V6OPS
Internet-Draft
Intended status: Informational
Expires: September 13, 2011

B. Carpenter
Univ. of Auckland

March 12, 2011

Advisory Guidelines for 6to4 Deployment
draft-carpenter-v6ops-6to4-teredo-advisory-03

Abstract

This document provides advice to network operators about deployment of the 6to4 technique for automatic tunneling of IPv6 over IPv4. It is principally addressed to Internet Service Providers, including those that do not yet support IPv6, and to Content Providers. The intention of the advice is to minimise both user dissatisfaction and help desk calls.

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 September 13, 2011.

Table of Contents

1. Introduction

A technique for automatic tunneling of IPv6 over IPv4, intended for situations where a user may wish to access IPv6-based services via a network that does not support IPv6, was defined a number of years ago. It is known as 6to4 [RFC3056], [RFC3068] and is quite widely deployed in end systems, especially desktop and laptop computers. Also, 6to4 is supported in a number of popular models of CPE routers, some of which have it enabled by default, leading to quite widespread unintentional deployment by end users.

Unfortunately, experience shows that the method has some problems in current deployments that can lead to connectivity failures. These failures either cause long retry delays or complete failures for users trying to connect to services. In many cases, the user may be quite unaware that 6to4 is in use, and when the user contacts a help desk, in all probability the help desk is unable to correctly diagnose the problem. Anecdotally, many help desks simply advise users to disable IPv6, thus defeating the whole purpose of the mechanism, which was to encourage early adoption of IPv6.

There is additional discussion of operational issues in [I-D.vandevelde-v6ops-harmful-tunnels]. The main goal of the present document is to offer advice to network operators on how to deal with this situation more constructively than by disabling 6to4. It briefly describes the principle of operation, then describes the problems observed, and finally offers specific advice on the available methods of avoiding the problems. Note that some of this advice applies to ISPs that do not yet support IPv6, since their customers and help desks are significantly affected in any case. Other advice applies to content providers.

We do not discuss here details of this situation that are mainly outside the scope of network operators:

1. Operating system preferences between IPv4 and IPv6 when both appear to be available [I-D.ietf-6man-rfc3484-revise].
2. Ensuring that application software deals gracefully with connectivity problems [I-D.wing-v6ops-happy-eyeballs-ipv6].
3. Some content providers have chosen to avoid the problem by hiding their IPv6 address except from customers of pre-qualified networks [I-D.ietf-v6ops-v6-aaaa-whitelisting-implications].

Note to readers of earlier versions: references to Teredo have been removed from this document. Sorry about the file name.

2. Principles of Operation

There are two variants of 6to4 which are referred to here as "Router 6to4" and "Anycast 6to4". To understand Anycast 6to4, it is necessary first to understand Router 6to4.

2.1. Router 6to4

Router 6to4 is the original version, documented in [RFC3056]. The model assumes that a user site operates native IPv6, but that its ISP provides no IPv6 service. The site border router acts as a 6to4 router. If its external global 32-bit IPv4 address is V4ADDR, the site automatically inherits the IPv6 prefix 2002:V4ADDR::/48. (The explanation in RFC 3056 is somewhat confusing, as it refers to the obsolete "Top Level Aggregator" terminology.) The prefix 2002:V4ADDR::/48 will be used and delegated for IPv6 service within the user site.

Consider two such site border routers, with global IPv4 addresses 192.0.2.170 and 190.0.2.187, and therefore inheriting the IPv6 prefixes 2002:c000:2aa::/48 and 2002:c000:2bb::/48 respectively. The routers can exchange IPv6 packets by encapsulating them in IPv4 using protocol number 41, and sending them to each other at their respective IPv4 addresses. In fact, any number of 6to4 routers connected to the IPv4 network can directly exchange IPv6 packets in this way.

Some 6to4 routers are also configured as "Relay routers." They behave as just described, but in addition they obtain native IPv6 connectivity with a normal IPv6 prefix. They announce an IPv6 route to 2002::/16. For example, assume that the 6to4 router at 190.0.2.187 is a relay router, whose address on the 6to4 side is 2002:c000:2bb::1. Suppose that a host with the 6to4 address 2002:c000:2aa::123 sends an IPv6 packet to a native IPv6 destination such as 2001:db8:123:456::321. Assume that the 6to4 router at 192.0.2.170 has its IPv6 default route set to 2002:c000:2bb::1, i.e. the relay. The packet will be delivered to the relay, encapsulated in IPv4. After decapsulation, the relay will forward the packet into native IPv6 for delivery. When the remote host replies, the packet (source 2001:db8:123:456::321, destination 2002:c000:2aa::123) will find a route to 2002::/16 and hence be delivered to a 6to4 relay. The process will be reversed and the packet will be encapsulated and forwarded to the 6to4 router at 192.0.2.170 for final delivery.

Note that this process does not require the same relay to be used in both directions. The outbound packet will go to whichever relay is configured as the default IPv6 router at the source router, and the return packet will go to whichever relay is announcing a route to

2002::/16 in the vicinity of the remote IPv6 host.

There are of course many further details in RFC 3056, most of which are irrelevant to current operational problems.

2.2. Anycast 6to4

Router 6to4 assumes that 6to4 routers and relays will be managed and configured cooperatively. In particular, 6to4 sites need to find a relay router willing to carry their outbound traffic, which becomes their default IPv6 router (except for 2002::/16). The objective of the anycast variant, defined in [RFC3068], is to avoid any need for such configuration. The intention was to make the solution available for small or domestic users, even those with a single host or simple home gateway rather than a border router. This is achieved quite simply, by defining 192.88.99.1 as the default IPv4 address for a 6to4 relay, and therefore 2002:c058:6301:: as the default IPv6 router address for a 6to4 site.

Since Anycast 6to4 implies a default configuration for the user site, it does not require any particular user action. It does require an IPv4 anycast route to be in place to a relay at 192.88.99.1. As with Router 6to4, there is no requirement that the return path goes through the same relay.

3. Problems Observed

It should be noted that Router 6to4 was not designed to be an unmanaged solution. Quite the contrary: RFC 3056 contains a number of operational recommendations intended to avoid routing issues. In practice, there are few if any deployments of Router 6to4 following these recommendations. Mostly, Anycast 6to4 has been deployed. In this case, the user site (either a single host or a small broaband gateway) discovers that it doesn't have native IPv6 connectivity, but that it does have a global IPv4 address and can resolve AAAA queries, and therefore assumes that it can send 6to4 packets to 192.88.99.1.

Empirically, 6to4 appears to suffer from a significant level of connection failure; see <https://labs.ripe.net/Members/emileaben/6to4-how-bad-is-it-really> and <http://www.potaroo.net/ispcol/2010-12/6to4fail.html>. In experiments conducted on a number of dual stack web servers, the TCP connection failure rate has been measured. In these experiments, the client's connection attempt to a server was considered to have failed when the server received a TCP SYN packet and sent a SYN/ACK packet in response, but received no ACK packet to complete the initial TCP 3-way handshake. The experiment conducted by Aben recorded a failure

rate of between 9% and 20% of all 6to4 connection attempts. The experiment conducted by Huston has recorded a failure rate of between 9% and 19% of all 6to4 clients. In this latter experiment it was further noted that between 65% to 80% of all 6to4 clients who failed to connect using 6to4 were able to make a successful connection using IPv4, while the remainder did not make any form of IPv4 connection attempt, successful or otherwise, using the mapped IPv4 address as a source address. No connection attempts were recorded by the server using embedded RFC1918 IPv4 addresses.

There have been several possible reasons offered for this form of 6to4 connection failure. One is the use of private IPv4 addresses embedded in the 6to4 address, making the return path for the 6to4 tunnel infeasible, and the second is the use of local filters and firewalls that drop incoming IP packets that use IP protocol 41. If the former case were prevalent it would be reasonable to expect that a significant proportion of failed 6to4 connections would use embedded IPv4 addresses that are either drawn from the private use (RFC 1918) address ranges, contrary to RFC 3056, or from addresses that are not announced in the Internet's IPv4 inter-domain routing table. Neither case was observed to any significant volume in the experiments conducted by Huston. Furthermore, the experimental conditions were varied to use a return 6to4 tunnel with either the native IPv4 source address of the dual stack server or an IPv4 source address of 192.88.99.1. No change in the 6to4 connection failure rate was observed between these two configurations; however, other operators have reported significant problems when replying from the native address, caused by stateful firewalls at the user site. Given that the server used its own 6to4 relay for the return path, the only difference in the IP packet itself between the successful IPv4 connections and the failed 6to4 connections was the IP protocol number, which was 6 (TCP) for the successful IPv4 connections and 41 (IPv6 payload) for the failed 6to4 connections. The inference from these experiments is that one likely reason for the high connection failure rate for 6to4 connections is the use of local filters close to the end-user that block incoming packets with protocol 41.

In a dual stack context this connection failure rate was effectively masked by the ability of the client system to recover from the failure and make a successful connection using IPv4. In this case the only effect on the client system was a delay in making the connection of between 7 and 20 seconds as the client's system timed out on the 6to4 connection attempts (see [I-D.wing-v6ops-happy-eyeballs-ipv6]).

This experience and further analysis shows that specific operational problems with Anycast 6to4 include:

1. Outbound Black Hole: 192.88.99.1 does not generate 'destination unreachable' but in fact packets sent to that address are dropped. This can happen due to routing or firewall configuration, or even because the relay that the packets happen to reach contains an ACL such that they are discarded.

   This class of problem arises because the user's ISP is accepting a route to 192.88.99.0/24 despite the fact that it doesn't go anywhere useful. Either the user site or its ISP is dropping outbound Protocol 41 traffic, or the upstream operator is unwilling to accept incoming 6to4 packets from the user's ISP. The latter is superficially compatible with the design of Router 6to4 (referred to as "unwilling to relay" in RFC 3056). However, the simple fact of announcing a route to 192.88.99.0/24 in IPv4, coupled with the behavior described in RFC 3068, amounts to announcing a default route for IPv6 to all 6to4 sites that receive the IPv4 route. This violates the assumptions of RFC 3056.

   The effect of this problem on users is that their IPv6 stack believes that it has 6to4 connectivity, but in fact all outgoing IPv6 packets are black-holed. The prevalence of this problem is hard to measure, since the resulting IPv6 packets can never be observed from the outside.
2. Inbound Black Hole: In this case, 6to4 packets sent to 192.88.99.1 are correctly delivered to a 6to4 relay, and reply packets are returned, but they are dropped by an inbound Protocol 41 filter. As far as the user is concerned, the effect is the same as the previous case: IPv6 is a black hole. Many enterprise neworks are believed to be set up in this way. Connection attempts due to this case can be observed by IPv6 server operators, in the form of SYN packets from addresses in 2002::/16 followed by no response to the resulting SYN/ACK. From the experiments cited above, this appears to be a significant problem in practice.
3. No Return Relay: If the Outbound Black Hole problem does not occur, i.e. the outgoing packet does reach the intended native IPv6 destination, the target system will send a reply packet, to 2002:c000:2aa::123 in our example above. Then 2002::/16 may or may not be successfully routed. If it is not routed, the packet will be dropped (hopefully with 'destination unreachable'). According to RFC 3056, an unwilling relay "MUST NOT advertise any 2002:: routing prefix into the native IPv6 domain"; therefore, conversely, if this prefix is advertised the relay must relay packets regardless of source and destination. However, in practice the problem arises that some relays reject packets that they should relay, based on their IPv6 source address.

Whether the native IPv6 destination has no route to 2002::/16, or it turns out to have a route to an unwilling relay, the effect is the same: all return IPv6 packets are black-holed. While there is no direct evidence of the prevalence of this problem, it certainly exists in practice.

4. Large RTT: In the event that none of the above three problems applies, and a two-way path does in fact exist between a 6to4 host and a native host, the round trip time may be quite large and variable since the paths to the two relays are unmanaged and may be complex. Overloaded relays might also cause highly variable RTT.
5. PMTUD Failure: A common link MTU size observed on the Internet today is 1500 bytes. However, when using 6to4 the path MTU is less than this due to the encapsulation header. Thus a 6to4 client will normally see a link MTU that is less than 1500, but a native IPv6 server will see 1500. Path MTU Discovery does not always work, and this can lead to connectivity failures. Even if a TCP SYN/ACK exchange works, TCP packets with full size payloads may simply be lost. These failures are disconcerting even to an informed user, since a standard 'ping' from the client to the server will succeed, because it generates small packets, and the successful SYN/ACK exchange can be traced. Also, the failure may occur on some paths but not others, so a user may be able to fetch web pages from one site, but only ping another.
6. Reverse DNS Failure: Typically a 6to4-addressed host will not have a reverse DNS delegation. If reverse DNS is used as a pseudo-security check, it will fail.
7. Bogus Address Failure: By design, 6to4 does not work and will not activate itself if the available V4ADDR is a private address [RFC1918]. However it will also not work if the available V4ADDR is a "bogon", i.e. a global address that is being used by the operator as a private address. A common case of this is a legacy wireless network using 1.1.1.0/24 as if it was a private address. In this case, 6to4 will assume it is connected to the global Internet, but there is certainly no working return path.

   This failure mode will also occur if an ISP is operating a Carrier Grade NAT between its customers and the Internet, and is using global public address space as if it were private space to do so.
8. Faulty 6to4 Implementations: It has been reported that some 6to4 implementations attempt to activate themselves even when the available IPv4 address is an RFC 1918 address. This is in direct contradiction to RFC 3056, and will produce exactly the same failure mode as Bogus Address Failure. It is of course outside the ISP's control.

9. Difficult Fault Diagnosis: The existence of all the above failure modes creates a problem of its own: very difficult fault diagnosis, especially if the only symptom reported by a user is slow access to web pages, caused by a long timeout before fallback to IPv4. Tracking down anycast routing problems and PMTUD failures is particularly hard.

The practical impact of the above problems, which are by no means universal as there is considerable successful use of Anycast 6to4, has been measured at a fraction of 1% loss of attempted connections to content servers (see <http://www.fud.no/ipv6/>). While this seems low, it amounts to a significant financial impact for content providers. Also, end users frustrated by the poor response times caused by fall-back to IPv4 connectivity [I-D.wing-v6ops-happy-eyeballs-ipv6] are considered likely to generate help desk calls with their attendant costs.

A rather different operational problem caused incidentally by 6to4 is that, according to observations made by Tim Chown and James Morse at the University of Southampton, rogue Router Advertisements [RFC6104] predominantly convey a 2002::/16 prefix. This appears to be due to misbehaviour by devices acting as local IPv6 routers or connection-sharing devices but issuing RA messages on the wrong interface.

4. Advisory Guidelines

There are several types of operator involved, willingly or unwillingly, in the Anycast 6to4 scenario and they will all suffer if things work badly. There is a clear incentive for each of them to take appropriate action, as described below.

This document avoids formal normative language, because it is highly unlikely that the guidelines apply universally. Each operator will make its own decisions about which of the following guidelines are useful in its specific scenario.

4.1. Vendor Issues

Although this document is aimed principally at operators, there are some steps that implementers and vendors of 6to4 should take.

1. Some vendors of routers, including customer premises equipment, have not only included support for 6to4 in their products, but have enabled it by default. This is bad practice - it should always be a conscious decision by a user to enable 6to4. Many of the above problems only occur due to unintentional deployment of 6to4.

2. Similarly, host operating systems should not enable Anycast 6to4 by default; it should always be left to the user to switch it on.
3. Any 6to4 implementation that attempts to activate itself when the available IPv4 address is an RFC 1918 address is faulty and needs to be updated.
4. 6to4 implementations should adopt updated IETF recommendations on address selection [I-D.ietf-6man-rfc3484-revise].
5. 6to4 router or connection-sharing implementations must avoid issuing rogue RAs [RFC6104].

4.2. Consumer ISPs, and enterprise networks, that do not support IPv6 in any way

To reduce the negative impact of Anycast 6to4 deployed (probably unknowingly) by users, and consequent user dissatisfaction and help desk calls, such ISPs should check in sequence:

1. Does the ISP have a route to 192.88.99.1? (This means an explicit route, or knowledge that the default upstream provider has an explicit route. A default route doesn't count!)
2. If so, is it functional and stable?
3. If so, is the ping time reasonably short?
4. If so, does the relay willingly accept 6to4 traffic from the ISP's IPv4 prefixes? (Note that this is an administrative as well as a technical question - is the relay's operator willing to accept the traffic?)

Unless the answer to all these questions is 'yes', subscribers will be no worse off, and possibly better off, if the route to 192.88.99.1 is blocked and generates an IPv4 'destination unreachable'. There is little operational experience with this, however.

Some implementations also perform some form of 6to4 relay qualification. For example, one host implementation (Windows) tests the Protocol 41 reachability by sending an ICMPv6 echo request with Hop Limit=1 to the relay, expecting a response or Hop Limit exceeded error back. Lack of any response indicates that the 6to4 relay does not work so 6to4 is turned off [Savola].

A more constructive approach for such an ISP is to seek out a transit provider who is indeed willing to offer outbound 6to4 relay service, so that the answer to each of the questions above is positive.

In any case, such ISPs should always allow protocol 41 through their network and firewalls. Not only is this a necessary condition for 6to4 to work, but it also allows users who want to use a configured IPv6 tunnel service to do so.

Some operators, particularly enterprise networks, silently block

Protocol 41 on security grounds. Doing this on its own is bad practice, since it contributes to the problem and harms any users who are knowingly or unknowingly attempting to run 6to4. The strategic solution is to deploy native IPv6, making Protocol 41 redundant. In the short term, experimentation could be encouraged by allowing Protocol 41 for certain users, while returning appropriate ICMP responses as mentioned above. Unfortunately, if this is not done, the 6to4 problem cannot be solved.

Operators should never use "bogon" address space such as the example of 1.1.1.0/24 for customers, since IPv4 exhaustion means that all such addresses are likely to be in real use in the near future. (Also see [I-D.ietf-intarea-shared-addressing-issues].) An operator that is unable to immediately drop this practice should ensure that 192.88.99.1 generates IPv4 'destination unreachable'. It has been suggested that they could also run a dummy 6to4 relay at that address which always returns ICMPv6 'destination unreachable' as a 6to4 packet. However, these techniques are not very effective, since most current end-user 6to4 implementations will ignore them.

If an operator is providing legitimate global addresses to customers (neither RFC 1918 nor bogon addresses), and also running Carrier Grade NAT (Large Scale NAT) between this address space and the global address space of the Internet, then 6to4 cannot work properly. Such an operator should also take care to return 'destination unreachable' for 6to4 traffic. Alternatively, they could offer untranslated address space to the customers concerned.

A customer who is intentionally using 6to4 may also need to create AAAA records, and the operator should be able to support this, even if the DNS service itself runs exclusively over IPv4. However, customers should be advised to consider carefully whether their 6to4 service is sufficiently reliable for this.

Operators could, in principle, offer reverse DNS support for 6to4 users [RFC5158], although this is not straightforward for domestic customers.

Finally, enterprise operators who have complete administrative control of all end-systems may choose to disable 6to4 in those systems as an integral part of their plan to deploy IPv6.

4.2.1. 6to4 as the first step to IPv6 operation

An IPv4 operator could choose to install a well-managed 6to4 relay, connected to an IPv6-in-IPv4 tunnel to an IPv6 operator. This could serve as a small first step before the operator proceeds to native IPv6 deployment. The routing guidelines in Section 4.4 would apply.

4.3. Consumer ISPs, and enterprise networks, that do support IPv6

Once an operator does support IPv6 service, whether experimentally or in production, it is almost certain that users will get better results using this service than by continuing to use 6to4. Therefore, these operators are encouraged to advise their users to disable 6to4 and they should not create DNS records for any 6to4 addresses.

Such an operator may automatically fall into one of the following two categories (transit provider or content provider), so the guidelines in Section 4.4 or in Section 4.5 will apply instead.

Operators in this category should make sure that no routers are unintentionally or by default set up as active 6to4 relays. Unmanaged 6to4 relays will be a source of problems.

Operators in this category should consider whether they need to defend themselves against rogue RA messages [RFC6105].

4.4. Transit ISPs and Internet Exchange Points

We assume that transit ISPs and IXPs have IPv6 connectivity. To reduce the negative impact of Anycast 6to4 on all their client networks, it is strongly recommended that they each run an Anycast 6to4 relay service. This will have the additional advantage that they will terminate the 6to4 IPv4 packets, and can then forward the decapsulated IPv6 traffic according to their own policy. Otherwise, they will blindly forward all the encapsulated IPv6 traffic to a competitor who does run a relay.

It is of critical importance that routing to this service is carefully managed:

1. The IPv4 prefix 192.88.99.0/24 must be announced only towards client IPv4 networks whose outbound 6to4 packets will be accepted.
2. The IPv6 prefix 2002::/16 must be announced towards native IPv6. The relay must accept all traffic towards 2002::/16 that reaches it, so the scope reached by this announcement should be carefully planned. It must reach all client IPv6 networks of the transit ISP or IXP. If it reaches a wider scope, the relay will be offering a free ride to non-clients.
3. The evidence is mixed, but it seems best to ensure that when the relay sends 6to4 packets back towards a 6to4 user, they should have 192.88.99.1 as their IPv4 source address (not the relay’s unicast IPv4 address). This is to avoid problems if the user is behind a stateful firewall that drops inbound packets from addresses that have not been seen in outbound traffic.

4. The relay should be capable of responding correctly to ICMPv6 echo requests encapsulated in IPv4 protocol 41, typically with outer destination address 192.88.99.1 and inner destination address 2002:c058:6301::. (As noted previously, some 6to4 hosts are known to send echo requests with Hop Limit = 1, which allows them to rapidly detect the presence or absence of a relay in any case, but operators cannot rely on this behaviour.)
5. Protocol 41 must not be filtered in any IPv4 network or firewalls.
6. As a matter of general practice, which is essential for 6to4 to work well, IPv6 PMTUD must be possible, which means that ICMPv6 must not be blocked anywhere [RFC4890]. This also requires that the relay has a sufficiently high ICMP error generation threshold. For a busy relay, a typical default rate limit of 100 packets per second is too slow. On a busy relay, 1000pps or more might be needed. If ICMPv6 "Packet too Big" error messages are rate-limited, users will experience PMTUD failure.
7. The relay must have adequate performance, and since load prediction is extremely hard, it must be possible to scale it up or, perhaps better, to replicate it as needed. Since the relay process is stateless, any reasonable method of load sharing between multiple relays will do.
8. The relay must of course be connected directly to global IPv4 space, with no NAT.

Operators in this category should make sure that no routers are unintentionally or by default set up as active 6to4 relays. Unmanaged 6to4 relays will be a source of problems.

4.5. Content providers and their ISPs

We assume that content providers and their ISPs have IPv6 connectivity, and that content servers are dual stacked. There is a need to avoid the situation where a client host, configured with Anycast 6to4, suceeds in sending an IPv6 packet to the server, but the 6to4 return path fails as described above. To avoid this, there must be a locally positioned 6to4 relay. Large content providers are advised to operate their own relays, and ISPs should do so in any case. There must be a 2002::/16 route from the content server to the relay. As noted in the previous section, the corresponding route advertisement must be carefully scoped, since any traffic that arrives for 2002::/16 must be relayed.

Such a relay may be dedicated entirely to return traffic, in which case it need not respond to the 6to4 anycast address.

Nevertheless, it seems wisest to ensure that when the relay sends 6to4 packets back towards a 6to4 user, they should have 192.88.99.1

as their IPv4 source address (not the relay's unicast IPv4 address). As noted above, this is to avoid problems if the user is behind a stateful firewall that drops UDP packets from addresses that have not been seen in outbound traffic. However, it is also necessary that 192.88.99.1 is not blocked by upstream ingress filtering - this needs to be tested.

Without careful engineering, there is nothing to make the return path as short as possible. It is highly desirable to arrange the scope of advertisements for 2002::/16 such that content providers have a short path to the relay, and the relay should have a short path to the ISP border. Care should be taken about shooting off advertisements for 2002::/16 into BGP4; they will become traffic magnets. If every ISP with content provider customers operates a relay, there will be no need for any of them to be advertised beyond each ISP's own customers.

Protocol 41 must not be filtered in the ISP's IPv4 network or firewallls. If the relays are placed outside the content provider's firewall, the latter may filter protocol 41 if desired.

The relay must have adequate performance, and since load prediction is extremely hard, it must be possible to scale it up or, perhaps better, to replicate it as needed. Since the relay process is stateless, any reasonable method of load sharing between multiple relays will do.

The relay must of course be connected directly to global IPv4 space, with no NAT.

An option for content servers is to embed the relay function directly in the content server. This is in fact trivial, since it can be achieved by enabling a local 6to4 interface on the server, and using it to route 2002::/16 for outbound packets. (This might not allow use of 192.88.99.1 as the source address.) Further details are to be found at <http://www.potaroo.net/ispcol/2010-05/v6hints.html>. However, in this case Protocol 41 must be allowed by the firewalls.

Content providers who do embed the relay function in this way could, in theory, accept inbound 6to4 traffic as well. This is highly unadvisable since, according the the rules of 6to4, they would then have to relay traffic for other IPv6 destinations too. So they should not be reachable via 192.88.99.1. Also, they should certainly not create an AAAA record for their 6to4 address - their inbound IPv6 access should be native, and advertising a 6to4 address might well lead to uRPF ingress filtering problems.

To avoid the path MTU problem described above, content servers should

also set their IPv6 MTU to a safe value. From experience, 1280 bytes (the minimum allowed for IPv6) is recommended; again see <http://www.potaroo.net/ispcol/2010-05/v6hints.html>. Of course, ICMPv6 "Message too Big" must not be blocked or rate-limited anywhere [RFC4890].

Reverse DNS delegations are highly unlikely to exist for 6to4 clients, and are by no means universal for other IPv6 clients. Content providers (and in fact all service providers) should not rely on them as a pseudo-security check for IPv6 clients.

Operators and content providers should make sure that no routers are unintentionally or by default set up as active 6to4 relays. Unmanaged 6to4 relays will be a source of problems.

5. Tunnels Managed by ISPs

There are various ways, such as tunnel brokers [RFC3053], 6rd [RFC5969], L2TPv2 hub-and-spoke [RFC5571], and the proposed 6a44 [I-D.despres-softwire-6a44], by which Internet Service Providers can provide tunneled IPv6 service to subscribers in a managed way, in which the subscriber will acquire an IPv6 prefix under a normal provider-based global IPv6 prefix. Most of the issues described for 6to4 do not arise in these scenarios. However, for IPv6-in-IPv4 tunnels used by clients behind a firewall, it is essential that IPv4 Protocol 41 is not blocked.

As a matter of general practice, IPv6 PMTUD must be possible, which means that ICMPv6 "Message too Big" must not be blocked or rate-limited anywhere [RFC4890].

6. Security Considerations

There is a general discussion of security issues for IPv6-in-IPv4 tunnels in [I-D.ietf-v6ops-tunnel-security-concerns], and [I-D.ietf-v6ops-tunnel-loops] discusses possible malicious loops. [RFC3964] specifically discusses 6to4 security. In summary, tunnels create a challenge for many common security mechanisms, simply because a potentially suspect packet is encapsulated inside a harmless outer packet. All these considerations apply to the automatic mechanisms discussed in this document. However, it should be noted that if an operator provides well managed servers and relays for 6to4, non-encapsulated IPv6 packets will pass through well defined points (the native IPv6 interfaces of those servers and relays) at which security mechanisms may be applied.

A blanket recommendation to block Protocol 41 is not compatible with mitigating the 6to4 problems described in this document.

7. IANA Considerations

This document makes no request of the IANA.

8. Acknowledgements

Useful comments and contributions were made by Emile Aben, Tore Anderson, Jack Bates, Cameron Byrne, Remi Despres, Jason Fesler, Wes George, Geoff Huston, Eric Kline, Victor Kuarsingh, Martin Levy, David Malone, Martin Millnert, Keith Moore, Gabi Nakibly, Michael Newbery, Pekka Savola, Mark Smith, Nathan Ward, James Woodyatt, and others.

This document was produced using the xml2rfc tool [RFC2629].

9. Change log

draft-carpenter-v6ops-6to4-teredo-advisory-03: updated with additional security reference and addditional comments, 2011-03-12

draft-carpenter-v6ops-6to4-teredo-advisory-02: updated after further comments, removed references to Teredo, 2011-02-24

draft-carpenter-v6ops-6to4-teredo-advisory-01: updated after WG discussion, 2011-02-10

draft-carpenter-v6ops-6to4-teredo-advisory-00: original version, 2011-02-03

10. Informative References

[I-D.despres-softwire-6a44]
Despres, R., Carpenter, B., and S. Jiang, "Native IPv6 Across NAT44 CPEs (6a44)", draft-despres-softwire-6a44-01 (work in progress), October 2010.

[I-D.ietf-6man-rfc3484-revise]
Matsumoto, A., Kato, J., and T. Fujisaki, "Update to RFC 3484 Default Address Selection for IPv6", draft-ietf-6man-rfc3484-revise-01 (work in progress), October 2010.

[I-D.ietf-intarea-shared-addressing-issues]
Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", draft-ietf-intarea-shared-addressing-issues-05 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-ietf-v6ops-tunnel-loops-04 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-security-concerns]
Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns With IP Tunneling", draft-ietf-v6ops-tunnel-security-concerns-04 (work in progress), October 2010.

[I-D.ietf-v6ops-v6-aaaa-whitelisting-implications]
Livingood, J., "IPv6 AAAA DNS Whitelisting Implications", draft-ietf-v6ops-v6-aaaa-whitelisting-implications-03 (work in progress), February 2011.

[I-D.vandevelde-v6ops-harmful-tunnels]
Velde, G., Troan, O., and T. Chown, "Non-Managed IPv6 Tunnels considered Harmful", draft-vandevelde-v6ops-harmful-tunnels-01 (work in progress), August 2010.

[I-D.wing-v6ops-happy-eyeballs-ipv6]
Wing, D. and A. Yourtchenko, "Happy Eyeballs: Trending Towards Success with Dual-Stack Hosts", draft-wing-v6ops-happy-eyeballs-ipv6-01 (work in progress), October 2010.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

[RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, December 2004.

[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, May 2007.

[RFC5158] Huston, G., "6to4 Reverse DNS Delegation Specification", RFC 5158, March 2008.

[RFC5571] Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)", RFC 5571, June 2009.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, February 2011.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 2011.

[Savola] Savola, P., "Observations of IPv6 Traffic on a 6to4 Relay", ACM SIGCOMM CCR 35 (1) 23-28, 2006.

Author's Address

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Internet Engineering Task Force G. Chen
Internet-Draft H. Deng
Intended status: Informational China Mobile
Expires: September 15, 2011 March 14, 2011

NAT64-CPE Mode Operation for Opening Residential Service
draft-chen-v6ops-nat64-cpe-01

Abstract

The document has proposed an approach of NAT64-CPE mode, which would give residential service opportunities to be accessed by remote subscribers going through IPv6 networks. The document captures the fundamental NAT64 Functionalities with special cares to fit into CPE scenarios and don’t need cooperate with DNS64 any more. In addition, the CPE mode also allows IPv4 residential hosts to access IPv6 service. It will compatible with legacy residential hosts/servers and no further updates requirements to public DNS system.

Table of Contents

1. Introduction

Recently,IPv6 transition is fairly prevalent due to the depletion of IPv4 soon enough. However, the large number of installed CPE is IPv4-only based and likely to remain for several years. Considering the existing deployment approaches, majority of ISP assigned private IPv4 address to their customers,including residential servers and hosts. The nature of private IPv4 would block the end-to-end bi-directional communications. On the other hand, the goal of Internet services is to offer users ubiquitous experiences. User will be certainly supposed to able to enjoy such conveniences regardless of where we are. Therefore, ISP would take advantage of the accessibilities of residential services to provide plenty of services. Fortunately, IPv6 will get ISP end-to-end benefits. During IPv6 migration period, NAT64-CPE mode could overcome the obstacles to achieve final goals.

The document is aimed at proposing an approach of NAT64-CPE mode, which would give residential service opportunities to be accessed by remote subscribers going through IPv6 networks. The document captures the fundamental NAT64[NAT64] functionalities with special cares to fit into CPE scenarios. In these scenarios, the NAT64-CPE don’t need cooperate with DNS64[DNS64] any more, whereby this mechanism allows an IPv6-only client (i.e. either a host with only IPv6 stack, or a host with both IPv4 and IPv6 stack, but only with IPv6 connectivity or a host running an IPv6 only application) to initiate communications to an IPv4-only residential service server.

In addition, such CPE mode also allows IPv4 residential hosts to be capable of accessing IPv6 service. The packages generated by IPv4 hosts can be translated by NAT64-CPE, which will communicate with IPv6 servers.

The document is structured as follows. Section 2 describes appropriate scenario the NAT64-CPE mode fit to. Section 3 enumerates various functional parts for NAT64-CPE operation. Section 4 focus on the benefits the NAT64-CPE could bring. Section 5 is further securities consideration.

2. NAT64-CPE Mode Scenario Overviews

2.1. Communications between IPv6 hosts and IPv4 residential servers

Figure 1 illustrates a possible network scenario where an IPv6-only client attached to a dual-stack network, but the destination server is running on a private site where there is NAT64-CPE numbered with public IPv6 addresses and private IPv4 addresses. DNS is located in

dual stack Internet for naming-resolving.

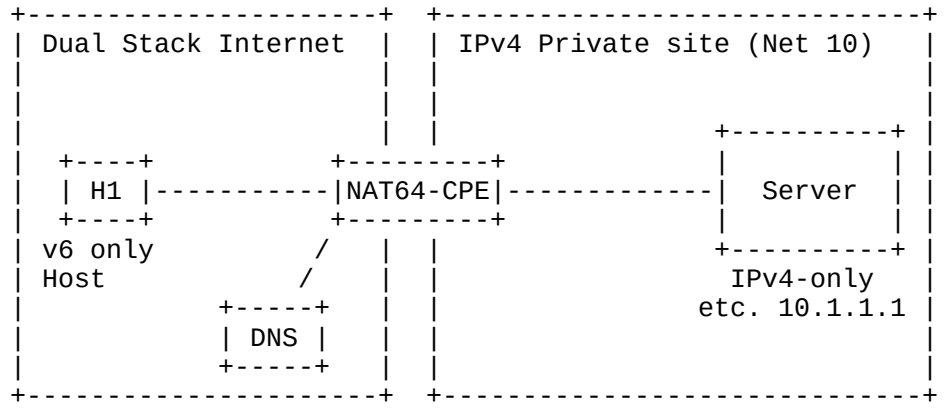


Figure 1: NAT64-CPE Network Scenario 1

This scenario appears in ISP network quite popular. As the instances, visitors go through distant network to take care of family affairs, like monitoring house security via residential camera, manipulating household appliances remotely prior to comeback home.

2.2. Communications between IPv6 servers and IPv4 residential hosts

Figure 2 illustrates a possible network scenario where an IPv4-only client connects to NAT64-CPE , and the destination server is running in IPv6 network. DNS is located in IPv6 network for naming-resolving. NAT6-CPE will have DNS-ALG capabilities for resolving IPv4 DNS query.

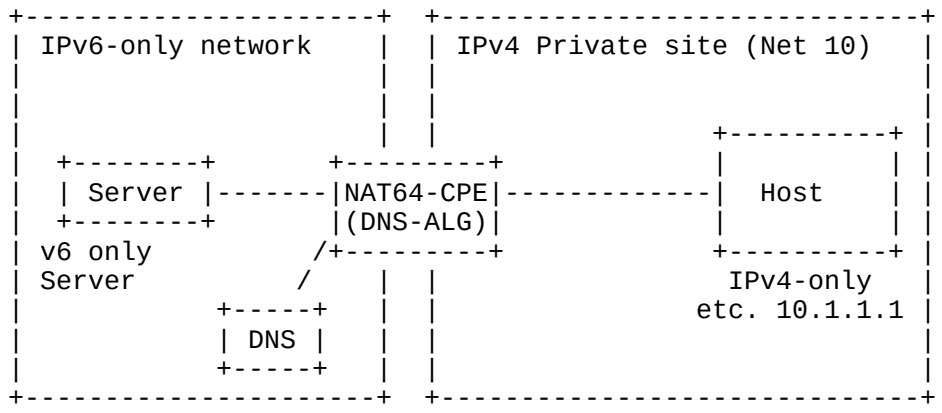


Figure 2: NAT64-CPE Network Scenario 2

This scenario could allow legacy IPv4 host to access IPv6 services. There are no additional requirements for IPv4 host.

3. NAT64-CPE Mode Operation

The whole process of NAT64-CPE operation involves CPE, DNS and addressing mechanism. This section illustrates different parts of functionalities for each scenario.

3.1. NAT64-CPE Operations for Scenario 1

3.1.1. CPE Functionalities Description

Two kinds of functions the NAT64-CPE would take on. First, it will perform the functionalities that normal CPE does except NAT44 forwarding, like assigning private IPv4 address to their attached residential servers. Additionally, CPE will allocate private IPv4 address to the servers depending on the server MAC address. Therefore, the server could always get constant private IPv4 address.

Second, CPE should carry NAT64 capable mode without integrating DNS64. According to normative handing, NAT64-CPE translates in-coming IPv6 destination address by striping NAT64 IPv6-prefix and maintains a IPv4 pool for translating IPv6 sources address. Therein, the NAT64 IPv6 prefix will be NSP specified in IPv6 Addressing of IPv4/IPv6 Translators. And, ISP will reserve distinct NSP for each CPE.

The prerequisite here is that NAT64-CPE should maintain address

mapping between inner IP address and outer IP address. PCP [PCP] could handle such problems. But that goes beyond the scope of this draft. Also, NAT64-CPE would install ALG, but it is optional.

3.1.2. DNS Configuration Consideration

Each residential services should be represented by FQDN format so as to users could easily remember and understand. The corresponding naming resource record should be stored as AAAA. The record's IPv6 address is synthesized by NAT64 prefix and private IPv4 address. The IPv6 format is compliant with assembling IPv6 address in DNS64.

The deployed DNS just follow regular DNS handling. There are no demands for performing DNS64 process.

3.1.3. NAT64-CPE Flow Example for Scenario 1

Figure 3 demonstrates the NAT64-CPE Mode operation flow, in which IPv6 host initiate service interaction with residential server remotely. The detailed actions that different entities performed was described afterwards.

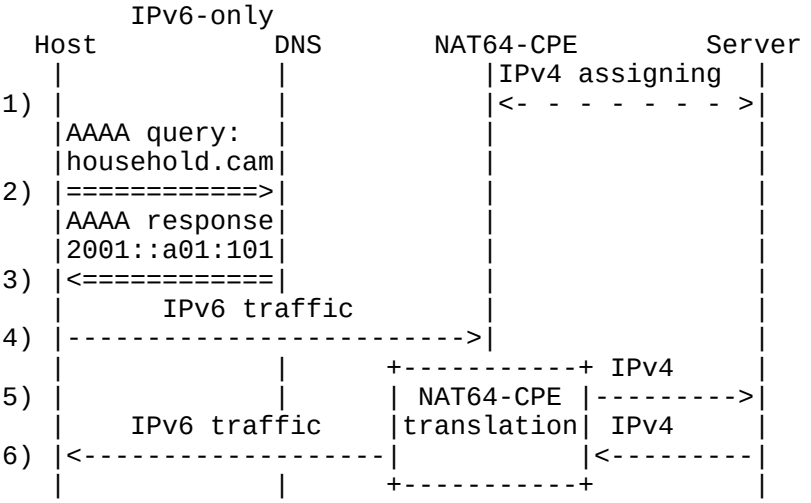


Figure 3: NAT64-CPE Flow Example for Scenario 1

(1) NAT64-CPE should be configured with NAT64 prefix, which is allocated by ISP. In that case, the NAT64 prefix is 2001::/64. NAT64-CPE assign private IPv4 address to the servers depending on the server MAC address. And, NAT64-CPE already has maintained the

mapping between inner IP address and outer IP address.

(2)IPv6-only host initiates AAAA query for resolving service name, for example, that is household.cam.

(3)DNS response AAAA record to the previous query. The IPv6 address of this service is synthesized by NAT64 prefix and assigned private IPv4 address. That is 2001::a01:101

(4)IPv6-only host send IPv6 traffic targeting to 2001::a01:101. The IPv6 traffic is routed to CPE.

(5)NAT64-CPE detects incoming IPv6 packets and algorithmically translated to IPv4 addresses by using the algorithm defined in [I-D.ietf-behave-address-format]. The translated IPv4 traffic is headed to IPv4-only residential server and perform somehow process.

(6)The residential IPv4 server responses these requests by IPv4 traffic, which will be sent to CPE. CPE performes reversed algorithm to translate IPv4 to IPv6 based on the maintained mapping information. And then, CPE generate IPv6 traffic and transmit to IPv6-only Host.

3.2. NAT64-CPE Operations for Scenario 2

3.2.1. CPE Functionalities Description

In this scenario, CPE is integrated with DNS-ALG, which will accept IPv4 DNS A query and translated into A and AAAA. After name-resolving is finished in IPv6 network, DNS-ALG take different actions according to DNS responses.

- DNS response only contains AAAA response. DNS-ALG will translate AAAA into A response. Meanwhile, a private IPv4 is required to be assigned to DNS A response. while, NAT64-CPE creates corresponding IPv6->IPv4 mapping in NAT6-CPE.

- DNS response contains AAAA and A response. DNS-ALG will return A response to IPv4 hosts. There are no additional states NAT64-CPE need to create.

- DNS response only contains A response. DNS-ALG will return A response to IPv4 hosts. There are no additional states NAT64-CPE need to create.

When IPv4 hosts get DNS responses through NAT64-CPE, it would create packets to communicate with remote servers. When the packet reaches the CPE box, CPE would recognize if the flow is belonging to existing

IP address mapping by determining IPv4 destination address and port. if CPE found matching records in NAT cache, NAT64-CPE would assign IPv6 address and port to translate the tuple of (Source Address, Source TCP port). Otherwise, NAT64-CPE will do NAT44 forwarding.

3.2.2. NAT64-CPE Flow Example for Scenario 2

Figure 4 demostrates the NAT64-CPE Mode operation flow, in which Internal IPv4 host initiate service interaction with remote IPv6 server.

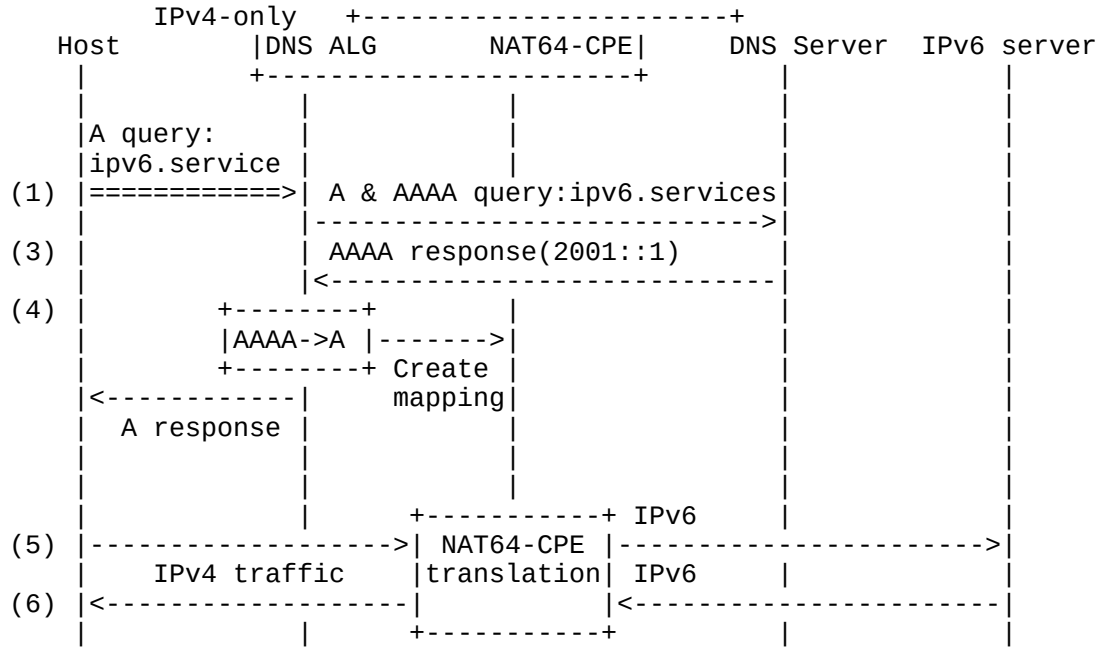


Figure 4: NAT64-CPE Flow Example for Scenario 2

4. NAT64-CPE Approach Discussion

Considering above description, NAT64-CPE has following specific features.

- o NAT64-CPE is capable of making residential server to be accessed, by means of which users could visit the IPv4-only server remotely.

- o NAT64-CPE is a solely NAT64 deployed solution in CPE environment. It will compatible with legacy residential servers and no further updates requirements to DNS. Therefore, it's liable to be deployed.

- o NAT64-CPE allows legacy IPv4 hosts to access IPv6 service. There are no additional requirements for IPv4 hosts.

5. Security Considerations

Essentially, there are strong demands to have thorough security mechanism to prevent privacy invasion in CPE scenario. The detailed considerations need to be further identified.

6. IANA Considerations

This memo includes no request to IANA.

7. Informative References

[DNS64] Bagnulo, M., "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", IETF Internet-draft draft-ietf-behave-dns64-10.txt, July 2010.

[NAT64] Bagnulo, M., "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12.txt (work in progress), July 2010.

[PCP] Wing, D., "Pinhole Control Protocol (PCP)", draft-ietf-pcp-base-06.txt (work in progress), February 2011.

Authors' Addresses

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

Email: chengang@chinamobile.com

Hui Deng
China Mobile
53A,Xibianmennei Ave.
Beijing 100053
P.R.China

Phone: +86-13910750201
Email: denghui02@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 3, 2012

G. Chen
China Mobile

Oct 2011

NAT64 Operational Considerations
draft-chen-v6ops-nat64-cpe-03

Abstract

The document has summarized NAT64 usages on different modes, in which NAT64 may serve for a large-scale network or would give enterprise or residential service opportunities to be accessed by IPv6 remote subscribers. The document has described different operations for each usage and proposed operational considerations for each particular NAT64-mode.

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 April 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

With fast developments of global Internet, the demands for IP address are rapidly increasing at present.This year, IANA announced that the global free pool of IPv4 depleted on 3 February. IPv6 is the only real option on the table. Operators have to accelerate the process of deploying IPv6 networks in order to address IP address strains.IPv6 deployment normally involves a step-wise approach where parts of the network should properly updated gradually. As IPv6 deployment progresses it may be simpler for operators and ICP/ISP to employ NAT64[RFC6146] functionalities at edge of IPv4 and IPv6 networks, since a significant part of network will still stay in IPv4 for long time. Especially, NAT64 could facilitate large ICP/ISP IPv6 transition process by eliminating upgrations of tremendous legacy IPv4 servers. Therefore, it’s quite popular to deploy NAT64 at the front of IDC to shift the entire service to be IPv6-enable.

Depending on different usage, NAT64 could be deployed on different places. The document has summarized NAT64 usages on different modes. Considering the existing deployment approaches, the memo has proposed different operational consideration for each particular NAT64-mode.

2. NAT64-CGN Deployment

2.1. Deployment in IDC

NAT has widely used in data center environments whenever IDC have to make your IPv4-only content available to IPv6 clients.

Figure 1 illustrates the usage where an IPv6-only host would like to initiate communications with IDC in IPv4 domain through NAT64. The NAT64 would accept IPv6 incoming session and distribute them to multiple IPv4 servers.

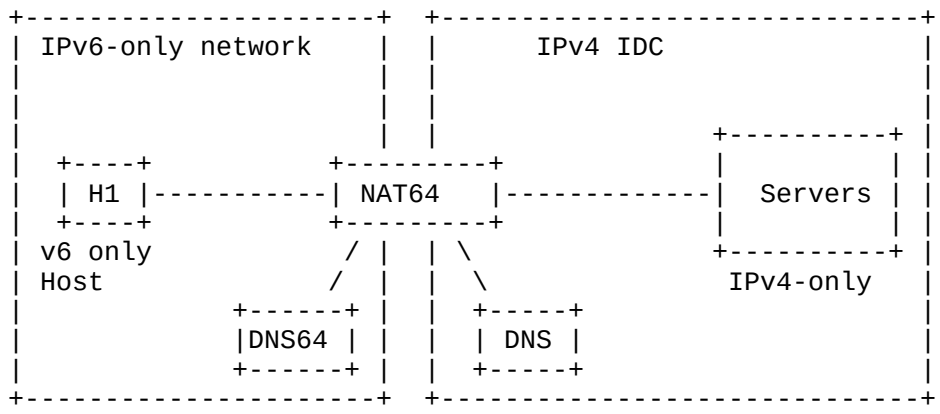


Figure 1: NAT64-CGN Mode Usage

NAT64 device in IDC may also take responsibilities of load balancer, which can accept incoming TCP/UDP sessions on a single virtual IPv6 interface or multiple IPv6 interfaces. Afterwards, it distributes them according to a specific algorithm it uses to multiple IPv4 servers. Ideally you could have a mix of IPv4 and IPv6 servers sitting behind the virtual IPv6 address.

Therein, NAT64 has to pick a new source IPv4 address and associated port number from local IPv4 address pool. DNS64 is a logical function that synthesizes DNS resource records(e.g., AAAA records containing IPv6 addresses) from DNS resource records actually contained in the DNS (e.g., A records containing IPv4 addresses).

2.2. Connecting with IPv4 Internet

NAT64 may also be used to connecting IPv6 users with IPv4 Internet. In this cases, NAT64 could colocated with BNG or Core Router to map legacy IPv4 servers into a NAT64 prefix and performs 6-to-4 address.

Therein, NAT64 would perform protocol translation mechanism and address translation mechanism. Protocol translation from an IPv4 packet header to an IPv6 packet header and vice versa is performed according to the IP/ICMP Translation Algorithm [RFC6145]. Address translation maps IPv6 transport addresses to IPv4 transport addresses and vice versa.

Following illustrates normal process for this usage.

- Step1: IPv6-only host performs an AAAA DNS query to DNS64 for the IPv6 address of the Pv4-only sever.
- Step2: DNS64 could not find the IPv6 address of the IPv4-only sever. So it tries to get the IPv4 address of the Pv4-only sever by sending A DNS query to DNS4.
- Step3: DNS4 return the A record to the DNS64.
- Step4: DNS64 map the IPv4 address to IPv6 address and send a synthetic AAAA record which is translated from A record to IPv6-only host.
- Step5: IPv6-only host send the IPv6 packet to the NAT64. NAT64 translates the IPv6 packet to IPv4 packet and send it to IPv4-only server.

2.3. NAT64-CGN Mode Requirements

According to above description for NAT64-CGN, the NAT64-CGN requirements are listed as following.

NAT64-CGN-R1: Each NAT64 device MUST have at least one unicast IPv6 prefix assigned to it, denoted Pref64::/n.

NAT64-CGN-R2:A NAT64 MUST have one or more unicast IPv4 addresses assigned to it.

NAT64-CGN-R3:Irrespective of the transport protocol used, the NAT64 MUST silently discard all incoming IPv6 packets containing a source address that contains the Pref64::/n.

NAT64-CGN-R4:The NAT64 MUST only process incoming IPv6 packets that contain a destination address that contains Pref64::/n. Likewise, the NAT64 MUST only process incoming IPv4 packets that contain a destination address that belongs to the IPv4 pool assigned to the NAT64.

NAT64-CGN-R5:NAT64 MUST support the algorithm for generating IPv6 representations of IPv4 addresses defined in RFC6052 as Address Translation Algorithms.

NAT64-CGN-R6:For incoming packets carrying TCP or UDP fragments with a non-zero checksum, NAT64 MAY elect to queue the fragments as they arrive and translate all fragments at the same time.

NAT64-CGN-R7: For incoming IPv4 packets carrying UDP packets with a zero checksum, if the NAT64 has enough resources, the NAT64 MUST

reassemble the packets and MUST calculate the checksum. If the NAT64 does not have enough resources, then it MUST silently discard the packets.

NAT64-CGN-R8: The NAT64 MAY require that the UDP, TCP, or ICMP header be completely contained within the fragment that contains fragment offset equal to zero.

NAT64-CGN-R9:The NAT64 MUST limit the amount of resources devoted to the storage of fragmented packets in order to protect from DoS attacks.

NAT64-CGN-R10:The NAT64 MUST make fragmentation process when MTU of incoming IPv4 traffic exceed maximum MTU on IPv6 side.

NAT64-CGN-R11:The NAT64 MAY let hosts and applications know IPv6 prefix used by the NAT64 and DNS64 so as to hosts have knowledge whether synthetic IPv6 address is targeted.

NAT64-CGN-R12:The NAT64 MAY decouple with DNS64 in order to establish communication with IPv4-only servers.

NAT64-CGN-R13:The NAT64 MAY take load-balancing functionalities incorporating with DNS64.

3. NAT64-CE Mode

NAT64-CE mode represents usages where there NAT64 is closed to customer edges, like enterprise network edge or residential network edge.

3.1. NAT64 at Enterprise Network Edge

Some enterprise would like to offers their employees with IPv6 access. However, the service may still stay in IPv4 domain. NAT64 useges in enterprise network could help shift all enterprise service to be IPv6 enable.

Figure 2 illustrates a network usage where an IPv6-only client attached to a dual-stack network, but the destination server is running on a private site where there is NAT64-CE numbered with public IPv6 addresses and private IPv4 addresses.

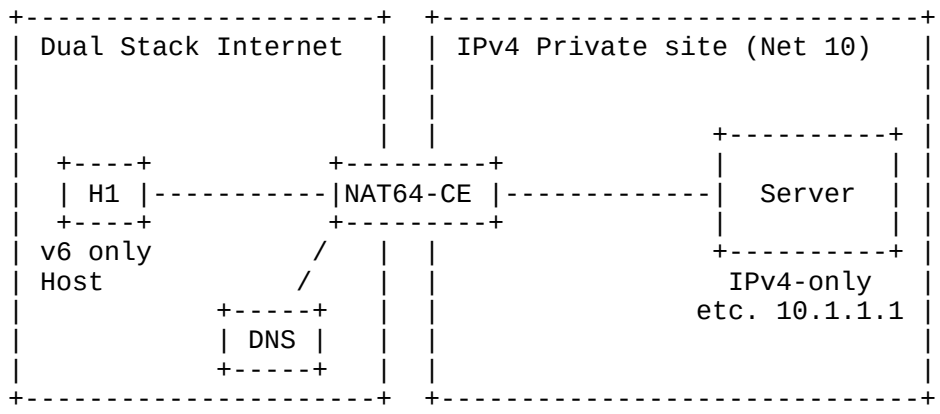


Figure 2: NAT64-CPE Mode Usage

3.2. NAT64 at Residential Network Edge

Residential servers are usually going beyond the operator's management. They may not able to IPv6-enable due to limitations of application supporting. In this case, ISP is still assigning private IPv4 address to servers. However, The nature of private IPv4 would block the end-to-end bi-directional communications. On the other hand, IPv6 will bring end-to-end benefits to operators. NAT64-CPE mode could let IPv6 users to access such IPv6-disable services in residential areas.

This scenario may appears in ISP network for several cases. As the instances, visitors go through distant network to take care of family affairs, like monitoring house security via residential camera, manipulating household appliances remotely prior to comeback home.

4. Security Considerations

Essentially, there are strong demands to have thorough security mechanism to prevent privacy invasion in NAT64-CPE scenario. The detailed considerations need to be further identified.

5. IANA Considerations

This memo includes no request to IANA.

6. Normative References

[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.

[RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

Author's Address

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

Email: chengang@chinamobile.com

IPv6 Operations
Internet-Draft
Intended status: Informational
Expires: September 15, 2011

T. Chown
University of Southampton
S. Venaas
Cisco Systems
March 14, 2011

World IPv6 Day Call to Arms
draft-chown-v6ops-call-to-arms-01

Abstract

The Internet Society (ISOC) has declared that June 8th 2011 will be World IPv6 Day, on which organisations are being encouraged to test their production IPv6 deployment capability. Many significant content providers and networks have stated they will take part. Given this date is likely to see more IPv6 traffic flowing across the Internet than has ever been seen before, it seems timely to issue a call to arms for operators and administrators to review and mitigate common causes of IPv6 connectivity problems. The increased traffic on World IPv6 Day should also create an excellent opportunity to observe the behaviour and performance of IPv6; it is thus very desirable to have appropriate measurement tools in place in advance. We discuss some appropriate tools from the network and application perspective.

Table of Contents

1. Introduction

Despite the recent exhaustion of the available IPv4 address pool, deployment of IPv6 remains limited. To help encourage organisations to trial production deployment, ISOC has declared June 8th 2011 as World IPv6 Day. Organisations are encouraged to use this day to test IPv6 in production, either by enabling clients in their network, or by making externally-facing services available over IPv6. At the current time, this would generally mean enabling dual-stack networking with IPv4 running alongside IPv6. However, IPv6-only networks are inevitable, and so some sites may choose to use June 8th to undertake some focused tests on that deployment model.

The purpose of this document is two-fold. One is to discuss common IPv6 connecivity issues that are likely to arise on June 8th, with a focus on dual-stack networking (which is likely to be how the vast majority of sites take part). This should help raise awareness of those problems and possible mitigations. The other is to encourage organisations to think about how they might get useful instrumentation on what happens in and to/from their networks on the day. Such measurement tools are likely to also be useful longer term - once deployed they could be left in place.

For most sites providing content, June 8th will be a chance to make some public facing services available over IPv6, such as web content using their production domain (e.g. www.example.com) rather than a contrived IPv6 test domain (e.g. www.ipv6.example.com). But also some enterprise sites may choose to enable IPv6 in user/client subnets, in which case the performance of those systems and the applications they run will be of paramount interest.

The document also includes a brief section on tools that might be used to test IPv6-only operation.

NB. This is a very rough draft of the document; feedback is welcomed. The scope of the document is purely informational to provoke discussion, and to encourage deployment steps for June 8th, which may remain in place after that date. We aim to have a relatively mature informational text ready well in advance of June.

2. Connectivity Issues

In this section we review some common causes of IPv6 connectivity issues. The topics below include initial thoughts for this early draft, and there is no significance to the order in which issues are listed. Some issues, such as transit arrangements, are not included - currently the focus is on end sites (or users) who may take part in

the World IPv6 Day.

2.1. Unmanaged Tunnels

One cause of connectivity problems is the use of unmanaged tunnels, in particular 'automated' methods that are not provisioned by the user's ISP. The most common example is 6to4 [RFC3056], or more specifically the 6to4 relay approach described in [RFC3068]. A native IPv6 host communicating with a 6to4 host will require both hosts to have access to an appropriately capable 6to4 relay (which may or may not be the same relay). If a host in a native IPv6 network has no route to 2002::/16 it cannot send traffic to a 6to4 host. Similarly, a 6to4 router that cannot reach the well-known IPv4 anycast relay address cannot send traffic to a native IPv6 network.

One approach to this problem is to encourage sites/ISPs to run local relays, as discussed in [I-D.carpenter-v6ops-6to4-teredo-advisory]. The alternative to reduce such problems is simply to obsolete 6to4, as proposed in [I-D.troan-v6ops-6to4-to-historic].

2.2. Tunnel Broker first-hop delays

IPv6 tunnel brokers, such as those provided by SixXS (http://www.sixxs.net) and Hurricane Electric (http://tunnelbroker.net) provide a more robust, managed approach to IPv6-in-IPv4 tunnelled access than 6to4. Individual users interested in IPv6 access for World IPv6 Day, in the absence of IPv6 support from their ISP, should consider registering to use a free tunnel broker. When choosing a broker service, it is prudent to pick one with a presence near to you that has a minimal round trip time. Providers such as SixXS and HE have tunnel broker servers in many countries. Beware picking a broker in another continent that may add 150ms+ to your round trip times.

2.3. Connection Timeouts

Where dual-stack systems - or rather the applications running on them - have a choice of IPv4 or IPv6 connectivity, timeouts can occur if there is no connectivity on the preferred protocol. For example, if both A and AAAA DNS records exists for a web server, and IPv6 connectivity is broken, there is likely to be some timeout for the browser before the connection drops back to IPv4.

A bigger problem exists if the application or OS tries IPv6 first and then does not fall back to IPv4. A bug in versions of Opera prior to 10.5 caused such behaviour, which was obviously a big issue for Opera users trying to access dual-stack web sites with broken IPv6 connectivity.

The author has undertaken some informal tests at his own site, which shows how different operating systems handle ICMP unreachables, if they are received. On Linux, web connections timeout after 20 seconds for 'no response', but immediately for unreachables. In contrast, Windows Vista was 20 seconds regardless of unreachables being received. Any non-trivial delay will cause significant user frustration.

This problem is probably the main reason that Google implemented a AAAA whitelisting system for its test sites. The sites had to demonstrate they had good IPv6 connectivity before being allowed into the test programme. The topic is discussed in [I-D.ietf-v6ops-v6-aaaa-whitelisting-implications]. For the sake of World IPv6 Day, it is expected that no such whitelisting is in place - that is, after all, the point of having a day dedicated to testing IPv6 in production.

An interesting suggestion to handle the problem is the 'happy eyeballs' approach described in [I-D.ietf-v6ops-happy-eyeballs]. This approach is now also being suggested for multiple interface systems, as per [I-D.chen-mif-happy-eyeballs-extension]. However some people feel this 'workaround' is simply masking underlying problems that should be fixed.

2.4. PMTU Discovery

IPv6 mandates that fragmentation is only undertaken by the sending node, and thus IPv6 requires working PMTU Discovery [RFC1981]. An existing RFC gives Recommendations for Filtering ICMPv6 Messages in Firewalls [RFC4890]; if this guidance is not followed, connectivity problems are likely to arise. Blindly filtering all ICMPv6 messages is not good practise.

The minimum MTU for IPv6 is 1280 bytes. Where PMTUD is not working or not implemented, the minimum MTU should be used.

2.5. Rogue Router Advertisements

Within a site, hosts may use IPv6 Stateless Address Autoconfiguration (SLAAC) [RFC4862]. However, it is possible for accidental (or malicious) rogue RAs to cause connectivity issues, as described in the Rogue Router Advertisement Problem Statement [RFC6104].

A typical cause of rogue RAs is Windows ICS, which can present a rogue 6to4 router on its wireless interface. This will cause hosts to potentially autoconfigure two global IPv6 addresses and pick the wrong default router, with unpredictable results. As a (bad) example the author experienced a scenario where he had a rogue 6to4 RA, but

because the rogue 6to4 was working he was able to access IPv6 networks outside his own network, but could not access most internal hosts inside his own network because he was unwittingly using 6to4 from outside into his own network, and thus being firewalled from those internal hosts.

In many cases, default address selection [RFC3484] (and [I-D.ietf-6man-rfc3484-revise]) would avoid such cases, because the address selection rules should prefer, or can be configured to prefer, native IPv6 over 6to4. However not all operating systems implement RFC 3484 yet, in particular MacOS X.

Adding ACLs to your switches to block ICMPv6 Type 134 packets on ports that do not have routers connected would also minimise rogue RAs. A more elegant solution is RA Guard [RFC6105], and another is use of SEcure Neighbour Discovery (SEND) [RFC3971]. However neither is widely implemented yet. Indeed, any reported operational experience of SEND in an enterprise network would be very welcome.

Finally, there is a tool called RAmond, available freely from http://ramond.sourceforge.net, that can be configured to detect and issue deprecating RAs against observed rogue RAs. This software is based on rafixd.

2.6. Tunnel performance

In scenarios where sites currently have manually configured tunnels to gain IPv6 connectivity, it may be the case that such encapsulation is performed by a router’s CPU, in which case unexpected high volumes of traffic may cause problems. Bear in mind that on World IPv6 Day, you may start using IPv6 by default for some high bandwidth applications that you had not used before, e.g. YouTube from Google.

2.7. AAAA record advertised but service not enabled

If enabling a service for World IPv6 Day, be aware of other existing services that may be running on the same system. If a server has multiple functions, all services should be IPv6 enabled before a AAAA record is entered into the DNS for services that may use that name.

A related consideration is to make sure that firewalls don’t just drop IPv6 packets to ports that are not in use. It’s better if the firewall or host sends a TCP RST to avoid a potential timeout. For example, if you add a AAAA record for your web server that also runs say FTP, where FTP is IPv4 only, either the firewall should have port 21 open or the firewall should be configured to send a TCP RST.

3. Instrumentation

In this section we discuss potential instrumentation approaches that may be configured for World IPv6 Day, and then retained longer term after the event. These should complement informal, subjective reports from users at participating sites. It is probably prudent to make users aware of the 'at risk' day, and actions they should take should the experience problems. It may also be desirable to undertake some form of user survey soon afterwards.

3.1. IPv6 traffic levels

It should be possible to measure raw IPv6 traffic levels independently on dual-stack switch/router platforms, given implementations of appropriate MIBs. Sites should take steps to ensure they have the tools in place to be able to view the relative levels of IPv4 and IPv6 traffic over time.

Application level measurement is also desirable, because handling of choice (preference) of protocol used lies with the application if both A and AAAA records are returned. Sites should be aware that due to IPv6 Privacy Extensions [RFC4941] application logs may show more apparent different clients connecting, due to clients cycling the source IPv6 address they use over time.

3.2. Network flow records

Where available, sites should seek to deploy network flow records for traffic, to maximise opportunities to analyse traffic patterns after the event, or in the case of reports of specific problems. Netflow v9 supports IPv6. Open source IPv6-capable Netflow collectors also exist, e.g. nfsen, from http://nfsen.sourceforge.net.

3.3. Client Web Access Success Rate

There have been some recent studies on the capabilities of web clients to access content on dual-stack servers by IPv4 or IPv6 in the presence of both A and AAAA records existing for a web domain.

One good example is that of [Anderson10], as reported at RIPE-61, where the author set up some application (web server) oriented tests for his newspaper content in Norway. The methodology was to add an invisible IFRAME to his site that would include IMG links randomly to 1x1 images that were served either via an IPv4-only target or a dual-stack target. Variation in the hit rates would imply IPv6 brokenness. By analysing the http metadata information could be gleaned on the cause of the brokenness. Results in Q4'2009 showed 0.2-0.3% brokenness, including the Opera bug mentioned above.

Recent figures published by Google suggest at most a 0.1% level of brokenness, indicating some improvement, but that level is still potentially 1 in 1000 users with a problem. Sites may wish to make their own measurements of IPv6 brokenness rather than relying on third party reports.

3.4. IPv4 Performance Comparison

Where a dual-stack service is deployed, measuring the relative performance of both protocols is desirable. This may primarily be a measurement of throughput or delay, but may also include availability/uptime measurement. A site may choose to set up its own performance measuring framework, for example using open source bandwidth and throughput test tools.

3.5. Security monitoring

We mentioned RAmond above in the context of watching for rogue RAs. There is another useful package called NDPmon, also available freely from http://ndpmon.sourceforge.net, that can be configured to watch for certain types of IPv6 'abuse' on your local network. It may be interesting to run the tool to confirm whether any 'bad' traffic is observed within your network on World IPv6 Day.

4. IPv6-only testing

The long-term IPv6 deployment plan is IPv6-only networking, rather than dual-stack. It is not clear how quickly significant IPv6-only networks will emerge, but testing of approaches to IPv6-only operation is desirable as soon as possible.

Some experience of NAT64 [I-D.ietf-behave-v6v4-xlate-stateful] has been described in [I-D.tan-v6ops-nat64-experiences], though this appears to have used only NAT-PT so far. An implementation of NAT64 is available at http://ecdysis.viagenie.ca. Operational experience of IVI is also desirable. An implementation of IVI is available at http://www.ivi2.org/IVI.

5. Conclusions

With the ISOC World IPv6 Day event due on June 8th 2011, this document aims to help focus attention on both improving awareness and mitigations of common causes of IPv6 connectivity problems, and encouraging sites and organisations to introduce appropriate instrumentation into their networks so they can observe traffic behaviour appropriately.

This is a very early version of the text, and is very drafty. All comments are very welcome towards a mature version in advance of June.

6. Security Considerations

There are no extra security consideration for this document.

7. IANA Considerations

There are no extra IANA consideration for this document.

8. Acknowledgments

To be added.

9. Informative References

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, May 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement

Problem Statement", RFC 6104, February 2011.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 2011.

[I-D.carpenter-v6ops-6to4-teredo-advisory]
Carpenter, B., "Advisory Guidelines for 6to4 Deployment", draft-carpenter-v6ops-6to4-teredo-advisory-03 (work in progress), March 2011.

[I-D.ietf-v6ops-happy-eyeballs]
Wing, D. and A. Yourtchenko, "Happy Eyeballs: Trending Towards Success with Dual-Stack Hosts", draft-ietf-v6ops-happy-eyeballs-00 (work in progress), March 2011.

[I-D.tan-v6ops-nat64-experiences]
Tan, J., Lin, J., and W. Li, "Experience from NAT64 applications", draft-tan-v6ops-nat64-experiences-00 (work in progress), March 2011.

[I-D.troan-v6ops-6to4-to-historic]
Troan, O., "Request to move Connection of IPv6 Domains via IPv4 Clouds (6to4) to Historic status", draft-troan-v6ops-6to4-to-historic-01 (work in progress), March 2011.

[I-D.ietf-v6ops-v6-aaaa-whitelisting-implications]
Livingood, J., "IPv6 AAAA DNS Whitelisting Implications", draft-ietf-v6ops-v6-aaaa-whitelisting-implications-03 (work in progress), February 2011.

[I-D.chen-mif-happy-eyeballs-extension]
Chen, G., "Happy Eyeballs Extension for Multiple Interfaces", draft-chen-mif-happy-eyeballs-extension-00 (work in progress), March 2011.

[I-D.ietf-6man-rfc3484-revise]
Matsumoto, A., Kato, J., and T. Fujisaki, "Update to RFC 3484 Default Address Selection for IPv6", draft-ietf-6man-rfc3484-revise-02 (work in progress), March 2011.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers",

draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[Anderson10]
Anderson, T., "Measuring and Combating IPv6 Brokenness", 2010, <http://ripe61.ripe.net/presentations/162-ripe61.pdf>.

Authors' Addresses

Tim Chown
University of Southampton
Highfield
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

Stig Venaas
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: stig@cisco.com

IPv6 Operations
Internet-Draft
Intended status: Informational
Expires: December 9, 2011

T. Chown
University of Southampton
M. Ford
Internet Society
S. Venaas
Cisco Systems

June 7, 2011

World IPv6 Day Call to Arms
draft-chown-v6ops-call-to-arms-03

Abstract

The Internet Society (ISOC) has declared that June 8th 2011 will be World IPv6 Day, on which some major organisations are going to make their content available over IPv6. With the likes of Google and Facebook providing IPv6 access to their production services and domains, it is very likely we will see more IPv6 traffic flowing across the Internet than has ever been seen before. With this in mind, it seems timely to issue a call to arms for systems and network administrators to review their organisation's IPv6 capabilities in order to mitigate common causes of IPv6 connectivity problems in advance of the day. The increased traffic on World IPv6 Day should also create an excellent opportunity to observe the behaviour and performance of IPv6; it is thus very desirable to have appropriate measurement tools in place in advance. We discuss some appropriate tools from the network and application perspective.

Status of this Memo

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

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

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

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

Copyright Notice

Table of Contents

1. Introduction

Despite the recent exhaustion of the available IPv4 address pool, deployment of IPv6 remains limited. To help encourage organisations to trial production deployment, ISOC has declared June 8th 2011 as World IPv6 Day [ISOC]. Organisations are encouraged to use this day to test IPv6 in production by making their main, externally-facing websites available over IPv6. Sites planning to turn on IPv6 for access in their network in the interest of World IPv6 Day should ensure this is completed well before the day, and commit to leaving it active after the event, and thus using the method they would choose to do so indefinitely. At the current time, this would generally mean enabling dual-stack networking with IPv4 running alongside IPv6. However, IPv6-only networks are ultimately inevitable, and so some sites may choose to use June 8th to undertake some focused tests on that deployment model.

The purpose of this document is two-fold. One is to discuss common IPv6 connectivity issues that are likely to arise on June 8th, with a focus on dual-stack networking (which is likely to be how the vast majority of sites take part). Most of the issues discussed in this text are those that would affect an end site or enterprise network running IPv6, but may be applicable elsewhere. Highlighting the issues should help raise awareness of those problems and possible mitigations. The other purpose is to encourage organisations to think about how they might get useful instrumentation in place to observe what happens in and to/from their networks on the day, both from the network and application perspective. Such measurement tools are likely to be useful in the longer term, so once deployed they could be left in place beyond June 8th.

For sites providing content, June 8th will be a chance to make some public facing services available over IPv6, most likely web content using their production domain (e.g. www.example.com) rather than a contrived IPv6 test domain (e.g. www.ipv6.example.com). Enabling public-facing Internet services is a reasonable first step for any organisation deploying IPv6. For ISPs, supporting IPv6 for their Internet-facing services (web, mail, etc.) and recording the impact of World IPv6 Day on their IPv4-only customers is an appropriate action. For sites enabling clients, doing so initially in their IT department may be appropriate; for educational sites enabling IPv6 on eduroam wireless networks could be appropriate given the underlying 802.1x authentication technology is IP version independent.

It should be emphasised that while World IPv6 Day is in many senses an ’experiment’ or ’test flight’ for IPv6, organisations should strongly consider deploying IPv6 in exactly the same robust way that they would do if they were deploying IPv6 and leaving it enabled

indefinitely. Similarly, applying measures to improve IPv6 robustness, e.g. improved ICMPv6 filtering practice, should be considered long term benefits. That they 'affect' the experiment is not a problem; indeed all measures that improve the robustness of IPv6 deployment should be seen as worthwhile. There will still be problems found, but these can at least be recognised and work done to make them better.

The document also includes a brief section on tools that might be used to test IPv6-only operation.

The scope of this document is purely informational to provoke discussion.

2. Connectivity Issues

In this section we review some common causes of IPv6 connectivity issues, oriented towards those that end sites or enterprises may have some ability to influence or mitigate. Some issues, such as transit arrangements, are not included - currently the focus is on end sites (or users) who may take part in the World IPv6 Day. Some IPv6 connectivity test sites are emerging, for example [testipv6]. There is no significance to the order in which issues are listed.

2.1. Unmanaged Tunnels

One cause of connectivity problems is the use of unmanaged tunnels, in particular 'automated' methods that are not provisioned by the user's ISP. The most common example is 6to4 [RFC3056], or more specifically the 6to4 relay approach described in [RFC3068]. A native IPv6 host communicating with a 6to4 host will require both hosts to have access to an appropriately capable 6to4 relay (which may or may not be the same relay). If a host in a native IPv6 network has no route to 2002::/16 it cannot send traffic to a 6to4 host. Similarly, a 6to4 router that cannot reach the well-known IPv4 anycast relay address cannot send traffic to a native IPv6 network. There are also potential issues with Protocol 41 filtering at site borders close to the client.

A presentation by Geoff Huston at IETF80 [Huston2011] highlighted the connection failure rates with 6to4, measured in excess of 15%, as well as the additional latency in 6to4 communications, with 6to4 showing an average additional 1.2s latency per retrieval.

One approach to this problem is to encourage sites/ISPs to run local relays, as discussed in [I-D.carpenter-v6ops-6to4-teredo-advisory]. This draft discusses how to make 6to4 more robust in situations where

there is a conscious decision to use it. Sites using 6to4 should consider deploying local relays to increase the chance of a good IPv6 experience. The alternative to reduce such problems is simply to move 6to4 to Historic, as proposed in [I-D.troan-v6ops-6to4-to-historic]. This would mean 6to4 would not be enabled by default anywhere, and once its usage had reduced enough, relays could be turned off.

There may still be some CPE routers that do enable 6to4 by default; it is likely that devices behind such routers will experience problems on World IPv6 Day.

Connection failures and latency with the Teredo protocol [RFC4380] were also highlighted by Geoff Huston's IETF80 presentation. Teredo connection failure rates were as high as 35%, with 1-3s additional latency. One of the connection issues is reliance on the ICMPv6 probe packet being able to reach the destination host; in practice filters may block these. Thus Teredo should not be considered a reliable means of accessing the IPv6 Internet.

2.2. Tunnel Broker first-hop delays

IPv6 tunnel brokers, such as those provided by SixXS (http://www.sixxs.net) and Hurricane Electric (http://tunnelbroker.net) provide a more robust, managed approach to IPv6-in-IPv4 tunnelled access than 6to4. Individual users interested in IPv6 access for World IPv6 Day, in the absence of IPv6 support from their ISP, should consider registering to use a free tunnel broker. It would be sensible to register for and test your broker client well in advance of IPv6 Day, and ideally plan to keep it available beyond that date, until your ISP provides IPv6 natively for you. One set of test sites to use would be the list cited on the ISOC World IPv6 Day site [ISOCsites].

When choosing a broker service, it is prudent to pick one with a presence near to you that has a minimal round trip time. Providers such as SixXS and HE have tunnel broker servers in many countries. Beware picking a broker in another continent that may add 150ms+ to your round trip times.

2.3. Connection Timeouts

One of the main drivers for IPv6 Day is identifying and fixing the problems that can lead to connection timeouts. Because unreliable IPv6 connectivity leads to intensely frustrating problems for end-users, it is essential that people motivated to deploy IPv6 connectivity, whether for themselves, or for a larger network, only do so in a well-supported, production-quality fashion.

Where dual-stack systems - or rather the applications running on them - have a choice of IPv4 or IPv6 connectivity, timeouts can occur if there is no connectivity on the preferred protocol. For example, if both A and AAAA DNS records exists for a web server, and IPv6 connectivity is broken, there is likely to be some timeout for the browser before the connection drops back to IPv4.

A bigger problem exists if the application or OS tries IPv6 first and then does not fall back to IPv4. A bug in versions of Opera prior to 10.5 caused such behaviour, which was obviously a big issue for Opera users trying to access dual-stack web sites with broken IPv6 connectivity.

The author has undertaken some informal tests at his own site, which shows how different combinations of browsers and operating systems behave in the event of IPv6 connections failing or when ICMP unreachables are received. On Linux/Firefox, web connections timeout after 20 seconds for 'no response', but immediately for unreachables. In contrast, Windows Vista/IE was 20 seconds regardless of unreachables being received. Any non-trivial delay will cause significant user frustration.

A more complete set of tests was run by Teemu Savolainen and reported at IETF80 [Savolainen2011]. Although the tests were only samples, they confirmed the results, also showing experiences across a much broader range of platforms, and that the problems with Vista/IE are repeated with Win 7/IE. It's thus clear that if major content providers enable IPv6 on World IPv6 Day, and end users for some reason try to access the content with broken IPv6 connectivity, they are likely to experience significant timeout issues.

This problem is probably the main reason that Google implemented a AAAA whitelisting system for its test sites. The sites had to demonstrate they had good IPv6 connectivity before being allowed into the test programme. The topic is discussed in [I-D.ietf-v6ops-v6-aaaa-whitelisting-implications]. For the sake of World IPv6 Day, it is expected that no such whitelisting is in place - that is, after all, the point of having a day dedicated to testing IPv6 in production.

An interesting suggestion to handle the problem is the 'happy eyeballs' approach described in [I-D.ietf-v6ops-happy-eyeballs]. This approach is now also being suggested for multiple interface systems, as per [I-D.chen-mif-happy-eyeballs-extension]. The happy eyeballs philosophy is to try both IPv4 and IPv6 together, and keep the first working connection up, remembering the result for future connection attempts. It may prefer IPv6 slightly in initial connections rather than trying connections exactly simultaneously.

It is an interesting approach, though some people are concerned about the additional connection load, or that this 'workaround' is simply masking underlying problems that should be fixed.

2.4. PMTU Discovery

IPv6 mandates that fragmentation is only undertaken by the sending node, and thus IPv6 requires working PMTU Discovery [RFC1981]. An existing RFC gives Recommendations for Filtering ICMPv6 Messages in Firewalls [RFC4890]; if this guidance is not followed, connectivity problems are likely to arise. Blindly filtering all ICMPv6 messages is not good practise. Filtering ICMP is a common practice in some IPv4 networks today. Adopting the same approach to ICMPv6 when deploying IPv6 networks will cause connectivity issues for users of the network filtering ICMPv6 and hosts trying to reach the filtered network. RFC 4890 is therefore an important document for IPv6 deployment engineers to read and it is similarly important to verify that IPv6 firewall deployments support appropriate configurations for ICMPv6 filtering.

The minimum MTU for IPv6 is 1280 bytes. Checking the MTU is an important step when connectivity issues arise. Where PMTUD is not working or not implemented, the using the minimum MTU is likely to resolve the problem, though not give optimal performance (the cause should still be investigated and resolved for longer term benefit). Tunnel broker services such as SixXS and HE set their MTUs to default to 1280, probably due to the varying conditions their customers may be in. However, it is preferable for enterprise networks to configure appropriate ICMPv6 filtering to allow PMTUD to operate and establish the most efficient MTUs for a link.

2.5. Rogue Router Advertisements

Within a site, hosts may use IPv6 Stateless Address Autoconfiguration (SLAAC) [RFC4862]. However, it is possible for accidental (or malicious) rogue RAs to cause connectivity issues, as described in the Rogue Router Advertisement Problem Statement [RFC6104].

A typical cause of rogue RAs is Windows ICS, which can present a rogue 6to4 router on its wireless interface. This will cause hosts to potentially autoconfigure two global IPv6 addresses and pick the wrong default router, with unpredictable results. As a (bad) example the author experienced a scenario where he had a rogue 6to4 RA, but because the rogue 6to4 was working he was able to access IPv6 networks outside his own network, but could not access most internal hosts inside his own network because he was unwittingly using 6to4 from outside into his own network, and thus being firewalled from those internal hosts.

In many cases, default address selection [RFC3484] (and [I-D.ietf-6man-rfc3484-revise]) would avoid such cases, because the address selection rules should prefer, or can be configured to prefer, native IPv6 over 6to4. However not all operating systems implement RFC 3484 yet, in particular MacOS X (though support may be appearing in Lion). Where rogue RAs cause broken IPv6 behaviour, the timeout issues discussed above may apply.

Adding ACLs to your switches to block ICMPv6 Type 134 packets on ports that do not have routers connected would also minimise the impact of rogue RAs. A more elegant solution is RA Guard [RFC6105], and another is use of SEcure Neighbour Discovery (SEND) [RFC3971]. However neither is widely implemented yet. Indeed, any reported operational experience of SEND in an enterprise network would be very welcome.

Finally, there is a tool called RAmond, available freely from http://ramond.sourceforge.net, that can be configured to detect and issue deprecating RAs against observed rogue RAs. This software is based on rafixd.

2.6. Tunnel performance

In scenarios where sites currently have manually configured tunnels to gain IPv6 connectivity, it may be the case that such encapsulation is performed by a router’s CPU, in which case unexpected high volumes of traffic may cause problems. Bear in mind that on World IPv6 Day, you may start using IPv6 by default for some high bandwidth applications that you had not used before, e.g. YouTube from Google. It may be prudent to estimate your load for such applications in advance, and test the capability of your tunnelling solution to handle that load.

2.7. AAAA record advertised but service not enabled

If enabling a service for World IPv6 Day, be aware of other existing services that may be running on the same system. If a server has multiple functions, all services should be IPv6 enabled before a AAAA record is entered into the DNS for services that may use that name.

A related consideration is to make sure that firewalls don’t just drop IPv6 packets to ports that are not in use. It’s better if the firewall or host sends an unreachable indication or a TCP RST to avoid a potential timeout. For example, if you add a AAAA record for your web server that also runs say FTP, where FTP is IPv4 only, either the firewall should have port 21 open or the firewall should be configured ta send a TCP RST. There are of course tradeoffs in enabling ICMP unreachables.

2.8. IPv6 Reverse DNS

Presence of IPv6 reverse DNS records is used by many systems as a security method. For example, many mail exchangers will only accept SMTP connections from IP addresses with a reverse DNS entry. It is thus important for such records to exist where, for example, a site is sending mail out over IPv6 transport. It is not necessarily the case that such connections will fall back to IPv4 if reverse records are not present.

3. Instrumentation

In this section we discuss potential instrumentation approaches that may be configured in advance of World IPv6 Day, and then retained longer term after the event. These are particularly useful if your site is turning on AAAA records for its production web presence (for example) and wants to get the best insight into how the systems performed and the nature of the end user experience.

These measurements should complement informal, subjective reports from users at participating sites. It is probably prudent to make at least your organisation’s IT staff aware of the ’at risk’ day, and actions they should take should they experience problems. It may also be desirable to undertake some form of user survey soon afterwards; whether you inform general users in advance is an issue for each site. The ARIN IPv6 wiki is a good source of such advice [ARINwiki].

3.1. IPv6 traffic levels

It should be possible to measure raw IPv6 traffic levels independently on dual-stack switch/router platforms, given implementations of appropriate MIBs. Sites should take steps to ensure they have the tools in place to be able to view the relative levels of IPv4 and IPv6 traffic over time.

Application level measurement is also desirable, because handling of choice (preference) of protocol used lies with the application if both A and AAAA records are returned. Sites should be aware that due to IPv6 Privacy Extensions [RFC4941] application logs may show more apparent different clients connecting, due to clients cycling the source IPv6 address they use over time.

The types of information gathered might for example include:

- IPv6 traffic volume, sources of IPv6 traffic by AS, types of IPv6 traffic (e.g. native, 6to4, Teredo, tunnelled);

- o IPv6 application mix, comparison with IPv4;
- o The number and type of IPv6 client connections.

3.2. Network flow records

Where available, sites should seek to generate and record network flow records for traffic, to maximise opportunities to analyse traffic patterns after the event, or in the case of reports of specific problems. Netflow v9 supports IPv6. Open source IPv6-capable Netflow collectors also exist, e.g. nfsen, from http://nfsen.sourceforge.net.

3.3. Client Web Access Success Rate

There have been some recent studies on the capabilities of web clients to access content on dual-stack servers by IPv4 or IPv6 in the presence of both A and AAAA records existing for a web domain.

One good example is that of [Anderson10], as reported at RIPE-61, where the author set up some application (web server) oriented tests for his newspaper content in Norway. The methodology was to add an invisible IFRAME to his site that would include IMG links randomly to 1x1 images that were served either via an IPv4-only target or a dual-stack target. Variation in the hit rates would imply IPv6 brokenness. By analysing the http metadata information could be gleaned on the cause of the brokenness. Results in Q4'2009 showed 0.2-0.3% brokenness, including the Opera bug mentioned above.

Recent figures published by Google suggest at most a 0.1% level of brokenness, indicating some improvement, but that level is still potentially 1 in 1000 users with a problem.

3.4. Tools to measure IPv6 brokenness

Sites may wish to make their own measurements of IPv6 brokenness rather than relying on third party reports. There are some openly available tools available that work along similar principles to the method proposed by Tore Anderson above.

The APNIC Labs test tool uses a combination of JavaScript and Google Analytics to measure various types of brokenness [APNIC]. Eric Vyncke's tool [Vyncke] measures a slightly smaller set of types of brokenness, but also looks very useful, with additional reports on the browser type for each failure. The author is currently using the latter tool, and plans to enable the APNIC measurement system shortly when other Analytics updates are applied locally.

3.5. IPv4 Performance Comparison

Where a dual-stack service is deployed, measuring the relative performance of both protocols is desirable. This may primarily be a measurement of throughput or delay, but may also include availability/uptime measurement. A site may choose to set up its own performance measuring framework, for example using open source bandwidth and throughput test tools. Participants in World IPv6 Day will be monitored from a broad range of locations and measurements will be available to show availability of AAAA records, reachability to http service, latency and availability over time.

3.6. User Tickets

It is possible a higher than usual user ticket rate for connectivity issues may be experienced. being able to categorise these cases for subsequent analysis is desirable.

3.7. Security monitoring

We mentioned RAmond above in the context of watching for rogue RAs. There is another useful package called NDPmon, also available freely from http://ndpmon.sourceforge.net, that can be configured to watch for certain types of IPv6 'abuse' on your local network. It may be interesting to run the tool to confirm whether any 'bad' traffic is observed within your network on World IPv6 Day.

4. IPv6-only testing

The long-term IPv6 deployment plan is IPv6-only networking, rather than dual-stack. It is not clear how quickly significant IPv6-only networks will emerge, but testing of approaches to IPv6-only operation is desirable as soon as possible. A draft by Jari Arkko and Ari Keranen describes some such experiences [I-D.arkko-ipv6-only-experience].

Some experience of NAT64 [RFC6146] has been described in [I-D.tan-v6ops-nat64-experiences], though this appears to have used only NAT-PT so far. An implementation of NAT64 is available at http://ecdysis.viagenie.ca. Operational experience of IVI is also desirable. An implementation of IVI is available at http://www.ivi2.org/IVI.

5. Conclusions

With the ISOC World IPv6 Day event due on June 8th 2011, this

document aims to help focus attention on both improving awareness and mitigations of common causes of IPv6 connectivity problems, and encouraging sites and organisations to introduce appropriate instrumentation into their networks so they can observe traffic behaviour appropriately.

This is still an early version of the text, and is thus a little drafty. All comments are very welcome towards a mature version in advance of June.

6. Security Considerations

There are no extra security consideration for this document.

7. IANA Considerations

There are no extra IANA consideration for this document.

8. Acknowledgments

To be added.

9. Informative References

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless

Address Autoconfiguration", RFC 4862, September 2007.

[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, May 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, February 2011.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 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.

[I-D.carpenter-v6ops-6to4-teredo-advisory]
Carpenter, B., "Advisory Guidelines for 6to4 Deployment", draft-carpenter-v6ops-6to4-teredo-advisory-03 (work in progress), March 2011.

[I-D.ietf-v6ops-happy-eyeballs]
Wing, D. and A. Yourtchenko, "Happy Eyeballs: Trending Towards Success with Dual-Stack Hosts", draft-ietf-v6ops-happy-eyeballs-02 (work in progress), May 2011.

[I-D.tan-v6ops-nat64-experiences]
Tan, J., Lin, J., and W. Li, "Experience from NAT64 applications", draft-tan-v6ops-nat64-experiences-00 (work in progress), March 2011.

[I-D.troan-v6ops-6to4-to-historic]
Troan, O., "Request to move Connection of IPv6 Domains via IPv4 Clouds (6to4) to Historic status", draft-troan-v6ops-6to4-to-historic-01 (work in progress), March 2011.

[I-D.ietf-v6ops-v6-aaaa-whitelisting-implications]
Livingood, J., "IPv6 AAAA DNS Whitelisting Implications", draft-ietf-v6ops-v6-aaaa-whitelisting-implications-05 (work in progress), May 2011.

[I-D.chen-mif-happy-eyeballs-extension]

Chen, G. and C. Williams, "Happy Eyeballs Extension for Multiple Interfaces", draft-chen-mif-happy-eyeballs-extension-01 (work in progress), March 2011.

[I-D.ietf-6man-rfc3484-revise]
Matsumoto, A., Kato, J., and T. Fujisaki, "Update to RFC 3484 Default Address Selection for IPv6", draft-ietf-6man-rfc3484-revise-02 (work in progress), March 2011.

[I-D.arkko-ipv6-only-experience]
Arkko, J. and A. Keranen, "Experiences from an IPv6-Only Network", draft-arkko-ipv6-only-experience-03 (work in progress), April 2011.

[APNIC] "IPv6 Capability Tracker", <http://labs.apnic.net/>.

[Vyncke] Vyncke, E., "Estimation of IPv6 brokenness", <http://test4.vyncke.org/testv6/>.

[ARINwiki]
"ARIN IPv6 Wiki", <http://getipv6.info/index.php/Customer_problems_that_could_occur>.

[testipv6]
"Test IPv6", <http://www.test-ipv6.com/>.

[ISOC] "World IPv6 Day", <http://isoc.org/wp/worldipv6day/>.

[Huston2011]
Huston, G., "Stacking it Up: Experimental Observations on the operation of Dual Stack Services", 2011, <http://www.ietf.org/proceedings/80/slides/v6ops-1.pdf>.

[Savolainen2011]
Savolainen, T., "Experiences of host behaviour in broken IPv6 networks", 2011, <http://www.ietf.org/proceedings/80/slides/v6ops-12.pdf>.

[ISOCsites]
"IPv6 Enabled Websites", <http://www.worldipv6day.org/ipv6-enabled-websites>.

[Anderson10]
Anderson, T., "Measuring and Combating IPv6 Brokenness", 2010, <http://ripe61.ripe.net/presentations/162-ripe61.pdf>.

Authors' Addresses

Tim Chown
University of Southampton
Highfield
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

Mat Ford
Internet Society
Geneva,
Switzerland

Email: ford@isoc.org

Stig Venaas
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: stig@cisco.com

v6ops
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

D. Wing
A. Yourtchenko
Cisco
March 14, 2011

Happy Eyeballs: Trending Towards Success with Dual-Stack Hosts
draft-ietf-v6ops-happy-eyeballs-01

Abstract

This document describes how a dual-stack client can determine the functioning path to a dual-stack server. This provides a seamless user experience during initial deployment of dual-stack networks and during outages of IPv4 or outages of IPv6.

Table of Contents

1. Introduction

In order to use HTTP successfully over IPv6, it is necessary that the user enjoys nearly identical performance as compared to IPv4. A combination of today's applications, IPv6 tunneling and IPv6 service providers, and some of today's content providers all cause the user experience to suffer (Section 3). For IPv6, a content provider may ensure a positive user experience by using a DNS white list of IPv6 service providers who peer directly with them, e.g. [whitelist]. However, this is not scalable to all service providers worldwide, nor is it scalable for other content providers to operate their own DNS white list.

Instead, this document suggests a mechanism for applications to quickly determine if IPv6 or IPv4 is the most optimal to connect to a server. The suggestions in this document provide a user experience which is superior to connecting to ordered IP addresses which is helpful during the IPv6/IPv4 transition with dual stack hosts.

This problem is described also in [RFC1671]: "The dual-stack code may get two addresses back from DNS; which does it use? During the many years of transition the Internet will contain black holes. For example, somewhere on the way from IPng host A to IPng host B there will sometimes (unpredictably) be IPv4-only routers which discard IPng packets. Also, the state of the DNS does not necessarily correspond to reality. A host for which DNS claims to know an IPng address may in fact not be running IPng at a particular moment; thus an IPng packet to that host will be discarded on delivery. Knowing that a host has both IPv4 and IPng addresses gives no information about black holes. A solution to this must be proposed and it must not depend on manually maintained information. (If this is not solved, the dual stack approach is no better than the packet translation approach.)"

Following the procedures in this document, once a certain address family is successful, the application trends towards preferring that address family. Thus, repeated use of the application DOES NOT cause repeated probes over both address families.

While the application recommendations in this document are described in the context of HTTP clients ("web browsers"), it is also useful and applicable to other interactive applications.

Code which implements some of the ideas described in this document has been made available [Perreault] [Andrews].

2. Notational Conventions

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. Problem Statement

As discussed in more detail in Section 3.1, it is important that the same URI and hostname be used for IPv4 and IPv6. Using separate namespaces causes namespace fragmentation and reduces the ability for users to share URIs and hostnames, and complicates printed material that includes the URI or hostname.

As discussed in more detail in Section 3.2, IPv6 connectivity is sometimes broken entirely or slower than native IPv4 connectivity.

3.1. URIs and hostnames

URIs are often used between users to exchange pointers to content -- such as on social networks, email, instant messaging, or other systems. Thus, production URIs and production hostnames containing references to IPv4 or IPv6 will only function if the other party is also using an application, OS, and a network that can access the URI or the hostname.

3.2. IPv6 connectivity

When IPv6 connectivity is impaired, today’s IPv6-capable web browsers incur many seconds of delay before falling back to IPv4. This harms the user’s experience with IPv6, which will slow the acceptance of IPv6, because IPv6 is frequently disabled in its entirety on the end systems to improve the user experience.

Reasons for such failure include no connection to the IPv6 Internet, broken 6to4 or Teredo tunnels, and broken IPv6 peering.

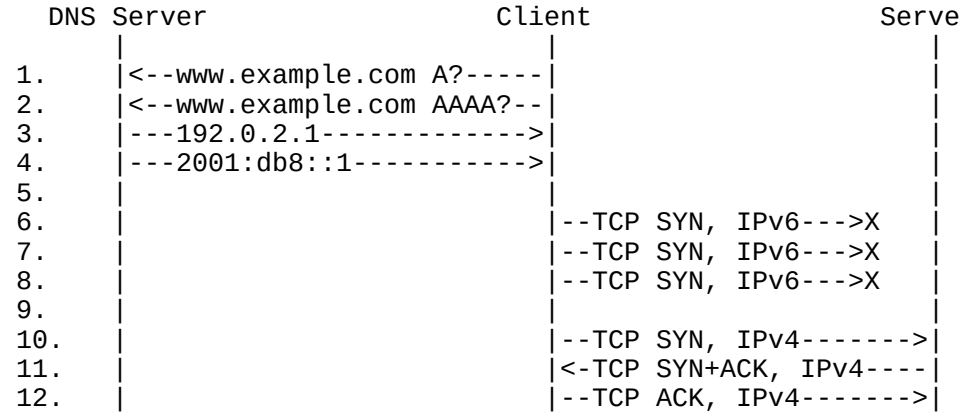


Figure 1: Existing behavior message flow

The client obtains the IPv4 and IPv6 records for the server (1-4). The client attempts to connect using IPv6 to the server, but the IPv6 path is broken (6-8), which consumes several seconds of time. Eventually, the client attempts to connect using IPv4 (10) which succeeds.

4. Client Recommendations

To provide fast connections for users, clients should make connections quickly over various technologies, automatically tune itself to avoid flooding the network with unnecessary connections (i.e., for technologies that have not made successful connections), and occasionally flush its self-tuning.

4.1. Dualstack behavior

If a TCP client supports IPv6 and IPv4 and is connected to IPv4 and IPv6 networks, it can perform the procedures described in this section.

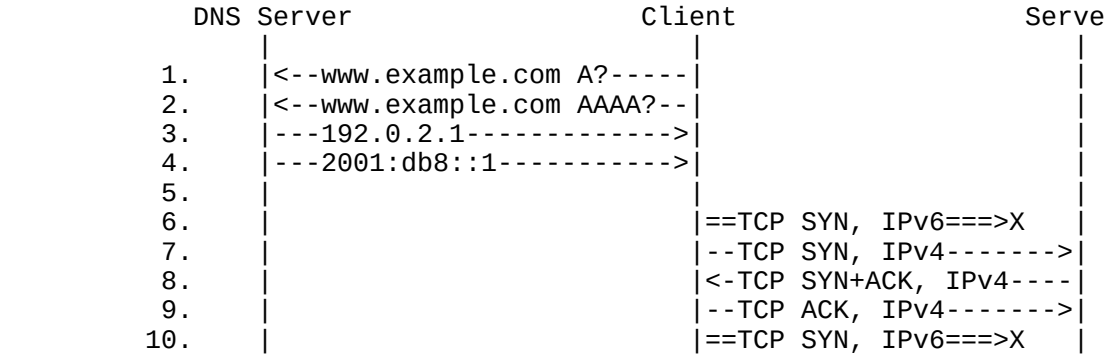


Figure 2: Happy Eyeballs flow 1, IPv6 broken

In the diagram above, the client sends two TCP SYNs at the same time over IPv6 (6) and IPv4 (7). In the diagram, the IPv6 path is broken but has little impact to the user because there is no long delay before using IPv4. The IPv6 path is retried until the application gives up (10).

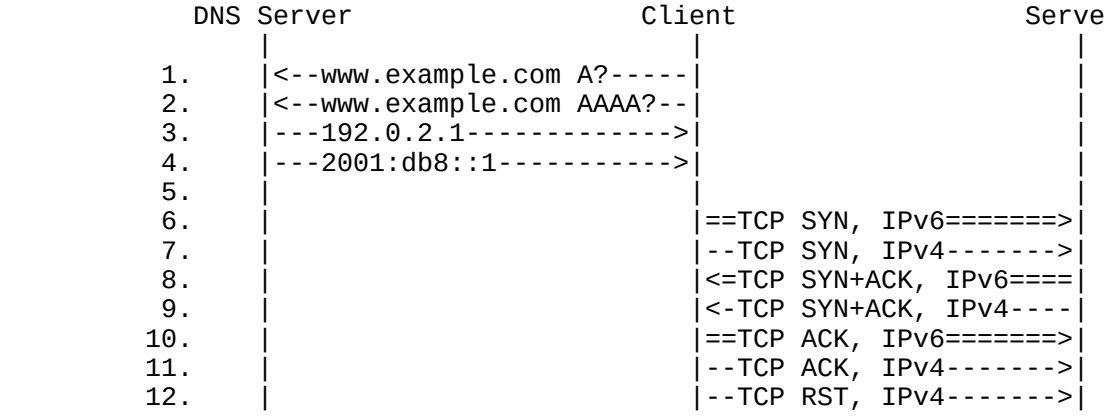


Figure 3: Happy Eyeballs flow 2, IPv6 working

The diagram above shows a case where both IPv6 and IPv4 are working, and IPv4 is abandoned (12).

4.2. Implementation details

4.2.1. Applications that use address records

This section details how to provide robust dual stack service for both IPv6 and IPv4, so that the user perceives very fast application response.

The TCP client application is configured with one application-wide value of P. A positive value indicates a preference for IPv6 and a negative value indicates a preference for IPv4. A value of 0 indicates equal weight, which means the A and AAAA queries and associated connection attempts will be sent as quickly as possible. The absolute value of P is the measure of a delay before initiating a DNS lookup and a connection attempt on the other address family. There are two P values maintained: one is application-wide and the other is specific per each destination (hostname and port).

The algorithm attempts to delay the DNS query until it expects that address family will be necessary; that is, if the preference is towards IPv6, then AAAA will be queried immediately and the A query will be delayed.

The TCP client application starts two concurrent execution flows (they will be referred to as "threads" but this reference does not imply the implementation detail of using the threading library, merely the property of mutual concurrency) in order to minimize the user-noticeable delay ("dead time") during the connection attempts:

thread 1: (IPv6)

* If P<0, wait for absolute value of p*10 milliseconds
* send DNS query for AAAA
* wait until DNS response is received
* Attempt to connect over IPv6 using TCP

thread 2: (IPv4)

* if P>0, wait for p*10 milliseconds
* send DNS query for A
* wait until DNS response is received
* Attempt to connect over IPv4 using TCP

The first thread that succeeds returns the completed connection to the parent code and aborts the other thread (Section 5.2).

After a connection is successful, we want to adjust the application-wide preference and the per-destination preference. The value of P is incremented (decremented) each time an IPv6 (IPv4) connection wins the race.. When a connection using the less-preferred address family

is successful, it indicates the wrong address family was used and the value of P is halved:

- If P>0 (indicating IPv6 is preferred over IPv4) and the first thread to finish was the IPv6 thread it indicates the IPv6 preference is correct and we need to re-enforce this by increasing the application-wide P value by 1. However, if the first thread to finish was the IPv4 thread it indicates an IPv6 connection problem occurred and we need to aggressively prefer IPv4 more by halving P and rounding towards 0.

- If P<0 (indicating IPv4 is preferred over IPv6) and the first thread to finish was the IPv4 thread it indicates the preference is correct and we need to re-enforce this gently by decreasing the application-wide P value by 1. However, if the first thread to finish was the IPv6 thread it indicates an IPv4 connection problem and we need to aggressively avoid IPv4 by halving P and rounding towards 0.

- If P=0 (indicating equal preference), P is incremented by one if the first thread to complete was the IPv6 thread, or decremented by one if the first thread to complete was the IPv4 thread.

After adjusting P, the resulting delay should never be larger than 4 seconds -- which is similar to the value used by many IPv6-capable TCP client applications to switch to an alternate A or AAAA record.

> Editor’s Note 01: Proof of concept tests on fast networks show that even smaller value (around 0.5 seconds) may be practical. More extensive testing would be useful to find the best upper boundary that still ensures a good user experience.
>
> Editor’s Note 02: A strict implementation of the above steps results in "P" being adjusted if there are no AAAA records or are no A records. This is undesirable. Thus, a future version of this specification is expected to recommend that "P" only be adjusted if there was both an A and AAAA record.

4.2.2. Applications that use the SRV records

For the purposes of this section, "client" is defined as the entity initiating the connection.

For protocols which support DNS SRV [RFC2782], the client performs the IN SRV query (e.g. IN SRV _xmpp-client._tcp.example.com) as normal. The client MUST perform the following steps:

1. Sort all SRV records according to priority (lowest priority first)

2. Process all of the SRV targets of the same priority with a weight greater than 0:

   A. Perform A/AAAA queries for each SRV target in parallel, as described in Section 4.2.1

   B. Connect to the IPv4/IPv6 addresses

   C. If at least one connection succeeds, stop processing SRV records

3. If there is no connection, process all of the SRV targets of the same priority with a weight of 0, as per steps 2.1 through 2.3 above

4. Repeat steps 2.1 through 2.3 for the next priority, until a connection is established or all SRV records have been exhausted

5. If there is still no connection, fallback to using the domain (e.g. example.com), following steps 2.1 through 2.3 above

It is RECOMMENDED, but not required, for the client to cache the winning connection's address information and reuse it on subsequent connections. If a significant network event occurs (e.g. network interface is activated/deactivated, IP address changes), the client MUST forget the cached address information and perform all of the steps from above. The definition of "significant network event" is intentionally vague.

5. Additional Considerations

This section discusses considerations and requirements that are common to new technology deployment.

5.1. Additional Network and Host Traffic

Additional network traffic and additional server load is created due to these recommendations and mitigated by application-wide and per-destination timer adjustments. The procedures described in this document retain a quality user experience while transitioning from IPv4-only to dual stack. The quality user experience benefits the user but to the detriment of the network and server that are serving the user.

5.2. Abandon Non-Winning Connections

It is RECOMMENDED that the non-winning connections be abandoned, even though they could be used to download content. This is because some web sites provide HTTP clients with cookies (after logging in) that incorporate the client's IP address, or use IP addresses to identify users. If some connections from the same HTTP client are arriving from different IP addresses, such HTTP applications will break. It's also important to abandon connections to avoid consuming server or middlebox (e.g., NAT) resources (file descriptors, memory, TCP control blocks) and avoid sending TCP or application-level keepalives on otherwise unused connections.

5.3. Flush or Expire Cache

Because every network has different characteristics (e.g., working or broken IPv6 connectivity) the IPv6/IPv4 preference value (P) SHOULD be reset to its default whenever the host is connected to a new network ([cx-osx], [cx-win]). However, in some instances the application and the host are unaware the network connectivity has changed so it is RECOMMENDED that per-destination values expire after 10 minutes of inactivity.

5.4. Determining Address Type

For some transitional technologies such as a dual-stack host, it is easy for the application to recognize the native IPv6 address (learned via a AAAA query) and the native IPv4 address (learned via an A query). For other transitional technologies [RFC2766] it is impossible for the host to differentiate a transitional technology IPv6 address from a native IPv6 address (see Section 4.1 of [RFC4966]). Replacement transitional technologies are attempting to bridge this gap. It is necessary for applications to distinguish between native and transitional addresses in order to provide the most seamless user experience.

Application awareness of transitional technologies, if implemented, SHOULD provide a facility to give the preference only to native IPv6 addresses.

5.5. Debugging and Troubleshooting

This mechanism is aimed at ensuring a reliable user experience regardless of connectivity problems affecting any single transport. However, this naturally means that applications employing these techniques are by default less useful for diagnosing issues with any particular transport. To assist in that regard, the applications implementing the proposal in this document SHOULD also provide a

mechanism to revert the behavior to that of a default provided by the operating system - the [RFC3484].

[[[To be discussed.

Some sites may wish to be informed when the the hosts adjust their "P" value, in order to troubleshoot the underlying cause. To help these sites, a strawman proposal is to send a syslog message or other notification to an address that may be configured by a site administrator in a centralized fashion. (The exact method TBD - DHCP option, domain name, etc.) This syslog message should be sent only first N times that the host expects to prefer IPv6 but has to use IPv4. I.e. the first N times it decreases the value of P. N - TBD.

]]]

5.6. DNS Behavior

Unique to DNS AAAA queries are the problems described in [RFC4074] which, if they still persist, require applications to perform an A query before the AAAA query.

[[Editor's Note 03: It is believed these defective DNS servers have long since been upgraded. If so, we can remove this section.]]

5.7. Middlebox Issues

Some devices are known to exhibit what amounts to a bug, when the A and AAAA requests are sent back-to-back over the same 4-tuple, and drop one of the requests or replies [DNS-middlebox]. However, in some cases fixing this behaviour may not be possible either due to the architectural limitations or due to the administrative constraints (location of the faulty device is unknown to the end hosts or not controlled by the end hosts). The algorithm described in this draft, in the case of this erroneous behaviour will eventually pace the queries such that this issue is will be avoided. The algorithm described in this draft also avoids calling the operating system's getaddrinfo() with "any", which should prevent the operating system from sending the A and AAAA queries on the same port.

For the large part, these issues are believed to be fixed, in which case the getaddrinfo() with AF_UNSPEC as the address family in its hints.

5.8. Multiple Interfaces

Interaction of the suggestions in this document with multiple interfaces, and interaction with the MIF working group, is for further study ([I-D.chen-mif-happy-eyeballs-extension] is devoted to this).

6. Content Provider Recommendations

Content providers SHOULD provide both AAAA and A records for servers using the same DNS name for both IPv4 and IPv6.

7. Security Considerations

[[Placeholder.]]

See Section 5.2.

8. Acknowledgements

The mechanism described in this paper was inspired by Stuart Cheshire's discussion at the IAB Plenary at IETF72, the author's understanding of Safari's operation with SRV records, Interactive Connectivity Establishment (ICE [RFC5245]), and the current IPv4/IPv6 behavior of SMTP mail transfer agents.

Thanks to Fred Baker, Jeff Kinzli, Christian Kuhtz, and Iljitsch van Beijnum for fostering the creation of this document.

Thanks to Scott Brim, Rick Jones, Stig Venaas, Erik Kline, Bjoern Zeeb, Matt Miller for providing feedback on the document.

Thanks to Javier Ubillos, Simon Perreault and Mark Andrews for the active feedback and the experimental work on the independent practical implementations that they created.

Also the authors would like to thank the following individuals who participated in various email discussions on this topic: Mohacsi Janos, Pekka Savola, Ted Lemon, Carlos Martinez-Cagnazzo, Simon Perreault, Jack Bates, Jeroen Massar, Fred Baker, Javier Ubillos, Teemu Savolainen, Scott Brim, Erik Kline, Cameron Byrne, Daniel Roesen, Guillaume Leclanche, Cameron Byrne, Mark Smith, Gert Doering, Martin Millnert, Tim Durack, Matthew Palmer.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

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

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

10.2. Informational References

[Andrews] Andrews, M., "How to connect to a multi-homed server over TCP", January 2011, <http://www.isc.org/community/blog/201101/how-to-connect-to-a-multi-h omed-server-over-tcp>.

[DNS-middlebox] Various, "DNS middlebox behavior with multiple queries over same source port", June 2009, <https://bugzilla.redhat.com/show_bug.cgi?id=505105>.

[I-D.chen-mif-happy-eyeballs-extension] Chen, G., "Happy Eyeballs Extension for Multiple Interfaces", draft-chen-mif-happy-eyeballs-extension-00 (work in progress), March 2011.

[Perreault] Perreault, S., "Happy Eyeballs in Erlang", February 2011, <http://www.viagenie.ca/news/index.html#happy_eyeballs_erlang>.

[RFC1671] Carpenter, B., "IPng White Paper on Transition and Other Considerations", RFC 1671, August 1994.

[RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.

[RFC4074] Morishita, Y. and T. Jinmei, "Common Misbehavior Against

DNS Queries for IPv6 Addresses", RFC 4074, May 2005.

[RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.

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

[cx-osx] Adium, "AIHostReachabilityMonitor", June 2009, <https://bugzilla.redhat.com/show_bug.cgi?id=505105>.

[cx-win] Microsoft, "NetworkChange.NetworkAvailabilityChanged Event", June 2009, <http://msdn.microsoft.com/en-us/library/system.net.networkinformation.networkchange.networkavailabilitychanged.aspx>.

[whitelist] Google, "Google IPv6 DNS Whitelist", January 2009, <http://www.google.com/intl/en/ipv6>.

Authors' Addresses

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

Email: dwing@cisco.com

Andrew Yourtchenko
Cisco Systems, Inc.
De Kleetlaan, 7
San Jose, Diegem B-1831
Belgium

Email: ayourtch@cisco.com

v6ops
Internet-Draft
Intended status: Standards Track
Expires: June 22, 2012

D. Wing
A. Yourtchenko
Cisco

December 20, 2011

Happy Eyeballs: Success with Dual-Stack Hosts
draft-ietf-v6ops-happy-eyeballs-07

Abstract

When a server's IPv4 path and protocol is working but the server's IPv6 path and protocol are not working, a dual-stack client application experiences significant connection delay compared to an IPv4-only client. This is undesirable because it causes the dual-stack client to have a worse user experience. This document specifies requirements for algorithms that reduce this user-visible delay, and provides an algorithm.

Status of this Memo

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

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

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

This Internet-Draft will expire on June 22, 2012.

Copyright Notice

Table of Contents

1. Introduction

In order to use applications over IPv6, it is necessary that users enjoy nearly identical performance as compared to IPv4. A combination of today's applications, IPv6 tunneling, IPv6 service providers, and some of today's content providers all cause the user experience to suffer (Section 3). For IPv6, a content provider may ensure a positive user experience by using a DNS white list of IPv6 service providers who peer directly with them (e.g., [whitelist]). However, this does not scale well (to the number of DNS servers worldwide or the number of content providers worldwide), and does not react to intermittent network path outages.

Instead, applications reduce connection setup delays themselves, by more aggressively making connections on IPv6 and IPv4. There are a variety of algorithms that can be envisioned. This document specifies requirements for any such algorithm, with the goals that the network and servers are not inordinately harmed with a simple doubling of traffic on IPv6 and IPv4, and the host's address preference is honored (e.g., [RFC3484]).

1.1. Additional Network and Host Traffic

Additional network traffic and additional server load is created due to the recommendations in this document, especially when connections to the preferred address family (usually IPv6) are not completing quickly.

The procedures described in this document retain a quality user experience while transitioning from IPv4-only to dual stack, while still giving IPv6 a slight preference over IPv4 (in order to remove load from IPv4 networks, most importantly to reduce the load on IPv4 network address translators). The improvement in the user experience benefits the user to only a small detriment of the network, DNS server, and server that are serving the user.

2. Notational Conventions

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. Problem Statement

The basis of the IPv6/IPv4 selection problem was first described in 1994 in [RFC1671],

> "The dual-stack code may get two addresses back from DNS; which does it use? During the many years of transition the Internet will contain black holes. For example, somewhere on the way from IPng host A to IPng host B there will sometimes (unpredictably) be IPv4-only routers which discard IPng packets. Also, the state of the DNS does not necessarily correspond to reality. A host for which DNS claims to know an IPng address may in fact not be running IPng at a particular moment; thus an IPng packet to that host will be discarded on delivery. Knowing that a host has both IPv4 and IPng addresses gives no information about black holes. A solution to this must be proposed and it must not depend on manually maintained information. (If this is not solved, the dual stack approach is no better than the packet translation approach.)"

As discussed in more detail in Section 3.1, it is important that the same hostname be used for IPv4 and IPv6.

As discussed in more detail in Section 3.2, IPv6 connectivity is broken to specific prefixes or specific hosts, or slower than native IPv4 connectivity.

The mechanism described in this document is directly applicable to connection-oriented transports (e.g., TCP, SCTP), which is the scope of this document. For connectionless transport protocols (e.g., UDP), a similar mechanism can be used if the application has request/response semantics (e.g., as done by ICE to select a working IPv6 or IPv4 media path [RFC6157]).

3.1. Hostnames

Hostnames are often used between users to exchange pointers to content -- such as on social networks, email, instant messaging, or other systems. Using separate namespaces (e.g., "ipv6.example.com") which are only accessible with certain client technology (e.g., an IPv6 client) and dependencies (e.g., a working IPv6 path) causes namespace fragmentation and reduces the ability for users to share hostnames. It also complicates printed material that includes the hostname.

The algorithm described in this document allows production hostnames to avoid these problematic references to IPv4 or IPv6.

3.2. Delay When IPv6 is not Accessible

When IPv6 connectivity is impaired, today’s IPv6-capable applications (e.g., web browsers, email clients, instant messaging clients) incur many seconds of delay before falling back to IPv4. This delays

overall application operation, including harming the user's experience with IPv6, which will slow the acceptance of IPv6, because IPv6 is frequently disabled in its entirety on the end systems to improve the user experience.

Reasons for such failure include no connection to the IPv6 Internet, broken 6to4 or Teredo tunnels, and broken IPv6 peering. The following diagram shows this behavior.

The algorithm described in this document allows clients to connect to servers without significant delay, even if a path or the server is slow or down.

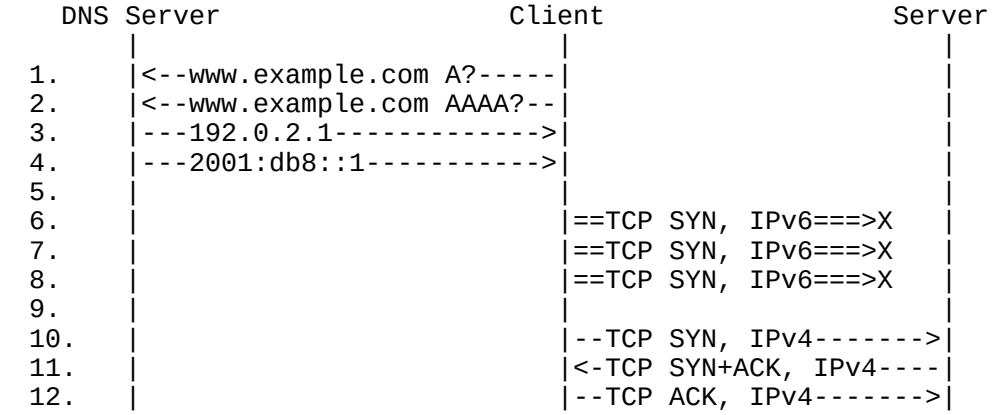


Figure 1: Existing behavior message flow

The client obtains the IPv4 and IPv6 records for the server (1-4). The client attempts to connect using IPv6 to the server, but the IPv6 path is broken (6-8), which consumes several seconds of time. Eventually, the client attempts to connect using IPv4 (10) which succeeds.

Delays experienced by users of various browser and operating system combinations have been studied [Experiences].

4. Algorithm Requirements

A Happy Eyeballs algorithm has two primary goals:

1. Provides fast connection for users, by quickly attempting to connect using IPv6 and (if that connection attempt is not quickly successful) to connect using IPv4.

2. Avoids thrashing the network, by not (always) making simultaneous connection attempts on both IPv6 and IPv4.

The basic idea is depicted in the following diagram:

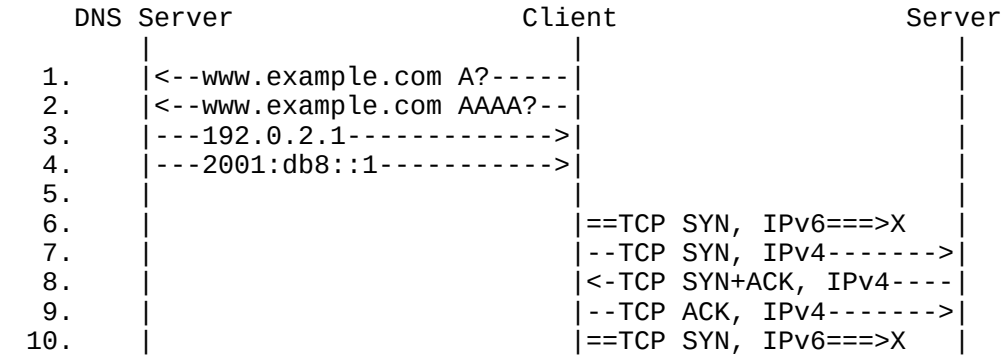


Figure 2: Happy Eyeballs flow 1, IPv6 broken

In the diagram above, the client sends two TCP SYNs at the same time over IPv6 (6) and IPv4 (7). In the diagram, the IPv6 path is broken but has little impact to the user because there is no long delay before using IPv4. The IPv6 path is retried until the application gives up (10).

After performing the above procedure, the client learns whether connections to the host's IPv6 or IPv4 address were successful. The client MUST cache information regarding the outcome of each connection attempt and uses that information to avoid thrashing the network with subsequent attempts. For example, in the example above, the cache indicates that the IPv6 connection attempt failed, and therefore the system will prefer IPv4 instead. Cache entries should be flushed when their age exceeds a system defined maximum on the order of ten minutes.

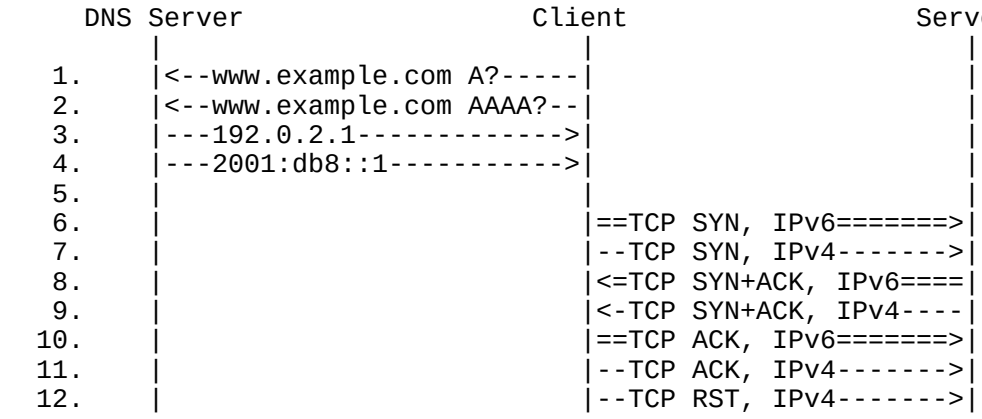


Figure 3: Happy Eyeballs flow 2, IPv6 working

The diagram above shows a case where both IPv6 and IPv4 are working, and IPv4 is abandoned (12).

Any Happy Eyeballs algorithm will persist in products for as long as the client host is dual-stacked, which will persist as long as there are IPv4-only servers on the Internet -- the so-called "long tail". Over time, as most content is available via IPv6, the amount of IPv4 traffic will decrease. This means that the IPv4 infrastructure will, over time, be sized to accommodate that decreased (and decreasing) amount of traffic. It is critical that a Happy Eyeballs algorithm not cause a surge of unnecessary traffic on that IPv4 infrastructure. To meet that goal, compliant Happy Eyeballs algorithms must adhere to the requirements in this section.

4.1. Delay IPv4

The transition to IPv6 is likely to produce a mix of different hosts within a subnetwork -- hosts that are IPv4-only, hosts that are IPv6-only (e.g., sensors), and dual-stack. This mix of hosts will exist both within an administrative domain (a single home, enterprise, hotel, or coffee shop) and between administrative domains. For example, a single home might have an IPv4-only television in one room and a dual-stack television in another room. As another example, another subscriber might have hosts that are all capable of dual-stack operation.

Due to IPv4 exhaustion, it is likely that a subscriber’s hosts (both IPv4-only hosts and dual-stack hosts) will be sharing an IPv4 address with other subscribers. The dual-stack hosts have an advantage: they can utilize IPv6 or IPv4, which means it can utilize the technique described in this document. The IPv4-only hosts have a

disadvantage: they can only utilize IPv4. If all hosts (dual-stack and IPv4-only) are using IPv4, there is additional contention for the shared IPv4 address. The IPv4-only hosts cannot avoid that contention (as they can only use IPv4) while the dual-stack hosts can avoid that contention by using IPv6.

As dual-stack hosts proliferate and content becomes available over IPv6, there will be proportionally less IPv4 traffic. This is true especially for dual-stack hosts that do not implement Happy Eyeballs, because those dual-stack hosts have a very strong preference to use IPv6 (with timeouts in the tens of seconds before they will attempt to use IPv4).

When deploying IPv6, both content providers and Internet Service Providers (who supply IPv4 address sharing mechanisms such as Carrier Grade NAT (CGN)) will want to reduce their investment in IPv4 equipment -- load balancers, peering links, and address sharing devices. If a Happy Eyeballs implementation treats IPv6 and IPv4 equally by connecting to whichever address family is fastest, it will contribute to load on IPv4. This load impacts IPv4-only devices (by increasing contention of IPv4 address sharing and increasing load on IPv4 load balancers). Because of this, ISPs and content providers will find it impossible to reduce their investment in IPv4 equipment. This means that costs to migrate to IPv6 are increased, because the investment in IPv4 cannot be reduced. Furthermore, using only a metric that measures connection speed ignores the value of IPv6 over IPv4 address sharing, such as shared penalty boxes and geo-location [RFC6269].

Thus, to avoid harming IPv4-only hosts which can only utilize IPv4, implementations MUST prefer the first IP address family returned by the host's address preference policy, unless implementing a stateful algorithm described in Section 4.2. This usually means giving preference to IPv6 over IPv4, although that preference can be over-ridden by user configuration or by network configuration [I-D.ietf-6man-addr-select-opt]. If the host's policy is unknown or not attainable, implementations MUST prefer IPv6 over IPv4.

4.2. Stateful Behavior when IPv6 Fails

Some Happy Eyeballs algorithms are stateful -- that is, the algorithm will remember that IPv6 always fails, or that IPv6 to certain prefixes always fails, and so on. This section describes such algorithms. Stateless algorithms, which do not remember the success/failure of previous connections, are not discussed in this section.

After making a connection attempt on the preferred address family (e.g., IPv6), and failing to establish a connection within a certain

time period (see Section 5.5), a Happy Eyeballs implementation will decide to initiate a second connection attempt using the same address family or the other address family.

Such an implementation MAY make subsequent connection attempts (to the same host or to other hosts) on the successful address family (e.g., IPv4). So long as new connections are being attempted by the host, such an implementation MUST occasionally make connection attempts using the host's preferred address family, as it may have become functional again, and it SHOULD do so every 10 minutes. The 10 minute delay before re-trying a failed address family avoids the simple doubling of connection attempts on both IPv6 and IPv4. Implementation note: this can be achieved by flushing Happy Eyeballs state every every 10 minutes, which does not significantly harm the application's subsequent connection setup time. If connections using the preferred address family are again successful, the preferred address family SHOULD be used for subsequent connections. Because this implementation is stateful, it MAY track connection success (or failure) based on IPv6 or IPv4 prefix (e.g., connections to the same prefix assigned to the interface are successful whereas connections to other prefixes are failing).

4.3. Reset on Network (re-)Initialization

Because every network has different characteristics (e.g., working or broken IPv6 or IPv4 connectivity), a Happy Eyeballs algorithm SHOULD re-initialize when the interface is connected to a new network. Interfaces can determine network (re-)initialization by a variety of mechanisms (e.g., DNAv4 [RFC4436], DNAv6 [RFC6059]).

If the client application is a web browser, see also Section 5.6.

4.4. Abandon Non-Winning Connections

It is RECOMMENDED that the non-winning connections be abandoned, even though they could -- in some cases -- be put to reasonable use.

Justification: This reduces the load on the server (file descriptors, TCP control blocks), stateful middleboxes (NAT and firewalls) and, if the abandoned connection is IPv4, reduces IPv4 address sharing contention.

HTTP: The design of some sites can break because of HTTP cookies that incorporate the client's IP address and require all connections be from the same IP address. If some connections from the same client are arriving from different IP addresses (or worse, different IP address families), such applications will break. Additionally for HTTP, using the non-winning connection

can interfere with the browser's Same Origin Policy (see Section 5.6).

5. Additional Considerations

This section discusses considerations related to Happy Eyeballs.

5.1. Determining Address Type

For some transitional technologies such as a dual-stack host, it is easy for the application to recognize the native IPv6 address (learned via a AAAA query) and the native IPv4 address (learned via an A query). While IPv6/IPv4 translation makes that difficult, IPv6/IPv4 translators do not need to be deployed on networks with dual stack clients, because dual stack clients can use their native IP address family.

5.2. Debugging and Troubleshooting

This mechanism is aimed at ensuring a reliable user experience regardless of connectivity problems affecting any single transport. However, this naturally means that applications employing these techniques are by default less useful for diagnosing issues with a particular address family. To assist in that regard, the implementations MAY also provide a mechanism to disable their Happy Eyeballs behavior via a user setting, and to provide data useful for debugging (e.g., a log or way to review current preferences).

5.3. Three or More Interfaces

A dual-stack host normally has two logical interfaces: an IPv6 interface and an IPv4 interface. However, a dual-stack host might have more than two logical interfaces because of a VPN (where a third interface is the tunnel address, often assigned by the remote corporate network) or because of multiple physical interfaces such as wired and wireless Ethernet, because the host belongs to multiple VLANs, or other reasons. The interaction of Happy Eyeballs with more than two logical interfaces is for further study.

5.4. A and AAAA Resource Records

It is possible that an DNS query for an A or AAAA resource record will return more than one A or AAAA address. When this occurs, it is RECOMMENDED that a Happy Eyeballs implementation order the responses following the host's address preference policy and then try the first address. If that fails after a certain time (see Section 5.5), the next address SHOULD be the IPv4 address.

If that fails to connect after a certain time (see Section 5.5), a Happy Eyeballs implementation SHOULD try the other addresses returned; the order of these connection attempts is not important.

On the Internet today, servers commonly have multiple A records to provide load balancing across their servers. This same technique would be useful for AAAA records, as well. However, if multiple AAAA records are returned to a non-Happy Eyeballs client that has broken IPv6 connectivity, it will further increase the delay to fall back to IPv4. Thus, web site operators with native IPv6 connectivity SHOULD NOT offer multiple AAAA records. If Happy Eyeballs is widely deployed in the future, this recommendation might be revisited.

5.5. Connection time out

The primary purpose of Happy Eyeballs is to reduce the wait time for a dual stack connection to complete, especially when the IPv6 path is broken and IPv6 is preferred. Aggressive time outs (on the order of tens of milliseconds) achieve this goal, but at the cost of network traffic. This network traffic may be billable on certain networks, will create state on some middleboxes (e.g., firewalls, IDS, NAT), and will consume ports if IPv4 addresses are shared. For these reasons, it is RECOMMENDED that connection attempts be paced to give connections a chance to complete. It is RECOMMENDED that connections attempts be paced 150-250ms apart, to balance human factors against network load. Stateful algorithms are expected to be more aggressive (that is, make connection attempts closer together), as stateful algorithms maintain an estimate of the expected connection completion time.

5.6. Interaction with Same Origin Policy

Web browsers implement a Same Origin Policy [RFC6454] which causes subsequent connections to the same hostname to go to the same IPv4 (or IPv6) address as the previous successful connection. This is done to prevent certain types of attacks.

The same-origin policy harms user-visible responsiveness if a new connection fails (e.g., due to a transient event such as router failure or load balancer failure). While it is tempting to use Happy Eyeballs to maintain responsiveness, web browsers MUST NOT change their Same Origin Policy because of Happy Eyeballs, as that would create an additional security exposure.

5.7. Implementation Strategies

The simplest venue for implementation of Happy Eyeballs is within the application itself. The algorithm specified in this document is

relatively simple to implement, and would require no specific support from the operating system beyond the commonly-available APIs that provide transport service. It could also be added to applications by way of a specific Happy Eyeballs API, replacing or augmenting the transport service APIs.

To improve IPv6 connectivity experience for legacy applications (e.g., applications which simply rely on the operating system's address preference order), operating systems may consider more sophisticated approaches. These can include changing default address selection sorting ([RFC3484]) based on configuration received from the network, or observing connection failures to IPv6 and IPV4 destinations.

6. Example Algorithm

What follows is the algorithm implemented in Google Chrome and Mozilla Firefox.

1. Call getaddinfo(), which returns a list of IP addresses sorted by the host's address preference policy.

2. Initiate a connection attempt with the first address in that list (e.g., IPv6).

3. If that connection does not complete within a short period of time (Firefox and Chrome use 300ms), initiate a connection attempt with the first address belonging to the other address family (e.g., IPv4)

4. The first connection that is established is used. The other connection is discarded.

If an algorithm were to cache connection success/failure, the caching would occur after step 4 determined which connection was successful.

Other example algorithms include [Perreault] and [Andrews].

7. Security Considerations

See Section 4.4 and Section 5.6.

8. Acknowledgements

The mechanism described in this paper was inspired by Stuart

Cheshire's discussion at the IAB Plenary at IETF72, the author's understanding of Safari's operation with SRV records, Interactive Connectivity Establishment (ICE [RFC5245]), the current IPv4/IPv6 behavior of SMTP mail transfer agents, and the implementation of Happy Eyeballs in Google Chrome and Mozilla Firefox.

Thanks to Fred Baker, Jeff Kinzli, Christian Kuhtz, and Iljitsch van Beijnum for fostering the creation of this document.

Thanks to Scott Brim, Rick Jones, Stig Venaas, Erik Kline, Bjoern Zeeb, Matt Miller, Dave Thaler, Dmitry Anipko, Brian Carpenter, and David Crocker for their feedback.

Thanks to Javier Ubillos, Simon Perreault and Mark Andrews for the active feedback and the experimental work on the independent practical implementations that they created.

Also the authors would like to thank the following individuals who participated in various email discussions on this topic: Mohacsi Janos, Pekka Savola, Ted Lemon, Carlos Martinez-Cagnazzo, Simon Perreault, Jack Bates, Jeroen Massar, Fred Baker, Javier Ubillos, Teemu Savolainen, Scott Brim, Erik Kline, Cameron Byrne, Daniel Roesen, Guillaume Leclanche, Mark Smith, Gert Doering, Martin Millnert, Tim Durack, Matthew Palmer.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

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

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

10.2. Informational References

[Andrews] Andrews, M., "How to connect to a multi-homed server over TCP", January 2011, <http://www.isc.org/community/blog/201101/how-to-connect-to-a-multi-h omed-server-over-tcp>.

[Experiences]

Savolainen, T., Miettinen, N., Veikkolainen, S., Chown, T., and J. Morse, "Experiences of host behavior in broken IPv6 networks", March 2011, <http://www.ietf.org/proceedings/80/slides/v6ops-12.pdf>.

[I-D.ietf-6man-addr-select-opt] Matsumoto, A., Fujisaki, T., Kato, J., and T. Chown, "Distributing Address Selection Policy using DHCPv6", draft-ietf-6man-addr-select-opt-01 (work in progress), June 2011.

[Perreault] Perreault, S., "Happy Eyeballs in Erlang", February 2011, <http://www.viagenie.ca/news/index.html#happy_eyeballs_erlang>.

[RFC1671] Carpenter, B., "IPng White Paper on Transition and Other Considerations", RFC 1671, August 1994.

[RFC4436] Aboba, B., Carlson, J., and S. Cheshire, "Detecting Network Attachment in IPv4 (DNAv4)", RFC 4436, March 2006.

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

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, November 2010.

[RFC6157] Camarillo, G., El Malki, K., and V. Gurbani, "IPv6 Transition in the Session Initiation Protocol (SIP)", RFC 6157, April 2011.

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

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, December 2011.

[whitelist] Google, "Google IPv6 DNS Whitelist", January 2009, <http://www.google.com/intl/en/ipv6>.

Appendix A. Changes

[RFC Editor: Please remove this section prior to publication as an RFC.]

A.1. changes from -06 to -07

- Changed "xmpp clients" to "instant messaging clients".
- For debugging/troubleshooting, providing a log of activity or a way to see current settings is useful.
- tweaked abstract
- "URIs and hostnames" -> "hostnames"
- tweaked text on caching
- interfaces (not hosts) notice when they are connected to a new network.
- encourage implementations to provide log or other way to view Happy Eyeballs settings.
- detailed that implementation can be in OS or in application.
- 150-250ms is for human factors

A.2. changes from -05 to -06

- Added paragraph describing current AAAA practice on the Internet (one AAAA record) due to non-Happy Eyeballs implementations, per opsdir review.
- fixed "=" in Figure 1.
- Removed text discussing A6. A6 is being deprecated in another document, and querying A6 is not a significant operational problem on the Internet.

A.3. changes from -04 to -05

- Updated citations.

A.4. changes from -03 to -04

- Make RFC3363 a non-normative reference.

A.5. changes from -03 to -04

- Better explained why IPv6 needs to be preferred
- Don’t query A6.

A.6. changes from -02 to -03

- Re-casted this specification as a list of requirements for a compliant algorithm, rather than trying to dictate a One True algorithm.

A.7. changes from -01 to -02

- Now honors host’s address preference (RFC3484 and friends)
- No longer requires thread-safe DNS library. It uses getaddrinfo()
- No longer describes threading.
- IPv6 is given a 200ms head start (Initial Headstart variable).
- If the IPv6 and IPv4 connection attempts were made at nearly the same time, wait Tolerance Interval milliseconds for both to complete before deciding which one wins.
- Renamed "global P" to "Smoothed P", and better described how it is calculated.
- introduced the exception cache. This contains the set of networks that only work with IPv4 (or only with IPv6), so that subsequent connection attempts use that address family without them causing serious affect to Smoothed P.
- encourages that every 10 minutes the exception cache and Smoothed P be reset. This allows IPv6 to be attempted again, so we don’t get ’stuck’ on IPv4.
- If we didn’t get both A and AAAA, abandon all Happy Eyeballs processing (thanks to Simon Perreault).
- added discussion of Same Origin Policy

- Removed discussion of NAT-PT and address learning; those are only used with IPv6-only hosts whereas this document is about dual-stack hosts contacting dual-stack servers.

A.8. changes from -00 to -01

- added SRV section (thanks to Matt Miller)

Authors' Addresses

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

Email: dwing@cisco.com

Andrew Yourtchenko
Cisco Systems, Inc.
De Kleetlaan, 7
Diegem B-1831
Belgium

Email: ayourtch@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 6, 2011

H. Singh
W. Beebee
Cisco Systems, Inc.
C. Donley
CableLabs
B. Stark
AT&T
O. Troan, Ed.
Cisco Systems, Inc.

March 5, 2011

Advanced Requirements for IPv6 Customer Edge Routers
draft-ietf-v6ops-ipv6-cpe-router-bis-00

Abstract

This document continues the work undertaken by the IPv6 CE Router Phase I work in the IETF v6ops Working Group. Advanced requirements or Phase II work is covered in this document.

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 September 6, 2011.

Copyright Notice

Table of Contents

1. Introduction

This document defines Advanced IPv6 features for a residential or small office router referred to as an IPv6 CE router. Typically these routers also support IPv4. The IPv6 End-user Network Architecture for such a router is described in [I-D.ietf-v6ops-ipv6-cpe-router]. This version of the document includes the requirements for Advanced features.

1.1. Requirements Language

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

2. Terminology

End-user Network	one or more links attached to the IPv6 CE router that connect IPv6 hosts.
IPv6 Customer Edge router	a node intended for home or small office use which forwards IPv6 packets not explicitly addressed to itself. The IPv6 CE router connects the end-user network to a service provider network.
IPv6 host	any device implementing an IPv6 stack receiving IPv6 connectivity through the IPv6 CE router
LAN interface	an IPv6 CE router’s attachment to a link in the end-user network. Examples are Ethernets (simple or bridged), 802.11 wireless or other LAN technologies. An IPv6 CE router may have one or more network layer LAN Interfaces.
Service Provider	an entity that provides access to the Internet. In this document, a Service Provider specifically offers Internet access using IPv6, and may also offer IPv4 Internet access. The Service Provider can provide such access over a variety of different transport methods such as DSL, cable, wireless, and others.

WAN interface	an IPv6 CE router's attachment to a link used to provide connectivity to the Service Provider network; example link technologies include Ethernets (simple or bridged), PPP links, Frame Relay, or ATM networks as well as Internet-layer (or higher-layer) "tunnels", such as tunnels over IPv4 or IPv6 itself.

3. Conceptual Configuration Variables

The CE Router maintains such a list of conceptual optional configuration variables.

1. Enable an IGP on the LAN.

4. Architecture

This document extends the architecture described in [I-D.ietf-v6ops-ipv6-cpe-router] to cover a strictly larger set of operational scenarios. In particular, QoS, multicast, DNS, routed network in the home, transition technologies, and conceptual configuration variables. This document also extends the model described in [I-D.ietf-v6ops-ipv6-cpe-router] to a two router topology where the two routers are connected back-to-back (the LAN of one router is connected to the WAN of the other router). This topology is depicted below:

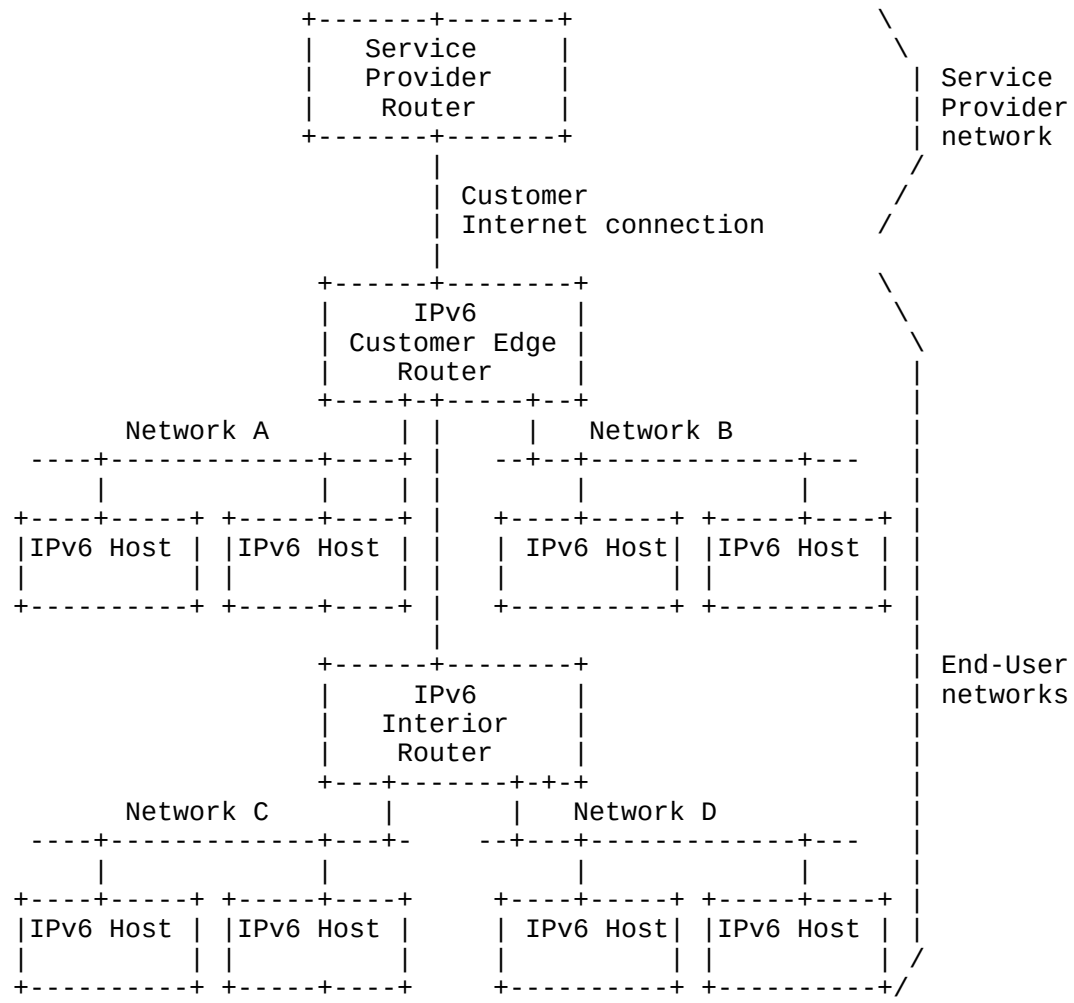


Figure 1.

For DNS, the operational expectation is that the end-user would be able to access home hosts from the home using DNS names instead of more cumbersome IPv6 addresses. Note that this is distinct from the requirement to access home hosts from outside the home.

End-users are expected to be able to receive multicast video in the home without requiring the CE router to include the cost of supporting full multicast routing protocols.

5. Advanced Features and Feature Requirements

The IPv6 CE router will need to support connectivity to one or more access network architectures. This document describes an IPv6 CE router that is not specific to any particular architecture or Service Provider, and supports all commonly used architectures.

5.1. DNS

D-1: For local DNS queries for configuration, the CE Router MAY include a DNS server to handle local queries. Non-local queries can be forwarded unchanged to a DNS server specified in the DNS server DHCPv6 option. The CE Router MAY also include DNS64 functionality which is specified in [I-D.bagnulo-behave-dns64].

D-2: The local DNS server MAY also handle renumbering from the Service Provider provided prefix for local names used exclusively inside the home (the local AAAA and PTR records are updated). This capability provides connectivity using local DNS names in the home after a Service Provider renumbering. A CE Router MAY add local DNS entries based on dynamic requests from the LAN segment(s). The protocol to carry such requests from hosts to the CE Router is yet to be described.

5.2. Multicast Behavior

This section is only applicable to a CE Router with at least one LAN interface. A host in the home is expected to receive multicast video. Note the CE Router resides at edge of the home and the Service Provider, and the CE Router has at least one WAN connection for multiple LAN connections. In such a multiple LAN to a WAN toplogy at the CE Router edge, it is not necessary to run a multicast routing protocol and thus MLD Proxy as specified in [RFC4605] can be used. The CE Router discovers the hosts via a MLDv2 Router implementation on a LAN interface. A WAN interface of the CE Router interacts with the Service Provider router by sending MLD Reports and replying to MLD queries for multicast Group memberships for hosts in the home.

The CE router SHOULD implement MLD Proxy as specified in [RFC4605]. For the routed topology shown in Figure 1, each router implements a MLD Proxy. If the CE router implements MLD Proxy, the requirements on the CE Router for MLD Proxy are listed below.

WAN requirements, MLD Proxy:

WMLD-1: Consistent with [RFC4605], the CE router MUST NOT implement the router portion of MLDv2 for the WAN interface.

LAN requirements, MLD Proxy:

LMMLD-1: The CPE Router MUST follow the model described for MLD Proxy in [RFC4605] to implement multicast.

LMMLD-2: Consistent with [RFC4605], the LAN interfaces on the CPE router MUST NOT implement an MLDv2 Multicast Listener.

LAN requirements:

LM-1: If the CE Router has bridging configured between the LAN interfaces, then the LAN interfaces MUST support snooping of MLD [RFC3810] messages.

5.3. ND Proxy

LAN requirements:

LNDP-1: If the CE Router has only one /64 prefix to be used across multiple LAN interfaces and the CE Router supports any two LAN interfaces that cannot bridge data between them because the two interfaces have disparate MAC layers, then the CE Router MUST support Proxying Neighbor Advertisements as specified in Section 7.2.8 of [RFC4861]. If any two LAN interfaces support bridging between the interfaces, then Proxying Neighbor Advertisements is not necessary between the two interfaces. Legacy 3GPP networks have the following requirements:

1. No DHCPv6 prefix is delegated to the CE Router.

2. Only one /64 is available on the WAN link.

3. The link types between the WAN interface and LAN interface(s) are disparate and, therefore, can’t be bridged.

4. No NAT66 is to be used.

5. Each LAN interface needs global connectivity.

6. Uses SLAAC to configure LAN interface addresses.

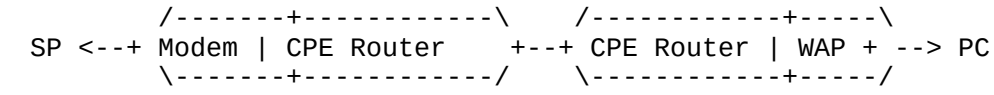
For these legacy 3GPP networks, the CPE Router MUST support ND Proxy between the WAN and LAN interface(s). If a CE

Router will never be deployed in an environment with these characteristics, then ND Proxy is not necessary.

5.4. Routed network behavior

CPE Router Behavior in a routed network:

R-1: One example of the CPE Router use in the home is shown below. The home has a broadband modem combined with a CPE Router, all in one device. The LAN interface of the device is connected to another standalone CPE Router that supports a wireless access point. To support such a network, this document recommends using prefix delegation of the prefix obtained either via IA_PD from WAN interface or a ULA from the LAN interface. The network interface of the downstream router MAY obtain an IA_PD via stateful DHCPv6. If the CPE router supports the routed network through a vendor specific automatic prefix delegation, the CPE router MUST support a DHCPv6 server or DHCPv6 relay agent. Further, if an IA_PD is used, the Service Provider or user MUST allocate an IA_PD or ULA prefix short enough to be delegated and subsequently used for SLAAC. Therefore, a prefix length shorter than /64 is needed. The CPE Router MAY support and IGP in the home network.



WAP = Wireless Access Point

Figure 2.

5.5. Transition Technologies Support

5.5.1. Dual-Stack(DS)-Lite

Even as users migrate from IPv4 to IPv6 addressing, a significant percentage of Internet resources and content will remain accessible only through IPv4. Also, many end-user devices will only support IPv4. As a consequence, Service Providers require mechanisms to allow customers to continue to access content and resources using IPv4 even after the last IPv4 allocations have been fully depleted. One technology that can be used for IPv4 address extension is DS-Lite.

DS-Lite enables a Service Provider to share IPv4 addresses among multiple customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) tunneling and Carrier Grade NAT. More specifically, Dual-Stack-Lite encapsulates IPv4 traffic inside an IPv6 tunnel at the IPv6 CE Router and sends it to a Service Provider Address Family Translation Router (AFTR). Configuration of the IPv6 CE Router to support IPv4 LAN traffic is outside the scope of this document.

The IPv6 CE Router SHOULD implement DS-Lite functionality as specified in [I-D.ietf-softwire-dual-stack-lite].

WAN requirements:

DLW-1: To facilitate IPv4 extension over an IPv6 network, if the CE Router supports DS-Lite functionality, the CE Router WAN interface MUST implement a B4 Interface as specified in [I-D.ietf-softwire-dual-stack-lite].

DLW-2: If the IPv6 CE Router implements DS-Lite functionality, the CE Router MUST support using a DS-Lite DHCPv6 option [I-D.ietf-softwire-ds-lite-tunnel-option] to configure the DS-Lite tunnel. The IPv6 CE Router MAY use other mechanisms to configure DS-Lite parameters. Such mechanisms are outside the scope of this document.

DLW-3: IPv6 CE Router MUST NOT perform IPv4 Network Address Translation (NAT) on IPv4 traffic encapsulated using DS-Lite.

DLW-4: If the IPv6 CE Router is configured with a public IPv4 address on its WAN interface, where public IPv4 address is defined as any address which is not in the private IP address space specified in [RFC1918] and also not in the reserved IP address space specified in [I-D.ietf-softwire-dual-stack-lite], then the IPv6 CE Router MUST disable the DS-Lite B4 element.

DLW-5: If DS-Lite is operational on the IPv6 CE Router, multicast data MUST NOT be sent on any DS-Lite tunnel.

5.5.2. 6rd

The IPv6 CE Router can be used to offer IPv6 service to a LAN, even when the WAN access network only supports IPv4. One technology that supports IPv6 service over an IPv4 network is IPv6 Rapid Deployment (6rd). 6rd encapsulates IPv6 traffic from the end user LAN inside IPv4 at the IPv6 CE Router and sends it to a Service Provider Border Relay (BR). The IPv6 CE Router calculates a 6rd delegated IPv6 prefix during 6rd configuration, and sub-delegates the 6rd delegated

prefix to devices in the LAN.

The IPv6 CE Router SHOULD implement 6rd functionality as specified in [RFC5969].

6rd requirements:

6RD-1: If the IPv6 CE Router implements 6rd functionality, the CE Router WAN interface MUST support at least one 6rd Virtual Interface and 6rd CE functionality as specified in [RFC5969].

6RD-2: If the IPv6 CE Router implements 6rd CE functionality, it MUST support using the 6rd DHCPv4 Option (212) for 6rd configuration. The IPv6 CE Router MAY use other mechanisms to configure 6rd parameters. Such mechanisms are outside the scope of this document.

6RD-3: If 6rd is operational on the IPv6 CE Router, multicast data MUST NOT be sent on any 6rd tunnel.

5.5.3. Transition Technologies Coexistence

Run the following four in parallel to provision CPE router connectivity to the Service Provider:

1. Initiate IPv4 address acquisition.

2. Initiate IPv6 address acquisition as specified by [I-D.ietf-v6ops-ipv6-cpe-router].

3. If 6rd is provisioned, initiate 6rd.

4. If DS-Lite is provisioned, initiate DS-Lite.

The default route for IPv6 through the native physical interface should have preference over the 6rd tunnel interface. The default route for IPv4 through the native physical interface should have preference over the DS-Lite tunnel interface.

5.6. Quality Of Service

Q-1: The CPE router MAY support differentiated services [RFC2474].

5.7. Unicast Data Forwarding

The null route introduced by the WPD-6 requirement in [I-D.ietf-v6ops-ipv6-cpe-router] has lower precedence than other routes except for the default route.

6. Security Considerations

None.

7. Acknowledgements

Thanks to the following people (in alphabetical order) for their guidance and feedback:

Mikael Abrahamsson, Merete Asak, Scott Beuker, Mohamed Boucadair, Rex Bullinger, Brian Carpenter, Remi Denis-Courmont, Gert Doering, Alain Durand, Katsunori Fukuoka, Tony Hain, Thomas Herbst, Kevin Johns, Stephen Kramer, Victor Kuarsingh, Francois-Xavier Le Bail, Chad Mikkelson, David Miles, Shin Miyakawa, Jean-Francois Mule, Michael Newbery, Carlos Pignataro, John Pomeroy, Antonio Querubin, Teemu Savolainen, Matt Schmitt, Hiroki Sato, Mark Townsley, Bernie Volz, James Woodyatt, Dan Wing and Cor Zwart

This draft is based in part on CableLabs' eRouter specification. The authors wish to acknowledge the additional contributors from the eRouter team:

Ben Bekele, Amol Bhagwat, Ralph Brown, Eduardo Cardona, Margo Dolas, Toerless Eckert, Doc Evans, Roger Fish, Michelle Kuska, Diego Mazzola, John McQueen, Harsh Parandekar, Michael Patrick, Saifur Rahman, Lakshmi Raman, Ryan Ross, Ron da Silva, Madhu Sudan, Dan Torbet and Greg White.

8. Contributors

The following people have participated as co-authors or provided substantial contributions to this document: Ralph Droms, Kirk Erichsen, Fred Baker, Jason Weil, Lee Howard, Jean-Francois Tremblay, Yiu Lee, John Jason Brzozowski and Heather Kirksey.

9. IANA Considerations

This memo includes no request to IANA.

10. References

10.1. Normative References

[I-D.bagnulo-behave-dns64]
Bagnulo, M., Sullivan, A., Matthews, P., Beijnum, I., and M. Endo, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-bagnulo-behave-dns64-02 (work in progress), March 2009.

[I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-09 (work in progress), March 2011.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 2011.

[I-D.ietf-v6ops-ipv6-cpe-router]
Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-ipv6-cpe-router-09 (work in progress), December 2010.

[I-D.vyncke-advanced-ipv6-security]
Vyncke, E. and M. Townsley, "Advanced Security for IPv6 CPE", draft-vyncke-advanced-ipv6-security-01 (work in progress), March 2010.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, January 1997.

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

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

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

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

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, May 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

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

[RFC4294] Loughney, J., "IPv6 Node Requirements", RFC 4294, April 2006.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC4779] Asadullah, S., Ahmed, A., Popoviciu, C., Savola, P., and J. Palet, "ISP IPv6 Deployment Scenarios in Broadband Access Networks", RFC 4779, January 2007.

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

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.

[RFC5072] S.Varada, Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, September 2007.

[RFC5571] Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)", RFC 5571, June 2009.

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

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

10.2. Informative References

[I-D.ietf-behave-v6v4-framework]
Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", draft-ietf-behave-v6v4-framework-10 (work in progress), August 2010.

[UPnP-IGD]
UPnP Forum, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)", November 2001, <http://www.upnp.org/standardizeddcps/igd.asp>.

Authors' Addresses

Hemant Singh
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 1622
Email: shemant@cisco.com
URI: http://www.cisco.com/

Wes Beebee
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 2030
Email: wbeebee@cisco.com
URI: http://www.cisco.com/

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

Email: c.donley@cablelabs.com

Barbara Stark
AT&T
725 W Peachtree St
Atlanta, GA 30308
USA

Email: barbara.stark@att.com

Ole Troan (editor)
Cisco Systems, Inc.
Veversmauet 8
N-5017 BERGEN,
Norway

Email: ot@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

H. Singh
W. Beebee
Cisco Systems, Inc.
C. Donley
CableLabs
B. Stark
ATT
O. Troan, Ed.
Cisco Systems, Inc.

July 11, 2011

Advanced Requirements for IPv6 Customer Edge Routers
draft-ietf-v6ops-ipv6-cpe-router-bis-01

Abstract

This document continues the work undertaken by the IPv6 CE Router Phase I work in the IETF v6ops Working Group. Advanced requirements or Phase II work is covered in this document.

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 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document defines Advanced IPv6 features for a residential or small office router referred to as an IPv6 CE router. Typically these routers also support IPv4. The IPv6 End-user Network Architecture for such a router is described in [RFC6204]. This version of the document includes the requirements for Advanced features.

1.1. Requirements Language

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

2. Terminology

End-user Network	one or more links attached to the IPv6 CE router that connect IPv6 hosts.
IPv6 Customer Edge router	a node intended for home or small office use which forwards IPv6 packets not explicitly addressed to itself. The IPv6 CE router connects the end-user network to a service provider network.
IPv6 host	any device implementing an IPv6 stack receiving IPv6 connectivity through the IPv6 CE router
LAN interface	an IPv6 CE router’s attachment to a link in the end-user network. Examples are Ethernets (simple or bridged), 802.11 wireless or other LAN technologies. An IPv6 CE router may have one or more network layer LAN Interfaces.
Service Provider	an entity that provides access to the Internet. In this document, a Service Provider specifically offers Internet access using IPv6, and may also offer IPv4 Internet access. The Service Provider can provide such access over a variety of different transport methods such as DSL, cable, wireless, and others.

WAN interface	an IPv6 CE router's attachment to a link used to provide connectivity to the Service Provider network; example link technologies include Ethernets (simple or bridged), PPP links, Frame Relay, or ATM networks as well as Internet-layer (or higher-layer) "tunnels", such as tunnels over IPv4 or IPv6 itself.

3. Conceptual Configuration Variables

The CE Router maintains such a list of conceptual optional configuration variables.

1. Enable an IGP on the LAN.

2. Configure 6rd configuration.

3. Configure IPv6 for 6rd to have IPv6 traffic go to the 6rd Border Relay vs. directly to peers.

4. Architecture

This document extends the architecture described in [RFC6204] to cover a strictly larger set of operational scenarios. In particular, QoS, multicast, DNS, routed network in the home, transition technologies, and conceptual configuration variables. This document also extends the model described in [RFC6204] to a two router topology where the two routers are connected back-to-back (the LAN of one router is connected to the WAN of the other router). This topology is depicted below:

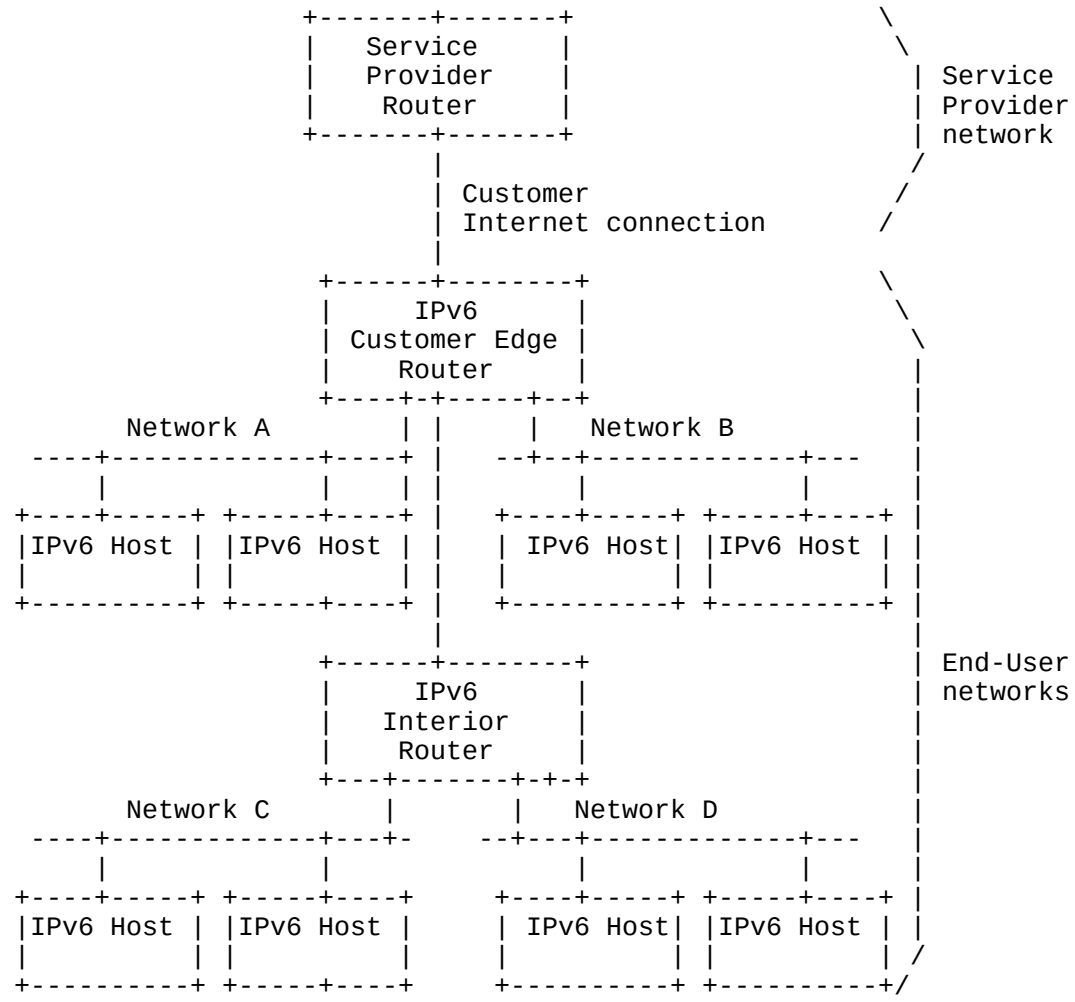


Figure 1.

For DNS, the operational expectation is that the end-user would be able to access home hosts from the home using DNS names instead of more cumbersome IPv6 addresses. Note that this is distinct from the requirement to access home hosts from outside the home.

End-users are expected to be able to receive multicast video in the home without requiring the CE router to include the cost of supporting full multicast routing protocols.

5. Advanced Features and Feature Requirements

The IPv6 CE router will need to support connectivity to one or more access network architectures. This document describes an IPv6 CE router that is not specific to any particular architecture or Service Provider, and supports all commonly used architectures.

5.1. DNS

D-1: The CE Router MAY include a DNS server authoritative for .local to handle local queries. If the service provider specifies one or more DNS resolvers in DHCP configuration options, the CE router SHOULD forward all non-local DNS queries unchanged to those servers. The CE Router MAY also include DNS64 functionality which is specified in [RFC6147].

5.2. Multicast Behavior

This section is only applicable to a CE Router with at least one LAN interface. A host in the home is expected to receive multicast video. Note the CE Router resides at edge of the home and the Service Provider, and the CE Router has at least one WAN connection for multiple LAN connections. In such a multiple LAN to a WAN toplogy at the CE Router edge, it is not necessary to run a multicast routing protocol and thus MLD Proxy as specified in [RFC4605] can be used. The CE Router discovers the hosts via a MLDv2 Router implementation on a LAN interface. A WAN interface of the CE Router interacts with the Service Provider router by sending MLD Reports and replying to MLD queries for multicast Group memberships for hosts in the home.

The CE router SHOULD implement MLD Proxy as specified in [RFC4605]. For the routed topology shown in Figure 1, each router implements a MLD Proxy. If the CE router implements MLD Proxy, the requirements on the CE Router for MLD Proxy are listed below.

WAN requirements, MLD Proxy:

WMLD-1: Consistent with [RFC4605], the CE router MUST NOT implement the router portion of MLDv2 for the WAN interface.

LAN requirements, MLD Proxy:

LMMLD-1: The CPE Router MUST follow the model described for MLD Proxy in [RFC4605] to implement multicast.

LMMLD-2: Consistent with [RFC4605], the LAN interfaces on the CPE router MUST NOT implement an MLDv2 Multicast Listener.

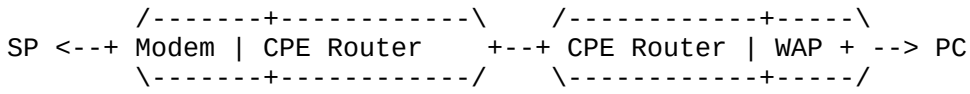
LAN requirements:

LM-1: If the CE Router has bridging configured between the LAN interfaces, then the LAN interfaces MUST support snooping of MLD [RFC3810] messages as per [RFC4541] .

5.3. Routed network behavior

CPE Router Behavior in a routed network:

R-1: One example of the CPE Router use in the home is shown below. The home has a broadband modem combined with a CPE Router, all in one device. The LAN interface of the device is connected to another standalone CPE Router that supports a wireless access point. To support such a network, this document recommends using prefix delegation of the prefix obtained either via IA_PD from WAN interface or a ULA from the LAN interface. The network interface of the downstream router MAY obtain an IA_PD via stateful DHCPv6. If the CPE router supports the routed network through a vendor specific automatic prefix delegation, the CPE router MUST support a DHCPv6 server or DHCPv6 relay agent. Further, if an IA_PD is used, the Service Provider or user MUST allocate an IA_PD or ULA prefix short enough to be delegated and subsequently used for SLAAC. Therefore, a prefix length shorter than /64 is needed. The CPE Router MAY support and IGP in the home network.



WAP = Wireless Access Point

Figure 2.

5.4. Transition Technologies Support

5.4.1. Dual-Stack(DS)-Lite

Even as users migrate from IPv4 to IPv6 addressing, a significant percentage of Internet resources and content will remain accessible

only through IPv4. Also, many end-user devices will only support IPv4. As a consequence, Service Providers require mechanisms to allow customers to continue to access content and resources using IPv4 even after the last IPv4 allocations have been fully depleted. One technology that can be used for IPv4 address extension is DS-Lite.

DS-Lite enables a Service Provider to share IPv4 addresses among multiple customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) tunneling and Carrier Grade NAT. More specifically, Dual-Stack-Lite encapsulates IPv4 traffic inside an IPv6 tunnel at the IPv6 CE Router and sends it to a Service Provider Address Family Translation Router (AFTR). Configuration of the IPv6 CE Router to support IPv4 LAN traffic is outside the scope of this document.

The IPv6 CE Router SHOULD implement DS-Lite functionality as specified in [I-D.ietf-softwire-dual-stack-lite].

WAN requirements:

DLW-1: To facilitate IPv4 extension over an IPv6 network, if the CE Router supports DS-Lite functionality, the CE Router WAN interface MUST implement a B4 Interface as specified in [I-D.ietf-softwire-dual-stack-lite].

DLW-2: If the IPv6 CE Router implements DS-Lite functionality, the CE Router MUST support using a DS-Lite DHCPv6 option [I-D.ietf-softwire-ds-lite-tunnel-option] to configure the DS-Lite tunnel. The IPv6 CE Router MAY use other mechanisms to configure DS-Lite parameters. Such mechanisms are outside the scope of this document.

DLW-3: IPv6 CE Router MUST NOT perform IPv4 Network Address Translation (NAT) on IPv4 traffic encapsulated using DS-Lite.

DLW-4: If the IPv6 CE Router is configured with a public IPv4 address on its WAN interface, where public IPv4 address is defined as any address which is not in the private IP address space specified in [RFC1918] and also not in the reserved IP address space specified in [I-D.ietf-softwire-dual-stack-lite], then the IPv6 CE Router MUST disable the DS-Lite B4 element.

DLW-5: If DS-Lite is operational on the IPv6 CE Router, multicast data MUST NOT be sent on any DS-Lite tunnel.

5.4.2. 6rd

The IPv6 CE Router can be used to offer IPv6 service to a LAN, even when the WAN access network only supports IPv4. One technology that supports IPv6 service over an IPv4 network is IPv6 Rapid Deployment (6rd). 6rd encapsulates IPv6 traffic from the end user LAN inside IPv4 at the IPv6 CE Router and sends it to a Service Provider Border Relay (BR). The IPv6 CE Router calculates a 6rd delegated IPv6 prefix during 6rd configuration, and sub-delegates the 6rd delegated prefix to devices in the LAN.

The IPv6 CE Router SHOULD implement 6rd functionality as specified in [RFC5969].

6rd requirements:

6RD-1: If the IPv6 CE Router implements 6rd functionality, the CE Router WAN interface MUST support at least one 6rd Virtual Interface and 6rd CE functionality as specified in [RFC5969].

6RD-2: If the IPv6 CE Router implements 6rd CE functionality, it MUST support user-entered configuration and using the 6rd DHCPv4 Option (212) for 6rd configuration. The IPv6 CE Router MAY use other mechanisms to configure 6rd parameters. Such mechanisms are outside the scope of this document.

6RD-3: If the CE router implements 6rd functionality, it MUST allow the user to specify whether all IPv6 traffic goes to the 6rd Border Relay, or whether other destinations within the same 6rd domain are routed directly to those destinations. The CE router MAY use other mechanisms to configure this. Such mechanisms are outside the scope of this document.

6RD-4: If 6rd is operational on the IPv6 CE Router, multicast data MUST NOT be sent on any 6rd tunnel.

5.4.3. Transition Technologies Coexistence

Run the following four in parallel to provision CPE router connectivity to the Service Provider:

1. Initiate IPv4 address acquisition.

2. Initiate IPv6 address acquisition as specified by [RFC6204].

3. If 6rd is provisioned, initiate 6rd.

4. If DS-Lite is provisioned, initiate DS-Lite.

The default route for IPv6 through the native physical interface should have preference over the 6rd tunnel interface. The default route for IPv4 through the native physical interface should have preference over the DS-Lite tunnel interface.

5.5. Quality Of Service

Q-1: The CPE router MAY support differentiated services [RFC2474].

5.6. Unicast Data Forwarding

The null route introduced by the WPD-6 requirement in [RFC6204] has lower precedence than other routes except for the default route.

5.7. Additional DHCPv6 WAN Requirement

When the WAN interface sends a DHCPV6 SOLICIT message, the CE router SHOULD request all mandatory information (IA_NA and IA_PD options) in the SOLICIT regardless of whether any partial information was received in response to previous SOLICITs.

6. Security Considerations

None.

7. Acknowledgements

Thanks to the following people (in alphabetical order) for their guidance and feedback:

Mikael Abrahamsson, Merete Asak, Scott Beuker, Mohamed Boucadair, Rex Bullinger, Brian Carpenter, Remi Denis-Courmont, Gert Doering, Alain Durand, Katsunori Fukuoka, Tony Hain, Thomas Herbst, Kevin Johns, Stephen Kramer, Victor Kuarsingh, Francois-Xavier Le Bail, Chad Mikkelson, David Miles, Shin Miyakawa, Jean-Francois Mule, Michael Newbery, Carlos Pignataro, John Pomeroy, Antonio Querubin, Teemu Savolainen, Matt Schmitt, Hiroki Sato, Mark Townsley, Bernie Volz, James Woodyatt, Dan Wing and Cor Zwart

This draft is based in part on CableLabs' eRouter specification. The authors wish to acknowledge the additional contributors from the eRouter team:

Ben Bekele, Amol Bhagwat, Ralph Brown, Eduardo Cardona, Margo Dolas,

Toerless Eckert, Doc Evans, Roger Fish, Michelle Kuska, Diego Mazzola, John McQueen, Harsh Parandekar, Michael Patrick, Saifur Rahman, Lakshmi Raman, Ryan Ross, Ron da Silva, Madhu Sudan, Dan Torbet and Greg White.

8. Contributors

The following people have participated as co-authors or provided substantial contributions to this document: Ralph Droms, Kirk Erichsen, Fred Baker, Jason Weil, Lee Howard, Jean-Francois Tremblay, Yiu Lee, John Jason Brzozowski and Heather Kirksey.

9. IANA Considerations

This memo includes no request to IANA.

10. References

10.1. Normative References

[I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-10 (work in progress), March 2011.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-11 (work in progress), May 2011.

[I-D.vyncke-advanced-ipv6-security]
Vyncke, E. and M. Townsley, "Advanced Security for IPv6 CPE", draft-vyncke-advanced-ipv6-security-01 (work in progress), March 2010.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080,

January 1997.

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

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

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

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

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, May 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

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

[RFC4294] Loughney, J., "IPv6 Node Requirements", RFC 4294, April 2006.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC4779] Asadullah, S., Ahmed, A., Popoviciu, C., Savola, P., and J. Palet, "ISP IPv6 Deployment Scenarios in Broadband Access Networks", RFC 4779, January 2007.

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

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.

[RFC5072] S.Varada, Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, September 2007.

[RFC5571] Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)", RFC 5571, June 2009.

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

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[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.

[RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

10.2. Informative References

[I-D.ietf-behave-v6v4-framework]
Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", draft-ietf-behave-v6v4-framework-10 (work in progress), August 2010.

[UPnP-IGD]
UPnP Forum, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)", November 2001, <http://www.upnp.org/standardizeddcps/igd.asp>.

Authors' Addresses

Hemant Singh
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 1622
Email: shemant@cisco.com
URI: http://www.cisco.com/

Wes Beebee
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 2030
Email: wbeebee@cisco.com
URI: http://www.cisco.com/

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

Email: c.donley@cablelabs.com

Barbara Stark
ATT
725 W Peachtree St
Atlanta, GA 30308
USA

Email: barbara.stark@att.com

Ole Troan (editor)
Cisco Systems, Inc.
Veversmauet 8
N-5017 BERGEN,
Norway

Email: ot@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: June 9, 2011

O. Troan, Ed.
Cisco
D. Miles
Alcatel-Lucent
S. Matsushima
SOFTBANK TELECOM Corp.
T. Okimoto
NTT West
D. Wing
Cisco

December 6, 2010

IPv6 Multihoming without Network Address Translation
draft-ietf-v6ops-multihoming-without-nat66-00

Abstract

Network Address and Port Translation (NAPT) works well for conserving global addresses and addressing multihoming requirements, because an IPv4 NAPT router implements three functions: source address selection, next-hop resolution and optionally DNS resolution. For IPv6 hosts one approach could be the use of IPv6 NAT. However, NAT should be avoided, if at all possible, to permit transparent host-to-host connectivity. In this document, we analyze the use cases of multihoming. We also describe functional requirements for multihoming without the use of NAT in IPv6 for hosts and small IPv6 networks that would otherwise be unable to meet minimum IPv6 allocation criteria .

Table of Contents

1. Introduction

IPv6 provides enough globally unique addresses to permit every conceivable host on the Internet to be uniquely addressed without the requirement for Network Address Port Translation (NAPT [RFC3022]) offering a renaissance in host-to-host transparent connectivity.

Unfortunately, this may not be possible due to the necessity of NAT even in IPv6, because of multihoming.

Multihoming is a blanket term to describe a host or small network that is connected to more than one upstream network. Whenever a host or small network (which does not meet minimum IPv6 allocation criteria) is connected to multiple upstream networks IPv6 addressing is assigned by each respective service provider resulting in hosts with more than one active IPv6 address. As each service provided is allocated a different address space from its Internet Registry, it in-turn assigns a different address space to the end-user network or host. For example, a remote access user may use a VPN to simultaneously connect to a remote network and retain a default route to the Internet for other purposes.

In IPv4 a common solution to the multihoming problem is to employ NAPT on a border router and use private address space for individual host addressing. The use of NAPT allows hosts to have exactly one IP address visible on the public network and the combination of NAPT with provider-specific outside addresses (one for each uplink) and destination-based routing insulates a host from the impacts of multiple upstream networks. The border router may also implement a DNS cache or DNS policy to resolve address queries from hosts.

It is our goal to avoid the IPv6 equivalent of NAT. To reach this goal, mechanisms are needed for end-user hosts to have multiple address assignments and resolve issues such as which address to use for sourcing traffic to which destination:

- If multiple routers exist on a single link the host must appropriately select next-hop for each connected network. Routing protocols that would normally be employed for router-to-router network advertisement seem inappropriate for use by individual hosts.

- Source address selection also becomes difficult whenever a host has more than one address within the same address scope. Current address selection criteria may result in hosts using an arbitrary or random address when sourcing upstream traffic. Unfortunately, for the host, the appropriate source address is a function of the upstream network for which the packet is bound for. If an

upstream service provider uses IP anti-spoofing or uRPF, it is conceivable that the packets that have inappropriate source address for the upstream network would never reach their destination.

- o In a multihomed environment, different DNS scopes or partitions may exist in each independent upstream network. A DNS query sent to an arbitrary upstream resolver may result in incorrect or poisoned responses.

In short, while IPv6 facilitates hosts having more than one address in the same address scope, the application of this causes significant issues for a host from routing, source address selection and DNS resolution perspectives. A possible consequence of assigning a host multiple identical-scoped addresses is severely impaired IP connectivity.

If a host connects to a network behind an IPv4 NAPT, the host has one private address in the local network. There is no confusion. The NAT becomes the gateway of the host and forwards the packet to an appropriate network when it is multihomed. It also operates a DNS cache server, which receives all DNS inquires, and gives a correct answer to the host.

In this document, we identify the functions present in multihomed IPv4 NAPT environments and propose requirements that address multihomed IPv6 environments without using IPv6 NAT.

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 RFC 2119 [RFC2119].

NAT66 or IPv6 NAT
: The terms "NAT66" and "IPv6 NAT" refer to [I-D.mrw-nat66].

NAPT
: Network Address Port Translation as described in [RFC3022]. In other contexts, NAPT is often pronounced "NAT" or written as "NAT".

Multihomed with multi-prefix (MHMP)
: A host implementation which supports the mechanisms described in this document. Namely source address selection policy, next-hop selection and DNS selection policy.

3. IPv6 multihomed network scenarios

In this section, we classify three scenarios of the multihoming environment.

3.1. Classification of network scenarios for multihomed host

Scenario 1:

In this scenario, two or more routers are present on a single link shared with the host(s). Each router is in turn connected to a different service provider network, which provides independent address assignment and DNS resolvers. A host in this environment would be offered multiple prefixes and DNS resolvers advertised from the two different routers.

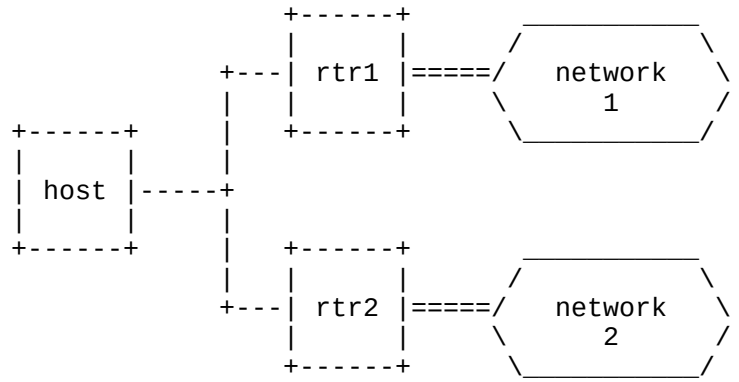


Figure 1: single uplink, multiple next-hop, multiple prefix (Scenario 1)

Figure 1 illustrates the host connecting to rtr1 and rtr2 via a shared link. Networks 1 and 2 are reachable via rtr1 and rtr2 respectively. When the host sends packets to network 1, the next-hop to network 1 is rtr1. Similarly, rtr2 is the next-hop to network 2.

- e.g., broadband service (Internet, VoIP, IPTV, etc.)

Scenario 2:

In this scenario, a single gateway router connects the host to two or more upstream service provider networks. This gateway router would receive prefix delegations from each independent service provider network and a different set of DNS resolvers. The gateway in turn advertises the provider prefixes to the host, and for DNS, may either

act as a lightweight DNS resolver/cache or may advertise the complete set of service provider DNS resolvers to the hosts.

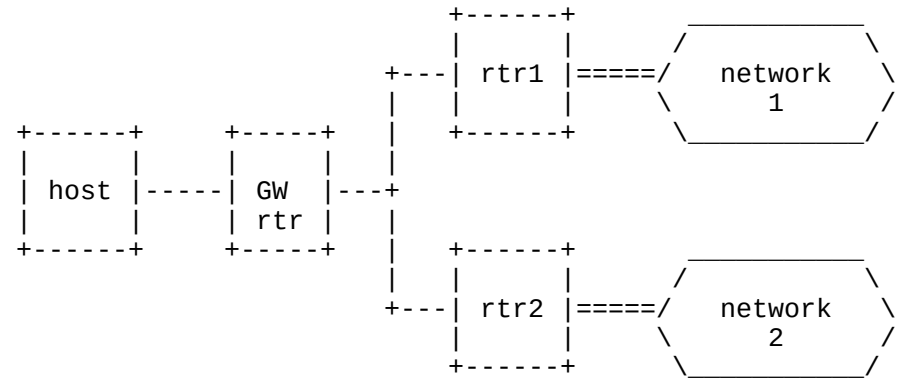


Figure 2: single uplink, single next-hop, multiple prefix (Scenario 2)

Figure 2 illustrates the host connected to GW rtr. GW rtr connects to networks 1 and 2 via rtr1 and rtr2, respectively. When the host sends packets to either network 1 or 2, the next-hop is GW rtr. When the packets are sent to network 1 (network 2), GW rtr forwards the packets to rtr1 (rtr2).

- e.g, Internet + VPN/ASP

Scenario 3:

In this scenario, a host has more than one active interfaces that connects to different routers and service provider networks. Each router provides the host with a different address prefix and set of DNS resolvers, resulting in a host with a unique address per link/ interface.

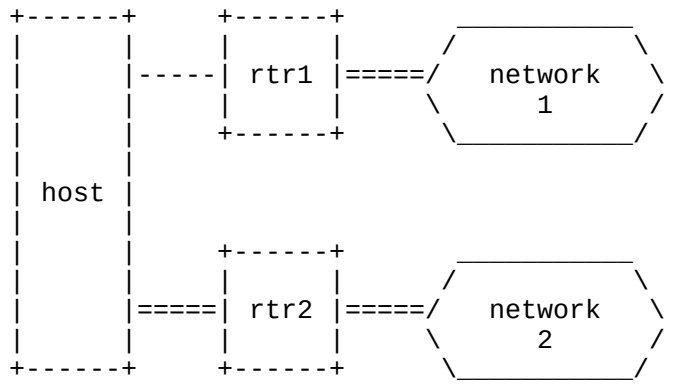


Figure 3: Multiple uplink, multiple next-hop, multiple prefix (Scenario 3)

Figure 3 illustrates the host connecting to rtr1 and rtr2 via a direct connection or a virtual link. When the host sends packets network 1, the next-hop to network 1 is rtr1. Similarly, rtr2 is the next-hop to network 2.

- e.g., Mobile Wifi + 3G, ISP A + ISP B

3.2. Multihomed network environment

In an IPv6 multihomed network, a host is assigned two or more IPv6 addresses and DNS resolvers from independent service provider networks. When this multihomed host attempts to connect with other hosts, it may incorrectly resolve the next-hop router, use an inappropriate source address, or use a DNS response from an incorrect service provider that may result in impaired IP connectivity.

Multihomed networks in IPv4 have been commonly implemented through the use of a gateway router with NAPT function (scenario 2 with NAPT). An analysis of the current IPv4 NAPT and DNS functions within the gateway router should provide a baseline set of requirements for IPv6 multihomed environments. A destination prefix/route is often used on the gateway router to separate traffic between the networks.

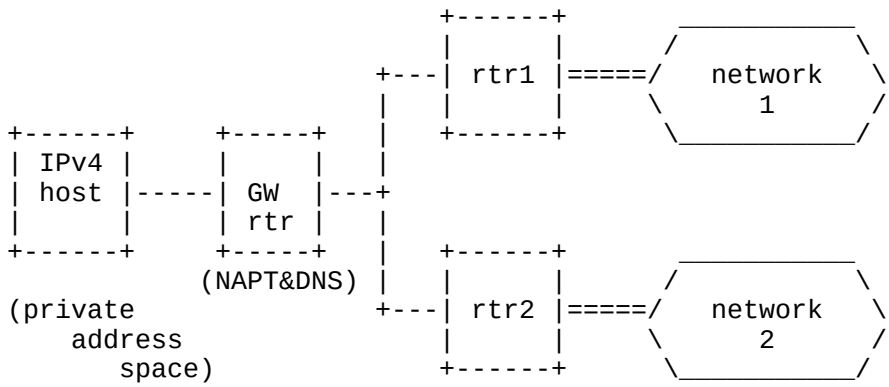


Figure 4: IPv4 Multihomed environment with Gateway Router performing NAPT

3.3. Multihomed Problem Statement

A multihomed IPv6 host has one or more assigned IPv6 addresses and DNS resolvers from each upstream service provider, resulting in the host having multiple valid IPv6 addresses and DNS resolvers. The host must be able to resolve the appropriate next-hop, the correct source address and DNS resolver to use based on the destination prefix. To prevent IP spoofing, operators will often implement IP filters and uRPF to discard traffic with an inappropriate source address, making it essential for the host to correctly resolve these three criteria before sourcing the first packet.

IPv6 has mechanisms for the provision of multiple routers on a single link and multiple address assignments to a single host. However, when these mechanisms are applied to the three scenarios in Section 3.1 a number of connectivity issues are identified:

Scenario 1:

The host has been assigned an address from each router and recognizes both rtr1 and rtr2 as valid default routers (in the default routers list).

- The source address selection policy on the host does not deterministically resolve a source address. Upstream uRPF or filter policies will discard traffic with source addresses that the operator did not assign.

- The host will select one of the two routers as the active default router. No traffic is sent to the other router.

Scenario 2:

The host has been assigned two different addresses from the single gateway router. The gateway router is the only default router on the link.

- The source address selection policy on the host does not deterministically resolve a source address. Upstream uRPF or filter policies will discard traffic with source addresses that the operator did not assign.

- The gateway router does not have a mechanism for determining which traffic should be sent to which network. If the gateway router is implementing host functions (ie, processing RA) then two valid default routers may be recognized.

Scenario 3:

A host has two separate interfaces and on each interface a different address is assigned. Each link has its own router.

- The host does not have enough information for determining which traffic should be sent to which upstream routers. The host will select one of the two routers as the active default router, and no traffic is sent to the other router.

- The default address selection rules select the address assigned to the outgoing interface as the source address. So, if a host has an appropriate routing table, an appropriate source address will be selected.

All scenarios:

- The host may use an incorrect DNS resolver for DNS queries.

4. Problem statement and analysis

The problems described in Section 3 can be classified into these three types:

- Wrong source address selection

- Wrong next-hop selection

- Wrong DNS server selection

This section reviews the problem statements presented above and the

proposed functional requirements to resolve the issues without employing IPv6 NAT.

4.1. Source address selection

A multihomed IPv6 host will typically have different addresses assigned from each service provider either on the same link (scenarios 1 & 2) or different links (scenario 3). When the host wishes to send a packet to any given destination, the current source address selection rules [RFC3484] may not deterministically resolve the correct source address when the host addressing was via RA or DHCPv6. [I-D.ietf-6man-addr-select-sol] describes the use of the policy table [RFC3484] to resolve this problem, but there is no mechanism defined to disseminate the policy table information to a host. A proposal is in [I-D.fujisaki-dhc-addr-select-opt] to provide a DHCPv6 mechanism for host policy table management.

Again, by employing DHCPv6, the server could restrict address assignment (of additional prefixes) only to hosts that support policy table management.

Scenario 1: "Host" needs to support the solution for this problem

Scenario 2: "Host" needs to support the solution for this problem

Scenario 3: If "Host" support the next-hop selection solution, there is no need to support the address selection functionality on the host.

4.2. Next-hop selection

A multihomed IPv6 host or gateway may have multiple uplinks to different service providers. Here each router would use Router Advertisements [RFC4861] for distributing default route/next-hop information to the host or gateway router.

In this case, the host or gateway router may select any valid default router from the default routers list, resulting in traffic being sent to the wrong router and discarded by the upstream service provider. Using the above scenarios as an example, whenever the host wishes to reach a destination in network 2 and there is no connectivity between networks 1 and 2 (as is the case for a walled-garden or closed service), the host or gateway router does not know whether to forward traffic to rtr1 or rtr2 to reach a destination in network 2. The host or gateway router may choose rtr1 as the default router, and traffic fails to reach the destination server. The host or gateway router requires route information for each upstream service provider, but the use of a routing protocol between a host and router causes

both configuration and scaling issues. For IPv4 hosts, the gateway router is often pre-configured with static route information or uses of Classless Static Route Options [RFC3442] for DHCPv4. Extensions to Router Advertisements through Default Router Preference and More-Specific Routes [RFC4191] provides for link-specific preferences but does not address per-host configuration in a multi-access topology because of its reliance on Router Advertisements. A DHCPv6 option, such as that in [I-D.dec-dhcpv6-route-option], is preferred for host-specific configuration. By employing a DHCPv6 solution, a DHCPv6 server could restrict address assignment (of additional prefixes) only to hosts that support more advanced next-hop and address selection requirements.

Scenario 1: "Host" needs to support the solution for this problem

Scenario 2: "GW rtr" needs to support the solution for this problem

Scenario 3: "Host" needs to support the solution for this problem

4.3. DNS server selection

A multihomed IPv6 host or gateway router may be provided multiple DNS resolvers through DHCPv6 or the experimental [RFC5006]. When the host or gateway router sends a DNS query, it would normally choose one of the available DNS resolvers for the query.

In the IPv6 gateway router scenario, the Broadband Forum [TR124] required that the query be sent to all DNS resolvers, and the gateway waits for the first reply. In IPv6, given our use of specific destination-based policy for both routing and source address selection, it is desirable to extend a policy-based concept to DNS resolver selection. Doing so can minimize DNS resolver load and avoid issues where DNS resolvers in different networks have connectivity issues, or the DNS resolvers are not publicly accessible. In the worst case, a DNS query may be unanswered if sent towards an incorrect resolver, resulting in a lack of connectivity.

An IPv6 multihomed host or gateway router should have the ability to select appropriate DNS resolvers for each service based on the domain space for the destination, and each service should provide rules specific to that network. [I-D.savolainen-mif-dns-server-selection] proposes a solution for DNS server selection policy enforcement solution with a DHCPv6 option.

Scenario 1: "Host" needs to support the solution for this problem

Scenario 2: "GW rtr" needs to support the solution for this problem

Scenario 3: "Host" needs to support the solution for this problem

5. Requirements

This section describes requirements that any solution multi-address and multi-uplink architectures need to meet.

5.1. End-to-End transparency

End-to-end transparency is a basic concept of the Internet. [RFC4966] states, "One of the major design goals for IPv6 is to restore the end-to-end transparency of the Internet. Therefore, because IPv6 is expected to remove the need for NATs and similar impediments to transparency, developers creating applications to work with IPv6 may be tempted to assume that the complex mechanisms employed by an application to work in a 'NATted' IPv4 environment are not required." The IPv6 multihoming solution SHOULD guarantee end-to-end transparency by avoiding IPv6 NAT.

5.2. Policy enforcement

The solution SHOULD have a function to enforce a policy on sites/ nodes. In particular, in a managed environment such as enterprise networks, an administrator has to control all nodes in his or her network.

The enforcement mechanisms should have:

- a function to distribute policies to nodes dynamically to update their behavior. When the network environment changes and the nodes' behavior has to be changed, a network administrator can modify the behavior of the nodes.

- a function to control every node centrally. A site administrator or a service provider could determine or could have an effect on the behavior at their users' hosts.

- a function to control node-specific behavior. Even when multiple nodes are on the same subnet, the mechanism should be able to provide a method for the network administrator to make nodes behave differently. For example, each node may have a different set of assigned prefixes. In such a case, the appropriate behavior may be different.

5.3. Scalability

The solution will have to be able to manage a large number of sites/ nodes. In services for residential users, provider edge devices have to manage thousands of sites. In such environments, sending packets periodically to each site may affect edge system performance.

6. Implementation approach

As mentioned in Section 4, in the multi-prefix environment, we have three problems in source address selection, next-hop selection, and DNS resolver selection. In this section, possible solution mechanisms for each problem are introduced and evaluated against the requirements in Section 5.

6.1. Source address selection

Possible solutions and their evaluation are summarized in [I-D.ietf-6man-addr-select-sol]. When those solutions are examined against the requirements in Section 5, the proactive approaches, such as the policy table distribution mechanism and the routing system assistance mechanism, are more appropriate in that they can propagate the network administrator’s policy directly. The policy distribution mechanism has an advantage with regard to the host’s protocol stack impact and the staticness of the assumed target network environment.

6.2. Next-hop selection

As for the source address selection problem, both a policy-based approach and a non policy-based approach are possible with regard to the next-hop selection problem. Because of the same requirements, the policy propagation-based solution mechanism, whatever the policy, should be more appropriate.

Routing information is a typical example of policy related to next-hop selection. If we assume source address-based routing at hosts or intermediate routers, the pairs of source prefixes and next-hops can be another example of next-hop selection policy.

The routing information-based approach has a clear advantage in implementation and is already commonly used.

The existing proposed or standardized routing information distribution mechanisms are routing protocols, such as RIPng and OSPFv3, the router advertisement (RA) extension option defined in [RFC4191], the DHCPv6 route information option proposed in [I-D.dec-dhcpv6-route-option], and the [TR069] standardized at BBF.

The RA-based mechanism has difficulty in per-host routing information distribution. The dynamic routing protocols such as RIPng are not usually used between the residential users and ISP networks because of their scalability implications. The DHCPv6 mechanism does not have these difficulties and has the advantages of its relaying functionality. It is commonly used and is thus easy to deploy.

[TR069], mentioned above, is a possible solution mechanism for routing information distribution to customer-premises equipment (CPE). It assumes, however, IP reachability to the Auto Configuration Server (ACS) is established. Therefore, if the CPE requires routing information to reach the ACS, [TR069] cannot be used to distribute this information.

6.3. DNS resolver selection

As in the above two problems, a policy-based approach and non policy-based approach are possible. In a non policy-based approach, a host or a home gateway router is assumed to send DNS queries to several DNS servers at once or to select one of the available servers.

In the non policy-based approach, by making a query to a resolver in a different service provider to that which hosts the service, a user could be directed to unexpected IP address or receive an invalid response, and thus cannot connect to the service provider’s private and legitimate service. For example, some DNS servers reply with different answers depending on the source address of the DNS query, which is sometimes called split-horizon. When the host mistakenly makes a query to a different provider’s DNS to resolve a FQDN of another provider’s private service, and the DNS resolver adopts the split-horizon configuration, the queried server returns an IP address of the non-private side of the service. Another problem with this approach is that it causes unnecessary DNS traffic to the DNS resolvers that are visible to the users.

The alternative of a policy-based approach is documented in [I-D.savolainen-mif-dns-server-selection], where several pairs of DNS resolver addresses and DNS domain suffixes are defined as part of a policy and conveyed to hosts in a new DHCP option. In an environment where there is a home gateway router, that router can act as a DNS proxy, interpret this option and distribute DNS queries to the appropriate DNS servers according to the policy.

7. Considerations for host without multi-prefix support

This section presents an alternative approach to mitigate the problem in a multihomed network. This approach will help IPv6 hosts that are

not capable of the enhancements for the source address selection policy, next-hop selection policy, and DNS selection policy described in Section 6.

7.1. IPv6 NAT

In a typical IPv4 multihomed network deployment, IPv4 NAPT is practically used and it can eventually avoid assigning multiple addresses to the hosts and solve the next-hop selection problem. In a similar fashion, IPv6 NAT can be used as a last resort for IPv6 multihomed network deployments where one needs to assign a single IPv6 address to a host.

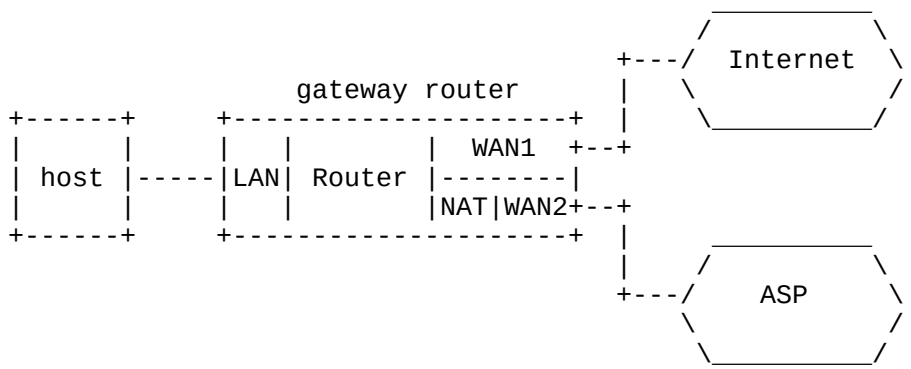


Figure 5: Legacy Host

The gateway router also has to support the two features, next-hop selection and DNS server selection, shown in Section 6.

The implementation and issues of IPv6 NAT are out of the scope of this document. They may be covered by another document under discussion [I-D.mrw-nat66].

7.2. Co-exisitence consideration

The above scenario relies on the assumption that only hosts without multi-prefix support are connected to the GW rtr in scenario 2. To allow the coexistence of non-MHMP hosts and MHMP hosts(i.e. hosts supporting multi-prefix with the enhancements for the source address selection), GW-rtr may need to treat those hosts separately.

An idea to achieve this is that GW-rtr identifies the hosts, and then assigns single prefix to non-MHMP hosts and assigns multiple prefix to MHMP hosts. In this case, GW-rtr can perform IPv6 NAT only for

the traffic from MHMP hosts if its source address is not appropriate.

Another idea is that GW-rtr assigns multiple prefix to the both hosts, and it performs IPv6 NAT for the traffic from non-MHMP hosts if its source address is not appropriate.

In scenario 1 and 3, the non-MHMP hosts can be placed behind the NAT box. In this case, non-MHMP host can access the service through the NAT box.

The implementation of identifying non-MHMP hosts and NAT policy is outside the scope of this document.

8. Security Considerations

This document does not define any new mechanisms. Each solution mechanisms should consider security risks independently. Security risks that occur as a result of combining solution mechanisms should be considered in another document.

9. IANA Considerations

This document has no IANA actions.

10. Contributors

The following people contributed to this document: Akiko Hattori, Arifumi Matsumoto, Frank Brockners, Fred Baker, Tomohiro Fujisaki, Jun-ya Kato, Shigeru Akiyama, Seiichi Morikawa, Mark Townsley, Wojciech Dec, Yasuo Kashimura, Yuji Yamazaki

11. References

11.1. Normative References

[I-D.dec-dhcpv6-route-option]
Dec, W., Mrugalski, T., Sun, T., and B. Sarikaya, "DHCPv6 Route Option", draft-dec-dhcpv6-route-option-05 (work in progress), September 2010.

[I-D.fujisaki-dhc-addr-select-opt]
Fujisaki, T., Matsumoto, A., and R. Hiromi, "Distributing Address Selection Policy using DHCPv6", draft-fujisaki-dhc-addr-select-opt-09 (work in progress),

March 2010.

[I-D.ietf-6man-addr-select-sol]
Matsumoto, A., Fujisaki, T., and R. Hiromi, "Solution approaches for address-selection problems", draft-ietf-6man-addr-select-sol-03 (work in progress), March 2010.

[I-D.mrw-nat66]
Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Address Translation (NAT66)", draft-mrw-nat66-00 (work in progress), October 2010.

[I-D.savolainen-mif-dns-server-selection]
Savolainen, T. and J. Kato, "Improved DNS Server Selection for Multi-Homed Nodes", draft-savolainen-mif-dns-server-selection-05 (work in progress), November 2010.

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

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

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

11.2. Informative References

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

[RFC3442] Lemon, T., Cheshire, S., and B. Volz, "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4", RFC 3442, December 2002.

[RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.

[RFC5006] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Option for DNS Configuration",

RFC 5006, September 2007.

[TR069] The BroadBand Forum, "TR-069, CPE WAN Management Protocol v1.1, Version: Issue 1 Amendment 2", December 2007.

[TR124] The BroadBand Forum, "TR-124i2, Functional Requirements for Broadband Residential Gateway Devices (work in progress)", May 2010.

Authors' Addresses

Ole Troan (editor)
Cisco
Bergen
Norway

Email: ot@cisco.com

David Miles
Alcatel-Lucent
Melbourne
Australia

Email: david.miles@alcatel-lucent.com

Satoru Matsushima
SOFTBANK TELECOM Corp.
Tokyo
Japan

Email: satoru.matsushima@tm.softbank.co.jp

Tadahisa Okimoto
NTT West
Osaka
Japan

Email: t.okimoto@rdc.west.ntt.co.jp

Dan Wing
Cisco
170 West Tasman Drive
San Jose
USA

Email: dwing@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 15, 2011

G. Nakibly
National EW Research &
Simulation Center
F. Templin
Boeing Research & Technology

March 14, 2011

Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations

draft-ietf-v6ops-tunnel-loops-05.txt

Abstract

This document is concerned with security vulnerabilities in IPv6-in-IPv4 automatic tunnels. These vulnerabilities allow an attacker to take advantage of inconsistencies between the IPv4 routing state and the IPv6 routing state. The attack forms a routing loop which can be abused as a vehicle for traffic amplification to facilitate DoS attacks. The first aim of this document is to inform on this attack and its root causes. The second aim is to present some possible mitigation measures.

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 September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

IPv6-in-IPv4 tunnels are an essential part of many migration plans for IPv6. They allow two IPv6 nodes to communicate over an IPv4-only network. Automatic tunnels that assign non-link-local IPv6 prefixes with stateless address mapping properties (hereafter called "automatic tunnels") are a category of tunnels in which a tunneled packet's egress IPv4 address is embedded within the destination IPv6 address of the packet. An automatic tunnel's router is a router that respectively encapsulates and decapsulates the IPv6 packets into and out of the tunnel.

Ref. [USENIX09] pointed out the existence of a vulnerability in the design of IPv6 automatic tunnels. Tunnel routers operate on the implicit assumption that the destination address of an incoming IPv6 packet is always an address of a valid node that can be reached via the tunnel. The assumption of path validity poses a denial of service risk as inconsistency between the IPv4 routing state and the IPv6 routing state allows a routing loop to be formed.

An attacker can exploit this vulnerability by crafting a packet which is routed over a tunnel to a node that is not participating in that tunnel. This node may forward the packet out of the tunnel to the native IPv6 network. There the packet is routed back to the ingress point that forwards it back into the tunnel. Consequently, the packet loops in and out of the tunnel. The loop terminates only when the Hop Limit field in the IPv6 header of the packet is decremented to zero. This vulnerability can be abused as a vehicle for traffic amplification to facilitate DoS attacks [RFC4732].

Without compensating security measures in place, all IPv6 automatic tunnels that are based on protocol-41 encapsulation [RFC4213] are vulnerable to such an attack including ISATAP [RFC5214], 6to4 [RFC3056] and 6rd [RFC5969]. It should be noted that this document does not consider non-protocol-41 encapsulation attacks. In particular, we do not address the Teredo [RFC4380] attacks described in [USENIX09]. These attacks are considered in [I-D.gont-6man-teredo-loops].

The aim of this document is to shed light on the routing loop attack and describe possible mitigation measures that should be considered by operators of current IPv6 automatic tunnels and by designers of future ones. We note that tunnels may be deployed in various operational environments, e.g. service provider network, enterprise network, etc. Specific issues related to the attack which are derived from the operational environment are not considered in this document.

2. A Detailed Description of the Attack

In this section we shall denote an IPv6 address of a node reached via a given tunnel by the prefix of the tunnel and an IPv4 address of the tunnel end point, i.e., Addr(Prefix, IPv4). Note that the IPv4 address may or may not be part of the prefix (depending on the specification of the tunnel's protocol). The IPv6 address may be dependent on additional bits in the interface ID, however for our discussion their exact value is not important.

The two victims of this attack are routers - R1 and R2 - of two different tunnels - T1 and T2. Both routers have the capability to forward IPv6 packets in and out of their respective tunnels. The two tunnels need not be based on the same tunnel protocol. The only condition is that the two tunnel protocols be based on protocol-41 encapsulation. The IPv4 address of R1 is IP1, while the prefix of its tunnel is Prf1. IP2 and Prf2 are the respective values for R2. We assume that IP1 and IP2 belong to the same address realm, i.e., they are either both public, or both private and belong to the same internal network. The following network diagram depicts the locations of the two routers. The numbers indicate the packets of the attack and the path they traverse as described below.

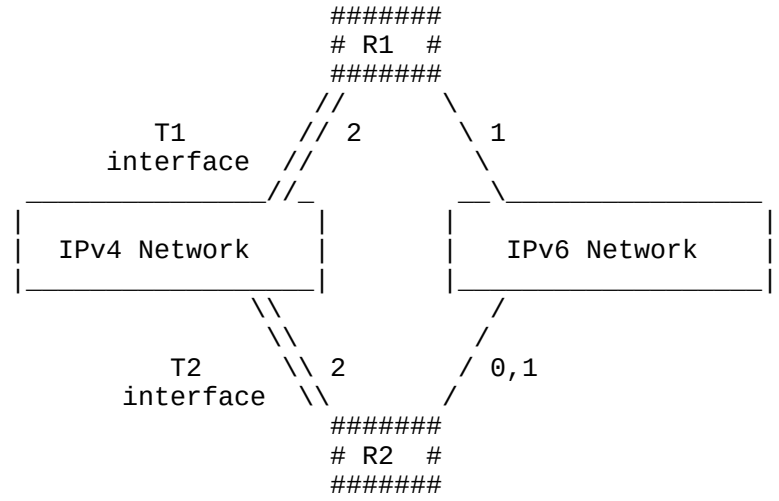


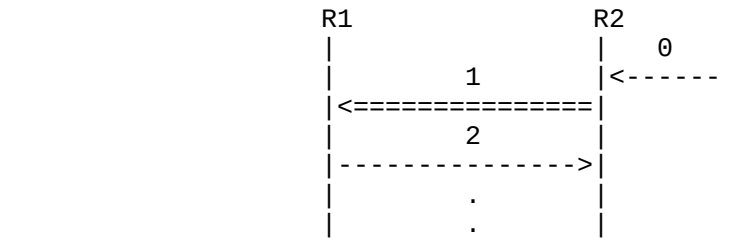
Figure 1: The network setting of the attack

The attack is depicted in Figure 2. It is initiated by sending an IPv6 packet (packet 0 in Figure 2) destined to a fictitious end point that appears to be reached via T2 and has IP1 as its IPv4 address,

i.e., Addr(Prf2, IP1). The source address of the packet is a T1 address with Prf1 as the prefix and IP2 as the embedded IPv4 address, i.e., Addr(Prf1, IP2). As the prefix of the destination address is Prf2, the packet will be routed over the IPv6 network to T2.

We assume that R2 is the packet's entry point to T2. R2 receives the packet through its IPv6 interface and forwards it over its T2 interface encapsulated with an IPv4 header having a destination address derived from the IPv6 destination, i.e., IP1. The source address is the address of R2, i.e., IP2. The packet (packet 1 in Figure 2.) is routed over the IPv4 network to R1, which receives the packet on its IPv4 interface. It processes the packet as a packet that originates from one of the end nodes of T1.

Since the IPv4 source address corresponds to the IPv6 source address, R1 will decapsulate the packet. Since the packet's IPv6 destination is outside of T1, R1 will forward the packet onto a native IPv6 interface. The forwarded packet (packet 2 in Figure 2) is identical to the original attack packet. Hence, it is routed back to R2, in which the loop starts again. Note that the packet may not necessarily be transported from R1 over native IPv6 network. R1 may be connected to the IPv6 network through another tunnel.



1 - IPv4: IP2 --> IP1
IPv6: Addr(Prf1,IP2) --> Addr(Prf2,IP1)
0,2- IPv6: Addr(Prf1,IP2) --> Addr(Prf2,IP1)

Legend: ====> - tunneled IPv6, ---> - native IPv6

Figure 2: Routing loop attack between two tunnels' routers

The crux of the attack is as follows. The attacker exploits the fact that R2 does not know that R1 does not take part of T2 and that R1 does not know that R2 does not take part of T1. The IPv4 network acts as a shared link layer for the two tunnels. Hence, the packet is repeatedly forwarded by both routers. It is noted that the attack will fail when the IPv4 network can not transport packets between the tunnels. For example, when the two routers belong to different IPv4 address realms or when ingress/egress filtering is exercised between

the routes.

The loop will stop when the Hop Limit field of the packet reaches zero. After a single loop the Hop Limit field is decreased by the number of IPv6 routers on path from R1 and R2. Therefore, the number of loops is inversely proportional to the number of IPv6 hops between R1 and R2.

The tunnel pair T1 and T2 may be any combination of automatic tunnel types, e.g., ISATAP, 6to4 and 6rd. This has the exception that both tunnels can not be of type 6to4, since two 6to4 routers can not belong to different tunnels (there is only one 6to4 tunnel in the Internet). For example, if the attack were to be launched on an ISATAP router (R1) and 6to4 relay (R2), then the destination and source addresses of the attack packet would be 2002:IP1:* and Prf1::0200:5EFE:IP2, respectively.

3. Proposed Mitigation Measures

This section presents some possible mitigation measures for the attack described above. For each measure we shall discuss its advantages and disadvantages.

The proposed measures fall under the following three categories:

- Verification of end point existence
- Operational measures
- Destination and source addresses checks

3.1. Verification of end point existence

The routing loop attack relies on the fact that a router does not know whether there is an end point that can reached via its tunnel that has the source or destination address of the packet. This category includes mitigation measures which aim to verify that there is a node which participate in the tunnel and its address corresponds to the packet’s destination or source addresses, as appropriate.

3.1.1. Neighbor Cache Check

One way that the router can verify that an end host exists and can be reached via the tunnel is by checking whether a valid entry exists for it in the neighbor cache of the corresponding tunnel interface. The neighbor cache entry can be populated through, e.g., an initial reachability check, receipt of neighbor discovery messages,

administrative configuration, etc.

When the router has a packet to send to a potential tunnel host for which there is no neighbor cache entry, it can perform an initial reachability check on the packet's destination address, e.g., as specified in the second paragraph of Section 8.4 of [RFC5214]. (The router can similarly perform a "reverse reachability" check on the packet's source address when it receives a packet from a potential tunnel host for which there is no neighbor cache entry.) This reachability check parallels the address resolution specifications in Section 7.2 of [RFC4861], i.e., the router maintains a small queue of packets waiting for reachability confirmation to complete. If confirmation succeeds, the router discovers that a legitimate tunnel host responds to the address. Otherwise, the router discards subsequent packets and returns ICMP destination unreachable indications as specified in Section 7.2.2 of [RFC4861].

Note that this approach assumes that the neighbor cache will remain coherent and not subject to malicious attack, which must be confirmed based on specific deployment scenarios. One possible way for an attacker to subvert the neighbor cache is to send false neighbor discovery messages with a spoofed source address.

3.1.2. Known IPv4 Address Check

Another approach that enables a router to verify that an end host exists and can be reached via the tunnel is simply by pre-configuring the router with the set of IPv4 addresses that are authorized to use the tunnel. Upon this configuration the router can perform the following simple checks:

- When the router forwards an IPv6 packet into the tunnel interface with a destination address that matches an on-link prefix and that embeds the IPv4 address IP1, it discards the packet if IP1 does not belong to the configured list of IPv4 addresses.

- When the router receives an IPv6 packet on the tunnel's interface with a source address that matches a on-link prefix and that embeds the IPv4 address IP2, it discards the packet if IP2 does not belong to the configured list of IPv4 addresses.

3.2. Operational Measures

The following measures can be taken by the network operator. Their aim is to configure the network in such a way that the attacks can not take place.

3.2.1. Avoiding a Shared IPv4 Link

As noted above, the attack relies on having an IPv4 network as a shared link-layer between more than one tunnel. From this the following two mitigation measures arise:

3.2.1.1. Filtering IPv4 Protocol-41 Packets

In this measure a tunnel router may drop all IPv4 protocol-41 packets received or sent over interfaces that are attached to an untrusted IPv4 network. This will cut-off any IPv4 network as a shared link. This measure has the advantage of simplicity. However, such a measure may not always be suitable for scenarios where IPv4 connectivity is essential on all interfaces.

3.2.1.2. Operational Avoidance of Multiple Tunnels

This measure mitigates the attack by simply allowing for a single IPv6 tunnel to operate in a bounded IPv4 network. For example, the attack can not take place in broadband home networks. In such cases there is a small home network having a single residential gateway which serves as a tunnel router. A tunnel router is vulnerable to the attack only if it has at least two interfaces with a path to the Internet: a tunnel interface and a native IPv6 interface (as depicted in Figure 1). However, a residential gateway usually has only a single interface to the Internet, therefore the attack can not take place. Moreover, if there are only one or a few tunnel routers in the IPv4 network and all participate in the same tunnel then there is no opportunity for perpetuating the loop.

This approach has the advantage that it avoids the attack profile altogether without need for explicit mitigations. However, it requires careful configuration management which may not be tenable in large and/or unbounded IPv4 networks.

3.2.2. A Single Border Router

It is reasonable to assume that a tunnel router shall accept or forward tunneled packets only over its tunnel interface. It is also reasonable to assume that a tunnel router shall accept or forward IPv6 packets only over its IPv6 interface. If these two interfaces are physically different then the network operator can mitigate the attack by ensuring that the following condition holds: there is no path between these two interfaces that does not go through the tunnel router.

The above condition ensures that an encapsulated packet which is transmitted over the tunnel interface will not get to another tunnel

router and from there to the IPv6 interface of the first router. The condition also ensures the reverse direction, i.e., an IPv6 packet which is transmitted over the IPv6 interface will not get to another tunnel router and from there to the tunnel interface of the first router. This condition is essentially translated to a scenario in which the tunnel router is the only border router between the IPv6 network and the IPv4 network to which it is attached (as in broadband home network scenario mentioned above).

3.2.3. A Comprehensive List of Tunnel Routers

If a tunnel router can be configured with a comprehensive list of IPv4 addresses of all other tunnel routers in the network, then the router can use the list as a filter to discard any tunneled packets coming from other routers. For example, a tunnel router can use the network's ISATAP Potential Router List (PRL) [RFC5214] as a filter as long as there is operational assurance that all ISATAP routers are listed and that no other types of tunnel routers are present in the network.

This measure parallels the one proposed for 6rd in [RFC5969] where the 6rd BR filters all known relay addresses of other tunnels inside the ISP's network.

This measure is especially useful for intra-site tunneling mechanisms, such as ISATAP and 6rd, since filtering can be exercised on well-defined site borders.

3.2.4. Avoidance of On-link Prefixes

The looping attack exploits the fact that a router is permitted to assign non-link-local IPv6 prefixes on its tunnel interfaces, which could cause it to send tunneled packets to other routers that do not configure an address from the prefix. Therefore, if the router does not assign non-link-local IPv6 prefixes on its tunnel interfaces there is no opportunity for it to initiate the loop. If the router further ensures that the routing state is consistent for the packets it receives on its tunnel interfaces there is no opportunity for it to propagate a loop initiated by a different router.

This mitigation is available only to ISATAP routers, since the ISATAP stateless address mapping operates only on the Interface Identifier portion of the IPv6 address, and not on the IPv6 prefix. . The mitigation is also only applicable on ISATAP links on which IPv4 source address spoofing is disabled. The following sections discuss the operational configurations necessary to implement the mitigation.

3.2.4.1. ISATAP Router Interface Types

ISATAP provides a Potential Router List (PRL) to further ensure a loop-free topology. Routers that are members of the provider network PRL configure their provider network ISATAP interfaces as advertising router interfaces (see: [RFC4861], Section 6.2.2), and therefore may send Router Advertisement (RA) messages that include non-zero Router Lifetimes. Routers that are not members of the provider network PRL configure their provider network ISATAP interfaces as non-advertising router interfaces.

3.2.4.2. ISATAP Source Address Verification

ISATAP nodes employ the source address verification checks specified in Section 7.3 of [RFC5214] as a prerequisite for decapsulation of packets received on an ISATAP interface. To enable the on-link prefix avoidance procedures outlined in this section, ISATAP nodes must employ an additional source address verification check; namely, the node also considers the outer IPv4 source address correct for the inner IPv6 source address if:

- a forwarding table entry exists that lists the packet’s IPv4 source address as the link-layer address corresponding to the inner IPv6 source address via the ISATAP interface.

3.2.4.3. ISATAP Host Behavior

ISATAP hosts send Router Solicitation (RS) messages to obtain RA messages from an advertising ISATAP router. Whether or not non-link-local IPv6 prefixes are advertised, the host can acquire IPv6 addresses, e.g., through the use of DHCPv6 stateful address autoconfiguration [RFC3315].

To acquire addresses, the host performs standard DHCPv6 exchanges while mapping the IPv6 "All_DHCP_Relay_Agents_and_Servers" link-scoped multicast address to the IPv4 address of the advertising router (hence, the advertising router must configure either a DHCPv6 relay or server function). The host should also use DHCPv6 Authentication in environments where authentication of the DHCPv6 exchanges is required.

After the host receives IPv6 addresses, it assigns them to its ISATAP interface and forwards any of its outbound IPv6 packets via the advertising router as a default router. The advertising router in turn maintains IPv6 forwarding table entries that list the IPv4 address of the host as the link-layer address of the delegated IPv6 addresses.

3.2.4.4. ISATAP Router Behavior

In many use case scenarios (e.g., enterprise networks, MANETs, etc.), advertising and non-advertising ISATAP routers can engage in a proactive dynamic IPv6 routing protocol (e.g., OSPFv3, RIPng, etc.) so that IPv6 routing/forwarding tables can be populated and standard IPv6 forwarding between ISATAP routers can be used. In other scenarios (e.g., large ISP networks, etc.), this might be impractical dues to scaling issues. When a proactive dynamic routing protocol cannot be used, non-advertising ISATAP routers send RS messages to obtain RA messages from an advertising ISATAP router, i.e., they act as "hosts" on their non-advertising ISATAP interfaces.

Non-advertising routers can also acquire IPv6 prefixes, e.g., through the use of DHCPv6 Prefix Delegation [RFC3633] via an advertising router in the same fashion as described above for host-based DHCPv6 stateful address autoconfiguration. The advertising router in turn maintains IPv6 forwarding table entries that list the IPv4 address of the non-advertising router as the link-layer address of the next hop toward the delegated IPv6 prefixes.

After the non-advertising router acquires IPv6 prefixes, it can sub-delegate them to routers and links within its attached IPv6 edge networks, then can forward any outbound IPv6 packets coming from its edge networks via other ISATAP nodes on the link.

3.2.4.5. Reference Operational Scenario

Figure 3 depicts a reference ISATAP network topology for operational avoidance of on-link non-link-local IPv6 prefixes. The scenario shows an advertising ISATAP router (’A’), two non-advertising ISATAP routers (’B’, ’D’), an ISATAP host (’F’), and three ordinary IPv6 hosts (’C’, ’E’, ’G’) in a typical deployment configuration:

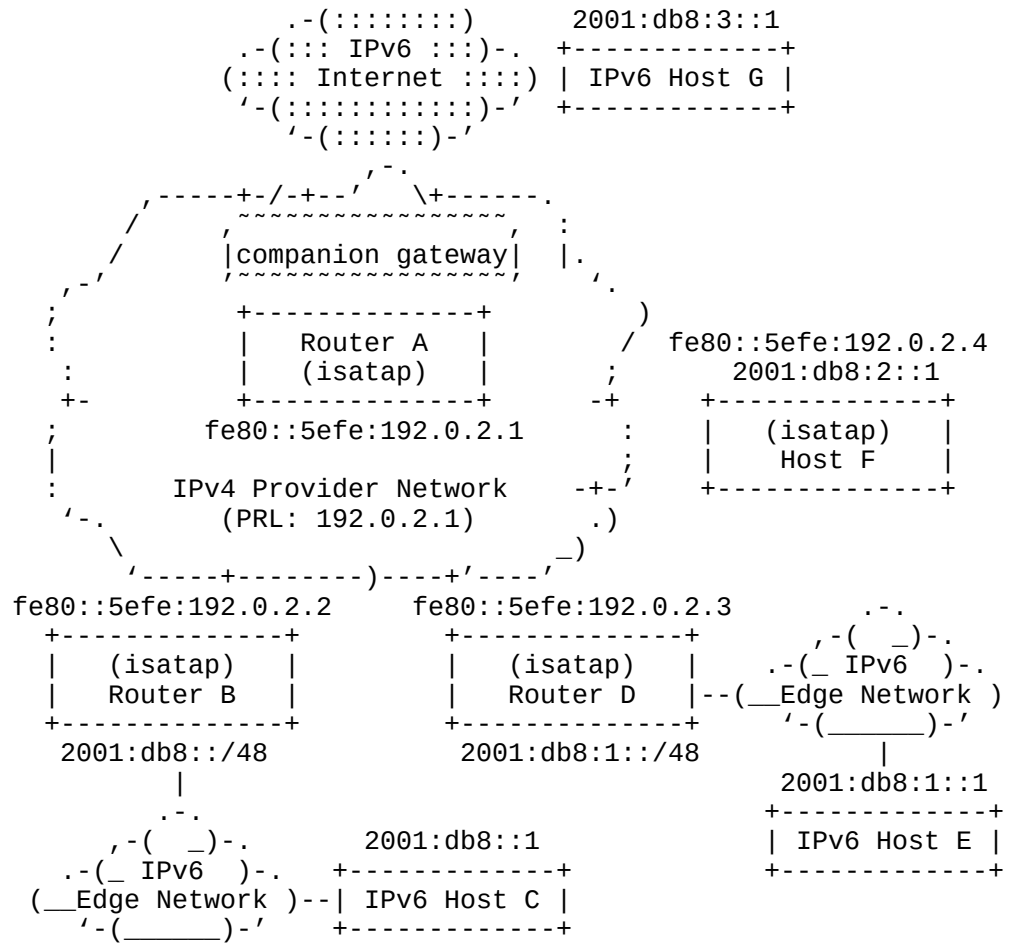


Figure 3: Reference ISATAP Network Topology

In Figure 3, advertising ISATAP router 'A' within the IPv4 provider network connects to the IPv6 Internet, either directly or via a companion gateway. 'A' configures a provider network IPv4 interface with address 192.0.2.1 and arranges to add the address to the provider network PRL. 'A' next configures an advertising ISATAP router interface with link-local IPv6 address fe80::5efe:192.0.2.1 over the IPv4 interface.

Non-advertising ISATAP router 'B' connects to one or more IPv6 edge networks and also connects to the provider network via an IPv4 interface with address 192.0.2.2, but it does not add the IPv4 address to the provider network PRL. 'B' next configures a non-advertising ISATAP router interface with link-local address fe80::

5efe:192.0.2.2, then receives the IPv6 prefix 2001:db8::/48 through a DHCPv6 prefix delegation exchange via 'A'. 'B' then engages in an IPv6 routing protocol over its ISATAP interface and announces the delegated IPv6 prefix. 'B' finally sub-delegates the prefix to its attached edge networks, where IPv6 host 'C' autoconfigures the address 2001:db8::1.

Non-advertising ISATAP router 'D' connects to the provider network, configures its ISATAP interface, receives a DHCPv6 prefix delegation, and engages in the IPv6 routing protocol the same as for router 'B'. In particular, 'D' configures the IPv4 address 192.0.2.3, the ISATAP link-local address fe80::5efe:192.0.2.3, and the delegated IPv6 prefix 2001:db8:1::/48. 'D' finally sub-delegates the prefix to its attached edge networks, where IPv6 host 'E' autoconfigures IPv6 address 2001:db8:1::1.

ISATAP host 'F' connects to the provider network via an IPv4 interface with address 192.0.2.4, and also configures an ISATAP host interface with link-local address fe80::5efe:192.0.2.4 over the IPv4 interface. 'F' next configures a default IPv6 route with next-hop address fe80::5efe:192.0.2.1 via the ISATAP interface, then receives the IPv6 address 2001:db8:2::1 from a DHCPv6 address configuration exchange via 'A'. When 'F' receives the IPv6 address, it assigns the address to the ISATAP interface but does not assign a non-link-local IPv6 prefix to the interface.

Finally, IPv6 host 'G' connects to an IPv6 network outside of the ISATAP domain. 'G' configures its IPv6 interface in a manner specific to its attached IPv6 link, and autoconfigures the IPv6 address 2001:db8:3::1.

Following this autoconfiguration, when host 'C' has an IPv6 packet to send to host 'E', it prepares the packet with source address 2001:db8::1 and destination address 2001:db8:1::1, then sends the packet into the edge network where it will eventually be forwarded to router 'B'. 'B' then uses ISATAP encapsulation to forward the packet to router 'D', since it has discovered a route to 2001:db8:1::/48 with next hop 'D' via dynamic routing over the ISATAP interface. Router 'D' finally forwards the packet to host 'E'.

In a second scenario, when 'C' has a packet to send to ISATAP host 'F', it prepares the packet with source address 2001:db8::1 and destination address 2001:db8:2::1, then sends the packet into the edge network where it will eventually be forwarded to router 'B' the same as above. 'B' then uses ISATAP encapsulation to forward the packet to router 'A' (i.e., a router that advertises "default"), which in turn forwards the packet to 'F'. Note that this operation entails two hops across the ISATAP link (i.e., one from 'B' to 'A',

and a second from 'A' to 'F'). If 'F' also participates in the dynamic IPv6 routing protocol, however, 'B' could instead forward the packet directly to 'F' without involving 'A'.

In a final scenario, when 'C' has a packet to send to host 'G' in the IPv6 Internet, the packet is forwarded to 'B' the same as above. 'B' then forwards the packet to 'A', which forwards the packet into the IPv6 Internet.

3.2.4.6. Scaling Considerations

Figure 3 depicts an ISATAP network topology with only a single advertising ISATAP router within the provider network. In order to support larger numbers of non-advertising ISATAP routers and ISATAP hosts, the provider network can deploy more advertising ISATAP routers to support load balancing and generally shortest-path routing.

Such an arrangement requires that the advertising ISATAP routers participate in an IPv6 routing protocol instance so that IPv6 address/prefix delegations can be mapped to the correct router. The routing protocol instance can be configured as either a full mesh topology involving all advertising ISATAP routers, or as a partial mesh topology with each advertising ISATAP router associating with one or more companion gateways and a full mesh between companion gateways.

3.2.4.7. On-Demand Dynamic Routing

With respect to the reference operational scenario depicted in Figure 3, there will be many use cases in which a proactive dynamic IPv6 routing protocol cannot be used. For example, in large ISP network deployments it would be impractical for all Customer-Edge and Provider-Edge routers to engage in a common routing protocol instance due to scaling considerations.

In those cases, an on-demand routing capability can be enabled in which ISATAP nodes send initial packets via an advertising ISATAP router and receive redirection messages back. For example, when a non-advertising ISATAP router 'B' has a packet to send to a host located behind non-advertising ISATAP router 'D', it can send the initial packets via advertising router 'A' which will return redirection messages to inform 'B' that 'D' is a better first hop. Protocol details for this ISATAP redirection are specified in [I-D.templin-intarea-vet].

3.3. Destination and Source Address Checks

Tunnel routers can use a source address check mitigation when they forward an IPv6 packet into a tunnel interface with an IPv6 source address that embeds one of the router's configured IPv4 addresses. Similarly, tunnel routers can use a destination address check mitigation when they receive an IPv6 packet on a tunnel interface with an IPv6 destination address that embeds one of the router's configured IPv4 addresses. These checks should correspond to both tunnels' IPv6 address formats, regardless of the type of tunnel the router employs.

For example, if tunnel router R1 (of any tunnel protocol) forwards a packet into a tunnel interface with an IPv6 source address that matches the 6to4 prefix 2002:IP1::/48, the router discards the packet if IP1 is one of its own IPv4 addresses. In a second example, if tunnel router R2 receives an IPv6 packet on a tunnel interface with an IPv6 destination address with an off-link prefix but with an interface identifier that matches the ISATAP address suffix ::0200:5EFE:IP2, the router discards the packet if IP2 is one of its own IPv4 addresses.

Hence a tunnel router can avoid the attack by performing the following checks:

- When the router forwards an IPv6 packet into a tunnel interface, it discards the packet if the IPv6 source address has an off-link prefix but embeds one of the router's configured IPv4 addresses.

- When the router receives an IPv6 packet on a tunnel interface, it discards the packet if the IPv6 destination address has an off-link prefix but embeds one of the router's configured IPv4 addresses.

This approach has the advantage that that no ancillary state is required, since checking is through static lookup in the lists of IPv4 and IPv6 addresses belonging to the router. However, this approach has some inherent limitations

- The checks incur an overhead which is proportional to the number of IPv4 addresses assigned to the router. If a router is assigned many addresses, the additional processing overhead for each packet may be considerable. Note that an unmitigated attack packet would be repetitively processed by the router until the Hop Limit expires, which may require as many as 255 iterations. Hence, an unmitigated attack will consume far more aggregate processing overhead than per-packet address checks even if the router assigns a large number of addresses.

- The checks should be performed for the IPv6 address formats of every existing automatic IPv6 tunnel protocol (which uses protocol-41 encapsulation). Hence, the checks must be updated as new protocols are defined.

- Before the checks can be performed the format of the address must be recognized. There is no guarantee that this can be generally done. For example, one can not determine if an IPv6 address is a 6rd one, hence the router would need to be configured with a list of all applicable 6rd prefixes (which may be prohibitively large) in order to unambiguously apply the checks.

- The checks cannot be performed if the embedded IPv4 address is a private one [RFC1918] since it is ambiguous in scope. Namely, the private address may be legitimately allocated to another node in another routing region.

The last limitation may be relieved if the router has some information that allows it to unambiguously determine the scope of the address. The check in the following subsection is one example for this.

3.3.1. Known IPv6 Prefix Check

A router may be configured with the full list of IPv6 subnet prefixes assigned to the tunnels attached to its current IPv4 routing region. In such a case it can use the list to determine when static destination and source address checks are possible. By keeping track of the list of IPv6 prefixes assigned to the tunnels in the IPv4 routing region, a router can perform the following checks on an address which embeds a private IPv4 address:

- When the router forwards an IPv6 packet into its tunnel with a source address that embeds a private IPv4 address and matches an IPv6 prefix in the prefix list, it determines whether the packet should be discarded or forwarded by performing the source address check specified in Section 3.3. Otherwise, the router forwards the packet.

- When the router receives an IPv6 packet on its tunnel interface with a destination address that embeds a private IPv4 address and matches an IPv6 prefix in the prefix list, it determines whether the packet should be discarded or forwarded by performing the destination address check specified in Section 3.3. Otherwise, the router forwards the packet.

The disadvantage of this approach is the administrative overhead for maintaining the list of IPv6 subnet prefixes associated with an IPv4

routing region may become unwieldy should that list be long and/or frequently updated.

4. Recommendations

In light of the mitigation measures proposed above we make the following recommendations in decreasing order:

1. When possible, it is recommended that the attacks are operationally eliminated (as per one of the measures proposed in Section 3.2).

2. For tunnel routers that keep a coherent and trusted neighbor cache which includes all legitimate end-points of the tunnel, we recommend exercising the Neighbor Cache Check.

3. For tunnel routers that can implement the Neighbor Reachability Check, we recommend exercising it.

4. For tunnels having small and static list of end-points we recommend exercising Known IPv4 Address Check.

5. We generally do not recommend using the Destination and Source Address Checks since they can not mitigate routing loops with 6rd routers. Therefore, these checks should not be used alone unless there is operational assurance that other measures are exercised to prevent routing loops with 6rd routers.

As noted earlier, tunnels may be deployed in various operational environments. There is a possibility that other mitigations may be feasible in specific deployment scenarios. The above recommendations are general and do not attempt to cover such scenarios.

5. IANA Considerations

This document has no IANA considerations.

6. Security Considerations

This document aims at presenting possible solutions to the routing loop attack which involves automatic tunnels' routers. It contains various checks that aim to recognize and drop specific packets that have strong potential to cause a routing loop. These checks do not introduce new security threats.

7. Acknowledgments

This work has benefited from discussions on the V6OPS, 6MAN and SECDIR mailing lists. Remi Despres, Christian Huitema, Dmitry Anipko, Dave Thaler and Fernando Gont are acknowledged for their contributions.

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.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

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

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

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

8.2. Informative References

[I-D.gont-6man-teredo-loops] Gont, F., "Mitigating Teredo Rooting Loop Attacks", draft-gont-6man-teredo-loops-00 (work in progress), September 2010.

[I-D.templin-intarea-vet]
Templin, F., "Virtual Enterprise Traversal (VET)", draft-templin-intarea-vet-23 (work in progress), January 2011.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4732] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[USENIX09]
Nakibly, G. and M. Arov, "Routing Loop Attacks using IPv6 Tunnels", USENIX WOOT, August 2009.

Authors' Addresses

Gabi Nakibly
National EW Research & Simulation Center
P.O. Box 2250 (630)
Haifa 31021
Israel

Email: gnakibly@yahoo.com

Fred L. Templin
Boeing Research & Technology
P.O. Box 3707 MC 7L-49
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 8, 2011

G. Nakibly
National EW Research &
Simulation Center
F. Templin
Boeing Research & Technology

May 7, 2011

Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations

draft-ietf-v6ops-tunnel-loops-07.txt

Abstract

This document is concerned with security vulnerabilities in IPv6-in-IPv4 automatic tunnels. These vulnerabilities allow an attacker to take advantage of inconsistencies between the IPv4 routing state and the IPv6 routing state. The attack forms a routing loop which can be abused as a vehicle for traffic amplification to facilitate DoS attacks. The first aim of this document is to inform on this attack and its root causes. The second aim is to present some possible mitigation measures. It should be noted that at the time of this writing there are no known reports of malicious attacks exploiting these vulnerabilities. Nonetheless, these vulnerabilities can be activated by accidental misconfiguarion.

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 November 8, 2011.

Copyright Notice

Table of Contents

1. Introduction

IPv6-in-IPv4 tunnels are an essential part of many migration plans for IPv6. They allow two IPv6 nodes to communicate over an IPv4-only network. Automatic tunnels that assign IPv6 prefixes with stateless address mapping properties (hereafter called "automatic tunnels") are a category of tunnels in which a tunneled packet's egress IPv4 address is embedded within the destination IPv6 address of the packet. An automatic tunnel's router is a router that respectively encapsulates and decapsulates the IPv6 packets into and out of the tunnel.

Ref. [USENIX09] pointed out the existence of a vulnerability in the design of IPv6 automatic tunnels. Tunnel routers operate on the implicit assumption that the destination address of an incoming IPv6 packet is always an address of a valid node that can be reached via the tunnel. The assumption of path validity can introduce routing loops as the inconsistency between the IPv4 routing state and the IPv6 routing state allows a routing loop to be formed. Although those loops will not trap normal data, they will catch traffic targeted at addresses that have become unavailable, and misconfigured traffic can enter the loop.

The looping vulnerability can be triggered accidentally or exploited maliciously by an attacker crafting a packet which is routed over a tunnel to a node that is not associated with the packet's destination. This node may forward the packet out of the tunnel to the native IPv6 network. There the packet is routed back to the ingress point that forwards it back into the tunnel. Consequently, the packet loops in and out of the tunnel. The loop terminates only when the Hop Limit field in the IPv6 header of the packet is decremented to zero. This vulnerability can be abused as a vehicle for traffic amplification to facilitate DoS attacks [RFC4732].

Without compensating security measures in place, all IPv6 automatic tunnels that are based on protocol-41 encapsulation [RFC4213] are vulnerable to such an attack including ISATAP [RFC5214], 6to4 [RFC3056] and 6rd [RFC5969]. It should be noted that this document does not consider non-protocol-41 encapsulation attacks. In particular, we do not address the Teredo [RFC4380] attacks described in [USENIX09]. These attacks are considered in [I-D.gont-6man-teredo-loops].

The aim of this document is to shed light on the routing loop attack and describe possible mitigation measures that should be considered by operators of current IPv6 automatic tunnels and by designers of future ones. We note that tunnels may be deployed in various operational environments, e.g. service provider network, enterprise

network, etc. Specific issues related to the attack which are derived from the operational environment are not considered in this document.

Routing loops pose a risk to the stability of a network. Furthermore, they provide an opening for denial of service attacks that exploit the existence of the loop to increase the traffic load in the network. Section 3 of this document discusses a number of mitigation measures. The most desirable mitigation, however, is to operate the network in such a way that routing loops can not take place (see Section 3.2).

2. A Detailed Description of the Attack

In this section we shall denote an IPv6 address of a node by an IPv6 prefix assigned to the tunnel and an IPv4 address of the tunnel end point, i.e., Addr(Prefix, IPv4). Note that the IPv4 address may or may not be part of the prefix (depending on the specification of the tunnel's protocol). The IPv6 address may be dependent on additional bits in the interface ID, however for our discussion their exact value is not important.

The two victims of this attack are routers - R1 and R2 - that service two different tunnel prefixes - Prf1 and Prf2. Both routers have the capability to forward IPv6 packets in and out of their respective tunnels. The two tunnels need not be based on the same tunnel protocol. The only condition is that the two tunnel protocols be based on protocol-41 encapsulation. The IPv4 address of R1 is IP1, while the prefix of its tunnel is Prf1. IP2 and Prf2 are the respective values for R2. We assume that IP1 and IP2 belong to the same address realm, i.e., they are either both public, or both private and belong to the same internal network. The following network diagram depicts the locations of the two routers. The numbers indicate the packets of the attack and the path they traverse as described below.

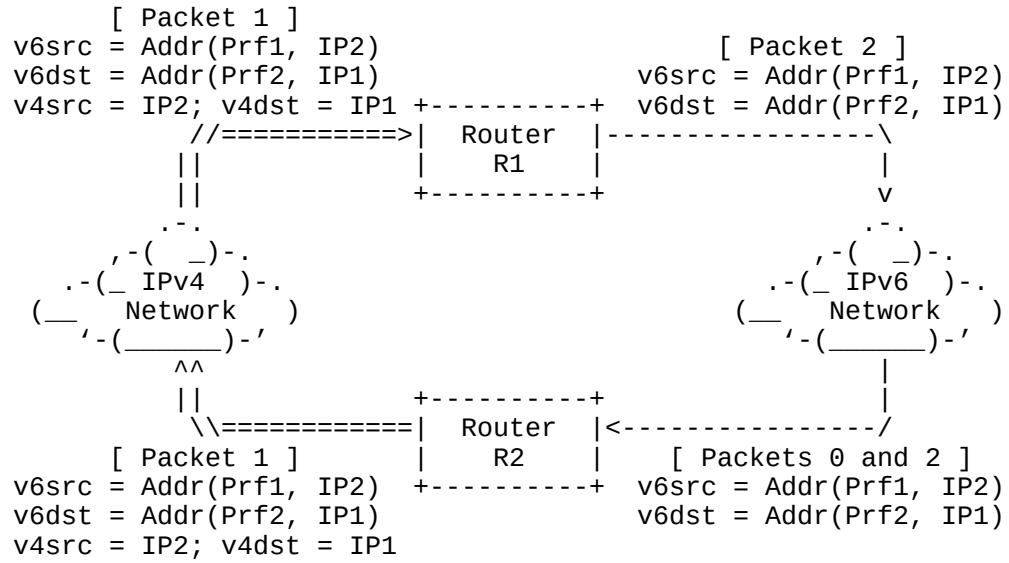


Figure 1: The network setting of the attack

The attack is depicted in Figure 2. It is initiated by an accidentally or maliciously produced IPv6 packet (packet 0 in Figure 2) destined to a fictitious end point that appears to be reached via Prf2 and has IP1 as its IPv4 address, i.e., Addr(Prf2, IP1). The source address of the packet is an address with Prf1 as the prefix and IP2 as the embedded IPv4 address, i.e., Addr(Prf1, IP2). As the prefix of the destination address is Prf2, the packet will be routed over the IPv6 network to R2.

R2 receives the packet through its IPv6 interface and forwards it into the tunnel with an IPv4 header having a destination address derived from the IPv6 destination, i.e., IP1. The source address is the address of R2, i.e., IP2. The packet (packet 1 in Figure 2) is routed over the IPv4 network to R1, which receives the packet on its IPv4 interface. It processes the packet as a packet that originates from one of the end nodes of Prf1.

Since the IPv4 source address corresponds to the IPv6 source address, R1 will decapsulate the packet. Since the packet's IPv6 destination is outside of Prf1, R1 will forward the packet onto a native IPv6 interface. The forwarded packet (packet 2 in Figure 2) is identical to the original attack packet. Hence, it is routed back to R2, in which the loop starts again. Note that the packet may not necessarily be transported from R1 over native IPv6 network. R1 may be connected to the IPv6 network through another tunnel.

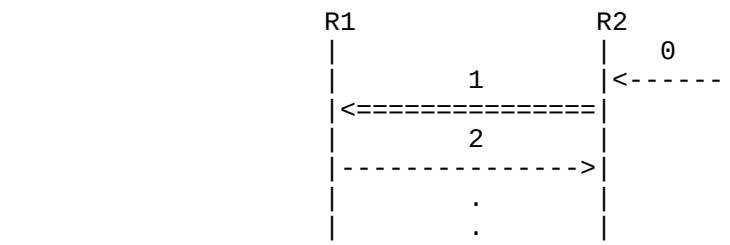


Figure 2: Routing loop attack between two tunnels' routers

The crux of the attack is as follows. The attacker exploits the fact that R2 does not know that R1 does not configure addresses from Prf2 and that R1 does not know that R2 does not configure addresses from Prf1. The IPv4 network acts as a shared link layer for the two tunnels. Hence, the packet is repeatedly forwarded by both routers. It is noted that the attack will fail when the IPv4 network can not transport packets between the tunnels. For example, when the two routers belong to different IPv4 address realms or when ingress/ egress filtering is exercised between the routers.

The loop will stop when the Hop Limit field of the packet reaches zero. After a single loop the Hop Limit field is decreased by the number of IPv6 routers on path from R1 to R2. Therefore, the number of loops is inversely proportional to the number of IPv6 hops between R1 and R2.

The tunnels used by R1 and R2 may be any combination of automatic tunnel types, e.g., ISATAP, 6to4 and 6rd. This has the exception that both tunnels can not be of type 6to4, since two 6to4 routers share the same IPv6 prefix, i.e., there is only one 6to4 prefix (2002::/16) in the Internet. For example, if the attack were to be launched on an ISATAP router (R1) and 6to4 relay (R2), then the destination and source addresses of the attack packet would be 2002: IP1:* and Prf1::0200:5EFE:IP2, respectively.

3. Proposed Mitigation Measures

This section presents some possible mitigation measures for the attack described above. For each measure we shall discuss its advantages and disadvantages.

The proposed measures fall under the following three categories:

- o Verification of end point existence
- o Operational measures
- o Destination and source addresses checks

3.1. Verification of end point existence

The routing loop attack relies on the fact that a router does not know whether there is an end point that can reached via its tunnel that has the source or destination address of the packet. This category includes mitigation measures which aim to verify that there is a node which participate in the tunnel and its address corresponds to the packet's destination or source addresses, as appropriate.

3.1.1. Neighbor Cache Check

One way that the router can verify that an end host exists and can be reached via the tunnel is by checking whether a valid entry exists for it in the neighbor cache of the corresponding tunnel interface. The neighbor cache entry can be populated through, e.g., an initial reachability check, receipt of neighbor discovery messages, administrative configuration, etc.

When the router has a packet to send to a potential tunnel host for which there is no neighbor cache entry, it can perform an initial reachability check on the packet's destination address, e.g., as specified in the second paragraph of Section 8.4 of [RFC5214]. (The router can similarly perform a "reverse reachability" check on the packet's source address when it receives a packet from a potential tunnel host for which there is no neighbor cache entry.) This reachability check parallels the address resolution specifications in Section 7.2 of [RFC4861], i.e., the router maintains a small queue of packets waiting for reachability confirmation to complete. If confirmation succeeds, the router discovers that a legitimate tunnel host responds to the address. Otherwise, the router discards subsequent packets and returns ICMP destination unreachable indications as specified in Section 7.2.2 of [RFC4861].

Note that this approach assumes that the neighbor cache will remain coherent and not subject to malicious attack, which must be confirmed based on specific deployment scenarios. One possible way for an attacker to subvert the neighbor cache is to send false neighbor discovery messages with a spoofed source address.

3.1.2. Known IPv4 Address Check

Another approach that enables a router to verify that an end host exists and can be reached via the tunnel is simply by pre-configuring the router with the set of IPv4 addresses that are authorized to use the tunnel. Upon this configuration the router can perform the following simple checks:

- When the router forwards an IPv6 packet into the tunnel interface with a destination address that matches an on-link prefix and that embeds the IPv4 address IP1, it discards the packet if IP1 does not belong to the configured list of IPv4 addresses.

- When the router receives an IPv6 packet on the tunnel's interface with a source address that matches a on-link prefix and that embeds the IPv4 address IP2, it discards the packet if IP2 does not belong to the configured list of IPv4 addresses.

3.2. Operational Measures

The following measures can be taken by the network operator. Their aim is to configure the network in such a way that the attacks can not take place.

3.2.1. Avoiding a Shared IPv4 Link

As noted above, the attack relies on having an IPv4 network as a shared link-layer between more than one tunnel. From this the following two mitigation measures arise:

3.2.1.1. Filtering IPv4 Protocol-41 Packets

In this measure a tunnel router may drop all IPv4 protocol-41 packets received or sent over interfaces that are attached to an untrusted IPv4 network. This will cut-off any IPv4 network as a shared link. This measure has the advantage of simplicity. However, such a measure may not always be suitable for scenarios where IPv4 connectivity is essential on all interfaces. Most notably, filtering of IPv4 protocol-41 packets that belong to a 6to4 tunnel can have real adverse affects on unsuspecting users [I-D.ietf-v6ops-6to4-advisory].

3.2.1.2. Operational Avoidance of Multiple Tunnels

This measure mitigates the attack by simply allowing for a single IPv6 tunnel to operate in a bounded IPv4 network. For example, the attack can not take place in broadband home networks. In such cases there is a small home network having a single residential gateway

which serves as a tunnel router. A tunnel router is vulnerable to the attack only if it has at least two interfaces with a path to the Internet: a tunnel interface and a native IPv6 interface (as depicted in Figure 1). However, a residential gateway usually has only a single interface to the Internet, therefore the attack can not take place. Moreover, if there are only one or a few tunnel routers in the IPv4 network and all participate in the same tunnel then there is no opportunity for perpetuating the loop.

This approach has the advantage that it avoids the attack profile altogether without need for explicit mitigations. However, it requires careful configuration management which may not be tenable in large and/or unbounded IPv4 networks.

3.2.2. A Single Border Router

It is reasonable to assume that a tunnel router shall accept or forward tunneled packets only over its tunnel interface. It is also reasonable to assume that a tunnel router shall accept or forward IPv6 packets only over its IPv6 interface. If these two interfaces are physically different then the network operator can mitigate the attack by ensuring that the following condition holds: there is no path between these two interfaces that does not go through the tunnel router.

The above condition ensures that an encapsulated packet which is transmitted over the tunnel interface will not get to another tunnel router and from there to the IPv6 interface of the first router. The condition also ensures the reverse direction, i.e., an IPv6 packet which is transmitted over the IPv6 interface will not get to another tunnel router and from there to the tunnel interface of the first router. This condition is essentially translated to a scenario in which the tunnel router is the only border router between the IPv6 network and the IPv4 network to which it is attached (as in broadband home network scenario mentioned above).

3.2.3. A Comprehensive List of Tunnel Routers

If a tunnel router can be configured with a comprehensive list of IPv4 addresses of all other tunnel routers in the network, then the router can use the list as a filter to discard any tunneled packets coming from or destined to other routers. For example, a tunnel router can use the network's ISATAP Potential Router List (PRL) [RFC5214] as a filter as long as there is operational assurance that all ISATAP routers are listed and that no other types of tunnel routers are present in the network.

This measure parallels the one proposed for 6rd in [RFC5969] where

the 6rd BR filters all known relay addresses of other tunnels inside the ISP's network.

This measure is especially useful for intra-site tunneling mechanisms, such as ISATAP and 6rd, since filtering can be exercised on well-defined site borders. A specific ISATAP operational scenario for which this mitigation applies is described in Section 3 of [I-D.templin-v6ops-isops].

3.2.4. Avoidance of On-link Prefixes

The looping attack exploits the fact that a router is permitted to assign non-link-local IPv6 prefixes on its tunnel interfaces, which could cause it to send tunneled packets to other routers that do not configure an address from the prefix. Therefore, if the router does not assign non-link-local IPv6 prefixes on its tunnel interfaces there is no opportunity for it to initiate the loop. If the router further ensures that the routing state is consistent for the packets it receives on its tunnel interfaces there is no opportunity for it to propagate a loop initiated by a different router.

This mitigation is available only to ISATAP routers, since the ISATAP stateless address mapping operates only on the Interface Identifier portion of the IPv6 address, and not on the IPv6 prefix. The mitigation is also only applicable on ISATAP links on which IPv4 source address spoofing is disabled. The following sections discuss the operational configurations necessary to implement the mitigation.

3.2.4.1. ISATAP Router Interface Types

ISATAP provides a Potential Router List (PRL) to further ensure a loop-free topology. Routers that are members of the PRL for the site configure their site-facing ISATAP interfaces as advertising router interfaces (see: [RFC4861], Section 6.2.2), and therefore may send RA messages that include non-zero Router Lifetimes. Routers that are not members of the PRL for the site configure their site-facing ISATAP interfaces as non-advertising router interfaces.

3.2.4.2. ISATAP Source Address Verification

ISATAP nodes employ the source address verification checks specified in Section 7.3 of [RFC5214] as a prerequisite for decapsulation of packets received on an ISATAP interface. To enable the on-link prefix avoidance procedures outlined in this section, ISATAP nodes must employ an additional source address verification check; namely, the node also considers the outer IPv4 source address correct for the inner IPv6 source address if:

- o a forwarding table entry exists that lists the packet's IPv4 source address as the link-layer address corresponding to the inner IPv6 source address via the ISATAP interface.

3.2.4.3. ISATAP Host Behavior

ISATAP hosts send RS messages to obtain RA messages from an advertising ISATAP router. Whether or not non-link-local IPv6 prefixes are advertised, the host can acquire IPv6 addresses, e.g., through the use of DHCPv6 stateful address autoconfiguration [RFC3315].

To acquire addresses, the host performs standard DHCPv6 exchanges while mapping the IPv6 "All_DHCP_Relay_Agents_and_Servers" link-scoped multicast address to the IPv4 address of the advertising router (hence, the advertising router must configure either a DHCPv6 relay or server function). The host should also use DHCPv6 Authentication in environments where authentication of the DHCPv6 exchanges is required.

After the host receives IPv6 addresses, it assigns them to its ISATAP interface and forwards any of its outbound IPv6 packets via the advertising router as a default router. The advertising router in turn maintains IPv6 forwarding table entries that list the IPv4 address of the host as the link-layer address of the delegated IPv6 addresses.

3.2.4.4. ISATAP Router Behavior

In many use case scenarios (e.g., enterprise networks, MANETs, etc.), advertising and non-advertising ISATAP routers can engage in a proactive dynamic IPv6 routing protocol (e.g., OSPFv3, RIPng, etc.) over their ISATAP interfaces so that IPv6 routing/forwarding tables can be populated and standard IPv6 forwarding between ISATAP routers can be used. In other scenarios (e.g., large enterprise networks, etc.), this might be impractical due to scaling issues. When a proactive dynamic routing protocol cannot be used, non-advertising ISATAP routers send RS messages to obtain RA messages from an advertising ISATAP router, i.e., they act as "hosts" on their non-advertising ISATAP interfaces.

Non-advertising ISATAP routers can also acquire IPv6 prefixes, e.g., through the use of DHCPv6 Prefix Delegation [RFC3633] via an advertising router in the same fashion as described above for host-based DHCPv6 stateful address autoconfiguration. The advertising router in turn maintains IPv6 forwarding table entries that list the IPv4 address of the non-advertising router as the link-layer address of the next hop toward the delegated IPv6 prefixes.

After the non-advertising router acquires IPv6 prefixes, it can sub-delegate them to routers and links within its attached IPv6 edge networks, then can forward any outbound IPv6 packets coming from its edge networks via other ISATAP nodes on the link.

3.2.4.5. Reference Operational Scenario

Figure 3 depicts a reference ISATAP network topology for operational avoidance of on-link non-link-local IPv6 prefixes. The scenario shows two advertising ISATAP routers ('A', 'B'), two non-advertising ISATAP routers ('C', 'E'), an ISATAP host ('G'), and three ordinary IPv6 hosts ('D', 'F', 'H') in a typical deployment configuration:

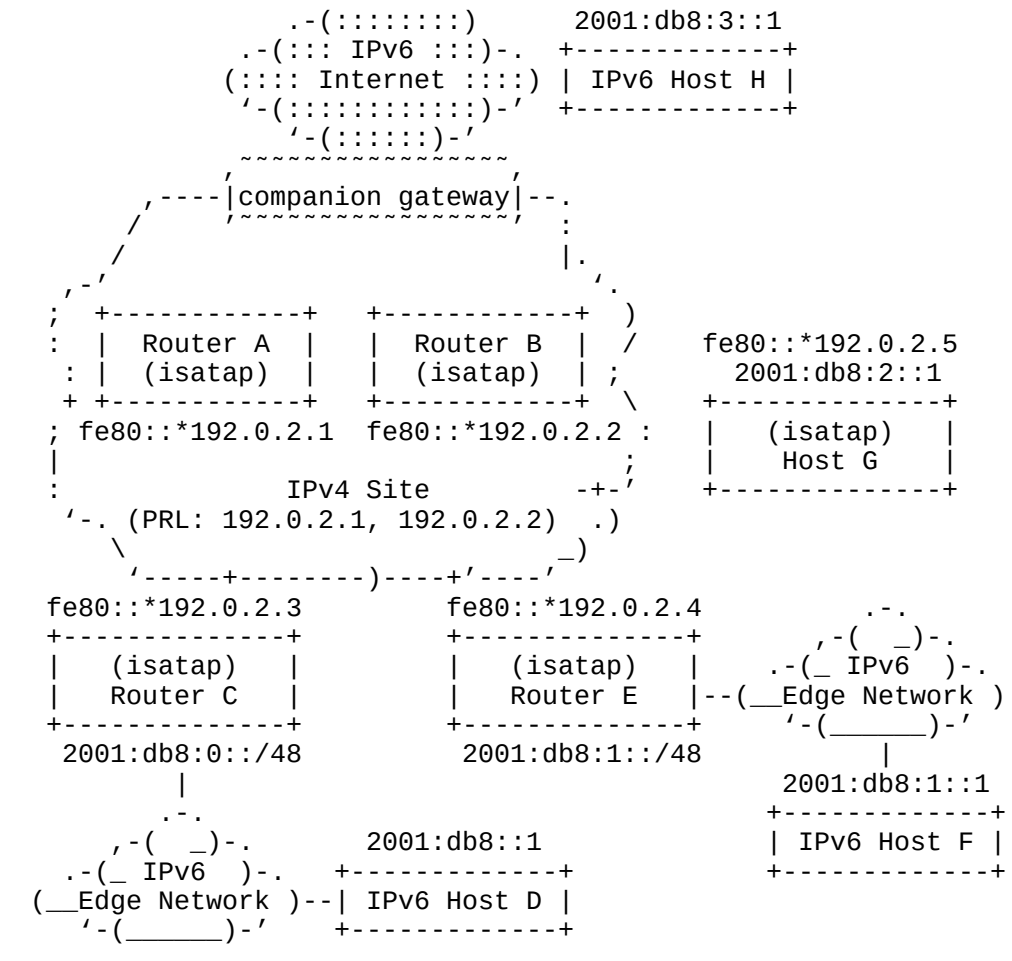


Figure 3: Reference ISATAP Network Topology

In Figure 3, advertising ISATAP routers 'A' and 'B' within the IPv4 site connect to the IPv6 Internet, either directly or via a companion gateway. 'A' configures a provider network IPv4 interface with address 192.0.2.1 and arranges to add the address to the provider network PRL. 'A' next configures an advertising ISATAP router interface with link-local IPv6 address fe80::5efe:192.0.2.1 over the IPv4 interface. In the same fashion, 'B' configures the IPv4 interface address 192.0.2.2, adds the address to the PRL, then configures the IPv6 ISATAP interface link-local address fe80::5efe: 192.0.2.2.

Non-advertising ISATAP router 'C' connects to one or more IPv6 edge networks and also connects to the site via an IPv4 interface with address 192.0.2.3, but it does not add the IPv4 address to the site's PRL. 'C' next configures a non-advertising ISATAP router interface with link-local address fe80::5efe:192.0.2.3, then receives the IPv6 prefix 2001:db8::/48 through a DHCPv6 prefix delegation exchange via one of 'A' or 'B'. 'C' then engages in an IPv6 routing protocol over its ISATAP interface and announces the delegated IPv6 prefix. 'C' finally sub-delegates the prefix to its attached edge networks, where IPv6 host 'D' autoconfigures the address 2001:db8::1.

Non-advertising ISATAP router 'E' connects to the site, configures its ISATAP interface, receives a DHCPv6 prefix delegation, and engages in the IPv6 routing protocol the same as for router 'C'. In particular, 'E' configures the IPv4 address 192.0.2.4, the ISATAP link-local address fe80::5efe:192.0.2.4, and the delegated IPv6 prefix 2001:db8:1::/48. 'E' finally sub-delegates the prefix to its attached edge networks, where IPv6 host 'F' autoconfigures IPv6 address 2001:db8:1::1.

ISATAP host 'G' connects to the site via an IPv4 interface with address 192.0.2.5, and also configures an ISATAP host interface with link-local address fe80::5efe:192.0.2.5 over the IPv4 interface. 'G' next configures a default IPv6 route with next-hop address fe80:: 5efe:192.0.2.2 via the ISATAP interface, then receives the IPv6 address 2001:db8:2::1 from a DHCPv6 address configuration exchange via 'B'. When 'G' receives the IPv6 address, it assigns the address to the ISATAP interface but does not assign a non-link-local IPv6 prefix to the interface.

Finally, IPv6 host 'H' connects to an IPv6 network outside of the ISATAP domain. 'H' configures its IPv6 interface in a manner specific to its attached IPv6 link, and autoconfigures the IPv6 address 2001:db8:3::1.

Following this autoconfiguration, when host 'D' has an IPv6 packet to send to host 'F', it prepares the packet with source address 2001:db8::1 and destination address 2001:db8:1::1, then sends the packet into the edge network where it will eventually be forwarded to router 'C'. 'C' then uses ISATAP encapsulation to forward the packet to router 'E', since it has discovered a route to 2001:db8:1::/48 with next hop 'E' via dynamic routing over the ISATAP interface. Router 'E' finally forwards the packet to host 'F'.

In a second scenario, when 'D' has a packet to send to ISATAP host 'G', it prepares the packet with source address 2001:db8::1 and destination address 2001:db8:2::1, then sends the packet into the edge network where it will eventually be forwarded to router 'C' the same as above. 'C' then uses ISATAP encapsulation to forward the packet to router 'A' (i.e., a router that advertises "default"), which in turn forwards the packet to 'G'. Note that this operation entails two hops across the ISATAP link (i.e., one from 'C' to 'A', and a second from 'A' to 'G'). If 'G' also participates in the dynamic IPv6 routing protocol, however, 'C' could instead forward the packet directly to 'G' without involving 'A'.

In a third scenario, when 'D' has a packet to send to host 'H' in the IPv6 Internet, the packet is forwarded to 'C' the same as above. 'C' then forwards the packet to 'A', which forwards the packet into the IPv6 Internet.

In a final scenario, when 'G' has a packet to send to host 'H' in the IPv6 Internet, the packet is forwarded directly to 'B', which forwards the packet into the IPv6 Internet.

3.2.4.6. Scaling Considerations

Figure 3 depicts an ISATAP network topology with only two advertising ISATAP routers within the provider network. In order to support larger numbers of non-advertising ISATAP routers and ISATAP hosts, the provider network can deploy more advertising ISATAP routers to support load balancing and generally shortest-path routing.

Such an arrangement requires that the advertising ISATAP routers participate in an IPv6 routing protocol instance so that IPv6 address/prefix delegations can be mapped to the correct router. The routing protocol instance can be configured as either a full mesh topology involving all advertising ISATAP routers, or as a partial mesh topology with each advertising ISATAP router associating with one or more companion gateways. Each such companion gateway would in turn participate in a full mesh between all companion gateways.

3.2.4.7. On-Demand Dynamic Routing

With respect to the reference operational scenario depicted in Figure 3, there will be many use cases in which a proactive dynamic IPv6 routing protocol cannot be used. For example, in large enterprise network deployments it would be impractical for all routers to engage in a common routing protocol instance due to scaling considerations.

In those cases, an on-demand routing capability can be enabled in which ISATAP nodes send initial packets via an advertising ISATAP router and receive redirection messages back. For example, when a non-advertising ISATAP router 'B' has a packet to send to a host located behind non-advertising ISATAP router 'D', it can send the initial packets via advertising router 'A' which will return redirection messages to inform 'B' that 'D' is a better first hop. Protocol details for this ISATAP redirection are specified in [I-D.templin-aero].

3.3. Destination and Source Address Checks

Tunnel routers can use a source address check mitigation when they forward an IPv6 packet into a tunnel interface with an IPv6 source address that embeds one of the router's configured IPv4 addresses. Similarly, tunnel routers can use a destination address check mitigation when they receive an IPv6 packet on a tunnel interface with an IPv6 destination address that embeds one of the router's configured IPv4 addresses. These checks should correspond to both tunnels' IPv6 address formats, regardless of the type of tunnel the router employs.

For example, if tunnel router R1 (of any tunnel protocol) forwards a packet into a tunnel interface with an IPv6 source address that matches the 6to4 prefix 2002:IP1::/48, the router discards the packet if IP1 is one of its own IPv4 addresses. In a second example, if tunnel router R2 receives an IPv6 packet on a tunnel interface with an IPv6 destination address with an off-link prefix but with an interface identifier that matches the ISATAP address suffix ::0200:5EFE:IP2, the router discards the packet if IP2 is one of its own IPv4 addresses.

Hence a tunnel router can avoid the attack by performing the following checks:

- When the router forwards an IPv6 packet into a tunnel interface, it discards the packet if the IPv6 source address has an off-link prefix but embeds one of the router's configured IPv4 addresses.

- When the router receives an IPv6 packet on a tunnel interface, it discards the packet if the IPv6 destination address has an off-link prefix but embeds one of the router's configured IPv4 addresses.

This approach has the advantage that no ancillary state is required, since checking is through static lookup in the lists of IPv4 and IPv6 addresses belonging to the router. However, this approach has some inherent limitations:

- The checks incur an overhead which is proportional to the number of IPv4 addresses assigned to the router. If a router is assigned many addresses, the additional processing overhead for each packet may be considerable. Note that an unmitigated attack packet would be repetitively processed by the router until the Hop Limit expires, which may require as many as 255 iterations. Hence, an unmitigated attack will consume far more aggregate processing overhead than per-packet address checks even if the router assigns a large number of addresses.

- The checks should be performed for the IPv6 address formats of every existing automatic IPv6 tunnel protocol (which uses protocol-41 encapsulation). Hence, the checks must be updated as new protocols are defined.

- Before the checks can be performed the format of the address must be recognized. There is no guarantee that this can be generally done. For example, one can not determine if an IPv6 address is a 6rd one, hence the router would need to be configured with a list of all applicable 6rd prefixes (which may be prohibitively large) in order to unambiguously apply the checks.

- The checks cannot be performed if the embedded IPv4 address is a private one [RFC1918] since it is ambiguous in scope. Namely, the private address may be legitimately allocated to another node in another routing region.

The last limitation may be relieved if the router has some information that allows it to unambiguously determine the scope of the address. The check in the following subsection is one example for this.

3.3.1. Known IPv6 Prefix Check

A router may be configured with the full list of IPv6 subnet prefixes assigned to the tunnels attached to its current IPv4 routing region. In such a case it can use the list to determine when static destination and source address checks are possible. By keeping track

of the list of IPv6 prefixes assigned to the tunnels in the IPv4 routing region, a router can perform the following checks on an address which embeds a private IPv4 address:

- When the router forwards an IPv6 packet into its tunnel with a source address that embeds a private IPv4 address and matches an IPv6 prefix in the prefix list, it determines whether the packet should be discarded or forwarded by performing the source address check specified in Section 3.3. Otherwise, the router forwards the packet.

- When the router receives an IPv6 packet on its tunnel interface with a destination address that embeds a private IPv4 address and matches an IPv6 prefix in the prefix list, it determines whether the packet should be discarded or forwarded by performing the destination address check specified in Section 3.3. Otherwise, the router forwards the packet.

The disadvantage of this approach is the administrative overhead for maintaining the list of IPv6 subnet prefixes associated with an IPv4 routing region may become unwieldy should that list be long and/or frequently updated.

4. Recommendations

In light of the mitigation measures proposed above we make the following recommendations in decreasing order:

1. When possible, it is recommended that the attacks are operationally eliminated (as per one of the measures proposed in Section 3.2).

2. For tunnel routers that keep a coherent and trusted neighbor cache which includes all legitimate end-points of the tunnel, we recommend exercising the Neighbor Cache Check.

3. For tunnel routers that can implement the Neighbor Reachability Check, we recommend exercising it.

4. For tunnels having small and static list of end-points we recommend exercising Known IPv4 Address Check.

5. We generally do not recommend using the Destination and Source Address Checks since they can not mitigate routing loops with 6rd routers. Therefore, these checks should not be used alone unless there is operational assurance that other measures are exercised to prevent routing loops with 6rd routers.

As noted earlier, tunnels may be deployed in various operational environments. There is a possibility that other mitigations may be feasible in specific deployment scenarios. The above recommendations are general and do not attempt to cover such scenarios.

5. IANA Considerations

This document has no IANA considerations.

6. Security Considerations

This document aims at presenting possible solutions to the routing loop attack which involves automatic tunnels' routers. It contains various checks that aim to recognize and drop specific packets that have strong potential to cause a routing loop. These checks do not introduce new security threats.

7. Acknowledgments

This work has benefited from discussions on the V6OPS, 6MAN and SECDIR mailing lists. The document has further benefitted from comments received from members of the IESG during their review. Dmitry Anipko, Fred Baker, Stewart Bryant, Remi Despres, Adrian Farrell, Fernando Gont, Christian Huitema, Joel Jaeggli, and Dave Thaler are acknowledged for their contributions.

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.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

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

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

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

8.2. Informative References

[I-D.gont-6man-teredo-loops]
Gont, F., "Mitigating Teredo Rooting Loop Attacks", draft-gont-6man-teredo-loops-00 (work in progress), September 2010.

[I-D.ietf-v6ops-6to4-advisory]
Carpenter, B., "Advisory Guidelines for 6to4 Deployment", draft-ietf-v6ops-6to4-advisory-01 (work in progress), April 2011.

[I-D.templin-aero]
Templin, F., "Asymmetric Extended Route Optimization (AERO)", draft-templin-aero-00 (work in progress), March 2011.

[I-D.templin-v6ops-isops]
Templin, F., "Operational Guidance for IPv6 Deployment in IPv4 Sites using ISATAP", draft-templin-v6ops-isops-00 (work in progress), May 2011.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4732] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[USENIX09]
Nakibly, G. and M. Arov, "Routing Loop Attacks using IPv6 Tunnels", USENIX WOOT, August 2009.

Authors' Addresses

Gabi Nakibly
National EW Research & Simulation Center
P.O. Box 2250 (630)
Haifa 31021
Israel

Email: gnakibly@yahoo.com

Fred L. Templin
Boeing Research & Technology
P.O. Box 3707 MC 7L-49
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 8, 2011

J. Brzozowski
Comcast Cable Communications

March 7, 2011

Comcast IPv6 Experiences
draft-jjmb-v6ops-comcast-ipv6-experiences-00

Abstract

This document outlines the various technologies Comcast has trialed as part of the company's ongoing IPv6 initiatives. The focus here are the technologies and experiences specific to enabling IPv6 for subscriber services like high speed data or Internet. Comcast has learned a great deal about various technologies that we feel are important to share with the community.

Requirements Language

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

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 September 8, 2011.

Copyright Notice

Table of Contents

1. Introduction

Beginning in early 2010 Comcast announced plans to leverage the work the company has been doing related to IPv6 to conduct a number of IPv6 technology trials. These trials were specifically aimed at enabling IPv6 for subscriber services. The purpose of this document is to outline the technologies that have been trialed thus far along with experiences and observations that adopters of the same may find valuable in their own planning and deployment processes.

Further, there may be some additional feedback that the various groups within the IETF may wish to take into account as part of ongoing standards efforts.

2. 6to4

During production deployment planning the widespread use of 6to4 [RFC3068] to access content and services over IPv6 was assessed. In some scenarios 6to4 usage increased several hundred times. At the time Comcast had not deployed its own 6to4 relay infrastructure as such open relays being operated by independent third parties were by default used to facilitate 6to4-based communications. The deployment and default use of open 6to4 relays appears to be a key variable behind the sub-optimal performance associated with the of 6to4. Operators that have not deployed IPv6 or have IPv6 incapable infrastructures should note that the use of 6to4 is likely occuring today across their infrastructure. Many operating systems and home networking devices continue to support the same and in some cases have 6to4 and other transition technologies enabled by default.

As a community there appears to be some consensus that long term the use of 6to4 is not desirable, however, in the near term is it clear that 6to4 will be used in specific scenarios. The expectation and goal is to see 6to4 usage diminish over time until use of the same is displaced by an alternate technique to access content and services over IPv6. While the debate continues over how and when to deprecate 6to4, it is clear that 6to4 should not be recommended as a primary mechanism to access content and services over IPv6.

[@todo - pointers to active documents pertaining to deprecating 6to4 and other transition technologies]

As part of Comcast’s IPv6 deployment a series of five (5) 6to4 relays were planned for deployment in a geographically dispersed configuration. The purpose of these relays was to reduce the latency typically associated with 6to4 usage. The use of off network, open 6to4 relays was analyzed and determined to yield nearly unusable

conditions depending on the geographic location of the end user relative to the open 6to4 relay. By deploying on network 6to4 relays latency in most cases was reduced by over 50%, which instantly yielded considerable improvements from an end user point of view. To be clear the objective behind deploying 6to4 relays was simply to reduce latency and improve the end user experience. Additionally, it is important to note that deploying the infrastructure required to support 6to4 was very straightforward and immediately noticeable from an end user point of view.

[@todo - additional deployment details and diagrams will be added to this section]

3. 6RD

6RD [draft-townsley-ipv6-6rd-01] is another transition technology similar to 6to4 that Comcast has deployed as part of technology trials. While 6RD shared many similarities with 6to4 technologically there were a number of differences noted with the same that adopters of the same should consider as part of their own deployments.

As advertised 6RD frees adopters of the same from many restrictions typically associated with 6to4 namely the use of anycast addressing (IPv4 and IPv6) and the infrastructure, like 6to4, is straightforward to deploy. However, at the time of deployment it was observed that a limited number of border relay (BR) implementations were available. This appears to be an evolving area with more implementations becoming available. Similarly it was observed that there we few if any customer edge (CE) implementations available to support a trial of the technology. As such engineering implementations were leveraged to evaluate 6RD. Further, there were no implementations available that supported the 6RD DHCPv4 options [draft-ietf-softwire-ipv6-6rd-03] as such every 6RD CE used for trial was manually configured with the necessary configuration required to enable 6RD. In order to support a wide scale production deployment leveraging 6RD an operator would have to ensure their DHCP infrastructure supports the required 6RD DHCPv4 options along with targeted 6RD CE devices.

Trial configurations included two (2) 6RD BRs which were intentionally deployed in part of the country. An anycast design was used to enable 6RD with a well known IPv4 anycast address and FQDN for the 6RD BR. The use of the same eased configuration and deployment. Additionally, an IPv6 /32 was used to support the 6RD trials as such subscriber devices were only able to yield a usable an IPv6 /64 on the LAN side of the 6RD CE.

The quantity and location of the 6RD BRs is a key variable when planning the deployment of 6RD. Comcast specifically deployed a limited quantity of the same resulting in some end users being "closer" to the BRs that others. Proximity to the 6RD BRs is an important factor in end user experience. While 6RD yields some improvements over 6to4, 6RD is ultimately a tunneling technology there proximity to the relay, in this case border relay, is an important variable.

Placement and quantity of 6RD BRs is also a significant variable to consider when assessing impacts to IPv6 geo-location information. A centralized approach to deploying 6RD BRs will yield undesirable impacts to IPv6 geo-location in that end users leveraging a particular 6RD BR that is geographically distant will not accurately represent the true origin of the end user request. Conversely, deploying 6RD BRs that are near to end users may require a substantial quantity of 6RD BRs depending on the operator network.

[@todo - add trial details and diagrams]

4. Native Dual Stack

Native dual stack is central to Comcast’s IPv6 program for trial and production deployment. Native dual stack is the model where IPv4 services remain as-is with native IPv6 support added or introduced in parallel or simultaneously. Many of the details surrounding how this is achieved are documented as part of the Cablelabs Data Over Cable Service Interface Specification 3.0 [DOCSIS3.0]. However, relevant trial and deployment specific information that is of interest to the IETF community will be documented.

[@todo - add reference to DOCSIS]

Native dual stack trials depend on the upgrade and enablement of Cable Modem Termination Systems [CMTS]. A CMTS is a device that end users in a cable network connect directly to using their cable modem [CM]. As with IPv4, native support for IPv6 is critical for the delivery of services to end users in a DOCSIS network. Anything less could yield an undesirable end user experience or instability in the operator network that could adversely impact larger populations of users.

Given the CMTS requirements native dual stack trials have initially been limited to specific areas of the network. Further, where CMTS platforms have been upgraded and enabled to support IPv6 end users have been incrementally enabled with support for IPv6. Again this is to ensure a controlled introduction with a specific focus on

maintaining stability. Initially, a limited combination of cable modem and IGD devices were used to support trial activities. Overtime diversity for both cable modem and IGDs are expected. To date a number of cable modems support the ability to enable native dual stack connectivity to CPEs devices. A subset of DOCSIS 2.0 and all DOCSIS 3.0 devices support this capability. The population of DOCSIS devices that support these capabilities varies from operator to operator.

Trial enablement requires the stateful provisioning of an IGD using stateful DHCPv6 [RFC3315] for the IGD WAN interface and delegated prefixes [RFC3633] for LAN side connectivity. The quantity of devices supporting a native dual stack mode of operation is growing. While some devices are upgradable to support native dual stack many devices deployed today are not upgradable to support this functionality. Early implementations of devices or devices that are upgradable to support native IPv6 were found to only require an IPv6 /64 for LAN side connectivity. This has been an acceptable mode of operation, however, over time IGDs will be required to support more advanced functionality including the ability to support multiple, routed IPv6 LANs. While support for a single IPv6 /64 is in place today support for shorter IPv6 prefixes is also supported. It is important for operators to ensure they design and plan support across their infrastructures for delegated prefixes that are shorter than /64.

[@todo - add information about in home consumer devices and rDNS]

[@todo - add trial details and diagrams]

5. Conclusion

[@todo - to be completed]

6. IANA Considerations

This document makes no request of IANA.

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

7. Security Considerations

There are no security considerations at this time.

8. Acknowledgements

Thanks to the Comcast team supporting the various trial and production deployment activities. A list will be supplied in a future version of this draft.

9. Normative References

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

Author's Address

John Jason Brzozowski
Comcast Cable Communications
Philadelphia, PA
USA

Phone: +1-484-962-0060
Email: john_brzozowski@cable.comcast.com

IPv6 Operations
Internet-Draft
Intended status: Informational
Expires: April 4, 2012

J. Brzozowski
C. Griffiths
Comcast

October 2, 2011

Comcast IPv6 Trial/Deployment Experiences
draft-jjmb-v6ops-comcast-ipv6-experiences-02

Abstract

This document outlines the various technologies Comcast has trialed as part of the company's ongoing IPv6 initiatives. The focus here are the technologies and experiences specific to enabling IPv6 for subscriber services like high speed data or Internet. Comcast has learned a great deal about various technologies that we feel are important to share with the community.

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 April 4, 2012.

Copyright Notice

Table of Contents

1. Requirements Language

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

2. Introduction

Beginning in early 2010 Comcast announced plans to leverage the work the company has been doing related to IPv6 to conduct a number of IPv6 technology trials. These trials were specifically aimed at enabling IPv6 for subscriber services. The purpose of this document is to outline the technologies that have been trialed thus far along with experiences and observations that adopters of the same may find valuable in their own planning and deployment processes.

Further, there may be some additional feedback that the various groups within the IETF may wish to take into account as part of ongoing standards efforts.

3. 6to4

During production deployment planning the widespread use of 6to4 [RFC3068] to access content and services over IPv6 was assessed. In some scenarios 6to4 usage increased several hundred times. At the time Comcast had not deployed its own 6to4 relay infrastructure as such open relays being operated by independent third parties were by default used to facilitate 6to4-based communications. The deployment and default use of open 6to4 relays appears to be a key variable behind the sub-optimal performance associated with the use of 6to4. An important thing to note is that some home gateway vendors have turned on 6to4 by default, and in some of these implementations, they have not presented a user interface a user interface to disable it. For operators that have not deployed IPv6 or have IPv6 incapable infrastructures should note that the use of 6to4 is likely occurring today across their infrastructure. Many operating systems and home networking devices continue to support 6to4 and in some cases have 6to4 and other transition technologies enabled by default.

As a community there appears to be some consensus that long term the use of 6to4 is not desirable, however, in the near term it is clear that 6to4 will be used in specific scenarios. The expectation and goal is to see 6to4 usage diminish over time until use of the same is displaced by an alternate technique to access content and services over IPv6. While the debate continues over how and when to deprecate 6to4, it is clear that 6to4 should not be recommended as a primary

mechanism to access content and services over IPv6.

The following documents outline the recommendations surrounding the use and status of 6to4 from a standards point of view:

1. [draft-ietf-v6ops-6to4-advisory]

2. [draft-ietf-v6ops-6to4-to-historic]

Comcast deployed a series of five (5) 6to4 relays in a geographically dispersed configuration across our network. The purpose of these relays was to reduce the latency typically associated with 6to4 usage. During our analysis, the use of off network, open 6to4 relays was determined to yield nearly unusable conditions depending on the geographic location of the end user relative to the open 6to4 relay. By deploying on-network 6to4 relays, latency in most cases was reduced by over 50%, which instantly yielded considerable improvements from an end user point of view. The simplistic design and deployment of these relays enabled us to rapidly put them in network, and in some cases create a better experience for some of our users who had 6to4 enabled.

Through the use of commodity x86 based servers that run a standard Linux Operating System, we reduced deployment and operating costs, while still maintaining a fault tolerant design. Each 6to4 relay was dual stacked, and with a simple kernel module, we enabled the 6to4 configuration. Some 6to4 specific configurations were required to ensure compatibility across a wide range of end points. The logic to anycast the 6to4 records was handled by the network infrastructure providing connectivity to the 6to4 relays, and health checking enabled us to automatically remove the route for any relay from the routing table in case of failure.

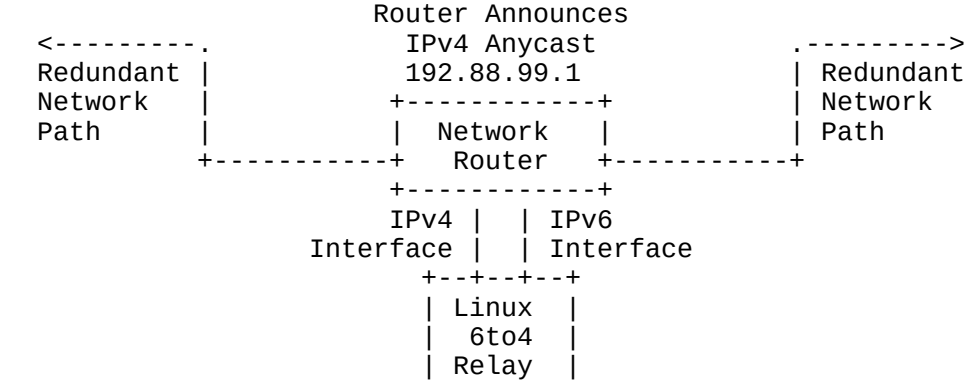


Figure 1: Comcast 6to4 Data Center View

4. 6RD

6RD [draft-townsley-ipv6-6rd] is another transition technology similar to 6to4 that Comcast has deployed as part of technology trials. While 6RD yields some improvements over 6to4, 6RD is ultimately a tunneling technology. As such, it is subject to the challenges faced by other tunneling technologies.

As advertised, 6RD frees adopters from some restrictions typically associated with 6to4. The use of anycast addressing (IPv4 and IPv6) is no longer required and the infrastructure, like 6to4, is straightforward to deploy. However, at the time of deployment it was observed that a limited number of border relay (BR) implementations were available. This appears to be an evolving area with more implementations becoming available. Similarly it was observed that there we few if any customer edge (CE) implementations available to support a trial of the technology. As such engineering implementations were leveraged to evaluate 6RD. Further, there were no implementations available that supported the 6RD DHCPv4 options [draft-ietf-softwire-ipv6-6rd]. Because of this, every 6RD CE used for trial was manually configured with the necessary information required to enable 6RD. In order to support a wide scale production deployment leveraging 6RD an operator would have to ensure their DHCP infrastructure supports the required 6RD DHCPv4 options along with targeted 6RD CE devices.

Trial configurations included two (2) 6RD BRs, which were intentionally deployed in geographically dispersed configuration. An anycast design was used to enable 6RD with a well known IPv4 anycast address and FQDN for the 6RD BR. The use of anycast eased manual configuration and deployment. Additionally, an IPv6 /32 was used to support the 6RD trials permitting subscriber devices were able to yield a usable IPv6 /64 on the LAN side of the 6RD CE.

The quantity and location of the 6RD BRs is a key variable when planning the deployment of 6RD. Comcast specifically deployed a limited quantity of BRs resulting in some end users being "closer" to the BRs than others. Proximity to the 6RD BRs is an important factor that impacts the end user experience. While 6RD yields some improvements over 6to4, 6RD is ultimately a tunneling technology as

such use of the same is subject to the challenges faced by other tunneling technologies.

Placement and quantity of 6RD BRs is also a significant variable to consider when assessing impacts to performance and IPv6 geo-location. A centralized approach to deploying 6RD BRs will yield undesirable impacts to IPv6 geo-location in that end users leveraging a particular 6RD BR that is geographically distant from their true location will not accurately represent the origin of the end user request. Conversely, deploying 6RD BRs that are near to end users may require a substantial quantity of 6RD BRs depending on the operator network.

The following provides an overview of the Comcast 6RD trial network design:

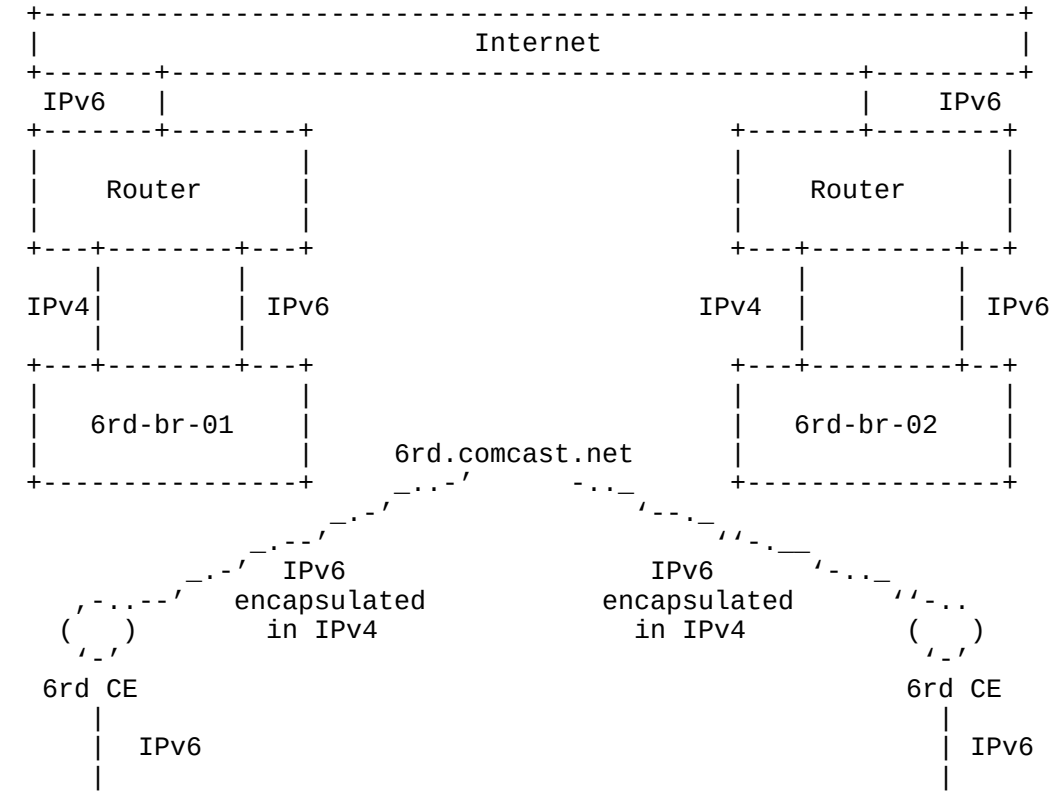


Figure 2: Comcast 6RD Overview

5. Native Dual Stack

Native dual stack is central to Comcast's IPv6 program for trial and production deployment. Native dual stack is the model where IPv4 services remain as-is with native IPv6 support introduced in parallel or simultaneously. Many of the details surrounding how this is achieved are documented as part of the CableLabs Data Over Cable Service Interface Specification (DOCSIS) 3.0 [DOCSIS3.0]. However, relevant trial and deployment specific information that is of interest to the IETF community will be documented.

Native dual stack trials depend on the upgrade and enablement of Cable Modem Termination Systems [CMTS] to support IPv6. A CMTS is a device that end users in a cable network connect directly to using their cable modem [CM]. As with IPv4, native support for IPv6 is critical for the delivery of services to end users in a DOCSIS network. Anything less could yield an undesirable end user experience or instability in the operator network that could adversely impact larger populations of users.

Given the CMTS requirements, native dual stack trials have initially been limited to specific areas of the network. Further, where CMTS platforms have been upgraded and enabled to support IPv6 end users have been incrementally enabled with support for IPv6. Again this is to ensure a controlled introduction with a specific focus on maintaining stability. Initially, a limited combination of cable modem and IGD devices are being used to support trial activities. Over time diversity for both cable modem and IGDs are expected to grow. To date a number of cable modems support the ability to enable native dual stack connectivity to CPEs devices behind them. A subset of pre-DOCSIS 3.0 and all DOCSIS 3.0 devices support this capability. The population of DOCSIS devices that support these capabilities varies from operator to operator.

Trial enablement requires the stateful provisioning of an IGD using stateful DHCPv6 [RFC3315] for the IGD WAN interface and delegated prefixes [RFC3633] for LAN side connectivity. Similarly, trial supported direct attachment of IPv6 capable CPE devices to the CM. In this configuration the CPE is provisioned with one or more IPv6 addresses via stateful DHCPv6 [RFC3315] in similar fashion to the IGD WAN interface. The quantity of devices supporting a native dual stack mode of operation is growing. While some devices are upgradable to support native dual stack many devices deployed today are not upgradable to support this functionality. Early implementations of devices or devices that are upgradable to support native IPv6 were found to only require and/or support the use of an IPv6 /64 for LAN side connectivity. This has been an acceptable mode of operation, however, over time IGDs will be required to support

more advanced functionality including the ability to support multiple, routed IPv6 LANs. While support for a single IPv6 /64 is in place today support for shorter IPv6 prefixes is also supported. It is important for operators to ensure they design and plan support across their infrastructures for delegated prefixes that are shorter than /64.

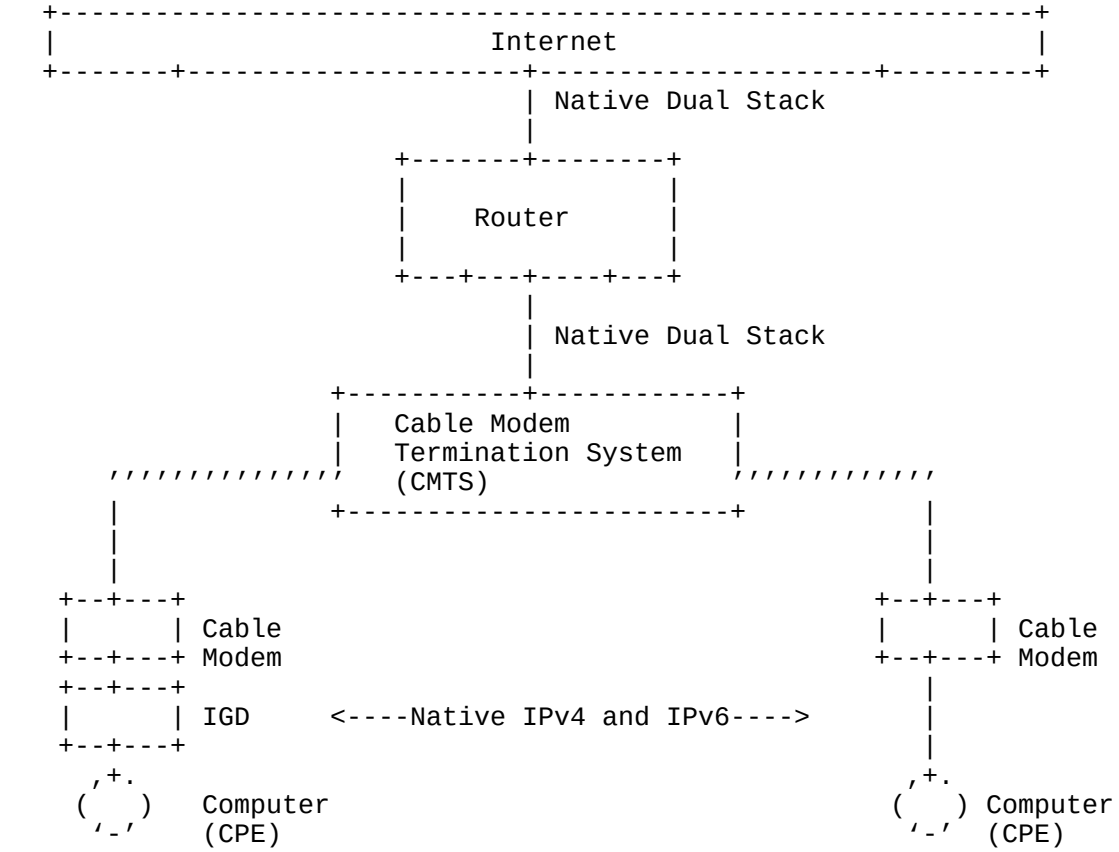


Figure 3: Comcast Native Dual Stack

6. Dual Stack Lite

Part of Comcast's trial plans includes the trialing of Dual Stack Lite. At this time trial planning for the same is underway. While Comcast plans on trialing Dual Stack Lite there are no plans at this

time to deploy Dual Stack Lite beyond a limited technology trial.

7. Content and Services

During early phases of our trials Comcast leveraged reverse proxies to expedite the availability of content natively over IPv6. Open source technology running on Linux based servers was used to enable the reverse proxies. To ensure that the origin content, which is IPv4 only, is available natively over IPv6 the proxy servers required native dual stack connectivity. This model allowed us to ensure that Internet facing access to Comcast content occurred natively over IPv6.

As third party CDNs introduce production quality support for IPv6 we plan to move away from the use of proxy servers and fully towards native dual stack for Comcast content and services. Native dual stack content is but the first step to ensure the same can be IPv6 only at some point in the future. Observations from Comcast's participation in World IPv6 day suggest it is premature to rely on IPv6-only content at this time

Further as part of our trials Comcast has also recently enabled IPv6 Message Transfer Agents (MTA), in a limited fashion, to allow a subset of Comcast trial users to send electronic mail using SMTP over IPv6.. Due to the limited availability of spam mitigation for IPv6 Comcast trials does not include the receipt of electronic mail over IPv6. In order to enable the receipt of electronic mail over IPv6 spam mitigation must be in place.

8. Backoffice

We made the decision early on in our design discussions to move all systems to a dual-stack design since we felt that this was the best way to transition to IPv6. The re-architect of many core systems like DNS, DHCP, OSS/BSS, and Billing systems took many years to plan and complete and this approach has paid off and allowed us to rapidly move towards support for dual-stack at the edge of our network, including support for our customers devices.

9. World IPv6 Day

During World IPv6 day, Comcast observed a significant increase in native IPv6 traffic once content providers enabled AAAA records for their websites. The resulting traffic has continued to increase even after World IPv6 when about 50% of the websites that participated in

World IPv6 Day left their AAAA records enabled after the day. We view this as a positive sign for continuing to drive more IPv6 traffic.

10. Conclusion

To date Comcast trial activities have yielded important, useful information about the various technologies that are available to facilitate the transition to IPv6. Observations and experience to date confirms that native dual stack is the preferred approach to transition to IPv6, where possible. While the various tunneling technologies are indeed straightforward to deploy there are a number of variables that must be considered when planning to deploy the same.

Support for native dual stack continues to evolve across various broadband technologies and within consumer electronics. As evidenced by World IPv6 Day many of the world’s largest content providers are also making progress with their IPv6 capabilities.

11. IANA Considerations

This document makes no request of IANA.

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

12. Security Considerations

There are no security considerations at this time.

13. Acknowledgements

Thanks to the Comcast team supporting the various trial and production deployment activities:

Jonathan Boyer

Chris Griffiths

Tom Klieber

Yiu Lee

Jason Livingood

Anthony Veiga

Joel Warburton

Richard Woundy

14. Normative References

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

Appendix A. Document Change Log

[RFC Editor: This section is to be removed before publication]

-02: Grammatical items and re-wording of some sections. We have also added a new World IPv6 Day section.

-01: Added C. Griffiths as co-author. Currently working on ascii art and several new sections.

-00: First version published.

Authors' Addresses

John Jason Brzozowski
Comcast Cable Communications
One Comcast Center
1701 John F. Kennedy Boulevard
Philadelphia, PA 19103
US

Email: john_brzozowski@cable.comcast.com
URI: http://www.comcast.com

Chris Griffiths
Comcast Cable Communications
One Comcast Center
1701 John F. Kennedy Boulevard
Philadelphia, PA 19103
US

Email: chris_griffiths@cable.comcast.com
URI: http://www.comcast.com

IPv6 Operations Working Group
Internet Draft
Intended status: Experimental
Expires: September 2011

Z. Kanizsai
BME
L. Bokor
BME
G. Jeney
BME
G. Panza
CEFRIEL

March 8, 2011

IPv6 anycast based feedback data aggregation
draft-kanizsai-v6ops-anycast-data-aggregation-00.txt

Status of this Memo

Abstract

This document describes how to use anycast addresses for collecting feedback information on the reverse link in case of multicast forward transmission. The application for anycast addressing in the case of multicast transmission is the novelty. The draft describes the fundamentals and requirements about how to collect and aggregate feedback information if anycasting is applied.

Table of Contents

1. Introduction

Cross-optimized communication architectures deeply rely on collection and evaluation of different feedback information provided by network nodes periodically or on-demand. However, information is often represented in a redundant way (e.g., series of measurement data with same values can be shortly represented by a single value together with zero standard deviation). As a technique to remove redundancy and achieve efficient communication, data aggregation has been introduced and widely investigated in the literature. The aim of data aggregation is to cut back the amount of data to be transmitted while still distributing the required information about events of interest. An adequate aggregation scheme can reduce the usage of bandwidth/energy/computational power of all architectural components and nodes in the network. Data aggregation considers two main aspects. On one hand, data-centric aggregation schemes are designed to address the encoding, calculation, and compression of aggregatable data coming from multiple sources (using aggregation functions such as MAX, MIN, AVERAGE, or the probabilistic aggregation). On the other hand, routing-centric aggregation mechanisms are supposed to cover routing problems: how (e.g., when and where) information pieces (i.e., datagrams) can meet each other in order to be aggregated. In the sections below we provide a general solution for the latter issue by introducing a feedback data aggregation architecture deploying designated entities inside the network (called aggregation servers) that collect individual feedback information pieces and relay the newly composed aggregated data towards further processing in an optimal way, all based on IPv6 anycasting.

2. IPv6 anycast based feedback data aggregation

In this section anycast based feedback aggregation is introduced according to different points of view, such as the typical scenario where this technique can be used, the required new entities introduced in the network and the newly proposed addressing architecture for efficient operation.

2.1. Application and usage scenarios

The typical scenario where the application of IPv6 anycast based feedback aggregation can be beneficial is depicted in Figure 1.

This scenario includes one single Server or Source of a general service (S), i.e. content(s), which requires feedback information from the subscribers called User Entities (UE). The UEs' connection type can be fixed or mobile as well. The feedback information helps the server to provide the best service given the current network

conditions by adaptively modifying the server working parameters as required.

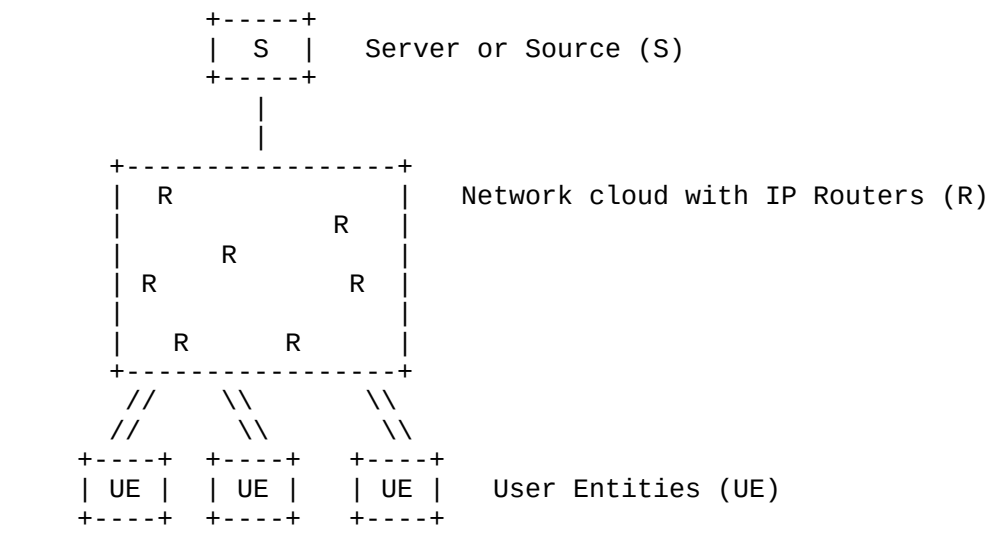


Figure 1 A typical scenario for anycast based aggregation

A feedback message is usually composed of several small numeric values which provide information about the actual quality of the service (QoS), network parameters, etc. These values are more often generated with high frequency and are only a few bits or bytes long, so sending each one to the server separately in an IP packet is a critical waste of bandwidth [FeedAgg]. To avoid this, a good solution is using IPv6 anycasting [RFC1546] and feedback aggregation in combination.

2.2. Terminology

- o Server or Source (S): A node in the network which provides for example adaptive multimedia streaming to the User Entities. This service is identified with a service anycast address which is practically identical and indistinguishable from the IPv6 unicast address of the server [RFC4291].

- o User Entity (UE): A user terminal which is able to subscribe to the service provided by the Server or Source. It measures some parameters of the received service data (e.g. multimedia stream) continuously and sends this information back to the adaptive Server to keep the QoS of the service as high as possible despite the constantly changing network conditions.

- o Anycast Capable Router (ACR): An IP packet router which is capable of handling anycast addresses and services in the network. The anycast service providers (i.e. the Feedback Aggregation Servers) are registering themselves in the closest ACR. They send their unicast address and the ID of the anycast service they are intended to participate in. During the operation of the ACR the packets that are addressed to the service's anycast address are forwarded to "at least one and preferably only one" service provider according to a parameter like hop count, the load of the servers, etc [RFC1546] [RFC4786].

- o Anycast routing protocol: A routing protocol running on the ACRs besides the normal routing process. This protocol maintains the anycast group information which is updated by the service providers periodically. A packet addressed to an anycast address is routed according to the current group state information to "at least one and preferably only one" service provider [RFC1546].

- o Feedback Aggregation Server (FAS): A server node which processes the incoming IP messages addressed to the anycast address of the service. The individual feedback messages are decapsulated and the FAS, which is aware of the feedback types of the given service, stores them in separate queues. Every feedback type has a lifetime, so the various types of feedback messages must be sent within different time constraints. When the timer expires for the first message in a queue, the complete content of this queue is placed in a new IP packet and sent to the server of the service.

- o Feedback Aggregation Address (FAA): This is the address the IPv6 packets, containing individual feedback messages, are addressed to. The FAA identifies the anycast group of the FASs and the Source. In practice, this address should be one of the Source's unicast addresses. This provides feedback delivery also in the cases when no ACR is present in the path from the UE back to the Source.

2.3. Protocol Operation Overview and Addressing

The service Source and the Feedback Aggregation Servers are in the same anycast group, addressable with the same anycast address, which should be one of the unicast addresses of the server [RFC4291]. The Source and the Feedback Aggregation Server are marked in the same way in Figure 2 according to the above reason. The Server should have assigned at least two unicast addresses, one used as the anycast group address and the other used by the aggregated IP packets sent by the FASs. The unicast packet forwarding between the FASs and the Source prevents packet looping between ACRs and FASs (Figure 2).

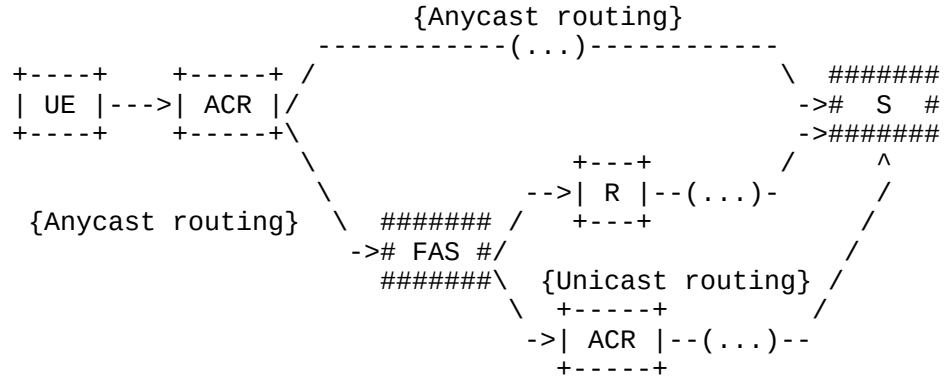


Figure 2 Possible paths for a feedback message

Using one of the unicast addresses of the service Source as anycast group address ensures that a packet is delivered to the proper destination even if it crosses only unicast capable routers along the path back to the source (Figure 3).

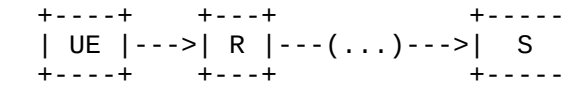


Figure 3 Feedback message path without ACRs

IPv6 anycasting helps reaching the aggregation servers in an optimal way: a UE addresses the feedback packets to the anycast address (FAA) of the aggregation servers (FAS), ensuring packets are delivered to the "closest" aggregation server (or directly to the service Source if this is the closest member of the anycast group (Figure 2 upper arrow)) using anycast routing protocol which is implemented in the intermediate Anycast Capable Routers (ACR). Note, that it is not necessary that all the routers be anycast capable: however, in this case, only sub-optimal transmission of feedback data is achievable. Furthermore, in this network scenario the stateless property of anycast communication [RFC1546] does not raise any problem, since the UEs send individual feedback packets and it makes no difference which aggregation server they are delivered to.

Aggregation servers supported by anycast communication provide Network-level (or System-level) aggregation in a (sub-)optimal way. After receiving feedback data from individual UEs the FASs aggregate the information and relay this newly composed aggregated data towards the adaptive service Source.

3. Benefits of using anycast based aggregation

In accordance with the literature, the aggregation ratio at network-level is determined by the length of the tracking history and the MTU size on the aggregation server's uplink. On average an aggregation ratio between 2 and 10 can be achieved. By applying this solution the overhead in the core network can be significantly reduced allowing also for an increased number of servable UEs for a given uplink transmission capacity of the Source.

4. Security Considerations

The above introduced solution does not raise new security issues or requirements, thus the considerations from [RFC1546] and [RFC4786] apply as well to this document.

5. IANA Considerations

This document has no new IANA considerations.

6. References

6.1. Normative References

[RFC1546] Partridge, C., Mendez, T. and W. Milliken, "Host Anycasting Service", RFC 1546, November 1993.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC4291] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 4291, February 2006.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", RFC 4786, December 2006.

6.2. Informative References

[FeedAgg] Kanizsai, Z., Bokor, L. and G. Jeney, "An Anycast based Feedback Aggregation Scheme for Efficient Network Transparency in Cross-layer Design", Submitted to Periodica Polytechnica Special Issue, Under review process

[AnyTerm] Hashimoto, M., Ata, S., Kitamura, H. and M. Murata, "IPv6 Anycast Terminolgy Definition", IETF Internet Draft, draft-doi-ipv6-anycast-func-term-05.txt, January 2006, work in progress

[AnyApp] Matsunaga, S., Ata, S., Kitamura, H. and M. Murata, "Applications of IPv6 Anycasting", IETF Internet Draft, draft-ata-ipv6-anycast-app-01.txt, February 2005, work in progress

7. Acknowledgments

This proposal results from the work carried out within the framework of OPTIMIX project (www.ict-optimix.eu) which is partly funded by the 7th Framework Programme (FP7) of the European Union's Information and Communication Technologies (ICT) under the contract FP7 No. INFSO-ICT-214625. The authors would like to thank all participants and contributors who take part in the studies.

The support of the Hungarian Government through the TAMOP-4.2.1/B-09/1/KMR-2010-0002 project at the Budapest University of Technology and Economics is also acknowledged.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Zoltan Kanizsai
Department of Telecommunications
Budapest University of Technology and Economics (BME)
Magyar Tudosok krt. 2. IB121
H-1117, Budapest
Hungary

Email: kanizsai@hit.bme.hu

Laszlo Bokor
Mobile Innovation Centre
Budapest University of Technology and Economics (BME)
Bertalan Lajos u. 2. Z301
H-1111, Budapest
Hungary

Email: goodzi@mcl.hu

Gabor Jeney
Department of Telecommunications
Budapest University of Technology and Economics (BME)
Magyar Tudosok krt. 2. IE450
H-1117, Budapest
Hungary

Email: jeneyg@hit.bme.hu

Gianmarco Panza
Digital Platform and Pervasive ICT Division
CEFRIEL - Politecnico di Milano
via Fucini 2, 20133 Milan
Italy

Email: Gianmarco.panza@cefriel.com

Individual Submission
Internet-Draft
Intended status: Informational
Expires: August 14, 2011

J. Korhonen, Ed.
Nokia Siemens Networks
J. Soininen
Renesas Mobile
B. Patil
T. Savolainen
G. Bajko
Nokia
K. Iisakkila
Renesas Mobile

February 10, 2011

IPv6 in 3GPP Evolved Packet System
draft-korhonen-v6ops-3gpp-eps-06

Abstract

Internet connectivity and use of data services in 3GPP based mobile networks has increased rapidly as a result of smart phones, broadband service via HSPA and HSPA+ networks, competitive service offerings by operators and a large number of applications. Operators who have deployed networks based on 3GPP architectures are facing IPv4 address shortages. With the impending exhaustion of available IPv4 addresses from the registries there is an increased emphasis for operators to migrate to IPv6. This document describes the support for IPv6 in 3GPP network architectures.

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 14, 2011.

Copyright Notice

Table of Contents

1. Introduction

IPv6 has been specified in the 3rd Generation Partnership Project (3GPP) standards since the early architectures developed for R99 General Packet Radio Service (GPRS). However, the support for IPv6 in commercially deployed networks by the end of 2010 is nearly non-existent. There are many factors that can be attributed to the lack of IPv6 deployment in 3GPP networks. The most relevant one is essentially the same as the reason for IPv6 not being deployed by other networks as well, i.e. the lack of business and commercial incentives for deployment. 3GPP network architectures have also evolved since 1999 (since R99). The most recent version of the 3GPP architecture, the Evolved Packet System (EPS), which is commonly referred to as SAE, LTE or Release-8, is a packet centric architecture. The number of subscribers and devices that are using the 3GPP networks for Internet connectivity and data services has also increased significantly. With the subscriber growth numbers projected to increase even further and the IPv4 addresses depletion problem looming in the near term, 3GPP operators and vendors have started the process of identifying the scenarios and solutions needed to transition to IPv6.

This document describes the establishment of IP connectivity in 3GPP network architectures, specifically in the context of IP bearers for 3GPP GPRS and for 3GPP EPS. It provides an overview of how IPv6 is supported as per the current set of 3GPP specifications. Some of the issues and concerns with respect to deployment and shortage of private IPv4 addresses within a single network domain are also discussed.

The IETF has specified a set of tools and mechanisms that can be utilized for transitioning to IPv6. In addition to operating dual-stack networks during the transition from IPv4 to IPv6 phase, the two alternative categories for the transition are encapsulation and translation. Most of the mechanisms available in the toolbox can be categorized into either translation or encapsulation approaches. The IETF continues to specify additional solutions for enabling the transition based on the deployment scenarios and operator/ISP requirements. There is no single approach for transition to IPv6 that can meet the needs for all deployments and models. The 3GPP scenarios for transition, described in [3GPP.23.975], can be addressed using transition mechanisms that are already available in the toolbox. The objective of transition to IPv6 in 3GPP networks is to ensure that:

1. Legacy devices and hosts which have an IPv4 only stack will continue to be provided with IP connectivity to the Internet and services,

2. Devices which are dual-stack can access the Internet either via IPv6 or IPv4. The choice of using IPv6 or IPv4 depends on the capability of:

   A. the application on the host,

   B. the support for IPv4 and IPv6 bearers by the network and/or,

   C. the capability of the server(s) and other end points.

3GPP networks are capable of providing a host with IPv4 and IPv6 connectivity today, albeit in many cases with upgrades to network elements such as the SGSN and GGSN.

2. 3GPP Terminology and Concepts

2.1. Terminology

Access Point Name

Access Point Name (APN) is a fully qualified domain name and resolves to a specific gateway in an operators network. The APNs are piggybacked on the administration of the DNS namespace.

Packet Data Protocol Context

A Packet Data Protocol (PDP) Context is the equivalent of a virtual connection between the host and a gateway.

General Packet Radio Service

General Packet Radio Service (GPRS) is a packet oriented mobile data service available to users of the 2G and 3G cellular communication systems Global System for Mobile communications (GSM), and specified by 3GPP.

Packet Data Network

Packet Data Network (PDN) is a packet based network that either belongs to the operator or is an external network such as Internet and corporate intranet. The user eventually accesses services in one or more PDNs. The operator’s packet domain network are separated from packet data networks either by GGSNs or PDN Gateways (PDN-GW).

Gateway GPRS Support Node

Gateway GPRS Support Node (GGSN) is a gateway function in GPRS, which provides connectivity to Internet or other PDNs. The host attaches to a GGSN identified by an APN assigned to it by an operator. The GGSN also serves as the topological anchor for addresses/prefixes assigned to the mobile host.

Packet Data Network Gateway

Packet Data Network Gateway (PDN-GW) is a gateway function in Evolved Packet System (EPS), which provides connectivity to Internet or other PDNs. The host attaches to a PDN-GW identified by an APN assigned to it by an operator. The PDN-GW also serves as the topological anchor for addresses/prefixes assigned to the mobile host.

Serving Gateway

Serving Gateway (SGW) is a gateway function in EPS, which terminates the interface towards E-UTRAN. The SGW is the Mobility Anchor point for layer-2 mobility (inter-eNodeB handovers). For each User Equipment connected with the EPS, at any given point of time, there is only one SGW. The SGW is essentially the user plane part of the GPRS' SGSN forwarding packets between a PDN-GW.

Serving Gateway Support Node

Serving Gateway Support Node (SGSN) is a network element that is located between the radio access network (RAN) and the gateway (GGSN). A per mobile host point to point (p2p) tunnel between the GGSN and SGSN transports the packets between the mobile host and the gateway.

GPRS tunnelling protocol

GPRS Tunnelling Protocol (GTP) [3GPP.29.060] [3GPP.29.274] is a tunnelling protocol defined by 3GPP. It is a network based mobility protocol and similar to Proxy Mobile IPv6 (PMIPv6) [RFC5213]. However, GTP also provides functionality beyond mobility such as inband signaling related to Quality of Service (QoS) and charging among others.

Evolved Packet System

Evolved Packet System (EPS) is an evolution of the 3GPP GPRS system characterized by higher-data-rate, lower-latency, packet-optimized system that supports multiple Radio Access Technologies

(RAT). The EPS comprises the Evolved Packet Core (EPC) together with the evolved radio access network (E-UTRA and E-UTRAN).

Mobility Management Entity

Mobility Management Entity (MME) is a network element that is responsible for control plane functionalities, including authentication, authorization, bearer management, layer-2 mobility, etc. The MME is essentially the control plane part of the GPRS' SGSN and not located on the user plane data path, i.e. user plane traffic bypasses the MME.

UMTS Terrestrial Radio Access Network

UMTS Terrestrial Radio Access Network (UTRAN) is communications network, commonly referred to as 3G, and consists of NodeBs (3G base station) and Radio Network Controllers (RNC) which make up the UMTS radio access network. The UTRAN allows connectivity between the mobile host/device and the core network. UTRAN comprises of WCDMA, HSPA and HSPA+ radio technologies.

Wideband Code Division Multiple Access

The Wideband Code Division Multiple Access (WCDMA) is the radio interface used in UMTS networks.

High Speed Packet Access

The High Speed Packet Access (HSPA) and the Evolved High Speed Packet Access (HSPA+) are enhanced versions of the WCDMA and UTRAN, thus providing more data throughput and lower latencies.

Evolved UTRAN

Evolved UTRAN (E-UTRAN) is communications network, sometimes referred to as 4G, and consists of eNodeBs (4G base station) which make up the E-UTRAN radio access network. The E-UTRAN allows connectivity between the mobile host/device and the core network.

eNodeB

The eNodeB is a base station entity that supports the Long Term Evolution (LTE) air interface.

GSM EDGE Radio Access Network

GSM EDGE Radio Access Network (GERAN) is communications network, commonly referred to as 2G or 2.5G, and consists of base stations

and Base Station Controllers (BSC) which make up the GSM EDGE radio access network. The GERAN allows connectivity between the mobile host/device and the core network.

UE, MS, MN and Mobile

The terms UE (User Equipment), MS (Mobile Station), MN (Mobile Node) and, mobile refer to the devices which are hosts with ability to obtain Internet connectivity via a 3GPP network. The terms UE, MS, MN and devices are used interchangeably within this document.

PCC

The Policy and Charging Control (PCC) framework is used for QoS policy and charging control. It is optional for 3GPP EPS but needed if dynamic policy and charging control by means of PCC rules based on user and services are desired.

HLR

The Home Location Register (HLR) is a pre-Release-5 database (the reality regarding releases is different, though) for a given subscriber. It is the entity containing the subscription-related information to support the network entities actually handling calls/sessions.

HSS

The Home Subscriber Server (HSS) is a database for a given subscriber and got introduced in 3GPP Release-5. It is the entity containing the subscription-related information to support the network entities actually handling calls/sessions.

2.2. The concept of APN

The Access Point Name (APN) essentially refers to a gateway in the 3GPP network. The 'complete' APN is expressed in a form of a Fully Qualified Domain Name (FQDN) and also piggybacked on the administration of the DNS namespace, thus effectively allowing the discovery of gateways using the DNS. Mobile hosts/devices can choose to attach to a specific gateway in the packet core. The gateway provides connectivity to the Packet Data Network (PDN) such as the Internet. An operator may also include gateways which do not provide Internet connectivity, rather a connectivity to closed network providing a set of operator's own services. A mobile host/device can be attached to one or more gateways simultaneously. The gateway in a 3GPP network is the GGSN or PDN-GW. Figure 1 below illustrates the

APN-based network connectivity concept.

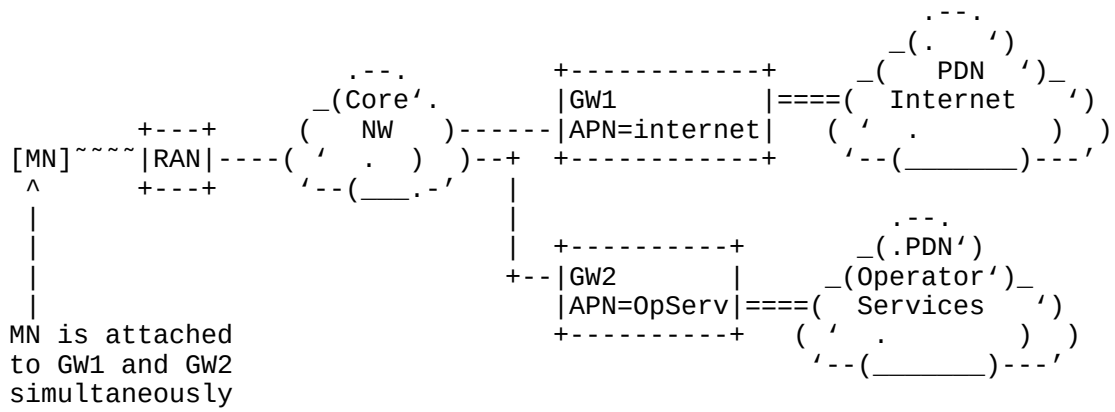


Figure 1: Mobile host/device attached to multiple APNs simultaneously

3. IP over 3GPP GPRS

3.1. Introduction to 3GPP GPRS

A simplified 2G/3G GPRS architecture is illustrated in Figure 2. This architecture basically covers the GPRS core network since R99 to Release-7, and radio access technologies such as GSM (2G), EDGE (2G, ofter referred as 2.5G), WCDMA (3G) and HSPA(+) (3G, often referred as 3.5G). The architecture shares obvious similarities with the Evolved Packet System (EPS) as will be seen in Section 4. Based on Gn/Gp interfaces, the GPRS core network functionality is logically implemented on two network nodes, the SGSN and the GGSN.

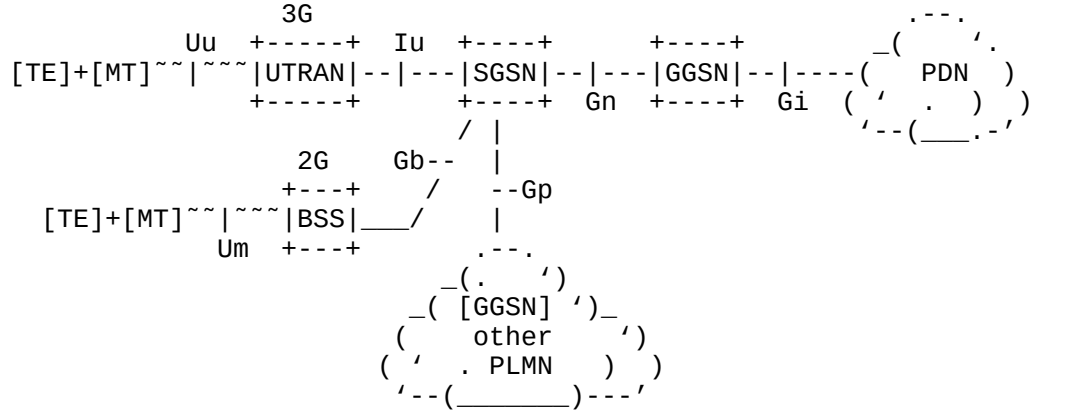


Figure 2: Overview of the 2G/3G GPRS Logical Architecture

Gn/Gp: These interfaces provide a network based mobility service for a mobile host and are used between a SGSN and a GGSN. The Gn interface is used when GGSN and SGSN are located inside one operator (i.e. PLMN). The Gp-interface is used if the GGSN and the SGSN are located in different operator domains (i.e. 'other' PLMN). GTP protocol is defined for the Gn/Gp interfaces (both GTP-C for the control plane and GTP-U for the user plane).

Gb: Is the Base Station System (BSS) to SGSN interface, which is used to carry information concerning packet data transmission and layer-2 mobility management. The Gb-interface is based on either on Frame Relay or IP.

Iu: Is the Radio Network System (RNS) to SGSN interface, which is used to carry information concerning packet data transmission and layer-2 mobility management. The user plane part of the Iu-interface (actually the Iu-PS) is based on GTP-U. The control plane part of the Iu-interface is based on Radio Access Network Application Protocol (RANAP).

Gi: It is the interface between the GGSN and a PDN. The PDN may be an operator external public or private packet data network or an intra-operator packet data network.

Uu/Um: Are either 2G or 3G radio interfaces between a mobile terminal and a respective radio access network.

The SGSN is responsible for the delivery of data packets from and to the mobile hosts within its geographical service area when a direct tunnel option is not used. If the direct tunnel is used, then the user plane goes directly between the RNS and the GGSN. The control plane traffic always goes through the SGSN. For each mobile host connected with the GPRS, at any given point of time, there is only one SGSN.

3.2. PDP Context

A PDP context is an association between a mobile host represented by one IPv4 address and/or one /64 IPv6 prefix and a PDN represented by an APN. Each PDN can be accessed via a gateway (typically a GGSN or PDN-GW). On the device/mobile host a PDP context is equivalent to a network interface. A host may hence be attached to one or more gateways via separate connections, i.e. PDP contexts. Each primary PDP context has its own IPv4 address and/or one /64 IPv6 prefix assigned to it by the PDN and anchored in the corresponding gateway.

Applications on the host use the appropriate network interface (PDP context) for connectivity to a specific PDN. Figure 3 represents a high level view of what a PDP context implies in 3GPP networks.

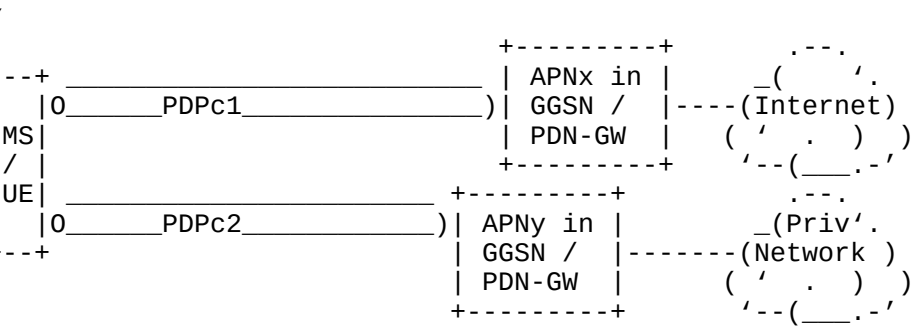


Figure 3: PDP contexts between the MS/UE and gateway

In the above figure there are two PDP contexts at the MS/UE (UE=User Equipment in 3GPP parlance). The 'PDPc1' PDP context that is connected to APNx provided Internet connectivity and the 'PDPc2' PDP context provides connectivity to a private IP network via APNy (as an example this network may include operator specific services such as MMS (Multi media service). An application on the host such as a web browser would use the PDP context that provides Internet connectivity for accessing services on the Internet. An application such as MMS would use APNy in the figure above because the service is provided through the private network.

4. IP over 3GPP EPS

4.1. Introduction to 3GPP EPS

In its most basic form, the EPS architecture consists of only two nodes on the user plane, a base station and a core network Gateway (GW). The basic EPS architecture is illustrated in Figure 4. The Mobility Management Entity (MME) node performs control-plane functionality and is separated from the node(s) that performs bearer-plane functionality (GW), with a well-defined open interface between them (S11). The optional interface S5 can be used to split the Gateway (GW) into two separate nodes, the Serving Gateway (SGW) and the PDN-GW. This allows independent scaling and growth of traffic throughput and control signal processing. The functional split of gateways also allows for operators to choose optimized topological locations of nodes within the network and enables various deployment models including the sharing of radio networks between different operators.

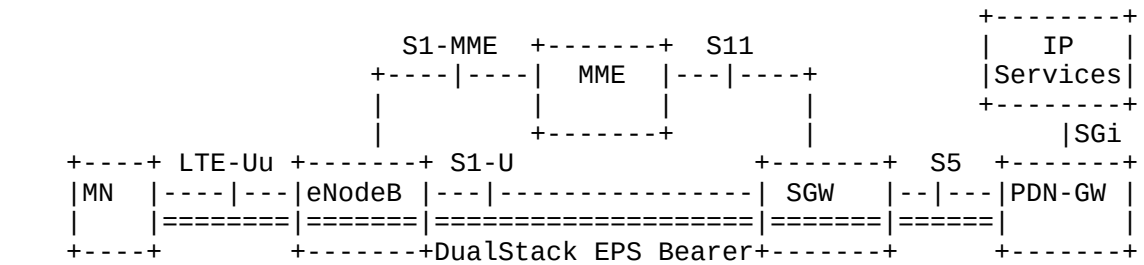


Figure 4: EPS Architecture for 3GPP Access

S5: It provides user plane tunnelling and tunnel management between SGW and PDN-GW, using GTP or PMIPv6 as the network based mobility management protocol.

S1-U: Provides user plane tunnelling and inter eNodeB path switching during handover between eNodeB and SGW, using the GTP-U protocol (GTP user plane).

S1-MME: Reference point for the control plane protocol between eNodeB and MME.

SGi: It is the interface between the PDN-GW and the packet data network. Packet data network may be an operator external public or private packet data network or an intra operator packet data network.

The eNodeB is a base station entity that supports the Long Term Evolution (LTE) air interface and includes functions for radio resource control, user plane ciphering, and other lower layer functions. MME is responsible for control plane functionalities, including authentication, authorization, bearer management, layer-2 mobility, etc.

The SGW is the Mobility Anchor point for layer-2 mobility. For each MN connected with the EPS, at any given point of time, there is only one SGW.

4.2. PDN Connection

A PDN connection is an association between a mobile host represented by one IPv4 address and/or one /64 IPv6 prefix, and a PDN represented by an APN. The PDN connection is the EPC equivalent of the GPRS PDP context. Each PDN can be accessed via a gateway (a PDN-GW). PDN is responsible for the IP address/prefix allocation to the mobile host. On the device/mobile host a PDN connection is equivalent to a network interface. A host may hence be attached to one or more gateways via

separate connections, i.e. PDN connections. Each PDN connection has its own IP address/prefix assigned to it by the PDN and anchored in the corresponding gateway. Applications on the host use the appropriate network interface (PDN connection) for connectivity.

4.3. EPS bearer model

The logical concept of a bearer has been defined to be an aggregate of one or more IP flows related to one or more services. An EPS bearer exists between the Mobile Node (MN i.e. a mobile host) and the PDN-GW and is used to provide the same level of packet forwarding treatment to the aggregated IP flows constituting the bearer. Services with IP flows requiring a different packet forwarding treatment would therefore require more than one EPS bearer. The mobile host performs the binding of the uplink IP flows to the bearer while the PDN-GW performs this function for the downlink packets.

In order to provide low latency for always on connectivity, a default bearer will be provided at the time of startup and an IPv4 address and/or IPv6 prefix gets assigned to the mobile host (this is different from GPRS, where mobile hosts are not automatically assigned with an IP address or prefix). This default bearer will be allowed to carry all traffic which is not associated with a dedicated bearer. Dedicated bearers are used to carry traffic for IP flows that have been identified to require a specific packet forwarding treatment. They may be established at the time of startup; for example, in the case of services that require always-on connectivity and better QoS than that provided by the default bearer. The default bearer and the dedicated bearer(s) associated to it share the same IP address(es)/prefix.

An EPS bearer is referred to as a GBR bearer if dedicated network resources related to a Guaranteed Bit Rate (GBR) value that is associated with the EPS bearer are permanently allocated (e.g. by an admission control function in the eNodeB) at bearer establishment/ modification. Otherwise, an EPS bearer is referred to as a non-GBR bearer. The default bearer is always non-GBR, with the resources for the IP flows not guaranteed at eNodeB, and with no admission control. However, the dedicated bearer can be either GBR or non-GBR. A GBR bearer has a Guaranteed Bit Rate (GBR) and Maximum Bit Rate (MBR) while more than one non-GBR bearer belonging to the same UE shares an Aggregate Maximum Bit Rate (AMBR). Non-GBR bearers can suffer packet loss under congestion while GBR bearers are immune to such losses.

5. Address Management

5.1. IPv4 Address Configuration

Mobile host's IPv4 address configuration is always performed during PDP context/EPS bearer setup procedures (on layer-2). DHCPv4-based [RFC2131] address configuration is supported by the 3GPP specifications, but is not used in wide scale. The mobile host must always support layer-2 based address configuration, since DHCPv4 is optional for both mobile hosts and networks.

5.2. IPv6 Address Configuration

IPv6 Stateless Address Autoconfiguration (SLAAC) as specified in [RFC4862] is the only supported address configuration mechanism. Stateful DHCPv6-based address configuration is not supported by 3GPP specifications [RFC3315]. On the other hand, Stateless DHCPv6-service to obtain other configuration information is supported [RFC3736]. This implies that the M-bit must always be set to zero and the O-bit may be set to one in the Router Advertisement (RA) sent to the UE.

3GPP network allocates each default bearer a unique /64 prefix, and uses layer-2 signaling to suggest user equipment an Interface Identifier that is guaranteed not to conflict with gateway's Interface Identifier. The UE may configure link local address using this Interface Identifier, but is allowed to use also other Interface Identifiers and as many globally scoped addresses as it needs. There is no restriction, for example, of using Privacy Extension for SLAAC [RFC4941] or other similar types of mechanisms.

In the 3GPP link model the /64 prefix assigned to the UE is always off-link (i.e. the L-bit in the Prefix Information Option (PIO) in the RA must be set to zero). If the advertised prefix is used for SLAAC then the A-bit in the PIO must be set to one. The details of the 3GPP link-model and address configuration is described in Section 11.2.1.3.2a of [3GPP.29.061]. More specifically, the GGSN/PDN-GW guarantees that the /64 prefix is unique for the mobile host. Therefore, there is no need to perform any Duplicate Address Detection (DAD) on addresses the mobile host creates (i.e., the 'DupAddrDetectTransmits' variable in the mobile host should be zero). The GGSN/PDN-GW is not allowed to generate any globally unique IPv6 addresses for itself using the /64 prefix assigned to the mobile host in the RA.

The current 3GPP architecture limits number of prefixes in each bearer to a single /64 prefix. If the mobile host finds more than one prefix in the RA, it only considers the first one and silently discard the others [3GPP.29.061]. Therefore, multi-homing within a single bearer is not possible. Renumbering without closing layer-2

connection is also not possible. The lifetime of /64 prefix is bound to lifetime of layer-2 connection even if the advertised prefix lifetime would be longer than the layer-2 connection lifetime.

5.3. Prefix Delegation

IPv6 prefix delegation is a part of Release-10 and is not covered by any earlier release. However, the /64 prefix allocated for each default bearer (and to the user equipment) may be shared to local area network by user equipment implementing Neighbor Discovery proxy (ND proxy) [RFC4389] functionality.

Release-10 prefix delegation uses the DHCPv6-based prefix delegation [RFC3633]. The model defined for Release-10 requires aggregatable prefixes, which means the /64 prefix allocated for the default bearer (and to the user equipment) must be part of the shorter delegated prefix. DHCPv6 prefix delegation has an explicit limitation described in Section 12.1 of [RFC3633] that a prefix delegated to a requesting router cannot be used by the delegating router (i.e., the PDN-GW in this case). This implies the shorter 'delegated prefix' cannot be given to the requesting router (i.e. the user equipment) as such but has to be delivered by the delegating router (i.e. the PDN-GW) in such a way the /64 prefix allocated to the default bearer is not part of the 'delegated prefix'. IETF is working on a solution for DHCPv6-based prefix delegation to exclude a specific prefix from the 'delegated prefix' [I-D.ietf-dhc-pd-exclude].

6. 3GPP Dual-Stack Approach to IPv6

6.1. 3GPP Networks Prior to Release-8

3GPP standards prior to Release-8 provide IPv6 access for cellular devices with PDP contexts of type IPv6 [3GPP.23.060]. For dual-stack access, a PDP context of type IPv6 is established in parallel to the PDP context of type IPv4, as shown in Figure 5 and Figure 6. For IPv4-only service, connections are created over the PDP context of type IPv4 and for IPv6-only service connections are created over the PDP context of type IPv6. The two PDP contexts of different type may use the same APN (and the gateway), however, this aspect is not explicitly defined in standards. Therefore, cellular device and gateway implementations from different vendors may have varying support for this functionality.

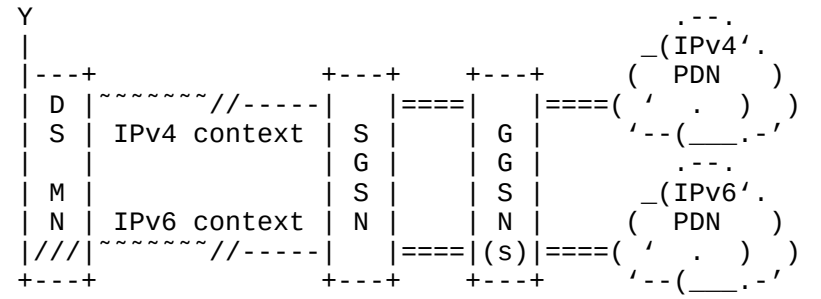


Figure 5: A dual-stack mobile host connecting to both IPv4 and IPv6 Internet using parallel IPv4-only and IPv6-only PDP contexts

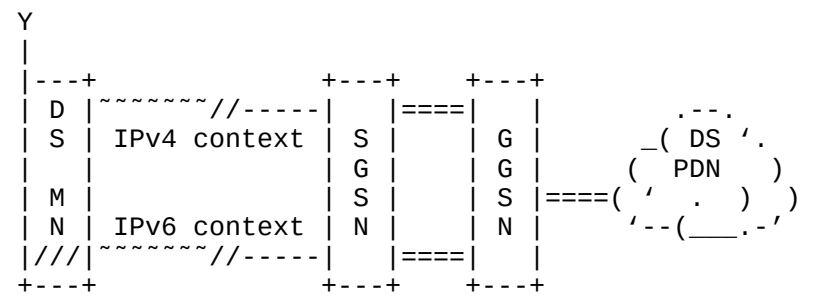


Figure 6: A dual-stack mobile host connecting to dual-stack Internet using parallel IPv4-only and IPv6-only PDP contexts

The approach of having parallel IPv4 and IPv6 type of PDP contexts open is not optimal, because two PDP contexts require double the signaling and consume more network resources than a single PDP context. In the figure above the IPv4 and IPv6 PDP contexts are attached to the same GGSN. While this is possible, the DS MS may be attached to different GGSNs in the scenario where one GGSN supports IPv4 PDN connectivity while another GGSN provides IPv6 PDN connectivity.

6.2. 3GPP Release-8 and -9 Networks

Since 3GPP Release-8, the powerful concept of a dual-stack type of PDN connection and EPS bearer have been introduced [3GPP.23.401]. This enables parallel use of both IPv4 and IPv6 on a single bearer (IPv4v6), as illustrated in Figure 7, and makes dual stack simpler than in earlier 3GPP releases. As of Release-9, GPRS network nodes also support dual-stack type (IPv4v6) PDP contexts.

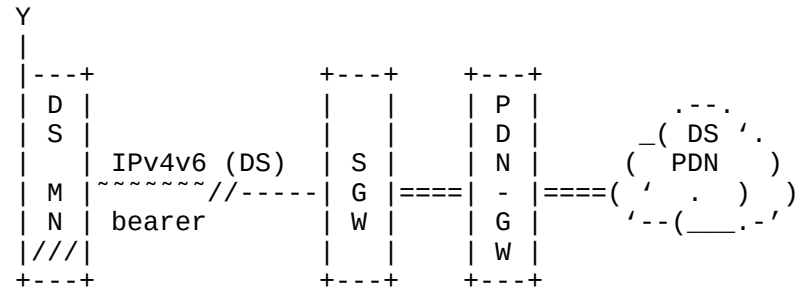


Figure 7: A dual-stack mobile host connecting to dual-stack Internet using a single IPv4v6 type PDN connection

The following is a description of the various PDP contexts/PDN bearer types that are specified by 3GPP:

1. For 2G/3G access to GPRS core (SGSN/GGSN) pre-Release-9 there are two IP PDP Types, IPv4 and IPv6. Two PDP contexts are needed to get dual stack connectivity.

2. For 2G/3G access to GPRS core (SGSN/GGSN) from Release-9 there are three IP PDP Types, IPv4, IPv6 and IPv4v6. Minimum one PDP context is needed to get dual stack connectivity.

3. For 2G/3G access to EPC core (PDN-GW via S4 Release-8 SGSN) from Release-8 there are three IP PDP Types, IPv4, IPv6 and IPv4v6 which gets mapped to PDN Connection type. Minimum one PDP Context is needed to get dual stack connectivity.

4. For LTE (E-UTRAN) access to EPC core from Release-8 there are three IP PDN Types, IPv4, IPv6 and IPv4v6. Minimum one PDN Connection is needed to get dual stack connectivity.

6.3. PDN Connection Establishment Process

The PDN connection establishment process is specified in detail in 3GPP specifications. Figure 8 illustrates the high level process and signaling involved in the establishment of a PDN connection.

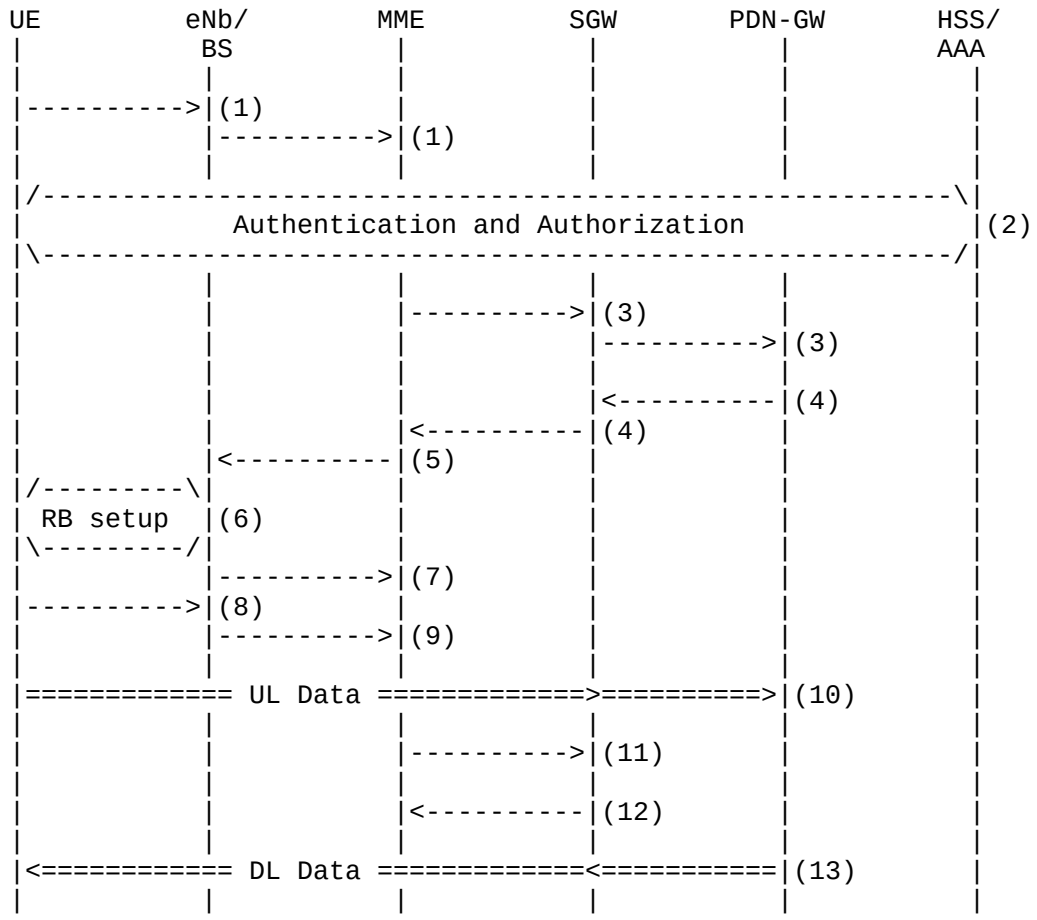


Figure 8: Simplified PDN connection setup procedure in Release-8

1. The UE (i.e the MS) requires a data connection and hence decides to establish a PDN connection with a PDN-GW. The UE sends an "Attach Request" (layer-2) to the BS. The BS forwards this attach request to the MME.

2. Authentication of the UE with the AAA server/HSS follows. If the UE is authorized for establishing a data connection, the following steps continue

3. The MME sends a "Create Session Request" message to the Serving-GW. The SGW forwards the create session request to the PDN-GW. The SGW knows the address of the PDN-GW to forward the create session request to as a result of this information having been obtained by the MME during the authentication/authorization

phase.

The UE IPv4 address and/or IPv6 prefix get assigned during this step. If a subscribed IPv4 address and/or IPv6 prefix is statically allocated for the UE for this APN, then the MME already passes the address information to the SGW and eventually to the PDN-GW in the "Create Session Request" message. Otherwise, the PDN-GW manages the address assignment to the UE (there is another variation to this where IPv4 address allocation is delayed until the UE initiates a DHCPv4 exchange but this is not discussed here).

4. The PDN-GW creates a PDN connection for the UE and sends "Create Session Response" message to the SGW from which the session request message was received from. The SGW forwards the response to the corresponding MME which originated the request.

5. The MME sends the "Attach Accept/Initial Context Setup request" message to the eNodeB/BS.

6. The radio bearer between the UE and the eNb is reconfigured based on the parameters received from the MME

7. The eNb sends "Initial Context Response" message to the MME.

8. The UE sends a "Direct Transfer" message to the eNodeB which includes the Attach complete signal.

9. The eNodeB forwards the Attach complete message to the MME.

10. The UE can now start sending uplink packets to the PDN GW.

11. The MME sends a "Modify Bearer Request" message to the SGW.

12. The SGW responds with a "Modify Bearer Response" message. At this time the downlink connection is also ready

13. The UE can now start receiving downlink packets

The type of PDN connection established between the UE and the PDN-GW can be any of the types described in the previous section. The DS PDN connection, i.e the one which supports both IPv4 and IPv6 packets is the default one that will be established if no specific PDN connection type is specified by the UE in Release-8 networks.

6.4. Mobility of 3GPP IPv4v6 Type of Bearers

3GPP discussed at length various approaches to support mobility between Release-8 and pre-Release-8 networks for the new dual-stack type of bearers.

The chosen approach for mobility is as follows, in short: if a mobile is known to be at risk for doing handovers between Release-8 and pre-Release-8 networks, only single stack bearers are used. Essentially meaning:

1. If a network knows a mobile may do handovers between Release-8 and pre-Release-8 networks (segment), network will only provide single stack bearers, even if the mobile host requests dual-stack bearers. This can happen e.g. if an operator is using pre-Release-8 SGSNs in some parts of the network. The single stack bearers of Release-8 are easy to map one-to-one to pre-Release-8 bearers.

2. If a network knows a mobile will not be able to do handover to pre-Release-8 network (segment), it will provide mobile with dual-stack bearers on request. This can happen e.g. if an operator has upgraded their SGSNs to support dual-stack bearers, or if an operator is running LTE-only network.

When a network operator and their roaming partners have upgraded their networks to Release-8, it is possible to use the new IPv4v6 dual-stack type of bearers. A Release-8 mobile device always requests for a dual-stack bearer, but accepts what is assigned by the network.

7. Dual-Stack Approach to IPv6 Transition in 3GPP Networks

3GPP networks can natively transport IPv4 and IPv6 packets between the mobile station/UE and the gateway (GGSN or PDN-GW) as a result of establishing either a dual-stack PDP context or parallel IPv4 and IPv6 PDP contexts.

Current deployments of 3GPP networks primarily support IPv4 only. These networks can be upgraded to also support IPv6 PDP contexts. By doing so devices and applications that are IPv6 capable can start utilizing the IPv6 connectivity. This will also ensure that legacy devices and applications continue to work with no impact. As newer devices start using IPv6 connectivity, the demand for actively used IPv4 connections is expected to slowly decrease, helping operators with a transition to IPv6. With a dual-stack approach, there is always the potential to fallback to IPv4. A device which may be

roaming in a network wherein IPv6 is not supported by the visited network could fall back to using IPv4 PDP contexts and hence the end user would at least get some connectivity. Unfortunately, dual-stack approach as such does not lower the number of used IPv4 addresses. Every dual-stack bearer still needs to given an IPv4 address, private or public. This is a major concern with dual-stack bearers concerning IPv6 transition. However, if the majority of active IP communication has moved over to IPv6, then in case of NAT44 [RFC1918] IPv4 connections the number of active IPv4 connections can still be expected to gradually decrease and thus giving some level of relief regarding NAT44 function scalability.

As the networks evolve to support Release-8 EPS architecture and the dual-stack PDP contexts, newer devices will be able to leverage such capability and have a single bearer which supports both IPv4 and IPv6. Since IPv4 and IPv6 packets are carried as payload within GTP between the MS and the gateway (GGSN/PDN-GW) the transport network capability in terms of whether it supports IPv4 or IPv6 on the interfaces between the eNodeB and SGW or, SGW and PDN-GW is immaterial.

8. Deployment issues

8.1. Overlapping IPv4 Addresses

Given the shortage of globally routable public IPv4 addresses, operators tend to assign private IPv4 addresses [RFC1918] to hosts when they establish an IPv4 only PDP context or an IPv4v6 type PDN context. About 16 million hosts can be assigned a private IPv4 address that is unique within a domain. However, in case of many operators the number of subscribers is greater than 16 million. The issue can be dealt with by assigning overlapping RFC 1918 IPv4 addresses to hosts. As a result the IPv4 address assigned to a host within the context of a single operator realm would no longer be unique. This has the obvious and know issues of NATed IP connection in the Internet. Direct host to host connectivity becomes complicated, unless the hosts are within the same private address range pool and/or anchored to the same gateway, referrals using IP addresses will have issues and so forth. These are generic issues and not only a concern of the EPS. However, 3GPP as such does not have any mandatory language concerning NAT44 functionality in EPC. Obvious deployment choices apply also to EPC:

1. Very large network deployments are partitioned, for example, based on a geographical areas. This partitioning allows overlapping IPv4 addresses ranges to be assigned to hosts that are in different areas. Each area has its own pool of gateways

that are dedicated for a certain overlapping IPv4 address range (referred here later as a zone). Standard NAT44 functionality enables the communication between hosts that are assigned the same IPv4 address but belong to different zones, yet are part of the same operator domain.

2. A mobile host/device attaches to a gateway as part of the attach process. The number of hosts that a gateway supports is in the order of 1 to 10 million. Hence all the hosts assigned to a single gateway can be assigned private IPv4 addresses. Operators with large subscriber bases have multiple gateways and hence the same [RFC1918] IPv4 address space can be reused across gateways. The IPv4 address assigned to a host is unique within the scope of a single gateway.

3. New services requiring direct connectivity between hosts should be build on IPv6. Possible existing IPv4-only services and applications requiring direct connectivity can be ported to IPv6.

8.2. IPv6 for transport

The various reference points of the 3GPP architecture such as S1-U, S5 and S8 are based on either GTP or PMIPv6. The underlying transport for these reference points can be IPv4 or IPv6. GTP has been able to operate over IPv6 transport (optionally) since R99 and PMIPv6 has supported IPv6 transport starting from its introduction in Release-8. The user plane traffic between the mobile host and the gateway can use either IPv4 or IPv6. These packets are essentially treated as payload by GTP/PMIPv6 and transported accordingly with no real attention paid to the information (at least from a routing perspective) contained in the IPv4 or IPv6 headers. The transport links between the eNodeB and the SGW, and the link between the SGW and PDN-GW can be migrated to IPv6 without any direct implications to the architecture.

Currently, the inter-operator (for 3GPP technology) roaming networks are all IPv4 only (see Inter-PLMN Backbone Guidelines [GSMA.IR.34]). Eventually these roaming networks will also get migrated to IPv6, if there is a business reason for that. The migration period can be prolonged considerably because the 3GPP protocols always tunnel user plane traffic in the core network and as described earlier the transport network IP version is not in any way tied to user plane IP version. Furthermore, the design of the inter-operator roaming networks is such that the user plane and transport network IP addressing is completely separated from each other. The inter-operator roaming network itself is also completely separated from the Internet. Only those core network nodes that must be connected to the inter-operator roaming networks are actually visible there, and

be able to send and receive (tunneled) traffic within the inter-operator roaming networks. Obviously, in order the roaming to work properly, the operators have to agree on supported protocol versions so that the visited network does not, for example, unnecessarily drop user plane IPv6 traffic.

8.3. Operational Aspects of Running Dual-Stack Networks

Operating dual-stack networks does imply cost and complexity to a certain extent. However these factors are mitigated by the assurance that legacy devices and services are unaffected and there is always a fallback to IPv4 in case of issues with the IPv6 deployment or network elements. The model also enables operators to develop operational experience and expertise in an incremental manner.

Running dual-stack networks requires the management of multiple IP address spaces. Tracking of hosts needs to be expanded since it can be identified by either an IPv4 address or IPv6 prefix. Network elements will also need to be dual-stack capable in order to support the dual-stack deployment model.

Deployment and migration cases described in Section 6.1 for providing dual-stack like capability may mean doubled resource usage in operator’s network. This is a major concern against providing dual-stack like connectivity using techniques discussed in Section 6.1. Also handovers between networks with different capabilities in terms of networks being dual-stack like service capable or not, may turn out hard to comprehend for users and for application/services to cope with. These facts may add other than just technical concerns for operators when planning to roll out dual-stack service offerings.

8.4. Operational Aspects of Running a Network with IPv6 Only Bearers

It is possible to allocate IPv6 only type bearers to mobile hosts in 3GPP networks. IPv6 only bearer type has been part of the 3GPP specification since the beginning. In 3GPP Release-8 (and later) it was defined that a dual-stack mobile host (or when the radio equipment has no knowledge of the host IP stack capabilities) must first attempt to establish a dual-stack bearer and then possibly fall back to single IP version bearer. A Release-8 (or later) mobile host with IPv6 only stack can directly attempt to establish an IPv6 only bearer. The IPv6 only behavior is up to a subscription provisioning or a PDN-GW configuration, and the fallback scenarios do not necessarily cause additional signaling.

Although the bullets below introduce IPv6 to IPv4 address translation and specifically discuss NAT64 technology [I-D.ietf-behave-v6v4-framework], the current 3GPP Release-8

architecture does not describe the use of address translation or NAT64. It is up to a specific deployment whether address translation is part of the network or not. Some operational aspects to consider for running a network with IPv6 only bearers:

- The mobile hosts must have an IPv6 capable stack and a radio interface capable of establishing an IPv6 PDP context or PDN connection.

- The GGSN/PDN-GW must be IPv6 capable in order to support IPv6 bearers. Furthermore, the SGSN/MME must allow the creation of PDP Type or PDN Type of IPv6.

- Many of the common applications are IP version agnostic and hence would work using an IPv6 bearer. However, applications that are IPv4 specific would not work.

- Inter-operator roaming is another aspect which causes issues, at least during the ramp up phase of the IPv6 deployment. If the visited network to which outbound roamers attach to does not support PDP/PDN Type IPv6, then there needs to be a fallback option. The fallback option in this specific case is mostly up to the mobile host to implement. Several cases are discussed in the following sections.

- If and when a mobile host using IPv6 only bearer needs to access to IPv4 Internet/network, a translation of some type from IPv6 to IPv4 has to be deployed in the network. NAT64 (and DNS64) is one solution that can be used for this purpose and works for a certain set of protocols (read TCP and UDP, and when applications actually use DNS for resolving name to IP addresses).

8.5. Restricting Outbound IPv6 Roaming

Roaming was briefly touched upon in Sections 8.2 and 8.4. While there is interest in offering roaming service for IPv6 enabled mobile hosts and subscriptions, not all visited networks are prepared for IPv6 outbound roamers. There are basically two issues. First, the visited network (S4-)SGSN does not support the IPv6 PDP Context or IPv4v6 PDP Context types. These should mostly concern pre-Release-8 networks but there is no definitive rule as the deployed feature sets vary depending on implementations and licenses. Second, the visited network might not be commercially ready for IPv6 outbound roamers, while everything might work technically at the user plane level. This would lead to "revenue leakage" especially from the visited operator point of view (note that the use of visited network GGSN/ PDN-GW does not really exist in real deployments today). Therefore, it might be in the interest of operators to prohibit roaming

selectively within specific visited networks.

Unfortunately, it is not mandatory to implement/deploy 3GPP standards based solution to selectively prohibit IPv6 roaming without also prohibiting other packet services (such as IPv4 roaming). However, there are few possibilities how this can be done in real deployments. The examples given below are either optional and/or vendor specific features to the 3GPP EPC:

- Using Policy and Charging Control (PCC) [3GPP.23.203] functionality and its rules to fail, for example, the bearer authorization when a desired criteria is met. In this case that would be PDN/PDP Type IPv6/IPv4v6 and a specific visited network. The rules can be provisioned either in the home network or locally in the visited network.

- Some Home Location Register (HLR) and Home Subscriber Server (HSS) subscriber databases allow prohibiting roaming in a specific (visited) network for a specified PDN/PDP Type.

The obvious problems are that these solutions are not mandatory, are not unified across networks, and therefore also lack well-specified fall back mechanism from the mobile host point of view.

8.6. Inter-rat Handovers and IP Versions

It is obvious that when operators start to incrementally deploy EPS (and E-UTRAN) along with the existing UTRAN/GERAN, handovers between different radio technologies (inter-rat handovers) become inevitable. In case of inter-rat handovers 3GPP supports the following IP addressing scenarios:

- E-UTRAN IPv4v6 bearer has to map one to one to UTRAN/GERAN IPv4v6 bearer.

- E-UTRAN IPv6 bearer has to map one to one to UTRAN/GERAN IPv6 bearer.

- E-UTRAN IPv4 bearer has to map one to one to UTRAN/GERAN IPv4 bearer.

Other types of configurations are considered network planning mistakes. What the above rules essentially imply is that the network migration has to be planned and subscriptions provisioned based on the lowest common nominator, if inter-rat handovers are desired. For example, if some part of the UTRAN network cannot serve anything but IPv4 bearers, then the E-UTRAN is also forced to provide only IPv4 bearers. Various combinations of subscriber provisioning regarding

IP versions are discussed further in Section 8.7.

8.7. Provisioning of IPv6 Subscribers and Various Combinations During Initial Network Attachment

Subscribers' provisioned PDP/PDN Types have multiple configurations. The supported PDP/PDN Type is provisioned per each APN for every subscriber. The following PDN Types are possible in the HSS for a Release-8 subscription [3GPP.23.401]:

- IPv4v6 PDN Type (note that IPv4v6 PDP Type does not exist in HLR).
- IPv6 only PDN Type
- IPv4 only PDN Type.
- IPv4_or_IPv6 PDN Type (note that IPv4_or_IPv6 PDP Type does not exist in HLR).

A Release-8 dual-stack mobile host must always attempt to establish a PDP/PDN Type IPv4v6 bearer. The same also applies when the modem part of the mobile host does not have exact knowledge whether the host operating system IP stack is a dual-stack capable or not. A mobile host that is IPv6 only capable must attempt to establish a PDP/PDN Type IPv6 bearer. Last, a mobile host that is IPv4 only capable must attempt to establish a PDN/PDP Type IPv4 bearer.

In a case the PDP/PDN Type requested by a mobile host does not match what has been provisioned for the subscriber in the HSS (or HLR), the mobile host possibly falls back to a different PDP/PDN Type. The network (i.e. the MME or the SGSN) is able to inform the mobile host during the network attachment signaling why it did not get the requested PDP/PDN Type. These response/cause codes are documented in [3GPP.24.008][3GPP.24.301]. Possible fall back cases include (as documented in [3GPP.23.401]):

- Requested & provisioned PDP/PDN Types match -> requested.
- Requested IPv4v6 & provisioned IPv6 -> IPv6 and a mobile host receives indication that IPv6-only bearer is allowed.
- Requested IPv4v6 & provisioned IPv4 -> IPv4 and the mobile host receives indication that IPv4-only bearer is allowed.
- Requested IPv4v6 & provisioned IPv4_or_IPv6 -> IPv4 or IPv6 is selected by the MME based on an unspecified criteria. The mobile host may then attempt to establish, based on the mobile host implementation, a parallel bearer of a different PDP/PDN Type.

- o Other combinations cause the bearer establishment to fail.

In addition to PDP/PDN Types provisioned in the HSS, it is also possible for a PDN-GW (and a MME) to affect the final selected PDP/PDN Type:

- o Requested IPv4v6 & configured IPv4 or IPv6 in the PDN-GW -> IPv4 or IPv6. If the MME operator had included the "Dual Address Bearer Flag" into the bearer establishment signaling, then the mobile host receives an indication that IPv6-only or IPv4-only bearer is allowed.

- o Requested IPv4v6 & configured IPv4 or IPv6 in the PDN-GW -> IPv4 or IPv6. If the MME operator had not included the "Dual Address Bearer Flag" into the bearer establishment signaling, then the mobile host may attempt to establish, based on the mobile host implementation, a parallel bearer of different PDP/PDN Type.

If for some reason a SGSN does not understand the requested PDP Type, then the PDP Type is handled as IPv4. If for some reason a MME does not understand the requested PDN Type, then the PDN Type is handled as IPv6.

9. IANA Considerations

This document has no requests to IANA.

10. Security Considerations

This document does not introduce any security related concerns.

11. Summary and Conclusion

The 3GPP network architecture and specifications enable the establishment of IPv4 and IPv6 connections through the use of appropriate PDP context types. The current generation of deployed networks can support dual-stack connectivity if the packet core network elements such as the SGSN and GGSN have the capability. With Release-8, 3GPP has specified a more optimal PDP context type which enables the transport of IPv4 and IPv6 packets within a single PDP context between the mobile station and the gateway.

As devices and applications are upgraded to support IPv6 they can start leveraging the IPv6 connectivity provided by the networks while maintaining the fall back to IPv4 capability. Enabling IPv6

connectivity in the 3GPP networks by itself will provide some degree of relief to the IPv4 address space as many of the applications and services can start to work over IPv6. However without comprehensive testing of different applications and solutions that exist today and are widely used, for their ability to operate over IPv6 PDN connections, an IPv6 only access would cause disruptions.

12. Acknowledgements

The authors thank Shabnam Sultana, Sri Gundavelli, Hui Deng, and Zhenqiang Li, Mikael Abrahamsson, James Woodyatt and Cameron Byrne for their reviews and comments on this document.

13. Informative References

[3GPP.23.060]
3GPP, "General Packet Radio Service (GPRS); Service description; Stage 2", 3GPP TS 23.060 8.8.0, March 2010.

[3GPP.23.203]
3GPP, "Policy and charging control architecture (PCC)", 3GPP TS 23.203 8.11.0, September 2010.

[3GPP.23.401]
3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", 3GPP TS 23.401 10.2.1, January 2011.

[3GPP.23.975]
3GPP, "IPv6 Migration Guidelines", 3GPP TR 23.975 1.1.1, June 2010.

[3GPP.24.008]
3GPP, "Mobile radio interface Layer 3 specification", 3GPP TS 24.008 8.12.0, December 2010.

[3GPP.24.301]
3GPP, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)", 3GPP TS 24.301 8.8.0, December 2010.

[3GPP.29.060]
3GPP, "General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface", 3GPP TS 29.274 8.8.0, April 2010.

[3GPP.29.061]

3GPP, "Interworking between the Public Land Mobile Network (PLMN) supporting packet based services and Packet Data Networks (PDN)", 3GPP TS 29.061 8.5.0, April 2010.

[3GPP.29.274]
3GPP, "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C)", 3GPP TS 29.060 8.11.0, December 2010.

[GSMA.IR.34]
GSMA, "Inter-PLMN Backbone Guidelines", GSMA PRD IR.34.4.9, March 2010.

[I-D.ietf-behave-v6v4-framework]
Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", draft-ietf-behave-v6v4-framework-10 (work in progress), August 2010.

[I-D.ietf-dhc-pd-exclude]
Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", draft-ietf-dhc-pd-exclude-01 (work in progress), January 2011.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

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

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless

Address Autoconfiguration", RFC 4862, September 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

Authors' Addresses

Jouni Korhonen (editor)
Nokia Siemens Networks
Linnoitustie 6
FI-02600 Espoo
FINLAND

Email: jouni.nospam@gmail.com

Jonne Soininen
Renesas Mobile

Email: jonne.soininen@renesasmobile.com

Basavaraj Patil
Nokia
6021 Connection drive
Irving, TX 75039
USA

Email: basavaraj.patil@nokia.com

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
FINLAND

Email: teemu.savolainen@nokia.com

Gabor Bajko
Nokia
323 Fairchild drive 6
Mountain view, CA 94043
USA

Email: gabor.bajko@nokia.com

Kaisu Iisakkila
Renesas Mobile

Email: kaisu.iisakkila@renesasmobile.com

V6ops WG
Internet-Draft
Intended status: Informational
Expires: September 15, 2011

V. Kuarsingh, Ed.
Rogers Communications
Y. Lee
Comcast
O. Vautrin
Juniper Networks
March 14, 2011

6to4 Provider Managed Tunnels
draft-kuarsingh-v6ops-6to4-provider-managed-tunnel-02

Abstract

6to4 Provider Managed Tunnels (6to4-PMT) provide a framework which can help manage 6to4 [RFC3056] tunnels operating an an anycast [RFC3068] configuration. The 6to4-PMT framework is intended to serve as an option to operators to help improve the experience of 6to4 operation when conditions of the network may provide sub-optimal performance or break normal 6to4 operation. 6to4-PMT provides a stable provider prefix and forwarding environment by utilizing existing 6to4 Relays with an added function of IPv6 Prefix Translation. This operation may be particularly important in NAT444 infrastructures where a customer endpoint may be assigned a non-RFC1918 address thus breaking the return path for anycast [RFC3068] based 6to4 operation.

Table of Contents

1. Introduction

6to4 [RFC3056] tunneling along with the anycast operation described in [RFC3068] is widely deployed in modern Operating Systems and off the shelf gateways sold throughout the retail and OEM channels. Anycast [RFC3068] based 6to4 allows for tunneled IPv6 connectivity through IPv4 clouds without explicit configuration of a relay address. Since the overall system utilizes anycast forwarding in both directions, flow paths are difficult to determine, tend to follow separate paths in either direction, and often change based on network conditions. The return path is normally uncontrolled by the local operator and can contribute to poor performance for IPv6, and can also act as a breakage point. Many of the challenges with 6to4 are described in [draft-carpenter-v6ops-6to4-teredo-advisor]. A specific critical use case for problematic anycast 6to4 operation is related to when the consumer endpoints are downstream from a northbound NAT44 function when assigned non-RFC1918 addresses (common future case in wireline networks and very common in wireless networks).

Operators which are actively deploying IPv6 networks and operate legacy IPv4 access environments may want to utilize the existing 6to4 behavior in customer site resident hardware and software as an interim option to reach the IPv6 Internet in advance of being able to offer full native IPv6. Operators may also need to address the brokenness related to 6to4 operation originating from behind a provider NAT function. 6to4-PMT offers a operator the opportunity to utilize IPv6 Prefix Translation to enable deterministic and an unbroken path to and from the Internet for IPv6 based traffic sourced originally from these 6to4 customer endpoints.

6to4-PMT translates the prefix portion of the IPv6 address from the 6to4 generated prefix to a provider assigned prefix which is used to represent the source. This translation will then provide a stable forward and return path for the 6to4 traffic by allowing the existing IPv6 routing and policy environment to control the traffic. 6to4-PMT is primarily intended to be used in a stateless manner to maintain many of the elements inherent in normal 6to4 operation. Alternatively, 6to4-PMT can be used in a stateful translation mode should the operator choose this option.

2. Motivation

Many operators endeavor to deploy IPv6 as soon as possible so as to ensure uninterrupted connectivity to all Internet applications and content through the IPv4 to IPv6 transition process. The IPv6 preparations within these organizations are often faced with both

financial challenges and timing issues related to deploying IPv6 to the network edge and related transition technologies. Many of the new technologies addressing IPv4 to IPv6 transition will require the replacement of the customer CPE to support technologies like 6RD [RFC5969], Dual Stack Lite [draft-ietf-softwire-dual-stack-lite] and Native Dual Stack.

Operators face a number of challenges related to home equipment replacement. Operator initiated replacement of this equipment will take time due to the nature of mass equipment refresh programs or may require the consumer to replace their own gear. Replacing consumer owned and operated equipment, compounded by the fact that there is also a general unawareness of what IPv6 is, also adds the the challenges faced by operators. It is also important to note that 6to4 is found in much of the equipment in networks today which do not as of yet, or will not, support 6RD and/or Native Dual Stack.

Operators may still be motivated to provide a form of IPv6 connectivity to customers and would want to mitigate potential issues related to IPv6-only deployments elsewhere on the Internet. Operators also need to mitigate issues related to the fact that 6to4 operation often is on by default and may be subject to erroneous behavior. The undesired behavior may be related to the use of non-RFC1918 addresses on CPE equipment which operate behind large NATs, or other conditions as described in a general advisory as laid out in [draft-carpenter-v6ops-6to4-teredo-advisory].

6to4-PMT allows a operator to help mitigate such challenges by leveraging the existing 6to4 deployment base, while maintaining operator control of access to the IPv6 Internet. It is intended for use when better options, such as 6RD or native IPv6, are not yet viable. One of key objectives of 6to4-PMT is to also help reverse the negative impacts of 6to4 in NAT444 environments. The 6to4-PMT operation can also be used immediately and the default parameters are often enough to allow it to operate in a 6to4-PMT environment. Once native IPv6 is available to the endpoint, the 6to4-PMT operation is no longer needed and will cease to be used based on correct address selection behaviors in end hosts [RFC3484].

6to4-PMT thus helps operators remove the impact of 6to4 in NAT444 environments, deals with the fact that 6to4 is often on by default, allows access to IPv6-only endpoints from IPv4-only addressed equipment and provides relief from may challenges related to mis-configuations in other networks. Due to the simple nature of 6to4-PMT, it can also be implemented in a cost effective and simple manner allowing operators to concentrate their energy on deploying native IPv6.

3. 6to4 Provider Managed Tunnels

3.1. 6to4 Provider Managed Tunnel Model

The 6to4 managed tunnel model behaves like a standard 6to4 service between the customer IPv6 host or gateway and the 6ot4-PMT Relay (within the provider domain). The 6to4-PMT Relay shares properties with 6RD [RFC5969] by decapsulating and forwarding embedded IPv6 flows, within an IPv4 packet, to the IPv6 Internet. The model provides an additional function which translates the source 6to4 prefix to a provider assigned prefix which is not found in 6RD [RFC5969] or traditional 6to4 operation.

The 6to4-PMT Relay is intended to provide a stateless (or stateful) mapping of the 6to4 prefix to a provider supplied prefix by mapping the embedded IPv4 address in the 6to4 prefix to the provider prefix.

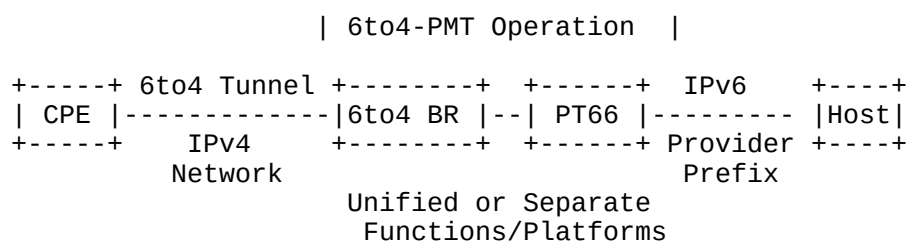


Figure 1: 6to4-PMT Functional Model

This mode of operation is seen as beneficial when compared to broken 6to4 paths and or environments where 6to4 operation may be functional but highly degraded.

3.2. Traffic Flow

Traffic in the 6to4-PMT model is intended to be controlled by the operator’s IPv6 peering operations. Egress traffic is managed through outgoing routing policy, and incoming traffic is influenced by the operator assigned prefix advertisements.

The routing model is as predictable as native IPv6 traffic and legacy IPv4 based traffic. Figure 2 provides a view of the routing topology needed to support this relay environment. The diagram references PrefixA as 2002::/16 and PrefixB as the example 2001:db8::/32.

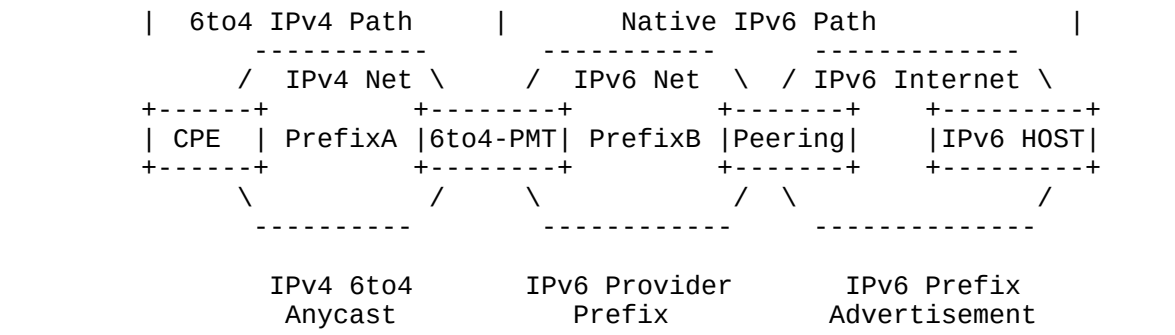


Figure 2: 6to4-PMT Flow Model

Traffic between two 6to4 enabled devices would use the IPv4 path for communication according to RFC3056. 6to4-PMT is intended to be deployed in conjunction with the 6to4 relay function in an attempt to help simplify it’s deployment. The model can also provide the ability for an operator to forward both 6to4-PMT (translated) and normal 6to4 flows (untranslated) simultaneously based on policy.

3.3. Prefix Translation

The IPv6 Prefix Translation is a key part of the system as a whole. The 6to4-PMT framework is a combination of two concepts: 6to4 [RFC3056] and IPv6 Prefix Translation. IPv6 Prefix Translation has some similarities to concepts discussed in [draft-mrw-nat66]. The only change in this particular case is that the provider would build specific rules on the translator to map the 6to4 prefix to an appropriate provider assigned prefix.

The provider can use any prefix mapping strategy they so choose, but the simpler the better. Simple direct bit mapping can be used such as in Figure 2, or more advanced forms of translation can be used to achieve higher address compression.

Figure 2 shows a 6to4 Prefix with a Subnet-ID of "0000" mapped to a provider globally unique prefix (2001:db8::/32). With this simple form of translation, there is support for only one Subnet-ID per provider assigned prefix. In characterization of deployed OSs and gateways, a subnet-id of "0000" is the most common default case.

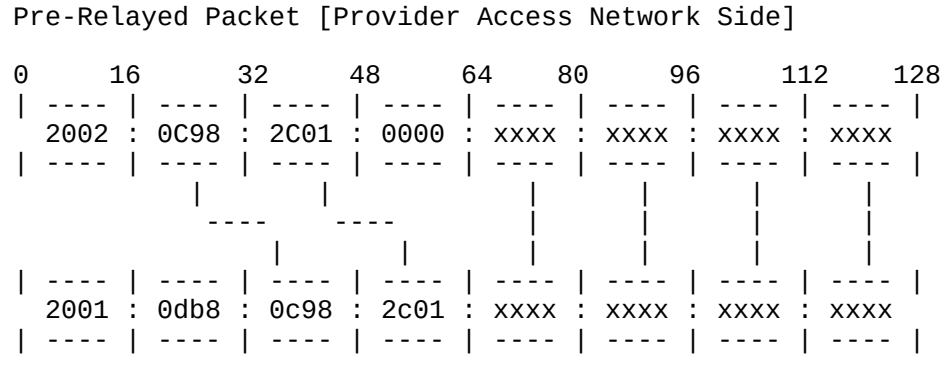


Figure 3: 6to4-PMT Prefix Mapping

Additional prefix compression techniques can be used such as those described in [draft-tremblay-pt66ac]. These techniques would allow for a more flexible implementation potentially supporting more Subnet-IDs per provider prefix.

3.4. Translation State

It is preferred that the overall system use deterministic prefix translation mappings such that stateless operation can be implemented. This allows the provider to place N number of relays within the network without the need to manage translation state.

If stateful operation is chosen, the operation would need to validate state and routing requirements particular to that type of deployment. The full body of considerations for this type of deployment are not within this scope of this document.

4. Deployment Considerations and Requirements

4.1. Customer Opt-out

A provider enabling this function should provide a method to allow customers to opt-out of such a service should the customer choose to maintain normal 6to4 operation irrespective of degraded performance. In cases where the customer is behind a NAT44 device (Provider CGN), the customer would not be advised to opt-out and can also be assisted to turn off 6to4.

Since the 6to4-PMT system is targeted at customers who are relatively unaware of IPv6 and IPv4, and normally run network equipment with a

default configuration, an opt-out strategy is recommended. This method provides the 6to4-PMT operation for non-IPv6 savvy customers whose equipment may turn on 6to4 automatically and allows savvy customers to easily configure they way around the PMT function.

Capable customers can also disable anycast based 6to4 entirely and use traditional 6to4 or other tunneling mechanisms if they are so inclined. This is not considered the normal case, and most endpoints with auto-6to4 operation will be subject to 6to4-PMT operation since most users are unaware of it’s existence. 6to4-PMT is targeted as an option for stable IPv6 connectivity for average consumers.

4.2. ISP Shared Space Considerations

6to4-PMT operation can also be used to mitigate a known problem with 6to4 when ISP Shared Space [draft-weil-shared-transition-space-request-01] or public but non-routed IPv4 space is used. Public but un-routed address space would cause many deployed OSs and network equipment to potentially auto-enable 6to4 operation even without a valid return path (such as behind NAT44 provider function). Operators’ desire to use public but un-routed IP space is considered highly likely based on points made in [draft-weil-shared-transition-space-request] and in reports such as [wide-tr-kato-as112-rep-01].

Such hosts, in normal cases, would send 6to4 traffic to the IPv6 Internet via the IPv4 anycast relay, which would in fact provide broken IPv6 connectivity since the return path is based on an address that is not routed or assigned to the source Network. The use of 6to4-PMT would help reverse these effects by translating the 6to4 prefix to a provided assigned prefix, masking this automatic and undesired behavior.

4.3. End to End Transparency

6to4-PMT mode operation removes the traditional end to end transparency of 6to4. Remote hosts would connect to a translated IPv6 address versus the original 6to4 based prefix. This can be seen as a disadvantage of the 6to4-PMT system. This lack of transparency should also be contrasted with the normal operating state of 6to4 which provides uncontrolled and often high latency prone connectivity. The lack of transparency is however a better form of operation when extreme poor performance, broken IPv6 connectivity, or no IPv6 connectivity is considered as the alternative.

4.4. Routing Requirements

The provider would need to advertise the anycast IP range within the IPv4 routing environment (service customers of interest) to attract the 6to4 upstream traffic. To control this environment and make sure all northbound traffic lands on a provider BR, the operator may filter the anycast range from being advertised from customer endpoints.

The provider would not be able to control route advertisements inside the customer domain, but this use case is out of this document's scope. It is likely in this case the end network/customer understands IPv6 operation and is maintaining their own environment.

The provider would also likely want to advertise the 2002::/16 range within their own network to help bridge within their own network (Native IPv6 to 6to4-IPv6 based endpoint).

4.5. Relay Deployments

The 6to4-PMT function can be deployed onto existing 6to4 relays (if desired) to help minimize network complexity. If used on Linux based relays, 6to4-PMT can be a low cost add-on which can help align normal 6to4 and 6to4-PMT operation. The only additional considerations beyond normal 6to4 relay operation would include the need to route specific IPv6 address ranges to the IPv6 side interface to manage return traffic.

5. IANA Considerations

No IANA considerations are defined at this time.

6. Security Considerations

6to4-PMT operation would be subject to the same security concerns as normal 6to4 operation and with the operation of tunnels.

7. Acknowledgements

Thanks to the following people for their textual contributions and/or guidance on 6to4 deployment considerations: Dan Wing, Wes George, Scott Beuker, JF Tremblay, John Brzozowski, and Chris Donley

Additional thanks to the following for assisting with the coding and testing of 6to4-PMT: Marc Blanchet, John Cianfarani, and Nik Lavorato

8. References

8.1. Normative References

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

8.2. Informative References

[I-D.ietf-softwire-dual-stack-lite] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 2011.

[I-D.mrw-nat66] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", draft-mrw-nat66-10 (work in progress), March 2011.

[I-D.tremblay-pt66ac] Tremblay, J. and S. Beuker, "Addressing bit compression for stateless IPv6 prefix translation", draft-tremblay-pt66ac-00 (work in progress), November 2010.

[I-D.weil-shared-transition-space-request] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA Reserved IPv4 Prefix for Shared Transition Space", draft-weil-shared-transition-space-request-01 (work in progress), November 2010.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

Authors' Addresses

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

Email: victor.kuarsingh@rci.rogers.com
URI: http://www.rogers.com

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: yiu_lee@cable.comcast.com
URI: http://www.comcast.com

Olivier Vautrin
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA 94089
U.S.A.

Email: olivier@juniper.net
URI: http://www.juniper.net

v6ops
Internet-Draft
Intended status: Informational
Expires: January 11, 2013

V. Kuarsingh, Ed.
Rogers Communications
Y. Lee
Comcast
O. Vautrin
Juniper Networks
July 10, 2012

6to4 Provider Managed Tunnels
draft-kuarsingh-v6ops-6to4-provider-managed-tunnel-07

Abstract

6to4 Provider Managed Tunnels (6to4-PMT) provide a framework which can help manage 6to4 tunnels operating in an anycast configuration. The 6to4-PMT framework is intended to serve as an option for operators to help improve the experience of 6to4 operation when conditions of the network may provide sub-optimal performance or break normal 6to4 operation. 6to4-PMT provides a stable provider prefix and forwarding environment by utilizing existing 6to4 relays with an added function of IPv6 Prefix Translation. This operation may be particularly important in NAT444 infrastructures where a customer endpoint may be assigned a non-RFC1918 address thus breaking the return path for anycast based 6to4 operation. 6to4-PMT has successfully been used in a production network, has been implemented as open source code, and implemented by a major routing vendor.

Table of Contents

1. Introduction

6to4 [RFC3056] tunnelling along with the anycast operation described in [RFC3068] is widely deployed in modern Operating Systems and off the shelf gateways sold throughout the retail and OEM channels. Anycast [RFC3068] based 6to4 allows for tunnelled IPv6 connectivity through IPv4 clouds without explicit configuration of a relay address. Since the overall system utilizes anycast forwarding in both directions, flow paths are difficult to determine, tend to follow separate paths in either direction, and often change based on network conditions. The return path is normally uncontrolled by the local operator and can contribute to poor performance for IPv6, and can also act as a breakage point. Many of the challenges with 6to4 are described in [RFC6343]. A specific critical use case for problematic anycast 6to4 operation is related to conditions where the consumer endpoints are downstream from a northbound CGN [RFC6264] function when assigned non-RFC1918 IPv4 addresses, which are not routed on interdomain links.

Operators which are actively deploying IPv6 networks and operate legacy IPv4 access environments may want to utilize the existing 6to4 behaviour in customer site resident hardware and software as an interim option to reach the IPv6 Internet in advance of being able to offer full native IPv6. Operators may also need to address the brokenness related to 6to4 operation originating from behind a provider NAT function. 6to4-PMT offers an operator the opportunity to utilize IPv6 Prefix Translation to enable deterministic traffic flow and an unbroken path to and from the Internet for IPv6 based traffic sourced originally from these 6to4 customer endpoints.

6to4-PMT translates the prefix portion of the IPv6 address from the 6to4 generated prefix to a provider assigned prefix which is used to represent the source. This translation will then provide a stable forward and return path for the 6to4 traffic by allowing the existing IPv6 routing and policy environment to control the traffic. 6to4-PMT is primarily intended to be used in a stateless manner to maintain many of the elements inherent in normal 6to4 operation. Alternatively, 6to4-PMT can be used in a stateful translation mode should the operator choose this option.

2. Motivation

Many operators endeavour to deploy IPv6 as soon as possible so as to ensure uninterrupted connectivity to all Internet applications and content through the IPv4 to IPv6 transition process. The IPv6 preparations within these organizations are often faced with both financial challenges and timing issues related to deploying IPv6 to

the network edge and related transition technologies. Many of the new technologies available for IPv4 to IPv6 transition will require the replacement of the customer CPE to support technologies like 6RD [RFC5969], Dual-Stack Lite [RFC6333] and Native Dual Stack.

Operators face a number of challenges related to home equipment replacement. Operator initiated replacement of this equipment will take time due to the nature of mass equipment refresh programs or may require the consumer to replace their own gear. Replacing consumer owned and operated equipment, compounded by the fact that there is also a general unawareness of what IPv6 is, also adds to the challenges faced by operators. It is also important to note that 6to4 is found in much of the equipment found in networks today which do not as of yet, or will not, support 6RD and/or Native IPv6.

Operators may still be motivated to provide a form of IPv6 connectivity to customers and would want to mitigate potential issues related to IPv6-only deployments elsewhere on the Internet. Operators also need to mitigate issues related to the fact that 6to4 operation often is on by default and may be subject to erroneous behaviour. The undesired behaviour may be related to the use of non-RFC1918 addresses on CPE equipment which operate behind large operator NATs, or other conditions as described in a general advisory as laid out in [RFC6343].

6to4-PMT allows an operator to help mitigate such challenges by leveraging the existing 6to4 deployment base, while maintaining operator control of access to the IPv6 Internet. It is intended for use when better options, such as 6RD or Native IPv6, are not yet viable. One of key objectives of 6to4-PMT is to also help reverse the negative impacts of 6to4 in CGN environments. The 6to4-PMT operation can also be used immediately with the default parameters which are often enough to allow it to operate in a 6to4-PMT environment. Once native IPv6 is available to the endpoint, the 6to4-PMT operation is no longer needed and will cease to be used based on correct address selection behaviours in end hosts [RFC3484].

6to4-PMT thus helps operators remove the impact of 6to4 in CGN environments, deals with the fact that 6to4 is often on by default, allows access to IPv6-only endpoints from IPv4-only addressed equipment and provides relief from many challenges related to mis-configurations in other networks which control return flows via foreign relays. Due to the simple nature of 6to4-PMT, it can also be implemented in a cost effective and simple manner allowing operators to concentrate their energy on deploying Native IPv6.

3. 6to4 Provider Managed Tunnels

3.1. 6to4 Provider Managed Tunnel Model

The 6to4 managed tunnel model behaves like a standard 6to4 service between the customer IPv6 host or gateway and the 6to4-PMT Relay (within the provider domain). The 6to4-PMT Relay shares properties with 6RD [RFC5969] by decapsulating and forwarding encapsulated IPv6 flows within an IPv4 packet, to the IPv6 Internet. The model provides an additional function which translates the source 6to4 prefix to a provider assigned prefix which is not found in 6RD [RFC5969] or traditional 6to4 operation.

The 6to4-PMT Relay is intended to provide a stateless (or stateful) mapping of the 6to4 prefix to a provider supplied prefix.

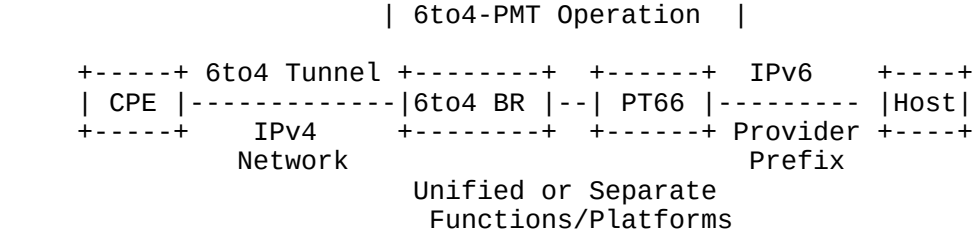


Figure 1: 6to4-PMT Functional Model

This mode of operation is seen as beneficial when compared to broken 6to4 paths and/or environments where 6to4 operation may be functional but highly degraded.

3.2. Traffic Flow

Traffic in the 6to4-PMT model is intended to be controlled by the operator’s IPv6 peering operations. Egress traffic is managed through outgoing routing policy, and incoming traffic is influenced by the operator assigned prefix advertisements using normal interdormain routing functions.

The routing model is as predictable as native IPv6 traffic and legacy IPv4 based traffic. Figure 2 provides a view of the routing topology needed to support this relay environment. The diagram references PrefixA as 2002::/16 and PrefixB as the example 2001:db8::/32.

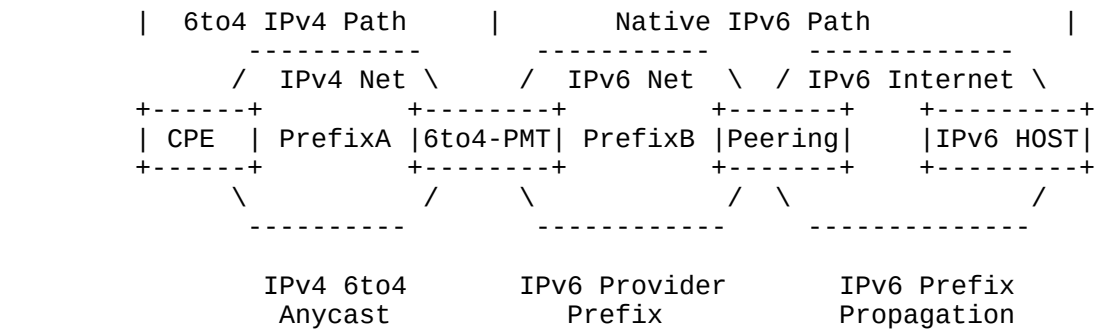


Figure 2: 6to4-PMT Flow Model

Traffic between two 6to4 enabled devices would use the IPv4 path for communication according to RFC3056 unless the local host still prefers traffic via a relay. 6to4-PMT is intended to be deployed in conjunction with the 6to4 relay function in an attempt to help simplify it's deployment. The model can also provide the ability for an operator to forward both 6to4-PMT (translated) and normal 6to4 flows (untranslated) simultaneously based on configured policy.

3.3. Prefix Translation

IPv6 Prefix Translation is a key part of the system as a whole. The 6to4-PMT framework is a combination of two concepts: 6to4 [RFC3056] and IPv6 Prefix Translation. IPv6 Prefix Translation, as used in 6to4-PMT, has some similarities to concepts discussed in [RFC6296]. 6to4-PMT would provide prefix translation based on specific rules configured on the translator which maps the 6to4 2002::/16 prefix to an appropriate provider assigned prefix. In most cases, a ::/32 prefix would work best in 6to4-PMT which matches common RIR prefix assignments to operators.

The provider can use any prefix mapping strategy they so choose, but the simpler the better. Simple direct bit mapping can be used, or more advanced forms of translation should the operator want to achieve higher address compression. More advanced forms of translation may require the use of stateful translation.

Figure 3 shows a 6to4 Prefix with a Subnet-ID of "0000" mapped to a provider assigned globally unique prefix (2001:db8::/32). With this simple form of translation, there is support for only one Subnet-ID per provider assigned prefix. In characterization of deployed OSs and gateways, a Subnet-ID of "0000" is the most common default case followed by Subnet-ID "0001". Use of Subnet-ID can be referenced in [RFC4291]. It should be noted that in normal 6to4 operation the endpoint (network) has access to 65,536 (16-bits) Subnet IDs. In the

6to4-PMT case as described above using the mapping in Figure 3, all but the one Subnet-ID used for 6to4-PMT would still operate under normal 6to4 operation.

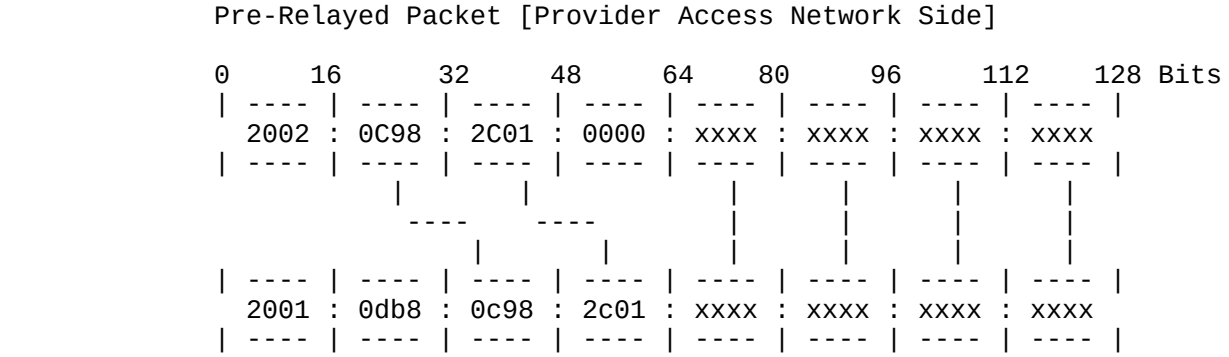


Figure 3: 6to4-PMT Prefix Mapping

3.4. Translation State

It is preferred that the overall system use deterministic prefix translation mappings such that stateless operation can be implemented. This allows the provider to place N number of relays within the network without the need to manage translation state. Deterministic translation also allows a customer to use inward services using the translated (provider prefix) address.

If stateful operation is chosen, the operator would need to validate state and routing requirements particular to that type of deployment. The full body of considerations for this type of deployment are not within this scope of this document.

4. Deployment Considerations and Requirements

4.1. Customer Opt-out

A provider enabling this function should provide a method to allow customers to opt-out of such a service should the customer choose to maintain normal 6to4 operation irrespective of degraded performance. In cases where the customer is behind a CGN device, the customer would not be advised to opt-out and can also be assisted to turn off 6to4.

Since the 6to4-PMT system is targeted at customers who are relatively

unaware of IPv6 and IPv4, and normally run network equipment with a default configuration, an opt-out strategy is recommended. This method provides 6to4-PMT operation for non-IPv6 savvy customers whose equipment may turn on 6to4 automatically and allows savvy customers to easily configure their way around the 6to4-PMT function.

Capable customers can also disable anycast based 6to4 entirely and use traditional 6to4 or other tunnelling mechanisms if they are so inclined. This is not considered the normal case, and most endpoints with auto-6to4 functions will be subject to 6to4-PMT operation since most users are unaware of it’s existence. 6to4-PMT is targeted as an option for stable IPv6 connectivity for average consumers.

4.2. Shared CGN Space Considerations

6to4-PMT operation can also be used to mitigate a known problem with 6to4 when shared address space [RFC6598] or Global Unicast Addresses (GUA) are used behind a CGN and not routed on the Internet. Non-RFC1918, yet un-routed (on interdomain links) address space would cause many deployed OSs and network equipment to potentially auto-enable 6to4 operation even without a valid return path (such as behind a CGN function). The Operators’ desire to use non-RFC1918 addresses, such as shared address space [RFC6598], is considered highly likely based on real world deployments.

Such hosts, in normal cases, would send 6to4 traffic to the IPv6 Internet via the anycast relay, which would in fact provide broken IPv6 connectivity since the return path flow is built using an IPv4 address that is not routed or assigned to the source Network. The use of 6to4-PMT would help reverse these effects by translating the 6to4 prefix to a provider assigned prefix, masking this automatic and undesired behaviour.

4.3. End to End Transparency

6to4-PMT mode operation removes the traditional end to end transparency of 6to4. Remote hosts would connect to a 6to4-PMT serviced host using a translated IPv6 address versus the original 6to4 address based on the 2002::/16 well-known prefix. This can be seen as a disadvantage of the 6to4-PMT system. This lack of transparency should also be contrasted with the normal operating state of 6to4 which provides uncontrolled and often high latency prone connectivity. The lack of transparency is however a better form of operation when extreme poor performance, broken IPv6 connectivity, or no IPv6 connectivity is considered as the alternative.

4.4. Path MTU Discovery Considerations

The MTU will be subject to a reduced value due to standard 6to4 tunnelling operation. Under normal 6to4 operation, the 6to4 service agent would send an ICMP Packet Too Big Message as part of Path MTU Discovery as described in [RFC4443] and [RFC1981] respectively. In 6to4-PMT operation, the PMT Service agent should be aware of the reduced 6to4 MTU and send ICMP messages using the translated address accordingly.

It is also possible to pre-constrain the MTU at the upstream router from the 6to4-PMT service agents which would then have the upstream router send the appropriate ICMP Packet Too Big Messages.

4.5. Checksum Management

Checksum management for 6to4-PMT can be implemented in one of two ways. The first deployment model is based on the stateless 6to4-PMT operational mode. In this case, checksum modifications are made using the method described in [RFC3022] section 4.2. The checksum is modified to match the parameters of the translated address of the source 6to4-PMT host. In the second deployment model where stateful 6to4-PMT translation is used, the vendor can implement checksum neutral mappings as defined in [RFC6296].

4.6. Application Layer Gateways

Vendors can choose to deploy ALGs on their platforms that perform 6to4-PMT if they so choose. No ALGs were deployed as part of the open source and vendor product deployments of 6to4-PMT. In the vendor deployment case, the same rules were used as with their NPTv6 [RFC6296] base code.

4.7. Routing Requirements

The provider would need to advertise the well-known IP address range used for normal anycast 6to4 [RFC3068] operation within the local IPv4 routing environment. This advertisement would attract the 6to4 upstream traffic to a local relay. To control this environment and make sure all northbound traffic lands on a provider controlled relay, the operator may filter the anycast range from being advertised from customer endpoints toward the local network (upstream propagation).

The provider would not be able to control route advertisements inside the customer domain, but that use case is not in scope for this document. It is likely in that case the end network/customer understands 6to4 and is maintaining their own relay environment and

therefore would not be subject to the operators 6to4 and/or PMT operation.

The provider would also likely want to advertise the 2002::/16 range within their own network to help bridge traditional 6to4 traffic within their own network (Native IPv6 to 6to4-PMT based endpoint). It would also be advised that the local 6to4-PMT operator not leak the well-known 6to4 anycast IPv4 prefix to neighbouring Autonomous Systems to prevent PMT operation for neighbouring networks. Policy configuration on the local 6to4-PMT relay can also be used to disallow PMT operation should the local provider service downstream customer networks.

4.8. Relay Deployments

The 6to4-PMT function can be deployed onto existing 6to4 relays (if desired) to help minimize network complexity and cost. 6to4-PMT has already been developed on Linux based platforms which are package add-ons to the traditional 6to4 code. The only additional considerations beyond normal 6to4 relay operation would include the need to route specific IPv6 provider prefix ranges used for 6to4-PMT operation towards peers and transit providers.

5. IANA Considerations

No IANA considerations are defined at this time.

6. Security Considerations

6to4-PMT operation would be subject to the same security concerns as normal 6to4 operation. 6to4-PMT is also not plainly perceptible by external hosts and local entities appear as Native IPv6 hosts to the external hosts.

7. Acknowledgements

Thanks to the following people for their textual contributions and/or guidance on 6to4 deployment considerations: Dan Wing, Wes George, Scott Beuker, JF Tremblay, John Brzozowski, Chris Metz and Chris Donley

Additional thanks to the following for assisting with the coding and testing of 6to4-PMT: Marc Blanchet, John Cianfarani, Tom Jefferd, Nik Lavorato, Robert Hutcheon and Ida Leung

8. References

8.1. Normative References

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

8.2. Informative References

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

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

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264, June 2011.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

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

[RFC6343] Carpenter, B., "Advisory Guidelines for 6to4 Deployment", RFC 6343, August 2011.

[RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and

M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.

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

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: yiu_lee@cable.comcast.com
URI: http://www.comcast.com

Olivier Vautrin
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA 94089
U.S.A.

Email: olivier@juniper.net
URI: http://www.juniper.net

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: April 24, 2011

Qiong Sun
Chongfeng Xie
China Telecom
March 15, 2011

LAFT6: Lightweight address family transition for IPv6
draft-sun-v6ops-laft6-01.txt

Abstract

With the approaching exhaustion of IPv4 address space, large-scale ISPs are now facing the option to deploy IPv6 in a timely manner. However, most existing IPv6 transition solutions have tradeoff between scalability and efficiency. This draft proposes a lightweight address family transition mechanism named LAFT6. It only needs to maintain per-subscriber state entries in core network and there is no specific address format requirement for users' IPv6 address. It is a lightweight solution in terms of state-management, addressing and routing. The experimental results have shown that LAFT6 is scalable and can be rapidly deployed in commercial ISP network.

Status of this Memo

Table of Contents

1. Introduction

Global IPv4 address exhaustion is becoming reality. The dramatic growth of the Internet is accelerating the consumption of available IPv4 addresses, which makes the address shortage problem even worse. It is widely accepted that IPv6 is the only answer to solve the address shortage problem and sustain the long-term growth of the Internet. However, IPv6 deployment is a huge systematic project and a lot of challenges will arise especially in large SP operational network.

In order to facilitate smooth migration to IPv6-based Internet, many factors need to be taken into consideration, e.g. rapid deployment, scalability, backward compatibility, legal traceability and IPv6 encouragement. Thereinto, high scalability and efficiency are two factors which cannot be easily accomplished in the same time.

Currently, most existing IPv6 transition mechanisms can be wildly divided into stateful and stateless solutions. Stateful ones, e.g. DS-Lite[I-D.ietf-softwire-dual-stack-lite], NAT64 [I-D.ietf-behave-v6v4-xlate-stateful], etc., need to keep per-session state in CGN. This will result in severe scalability problem especially for large-scale ISP networks. Moreover, the dynamic feature of session-based states will also bring a great burden on load balancing with state synchronization and legal traceability.

While for stateless solutions, it has strict IPv6 address-format restriction in order to achieve algorithm-based translation. It is very scalable, simple and straightforward; however, it would also have impact on existing address allocation systems, CPE prefix delegation models and routing systems. Since these IPv6 addresses with embedded port index are not continuous anymore, existing DHCPv6 server have to introduce additional function to deal with port-range allocation, either by specifying individual IPv6 address for each end subscriber manually in DHCPv6 pool configuration, or taking modifications for automatic configuration. And then, CPE should re-allocate IPv6 address to end-user based on PD prefix, and announce individual IPv6 routing entry to access

routers. These functionalities have not been fully supported by existing systems.

Therefore, existing solutions do not have a good tradeoff between stateful and stateless ones.

For large scale operators, it is of vital importance to achieve high scalability and simplicity in core network. It is one of the key principles in the overall development of Internet and it would also be very important in the future. Although it is inevitable to multiplex IPv4 address with port range in IPv4/IPv6 coexistence period when the common problems discussed in [I-D.ietf-intarea-shared-addressing-issues] could not be totally avoided, a lightweight state-management solution is still encouraged. It could simplify the packet processing procedure, state synchronization and traffic logging, etc., compared to session-based stateful solution.

Furthermore, it is also very important for network operators to adopt flexible addressing. A solution with no specific addressing requirement can make use of existing IPv6 addressing and routing model as much as possible. As a result, it can be rapidly deployed in operational network when facing pressing IPv4 address shortage problem. Network operators could further define more flexible addressing plans according to different service requirement.

In this document, we propose a scalable lightweight solution named LAFT6. It only needs to maintain per-subscriber state entries in core network and there is no specific address format requirement for each IPv6 address.

2. Terminologies

LAFT-CGW LAFT Customer-side Gateway

LAFT-NGW LAFT Network-side Gateway

Addr4-user IPv4 address for end node

Addr4-CGW-pub Multiplex Public IPv4 address for LAFT-CGW

Addr6-user IPv6 address for end node

Addr6-CGW IPv6 address for LAFT-CGW

Addr6-NGW IPv6 address for LAFT-CGW

Port-range-CGW Port range of LAFT-CGW

Pref64 Prefix for translation

4to4 table: binding table in LAFT-CGW for IPv4 sources with IPv4 receivers

6to4 table: binding table in LAFT-CGW for IPv6 sources with IPv4 receivers

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

3. LAFT6 Overview

3.1. Features of LAFT6

Instead of relying on a large carrier-grade session-based NAT, LAFT6 solution is built on IPv4-in-IPv6 tunnels to reach a lightweight subscriber-based NAT in the carrier side. Two major features of LAFT6 are lightweight state management and addressing.

For lightweight state management, LAFT6-NGW only needs to maintain the mapping pair between IPv4 address (denoted by Addr4-CGW-pub), available port range (denoted by Port-range-CGW) with IPv6 address(denoted by Addr6-CGW)for each subscriber. It would be stable during one login process for each subscriber. Thus, it could dramatically reduce the size of state database compared to session-based solutions, e.g. DS-Lite, NAT444, NAT64, etc. There are numbers of benefits to this feature:

- The state management in Carrier-grade NAT is largely simplified, including searching, inserting and deleting process, not only due to the fact that the size of state database has been reduced to a great extend, but also the number of dimension for each state has been decreased from 5-dimensional session-based tuple to 3-dimensional subscriber-based tuple (IPv6 address, IPv4 address and port range). Therefore, it can easily support larger amount of subscribers when deployed in the same placement than session-based solutions.

- It can simplify the complexity in traffic logging system. It only needs to record IPv4 address, available port range, IPv6 address and time stamp for each subscriber. While for session-based solution, logging system needs to record the 6-dimensional tuple including IPv4 source address, IPv4 destination address, source port, destination port, protocol and timestamp, other than solutions taking into account the advice given in [I-D.behave-natx4-log-reduction].

- State synchronization and HA (High Availability) is easier to achieve since the binding state for a specific subscriber is more stable during the whole login process.

For lightweight addressing, LAFT6 has no additional requirements on IPv6 address format. In this way, the addressing and routing of the service provider access network is simplified by leveraging existing IPv6 addressing and routing system. And more flexible addressing for different services and applications can be achieved conveniently. As a result, LAFT6 can be deployed rapidly in operational network.

3.2. Deployment scenario

The LAFT6 function is implemented in a customer side gateway (LAFT-CGW) and a carried grade gateway (LAFT-NGW) (as depicted in Figure1).

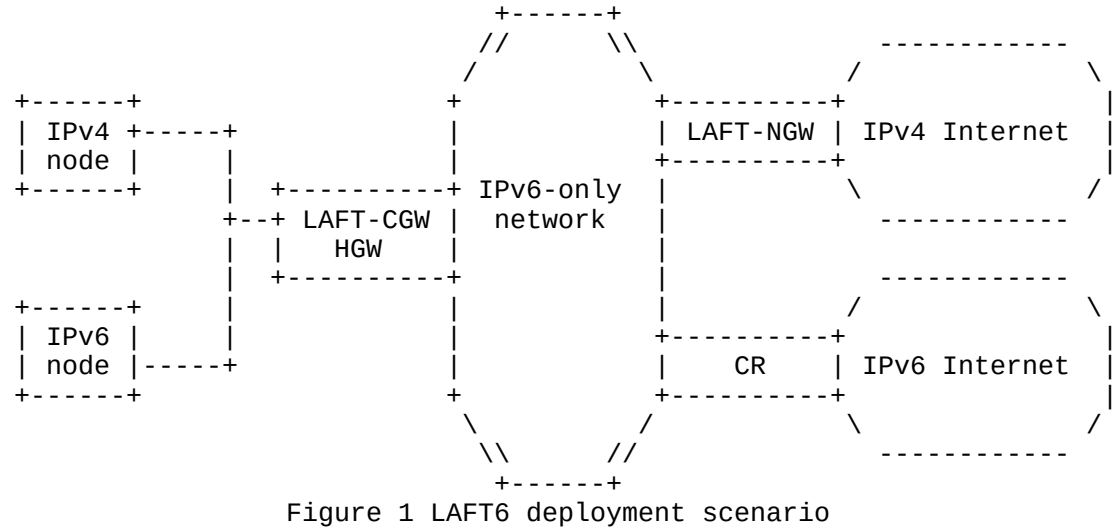


Figure 1 LAFT6 deployment scenario

It is mainly designed for end nodes with a customer gateway in broadband access network. For example, in the future home network, it would be common that there are IPv4-only computers, dual-stack computers,

IPv6-only sensors located behind the same home gateway. Therefore, LAFT-CGW is designed to accommodate different scenarios gracefully and it is up to LAFT-CGW to determine which scenario it would like to support, while LAFT-NGW is simple and it can deal with different scenarios in the same way.

LAFT6 can be applied to both scenarios where IPv4 sources communicate IPv4 receivers (denoted as 4to4) and IPv6 sources communicate with IPv4 receivers (denoted as 6to4). It can also support IPv6 sources communicating with IPv6 receivers in a native way without further treatment.

In home Local Area network, which is characterized by the presence of a home gateway, or CPE, provisioned only with IPv6 by the service provider, a LAFT CPE is an IPv6-aware device with a LAFT-CGW Interface implemented in the WAN interface. LAFT-NGW element is implemented in a device which has (at least) two interfaces, an IPv4 interface connected to the IPv4 network, and an IPv6 interface connected to the IPv6 network. It is usually deployed in core network.

3.3. LAFT6 solution overview

In the initial phase, the end user and LAFT-CGW(e.g. located in Home gateway) would get their individual IPv6 address (denoted as Addr6-user and Addr6-CGW) through traditional PPPoE, IPoE, etc. Besides, the end user would also get its private IPv4 address (denoted as Addr4-user) which is determined by LAFT-CGW only. After that, LAFT-CGW would be allocated a port range (denoted as Port-range-CGW), a public address (denoted as Addr4-CGW-pub) via PCP protocol [I-D.tsou-pcp-natcoord], and notify LAFT-NGW with its own IPv6 address (Addr6-CGW). LAFT-NGW would keep 3-tuple, which includes Addr4-CGW-pub, Port-range-CGW and Addr6-CGW.

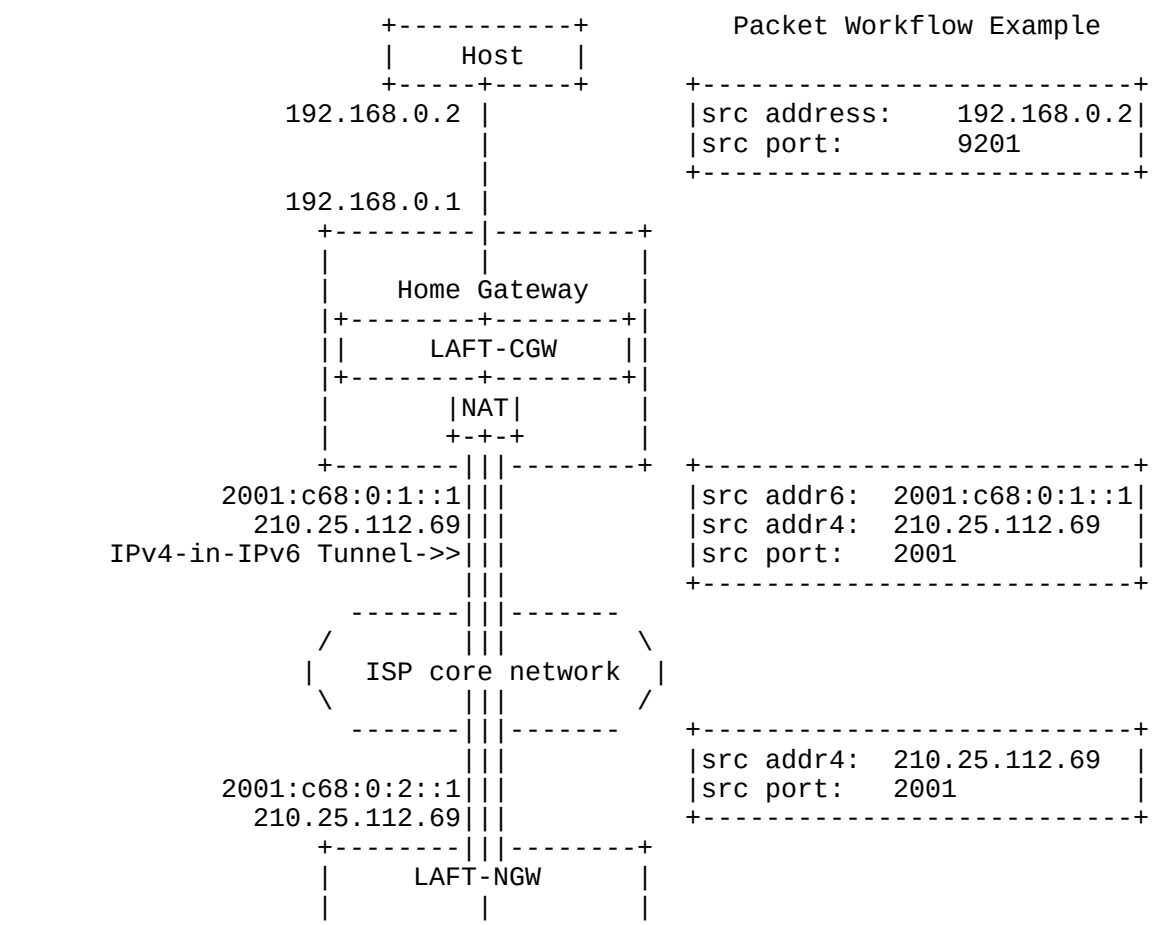
In LAFT6, LAFT-CGW can deal with session-based packet translation like traditional NAT, except that it would change the randomly generated source port to a valid port number within the range of Port-range-CGW. At the same time, it will deal with different scenarios accordingly. While in LAFT-NGW, on the other hand, it only needs to do packet encapsulation/ de-encapsulation according to the address mapping table in LAFT-NGW for different scenarios. Although it is a little complicated in LAFT-CGW, there will be not many new functionalities to implement since customer-side NAT is a quite common function for most current CPEs.

3.4. LAFT6 workflow

For upstream packets originated from IPv4 sources and destined for a receiver located in the IPv4 network, LAFT-CGW will firstly change the randomly generated source port to a valid port selected from Port-range-

CGW, then change the private IPv4 address to its Addr4-CGW-pub and encapsulate in IPv6 packet directed to LAFT-NGW. The corresponding mapping table with IPv4 addresses and port number will be maintained in LAFT-CGW. The LAFT-NGW will only need to de-encapsulate the packet and forward them as IPv4 packets through the IPv4 network to the IPv4 receiver. There is no packet translation in LAFT-NGW anymore. For downstream IPv4 packet, LAFT-NGW will extract the IPv4 destination address and destination port from the incoming packet, lookup the mapping table, determine the corresponding IPv6 address and then tunneled to LAFT-CGW. LAFT-CGW will also de-encapsulate the packet, lookup the mapping table for each packet, determine the original port number and private IPv4 address, and translate the packet back again.

The workflow of 4to4 communication is depicted in the following Figure 2.



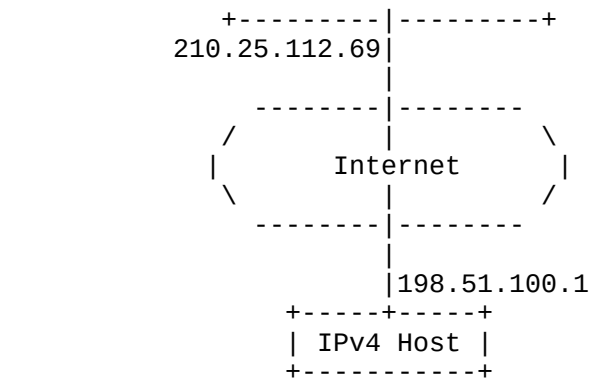
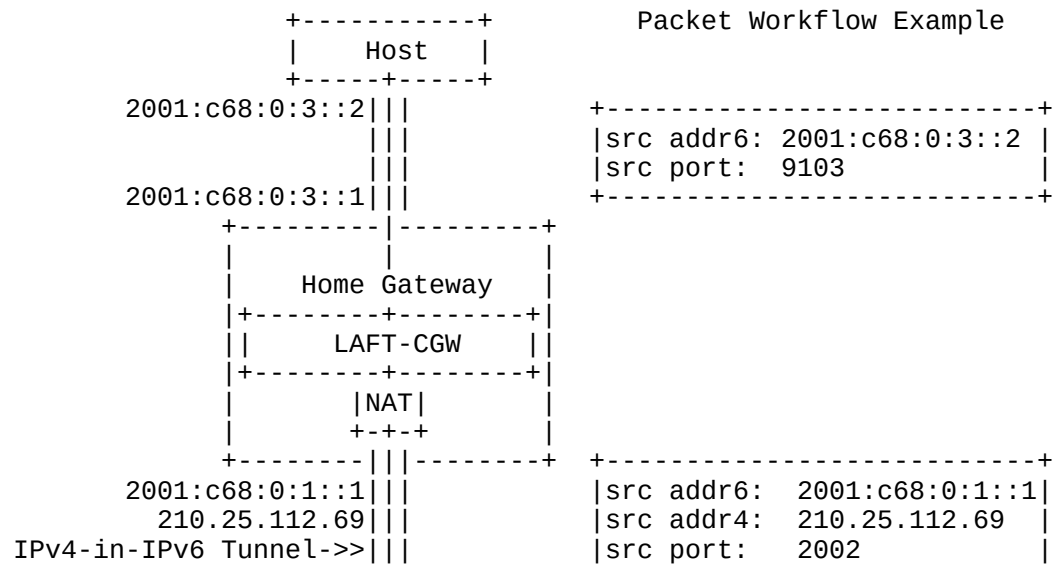


Figure 2 Workflow of 4to4 communication

For packets generated from IPv6 sources for a receiver located in the IPv4 network, the LAFT-CGW will not only need to change the random source port to a valid port in the Port-range-CGW,and change the IPv6 address to Addr4-CGW-pub, but also translate the IPv6 packet to an IPv4 packet. Then, the traversed IPv4 packet with Addr4-CGW-pub will also be encapsulated in IPv6 packet and then tunneled to LAFT-NGW. In this IPv6 header, the source address is Addr6-CGW, rather than Addr6-user in the original IPv6 packet generated by IPv6 source. LAFT-NGW will take the same procedure as in the 4to4 scenario.

The workflow of 6to4 communication is depicted in the following Figure 3.



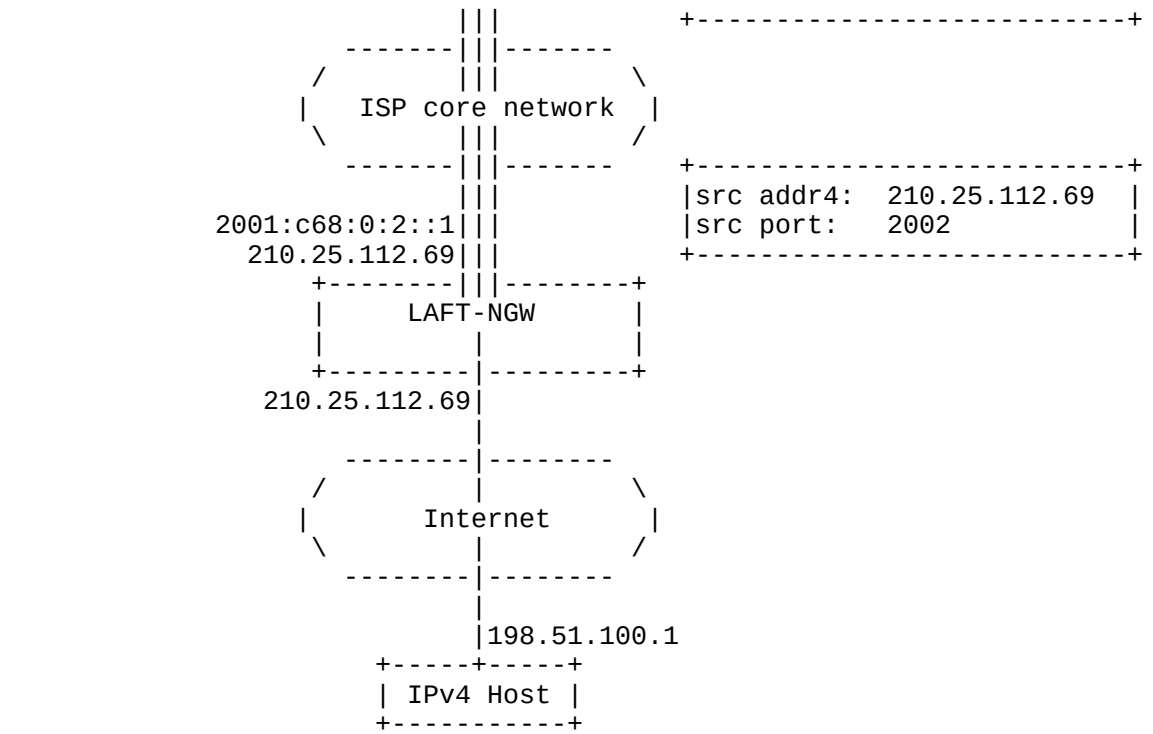


Figure 3 Workflow of 6to4 communication

4. LAFT-CGW element

The LAFT-CGW element is a function implemented on CPE. LAFT-CGW need to perform initialization, build extended binding table, do packet translation, encapsulation, fragmentation and reassembly, and DNS proxy, etc.

4.1. Initialization

In the initialization phase, each LAFT-CGW device will get its own IPv6 address by existing user authentication process, and it will also get Addr4-CGW-pub and Port-range-CGW by extended protocols []. Furthermore, it would allocate private IPv4 address to IPv4 or dual-stack end-users.

<We will add additional port range considerations in the next version>.

4.2. Extended Binding Table

There are two kinds of binding tables in LAFT-CGW element: one is 4to4 scenario (denoted as 4to4 table) and the other is 6to4 scenario (denoted as 6to4 table).

There are conceptual dynamic data structures to construct the 4to4 and 6to4 binding table, with TCP, UDP and ICMP respectively. In case of TCP and UCP, each state entry specifies a mapping between the IP address and port number:

(X',x) <--> (T,y)

where X' is Addr4-user in 4to4 mapping table and is Addr6-user in 6to4 table, T is the Addr4-CGW-address for both 4to4 table and 6to4 table, x is the original random port created by upper-layer application for LAFT-CGW and y is the translated port in Port-range-CGW. A given IPv6 or IPv4 transport address can appear in at most one entry in a binding table since TCP and UDP have separate binding tables because TCP and UDP have different port number space.

In the case of the ICMP Query, each ICMP Query binding entry specifies a mapping between an (IP address, ICMP Identifier) pair.

(X',i1) <--> (T,i2)

where X' and T are the same as in the above table, i1 and i2 are an ICMP Identifiers in 4to4 case and are an ICMPv6 Identifiers in 6to4 case. A given (IPv6 or IPv4 address, ICMPv6 or ICMPv4 Identifier) pair can appear in at most one entry in the ICMP Query.

Each upstream packet will construct a binding entry in case there is no existing binding entry in the extended binding table. It will determine the traversed port number for each packet. For downstream packet, LAFT-CGW knows how to re-construct IPv4/IPv6 packet when the packets comes back from by doing a reverse look-up in the extended IPv4 NAT binding table.

4.3. Packet Translation

In LAFT-CGW, packet translation includes network-layer header translation and transport-layer header translation.

- Network-layer header translation

For both IPv4 and IPv6 packet, it will change the end user address to Addr4-CGW. For IPv6 packet, it MUST be performed according to SIIT [RFC2765] except the source addresses in the header.

o Transport-layer header translation

In LAFT-CGW, source port should be changed to the conversed port according to extended binding table. Since the TCP and UDP headers [RFC0793] [RFC0768] consist of check sums which include the IP header, the recalculation and updating of the transport-layer headers MUST be performed.

4.4. Encapsulation/De-encapsulation

The tunnel is a multi-point to point IPv4-in-IPv6 tunnel ending on a service provider LAFT-NGW. The upstream IPv4 packet will be encapsulated in IPv6 header and tunneled to LAFT-NGW. In the same way, the downstream IPv6 tunneled packet will be de-encapsulated in LAFT-CGW.

4.5. Fragmentation and Reassembly

Fragmentation and Reassembly is the most time-consuming task for LAFT-CGW. Thus, it is better to deal with this problem, either by increasing the MTU size of all the links between the LAFT-CGW element and the LAFT-NGW elements or by limiting the size of packet generated by end users.

However, as not all service providers will be able to control the links or the packet size, the LAFT-CGW element MUST perform fragmentation and reassembly if the outgoing link MTU cannot accommodate the packet IPv6 transmission due to the addition of the extra IPv6 header.

Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the de-capsulation of the IPv6 header. The IETF standard for Fragmentation and MTU Handling is defined in [I-D.ietf-behave-v6v4-xlate], which contains updated technical specifications.

4.6. LAFT-NGW discovery

In order to configure the IPv4-in-IPv6 tunnel, the LAFT-CGW element needs the IPv6 address of the LAFT-NGW element. In order to guarantee interoperability, a LAFT-CGW element SHOULD implement the DHCPv6 option defined in [I-D.ietf-softwire-ds-lite-tunnel-option].

4.7. DNS

A LAFT-CGW element is only configured from the service provider with IPv6 address, so it can only learn the address of a DNS recursive server through DHCPv6 (or other similar method over IPv6). As DHCPv6 only defines an option to get the IPv6 address of such a DNS recursive server, the LAFT-CGW element cannot easily discover the IPv4 address of such a recursive DNS server, and as such will have to perform all DNS resolution over IPv6. As a result, the LAFT-CGW element SHOULD implement a DNS proxy, following the recommendations of [RFC5625].

When LAFT6 is applied to 6to4 scenario, it should also perform DNS64 [I-D.ietf-behave-dns64] in case there is no DNS64 server located in the local network. It should be configured with a Pref64 to synthesize IPv6 address. For AAAA request, the DNS64 will query the AAAA response. If there is no response for a certain period, it will reconstruct an A request and synthesize an AAAA response.

5. LAFT-NGW element

An LAFT-NGW element is the combination of an IPv4-in-IPv6 tunnel end-point. LAFT-NGW element is a light-weight end-point which only needs to do port-range management, encapsulation, fragmentation and reassembly, etc.

5.1. Port-range Binding Table

In LAFT-NGW, there is one Port-range Binding tale for TCP, UDP and ICMP. Each state entry specifies a mapping between the IP address, port range:

(X’) <--> (T,pr)

where X’ is the IPv6 address of LAFT-CGW (Addr6-CGW), T is the user’s Addr4-CGW, and pr is Port-range-CGW for each user. pr can be continuous, discrete or partial random. This binding table can be used for both 4to4 and 6to4 scenarios.

5.2. Encapsulation

The tunnel is a point-to-multipoint IPv4-in-IPv6 tunnel ending at the LAFT-CGW elements.

5.3. Fragmentation and Reassembly

Fragmentation and Reassembly will be performed if the underlying link MTU cannot accommodate packet transmission due to addition of the

extra IPv6 header of the tunnel. Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the de-capsulation of the IPv6 header.

Methods to avoid fragmentation, such as rewriting the TCP MSS option or using technologies such as Subnetwork Encapsulation and Adaptation Layer defined in [I-D.templin-seal] are out of scope for this document.

5.4. DNS

As noted previously, LAFT6 node implementing a LAFT-CGW element will perform DNS resolution over IPv6. As such, very few, if any, DNS packets will flow through the LAFT-NGW element.

6. Deployment Considerations for Broadband Provider

6.1. Addressing and Routing

In LAFT6, there is no additional addressing and routing requirements. Thus, the process of IPv6 address assignment and routing entry establishment can be integrated with existing IPv6 address allocation process, e.g. using PPPoE or IPoE, etc.

6.2. DNS

In LAFT6, since there is no IPv4 DNS server in IPv6-only network, it is recommended that LAFT-CGW should implement IPv4-to-IPv6 DNS Proxy to convert an IPv4 DNS request/response to IPv6 DNS request/response accordingly.

When applied to scenarios for IPv6 sources communicating with IPv6 receivers, DNS64 should be deployed. In case there is no DNS64 deployed in IPv6 operational network, it is recommended that DNS64 should be integrated into LAFT-CGW.

6.3. AAA and User Management

User authentication can be used to control who can have the LAFT6 connectivity service. In the initial phase of deployment, the maximum number of port number for subscribers can be configured uniformly in LAFT-NGW. But it is still recommended that AAA would define the maximum number of port number for different subscribers to offer better security and differentiated service.

6.4. Placement in Large SP Network

Normally, LAFT-NGW can be deployed in "centralized model" and "distributed model".

In "centralized model", LAFT-NGW could be deployed in the place of Core Router. Since LAFT-NGW has good scalability and it can handle numerous concurrent sessions, we strongly recommend adopting "centralized model" for LAFT6 as it is a cost-effective way and easy to manage.

In "distributed model", LAFT-NGW is usually be integrated with BRAS/SR. Since the newly emerging customers might be distributed in the whole Metro area, we have to deploy LAFT-NGW on all BRAS/SRs. This will cost a lot in the initial phase of IPv6 transition period.

6.5. ALG consideration

Currently, LAFT6 can support most of existing applications, such as HTTP, SSH and Telnet. However, some applications are designed such that IP addresses are used to identify application-layer entities (e.g. FTP). In these cases, application layer gateway (ALG) is unavoidable, and it can be integrated into the LAFT-CGW.

Since there is no address and port mapping in LAFT-NGW, there is no ALG needed anymore in carrier side network.

7. Security Considerations

There are no security considerations in this document.

8. IANA Considerations

This memo adds no new IANA considerations.

9. Appendix A: Experimental Result

We have tested LAFT6 using the prototype in our operational network of HuNan province, China. The major objective listed in the following is to verify the functionality and performance of LAFT6:

O Verify how to deploy LAFT6 in practical network.

O Verify what applications can be used in LAFT6.

O Verify the effect of user experience with limited ports.

O Verify the performance of LAFT6.

9.1. Experiment environment

The network topology for this experiment is depicted in Figure 4.

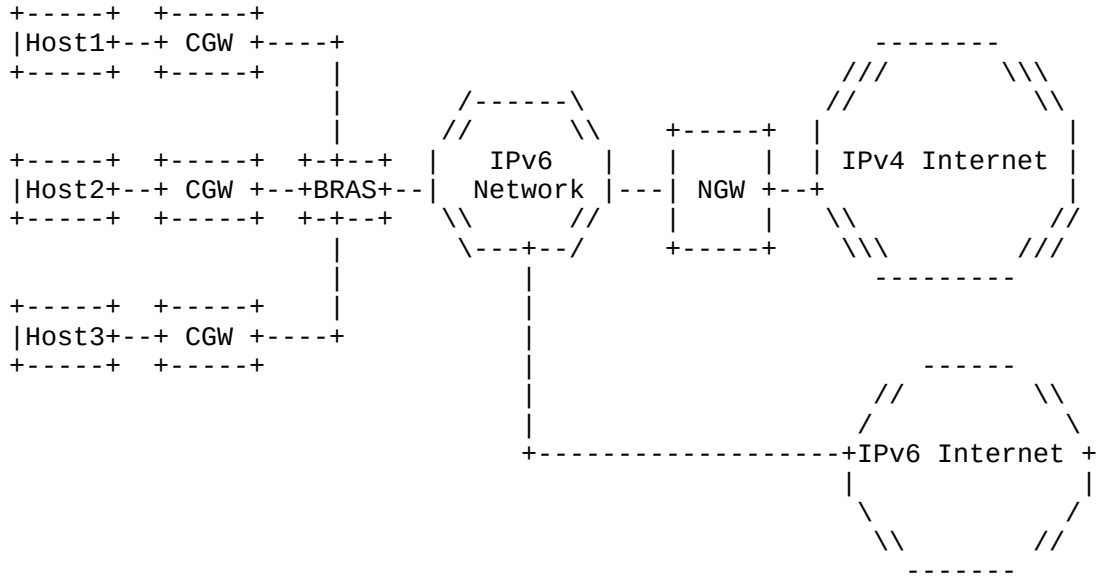


Figure 4 LAFT6 topology in the test

There are three key components in the test:

o The Hosts are dual-stack or IPv6-only customers, who could run IPv4 application, IPv6 application or dual stack application.

o The Home Gateways (Hgw) are LAFT-CGW in user side. It would do packet translation, encapsulation, fragmentation and reassembly, and DNS proxy, etc.

o The LAFT-NGW encapsulate/de-encapsulate the packet according to the mapping table in LAFT-NGW.

9.2. Experiment configuration

For address configuration, each host will get it IPv6 address through PPPoE process. And there is no explicit routing configuration needed.

For port configuration, we allocate each user with 2000 maximum available ports in LAFT-NGW. We have not implement AAA system with additional port-number notification.

For DNS configuration, since LAFT-CGW has implemented DNS64 itself, there is no DNS64 needed anymore in our operational network.

9.3. Experiment results

In our trial, we mainly have taken the following tests:

- Application test: The applications we have tested include web, email, Instant Message, ftp, telnet, SSH, video, Video Camera, P2P, online game, Voip, VPN and so on.

- Operating System test: The OS we have tested includes Win7, VISTA, windows XP.

- Performance test: We have measured the parameters of concurrent session number, throughput performance.

The experimental results are listed as follows:

Type	Experiment Result
Application test	LAFT hosts can support web, email, im, ftp , telnet, SSH, video, Video Camera, P2P, online game, voip, and so on.
Operating System test	LAFT can widely support Win7, VISTA, windows XP.
Performance test	The performance test for LAFT-NGW is carried out on a normal PC. Due to limitation of the PC hardware, the overall throughput is not quite good. However, it can still support more than one hundred million concurrent sessions

Figure 5 LAFT6 test result

9.4. Conclusions

From the experiment, we can have the following conclusions:

- o LAFT6 has good scalability. LAFT-NGW is a lightweight solution which only maintains per-subscriber state information. As a result, it can easily support a large amount of concurrent subscribers.

- o LAFT6 can be deployed rapidly. There is no modification to existing addressing and routing system in our operational network. And it is simple to achieve traffic tracing and logging.

- o LAFT6 can support a majority of current IPv4 applications and it can support a variety of Operating Systems.

10. References

[I-D.ietf-softwire-dual-stack-lite] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010

[I-D.ietf-behave-v6v4-xlate-stateful] Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.ietf-intarea-shared-addressing-issues] M. Ford, Ed., M. Boucadair, A. Durand, P. Levis, P. Roberts, "Issues with IP Address Sharing", draft-ietf-intarea-shared-addressing-issues-04 (work in progress), February 2011.

[I-D.behave-natx4-log-reduction] T. Tsou, W. Li, T. Taylor, "Port Management To Reduce Logging In Large-Scale NATs", draft-tsou-behave-natx4-log-reduction-02(work in progress), September 2010.

[I-D.ietf-softwire-ds-lite-tunnel-option] D. Hankins, T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-08 (work in progress), January 2011

[RFC5625] R. Bellis, "DNS Proxy Implementation Guidelines", RFC5625, August 2009.

[I-D.ietf-behave-dns64] Bagnulo, M., "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-dns64-10.txt, July 2010.

[I-D.tsou-pcp-natcoord] T.Tsou, C.Zhou, Q.Sun, T.Taylor, "Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation", draft-tsou-pcp-natcoord-00.txt, March 4, 2011.

11. Acknowledgments

The authors would like to thank to Fred Baker and Tony Hain for his continuous suggestion around this topic over the years. Thanks to Qian Wang, Jie Hu and Fan Shi for useful feedback.

Authors' Addresses

Qiong SUN
China Telecom Beijing Research Institute
Room 708 No.118, Xizhimenneidajie, xicheng District Beijing 100035
China

Phone: <86 10 58552936>
Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom Beijing Research Institute
Room 708 No.118, Xizhimenneidajie, xicheng District Beijing 100035
China

Phone: <86 10 58552116>
Email: xiechf@ctbri.com.cn

Network Working Group
Internet Draft
Intended status: Informational
Expires: April 24, 2011

Qiong Sun
Chongfeng Xie
China Telecom
Xing Li
CongXiao Bao
CERNET Center/Tsinghua University
Ming Feng
China Telecom

March 6, 2011

Considerations for Stateless Translation (IVI/dIVI) in Large SP Network
draft-sunq-v6ops-ivi-sp-02.txt

Abstract

With the approaching exhaustion of IPv4 address space, large-scale SPs are now faced with the only real option to deploy IPv6 in a timely manner. In order to achieve smooth transition to IPv6, migration tools should be introduced for different deployment models. Among different IPv6 transition mechanisms, dIVI is a prefix-specific and stateless address mapping method which can directly translate IPv4 packet to IPv6 packet. This document describes the challenges and requirements for large SP to deploy IPv6 in operational network, the experimental results of dIVI in our laboratory and the considerations for dIVI deployment in large SP operational network.

This Internet-Draft will expire on September 6,2011.

Copyright Notice

Table of Contents

1. Introduction

The dramatic growth of the Internet is accelerating the exhaustion of available IPv4 addressing pool. It is widely accepted that IPv6 is the only real option on the table for the continued growth of the Internet. However, IPv6 deployment is a huge systematic project and a lot of challenges will arise especially in large SP operational network.

In order to achieve smooth transition to IPv6, migration tools should be introduced for different deployment models, among which dIVI is a stateless translation mechanism with good scalability. This document describes the challenges and requirements for large SPs in IPv6 transition period. Then, we introduce dIVI experimental results in our laboratory. And finally, the considerations for designing and deploying dIVI in operational network are discussed in terms of addressing and routing, DNS deployment requirement, AAA support and user management, network management, CPE requirement and Xlate placement.

2. Terminologies

This document uses the terminologies defined in [I-D.ietf-behave-v6v4-framework], [I-D.ietf-behave-v6v4-xlate], [I-D.ietf-behave-address-format], [I-D.ietf-behave-v6v4-xlate-stateful], and [I-D.xli-behave-ivi].

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

3. Problem Statement

It is well known that the pool of public IPv4 addressing is nearing its exhaustion. The '/8' IANA blocks for Regional Internet Registries (RIRs) are projected to 'run-out' towards the end of 2011. Credible estimates based on past behavior suggest that the RIRs will exhaust their remaining address space by early 2012, apart from the development of a market in IPv4 address space. Thus, it will become much more difficult to get available public IPv4 addresses. In the same time, a lot of emerging applications, e.g. Apple's iPad, Motion's BlackBerry, etc. will quickly deplete the available addresses. This has led to a hightened awareness among the providers to consider introducing IPv6 to keep the Internet operational.

It has been widely accepted that the end goal of IPv6 transition is to achieve an end-to-end IPv6-only network, and IPv4 can eventually

be turned off. However, since it will have impact on almost the entire world, it will take a considerable period of time to reach the ultimate goal. As a result, IPv4 and IPv6 need to coexist during the whole transition period. In this document, we mainly focus on IPv6 migration issues from large ISP's point of view. In order to facilitate smooth IPv6 migration, some factors need to be taken into consideration especially for large SPs. There are ten major requirements:

1. It should deploy in an incremental fashion and the overall transition process should be stable and operational.

2. It should largely reduce IPv4 public address consumption.

3. It should accelerate the deployment of IPv6, rather than just prolonging the lifecycle of IPv4 by introducing multiple layers of NAT.

4. There should be no perceived degradation of customer experience. As a result, IPv6 transition mechanisms should provide IPv4 service continuity.

5. It should achieve scalability, simplicity and high availability, especially for large-scale SPs.

6. It should have user management and network management ability.

7. It should support user authentication, authorization and accounting as an essential part of operational network.

8. Most ISPs need some kind of mechanisms to trace the origin of traffic in their networks. This should also be available for IPv6 traffic.

9. It should have good throughput performance and massive concurrent session support.

10. It should maintain the deployment concepts and business models which have been proven and used with existing revenue generating IPv4 services.

All existing IPv6 transition mechanisms can be widely divided into three categories: dual-stack solution, translation-based solution and tunnel-based solution. In this document, we mainly concentrate on stateless translation mechanism: dIVI. The original stateless IPv4/IPv6 translation (stateless 1:1 IVI) is scalable, [I-D.ietf-behave-v6v4-framework], [I-D.ietf-behave-v6v4-xlate], [I-D.ietf-

behave-address-format],[I-D.xli-behave-ivi]. But it cannot use the IPv4 addresses effectively. The stateless dIVI[I-D.xli-behave-divi] is a double translation mechanism which includes a 1:N stateless translator and a 1:1 Hgw translator. The 1:N stateless translator is implemented in the border between the IPv6 network and the IPv4 Internet. It translates the packets between IPv4 and IPv6 with the 1:N stateless address mapping. The 1:1 Hgw translator is implemented between an IPv6 network and user's end system. It translates the packets between IPv4 and IPv6 with 1:1 stateless address mapping. In addition, the home gateway translator maps random source port numbers to restricted port number based on the extended IPv4-translatable address format and keeps the mapping table in database for the port number mapping of the retuning packets and all the packets in the same session.

dIVI support bidirectional communication initiated from IPv4 and IPv6. It can be deployed in an IPv6-only access network, in which operational and maintenance cost can be reduced. It has very good scalability and can largely reduce IPv4 address consumption.

In this document, we firstly demonstrate the laboratory experimental results of dIVI in section 4. We can see that dIVI can support most of the current IPv4 applications in IPv6-only access network, while largely reducing IPv4 address consumption. And then dIVI deployment model and considerations in large operational network are proposed in section 5 and section 6 respectively. Some important factors need to be taken into account when introducing dIVI. Since most challenges in dIVI have no big difference compared to an IPv6-only environment, we strongly recommend that related network elements should take the corresponding modifications in order to guarantee the IPv6 transition process to be operational and manageable.

4. Laboratory experiment

We have tested dIVI using the prototype in our laboratory. The major objective listed in the following is to verify the functionality and performance of dIVI:

O Verify how to deploy dIVI in practical network.

O Verify what applications can be used in dIVI.

O Verify what Operating Systems can be supported in dIVI.

O Verify the effect of user experience with limited ports.

O Verify the performance of dIVI.

4.1. Experiment environment

We have tested dIVI in our laboratory. The network topology is depicted in Figure 1.

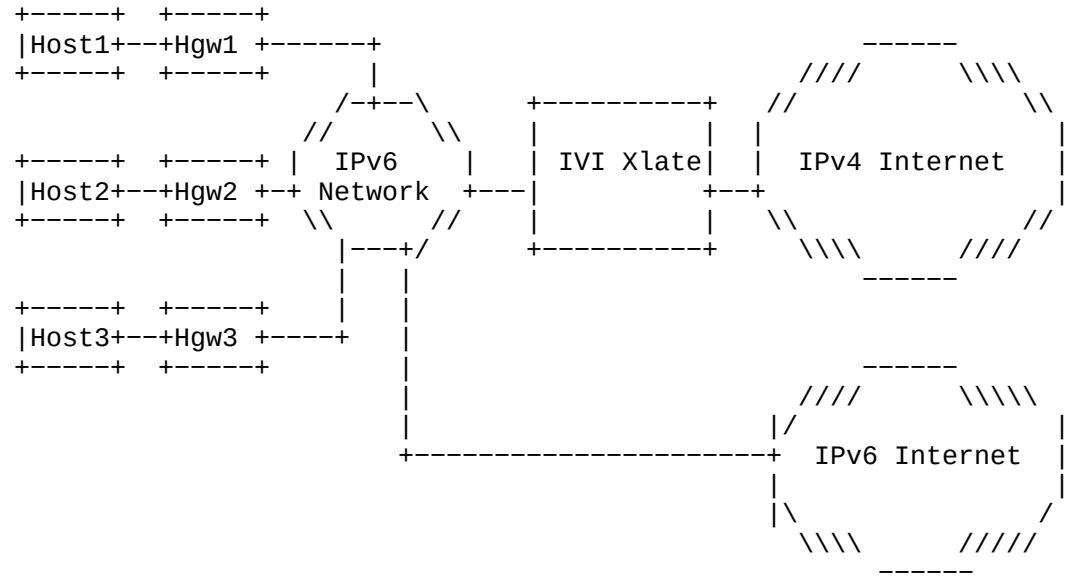


Figure 1 dIVI topology in the test

There are three key components in the test:

- The Hosts are dual-stack customers, who could run IPv4 application, IPv6 application or dual stack application.

- The Home Gateways (Hgw) are dIVI translator in user side. It translates the packets between IPv4 and IPv6 with 1:1 stateless address mapping, and maps random source port numbers to restricted port number.

- The Xlate translates the packets between IPv4 and IPv6 with the 1:N stateless address mapping.

The network between Hgw and Xlate is IPv6-only, and the network behind Hgw is dual-stack. Thus, the hosts behind Hgw can communicate with both IPv4 Internet and IPv6 Internet.

4.2. Experiment configuration

For address configuration, each host will use two IPv6 addresses: one is IVI6 address which is synthesized in Hgw with the IVI4 address and port-related information, and the other is non-IVI IPv6 address which is used for native IPv6 communication. We should notice that only non-IVI IPv6 address needs be allocated to end users. Besides, each user will get an IVI4 address from Hgw.

For routing configuration, both IVI address and non-IVI address need to be imported into the IPv6-only network.

For port configuration, since there are 65536 TCP/UDP ports for each IP address, and in fact one can use hundreds only for normal applications, so one IPv4 address can be shared by multiple customers. In our experiment, we selected ratio to be 128. That is to say, one IPv4 address is shared by 128 users, and there are 512 available ports per user.

For DNS configuration, there is no need to have additional DNS64 for dIVI. Only an IPv6 DNS server with A/AAAA records is needed and the DNS address is manually configured in Hgw. Besides, Hgw has implemented DNS Proxy and it will convert an IPv4 DNS request/response to IPv6 DNS request/response.

For ALG configuration, there is no need to deploy specific ALG for IPv4 applications in dIVI.

4.3. Experiment results

In our laboratory, we mainly have taken the following tests:

- Application test: The applications we have tested include web, email, Instant Message, ftp, telnet, SSH, video, Video Camera, P2P, online game, Voip, VPN and so on.

- Operating System test: The OS we have tested includes Win7, VISTA, windows XP.

- Performance test: We have measured the parameters of concurrent session number, throughput performance.

The experimental results are listed as follows:

Type	Experiment Result
Application test	dIVI hosts can support web, email, im, ftp , telnet, SSH, video, Video Camera, P2P, online game, voip, and so on.
Operating System test	dIVI can widely support Win7, VISTA, windows XP.
Performance test	The performance test for dIVI Xlate is carried out on a normal PC. Due to limitation of the PC hardware, the overall throughput is not quite good. However, it can still support more than one hundred million concurrent sessions.

Figure 2 dIVI test result

From the experiment, we can have the following conclusions:

1. dIVI can have good scalability. Xlate does not need to maintain any session state, and only limited session states have to been maintained in Hgw for its customer.

2. dIVI can be deployed in an incremental way. The most complicated part of dIVI is addressing and routing. The configuration for DNS and ALG is very simple.

3. dIVI can support a majority of current IPv4 applications.

4. dIVI can support a variety of Operating Systems.

5. With the ratio of 128 (512 maximum concurrent sessions), there is no perceived degradation of customer experience.

However, in the current status of equipment, e.g. BRAS, end system, etc., the support for IPv6-only function has not been fully accomplished. Therefore, there are still some limitations which would be improved in the next version of dIVI development when deploying dIVI prototype in practical operational network:

1. Address assignment process have not been integrated with existing address allocation system.

2. Currently, IVI routing entries are configured manually.

3. Hgw has not integrated PPPoE functionality with dIVI functionality.

4. AAA system has not supported IVI-related (or IPv6-only) functions.

With regard to the listed IPv6 transition requirements in section 3, most of them can be satisfied by dIVI, except for the requirement of network management and user management. These two points should be paid special attention for large SPs, which will be further discussed in section 6.

5. dIVI Deployment Scenario

In order to investigate the ways to deploy dIVI in operational network, we firstly briefly discuss network architecture for large SP network. Then dIVI deployment model is introduced.

5.1. Network Architecture for Large SP Network

In large SPs, the generic network topology can be divided into four main parts (as depicted in Figure3): the Customer Premises Network (CPN), the Access Network (AN), the Metro Area Network (MAN), and the Backbone Network.

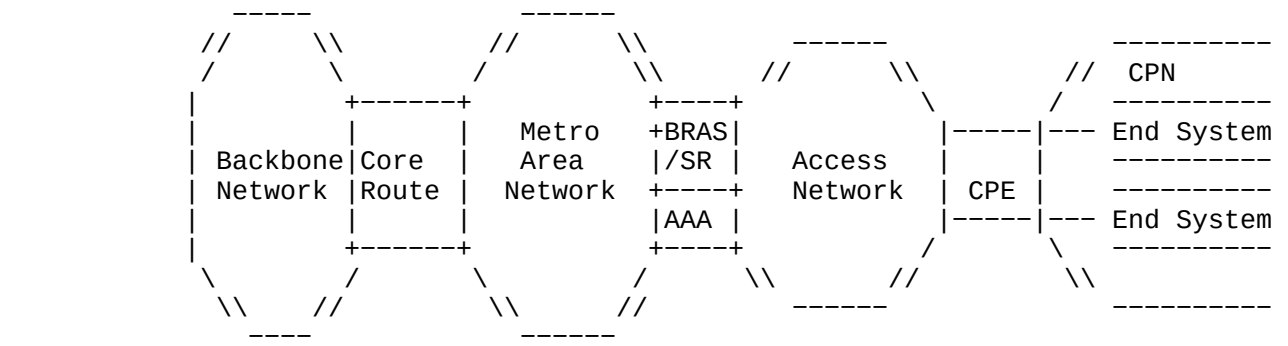


Figure 3 Network Architecture for Large SP Network

1. CPN is the part of the network by a customer when connecting to an ISP's network which includes the CPE and the last hop link.

2. In Access Network, a very wide variety of access technologies can be used, including ADSL, Ethernet, PON, ATM, WIFI, etc.

3. MAN is the aggregated network for customers in one single metro, with the vast range of size. In most metro networks, BRAS is connected to Core Router directly, while for a small portion of large metro networks, BRAS is connected to Core Routers via aggregated routers.

4. Backbone network is to offer transit service between MANs and other ISPs.

There are typically two network models for fixed broadband access service: one is to access using PPPoE/PPPoA authentication method while the other is to use IPoE. The first one is usually applied to Residential network and SOHO networks. Subscribers in CPNs can access broadband network by PPP dial-up authentication. BRAS is the key network element which takes full responsibility of IP address assignment, user authentication, traffic aggregation, PPP session termination, etc. Then IP traffic is forwarded to Core Routers through Metro Area Network, and finally transited to external Internet via Backbone network.

The second network scenario is usually applied to large enterprise networks. Subscribers in CPNs can access broadband network by IPoE authentication. IP address is normally assigned by DHCP server, or static configuration.

5.2. dIVI Deployment Scenario in Operational Network

The deployment model of dIVI in operational network is depicted in Figure4.

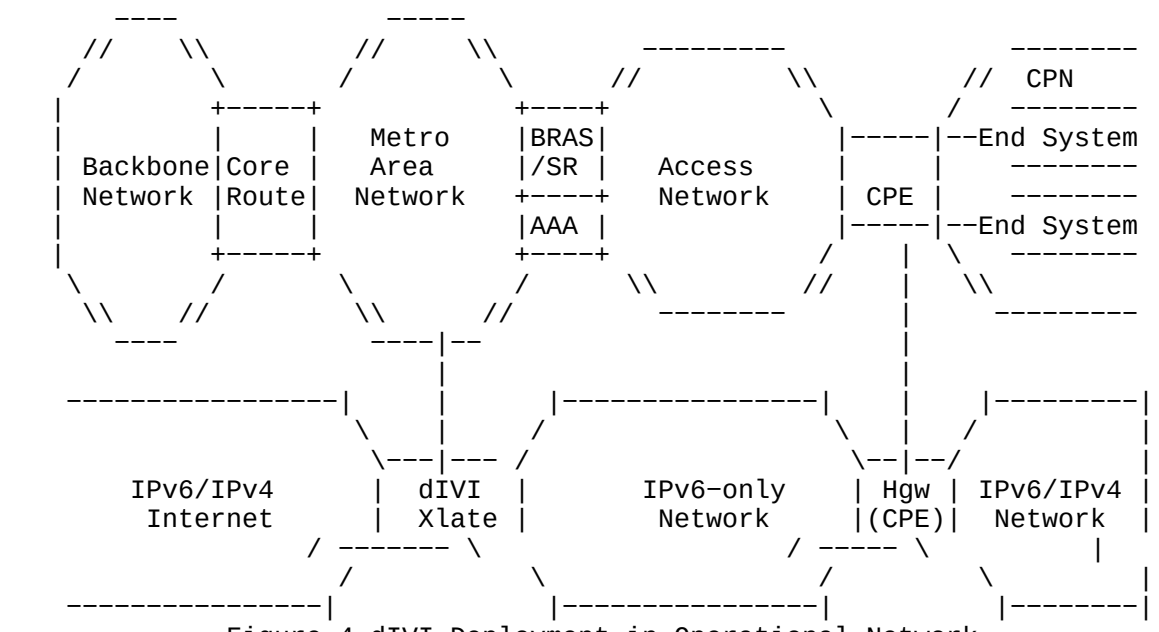


Figure 4 dIVI Deployment in Operational Network

In stateless dIVI, the network between Hgw and Xlate is an IPv6-only network, in which the operational and maintenance cost can be greatly reduced. The access network behind Hgw can either be IPv4-only or dual-stack. Thus, IPv4-only system and dual-stack system can communication with IPv4 Internet using shared IPv4 address by dIVI and the dual-stack system can also communicate with IPv6 Internet directly.

In operational network, Hgw can usually be integrated with CPE, while Xlate can be in someplace of MAN or Backbone Network. Subscribers can get IPv6 address from BRAS/SR after user authentication stage. Then, IVI-related route should be imported into the IPv6-only network between Xlate and Hgw. The detailed considerations for dIVI deployment will be discussed in section 6.

6. Considerations for dIVI deployment

This section describes the considerations for dIVI deployment in large operational network.

The major differences between dIVI deployment in laboratory and operational network lie in:

1. Operational network is a commercial network with strict user management requirement, while in laboratory it is simple and straightforward.

2. Operational network has different kinds of network equipments, e.g. BRAS/SR, CPE, Radius, etc. It would be more difficult to take modifications on all of these equipments.

3. Operational network has a large number of customers. Thus, it would be impossible to take manual configuration for all customers.

In this section, we try to outline considerations for dIVI deployment on large SP network. Some of the features are not specific to dIVI, but rather to a general requirement on all IPv6 transition techniques.

6.1. Addressing

In dIVI, there is no need to allocate IVI6 address explicitly to end users. Thus, the process of IPv6 address assignment can be integrated with existing IPv6 address allocation process. Only CPE will need to get IVI4 address, reallocate it to end user, do port-mapping and traffic translation with port-related information. Here are some basic considerations in dIVI addressing:

- Determine IVI6 prefix for each Xlate. Operators should use its own prefix as an IVI6 prefix, i.e. pref=2001:db8:a4a6::/48, in order to perform stateless translation. Address allocation process in BRAS/SR should be consistent with Xlate.

- Determine the embedded IVI4 address and port multx ratio. Operators should estimate the scale of subscribers in a certain region, evaluate the number of remaining IPv4 address, and analyze the number of concurrent ports. It is a tradeoff between multx ratio and concurrent port numbers. The bigger the multx ratio is, the more an IPv4 address can be shared by multiple subscribers and the less concurrent port number can be used per subscriber. From the above test in our laboratory, we choose multx ratio to be 128 and it is enough for current usage.

- Determine the ways to distribute the configuration profile including IVI4 address and port multx ratio to Hgw automatically, either by extended DHCP option, or other protocols.

6.2. Routing

In dIVI, IVI4(i) and IVI6(i) will be aggregated to ISP's IPv4 address and ISP's IPv6 address. They will not affect the global IPv4 and IPv6 routing tables

In dIVI deployment model, Hgws are normally configured with a default route that points to the BRAS/SR. The routers between BRAS/SR and Xlate run IPv6 dynamic routing protocols (IGP or BGP), and routers in the upper level of Xlate run IPv4 dynamic routing protocols. Therefore, the aggregated IVI6 routing directing to the upper routers will be learned/inserted by in IPv6-only domain. And the IVI6 route directing to Hgws should also be configured in BRAS/SR.

6.3. DNS

In dIVI, there is no DNS64/DNS46 needed anymore. An IPv6 DNS server is needed to process IPv6 DNS request/response, and the address of IPv6 DNS server should be passed to Hgw.

Since there is no IPv4 DNS server in IPv6-only network, it is recommended that Hgw should implement IPv4-to-IPv6 DNS Proxy to convert an IPv4 DNS request/response to IPv6 DNS request/response accordingly.

6.4. AAA and User Management

User authentication can be used to control who can have the dIVI connectivity service. This is not always required when a customer of IPv4 service automatically can have access to the dIVI service. However, it is highly recommended that IPv6-only customers should be authenticated separately. It is good for security, trouble shooting, user accounting, etc. There are some major points that AAA systems need to be taken into consideration:

- User authentication function needs to be extended to support the identification of IPv6-only subscriber, with additional dIVI-related profile for subscribers, e.g. IVI6 address, IVI4 address, non-IVI address, etc.

- User accounting and management function needs to be extended to identify dIVI user (or IPv6-only user) separately.

In summary, the major challenge of dIVI for the AAA and User Management is no big difference compared to an IPv6-only environment.

6.5. Network management

There are two issues to manage dIVI in operational network:

- Manage IPv6-only network. Operators should be able to manage IPv6-only network, including IPv6 MIB modules, Netflow Records, log information, etc.

- Manage the translation process. There are some exceptions that the MIB modules need to add dIVI related features, e.g. dIVI device management, dIVI traffic monitoring, etc.

6.6. dIVI CPE Requirements and Configuration

In dIVI, CPE is an important network element. It should perform DHCP server, integrated user authentication function, traffic translation and port mapping, DNS proxy, etc. The major operations in dIVI CPE include:

- Address assignment: dIVI CPE should support IVI4 address assignment by DHCP process to end users. It should also support IPv6 address assignment, either by stateful DHCP or stateless auto-configuration.

- Integrated user authentication function: dIVI CPE should integrate with existing user authentication function, e.g. PPPoE/PPPoA, etc.

- DNS: CPE should enable RFC 5006, well-known addresses, and DHCPv6 in order to maximize the likelihood of dIVI Hgw being able to use DNS without manual configuration. Besides, dIVI CPE should also support IPv4-to-IPv6 DNS proxy.

6.7. dIVI Xlate Placement in Large SP Network

Normally, dIVI Xlate can be deployed in "centralized model" and "distributed model".

In "centralized model", dIVI Xlate could be deployed in the place of Core Router, or even in the entrance of ICP. Since dIVI is a stateless method with better scalability than stateful ones, it can handle numerous concurrent sessions.

In "distributed model", dIVI Xlate is usually be integrated with BRAS/SR. So each Xlate should be configured with its own IVI6 prefix

and is responsible for translating the traffic of its own region. The number of subscribers is normally limited, so does the number of IVI routing entries. However, the network infrastructure should still be upgraded to dual-stack support in MAN and backbone network, and so the decreased operational cost is rather limited. Besides, since the newly emerging customers might be distributed in the whole Metro area, we have to deploy Xlate on all BRAS/SRs. This will cost a lot in the initial phase of IPv6 transition period.

In summary, we strongly recommend adopting "centralized model" for dIVI. It is a cost-effective way and easy to manage.

6.8. ALG consideration

dIVI does not require ALG, this is a very important feature in the initial development phase of IPv6.

7. Security Considerations

There are no security considerations in this document.

8. IANA Considerations

This memo adds no new IANA considerations.

Note to RFC Editor: This section will have served its purpose if it correctly tells IANA that no new assignments or registries are required, or if those assignments or registries are created during the RFC publication process. From the author's perspective, it may therefore be removed upon publication as an RFC at the RFC Editor's discretion.

9. References

9.1. Normative References

[I-D.ietf-behave-address-format] C., Bao, Huitema, C., Bagnulo, M., Boucadair, M., and X.Li, "IPv6 Addressing of IPv4/IPv6 Translators", draft-ietf-behave-address-format-10 (work in progress), August 2009.

[I-D.ietf-behave-dns64] Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-dns64-11 (work in progress),October 2009.

[I-D.ietf-behave-v6v4-framework] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", draft-ietf-behave-v6v4-framework-10 (work in progress), October 2009.

[I-D.ietf-behave-v6v4-xlate] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", draft-ietf-behave-v6v4-xlate-23 (work in progress), October 2009.

[I-D.ietf-behave-v6v4-xlate-stateful] Bagnulo, M., Matthews, P., I. Beijnum, "IP/ICMP Translation Algorithm", draft-ietf-behave-v6v4-xlate-12 (work in progress), October 2009.

[I-D.xli-behave-divi] Li,X., Bao, C., and Zhang, H., "Address-sharing stateless double IVI", draft-xli-behave-divi-01, April 29, 2010.

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

10. Acknowledgments

The authors would like to thank to Fred Baker for his continuous suggestion around this topic over the years. Thanks to Qian Wang, Jie Hu and Fan Shi for useful feedback.

Authors' Addresses

Qiong SUN
China Telecom Beijing Research Institute
Room 708 No.118, Xizhimenneidajie, xicheng District Beijing 100035
China

Phone: <86 10 58552636>
Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom Beijing Research Institute
Room 708 No.118, Xizhimenneidajie, xicheng District Beijing 100035
China

Phone: <86 10 58552116>
Email: xiechf@ctbri.com.cn

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University

Phone: <86 10 62785983>
Email: xing@cernet.edu.cn

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University

Phone: <86 10 62785983>
Email: congxiao@cernet.edu.cn >

Ming Feng
China Telecom
No.31, Jinrong Ave,Xicheng District,100032

Phone: <86 10 58501428>
Email: fengm@chinatelecom.com.cn

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 8, 2011

J. Tan
J. Lin
W. Li
China Telecom
March 7, 2011

Experience from NAT64 applications
draft-tan-v6ops-nat64-experiences-00

Abstract

This document discusses our experiences from deploying NAT64 devices for various Internet applications. Before the final transition to an IPv6-only network, NAT64 is one of the possible technologies which may be used to give users access to the IPv4-only parts of the Internet via an IPv6-only network. This document analyzes the testing results for a number of popular applications and describes the problems to be solved in the period of transition from IPv4 to IPv6.

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 September 8, 2011.

Copyright Notice

Table of Contents

1. Introduction

This document discusses our experiences from deploying NAT64 [I-D.ietf-behave-v6v4-xlate-stateful] devices for various Internet applications, both traditional and new. The main conclusion is that it is possible to deploy NAT64 devices at the edge of IPv6-only networks, but there are a number of issues unsolved such as lack of IPv6 support in PPTP and VPN connections.

> Note: Since no NAT64 and DNS64 devices are available for the time being, NAT-PT was used instead. Tests were run both with DNS-ALG enabled and with DNS-ALG disabled.

2. Network Topology Setup

The operating system we tested was Microsoft Windows 7 and the tested applications are the currently updated version. The IPv6 prefix was delegated as 2001:c68:100:100::/64, while the global IPv6 address was configured via SLAAC. The NAT-PT device was connected to the edge router of CNGI (China Next Generation Internet). A static route was configured in this device to direct packets destined to prefix 2001:c68:100:2:: (the prefix for the IPv6 DNS server) to the NAT-PT device.

In the NAT-PT device, the IPv4 addresses of the target websites or servers were mapped to global IPv6 addresses through dynamic or static mappings. When the tested terminal sent packets to these global IPv6 addresses, they were routed to the NAT-PT device which performed protocol translation and address translation.

The address of IPv6 DNS server was manually configured to 2001:c68:100:2:1::ca61:6, with the DNS-ALG enabled at the NAT-PT device. The AAAA DNS query from the terminal was transformed to an A query to the ISP’s IPv4 DNS cache server via NAT-PT translation. We also implemented a dual-stack DNS cache server and added AAAA entries for the tested websites. The global IPv6 addresses of those websites are based on the NAT-PT’s prefix and its IPv4 public addresses. The terminal has been setup the DNS server address as the IPv6 address of this dual-stack DNS cache server while the DNS-ALG is disabled at NAT-PT.

The NAT Log information was acquired by remote monitoring. The logs contained the 5-tuple, set up time and end-up time. The traffic Log was manually exported.

3. Experiences With Various Use Cases

This section discusses specific issues with various applications and appliances. Issues to be solved are also described.

3.1. Web Applications

All HTTP/HTTPs based web browsers, i.e., comprehensive portal website, webmail,search engine and HTTP download, that we have tried so far seem to work well without problems. When the DNS-ALG option of NAT-PT is enabled and AAAA record request is not restrained, we can visit websites which have AAAA records in the public DNS, e.g., ipv6.google.com. The address is the "real" IPv6 address. However, if an IPv6 host initiates an AAAA record request for some website, e.g., www.abc.com, but there is no corresponding AAAA record in the DNS, the IPv4 version of the website cannot normally be visited since the DNS-ALG does not convert this AAAA record request to an A record request. On the other hand, while the DNS-ALG option is enabledand the AAAA record request is restrained, we can visit the website or servers without IPv6 address by NAT-PT and DNS-ALG, but we cannot get the IPv6 address of the IPv6/Dual-stack websites or servers when the AAAA DNS record requests go through the NAT-PT.

HTTP downloading via domain name works well normally, for example, to upload and download HTTP netdisk service. However, problems exist in thosee file sharing websites which have policies based on source IP addresses. In that case, downloading does not work correctly and the users who are sharing a public IPv4 address needs to wait for a very long time before starting download from the same website at the same time. In addition, for some website resources which are downloaded directly without DNS query, downloading does not work as well. Especially when the website redirects the users to a separate IPv4 server by telling the browser the IPv4 address rather than the domain name of the server.

We also tested with some new web applications, e.g., Web-based video, Online music, Blog, SNS, Online shopping and Web-based map. Most popular video sharing websites in China do not support NAT-PT because the video resource is normally stored separately and the web server or application server redirects the resources’ IPv4 address to the user-end Flash player plug-in, which is not translated to an IPv6 address when it goes through the NAT-PT device.

Actually, whether using domain name lookup or connecting by address directly to the source depends on the content of the website, security policy and website provided, security policies and its software and hardware architecture. It may be not easy to change the existing architecture which makes the transition of such kind of

websites difficult and complex.

Blogs on most websites appear to work fine except for one webpage style and layout problem at blog.163.com. We believe that is because the CSS file is not loaded correctly.

E-bank web plug-in and client (software) do not work well with NAT-PT devices because the client end usually communicates directly with the server using a known IPv4 address. Even though the client performs a domain name lookup procedure, most of the client cannot recognize the translated IPv6 address. Besides, there is usually a logging server for security purpose that may not recognize IPv6 addresses and may not identify and distinguish users by the shared IPv4 address.

Google maps display normally when the browser opens six windows/tabs of maps simultaneously to watch different sites. The sessions are limited separately to 250 and 50 in this testing.

3.2. Email Client

The impact of NAT64 on email protocols (POP3, SMTP and IMAP) worked normally. The Microsoft Live Mail 2011 and Microsoft Office Outlook support IPv6, while Foxmail 6.5 does not. But there may be a little problem. Users can only wait for a new version of the software and access their email account via webmail during the transition period.

3.3. Instant Messaging

We have tested several instance messaging applications in an IPv6-only network with NAT64 and the test results can be found in Table 1.

System	Status
QQ2010 client	NOT OK
WebQQ	OK
Windows Live Messenger	NOT OK
Ebuddy Web	OK
Fetion	NOT OK
Skype	NOT OK

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

Most of the instant messaging systems tested were not able to log onto the server, by reason of lacking IPv6 support in the clients. However, the web-based instant messaging sysytem works well, and may be considered as a transition tool for the instant messaging systems with a large number of clients before the new versions are released.

3.4. Peer-to-Peer (P2P) Applications

Each Peer-to-Peer (P2P) downloading software displays downloading resources on the information page, employing HTTP as transport. From the experiments we have done, most P2P software packages do not support IPv6, e.g., we failed to get connection with peers from BitComet. There are also P2P clients that claim to support IPv6, like uTorrent and emule. However, we did not succeed when trying to make IPv6 connections. The problem is probably that the peers' addresses of the contents stored in tracker server are mainly IPv4 addresses. When these addresses sent from the Tracker to the downloading peer is encapsulated in the payload,it cannot be translated when it passes through the NAT-PT device. As a result, even though the uTorrent client of an IPv6 host and the Tracker server support IPv6, the client still can not download IPv4 resources from IPv4 peers via NAT64 device.

3.5. Gaming

Another application we have tested is online games. We cannot log in to most gaming platforms unless they uses domain name. It is presumably because the game client does not support IPv6. We cannot make further experiments before the IPv6-supported clients are released.

3.6. VPN

The VPN testing is to estimate whether the VPN client can initial a connection to the remote access server through a NAT64 device. The testing is based on Windows Vista (as the VPN client) and Windows Server 2008 R2 Standard (as the RAAS). Two protocols were applied to connect to the remote access server: L2TP/IPSec and PPTP. The results show that the PPTP protocol does not support IPv6 while L2TP/IPSec technology supports IPv. However, the Internet Key Exchange failed when passing through the NAT64.

3.6.1. Stream Media Player

Other applications we have tested include online stream media player software (e.g. PPTV, PPStream, UUSee), third Party FTP client and Remote cooperation/assistant tools (e.g. pcAnywhere and Windows Remote Desktop). The online stream media player can download the playlist and advertisements normally, but it was unable to connect to the media server and play the media contents.

4. Conclusions

This document discusses our experiences from deploying NAT64 devices for various Internet applications. The main conclusion is that two problems exist from our experimention. First is the weakness of the IPv6 capability of user end clients, and the second problem is that the IPv4 addresses can not be translated when they are carried inside a packet's payload.

5. Security Considerations

None.

6. Informative References

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (Work in progress)", July 2010.

Authors' Addresses

Jinghua Tan
China Telecom
109, Zhongshan Ave. West,
Tianhe District, Guangzhou 510630
P.R. China

Phone:
Email: tanjh@gsta.com

Jinyan Lin
China Telecom
109, Zhongshan Ave. West,
Tianhe District, Guangzhou 510630
P.R. China

Phone:
Email: jasonlin.gz@gmail.com

Weibo Li
China Telecom
109, Zhongshan Ave. West,
Tianhe District, Guangzhou 510630
P.R. China

Phone:
Email: MWeiboLI@gmail.com

v6ops WG
Internet-Draft
Obsoletes: 3056 (if approved)
Intended status: Standards Track
Expires: September 11, 2011

O. Troan
G. Van de Velde
Cisco

March 10, 2011

Request to move Connection of IPv6 Domains via IPv4 Clouds (6to4) to Historic status
draft-troan-v6ops-6to4-to-historic-01.txt

Abstract

Experience with the "Connection of IPv6 Domains via IPv4 Clouds (6to4)" IPv6 transitioning mechanism has shown that the mechanism is unsuitable for widespread deployment and use in the Internet. This document requests that RFC3056 and the companion document "An Anycast Prefix for 6to4 Relay Routers" RFC3068 are moved to historic status.

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 September 11, 2011.

1. Introduction

The IPv6 transitioning mechanism "Connection of IPv6 Domains via IPv4 Clouds (6to4) described in [RFC3056] and the extension in "An Anycast Prefix for 6to4 Relay Routers" RFC3068 [RFC3068] have been shown to have severe practical problems being used in the Internet. This document requests that RFC3056 and RFC3068 be moved to Historic status as defined in section 4.2.4 [RFC2026].

See also the document Non-Managed IPv6 Tunnels considered Harmful [I-D.vandevelde-v6ops-harmful-tunnels] for details.

[I-D.kuarsingh-v6ops-6to4-provider-managed-tunnel] are proposing a mechanism using IPv6 NAT to solve the 6to4 reverse path problem.

[I-D.carpenter-v6ops-6to4-teredo-advisory] are proposing a set of suggestions to improve 6to4 reliability.

Declaring the mechanism historic is not expected to have immediate product implications. The IETF sees no evolutionary future for the mechanism and it is not recommended to include this mechanism in new implementations.

2. Conventions

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

3. 6to4 operational problems

6to4 is a mechanism designed to allow isolated IPv6 islands to reach each other using IPv6 over IPv4 automatic tunneling. To reach the native IPv6 Internet the mechanism uses relay routers both in the forward and reverse direction. The mechanism is supported in many IPv6 implementations. With the increased deployment of IPv6, the mechanism has been shown to have a number of fundamental shortcomings.

6to4 depends on relays both in the forward and reverse direction to enable connectivity with the native IPv6 Internet. A 6to4 node will send IPv4 encapsulated IPv6 traffic to a 6to4 relay, that is

connected both to the 6to4 cloud and to native IPv6. In the reverse direction a 2002::/16 route is injected into the native IPv6 routing domain to attract traffic from native IPv6 nodes to a 6to4 relay router. It is expected that traffic will use different relays in the forward and reverse direction. RFC3068 adds an extension that allows the use of a well known IPv4 anycast address to reach the nearest 6to4 relay in the forward direction.

One model of 6to4 deployment as described in section 5.2, RFC3056, suggests that a 6to4 router should have a set of managed connections (read BGP peers) to a set of 6to4 relay routers. While this makes the forward path more controlled, it does not help the reverse path. In any case this model has the same operational burden has manually configured tunnels and has seen no deployment in the public Internet.

6to4 issues:

- Use of relays. 6to4 depends on the charity of an unknown third-party to operate the relays between the 6to4 cloud and the native IPv6 Internet. With the use of mechanism specified in [RFC3068] in both directions, without it only in the reverse direction (from native to 6to4) [RFC3056].
- The placement of the relay can lead to increased latency, and in the case the relay is overloaded packet loss.
- There is generally no customer relationship or even a way for the end-user to know who the relay operator is, so no support is possible.
- In case of the reverse path 6to4 relay and the anycast forward 6to4 relay, these have to be open for any address. Only limited by the scope of the routing advertisement. 6to4 relays can be used to anonymize traffic and inject attacks into IPv6 that are very difficult to trace.
- 6to4 has no specified mechanism to handle the case where the protocol (41) is blocked in intermediate firewalls. It can not be expected that path MTU discovery across the Internet works reliably; ICMP messages may be blocked and in any case an IPv4 ICMP message rarely has enough of the original packet in it to be useful to proxy back to the IPv6 sender.
- As 6to4 tunnels across the Internet, the IPv4 addresses used must be globally reachable. RFC3056 states that a private address [RFC1918] MUST NOT be used. 6to4 will not work in networks that employ addresses with limited topological span.

4. Recommendations for 6to4 Relay Operators

See [I-D.carpenter-v6ops-6to4-teredo-advisory].

5. Recommendations for implementors

If the implementation continues to support 6to4, then the 6to4 functionality MUST NOT be enabled by default.

If the implementation continues to support 6to4, then the Source Address Selection algorithm [RFC3484] MUST use a 6to4 address as a last resort. I.e. only use it the node has no other means of IPv6 connectivity and the destination is IPv6 only.

6. IANA Considerations

This specification does not require any IANA actions.

7. Security Considerations

There are no new security considerations pertaining to this document. General security issues with tunnels are listed in [I-D.ietf-v6ops-tunnel-security-concerns] and more specifically to 6to4 in [I-D.ietf-v6ops-tunnel-loops] and [I-D.vandevelde-v6ops-harmful-tunnels].

8. Acknowledgements

The authors would like to acknowledge Fred Baker, Jack Bates, Cameron Byrne, Brian Carpenter, Gert Doering, Joel Jaeggli, Jason Livingood, Keith Moore, Daniel Roesen and Mark Townsley, for their contributions and discussions on this topic.

9. References

9.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains

via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

9.2. Informative References

[I-D.carpenter-v6ops-6to4-teredo-advisory]
Carpenter, B., "Advisory Guidelines for 6to4 Deployment", draft-carpenter-v6ops-6to4-teredo-advisory-02 (work in progress), February 2011.

[I-D.ietf-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-ietf-v6ops-tunnel-loops-04 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-security-concerns]
Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns With IP Tunneling", draft-ietf-v6ops-tunnel-security-concerns-04 (work in progress), October 2010.

[I-D.kuarsingh-v6ops-6to4-provider-managed-tunnel]
Kuarsingh, V., Lee, Y., and O. Vautrin, "6to4 Provider Managed Tunnels", draft-kuarsingh-v6ops-6to4-provider-managed-tunnel-01 (work in progress), February 2011.

[I-D.vandevelde-v6ops-harmful-tunnels]
Velde, G., Troan, O., and T. Chown, "Non-Managed IPv6 Tunnels considered Harmful", draft-vandevelde-v6ops-harmful-tunnels-01 (work in progress), August 2010.

Authors' Addresses

Ole Troan
Cisco
Oslo,
Norway

Email: ot@cisco.com

Gunter Van de Velde
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2704 5473
Email: gvandeve@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 16, 2011

T. Tsou, Ed.
Huawei Technologies(USA)
T. Taylor
Huawei Technologies
March 15, 2011

Multicast Transition to IPv6 Only in Broadband Deployments
draft-tsou-v6ops-multicast-transition-v6only-01

Abstract

This document proposes a multicast transition solution from the old IPv4 only network to the IPv6 only network. It enumerates the transition steps and then analyzes the transition cost in various dimensions.

This document is intended to eventually meet the criteria for a specification in the series envisioned by the v4-to-v6 transition framework.

Table of Contents

1. Introduction

[ID.v4v6tran-framework] defines the required content for a series of documents describing how to move from IPv4 to IPv6 for specific network scenarios. The present document is an initial sketch of one such document. Content will be added in later versions to allow it to meet the criteria set by [ID.v4v6tran-framework].

The handling of unicast during the transition from IPv4 to IPv6 is the focus of a considerable amount of activity within the BEHAVE and SOFTWIRES Working Groups, which have worked on tools such as NAT64, 6rd, and DS Lite. At the same time, even though some ISPs have chosen 6rd or dual stack as their unicast transition solution, they want to keep their IPv4 only multicast system unchanged as long as possible, simply because they have enough IPv4 multicast address. While IPv4 unicast addresses will soon be exhausted, ISPs have no motivation to update multicast until the day when there are few IPv4 unicast users, it is near the point where the IPv4 stack in network equipment can be turned off, and the update of the network to IPv6 only is nearly complete.

This document discusses a multicast transition model to keep the old IPv4 only multicast service while 6rd or dual stack is deployed for unicast transition, and then to migrate the IPv4 only multicast system to an IPv6 only multicast system "directly".

1.1. Requirements Language

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

2. Scope

The multicast framework proposed here corresponds to some unicast transition scenarios in the SOFTWIRES Working Group as follows:

1. The access network moves from IPv4 only to 6rd [RFC5969] and then to dual stack and finally to IPv6 only. This path may be preferred by some ISPs.

2. The access network moves from IPv4 only to dual stack directly, and then to IPv6 only. This path may be preferred by other ISPs.

3. The access network is deployed from its beginning as an IPv6 only network.

This document considers unicast transition Scenarios 1 and 2. Scenario 3 is not considered in this draft because DS-Lite is good as its unicast solution and [ID.qin-dslite-multicast] is good as its multicast solution.

3. Transition Steps

The multicast transition solution has four steps as described below.

3.1. The Old IPv4 Only Multicast Network

In this stage, both unicast and multicast services are based on an IPv4 only network.

3.2. First Upgrade

In this stage, the user IPTV terminal is either IPv4 only or has been upgraded to dual stack. The network is now dual stack. Multicast sources continue to be IPv4.

The IPv4 core network is changed to dual stack to support more and more use of IPv6 unicast, but not multicast in this stage. In a variation, some ISPs may choose to start with 6rd and then move to dual stack as their unicast transition solution. The corresponding multicast solution in that scenario is the same as for a direct move to dual stack.

In this stage, new dual stack IPTV terminals may be deployed only for compatibility with the future IPv6 only network.

This stage may exist in more than 15 years. At the end of the stage, IPv4 unicast traffic may only take up at most 10% of the total bandwidth.

Moreover, at the end of this stage, the IPv4 source servers should be updated to dual stack. The IPv6 stack is operated only for testing, just make sure it will work well in the next stage when the day comes that the ISP decides to turn off the IPv4 stack in all equipment in its network.

3.3. Second Upgrade, Dropping IPv4

In this stage the user terminal is either the dual stack device introduced in the previous stage or a new IPv6 only IPTV Terminal. The network and the IPTV source are both IPv6 only.

This stage begins when the ISP finds that the IPv4 unicast traffic in

its network is insignificant and decides to turn of all IPv4 stacks in all of its network devices. At that point the IPv4 only IPTV terminal will be useless, and the IPv4 stack of the source servers will be turned off, too.

3.4. The Pure IPv6 Multicast Network

In the final stage, the old dual stack IPTV terminals disappear. The network is purely IPv6.

4. Analysis of the Multicast Transition Cost

4.1. IPTV Source Server

The IPv4 IPTV source servers may operate for more than 15 years, so that the ISP investment in the old IPv4 IPTV system is well protected. Only at the end of stage 2 (Section 3.2) must the ISP add the IPv6 stack to the source server for testing in preparation for stage 3.

4.2. Bandwidth Consumed By Multicast

Because the production servers are always running just one IP version, bandwidth consumption for multicast is not affected by the transition. The only exception is at the end of the second stage, when the servers are upgraded to dual stack. Stray customer IPv6 traffic could boost bandwidth, but this can be prevented by proper filtering to allow IPv6 access only to test traffic for the moment.

4.3. Border Devices

The suggested evolution path avoids the need to deploy the NAT64 function in border routers, such as the Border Relay in 6rd unicast deployment. NAT64 can seriously degrade performance.

4.4. IPTV Terminal

The suggested evolution path requires an upgrade to IPTV terminals over a period in the order of 15 years, from IPv4 to dual stack. Later, the IPv4 stack in these terminals has to be turned off. In the long run they will be replaced by IPv6 terminals. The time frames involved are probably longer than the working life of an individual terminal, so that no extra investment is involved.

5. Security Considerations

TBD

6. IANA Considerations

This document requires no IANA action.

7. Acknowledgements

Thanks to Fred Baker for preliminary comments..

8. References

8.1. Informative References

[ID.qin-dslite-multicast]
Wang, Q., Qin, J., Sun, P., Boucadair, M., Jacquenet, C., and Y. Lee, "Multicast Extensions to DS-Lite Technique in Broadband Deployments (Work in progress)", January 2011.

[ID.v4v6tran-framework]
Carpenter, B., Jiang, S., and V. Kuarsingh, "Framework for IP Version Transition Scenarios", February 2011.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

8.2. Normative References

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

Authors' Addresses

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

Phone: +1 408 330 4424
Email: tena@huawei.com

Tom Taylor
Huawei Technologies
1852 Lorraine Ave
Ottawa, Ontario K1H 6Z8
Canada

Phone:
Email: tom111.taylor@bell.net

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 20, 2011

D. Zhang
Huawei Symantec

February 16, 2011

Solution Model of Source Address Tracing for Carrier Grade NAT (CGN)
draft-zhang-v6ops-cgn-source-trace-00.txt

Status of This Memo

Copyright Notice

Abstract

Since NAT function on CGN box will make the IPv4 address re-used by more than one user, the packets sent outside CGN are not able to be identifid where they are from or which user they belong to according to the source address within the packets. However, under some certain circumstances, knowing the original source IP address and the identity of the user who sends the packet out is necessary. This document states the requirement of source address tracing briefly, and discusses the possible solution models for this issue.

Table of Contents

1. Introduction

At this time of the document written, IPv4 addresses for IANA have been depleted. The network is going to experience a long migration stage to IPv6. Some transition solutions have been proposed, such as NAT444, DS-Lite, NAT64 and 6rd. In these solutions, translation is an important technology. The translator which executes the translation function in service provider network means Carrier Grade NAT (CGN) [I-D. draft-ietf-behave-lsn-requirements-00]. Here, the CGN scope in this document includes NAT44 and NAT64.

NAT function on CGN box may make the IPv4 address in its address pool re-used by more than one user probably. Thus the packet seen from the outside of CGN cannot be identified based on the source address in the packet, which means it is difficult to know what the original source IP address is before translation and which user sends the packet. As the service providers consider their deployment solution of IPv6 transition, the source address tracing issue has been emphasized explicitly. Under some certain circumstances it is required tracking the source of the packet. Therefore, it is helpful for service provides to give a clear introduction on how to achieve the tracing of source address. This document states the requirement for source address tracing and discusses the possible solution models for this issue.

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. Requirement Statement

The requirement of tracing the source address is forced by mainly three reasons. All the requirements below can happen not only in Fixed BroadBand (FBB) network, but also in Mobile BroadBand (MBB) network. And [I-D. draft-ietf-v6ops-v6-in-mobile-networks-03] indicates the same issue as well, espectially for mobile network.

3.1. Legal Requirement

CGN box produces log records in which contains the session information used for packet translation. Because of the huge number of NAT log, the log records will be exported to a external log server. In order to monitor Internet, for some carriers, they are demanded conserving the NAT logs for a few months. Once an illegal behavior is present on Internet, legal department will request the carrier find out the subscriber who did it. Depending on the

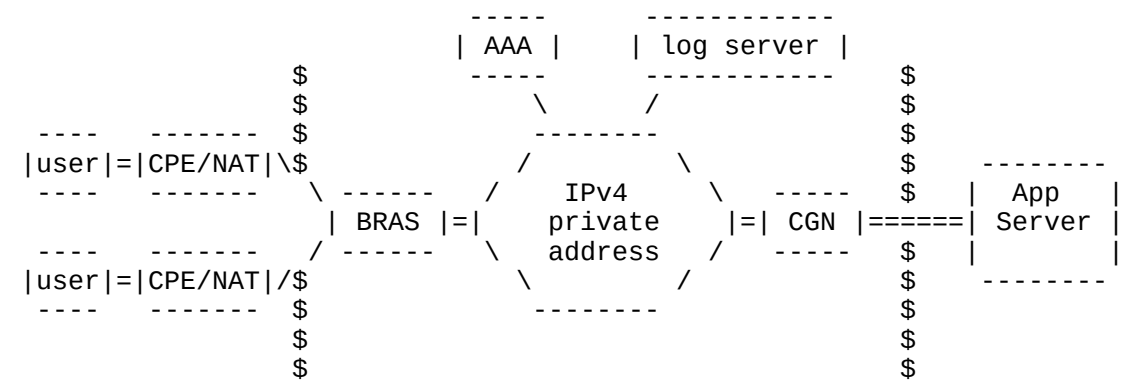
conserved logs, the carrier is capable of fulfilling the task.

This type of requirement can be regards as non-realtime requirement. Generally, the source address tracing may happen later than the NAT binding state has been deleted on NAT box.

3.2. Application Requirement

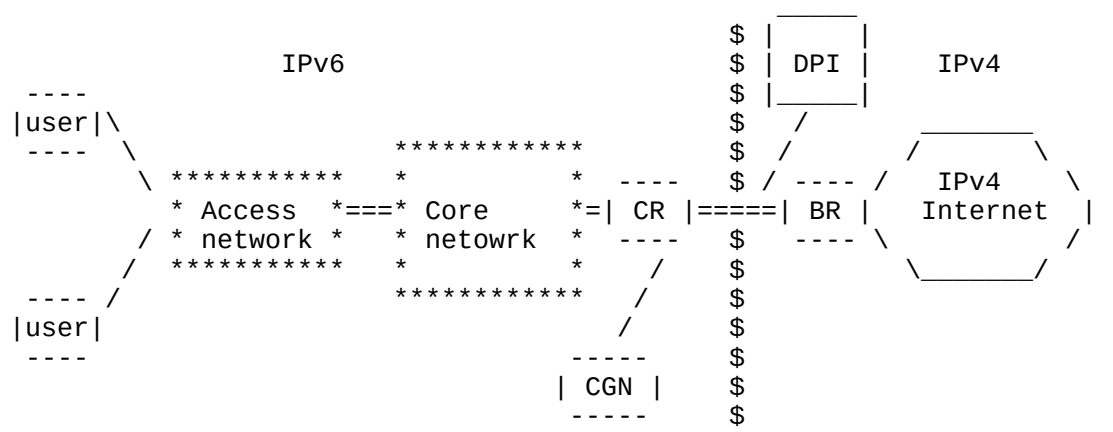
Carriers may provide some applications to the users. The application usually is a kind of application or service operated by the carrier itself. The application or service will provide the users who have subscribed it. For instance, a carrier supplies the subscribers with video, game, email and even broadband sevice management (maybe portal) by a unified web server. As a result, the subscriber identification may be needed. When a user visits the special application, the application server will identify the user by the source address. However, because of the deployed CGN, the IP address got from the source address field in the packet may be shared by more than one user. Thus, the application server must authenticate the subscriber by obtaining the original unique IP address assigned by the carrier, either an IPv4 private address or an IPv6 address. As a matter of fact, this requirement is a real-time trait, which is unlike the former one.

The figure depicts the NAT444 case. The service provider allocates non-duplicated IPv4 address to different user's CPE. When the application server receives a packet translated by CGN, it will be confused which user the packet belongs to. Therefore, the authentication of application server could be incorrect.



3.3. Existing Device Constriction

Service providers also offer value-added services by advanced technologies, such as DPI, P2P cache and so on. Here DPI is token as an example. It seems that most of the DPI products deployed in the network currently cannot support IPv6. It needs time for DPI vendors to develop IPv6 features. As depletion of IPv4 address, if a carrier intends to deploy IPv6-only network and use NAT64 to help users access IPv4 Internet, the placement of CGN should be considered carefully. This is a sort of real-time requirement as well.



The picture shows an example of possible network topology. For different service providers, there may be a variation. Since the existing DPI device does not support IPv6, the CGN with NAT64 function should be located lower, leaving the DPI in IPv4 realm. The carrier provides a value-added service, which depends on DIP to deal with billing and accounting. Because not all the users will subscribe the value-added service, and the source IPv4 addresses have been shared, DPI is not able to attach the traffic to the exact user. As a result source address tracing is inevitable in this case.

4. Solution models for Source Address Tracing

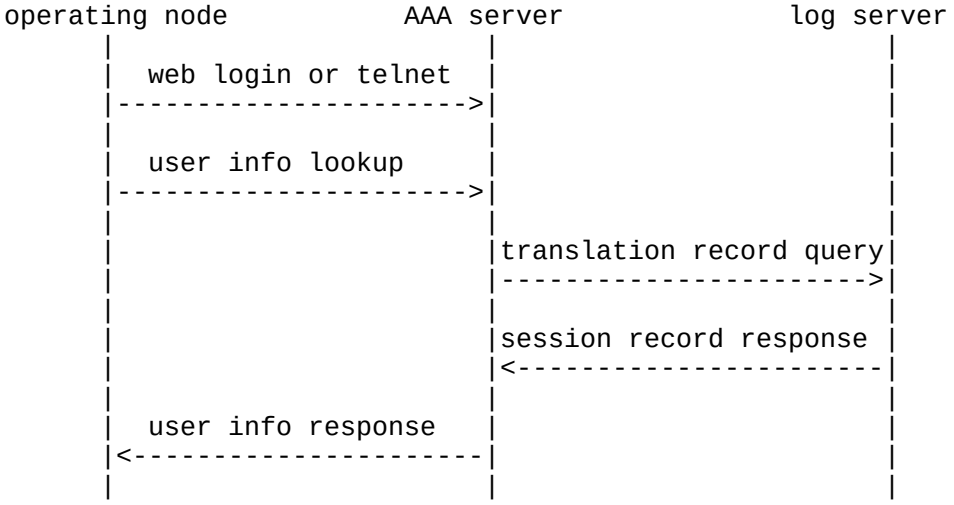
In order to meet the foregoing requirements, service providers have to take the solution of source address tracing into account. In this section, the possible tracing models are discussed for reference.

4.1. Non-realtime Tracing Model

The non-realtime tracing of source address is always suitable for legal requirement.

The aim of tracing is to find out the user information. The lookup action happening on AAA server is indispensable. Hence, the AAA server should be provided with necessary lookup means, such as web, telnet and so on. As AAA server only has the binding between the user information and its IP address assigned originally, the log server is requested working together with AAA server. AAA server will obtain the translation binding according to the public IP address and using time from the log server. The public address and using time may be given by legal department. In this process, a special interface on log server should be implemented for responding the AAA's query message.

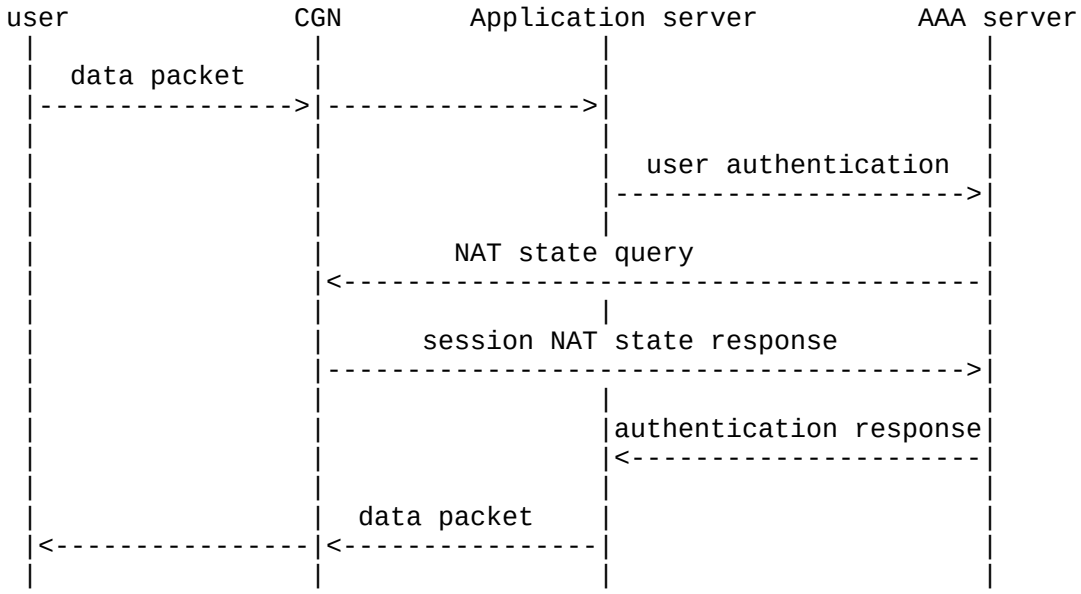
The process could be demonstrated by the following figure. The operating node would be any PC or terminal by which the carrier launches the source address tracing.

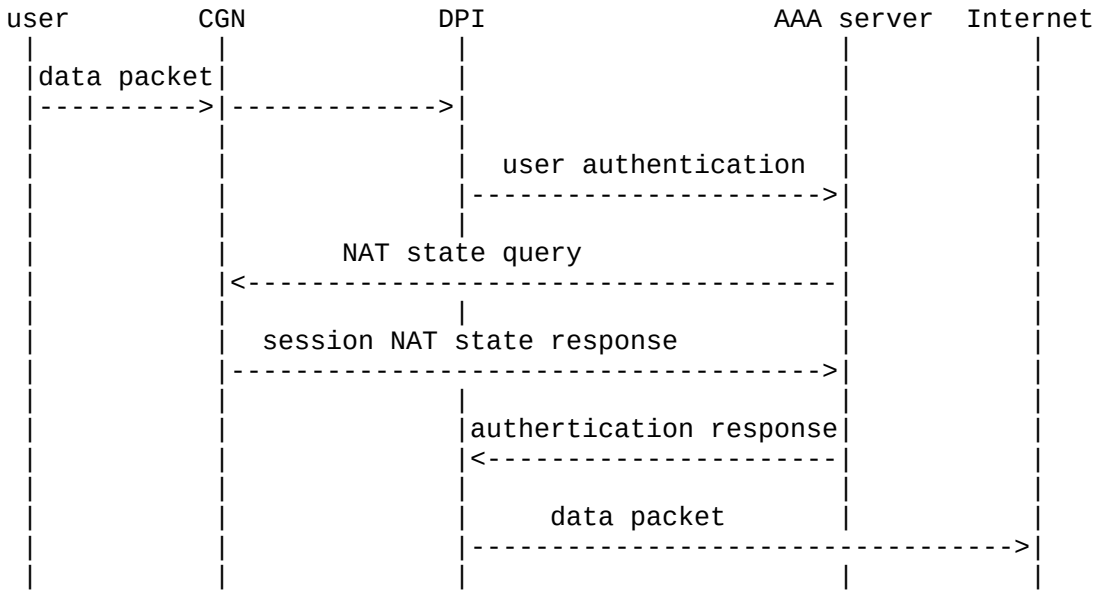


4.2. Realtime Tracing Model

The realtime tracing satisfies the need of user identity authentication of certain services or subscriber management, e.g. policy and billing. The tracing takes place when the user is online and initializing the service accessing. On account of this, the translation binding state is still preserved on CGN. Therefore, the NAT binding of session will be acquired from CGN, but not log server, in realtime tracing model. (If getting the binding state by log server with the same way as the non-realtime model, it could be unreliable. It is because that some CGN implementations may not send out the log message to log server before the session is expired or the session state is deleted.) So, CGN is demanded a interface for querying the NAT binding of session.

The tracing procedures for the second and third requirements in section 3 can be seen as follows.





5. Security Considerations

TBD

6. IANA Considerations

This document has no IANA actions.

7. Acknowledgement

8. References

8.1. Normative References

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

8.2. Informative References

[I-D. draft-ietf-behave-lsn-requirements-00] Yamagate, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for IP address sharing schemes", April 2011.

[I-D. draft-ietf-v6ops-v6-in-mobile-networks-03] Koodli, R., "Mobile Networks Considerations for IPv6 Deployment", January 2011.

Author’s Address

Dong Zhang
Huawei Symantec
China

EMail: zhangdong_rh@huaweisymantec.com