Network Working Group J. Arkko
Internet-Draft A. Keranen
Intended status: Informational Ericsson
Expires: August 10, 2012 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

Copyright Notice

Table of Contents

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