Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 14, 2008

M. Bagnulo
Huawei Labs at UC3M
F. Baker
Cisco Systems
I. van Beijnum
IMDEA Networks

May 13, 2008

IPv4/IPv6 Coexistence and Transition: Requirements for solutions
draft-ietf-v6ops-nat64-pb-statement-req-00

Status of this Memo

Abstract

This note presents the problem statement, analysis and requirements for solutions to IPv4/IPv6 coexistence and eventual transition in a scenario in which dual stack operation is not the norm.

Table of Contents

1. Introduction

This note addresses requirements for solutions to IPv4/IPv6 coexistence and eventual transition in a scenario in which dual stack operation is not the norm.

2. Problem statement

Operationally, we now expect the transition to be less a matter of connecting ever-growing IPv6 islands in an IPv4 network, and more a matter of the network becoming a patchwork quilt of IPv4, IPv6, and dual domains.

- Hosts now generally support IPv6 and IPv4 natively.
- As described in [RFC4213], the IETF community had expected administrations to turn on IPv6 in their existing IPv4 networks, resulting in a simple coexistence scenario.
- Increasingly, we hear statements that people want to move directly to an IPv6-only or IPv6-dominant network.

In this context, "IPv6-only" refers to a network or system that only runs IPv6, and "IPv6-dominant" refers to a network or system that may use IPv4 internally or with other clients, but in the context only routes IPv6 datagrams. "IPv4-only" and "IPv4-dominant" are defined similarly. Since these are indistinguishable to the peer, the terms "IPv4-only" and "IPv6-only" will be used in this paper and considered to subsume the "dominant" issues.

2.1. Transition scenarios

There are six obvious transition scenarios:

- IPv4 system connecting to an IPv4 system across an IPv4 network,
- An IPv6 system connecting to an IPv6 system across an IPv6 network,
- an IPv4 system connecting to an IPv4 system across an IPv6 network,
- an IPv6 system connecting to an IPv6 system across an IPv4 network,
- an IPv4 system connecting to an IPv6 system, or
- an IPv6 system connecting to an IPv4 system.

2.1.1. Simple transition scenarios

The simplest coexistence cases are about an IPv4 system connecting to an IPv4 system across an IPv4 network, or an IPv6 system connecting to an IPv6 system across an IPv6 network. The dual stack case, in which both endpoints and the relevant applications support IPv4 and IPv6 and the network supports at least one of the protocols, falls

into this case as the applications can connect using whichever stack is consistent end to end.

The IETF strongly prefers and recommends this scenario, as the operational matters are the simplest. Until the Internet reaches IPv4 address exhaustion, an IPv4 and an IPv6 address can be assigned to every interface, and the applications are supported. When it becomes necessary to deploy only IPv6 addresses, since all other systems have both, IPv6-only systems cleanly interoperate with existing systems.

2.1.2. Transition scenarios that do not require translation

[RFC4213] discusses the scenario in Figure 1, in which routers connect two dual domains via an IPv4-only domain. Obviously, this can be reversed: routers can connect two dual domains via an IPv6-only domain. Note that the connecting domain need not actually be IPv4-only or IPv6-only; to create this scenario, it need merely fail to offer IPv6 or IPv4 services to the neighboring domains.

```
                    ,-.             ,-.             ,-.
                  ,'   `.         ,'   `.         ,'   `.
                 ;       :       ;       :       ;       :
                 ; IPv4+ :       ; IPv4- :       ; IPv4+ :
                ;  IPv6   :     ;   only  :     ;  IPv6   :
                ;  Domain :     ;  Domain :     ;  Domain :
               ;           :   ;           :   ;           :
               |  +----+   |   |  +----+   |   |  +----+   |
               |  |IPv4|   |   |  |IPv4|   |   |  |IPv4|   |
               |  |Host+   |   |  |Host|   |   |  |Host|   |
               :  +----+\  ;   : /+----+\  ;   :/ +----+   ;
                : +----+ \+------+       +------+ +----+  ;
                : |IPv6+--+Router+=======+Router+-+IPv6|  ;
                 :|Host| ;+------+       +------+:|Host| ;
                 :+----+ ;       :       ;       :+----+ ;
                  `.   ,'         `.   ,'         `.   ,'
                    `-'             `-'             `-'
```

Figure 1: Disconnected continuity

In such a scenario, there are two obvious solutions: one can tunnel across the connecting domain, as shown, or one can translate between IP layers using something akin to traditional NAT technology. The tunnel approach offers some pros and some cons: it natively connects the dual domains, meaning that all applications should work, but they may have issues with the path MTU, and the tunnels require some form of configuration. The NAT approach similarly offers pros and cons: it offers something similar to standard routing, but it suffers from

the various ills of Network Address Translation on both sides, meaning that it may be difficult for the dual domains to offer services to each other.

In general, the IETF recommends the use of tunnels rather than a dual NAT.

There are at least three generic models that could be used to describe this kind of tunneling scenario:

- o Static tunnels with interior dynamic routing
- o Start-time negotiated tunnels to some central point with default routing (example in [I-D.stenberg-v6ops-pd-route-maintenance])
- o Dynamic tunnels with specific routing to islands (examples might include ISATAP [RFC4214] or a tunnel broker of some description)

Static tunnels with routing through them are commonly deployed today, both in VPNs and in overlay networks. The positive side is that they provide simple service; the negative is that the generally require manual configuration and can result in suboptimal routing.

A "start-time" tunnel might be useful in an access network that serves homes or SOHO environments. In this model, the ISP informs the CPE of a cross-network peer that it can create a tunnel to, reducing the case to one similar to static tunneling but without manual configuration.

A dynamic tunneling environment is an overlay model in which systems create tunnels to various peers across the connecting domain as needed, based on a priori knowledge of the correlation between remote prefixes and next hop routers. This has not been adequately described at this point, and therefore involves complexities in implementation and deployment.

2.1.3. Transition scenarios that require translation

Translation, as found in Figure 2, is considered in NAT-PT [RFC2766], which has in turn been set aside via [RFC4966]. In essence, translation is required when an IPv4-only system connects to an IPv6-only system or an IPv6-only system connects to an IPv4-only system. These systems need not actually be IPv4-only or IPv6-only; if the connecting network is IPv4-only or IPv6-only and provides no tunnel, but only offers IPv4 service to one and only offers IPv6 service to the other, the situation is equivalent.

```
                      ,-----.                 ,-----.
                    ,'       `.             ,'       `.
                   /           \           /           \
                  /   IPv4-only \         /   IPv6-only \
                 /    Domain   +-----------+  Domain     \
                ;              |Translation|              :
                |              |  Gateway  |              |
                :              +-----------+              ;
                 \     +----+    /       \     +----+    /
                  \    |IPv4|   /         \    |IPv6|   /
                   \   |Host|  /           \   |Host|  /
                    `. +----+,'             `. +----+,'
                      '-----'                 '-----'
```

Figure 2: Translation

In such a scenario, it is necessary for the network to create a translation gateway, at which datagrams from one system are translated forwarded to the other. The situation is in many ways reflexive, since most Internet sessions are bidirectional - TCP between an IPv4 and an IPv6 system translate data messages in one direction and acknowledgments in the other.

They are not reflexive, however, in the distribution of domain names. If the application is client-server and the server is in one of the domains, the name of the server need only be propagated to the other. Reverse lookups, frequently used in spam verification would require the client's name to be propagated into the server's domain. But in this there are issues. The address of the client (the TCP peer) as seen by the server is not the remote system in the other domain; it is the translator. This is readily worked around for an IPv6 server, as the IPv4 address of the remote peer can be embedded in a "privacy" address [RFC4941], making the reverse lookup viable. This doesn't work on the IPv4 side, however.

2.2. Requirements for the overall transition strategy

Given the problem statement presented here, we see the following requirements for a complete transition strategy:

1. Any transition strategy must contemplate a period of coexistence, with ultimate transition (e.g., turning off IPv4) being a business decision.
2. Many are delaying turning on IPv6 (initiating coexistence in their networks) as long as possible.
3. Some are turning off IPv4 immediately, at least as a customer service.

4. Therefore, dual stack approaches, tunneled architectures, and translation architectures are all on the table.
5. Any solution that makes translation between semi-connected islands "normal" has failed the fundamental architecture of the Internet and can expect service complexity to be an issue. [RFC3439]
6. Translation architectures must provide for the advertisement of IPv4 names to IPv6 systems and vice versa. The address advertised in the "far" domain must be that of the translating gateway.
7. Tunneling architectures must provide a way to minimize and ideally eliminate configuration of the tunnel.

3. Preliminary analysis for translation mechanisms

3.1. Application behavior taxonomy

The general purpose of NAT64 type of mechanisms is to enable communication between a v4-only node and a v6-only node. However, there is wide range of type of communications, when considering how they handle IP addresses. So, in order to properly characterize the problem, we need to do an analysis of the different application behavior in terms of the usage of their IP addresses. We will next present a taxonomy of the behavior of the application with respect of how they use the IP address. The support of the different type of behavior will impose a different set of constraints to the design of a NAT64 mechanisms. It is then important to decide which type of application behavior will be supported before starting to design a NAT64 mechanism. The proposed taxonomy is heavily based on the one presented in section 1.1 of draft-ietf-shim6-app-refer-00.txt.

The proposed application behavior taxonomy is the following:

Short-lived local handle. The IP addresses is never retained by the application. The only usage is for the application to pass it from the DNS APIs (e.g., getaddrinfo()) and the API to the protocol stack (e.g., connect() or sendto()). This type of communication can be either initiated by the v4-only node or by the v6-only node, resulting in two type of behaviors, v4-initiated short lived local handle and v6-initiated short lived local handle.

Long-lived application associations. The IP address is retained by the application for several instances of communication. However, it is always the same node that initiates the communication. This type of communication can be either initiated by the v4-only node or by the v6-only node, resulting in two type of behaviors, v4-initiated long-lived associations and v6-initiated long-lived associations.

Callbacks. The application at one end retrieves the IP address of the peer and uses that to later communicate "back" to the peer. This type of communication can be either initiated by the v4-only node or by the v6-only node, resulting in two type of behaviors, v4-initiated callback, meaning that the initial communication is initiated by the v6-only node, and later the v4-only node initiates the callback, and v6-initiated callback, meaning that the initial communication is initiated by the v4-only node, and later the v6-only node initiates the callback. An additional disticntion can be made based on the time-frame of the call back operation. There can be short-lived call-backs, where the receiver inmediatelly calls back to the initiator and long-lived call-backs where the receiver calls backs after a while.

Referrals. In an application with more than two parties, party B takes the IP address of party A and passes that to party C. After this party C uses the IP address to communicate with A. In this type of communication, the following 6 sub-cases are possible.

o A and B are v6-only nodes and C is a v4-only node;
o A and C are v6-only nodes and B is a v4-only node,
o B and C are v6-only nodes and A is a v4-only node,
o A and B are v4-only nodes and C is a v6-only node;
o A and C are v4-only nodes and B is a v6-only node,
o B and C are v4-only nodes and A is a v6-only node,

"Identity" comparison. Some applications might retain the IP address, not as a means to initiate communication as in the above cases, but as a means to compare whether a peer is the same as another peer. While this is insecure in general, it might be something which is used e.g., when TLS is used. This type of communication results in two sub-cases, when the v4-only node performs comparison of the v6-only node identity, and when the v6-only node performs comparison of the v4-only node identity

3.2. Placement of the NAT64 mechanisms

Another aspect that is critical to design a NAT64 mechanism is the placement of the mechanisms involved. In other words, what elements can be modified/updated to support the NAT64 mechanisms. We assume that the NAT64 box supports a set of mechanisms that are the core part of the solution, but some approaches may require the modification of additional elements. In particular, we can identify the following additional elements that may require modification to support a NAT64 approach.

Modification to v4-only nodes: one option is to require modification to existent v4-only nodes in order to support the NAT64 mechanism. This option would impose high deployment costs, because the existent

base of v4-only nodes is really big and there is no incentives for the v4-only nodes to install such mechanism, since it seems unlikely that v4-only nodes will have a strong need to communicate with v6-only nodes (at least at the initial stages of v6 deployment). However, it may be possible that this is the only viable solution for supporting some type of application behavior.

Modification to v6-only nodes: Another option is to require modifications to v6-only nodes. This option seems much more acceptable, since the existent base of v6-nodes is relatively small and there would be a strong incentive for v6-only nodes to communicate with v4-only nodes, since most of the contents are available only in v4 today. However, imposing modifications to v6-only nodes does make deployment of the solution more difficult, since update of current v6-implementations is needed. In addition, there is an architectural consideration, that we would be imposing v6-only nodes to support "NAT hacks" in order to enable communication with the v4 world, and that those modifications may stay forever, even when the need for communication with the v4-Internet is not so pressing.

Modification to both v4-only nodes and v6-only nodes. Another option is to require updates to both v4-only nodes and also to v6-only nodes. Needless to say that this would be the option with higher deployment costs.

No modification. Another option is that the NAT64 mechanisms does not require modification to any host and that the mechanism is fully contained in the NAT64 box. This was the case of the previously defined NAT-PT approach. However, it may be challenging to design a solution with this constraint that does not suffer the limitations suffered by the NAT-PT mechanism that lead the IETF community to deprecate it.

Another consideration related to the modification imposed by a NAT64 approach is about what elements in the nodes need to be updated. In particular, it is important to determine if only the IP layer on the affected nodes needs to be modified or f other elements in the nodes needs to be updated. In particular, it is critical to determine if applications need to e modified in order to support the NAT64 mechanism.

3.3. v4 addressing consideration

We assume that both the v6-only nodes and the v6 interface of the NAT64 boxes will have routable IPv6 addresses. However, on the v4 side, there are more options. Either the v4 interface of the NAT64 boxes and/or the v4-only nodes can have either v4 private addresses

or v4 public addresses. Actually, it is possible that the different combinations make sense. It seems clear that the case where public v4 addresses are used in both the v4 interface of the NAT64 box and the v4-only nodes is relevant. The case where the v4-only node has a private v4 address and the NAT64 box has a public address seems also possible, but here it seems reasonable to assume that a NAT box will exist between the v4 only node and the NAT64 box. The case where both the v4 node and the NAT64 box have v4 private addresses could also make sense, since this could apply to a scenario where a site that has v4 private addresses and v6 addresses could try to use a NAT64 box internally. The last case, where the v4 node has public address and the NAT64 box has a private address seems harder to justify though.

Another consideration related to v4 addressing of the NAT64 approach is the number of addresses required by the NAT64 box. It is possible that some NAT64 approaches require a pool of v4 addresses instead of a single v4 address. Considering the status of the v4 address space consumption, it may not be feasible to use a NAT64 approach that require a big number of v4 public addresses.

3.4. Name-space considerations

One of the major choices that are faced when designing a NAT4 mechanism that enable communication initiated by the v4-only node towards a v6-only node. In this case, the v4 only node needs to identify the v6 only node and the problem is that there is no means to permanently map the v6 address space in the v4 address space. So in order to enable a v4-only node to identify a v6-only node a name space other than the IPv4 address space is needed. We will next discuss some options that could be considered to identify v6 nodes in the v4 world.

A first option is to use IPv4 addresses to identify IPv6 nodes. The problem is that the v6 address space is much bigger than the v4 address space, so it is not possible to do permanent mapping between these two. This basically implies that dynamic mapping between a given v4 address and different v6 addresses are established. While this works for some type of application behavior, it does not support others, such as communications initiated by a v4 node towards a v6 node in a general case (it is possible for a given subset of v6 nodes, but not as a general solution)

A second option is to use IPv6 addresses themselves. In this case, the IPv4 node is aware of the IPv6 address of the destination and it uses it to identify the target at the NAT64 box. This option would likely imply modifications in the v4 nodes.

A third option is to use FQDN to identify nodes. In this case v4 nodes identify v6 nodes using FQDNs, which is already supported in the v4 world. The difficulties with such a approach is that DNS ALG are likely to be required.

A fourth option is to use a combination of IPv4 address, transport protocol and port for identification of a v6 node or a v6 flow.

3.5. Market timing considerations

We expect translation mechanism to require deployment in the very near term, prior to IPv4 address depletion, and to be interoperable with end systems that have been deployed in that timeframe. Since address space depletion is expected t occur in the 2010-2012 timeframe and host software tends to be changed primarily when people buy new hardware (every 2-3 years on average), we expect that this needs to be compatible with currently-deployed Windows (XP and Vista), MacOSX (Tiger and Leopard), Linux, and Solaris operating systems. That argues for a solution that requires no changes to host software that cannot be reasonably expected to deploy via patch update procedures - this is otherwise all solved in network devices.

4. Requirements for new generation of v4-v6 translation mechanisms

This list of requirements basically should contain all the aspects that should be considered when designing a new generation of translation mechanisms.

4.1. Basic Requirements that MUST be supported

These are the requirements for short term mechanism behaviour

R1: Changes in the hosts

The translation mechanism MUST NOT require changes in the v4-only nodes to support the Basic requirements described in this section, unless explicitly stated in the particular requirement. The translation mechanism MAY require changes to v6-only nodes.

R2: Basic communication support

o R2.1: Translation mechanim must support v6-initiated short-lived local handle (as defined in Section 3.1. (strong consensus on this)

o R2.2: Translation mechanim must support v4-initiated short-lived local handle (as defined in Section 3.1). (not clear if there is consensus for this)

- R2.2.1: v4 initiators can either use IPv4 public addresses or IPv4 private addresses and use a NAT.(The acceptance of R2.2.1 is subject to the acceptance of R2.2.

R3: Interaction with dual-stack hosts

Translation mechanism MUST allow using native connectivity when it is available. This means that if a v6-only nodes wants to communicate with a dual stack, it must use native v6 connectivity and if a v4-only nodes wants to communicate with a dual stack, it must use native v4 connectivity.(In this case, dual stack means a host with both IPv6 and IPv4 stacks, wich are both active, i.e. they have v4 and v6 connectivity).

R4: DNS semantics preservation

Any modifications to DNS responses associated with translation MUST NOT violate standard DNS semantics. This includes in particular that a DNS response (that has been modified by the translator mechanism) should not be invalid if it ends up in the wrong context, i.e. traversing a non expected part of the topology.

R5: Routing

IPv6 routing should not be affected in any way, and there should be no risk of importing "entropy" from the IPv4 routing tables into IPv6.

R6: Protocols supported

The translation mechanism MUST support at least TCP, UDP, ICMP, TLS.

R7: Behave requirements

The translation mechanism MUST be compliant with the requirements for IPv4 NATs defined in [I-D.ietf-behave-tcp] and in [RFC4787] when applicable. These requirements should be interpreted with the IPv6 side on the IPv6-IPv4 translator being the IPv4 private side of the conventional NAT.

R8: Fragmented packets

The translation mechanism MUST suport fragmented packets when the fragments arrive within an interval smaller or equal to 5 seconds. However, the translator device MUST avoid that the support for fragmented packets introduces a DoS attack vector (i.e. an attacker injecting a high number of fragments would result in a DoS attack to the device), so the device MUST implement some form of limitation to

the resources used by the fragmented packet support. for example a translator device may define a maximum amount of memory used for storing fragmented packet state (the actual amount of memory will depend on the intended usage of the box, carrier grade vs. set top box).

R9: Security

The adoption of the translation mechanism MUST not result in a significantly more vulnerable Internet

R10: DNSSec support

DNSSec support MUST NOT be prevented.

- R10.1: In particular, if an IPv6 node is initiating a communication with an IPv4 that is located behind a translator, the IPv6 initiator MUST be able to perform DNSSec verification of the DNS information of the IPv4 target. (strong consensus on this one).
- R10.2: In particular, if an IPv4 node is initiating a communication with an IPv6 that is located behind a translator, the IPv4 initiator MUST be able to perform DNSSec verification of the DNS information of the IPv4 target. This may require the modification of the IPv4 node as well. (not clear if there consensus on this one)

R11: IPsec support.

The translator MUST support communication between IPv4 node and IPv6 node using UDP Encapsulation of IPsec ESP Packets as defined in [RFC3948] as applicable. RFC3948 should be interpreted as with the IPv6 side on the IPv6-IPv4 translator being the IPv4 private side of the conventional NAT. IPsec support MAY require updating also the IPv4 side.

4.2. Important things that SHOULD be supported

I2: Operational flxibility

It should be possible to locate the translation device at an arbitrary point in the network (i.e. not at fixed points such as a site exit), so that there is full operational flexibility.

I3: Central Management

Any configuration need for an IPv6 host to make use of the mechanism should be possible centrally, e.g. a DHCP option.

I4: Richer application behaviour support

The translation mechanism SHOULD support the other types of application behaviours, including Long-lived application associations, callbacks and referrals.In order to support this. the translation mechanism MAY require changes to v4-only nodes too

I5: MIPv6 support

The translation mechanism SHOULD not prevent MIPv6 Route Optimization when the CN is a v4-only node

I6: SCTP support

The translation mechanism SHOULD not prevent a SCTP communication between a v6-only node and a v4-only node

I7: DCCP support

The translation mechanism SHOULD not prevent a DCCP communication between a v6-only node and a v4-only node

I8: Multicast support

The translation mechanism SHOULD not prevent multicast traffic between the v4-only nodes and the v6-only nodes.

5. Contributors

This draft contains contributions from Iljitsch van Beijnum, Brian Carpenter and Elwyn Davies (this doesn't mean that they agree on the draft, just that we have used text provided by them). We would like to acknowledge the comments from Dave Thaler, Michael Richardson, George Tsirtsis, Hesham Soliman, Yaron Sheffer and Kurt Lindqvist.

6. Security considerations

The requirements include R9 and R11 concerning security issues.

7. Acknowledgments

Marcelo Bagnulo is partly funded by Trilogy, a research project supported by the European Commission under its Seventh Framework Program.

8. References

8.1. Normative References

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

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, September 2007.

[I-D.ietf-behave-tcp]
Guha, S., "NAT Behavioral Requirements for TCP", draft-ietf-behave-tcp-07 (work in progress), April 2007.

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

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

8.2. Informative References

[RFC3439] Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439, December 2002.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC4214] Templin, F., Gleeson, T., Talwar, M., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 4214, October 2005.

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

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

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

[I-D.stenberg-v6ops-pd-route-maintenance]

Stenberg, M. and O. Troan, "IPv6 Prefix Delegation routing state maintenance approaches", draft-stenberg-v6ops-pd-route-maintenance-00 (work in progress), December 2007.

Authors' Addresses

Marcelo Bagnulo
Huawei Labs at Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

Phone: +1-408-526-4257
Fax: +1-413-473-2403
Email: fred@cisco.com

Iljitsch van Beijnum
IMDEA Networks
Madrid, Madrid 28911
Spain

Phone:
Fax:
Email: iljitsch@muada.com

Full Copyright Statement

Intellectual Property