this recommendation in order to ensure that it is possible to operate without dependencies on IPv4.

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 July 10, 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
1.1. Requirements Language	3
2. IPv6-only operation	3
2.1. Functional Parity with IPv4	3
2.2. IPv4 Sunset	4
3. IETF Actions	4
4. Requirements and Recommendations	6
5. Acknowledgements	7
6. IANA Considerations	7
7. Security Considerations	7
8. References	7
8.1. Normative References	7
8.2. Informative References	8
Authors' Addresses	9

1. Introduction

[RFC6540] gives guidance to implementers that in order to ensure interoperability and proper function after IPv4 exhaustion, IP-capable devices need to support IPv6, and cannot be reliant on IPv4, because global IPv4 exhaustion creates many circumstances where the use of IPv6 will no longer be optional. Since this is an IETF Best Current Practice recommendation, it is imperative that the results of IETF efforts enable implementers to follow that recommendation. This document provides recommendations and guidance as to how IETF itself should handle future work as it relates to Internet Protocol versions, and discusses the need for gap analyses on existing work.

When considering support for IPv4 vs IPv6 within IETF work, the general goal is to provide tools that enable networks and applications to operate seamlessly in any combination of IPv4-only, dual-stack, or IPv6-only as their needs dictate. However, as the IPv4 to IPv6 transition continues, it will become increasingly difficult to ensure interoperability and backward compatibility with IPv4-only networks and applications. As IPv6 deployment grows, IETF will naturally focus on features and protocols that enhance and extend IPv6, along with continuing work on items that are IP version agnostic. New features and protocols will not typically be introduced for use as IPv4-only. However, as of this document's writing, there is no formal requirement for all IETF work to support IPv6, either implicitly by being network-layer agnostic or explicitly by having an IPv6-specific implementation. Additionally, although reviews in RFC's 3789 [RFC3789] through RFC3796 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.

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. IPv6-only operation

At this document's writing, IPv6 has seen significant deployment. Most of these deployments are dual-stack, with IPv4 and IPv6 coexisting on the same networks. However, dual-stack is a waypoint in the transition from IPv4 to IPv6. The eventual end state is networks and end points that are IPv6-only. Some operators may take a long time to turn off IPv4, if they ever do, but the IETF must make sure that its standards can be deployed by even the first operators to turn off IPv4. Problems (and solutions) should be identified before they are encountered by the earliest adopters.

2.1. Functional Parity with IPv4

In order for IPv6-only operation to be realistic, IPv6 MUST have at least functional parity with IPv4. "Functional parity" means that any function that IPv4 enabled MUST also be enabled by IPv6. This does not mean that every feature that exists in IPv4 will exist in IPv6; different features may enable the same function. For instance, IPv4 supports some features that are no longer in use. In some cases it has not been practical to remove them in IPv4, or even to declare them historic, but it is unnecessary to carry them forward into IPv6. IPv6 also eliminates the need for some features that exist in IPv4; no effort to create unneeded features is required. Functional parity does not mean that all functions in IPv6 must also be possible in IPv4. Indeed, with IPv6 becoming the predominant protocol, new functionality should be developed in IPv6, and IETF effort SHOULD NOT be spent retrofitting features into the legacy protocol. The key at this point is 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.

2.2. IPv4 Sunset

Somewhat distinct from identifying the needed features for IPv6-only functional parity is the effort to identify what is necessary to disable or sunset IPv4 in a given network. Since many of the protocols in use today were designed to be fault-tolerant and very robust, actually removing them from a network once they are no longer needed is sometimes complex. Many implementations may not even have "off switches" because the assumption was that they would never be switched off in a normal network implementation. The Sunset4 Working Group was chartered to address these issues:

"The Working Group will point out specific areas of concern, provide recommendations, and standardize protocols that facilitate the graceful "sunsetting" of the IPv4 Internet in areas where IPv6 has been deployed. This includes the act of shutting down IPv4 itself, as well as the ability of IPv6-only portions of the Internet to continue to connect with portions of the Internet that remain IPv4-only. ... Disabling IPv4 in applications, hosts, and networks is new territory for much of the Internet today, and it is expected that problems will be uncovered including those related to basic IPv4 functionality, interoperability, as well as potential security concerns. The working group will report on common issues, provide recommendations, and, when necessary, protocol extensions in order to facilitate disabling IPv4 in networks where IPv6 has been deployed."

3. IETF Actions

In addition to a requirement for IPv6 support in the following section, this document recommends two major actions:

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). A document updating each of the below area-specific RFCs to identify which gaps have been addressed and which ones are either still outstanding or are now irrelevant may be an appropriate way to track this activity.

- o Internet Area [RFC3790]
- o Routing Area [RFC3791]
- o Security Area [RFC3792]
- o Sub-IP Area [RFC3793]

- o Transport Area [RFC3794]
- o Applications Area [RFC3795]
- o Operations and Management Area [RFC3796]

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 represents a significant amount of work in addition to IETF's existing workload, and there are basically two options for how to accomplish this significant document review. If existing IETF resources are to take on this work, one method would be for Area Directors to charter their existing Working Groups to undertake this review for relevant work, and charter their Directorates or other volunteers to review work that is not within the charter of any active Working Group. Another method would be to charter one (new or existing) Working Group or directorate to oversee this activity, with the assumption that the WG or directorate will pull in expertise from other areas and WGs as needed. The alternative is to use a similar model to the previous analysis in RFCs 3789-3796, in which ISOC funded dedicated resources whose primary duty was to complete this document audit.

RFC3789 [RFC3789] section 2 provides some guidance on methodology that can serve as a useful starting point for this effort.

"To perform this study, each class of IETF standards are investigated in order of maturity: Full, Draft, and Proposed, as well as Experimental. Informational and BCP RFCs are not addressed. RFCs that have been obsoleted by either newer versions or because they have transitioned through the standards process are not covered. RFCs which have been classified as Historic are also not included."

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 RFC 3789-96 were looking for "IPv4 dependency" or "usage of IPv4 addresses in standards". This document recommends a slightly more specific set of criteria for review: review should

include consideration of whether the specification can operate in an environment without IPv4. Reviews should include guidance on the use of 32-bit identifiers that are commonly populated by IPv4 addresses. Reviews should include consideration of protocols on which specifications depend or interact, to identify indirect dependencies on IPv4. Finally, reviews should consider how to migrate from an IPv4 environment to an IPv6 environment.

By necessity, this sort of gap analysis work is already happening in several places, e.g. draft-ietf-sunset4-gapanalysis [I-D.ietf-sunset4-gapanalysis], draft-george-mpls-ipv6-only-gap [I-D.george-mpls-ipv6-only-gap], and draft-klatsky-dispatch-ipv6-impact-ipv4 [I-D.klatsky-dispatch-ipv6-impact-ipv4]. These efforts are limited in scope, but may serve as a model for the larger effort necessary.

4. Requirements and Recommendations

While the primary goal of this effort is to ensure that existing IETF work has been properly evaluated and updated for IPv6-only support, ongoing focus is required for future work, whether via IESG evaluation, individual document reviews, or future WG charters. Due to the existing operational base of IPv4, it is not realistic to completely bar further work on IPv4 within the IETF at this time, nor to formally declare it historic. Until the time when IPv4 is no longer in wide use and/or declared historic, the IETF needs to continue to update IPv4-only protocols and features for vital operational or security issues. Similarly, the IETF needs to complete the work related to IPv4-to-IPv6 transition tools for migrating more traffic to IPv6. As the transition to IPv6-capable networks accelerates, it is also likely that some changes may be necessary in IPv4 protocols to facilitate decommissioning IPv4 in a way that does not create unacceptable impact to applications or users. These sorts of IPv4-focused activities, in support of security, transition, and decommissioning, should continue, accompanied by problem statements based on operational experience. Generally the focus should move away from IPv4-only work.

IETF should make updates to IPv4 protocols and features to facilitate IPv4 decommissioning

IETF work SHOULD explicitly support IPv6 or SHOULD be IP version agnostic (because it is implemented above the network layer), except IPv4-specific transition or address-sharing technologies.

IETF SHOULD NOT initiate new IPv4 extension technology development.

IETF work SHOULD function completely on IPv6-only nodes and networks, unless consensus exists that it is unnecessary to use a given feature or protocol on IPv6-only networks.

IETF SHOULD identify and update IPv4-only protocols and applications to support IPv6 unless consensus exists that it is unnecessary for a given feature or protocol.

5. Acknowledgements

Thanks to the following people for their comments: Jari Arkko, Ralph Droms, Scott Brim, Margaret Wasserman, Brian Haberman. Thanks also to Randy Bush, Mark Townsley, and Dan Wing for their discussion in IntArea WG at IETF 81 in Taipei, TW regarding transition technologies, IPv4 life extension, and IPv6 support.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document generates no new security considerations because it is not defining a new protocol. As existing work is analyzed for its ability to operate properly on IPv6-only networks, new security issues may be identified.

8. References

8.1. Normative References

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

8.2. Informative References

- [I-D.george-mpls-ipv6-only-gap]
George, W., Pignataro, C., Asati, R., Raza, K., Bonica, R., Papneja, R., Dhody, D., and V. Manral, "Gap Analysis for Operating IPv6-only MPLS Networks", draft-george-mpls-ipv6-only-gap-02 (work in progress), October 2013.
- [I-D.ietf-sunset4-gapanalysis]
Dionne, J., Perreault, S., Tsou, T., and C. Zhou, "Gap Analysis for IPv4 Sunset", draft-ietf-sunset4-gapanalysis-03 (work in progress), July 2013.
- [I-D.klatsky-dispatch-ipv6-impact-ipv4]
Klatsky, C., Shekh-Yusef, R., Hutton, A., and G. Salgueiro, "Interoperability Impacts of IPv6 Interworking with Existing IPv4 SIP Implementations", draft-klatsky-dispatch-ipv6-impact-ipv4-02 (work in progress), October 2013.
- [RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, April 2012.

Authors' Addresses

Wesley George
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA 20171
US

Phone: +1 703-561-2540
Email: wesley.george@twcable.com

Lee Howard
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA 20171
US

Phone: +1-703-345-3513
Email: lee.howard@twcable.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 7, 2015

S. Perreault
Jive Communications
W. George
Time Warner Cable
T. Tsou
Huawei Technologies (USA)
T. Yang
L. Li
China Mobile
JF. Tremblay
Viagenie
December 4, 2014

Turning off IPv4 Using DHCPv6 or Router Advertisements
draft-ietf-sunset4-noipv4-01

Abstract

This memo defines a new DHCPv6 option and a new Router Advertisement option to inform a dual-stack host or router that IPv4 can 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 June 7, 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	2
2. Terminology	3
3. Problems Being Addressed	3
3.1. Load on DHCPv4 Server and Relay	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	6
5. The No-IPv4 DHCPv6 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	11
9.1. Normative References	11
9.2. Informative References	11
9.3. URIs	11
Appendix A. Test Results of Terminals Behavior	11
Authors' Addresses	13

1. Introduction

When a dual-stack host makes a DHCPv4 request, it typically interprets the absence of a response as a failure condition. This may cause operational problems when deploying an IPv6-only network. Providing a way to inform hosts and routers that IPv4 is not available would prevent such problems and allow for smoother deployments.

One situation where problems arise is with a dual-stack home router provisioned with an IPv6-only WAN connection. It typically assigns an IPv4 address to its LAN interface, starts services on that interface and hands 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 install a default route pointing to the router and behave 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 if it would be desirable to turn it off, especially for applications that do not implement Happy Eyeballs [RFC6555].

Another situation relates to the load on DHCPv4 servers and relays. In large dual-stack network (LAN, WLAN), thousands of hosts, including mobile phones, may generate a significant amount of traffic by attempting to contact a DHCP server. If the servers and relays are configured in IPv6-only, the dual-stack or IPv4-only clients will broadcast DHCPDISCOVER messages endlessly, creating a DDOS-like attack on the network. This scenario has also been briefly described for DHCPv6 in [RFC7083]. Although DHCP mandates a exponential backoff, it is limited to 64 seconds, which may still generate significant traffic (see section 4.1 of [RFC2131]). Various operating systems also implement the backoff algorithms in different ways, or not at all, with different limit values. Some test results for a few popular operating systems are available in appendix.

A new mechanism is needed to indicate the absence of IPv4 connectivity. Considering the end goal is turn off all IPv4 connectivity, the chosen mechanism should be transported over IPv6. Therefore, this document 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. Problems Being Addressed

3.1. Load on DHCPv4 Server and Relay

When a DHCPv4 server or relay is present but intentionally does not react to DHCPDISCOVERs, the aggregated traffic generated by a large number of dual-stack hosts can represent a significant bandwidth load. This scenario is encountered with an ISP serving multiple types of subscribers where some are provisioned for IP4 service and others are not. It might not be feasible for operational reasons to block the useless requests before they reach the DHCPv4 servers, for example if the DHCPv4 servers themselves are the only ones with the knowledge of which nodes should or should not get an IPv4 address.

3.2. Bandwidth Consumption

In addition to the useless load on the DHCPv4 servers, the above scenario could also consume a significant amount of bandwidth, especially if the aggregated traffic from many clients goes through a low-bandwidth link or through a wireless 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 an autoconfigured IPv4 Link-Local Addresses (LLA) to send IPv4 packets on the LAN. In an IPv6-only environment, this behavior may waste a significant amount 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 to turn off IPv4. An equivalent option could conceivably be created for DHCPv4. The pros and cons are discussed below. 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 the need for a DHCPv4 option is not as obvious. More feedback on this topic would be appreciated.

4.2. DHCPv6 vs RA

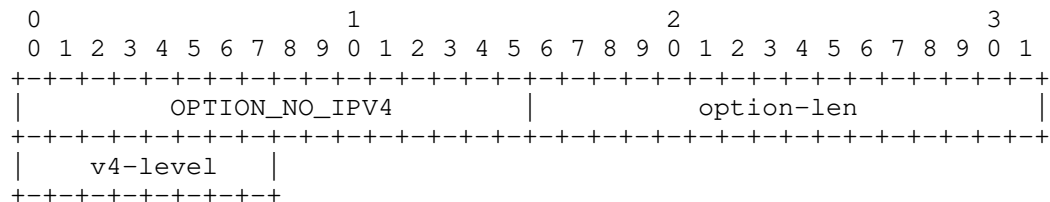
Both DHCPv6 and RA-based solutions are presented in this draft. It is expected that the working group will decide whether both solutions, only one, or none are desirable.

5. The No-IPv4 DHCPv6 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).

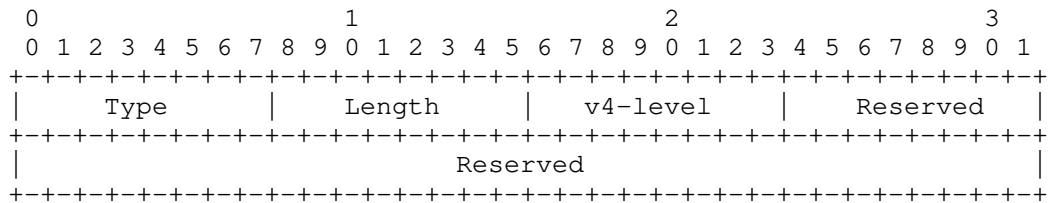
option-len 1.

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 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. (JFT: What happens if the timer below is not implemented and IPv4 completes before IPv6? Maybe we could recommend to run IPv6 provisioning first when `OPTION_NO_IPV4` is supported.)

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.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.
- [RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, November 2013.

9.3. URIs

- [1] <http://www.iana.org/assignments/dhcpv6-parameters>

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 various 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
Jive Communications
Quebec, QC
Canada

Email: sperreault@jive.com

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 Lianyan
China Mobile
32, Xuanwumenxi Ave.
Xicheng District, Beijing 100053
China

Email: lilianyan@chinamobile.com

Jean-Francois Tremblay
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Phone: +1 418 656 9254
Email: jean-francois.tremblay@viagenie.ca
URI: <http://viagenie.ca>