

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 10, 2012

J. Arkko
A. Keranen
Ericsson
February 7, 2012

Experiences from an IPv6-Only Network
draft-arkko-ipv6-only-experience-05

Abstract

This document discusses our experiences from moving a small number of users to an IPv6-only network, with access to the IPv4-only parts of the Internet via a NAT64 device. The document covers practical experiences as well as road blocks and opportunities for this type of a network setup. The document also makes some recommendations about where such networks are applicable and what should be taken into account in the network design. The document also discusses further work that is needed to make IPv6-only networking applicable in all environments.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 10, 2012.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
2. Technology and Terminology	4
3. Network Setup	4
3.1. The IPv6-Only Network	5
3.2. DNS Operation	6
4. General Experiences	7
5. Experiences with IPv6-Only Networking	9
5.1. Operating Systems	9
5.2. Programming Languages and APIs	10
5.3. Instant Messaging and VoIP	11
5.4. Gaming	12
5.5. Music Services	13
5.6. Appliances	13
5.7. Other Differences	13
6. Experiences with NAT64	13
6.1. IPv4 Address Literals	14
6.2. Comparison of Web Access via NAT64 to Other Methods	15
7. Future Work	15
8. Conclusions and Recommendations	16
9. Security Considerations	18
10. IANA Considerations	18
11. References	18
11.1. Normative References	18
11.2. Informative References	19
Appendix A. Acknowledgments	20
Authors' Addresses	20

1. Introduction

This document discusses our experiences from moving a small number of users to an IPv6-only network, with access to the IPv4-only parts of the Internet via a NAT64 device. This arrangement has been done with a permanent change in mind rather than as a temporary experiment, involves both office and home users, heterogeneous computing equipment, and varied applications. We have learned both practical details, road blocks and opportunities, as well as more general understanding of when such a configuration can be recommended and what should be taken into account in the network design.

The networks involved in this setup have been in dual-stack mode for considerable amount of time, in one case for over ten years. Our IPv6 connectivity is stable and in constant use with no significant problems. Given that the IETF is working on technology such as NAT64 [RFC6144] and several network providers are discussing the possibility of employing IPv6-only networking, we decided to take our network beyond the "comfort zone" and make sure that we understand the implications of having no IPv4 connectivity at all. This also allowed us to test a NAT64 device that is being developed by Ericsson.

The main conclusion is that it is possible to employ IPv6-only networking, though there are a number of issues such as lack of IPv6 support in some applications and bugs in untested parts of code. As a result, dual-stack [RFC4213] remains as our recommended model for general purpose networking at this time, but IPv6-only networking can be employed by early adopters or highly controlled networks. The document also suggests actions to make IPv6-only networking applicable in all environments. In particular, resolving problems with a few key applications would have a significant impact for enabling IPv6-only networking for large classes of users and networks. It is important that the Internet community understands these deployment barriers and works to remove them.

The rest of this document is organized as follows. Section 2 introduces some relevant technology and terms, Section 3 describes the network setup, Section 4 discusses our general experiences, Section 5 discusses experiences related to having only IPv6 networking available, and Section 6 discusses experiences related to NAT64 use. Finally, Section 7 presents some of our ideas for future work, Section 8 draws conclusions and makes recommendations on when and how one should employ IPv6-only networks, and Section 9 discusses relevant security considerations.

2. Technology and Terminology

In this document, the following terms are used. "NAT44" refers to any IPv4-to-IPv4 network address translation algorithm, both "Basic NAT" and "Network Address/Port Translator (NAPT)", as defined by [RFC2663].

"Dual-Stack" refers to a technique for providing complete support for both Internet protocols -- IPv4 and IPv6 -- in hosts and routers [RFC4213].

"NAT64" refers to a Network Address Translator - Protocol Translator defined in [RFC6144], [RFC6145], [RFC6146], [RFC6052], [RFC6147], and [RFC6384].

3. Network Setup

We have tested IPv6-only networking in two different network environments: office and home. In both environments all hosts had normal dual-stack native IPv4 and IPv6 Internet access already in place. The networks were also already employing IPv6 in their servers and DNS records. Similarly, the network was a part of whitelisting arrangement to ensure that IPv6-capable content providers would be able to serve their content to the network over IPv6.

The office environment has heterogeneous hardware with PCs, laptops, and routers running Linux, BSD, Mac OS X, and Microsoft Windows operating systems. Common uses of the network include e-mail, Secure Shell (SSH), web browsing, and various instant messaging and Voice over IP (VoIP) applications. The hardware in the home environment consists of PCs, laptops and a number of server, camera, and sensor appliances. The primary operating systems in this environment are Linux and Microsoft Windows operating systems. Common applications include web browsing, streaming, instant messaging and VoIP applications, gaming, file storage, and various home control applications. Both environments employ extensive firewalling practices, and filtering is applied for both IPv4 and IPv6 traffic. However, firewall capabilities, especially with older versions of firewall software, dictate some differences between the filtering applied for IPv4 and IPv6 since some features commonly supported for IPv4 were not yet implemented for IPv6. In addition, in the home environment the individual devices are directly accessible from the Internet on IPv6 (on select protocols such as SSH) but not on IPv4 due to lack of available public IPv4 addresses.

In both environments, volunteers had the possibility to opt-in for

the IPv6-only network. The number of users is small: there are roughly five permanent users and a dozen users who have been in the network at least for some amount of time. Each user had to connect to the IPv6-only wired or wireless network, and depending on their software, possibly configure their computer by indicating that there is no IPv4 and/or setting DNS server addresses. The users were also asked to report their experiences back to the organizers.

3.1. The IPv6-Only Network

The IPv6-only network was provided as a parallel network on the side of the already existing dual-stack network. It was important to retain the dual-stack network for the benefit of those users who did not decide to opt-in and also because we knew that there were some IPv4-only devices in the network. A separate wired access network was created using Virtual Local Area Networks (VLANs). This network had its own IPv6 prefix. A separate wireless network, bridged to the wired network, was also created. In our case, the new wireless network required additional access point hardware in order to accommodate advertising multiple wireless networks. The simple access point model that we employed in these networks did not allow this on a single device, although many other access points support this. All the secondary infrastructure resulted in some additional management burden and cost, however. An added complexity was that the home network already employed two types of infrastructure, one for family members and another one for visitors. In order to duplicate this model for the IPv6-only network there are now four separate networks, with several access points on each.

A stateful NAT64 [RFC6146] with integrated DNS64 was installed on the edge of the IPv6-only networks. No IPv4 routing or Dynamic Host Configuration Protocol (DHCP) was offered on these networks. The NAT64 device sends Router Advertisements (RAs) [RFC4861] from which the hosts learn the IPv6 prefix and can automatically configure IPv6 addresses for them. Each new IPv6-only network needed one new /64 prefix to be used in these advertisements. In addition, each NAT64 device needed another /64 prefix to be used for the representation of IPv4 destinations in the IPv6-only network. As a result, one IPv6-only network requires /63 of address space. This space was easily available in our networks, as IPv6 allocations are on purpose made in sufficiently large blocks. Additional address space needs can be accommodated from the existing block without registry involvement. Another option would have been to use the Well-Known Prefix [RFC6052] for the representation of IPv4 destinations in the IPv6-only network. In any case, the prefixes have to be listed in the intra-domain routing system so that they can be reached. In one case the increase from one block to multiple also made it necessary to employ an improved routing configuration. In addition to routing, the new

prefixes have to be listed in the appropriate firewall rules.

Setting up NAT64 and DNS64 by itself is easy and can be done quickly by experienced network manager. However, when duplicate infrastructure is needed for dual-stack and IPv6-only networks, the additional switches, cables, access points, etc., will take some amount of installation effort. In addition, if whitelisting agreements or IPv6 ISP connectivity is needed, setting these up requires negotiations with external partners.

3.2. DNS Operation

Router Advertisements are used to carry DNS Configuration options [RFC6106], listing the DNS64 as the DNS server the hosts should use. In addition, aliases were added to the DNS64 device to allow it to receive packets on the well-known DNS server addresses that Windows operating systems use (fec0:0:0:ffff::1, fec0:0:0:ffff::2, and fec0:0:0:ffff::3). At a later stage support for stateless DHCPv6 [RFC3736] was added. We do recommend enabling RFC 6106, well-known addresses, and stateless DHCPv6 in order to maximize the likelihood of different types of IPv6-only hosts being able to use DNS without manual configuration. DNS server discovery was never a problem in dual-stack networks, because DNS servers on the IPv4 side can easily provide IPv6 information (AAAA records) as well. With IPv6-only networking, it becomes crucial that the local DNS server can be reached via IPv6 as well. This is in principle exactly same as needing IPv4-based DNS and DNS discovery in IPv4-only networks. However, in IPv6 the discovery mechanisms are somewhat more complicated because there are several alternative techniques.

When a host served by the DNS64 asks for a domain name that does not have an AAAA (IPv6 address) record, but has an A (IPv4 address) record, an AAAA record is synthesized from the A record (as defined for DNS64 in [RFC6147]) and sent in the DNS response to the host. IP packets sent to this synthesized address are routed via the NAT64, translated to IPv4 by the NAT64, and forwarded to the queried host's IPv4 address; return traffic is translated back from IPv4 to IPv6 and forwarded to the host behind the NAT64 (as described in [RFC6144]). This allows the hosts in the IPv6-only network to contact any host in the IPv4 Internet as long as the hosts in the IPv4 Internet have DNS address records.

The NAT64 devices have standard dual-stack connectivity and their DNS64 function can use both IPv4 and IPv6 when requesting information from DNS. A destination that has both an A and AAAA records is not treated in any special manner, because the hosts in the IPv6-only network can contact the destination over IPv6. Destinations with only an A record will be given a synthesized AAAA record as explained

above. However, in one of our open visitor networks that is sharing the infrastructure with the home network we needed a special arrangement. Currently, the home network obtains its IPv6 connectivity through a tunnel via the office network, and it is undesirable to allow outsiders using the visitor network to generate traffic through the office network, even if the traffic is just passing by and forwarded to the IPv6 Internet. As a result, in the visitor network there is a special IPv6-only to IPv4-only configuration where the DNS64 never asks for AAAA records and always generates synthesized records. Therefore no traffic from the visitor network, even if it is destined to the IPv6 Internet, is routed via the office network but traffic from the home network can still use the IPv6 connectivity provided by the office network.

Note: This configuration may also be useful for other purposes. For instance, one drawback of standard behavior is that if a destination publishes AAAA records but has bad IPv6 connectivity, the hosts in the IPv6-only network have no fallback. In the dual-stack model a host can always try IPv4 if the IPv6 connection fails. In the special configuration IPv6 is only used internally at the site but never across the Internet, eliminating this problem. This is not a recommended mode of operation, but it is interesting to note that it may solve some issues.

Note that in NAT64 (unlike in its older variant [RFC4966]) it is possible to decouple the packet translation, IPv6 routing, and DNS64 functions. Since clients are configured to use a DNS64 as their DNS server, there is no need for having an Application Layer Gateway (ALG) on the path sniffing and spoofing DNS packets. This decoupling possibility was used by one of our users, as he is outside of our physical network and wants to communicate directly on IPv6 where it is possible without having to go through our central network equipment. His DNS queries go to our DNS64 and to establish communications to an IPv4 destination our central NAT64 is used. If there is a need to translate some packets, these packets find the translator device through normal IPv6 routing means since the synthesized addresses have our NAT64's prefix. However, for non-synthesized IPv6 addresses the packets are routed directly to the destination.

4. General Experiences

Based on our experiences, it is possible to live (and work) with an IPv6-only network. For instance, at the time of this writing, one of the authors has been in an IPv6-only network for about a year and a half and has had no major problems. Most things work well in the new environment; for example, we have been unable to spot any practical

difference in the web browsing (HTTP and HTTPS) experience. Also e-mail, software upgrades, operating system services, many chat systems and media streaming work well. On certain Symbian mobile handsets that we tried all applications work even on an IPv6-only network. In another case with Android operating system, all the basic applications worked without problems. In order to make the latter handset architecture support IPv6-only networks, however, a small change was needed in the operating system so that it could discover IPv6-only DNS servers.

However, in general there is some pain involved and thus IPv6-only networking is not suitable for everyone just yet. Switching IPv4 off does break many things as well. Some of the users in our environment left due to these issues, as they missed some key feature that they needed from their computing environment. These issues fall in several categories:

Bugs

We saw many issues that can be classified as bugs, likely related to so few people having tried the software in question in an IPv6-only network. For instance, some operating system facilities support IPv6 but have annoying problems that are only uncovered in IPv6-only networking.

Lack of IPv6 Support

We also saw many applications that do not support IPv6 at all. These range from minor, old tools (such as the Unix `dict(1)` command) to major applications that are important to our users (such as Skype) and even to entire classes of applications (many games have issues). As our experiment continued, we have seen improvements in some areas, such as gaming.

Protocol, Format, and Content Problems

There are many protocols that carry IP addresses in them, and using these protocols through a translator can lead to problems. In our current network setup we did not employ any ALGs except for FTP [RFC6384]. However, we have observed a number of protocol issues with IPv4 addresses. For instance, some instant messaging services do not work due to this. Finally, content on some web pages may refer to IPv4 address literals (i.e., plain IP addresses instead of host and domain names). This renders some links inaccessible in an IPv6-only network. While this problem is easily quantifiable in measurements, the authors have run into it only a couple of times during real-life web browsing.

Firewall Issues

We also saw a number of issues related to lack of features in IPv6 support in firewalls. In particular, while we did not experience any Maximum Transmission Unit (MTU) and fragmentation problems in our networks, there is potential for generating problems, as the support for IPv6 fragment headers is not complete in all firewalls and the NAT64 specifications call for use of the fragment header (even in situations where fragmentation has not yet occurred, e.g., if an IPv4 packet that is not a fragment does not have the Don't Fragment (DF) bit set).

In general, most of the issues relate to poor testing and lack of IPv6 support in some applications. IPv6 itself and NAT64 did not cause any major issues for us, once our setup and NAT64 software was stable. In general, the authors feel that with the exception of some applications, our experience with translation to reach the IPv4 Internet has been equal to our past experiences with NAT44-based Internet access. While translation implies loss of end-to-end connectivity, in practice direct connectivity has not been available to the authors in the IPv4 Internet either for a number of years.

It should be noted that the experience with a properly configured set of ALGs and work-arounds such as proxies may be different. Some of the problems we encountered can be solved through these means. For instance, a problematic application can be configured to use a proxy that in turn has both IPv4 and IPv6 access.

5. Experiences with IPv6-Only Networking

The overall experience was as explained above. The remainder of this section discusses specific issues with different operating systems, programming languages, applications, and appliances.

5.1. Operating Systems

Even operating systems have some minor problems with IPv6. For example, in Linux Router Advertisement (RA) information was not automatically updated when the network changes while the computer is on and required an unnecessary suspend/resume cycle to restore its proper state. We have also had issues with the `rdnssd` daemon, which first does not come as a default feature in Ubuntu and does not always appear to work reliably. To resolve these issues we had to configure the network manager to use a specific server address. Later, a new version of the Linux distribution that we used solved these problems, even if some problems still remained. For instance, in the latest Ubuntu Long Term Support release (10.04) we have

experienced that the network manager by default returns to an available IPv4 wireless network even if there is a previously used IPv6-only network available and the IPv4 network has no global connectivity before a web-based login is completed.

In Mac OS X (Snow Leopard) the network manager needed to be explicitly told to not expect IPv4. A more annoying issue was that in order to switch between an IPv6-only and IPv4-only networks, these settings had to be manually changed, making it undesirable for Mac OS X users to employ IPv6-only networks.

Also on Microsoft Windows 7 we experienced problems when relying on default, well-known DNS server addresses: without manual configuration, the host was unable to use the DNS addresses, even though the system displays them as current DNS server addresses.

Latest versions of the Android operating system support IPv6 on its wireless LAN interface, but due to lack of DNS discovery mechanisms, this does not work in IPv6-only networks. We corrected this, however, and prototype phones in our networks work now well even in an IPv6-only environment. This change, DNS Discovery Daemon (DDD) now exists as open source software. Interestingly, all applications that we have tried so far seem to work without problems with IPv6-only connectivity, though no exhaustive testing was done, nor did we try known troublesome applications.

While all these operating systems (or their predecessors) have supported IPv6 already for a number of years, these kind of small glitches seem to imply that they have not been thoroughly tested in networks lacking IPv4 connectivity. At the very least their usability leaves something to be desired.

5.2. Programming Languages and APIs

For applications to be able to support IPv6, they need access to the necessary APIs. Luckily, IPv6 seems to be well supported by majority of the commonly used APIs. The Perl programming language used to be an exception with only partial IPv6 support up to the version 5.14 (released May 14th 2011). This version finally includes full IPv6 support also in the core libraries and older modules are being updated as well. With previous versions of Perl, while IPv6 socket support is available as an extension module, it may not be possible to install this module without administrative rights. This has also resulted in other networking core libraries (such as FTP and SMTP) not being able to fully support IPv6 and thus many existing Perl programs using network functionality may not work properly in an IPv6-only environment.

5.3. Instant Messaging and VoIP

By far the biggest complaint from our group of users was that Skype stopped working. In some environments even Skype can be made to work through a proxy configuration, and this was verified in our setting but not used as a permanent solution. More generally, we tested a number of instance messaging applications in an IPv6-only network with NAT64 and the test results can be found from Table 1. The versions used in the tests were the latest versions available on summer 2010.

SYSTEM	STATUS
Facebook on the web (http)	OK
Facebook via a client (xmpp)	OK
Jabber.org chat service (xmpp)	OK
Gmail chat on the web (http)	OK
Gmail chat via a client (xmpp)	OK
Google Talk client	NOT OK
AIM (AOL)	NOT OK
ICQ (AOL)	NOT OK
Skype	NOT OK
MSN	NOT OK
Webex	NOT OK
Sametime	OK (NOW)

Table 1. Instant Messaging Applications in an IPv6-Only Network

Packet tracing revealed that the issues in AIM, ICQ, and MSN appear to be related to passing literal IPv4 addresses in the protocol. It remains to be determined whether this can be solved through configuration, proxies, or ALGs. The problem with the Google Talk client is that the software does not support IPv6 connections at this moment. We are continuing our tests with additional applications, and we have also seen changes over time. For instance, a new version of Sametime suddenly started working with IPv6-only networks, presumably due to the new version being more careful with the use of DNS names as opposed to IPv4 addresses. One problem in running these tests is to ensure that we can distinguish IPv6 and NAT64 issues from other issues, such as a generic issue on a given operating system platform.

Some of these problems are solvable, however. For instance, we used localhost as a proxy for Skype, and then used SSH to tunnel to an external web proxy, bypassing Skype's limitations with regards to connecting to IPv6 destinations or even IPv6 proxies.

5.4. Gaming

Another class of applications that we tried was games. We tried both web-based gaming and standalone gaming applications that have a "network" / "Internet" or "LAN" gaming modes. The results are shown in Table 2.

SYSTEM	STATUS
Web-based (e.g. armorgames)	OK
Runescape (on the web)	NOT OK
Flat out 2	NOT OK
Battlefield	NOT OK
Secondlife	NOT OK
Guild Wars	NOT OK
Age of Empires	NOT OK
Star Wars: Empire at War	NOT OK
Crysis	NOT OK
Lord of the Rings: Conquest	NOT OK
Rome Total War	NOT OK
Lord of the Rings: Battle for Middle Earth 2	NOT OK

Table 2. Gaming Applications in an IPv6-Only Network

Most web-based games worked well, as expected from our earlier good general web experience. However, we were also able to find one web-based game that failed to work (Runescape). This particular game is a Java application that fails on an attempt to perform a HTTP GET request. The reason remains unclear, but a likely theory is the use of an IPv4-literal in the application itself.

The experience with standalone games was far more discouraging. Without exception all games failed to enable either connections to ongoing games in the Internet or even LAN-based connections to other computers in the same IPv6-only LAN segment. This is somewhat surprising, and the results require further verification. Unfortunately, the games provide no diagnostics about their operation, so it is hard to guess what is going on. It is possible that their networking code employs older APIs that cannot use IPv6 addresses [RFC4038]. The inability to provide any LAN-based connectivity is even more surprising, as this must mean that they are unable to use IPv4 link local connectivity, which should have been available to the devices (IPv4 was not blocked; just that no DHCP answers were provided on IPv4).

While none of the standalone games we tested on summer 2010 were IPv6-capable, the situation has improved during the experiment. For instance, a popular on-line game, World of Warcraft, now has IPv6

support in its latest version and some of the older games that have been re-released as open source (e.g., Quake) have been patched IPv6-capable by the open source community.

5.5. Music Services

Most of the web-based music services appear to work fine, presumably because they employ TCP and HTTP as a transport. One notable exception is Spotify, which requires communication to specific IPv4 addresses. A proxy configuration similar to the one we used for Skype makes it possible to use Spotify as well.

5.6. Appliances

There are also problems with different appliances such as webcams. Many of them do not support IPv6 and hence will not work in an IPv6-only network. Also not all firewalls support IPv6. Or even if they do, they may still experience issues with some aspects of IPv6 such as fragments.

Some of these issues are easily solved when the appliance works as a server, such as what most webcams and our sensor gateway devices do. We placed the appliance in the IPv4 part of the network (in this case, in private address space), added its name to the local DNS, and simply allowed devices from the IPv6-only network reach it through NAT64.

5.7. Other Differences

One thing that becomes simplified in an IPv6-only network is source address selection [RFC3484]. As there is no IPv4 connectivity, the host only needs to consider its IPv6 source address. For global communications there is typically just one possible source address.

Some networks that advertise IPv6 addresses in their DNS records have in reality some problems. For instance, a popular short URL forwarding service has advertised a deprecated IPv4-compatible IPv6 address [RFC4291] in its AAAA record, making it impossible for this site to be reached unless either IPv4 or NAT64 translation to an IPv4 destination is used.

6. Experiences with NAT64

After correcting some initial bugs and stability issues, the NAT64 operation itself has been relatively problem free. There have been no unexplained DNS problems or lost sessions. With the exception of the specific applications mentioned above and IPv4 literals, the user

experience has been in line with using IPv4 Internet through a NAT44 device. These failures with the specific applications are clearly very different from the IPv4 experience, however.

The rest of this section discusses our measurements on specific issues. These tests and measurements were performed during year 2011 and present a snapshot of the situation on that time. More up-to-date measurement information can be found from various on-line tools such as [HE-IPv6].

6.1. IPv4 Address Literals

While browsing in general works, IPv4 literals embedded in the HTML code may break some parts of the web pages when using IPv6-only access. This happens because the DNS64 can not synthesize AAAA records for the literals since the addresses are not queried from the DNS. Luckily, the IPv4 literals seem to be fairly rarely encountered, at least so that they would be noticed, with regular web surfing. The authors have run into this issue only few times during the entire experiment. Only two of those cases had a practical impact (in YouTube, some of the third-party applications for downloading content did not work and one hotel's web page had a literal link to its reservation system).

We have attempted to measure the likelihood of running into an IPv4 literal in the web. To do this, we took the top 1,000 and 10,000 web sites from the Alexa popular web site list. With 1,000 top sites, 0.2% needed an IPv4 literal to render all components in their top page (e.g., images, videos, JavaScript, and Cascading Style Sheet (CSS) files). With 10,000 top sites, this number increases to 2%.

However, it is not clear what conclusions can be made about this. It is often the case that there are unresolvable or inaccessible components on a web page anyway for various reasons, and to understand the true impact we would have to know how "important" a given page component was. Also, we did not measure the number of links with IPv4 literals on these pages, nor did we attempt to search the site in any thorough manner for these literals.

As noted, personal anecdotal evidence says that IPv4 literals are not a big problem. But clearly, cleaning the most important parts of the web from IPv4 literals would be useful. With tools such as the popular web site list, some user pressure, and co-operation from the content providers the most urgent part of the problem could hopefully be solved as a one-time effort. While IPv4 literals still exist in the web, using a suitable HTTP proxy (e.g., [I-D.wing-behave-http-ip-address-literals]) can help to cope with them.

6.2. Comparison of Web Access via NAT64 to Other Methods

We also compared how well the web works behind a NAT64 compared to IPv4-only and native IPv6 access. For this purpose, we used wget to go through the same top web site lists as described in Section 6.1, again downloading everything needed to render their front page. The tests were repeated and average failure rate was calculated over all of the runs. Separate tests were conducted with an IPv4-only network, an IPv6-only network, and an IPv6-only network with NAT64.

When accessed with the IPv4-only network, our tests show that 1.9% of the sites experienced some sort of error or failure. The failure could be that the whole site was not accessible, or just that a single image (e.g., an advertisement banner) was not loaded properly. It should also be noted that access through wget is somewhat different from a regular browser: some web sites refuse to serve content to wget, browsers typically have DNS heuristics to fill in "www." in front of a domain name where needed, and so on. In addition to missing advertisement banners, temporary routing glitches and other mistakes, these differences also help to explain the reason for the high baseline error rate in this test. It should also be noted that variations in wget configuration options produced highly different results, but we believe that the options we settled on bear closest resemblance to real world browsing.

When we tried to access the same sites with native IPv6 (without NAT64), 96% of the sites failed to load correctly. This was as expected, given that most of the Internet content is not available on IPv6. The few exceptions included, for instance, sites managed by Google.

When the sites were accessed from the IPv6-only network via a NAT64 device, the failure rate increased to 2.1%. Most of these failures appear to be due to IPv4 address literals, and the increased failure rate matches that of IPv4 literal occurrence in the same set of top web sites. With the top 10,000 sites the failure rate with NAT64 increases similarly to our test on IPv4 address literals.

7. Future Work

One important set of measurements remains for future work. It would be useful to understand the effect of DNS64 and NAT64 to response time and end-to-end communication delays. Some users have anecdotal reports of slow web browsing response times, but we have been unable to determine if this was due to the IPv6-only network mechanisms or for some other reason. Measurements on pure DNS response times and packet round-trip delays does not show a significant difference to a

NAT44 environment. It would be particularly interesting to measure delays in the context of dual-stack vs. NAT64-based IPv6-only networking. When using dual-stack, broken IPv6 connectivity can be repaired by falling back to IPv4 use. With NAT64, this is not always possible as discussed in Section 3.2.

Also more programs, especially VoIP and Peer-to-Peer (P2P) applications should be tested with NAT64. In addition, tunneling and mobility protocols should be tested and especially Virtual Private Network (VPN) protocols and applications would deserve more thorough investigation.

8. Conclusions and Recommendations

The main conclusion is that it is possible to employ IPv6-only networking. For large classes of applications there are no downsides or the downsides are negligible. We have been unable to spot any practical difference in the web browsing experience, for instance. And IPv6 usage -- be it in dual-stack or IPv6-only form -- comes with inherent advantages, such as enabling direct end-to-end connectivity. In our case, we employed this by enabling direct connectivity to devices in a home network from anywhere in the (IPv6) Internet. There are, however, a number of issues as well, such as lack of IPv6 support in some applications or bugs in untested parts of the code.

Our experience with IPv6-only networking confirms that dual stack should still be our recommended model for general purpose networking at this point of time. However, IPv6-only networking can be employed by early adopters or highly controlled networks. One example of such controlled network is a mobile network with operator-driven selection of handsets. For instance, on some handsets that we tested, we were unable to see any functional difference between IPv4 and IPv6, today.

Our recommendations apply at the present time. With effort and time, deployment barriers can be removed and IPv6-only networking becomes applicable in all networking situations.

Some of the improvements are already in process in the form of new products and additional IPv6 support. For instance, we expect that the handset market will have a much higher number of IPv6-capable devices in the near future. But some of the changes do not come without the community spending additional effort. We have identified a number of actions that should be taken to improve the state of IPv6-only networking. These include:

DNS Discovery

The state of DNS discovery continues to be one of the main barriers for easy adoption of IPv6-only networking. Since DNS discovery is not a problem in dual-stack networking, there has been too little effort in testing and deploying the necessary components. For instance, it would be useful if RA-based DNS discovery came as a standard feature and not as an option in Linux distributions. Our hope is that recent standardization of the RA-based DNS discovery at the IETF will help this happen. Similar issues face other operating systems. The authors believe that at this time, prudent operational practices call for maximizing the number of offered automatic configuration mechanisms on the network side. It might be useful for an IETF document to provide guidance on operating DNS in IPv6-only networks.

Network Managers

Other key software components are the various network management and attachment tools in operating systems. These tools generally have the required functionality, but do not always appear to have been tested very extensively on IPv6, or let alone IPv6-only networks. Further work is required here.

Firewalls

More work is needed to ensure that IPv6 is supported in equal manner in various firewall products.

Application Support

But by far the most important action, for at least our group of users, would be to bring some key applications (e.g., instant messaging and VoIP applications and also games) to a state where they can be easily run on IPv6-only networks and behind a NAT64. To facilitate this, application programmers should use IP version agnostic APIs so that applications automatically use IPv4 or IPv6 depending on what is available. In some cases, it may also be necessary to add support for new types of ALGs.

IPv4 Literals

The web should be cleaned of IPv4 literals. Also IPv4 literals should be avoided in application protocol signaling messages.

Measurements and Analysis

It is also important to continue with testing, measurements, and analysis of what Internet technology works in IPv6-only networks, to what extent, at what speed, and where the remaining problems are.

Guidelines

It is also useful to provide guidance for network administrators and users on how to turn on IPv6-only networking.

As can be seen from the above list, there are only minor things that can be done through standardization. Most of the effort is practical and centers around improving various implementations.

9. Security Considerations

The use of IPv6 instead of IPv4 by itself does not make a big security difference. The main security requirement is that, naturally, network security devices need to be able to deal with IPv6 in these networks. This is though already required in all dual-stack networks. As noted, it is important, e.g., to ensure firewall capabilities. Security considerations for NAT64 and DNS64 are discussed in [RFC6146] and [RFC6147].

In our experience many of the critical security functions in a network end up being on the dual-stack part of the network anyway. For instance, our mail servers obviously still have to be able to communicate with both the IPv4 and IPv6 Internet, and as a result they and the associated spam & filtering components are not in the IPv6-only part of the network.

10. IANA Considerations

This document has no IANA implications.

11. References

11.1. Normative References

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

11.2. Informative References

- [RFC4038] Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E. Castro, "Application Aspects of IPv6 Transition", RFC 4038, March 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

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

[I-D.wing-behave-http-ip-address-literals]
Wing, D., "Coping with IP Address Literals in HTTP URIs with IPv6/IPv4 Translators",
draft-wing-behave-http-ip-address-literals-02 (work in progress), March 2010.

[HE-IPv6] Hurricane Electric, "Global IPv6 Deployment Progress Report", February 2012,
<<http://bgp.he.net/ipv6-progress-report.cgi>>.

Appendix A. Acknowledgments

The authors would like to thank the many people who have engaged in discussions around this topic, and particularly the people who were involved in building some of the new tools used in our network, our users who were interested in going where only few had dared to venture before, or people who helped us in this effort. In particular, we would like to thank Martti Kuparinen, Tero Kauppinen, Heikki Mahkonen, Jan Melen, Fredrik Garneij, Christian Gotare, Teemu Rinta-Aho, Petri Jokela, Mikko Sarela, Olli Arkko, Lasse Arkko, and Cameron Byrne. Also Marcelo Braun, Iljitsch van Beijnum, Miika Komu, and Jouni Korhonen have provided useful discussion and comments on the document.

Authors' Addresses

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Ari Keranen
Ericsson
Jorvas 02420
Finland

Email: ari.keranen@ericsson.com

BEHAVE
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2011

G. Camarillo
O. Novo
Ericsson
S. Perreault, Ed.
Viagenie
July 8, 2010

Traversal Using Relays around NAT (TURN) Extension for IPv6
draft-ietf-behave-turn-ipv6-11

Abstract

This document adds IPv6 support to Traversal Using Relays around NAT (TURN). IPv6 support in TURN includes IPv4-to-IPv6, IPv6-to-IPv6, and IPv6-to-IPv4 relaying. This document defines the REQUESTED-ADDRESS-FAMILY attribute for TURN. The REQUESTED-ADDRESS-FAMILY attribute allows a client to explicitly request the address type the TURN server will allocate (e.g., an IPv4-only node may request the TURN server to allocate an IPv6 address).

Status of this Memo

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

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

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

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

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 9, 2011.

Copyright Notice

Copyright (c) 2010 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 BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Overview of Operation	3
4. Creating an Allocation	4
4.1. Sending an Allocate Request	4
4.1.1. The REQUESTED-ADDRESS-FAMILY Attribute	4
4.2. Receiving an Allocate Request	5
4.2.1. Unsupported Address Family	6
4.3. Receiving an Allocate Error Response	6
5. Refreshing an Allocation	6
5.1. Sending a Refresh Request	6
5.2. Receiving a Refresh Request	6
6. CreatePermission	6
6.1. Sending a CreatePermission Request	7
6.2. Receiving a CreatePermission request	7
6.2.1. Peer Address Family Mismatch	7
7. Channels	7
7.1. Sending a ChannelBind Request	7
7.2. Receiving a ChannelBind Request	7
8. Packet Translations	7
8.1. IPv4-to-IPv6 Translations	8
8.2. IPv6-to-IPv6 Translations	9
8.3. IPv6-to-IPv4 Translations	11
9. Security Considerations	11
9.1. Tunnel Amplification Attack	12
10. IANA Considerations	13
10.1. New STUN Attribute	13
10.2. New STUN Error Codes	13
11. Acknowledgements	13
12. References	13
12.1. Normative References	13
12.2. Informative References	14
Authors' Addresses	14

1. Introduction

Traversal Using Relays around NAT (TURN) [I-D.ietf-behave-turn] is a protocol that allows for an element behind a NAT to receive incoming data over UDP or TCP. It is most useful for elements behind NATs without Endpoint-Independent Mapping [RFC4787] that wish to be on the receiving end of a connection to a single peer.

The base specification of TURN [I-D.ietf-behave-turn] only defines IPv4-to-IPv4 relaying. This document adds IPv6 support to TURN, which includes IPv4-to-IPv6, IPv6-to-IPv6, and IPv6-to-IPv4 relaying. This document defines the REQUESTED-ADDRESS-FAMILY attribute, which is an extension to TURN that allows a client to explicitly request the address type the TURN server will allocate (e.g., an IPv4-only node may request the TURN server to allocate an IPv6 address). This document also defines and registers new error response codes.

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].

3. Overview of Operation

When a user wishes a TURN server to allocate an address of a specific type, it sends an Allocate Request to the TURN server with a REQUESTED-ADDRESS-FAMILY attribute. TURN can run over UDP and TCP, and it allows for a client to request address/port pairs for receiving both UDP and TCP.

After the request has been successfully authenticated, the TURN server allocates a transport address of the type indicated in the REQUESTED-ADDRESS-FAMILY attribute. This address is called the relayed transport address.

The TURN server returns the relayed transport address in the response to the Allocate Request. This response contains a XOR-RELAYED-ADDRESS attribute indicating the IP address and port that the server allocated for the client.

TURN servers allocate a single relayed transport address per allocation request. Therefore, Allocate Requests cannot carry more than one REQUESTED-ADDRESS-FAMILY attribute. Consequently, a client that wishes to allocate more than one relayed transport address at a TURN server (e.g., an IPv4 and an IPv6 address) needs to perform

several allocation requests (one allocation request per relayed transport address).

A TURN server that supports a set of address families is assumed to be able to relay packets between them. If a server does not support the address family requested by a client, the server returns a 440 (Address Family not Supported) error response.

4. Creating an Allocation

The behavior specified here affects the processing defined in Section 6 of [I-D.ietf-behave-turn].

4.1. Sending an Allocate Request

A client that wishes to obtain a relayed transport address of a specific address type includes a REQUESTED-ADDRESS-FAMILY attribute, which is defined in Section 4.1.1, in the Allocate Request that it sends to the TURN server. Clients MUST NOT include more than one REQUESTED-ADDRESS-FAMILY attribute in an Allocate Request. The mechanisms to formulate an Allocate Request are described in Section 6.1 of [I-D.ietf-behave-turn].

Clients MUST NOT include a REQUESTED-ADDRESS-FAMILY attribute in an Allocate request that contains a RESERVATION-TOKEN attribute.

4.1.1. The REQUESTED-ADDRESS-FAMILY Attribute

The REQUESTED-ADDRESS-FAMILY attribute is used by clients to request the allocation of a specific address type from a server. The following is the format of the REQUESTED-ADDRESS-FAMILY attribute. Note that TURN attributes are TLV (Type-Length-Value) encoded, with a 16 bit type, a 16 bit length, and a variable-length value.

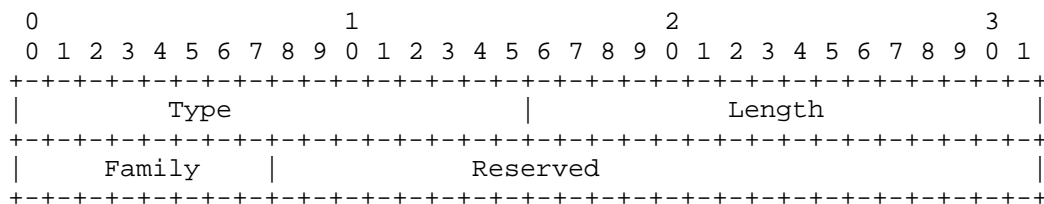


Figure 1: Format of REQUESTED-ADDRESS-FAMILY Attribute

Type: the type of the REQUESTED-ADDRESS-FAMILY attribute is 0x0017.

As specified in [RFC5389], attributes with values between 0x0000 and 0x7FFF are comprehension-required, which means that the client or server cannot successfully process the message unless it understands the attribute.

Length: this 16-bit field contains the length of the attribute in bytes. The length of this attribute is 4 bytes.

Family: there are two values defined for this field and specified in [RFC5389], Section 15.1: 0x01 for IPv4 addresses and 0x02 for IPv6 addresses.

Reserved: at this point, the 24 bits in the reserved field MUST be set to zero by the client and MUST be ignored by the server.

The REQUEST-ADDRESS-TYPE attribute MAY only be present in Allocate Requests.

4.2. Receiving an Allocate Request

Once a server has verified that the request is authenticated and has not been tampered with, the TURN server processes the Allocate request. If it contains both a RESERVATION-TOKEN and a REQUESTED-ADDRESS-FAMILY, the server replies with a 400 (Bad Request) Allocate Error Response. Following the rules in [RFC5389], if the server does not understand the REQUESTED-ADDRESS-FAMILY attribute, it generates an Allocate Error Response, which includes an ERROR-CODE attribute with response code 420 (Unknown Attribute). This response will contain an UNKNOWN-ATTRIBUTE attribute listing the unknown REQUESTED-ADDRESS-FAMILY attribute.

If the server can successfully process the request, it allocates a transport address for the TURN client, called the relayed transport address, and returns it in the response to the Allocate Request.

As specified in [I-D.ietf-behave-turn], the Allocate Response contains the same transaction ID contained in the Allocate Request and the XOR-RELAYED-ADDRESS attribute is set to the relayed transport address.

The XOR-RELAYED-ADDRESS attribute indicates the allocated IP address and port. It is encoded in the same way as the XOR-MAPPED-ADDRESS [RFC5389].

If the REQUESTED-ADDRESS-FAMILY attribute is absent, the server MUST allocate an IPv4 relayed transport address for the TURN client. If allocation of IPv4 addresses is disabled by local policy, the server

returns a 440 (Address Family not Supported) Allocate Error Response.

If the server does not support the address family requested by the client, it MUST generate an Allocate Error Response, and it MUST include an ERROR-CODE attribute with the 440 (Address Family not Supported) response code, which is defined in Section 4.2.1.

4.2.1. Unsupported Address Family

This document defines the following new error response code:

440 (Address Family not Supported): The server did not support the address family requested by the client.

4.3. Receiving an Allocate Error Response

If the client receives an Allocate error response with the 440 (Unsupported Address Family) error code, the client MUST NOT retry its request.

5. Refreshing an Allocation

The behavior specified here affects the processing defined in Section 7 of [I-D.ietf-behave-turn].

5.1. Sending a Refresh Request

To perform an allocation refresh, the client generates a Refresh Request as described in Section 7.1 of [I-D.ietf-behave-turn]. The client MUST NOT include any REQUESTED-ADDRESS-FAMILY attribute in its Refresh Request.

5.2. Receiving a Refresh Request

If a server receives a Refresh Request with a REQUESTED-ADDRESS-FAMILY attribute, and the attribute's value doesn't match the address family of the allocation, the server MUST reply with a 443 (Peer Address Family Mismatch) Refresh Error Response.

6. CreatePermission

The behavior specified here affects the processing defined in Section 9 of [I-D.ietf-behave-turn].

6.1. Sending a CreatePermission Request

The client MUST only include XOR-PEER-ADDRESS attributes with addresses of the same address family as the relayed transport address for the allocation.

6.2. Receiving a CreatePermission request

If an XOR-PEER-ADDRESS attribute contains an address of an address family different than the relayed transport address for the allocation, the server MUST generate an error response with the 443 (Peer Address Family Mismatch) response code, which is defined in Section 6.2.1.

6.2.1. Peer Address Family Mismatch

This document defines the following new error response code:

443 (Peer Address Family Mismatch): A peer address was of a different address family than the relayed transport address of the allocation.

7. Channels

The behavior specified here affects the processing defined in Section 11 of [I-D.ietf-behave-turn].

7.1. Sending a ChannelBind Request

The client MUST only include a XOR-PEER-ADDRESS attribute with an address of the same address family as the relayed transport address for the allocation.

7.2. Receiving a ChannelBind Request

If the XOR-PEER-ADDRESS attribute contains an address of an address family different than the relayed transport address for the allocation, the server MUST generate an error response with the 443 (Peer Address Family Mismatch) response code, which is defined in Section 6.2.1.

8. Packet Translations

The TURN specification [I-D.ietf-behave-turn] describes how TURN relays should relay traffic consisting of IPv4 packets (i.e., IPv4-to-IPv4 translations). The relay translates the IP addresses and

port numbers of the packets based on the allocation's state data. How to translate other header fields is also specified in [I-D.ietf-behave-turn]. This document addresses IPv4-to-IPv6, IPv6-to-IPv4, and IPv6-to-IPv6 translations.

TURN relays performing any translation MUST translate the IP addresses and port numbers of the packets based on the allocation's state information as specified in [I-D.ietf-behave-turn]. The following sections specify how to translate other header fields.

As discussed in Section 2.6 of [I-D.ietf-behave-turn], translations in TURN are designed so that a TURN server can be implemented as an application that runs in userland under commonly available operating systems and that does not require special privileges. The translations specified in the following sections follow this principle.

The descriptions below have two parts: a preferred behavior and an alternate behavior. The server SHOULD implement the preferred behavior. Otherwise, the server MUST implement the alternate behavior and MUST NOT do anything else.

8.1. IPv4-to-IPv6 Translations

Traffic Class

Preferred behavior: as specified in Section 3 of [I-D.ietf-behave-v6v4-xlate].

Alternate behavior: the relay sets the Traffic Class to the default value for outgoing packets.

Flow Label

Preferred behavior: The relay sets the Flow label to 0. The relay can choose to set the Flow label to a different value if it supports [RFC3697].

Alternate behavior: the relay sets the Flow label to the default value for outgoing packets.

Hop Limit

Preferred behavior: as specified in Section 3 of [I-D.ietf-behave-v6v4-xlate].

Alternate behavior: the relay sets the Hop Limit to the default value for outgoing packets.

Fragmentation

Preferred behavior: as specified in Section 3 of [I-D.ietf-behave-v6v4-xlate].

Alternate behavior: the relay assembles incoming fragments. The relay follows its default behavior to send outgoing packets.

For both preferred and alternate behavior, the DONT-FRAGMENT attribute ([I-D.ietf-behave-turn], Section 14.8) MUST be ignored by the server.

Extension Headers

Preferred behavior: the relay sends outgoing packet without any IPv6 extension headers, with the exception of the Fragmentation header as described above.

Alternate behavior: same as preferred.

8.2. IPv6-to-IPv6 Translations

Flow Label

The relay should consider that it is handling two different IPv6 flows. Therefore, the Flow label [RFC3697] SHOULD NOT be copied as part of the translation.

Preferred behavior: The relay sets the Flow label to 0. The relay can choose to set the Flow label to a different value if it supports [RFC3697].

Alternate behavior: the relay sets the Flow label to the default value for outgoing packets.

Hop Limit

Preferred behavior: the relay acts as a regular router with respect to decrementing the Hop Limit and generating an ICMPv6 error if it reaches zero.

Alternate behavior: the relay sets the Hop Limit to the default value for outgoing packets.

Fragmentation

Preferred behavior: If the incoming packet did not include a Fragment header and the outgoing packet size does not exceed the outgoing link's MTU, the relay sends the outgoing packet without a Fragment header.

If the incoming packet did not include a Fragment header and the outgoing packet size exceeds the outgoing link's MTU, the relay drops the outgoing packet and send an ICMP message of type 2 code 0 ("Packet too big") to the sender of the incoming packet. If the packet is being sent to the peer, the relay reduces the MTU reported in the ICMP message by 48 bytes to allow room for the overhead of a Data indication.

If the incoming packet included a Fragment header and the outgoing packet size (with a Fragment header included) does not exceed the outgoing link's MTU, the relay sends the outgoing packet with a Fragment header. The relay sets the fields of the Fragment header as appropriate for a packet originating from the server.

If the incoming packet included a Fragment header and the outgoing packet size exceeds the outgoing link's MTU, the relay **MUST** fragment the outgoing packet into fragments of no more than 1280 bytes. The relay sets the fields of the Fragment header as appropriate for a packet originating from the server.

Alternate behavior: the relay assembles incoming fragments. The relay follows its default behavior to send outgoing packets.

For both preferred and alternate behavior, the DONT-FRAGMENT attribute **MUST** be ignored by the server.

Extension Headers

Preferred behavior: the relay sends outgoing packet without any IPv6 extension headers, with the exception of the Fragmentation header as described above.

Alternate behavior: same as preferred.

8.3. IPv6-to-IPv4 Translations

Type of Service and Precedence

Preferred behavior: as specified in Section 4 of [I-D.ietf-behave-v6v4-xlate].

Alternate behavior: the relay sets the Type of Service and Precedence to the default value for outgoing packets.

Time to Live

Preferred behavior: as specified in Section 4 of [I-D.ietf-behave-v6v4-xlate].

Alternate behavior: the relay sets the Time to Live to the default value for outgoing packets.

Fragmentation

Preferred behavior: as specified in Section 4 of [I-D.ietf-behave-v6v4-xlate]. Additionally, when the outgoing packet's size exceeds the outgoing link's MTU, the relay needs to generate an ICMP error (ICMPv6 Packet Too Big) reporting the MTU size. If the packet is being sent to the peer, the relay SHOULD reduce the MTU reported in the ICMP message by 48 bytes to allow room for the overhead of a Data indication.

Alternate behavior: the relay assembles incoming fragments. The relay follows its default behavior to send outgoing packets.

For both preferred and alternate behavior, the DONT-FRAGMENT attribute MUST be ignored by the server.

9. Security Considerations

Translation between IPv4 and IPv6 creates a new way for clients to obtain IPv4 or IPv6 access which they did not have before. For example, an IPv4-only client having access to a TURN server implementing this specification is now able to access the IPv6 internet. This needs to be considered when establishing security and

monitoring policies.

The loop attack described in [I-D.ietf-behave-turn] Section 17.1.7 may be more easily done in cases where address spoofing is easier to accomplish over IPv6. Mitigation of this attack over IPv6 is the same as for IPv4.

All the security considerations applicable to STUN [RFC5389] and TURN [I-D.ietf-behave-turn] are applicable to this document as well.

9.1. Tunnel Amplification Attack

An attacker might attempt to cause data packets to loop numerous times between a TURN server and a tunnel between IPv4 and IPv6. The attack goes as follows.

Suppose an attacker knows that a tunnel endpoint will forward encapsulated packets from a given IPv6 address (this doesn't necessarily need to be the tunnel endpoint's address). Suppose he then spoofs two packets from this address:

1. An allocate request asking for a v4 address, and
2. A ChannelBind request establishing a channel to the IPv4 address of the tunnel endpoint

Then he has set up an amplification attack:

- o The TURN relay will re-encapsulate IPv6 UDP data in v4 and send it to the tunnel endpoint
- o The tunnel endpoint will decapsulate packets from the v4 interface and send them to v6

So if the attacker sends a packet of the following form...

```
IPv6: src=2001:DB9::1 dst=2001:DB8::2
UDP:  <ports>
TURN: <channel id>
IPv6: src=2001:DB9::1 dst=2001:DB8::2
UDP:  <ports>
TURN: <channel id>
IPv6: src=2001:DB9::1 dst=2001:DB8::2
UDP:  <ports>
TURN: <channel id>
...
```

Then the TURN relay and the tunnel endpoint will send it back and forth until the last TURN header is consumed, at which point the TURN relay will send an empty packet, which the tunnel endpoint will drop.

The amplification potential here is limited by the MTU, so it's not

huge: IPv6+UDP+TURN takes 334 bytes, so you could get a four-to-one amplification out of a 1500-byte packet. But the attacker could still increase traffic volume by sending multiple packets or by establishing multiple channels spoofed from different addresses behind the same tunnel endpoint.

The attack is mitigated as follows. It is RECOMMENDED that TURN relays not accept allocation or channel binding requests from addresses known to be tunneled, and that they not forward data to such addresses. In particular, a TURN relay MUST NOT accept Teredo or 6to4 addresses in these requests.

10. IANA Considerations

The IANA is requested to register the following values under the STUN Attributes registry and under the STUN Error Codes registry.

10.1. New STUN Attribute

0x0017: REQUESTED-ADDRESS-FAMILY

10.2. New STUN Error Codes

440 Address Family not Supported
443 Peer Address Family Mismatch

11. Acknowledgements

The authors would like to thank Alfred E. Heggstad, Dan Wing, Magnus Westerlund, Marc Petit-Huguenin, Philip Matthews, and Remi Denis-Courmont for their feedback on this document.

12. References

12.1. Normative References

[I-D.ietf-behave-turn]
Rosenberg, J., Mahy, R., and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)",
draft-ietf-behave-turn-16 (work in progress), July 2009.

[I-D.ietf-behave-v6v4-xlate]

Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", draft-ietf-behave-v6v4-xlate-10 (work in progress), February 2010.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, March 2004.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

12.2. Informative References

- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

Oscar Novo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Oscar.Novo@ericsson.com

Simon Perreault (editor)
Viagenie
2600 boul. Laurier, suite 625
Quebec, QC G1V 4W1
Canada

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

OPSAWG
Internet-Draft
Intended status: Informational
Expires: August 23, 2012

V. Kuarsingh, Ed.
J. Cianfarani
Rogers Communications
February 20, 2012

CGN Deployment with MPLS/VPNs
draft-kuarsingh-lsn-deployment-06

Abstract

This document specifies a framework to integrate a Network Address Translation layer into an operator's network to function as a Carrier Grade NAT (also known as CGN or Large Scale NAT). CGN is a concept also described in [I-D.ietf-behave-lsn-requirements] and describes the model as a dual layer translation model. Although operators may wish to deploy IPv6 to strategically overcome IPv4 exhaustion, near term needs may not be satisfied with an IPv6 deployment alone. This document provides a practical integration model which allows CGN to be integrated into the network meeting the connectivity needs of the customer while being mindful of not disrupting existing services and meeting the technical challenges that CGN brings. The model includes the use of MPLS/VPNs defined in [RFC4364] as a tool to achieve this goal. This document does not intend to defend the merits of CGN.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 23, 2012.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal

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

Table of Contents

1. Introduction	3
2. Motivation	3
3. CGN Network Deployment Requirements	4
3.1. Centralized versus Distributed Deployment	5
3.2. CGN and Traditional IPv4 Service Co-existence	6
3.3. CGN By-Pass	6
3.4. Routing Plane Separation	6
3.5. Flexible Deployment Options	7
3.6. IPv4 Overlap Space	7
3.7. Transactional Logging for LSN Systems	7
3.8. Additional CGN Requirements	8
4. MPLS/VPN based CGN Framework	8
4.1. Service Separation	9
4.2. Internal Service Delivery	10
4.2.1. Dual Stack Operation	11
4.3. Deployment Flexibility	12
4.4. Comparison of MPLS/VPN Option versus other CGN Attachment Options	13
4.4.1. IEEE 802.1Q	13
4.4.2. Policy Based Routing	14
4.4.3. Traffic Engineering	14
4.4.4. Multiple Routing Topologies	14
5. Experiences	14
6. Basic Integration and Requirements Support	14
7. Performance	15
8. IANA Considerations	16
9. Security Considerations	16
10. Conclusions	17
11. Acknowledgements	17
12. References	17
12.1. Normative References	17
12.2. Informative References	17
Authors' Addresses	18

1. Introduction

Operators are faced with near term IPv4 address exhaustion challenges. Many operators may not have a sufficient amount of IPv4 addresses in the future to satisfy the needs of their growing customer base. This challenge may also be present before or during an active transition to IPv6 somewhat complicating the overall problem space.

To face this challenge, operators may need to deploy CGN (Carrier Grade NAT) as described in [I-D.ietf-behave-lsn-requirements] to help extend the connectivity matrix once IPv4 addresses run out in the network. CGN's addition to the network requires integration in an often running state environment with working IPv4 and/or IPv6 services.

The addition of the CGN introduces an operator controlled and administered translation layer which needs to be added in a manner which does not overly disrupt existing services. This addition may also include interworking in a dual stack environment where the IPv4 path requires translation.

This document shows how MPLS/VPNs as described in [RFC4364] can be used to integrate the CGN infrastructure solving key problems faced by the operator. This model has also been tested and validated in real production network models and allows fluid operation with existing IPv4 and IPv6 services.

2. Motivation

The selection of CGN may be made by an operator based on a number of factors. The overall driver may be the depletion of IPv4 address pools which leaves little to no addresses for IPv4 service growth. IPv6 is considered the strategic answer, but it's applicability and usefulness in many networks is limited by the current access network and consumer home network. These environments often are filled with IPv4-Only equipment which may not be upgradable to IPv6.

The ability to replace IPv4-Only equipment may be out of the control of the operator, and even when it's in the administrative control; it poses both cost and technical challenges as operators build out massive programs for equipment retirement or upgrade. These issues leave an operator in a precarious position which may lead to the decision to deploy CGN. Other address IPv4 sharing options do exist which are more architecturally desirable, but the practical and workable approach in many cases is a CGN deployment using NAT444.

If the operator as has chosen to deploy CGN, they should this in a manner as not to negatively impact the existing IPv4 or IPv6 customer base. This will include solving a number of challenges since customers who's connections require translation will have network routing and flow needs which are different from legacy IPv4 connections.

The solution will also need to work in a dual stack environment where other options such as DS-Lite [RFC6333] are not yet viable. Even technologies like 6RD [RFC5969] still require an IPv4 connectivity path to service the customer endpoint. The solution will need to address basic Internet connectivity, on-net service offerings, back office management, billing, policy and security models already in place within the operator's network. CGN will often integrate quite readily with the aforementioned requirements where as other transition mechanism may not due to the requirements to support IPv6 as the base protocol for IPv4 connectivity.

3. CGN Network Deployment Requirements

If a service provider is considering a CGN deployment with a provider NAT44 function, there are a number of basic requirements which are of importance. Preliminary requirements may require the following from the incoming CGN system architecture:

- Support distributed (sparse) and centralized (dense) deployment models;
- Allow co-existence with traditional IPv4 based deployments, which provide global scoped IPs to CPEs;
- Provide a framework for CGN by-pass supporting non-translated flows between endpoints within a provider's network;
- Provide routing framework which allows the segmentation of routing control and forwarding paths between CGN and non-CGN mediated flows;
- Provide flexibility for operators to modify their deployments over time as translation demands change (connections, bandwidth, translation realms/zones and other vectors);
- Flexibility should include integration options for common access technologies such as DSL (BRAS), DOCSIS (CMTS), Mobile (GGSN/PGW/ASN-GW), and Ethernet access;

- Support deployment modes that allow for IPv4 address overlap within the operator's network (between various translation realms or zones);
- Allow for evolution to future dual-stack and IPv4/IPv6 transition deployment modes;
- Transactional logging and export capabilities to support auxiliary functions including abuse mitigation;
- Support for stateful connection synchronization between translation instances/elements (redundancy);
- Support for CGN Shared Space [I-D.weil-shared-transition-space-request] deployment modes if applicable;
- Allows for the enablement of CGN functionality (if required) while still minimizing costs and customer impact to the best extend possible;

Other requirements may be assessed on a operator-by-operator basis, but those listed above should be considered for any given deployment architecture.

3.1. Centralized versus Distributed Deployment

Centralized deployments of CGN (longer proximity to end user and/or higher densities of subscribers/connections to CGN instances) differ from distributed deployments of CGN (closer proximity to end user and/or lower densities of subscribers/connections to CGN instances). Service providers will likely deploy CGN translation points more centrally during initial phases. Early deployments will likely see light loading on these new systems since legacy IPv4 services will continue to operate with most endpoints using globally unique IPv4 addresses. Exceptional cases which may drive heavy usage in initial stages may include operators who already translate most IPv4 traffic and will migrate to a CGN implementation from legacy firewalls; or a green field deployment which may see quick growth in the number of new IPv4 endpoints which require Internet connectivity.

Over time, most providers will likely need to expand and possibly distribute the translation points as demand for the CGN system increases. The extent of the expansion of the CGN infrastructure will depend on factors such as growth in the number of IPv4 endpoints, status of IPv6 content on the Internet and the overall progress globally to an IPv6-dominate Internet (reducing the demand for IPv4 connectivity).

3.2. CGN and Traditional IPv4 Service Co-existence

Newer CGN serviced endpoints will exist alongside endpoints served by traditional IPv4 global IPs. Providers will need to rationalize these environments since both have distinct forwarding needs. Traditional IPv4 services will likely require (or be best served) direct forwarding towards Internet peering points while CGN mediated flows require access to a translator. CGN and non-CGN mediated flows post two fundamentally different forwarding needs.

The new CGN environments should not negatively impact the existing IPv4 service base by forcing all traffic to translation enabled network points since many flows do not require translation and this would reduce performance of the existing flows. This would also require massive scaling of the CGN which is a cost and efficiency concern as well.

Traffic flow and forwarding efficiency is considered important since networks are under considerable demand to deliver more and more bandwidth without the luxury of needless inefficiencies which can be introduced with CGN.

3.3. CGN By-Pass

The CGN environment is only needed for flows with translation requirements. Many flows which remain in a service provider environment, do not require translation. Such services include operator offered DNS Services, DHCP Services, NTP Services, Web Caching, Mail, News and other services which are local to the operator's network.

The operator may want to leverage opportunities to offer third parties a platform to also provide services without translation. CGN By-pass can be accomplished in many ways, but a simplistic, deterministic and scalable model is preferred.

3.4. Routing Plane Separation

Many operators will want to engineer traffic separately for CGN flows versus flows which are part of the more traditional IPv4 environment. Many times the routing of these two major flow types differ, therefore route separation may be required.

Routing plane separation also allows the operator to utilize other addressing techniques, which may not be feasible on a single routing plane. Such examples include the use of overlapping private address space [RFC1918] or use of other IPv4 space which may overlap globally within the operator's network.

3.5. Flexible Deployment Options

Service providers operate complex routing environments and offer a variety of IPv4 based services. Many operator environments utilize distributed peering infrastructures for transit and peering and these may span large geographical areas and regions. A CGN solution should offer the operator an ability to place CGN translation points at various points within their network.

The CGN deployment should also be flexible enough to change over time as demand for translation services increase. In turn, the deployment will need to then adapt as translation demand decreases caused by the transition of flows to IPv6. Translation points should be able to be placed and moved with as little re-engineering effort as possible minimizing the risks to the customer base.

Depending on hardware capabilities, security practices and IPv4 address availability, the translation environments may need to be segmented and/or scaled over time to meet organic IPv4 demand growth. Operators will want to seek deployment models which are conducive to meeting these goals as well.

3.6. IPv4 Overlap Space

IP address overlap for CGN translation realms may be required if insufficient IPv4 addresses are available within the service provider environment to assign internally unique IPs to the CGN customer base. The CGN deployment should provide mechanisms to manage IPv4 overlap if required.

3.7. Transactional Logging for LSN Systems

CGNs may require transactional logging since the source IP and related transport protocol information is not easily visible to external hosts and system.

If needed, the CGN systems should be able to generate logs which identify 'internal' host parameters (i.e. IP/Port) and associated them to external translated parameters imposed by the translator. The logged information should be stored on the CGN hardware and/or exported to an external system for processing. Operators may need to keep track of this information (securely) to meet regulatory and/or legal obligations. Further information can be found in [I-D.ietf-behave-lsn-requirements] with respect to CGN logging requirements (Logging Section).

3.8. Additional CGN Requirements

The CGN platform will also need to meet the needs of additional requirements such as Bulk Port Allocation and other CGN device specific functions. These additional requirements are captured within [I-D.ietf-behave-lsn-requirements].

4. MPLS/VPN based CGN Framework

The MPLS/VPN [RFC4364] framework for CGN segregates the 'pre-translated' realms within the service provider space into layer-3 MPLS/VPNs. The operator can deploy a single realm for all CGN based flows, or can deploy multiple realms based on translation demand and other factors such as geographical proximity. A realm in this model refers to a 'VPN' which shares a unique RD/RT combination, routing plane and forwarding behaviours.

The MPLS/VPN infrastructure provides control plane and forwarding separation for the traditional IPv4 service environment and CGN environment(s). The separation allows for routing information (such as default routes) to be propagated separately for CGN and non-CGN based customer flows. Traffic can be efficiently routed to the Internet for normal flows, and routed directly to translators for CGN mediated flows. Although many operators may run a "default-route-free" core, IPv4 flows which require translation must obviously be routed first to a translator, so a default route is acceptable for the pre-translated realms.

The physical location of the VRF Termination point for a MPLS/VPN enabled CGN can vary and be located anywhere within the operator's network. This model fully virtualizes the translation service from the base IPv4 forwarding environment which will likely carrying Internet bound traffic. The base IPv4 environment can continue to service traditional IPv4 customer flows plus post translated CGN flows.

Figure 1 provides a view of the basic model. The Access node provides CPE access to either the CGN VRF or the Global Routing Table, depending on whether the customer receives a private or public IP. Translator mediated traffic follows an MPLS LSP which can be setup dynamically and can span one hop, or many hops (with no need for complex routing policies). Traffic is then forwarded to the translator (shown below) which can be an external appliance or integrated into the VRF Termination (Provider Edge) router. Once traffic is translated, it is forwarded to the global routing table for general Internet forwarding. The Global Routing table can also be a separate VRF (Internet Access VPN/VRF) should the provider

choose to implement their Internet based services in that fashion. The translation services are effectively overlaid onto the network, but are maintained within a separate forwarding and control plane.

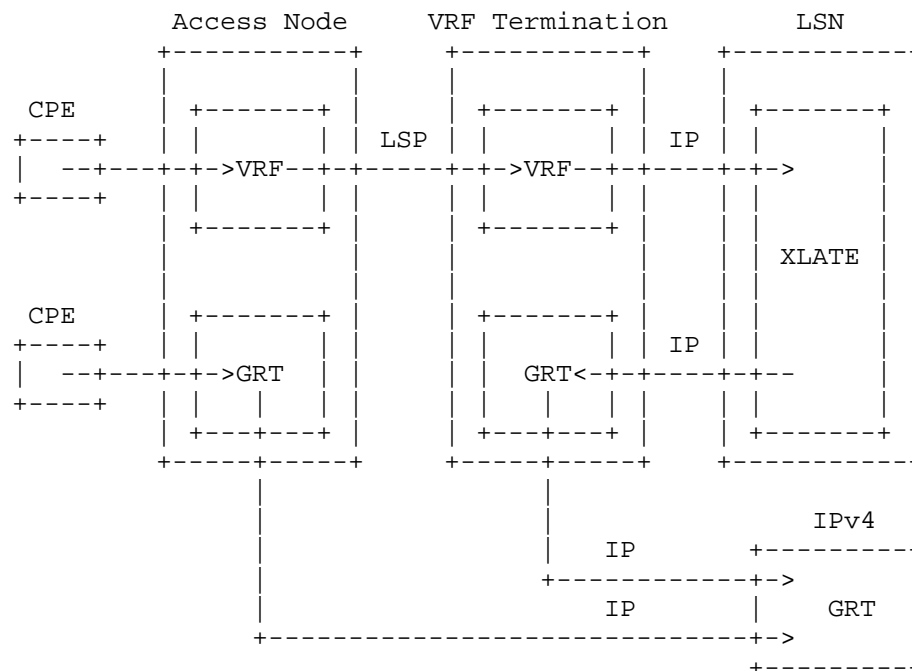


Figure 1: Basic MPLS/VPN CGN Model

If more than one VRF (translation realm) is used within the operator's network, each VPN instance can manage CGN flows independently for the respective realm. Various redundancy models can be used within this architecture to support failover from one physical CGN hardware instance to another. If state information needs to be passed or maintained between hardware instances, the vendor would need to enable this feature in a suitable manner.

4.1. Service Separation

The MPLS/VPN CGN framework supports route separation. The traditional IPv4 flows can be separated at the access node (Initial Layer 3 service point) from those which require translation. This type of service separation is possible on common technologies used for Internet access within many operator networks. Service separation can be accomplished on common access technology including

those used for DOCSIS (CMTS), Ethernet Access, DSL (BRAS), and Mobile Access (GGSN/ASN-GW) architectures.

4.2. Internal Service Delivery

Internal services can be delivered directly to the privately addressed endpoint within the CGN domain without translation. This can be accomplished using direct route exchange (import/export) between the CGN VRFs and the Services VRFs. The previous statement assumes the provider puts key services into a VRF for simple route exchange. This model allows the provider to maintain separate forwarding rules for translated flows, which require a pass through the translator to reach external network entities, versus those flows which need to access internal services. This operational detail can be advantageous for a number of reasons.

First, the provider can reduce the load on the translator since internal services do not need to be factored into the scaling of the CGN hardware. Secondly, more direct forwarding paths can be maintained providing better network efficiency. Thirdly, geographic locations of the translators and the services infrastructure can be deployed in a location in an independent manner. Additionally, the operator can allow CGN subject endpoints to be accessible via an untranslated path reducing the complexities of provider initiated management flows. This last point is of key interest since NAT removes transparency to the end device in normal cases.

Figure 2 below shows how internal services are provided untranslated since flows are sent directly from the access node to the services node/VRF via an MPLS LSP. This traffic is not forwarded to the CGN translator and therefore is not subject to problematic behaviours related to NAT. The services VRF contains routing information which can be "imported" into the access node VRF and the CGN VRF routing information can be "imported" into the Services VRF.

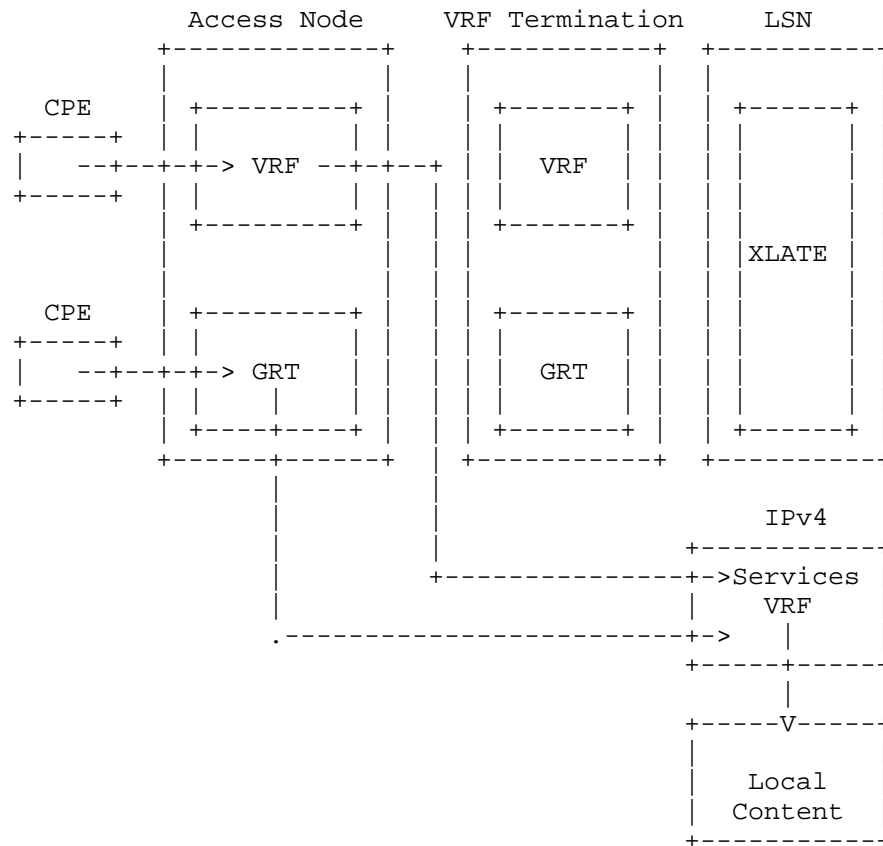


Figure 2: Internal Services and CGN By-Pass

This demonstrates the ability to offer CGN By-Pass in a simple and deterministic manner without the need of policy based routing or traffic engineering.

4.2.1. Dual Stack Operation

The MPLS/VPN CGN model can also be used in conjunction with IPv4/IPv6 dual stack service modes. Since many providers will use CGNs on an interim basis while IPv6 matures within the global Internet or due to technical constraints, a dual stack option is of strategic importance. Operators can offer this dual stack service for both traditional IPv4 (global IP) endpoints and CGN mediated endpoints.

Operators can separate the IP flows for IPv4 and IPv6 traffic, or use other routing techniques to move IPv6 based flows towards the GRT

(Global Routing Table or Instance) while allowing IPv4 flows to remain within the IPv4 CGN VRF for translator services.

The Figure 3 below shows how IPv4 translation services can be provided alongside IPv6 based services. The model shown allows the provider to enable CGN to manage IPv4 flows (translated) and IPv6 flows are routed without translation efficiently towards the Internet. Once again, forwarding of flows to the translator does not impact IPv6 flows which do not require this service.

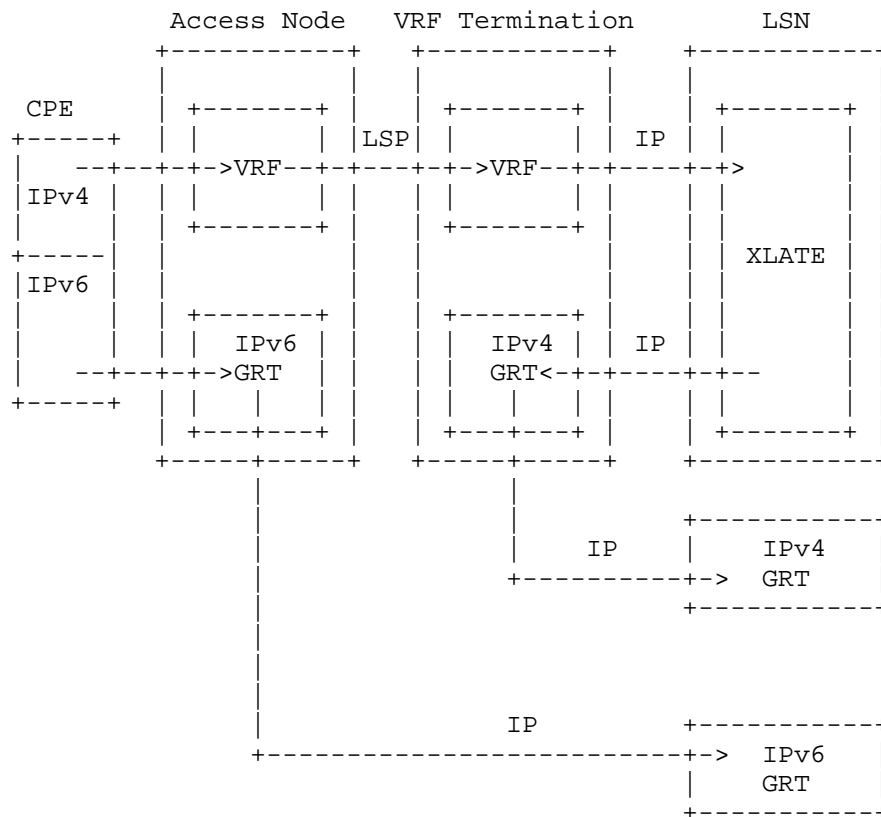


Figure 3: CGN with IPv6 Dual Stack Operation

4.3. Deployment Flexibility

The CGN translator services can be moved, separated or segmented (new translation realms) without the need to change the overall translation design. Since dynamic LSPs are used to forward traffic

from the access nodes to the translation points, the physical location of the VRF termination points can vary and be changed easily.

This type of flexibility allows the service provider to initially deploy more centralized translation services based on relatively low loading factors, and distribute the translation points over time to improve network traffic efficiencies and support higher translation load.

Although traffic engineered paths are not required within the MPLS/VPN deployment model, nothing precludes an operator from using technologies like MPLS with Traffic Engineering [RFC3031]. Additional routing mechanisms can be used as desired by the provider and can be seen as independent. There is no specific need to diversify the existing infrastructure in most cases.

4.4. Comparison of MPLS/VPN Option versus other CGN Attachment Options

Other integration architecture options exist which can attach CGN based service flows to a translator instance. Alternate options which can be used to attach such services include:

- IEEE 802.1Q for direct attachment to a next hop translator;
- Policy Based Routing (Static) to direct translation bound traffic to a network based translator;
- Traffic Engineering or;
- Multiple Routing Topologies

4.4.1. IEEE 802.1Q

IEEE 802.1Q can be used to associate separated traffic from the access node to the next hop router's CGN instance. This technology option may limit the CGN placement to the next hop router unless a second technology option is paired with it to extend connectivity deeper in the network.

This option is most effective if CGN instances are placed directly upstream of the access node. Distributed CGN instance placement is not likely an initial stage of the CGN deployment due to cost and demand factors.

4.4.2. Policy Based Routing

Policy Based Routing (PBR) provides another option to direct CGN mediated flows to a translator. PBR options, although possible, are difficult to maintain (static policy) and must be configured throughout the network with considerable maintenance overhead.

More centralized deployments may be difficult or too onerous to deploy using Policy Based Routing methods. Policy Based Routing would not achieve route separation (unless used with others options), and may add complexities to the providers' routing environment.

4.4.3. Traffic Engineering

Traffic Engineering can also be used to direct traffic from an access node towards a translator. Traffic Engineering, like MPLS-TE, may be difficult to setup and maintain. Traffic Engineering provides additional benefits if used with MPLS by adding potentials for faster path re-convergence. Traffic Engineering paths would need to be updated and redefined overtime as CGN translation points are augmented or moved.

4.4.4. Multiple Routing Topologies

Multiple routing topologies can be used to direct CGN based flows to translators. This option would achieve the same basic goal as the MPLS/VPN option but with additional implementation overhead and platform configuration complexity. Since operator based translation is expected to have an unknown lifecycle, and may see various degrees of demand (dependant on operator IPv4 Global space availability and shift of traffic to IPv6), it may be too large of an undertaking for the provider to enabled this as their primary option for CGN.

5. Experiences

6. Basic Integration and Requirements Support

The MPLS/VPN CGN environment has been successfully integrated into real network environments utilizing existing network service delivery mechanisms. It solves many issues related to provider based translation environments, while still subject to problematic behaviours inherent within NAT.

Key issues which are solved or managed with the MPLS/VPN option include:

- Centralized and Distributed Deployment model support
- Routing Plane Separation for CGN flows versus traditional IPv4 flows
- Flexible Translation Point Design (can relocate translators and split translation zones easily)
- Low maintenance overhead (dynamic routing environment with little maintenance of separate routing infrastructure other than management of MPLS/VPNs)
- CGN By-pass options (for internal and third party services which exist within the provider domain)
- IPv4 Translation Realm overlap support (can reuse IP addresses between zones with some impact to extranet service model)
- Simple failover techniques can be implemented with redundant translators, such as using a second default route

7. Performance

The MPLS/VPN CGN model was observed to support basic functions which are typically used by customers within an operator environment. Examples of successful operation include:

- Traditional Web (HTTP) Surfing (client initiated)
- Internet Video Streaming
- HTTP Based Client Connections
- High Connection Count sites (i.e. Google Maps)
- Email Transaction Support (POP, IMAP, SMTP)
- Instant Messaging Support (Online Status, File transfers, text chat)
- ICMP Operation (client initiated Echo, Traceroute)
- Peer to Peer application support (download)
- DNS (based on services extranet option, but was problematic when passed through a translator)

CGNs are still subject to problematic connectivity even within the MPLS/VPN technology approach. Problems which arise, or are not inherently addressed in this model include:

- Inward services from the Internet to the CPE
- Web session tracking
- Restricting usage and/or access based on source IP
- Abuse mitigation (masquerade of potential offenders)
- Increased network or server IDS false positives
- Increased customer risk for session hijacking
- Exceeding firewall TCP/UDP limits
- Customer identification (external site)
- Poor source based load balancing
- Customer usage tracking / Ad insertion
- Other applications or operations may be negatively impacted

8. IANA Considerations

There are not specific IANA considerations known at this time with the architecture described herein. Should a provider choose to use non-assigned IP address space within their translation realms, then considerations may apply.

9. Security Considerations

The same security considerations would typically exist for CGN deployments when compared with traditional IPv4 based services. With the MPLS/VPN model, the operator would want to consider security issues related to offering IP services over MPLS.

If a provider plans to operate the pre-translation realm (CPE towards translator IPv4 zone) as a non-public like network, then additional security measures may be needed to secure this environment. It is however the position in this document that CGN realms are public domains which utilize non-Internet routable IP addresses for endpoint addressing.

10. Conclusions

The MPLS/VPN delivery method for a CGN deployment is an effective and scalable way to deliver mass translation services. The architecture avoids the complex requirements of traffic engineering and policy based routing when combining these new service flows to existing IPv4 operation. This is advantageous since the NAT44/CGN environments should be introduced with as little impact as possible and these environments are expected to change over time.

The MPLS/VPN based CGN architecture solves many of this issues related to deploying this technology in existing operator networks.

11. Acknowledgements

Thanks to the following people for their participating in integrating and testing the CGN environment: Chris Metz, Syd Alam, Richard Lawson, John E Spence.

Additional thanks for the following people for the guidance on IPv6 transition considerations: John Jason Brzozowski, Chris Donley, Jason Weil, Lee Howard, Jean-Francois Tremblay

12. References

12.1. Normative References

- [I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-05 (work in progress), November 2011.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

12.2. Informative References

- [I-D.weil-shared-transition-space-request]
Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA Reserved IPv4 Prefix for Shared Address Space", draft-weil-shared-transition-space-request-15 (work in progress), February 2012.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets",

BCP 5, RFC 1918, February 1996.

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

Authors' Addresses

Victor Kuarsingh (editor)
Rogers Communications
8200 Dixie Road
Brampton, Ontario L6T 0C1
Canada

Email: victor.kuarsingh@gmail.com
URI: <http://www.rogers.com>

John Cianfarani
Rogers Communications
8200 Dixie Road
Brampton, Ontario L6T 0C1
Canada

Email: john.cianfarani@rci.rogers.com
URI: <http://www.rogers.com>

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 5, 2013

I. Yamagata
Y. Shirasaki
NTT Communications
A. Nakagawa
Japan Internet Exchange (JPIX)
J. Yamaguchi
Fiber 26 Network
H. Ashida
IS Consulting G.K.
July 4, 2012

NAT444
draft-shirasaki-nat444-06

Abstract

This document describes one of the network models that are designed for smooth transition to IPv6. It is called NAT444 model. NAT444 model is composed of IPv6, and IPv4 with Carrier Grade (CGN).

NAT444 is the only scheme not to require replacing Customer Premises Equipment (CPE) even if IPv4 address exhausted. But it must be noted that NAT444 has serious restrictions i.e. it limits the number of sessions per CPE so that rich applications such as AJAX and RSS feed cannot work well.

Therefore, IPv6 which is free from such a difficulty has to be introduced into the network at the same time. In other words, NAT444 is just a tool to make IPv6 transition easy to be swallowed. It is designed for the days IPv4 and IPv6 co-existence.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 5, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Definition of NAT444 Model	3
3. Behavior of NAT444 Model	4
4. Pros and Cons of NAT444 Model	5
4.1. Pros of NAT444 Model	5
4.2. Cons of NAT444 Model	5
5. Acknowledgements	6
6. IANA Considerations	6
7. Security Considerations	6
8. References	6
8.1. Normative References	6
8.2. Informative References	7
Appendix A. Example IPv6 Transition Scenario	7
Authors' Addresses	9

1. Introduction

The only permanent solution of the IPv4 address exhaustion is to deploy IPv6. Now, just before the exhaustion, it's time to make a transition to IPv6.

After the exhaustion, unless ISP takes any action, end users will not be able to get IPv4 address.

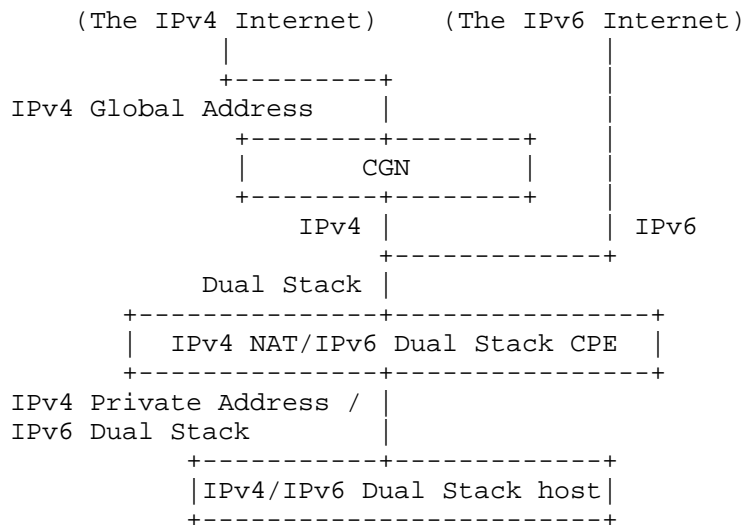
The servers that have only IPv4 address will continue to exist on the Internet after the IPv4 address exhaustion. In this situation, IPv6 only hosts cannot reach IPv4 only hosts.

This document explains NAT444 model that bridges the gap between the coming IPv6 Internet and the present IPv4 Internet.

2. Definition of NAT444 Model

NAT444 Model is a network model that uses two Network Address and Port Translators (NAPT) with three types of IPv4 address blocks.

The first NAPT is in CPE, and the second NAPT is in Carrier Grade NAT (CGN) [I-D.ietf-behave-lsn-requirements]. CGN is supposed to be installed in the ISP's network.



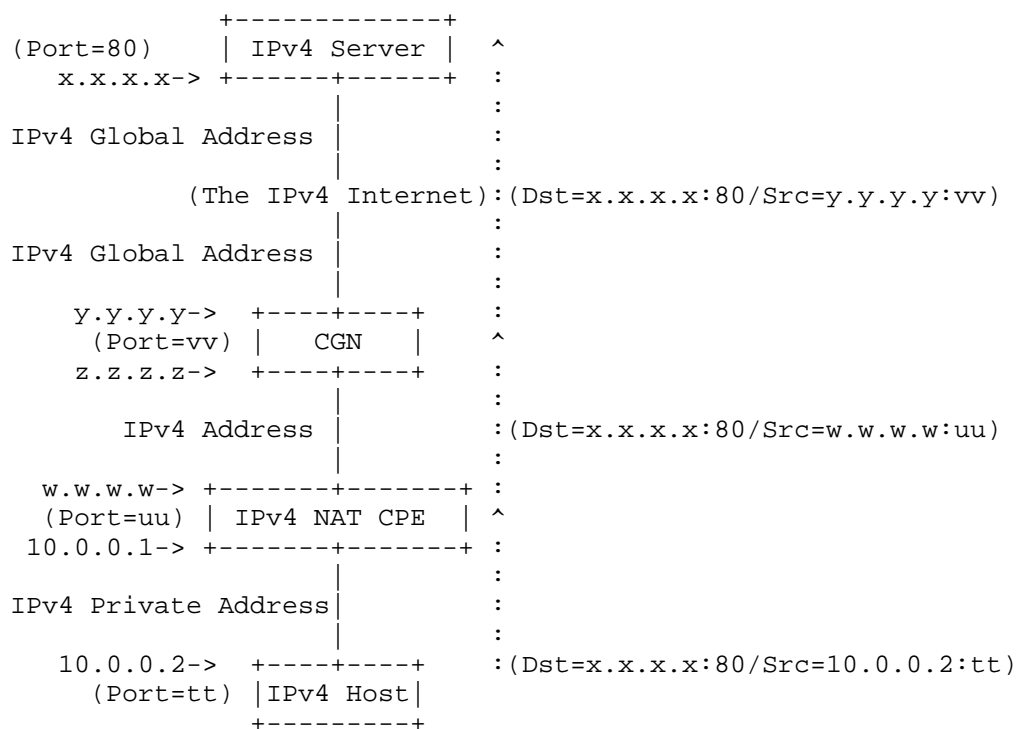
The first IPv4 address block is Private Address [RFC1918] inside CPE. The second one is an IPv4 Address block between CPEs and CGN. The third one is IPv4 Global Addresses that is outside CGN. The ISPs

using NAT444 provide IPv6 connectivity by dual stack model.

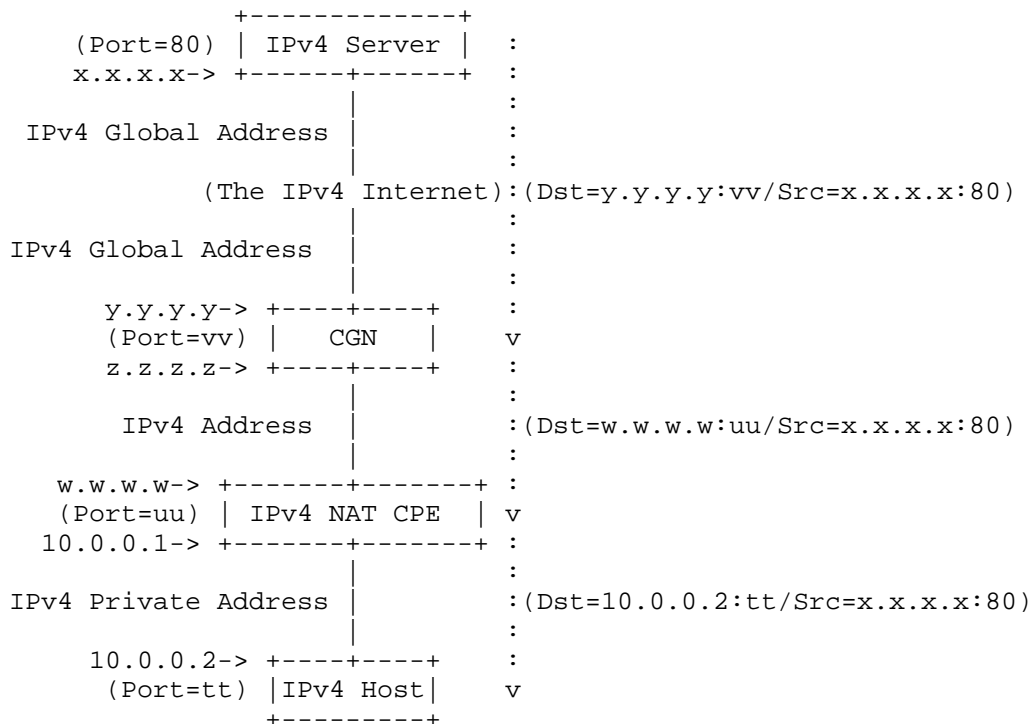
3. Behavior of NAT444 Model

The IPv6 packets from the host reach the IPv6 Internet without using NAT functionality.

The following figure shows the behavior of the IPv4 packet from the host to the IPv4 server via two NATs. The first NAT in CPE overwrites the Source IP Address and Source Port from 10.0.0.2:tt to w.w.w.w:uu. Then the second NAT in CGN overwrites them from w.w.w.w:uu to y.y.y.y:vv. Destination IP Address and Port are not overwritten.



The following figure explains the behavior of returning IPv4 packet via two NATs. The first NAT in CGN overwrites the Destination IP Address and Port Number from y.y.y.y:vv to w.w.w.w:uu. Then the second NAT in CPE overwrites them from w.w.w.w:u to 10.0.0.2:tt.



4. Pros and Cons of NAT444 Model

4.1. Pros of NAT444 Model

This network model has following advantages.

- This is the only network model that doesn't require replacing CPEs those are owned by customers.
- This network model is composed of the present technology.
- This network model doesn't require address family translation.
- This network model doesn't require DNS rewriting.
- This network model doesn't require additional fragment for the packets because it doesn't use tunneling technology.

4.2. Cons of NAT444 Model

This network model has some technical restrictions.

- Some application such as SIP requires special treatment, because IP address is written in the payload of the packet. Special treatment means application itself aware double NAT or both of two NATs

support inspecting and rewriting the packets.

- Because both IPv4 route and IPv6 route exist, it doubles the number of IGP route inside the CGN.
- UPnP doesn't work with double NATs.

5. Acknowledgements

Thanks for the input and review by Shin Miyakawa, Shirou Niinobe, Takeshi Tomochika, Tomohiro Fujisaki, Dai Nishino, JP address community members, AP address community members and JPNIC members.

6. IANA Considerations

There are no IANA considerations.

7. Security Considerations

Each customer inside a CGN looks using the same Global Address from outside an ISP. In case of incidents, the ISP must have the function to trace back the record of each customer's access without using only IP address.

If a Global Address of the CGN is listed on the blacklist, other customers who share the same address could be affected.

8. References

8.1. 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.
- [RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Software Problem Statement", RFC 4925, July 2007.
- [I-D.ietf-behave-lsn-requirements] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-07 (work in progress), June 2012.

8.2. Informative References

[I-D.shirasaki-isp-shared-addr]

Yamagata, I., Miyakawa, S., Nakagawa, A., Yamaguchi, J.,
and H. Ashida, "ISP Shared Address",
draft-shirasaki-isp-shared-addr-07 (work in progress),
January 2012.

[I-D.shirasaki-nat444-isp-shared-addr]

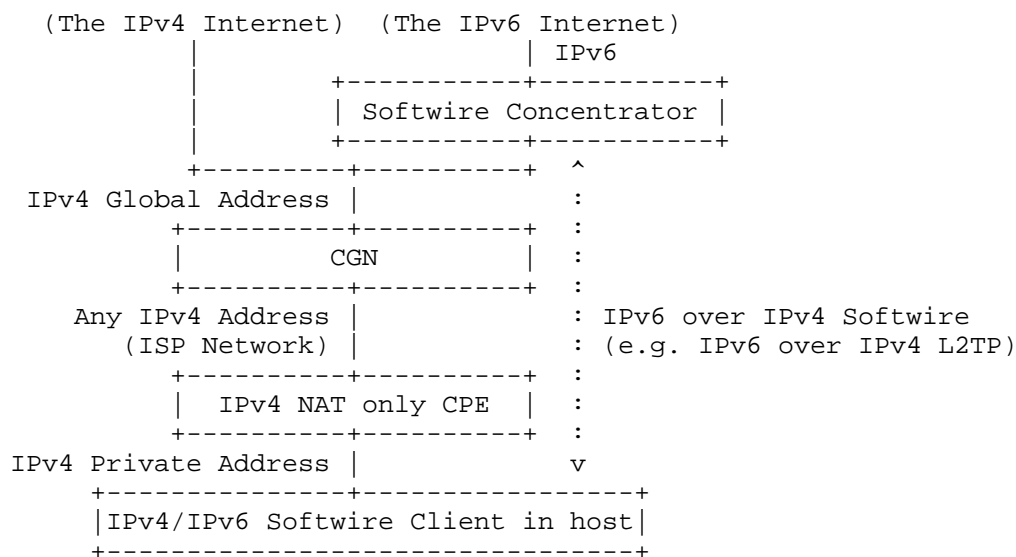
Yamaguchi, J., Shirasaki, Y., Miyakawa, S., Nakagawa, A.,
and H. Ashida, "NAT444 addressing models",
draft-shirasaki-nat444-isp-shared-addr-07 (work in
progress), January 2012.

Appendix A. Example IPv6 Transition Scenario

The steps of IPv6 transition are as follows.

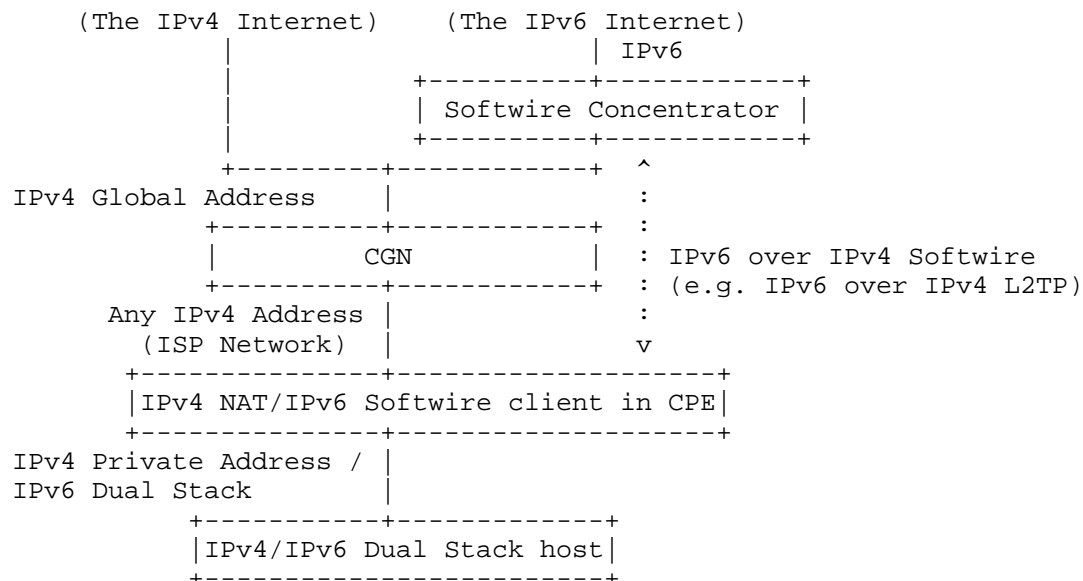
Step 1: Enabling software client in host

ISP provides IPv6 connectivity to customers with software [RFC4925].
ISP installs CGN and software concentrator in its network. A
software client in host connects to the IPv6 internet via ISP's
concentrator. ISP can use existing IPv4 equipments. Customers can
just use existing CPE.



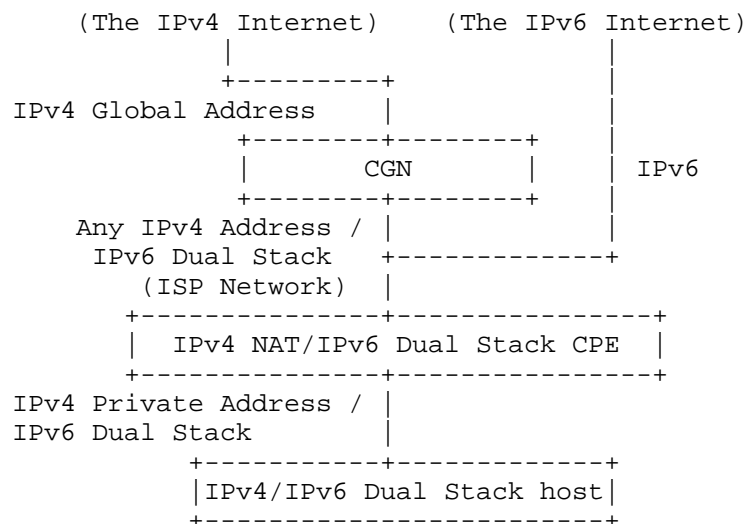
Step 2: Enabling software client in CPE

A customer enables software client in CPE. A software client in CPE connects to the IPv6 internet via ISP's concentrator. A Customer's network is now dual stack.



Step 3: Moving on to dual stack

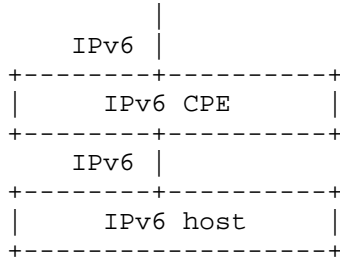
ISP provides dual stack access to CPE. A CPE uplink is now dual stack.



Step 4: Moving on to pure IPv6

IPv6 transition completes.

(The IPv6 Internet)



Authors' Addresses

Ikuhei Yamagata
NTT Communications Corporation
Granpark Tower 17F, 3-4-1 Shibaura, Minato-ku
Tokyo 108-8118
Japan

Phone: +81 3 6733 8671
Email: ikuhei@nttv6.jp

Yasuhiro Shirasaki
NTT Communications Corporation
NTT Hibiya Bldg. 7F, 1-1-6 Uchisaiwai-cho, Chiyoda-ku
Tokyo 100-8019
Japan

Phone: +81 3 6700 8530
Email: yasuhiro@nttv6.jp

Akira Nakagawa
Japan Internet Exchange Co., Ltd. (JPIX)
Otemachi Building 21F, 1-8-1 Otemachi, Chiyoda-ku
Tokyo 100-0004
Japan

Phone: +81 90 9242 2717
Email: a-nakagawa@jpix.ad.jp

Jiro Yamaguchi
Fiber 26 Network Inc.
Haraguchi bldg., 5F, 3-11-4 Kanda Jinbo-cho, Chiyoda-ku
Tokyo 101-0051
Japan

Phone: +81 50 3463 6109
Email: jiro-y@f26n.jp

Hiroyuki Ashida
IS Consulting G.K.
12-17 Odenma-cho, Nihonbashi, Chuo-ku
Tokyo 103-0011
Japan

Email: assie@hir.jp

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 6, 2013

J. Yamaguchi
Fiber 26 Network
Y. Shirasaki
S. Miyakawa
NTT Communications
A. Nakagawa
Japan Internet Exchange (JPIX)
H. Ashida
IS Consulting G.K.
July 5, 2012

NAT444 addressing models
draft-shirasaki-nat444-isp-shared-addr-08

Abstract

This document describes addressing models of NAT444. There are some addressing models of NAT444. The addressing models have some issues of network behaviors, operations, and addressing. This document helps network architects to use NAT444 after IPv4 address exhaustion.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 6, 2013.

Copyright Notice

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

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

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Addressing Models	4
2.1. Global Address	4
2.2. Private Address	4
2.2.1. Policy Based Routing Issue	4
2.2.2. Address Block Duplication Issue	4
2.2.3. Class-E Address (240/4)	4
2.2.4. ISP Shared Address	5
3. Example Architectures	5
3.1. Direct Routing inside CGN	5
3.2. CGN Bypassing	6
3.3. Global Address Customers inside CGN	7
4. Acknowledgements	7
5. IANA Considerations	8
6. Security Considerations	8
7. Normative References	8
Authors' Addresses	8

1. Introduction

NAT444 [I-D.shirasaki-nat444] is one of solutions after IPv4 address exhaustion. ISP can select some addressing models of NAT444. The addressing models have some issues of network behaviors, operations, and addressing. This document describes these issues and solutions. It boosts up to deploy the IPv6 Internet.

2. Addressing Models

The key of addressing model is the address block between Customer Premises Equipment (CPE) and Carrier Grade NAT (CGN) [I-D.ietf-behave-lsn-requirements]. It's mentioned in this section. The best addressing model is "ISP Shared Address" which is defined in [I-D.shirasaki-isp-shared-addr] and briefly described in this section.

2.1. Global Address

ISP cannot assign IPv4 Global Address any more after the exhaustion.

2.2. Private Address

It has two major problems.

2.2.1. Policy Based Routing Issue

If both source and destination address of the packet are inside CGN, it has to go through CGN. The reason is that some servers reject receiving packets when the source address of receiving packet is Private Address. Therefore packets have to go through the CGN for rewriting the source address from Private Address to Global Address. Additionally, if Private Address and Global Address co-exist inside CGN, the ISP has to use Policy Based Routing (PBR).

2.2.2. Address Block Duplication Issue

The Private Address in ISP's network could conflict with its customer's network address. Many CPEs between customer's network and ISP's network cannot route the packet under this situation. To avoid this, ISP has to negotiate with its all customers not to use the reserved Private Address block.

2.2.3. Class-E Address (240/4)

It is known that some equipment such as routers and servers reject packets from or to this address block. So, to use this address block

in ISP's network, ISP has to request its customers to replace their equipment. In addition to that, ISP might have to replace their equipment when it doesn't handle Class-E address packets properly.

2.2.4. ISP Shared Address

ISP Shared Address is the newly defined IPv4 address block that is to be allocated from IANA free pool. It doesn't have any problem. Spending some blocks from the exhausting IANA free pool could be regarded as a problem, but from long view, this problem is much smaller than its great merit. ISP Shared Address is defined in [I-D.shirasaki-isp-shared-addr].

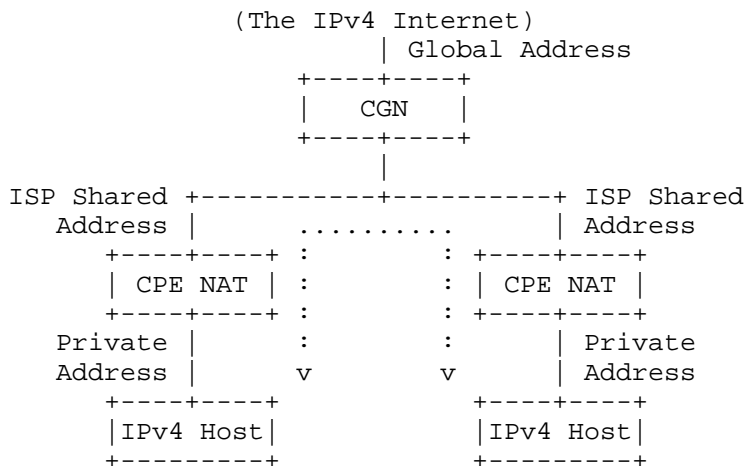
3. Example Architectures

This section explains example architectures how to design NAT444 with ISP Shared Address.

3.1. Direct Routing inside CGN

This architecture enables direct communication between customers inside same CGN. It has the following advantages.

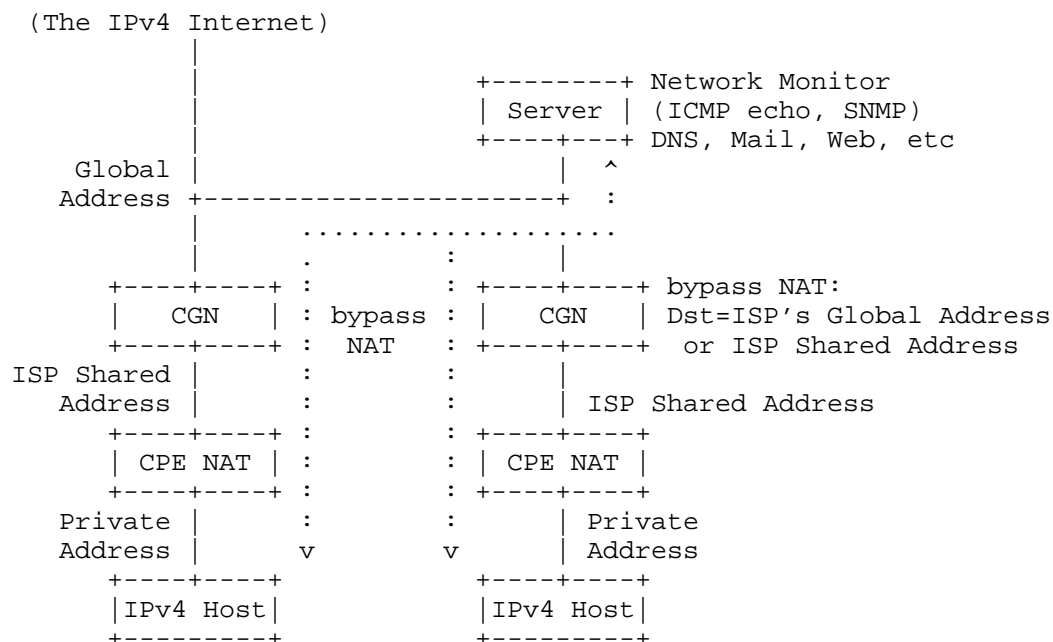
- o The packets don't go through CGN. (No hairpinning)
- o The customers inside CGN can use bidirectional applications (e.g. TV Conference, VPN).
- o No need to use Policy Based Routing.



3.2. CGN Bypassing

This architecture is bypassing the NAT function of CGN. It has the following advantage.

- o The customers inside an ISP can use bidirectional applications (e.g. TV Conference, VPN).
- o Any communication in single ISP doesn't consume CGN external port.
- o ISP's servers outside CGN can access CPE. (e.g. ICMP echo, SNMP, remote access)
- o ISP's servers outside CGN can distinguish which customer's connection it receives. (e.g. DNS, Mail)



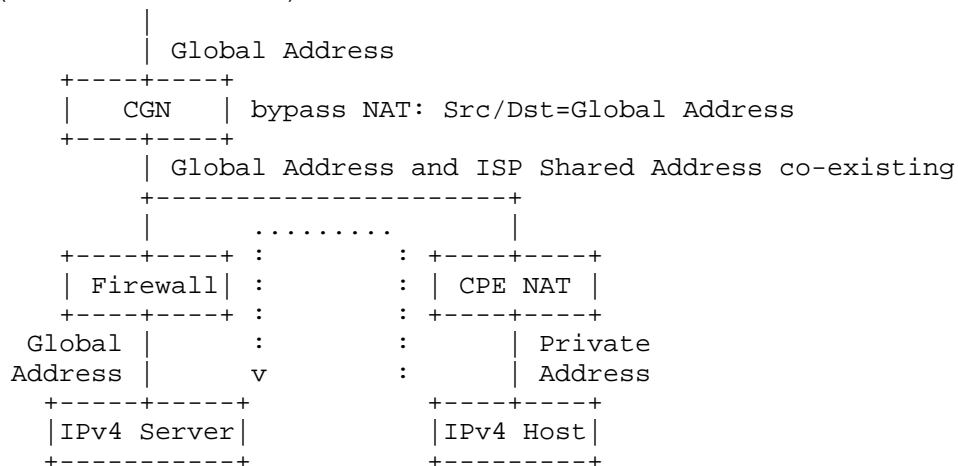
3.3. Global Address Customers inside CGN

This architecture enables co-existing Global Address and ISP Shared Address inside CGN.

It enables direct communications from ISP Shared Address customer to Global Address customer inside same CGN. It has the following advantage.

- o The ISP can put ISP Shared Address customer and Global Address customer in the same concentrator.
- o The customers inside CGN can use bidirectional applications (e.g. TV Conference, VPN).
- o No need to use Policy Based Routing.

(The IPv4 Internet)



4. Acknowledgements

Thanks for the input and review by Shirou Niinobe, Takeshi Tomochika, Tomohiro Fujisaki, Dai Nishino, JP address community members, AP address community members and JPNIC members.

5. IANA Considerations

IANA is to allocate a certain size of address block from IANA free pool. The size of it is described in [I-D.shirasaki-isp-shared-addr]

6. Security Considerations

There are no security considerations.

7. Normative References

[I-D.shirasaki-isp-shared-addr]

Yamagata, I., Miyakawa, S., Nakagawa, A., Yamaguchi, J.,
and H. Ashida, "ISP Shared Address",
draft-shirasaki-isp-shared-addr-07 (work in progress),
January 2012.

[I-D.ietf-behave-lsn-requirements]

Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A.,
and H. Ashida, "Common requirements for Carrier Grade NATs
(CGNs)", draft-ietf-behave-lsn-requirements-07 (work in
progress), June 2012.

[I-D.shirasaki-nat444]

Yamagata, I., Shirasaki, Y., Nakagawa, A., Yamaguchi, J.,
and H. Ashida, "NAT444", draft-shirasaki-nat444-05 (work
in progress), January 2012.

Authors' Addresses

Jiro Yamaguchi
Fiber 26 Network Inc.
Haraguchi bldg., 5F, 3-11-4 Kanda Jinbo-cho, Chiyoda-ku
Tokyo 101-0051
Japan

Phone: +81 50 3463 6109
Email: jiro-y@f26n.jp

Yasuhiro Shirasaki
NTT Communications Corporation
NTT Hibiya Bldg. 7F, 1-1-6 Uchisaiwai-cho, Chiyoda-ku
Tokyo 100-8019
Japan

Phone: +81 3 6700 8530
Email: yasuihiro@nttv6.jp

Shin Miyakawa
NTT Communications Corporation
Granpark Tower 17F, 3-4-1 Shibaura, Minato-ku
Tokyo 108-8118
Japan

Phone: +81 3 6733 8671
Email: miyakawa@nttv6.jp

Akira Nakagawa
Japan Internet Exchange Co., Ltd. (JPIX)
Otemachi Building 21F, 1-8-1 Otemachi, Chiyoda-ku
Tokyo 100-0004
Japan

Phone: +81 90 9242 2717
Email: a-nakagawa@jpix.ad.jp

Hiroyuki Ashida
IS Consulting G.K.
12-17 Odenma-cho, Nihonbashi, Chuo-ku
Tokyo 103-0011
Japan

Email: assie@hir.jp

