Internet Engineering Task Force (IETF)
Request for Comments: 6144
Category: Informational
ISSN: 2070-1721

F. Baker
Cisco Systems
X. Li
C. Bao
CERNET Center/Tsinghua University
K. Yin
Cisco Systems

April 2011

Framework for IPv4/IPv6 Translation

Abstract

This note describes a framework for IPv4/IPv6 translation. This is in the context of replacing Network Address Translation - Protocol Translation (NAT-PT), which was deprecated by RFC 4966, and to enable networks to have IPv4 and IPv6 coexist in a somewhat rational manner while transitioning to an IPv6 network.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6144.

Table of Contents

1. Introduction

This note describes a framework for IPv4/IPv6 translation. This is in the context of replacing NAT-PT (Network Address Translation - Protocol Translation) [RFC2766], which was deprecated by [RFC4966], and to enable networks to have IPv4 and IPv6 coexist in a somewhat rational manner while transitioning to an IPv6-only network.

NAT-PT was deprecated to inform the community that NAT-PT had operational issues and was not considered a viable medium- or long-term strategy for either coexistence or transition. It wasn't intended to say that IPv4<->IPv6 translation was bad, but the way that NAT-PT did it was bad, and in particular using NAT-PT as a general-purpose solution was bad. As with the deprecation of the RIP routing protocol [RFC1923] at the time the Internet was converting to Classless Inter-Domain Routing (CIDR), the point was to encourage network operators to actually move away from technology with known issues.

[RFC4213] describes the IETF's view of the most sensible transition model. The IETF recommends, in short, that network operators (transit providers, service providers, enterprise networks, small and medium businesses, SOHO (Small Office, Home Office) and residential customers, and any other kind of network that may currently be using IPv4) obtain an IPv6 prefix, turn on IPv6 routing within their networks and between themselves and any peer, upstream, or downstream neighbors, enable it on their computers, and use it in normal processing. This should be done while leaving IPv4 stable, until a point is reached that any communication that can be carried out could use either protocol equally well. At that point, the economic justification for running both becomes debatable, and network operators can justifiably turn IPv4 off. This process is comparable to that of [RFC4192], which describes how to renumber a network using the same address family without a flag day. While running stably with the older system, deploy the new. Use the coexistence period to work out such kinks as they arise. When the new is also running stably, shift production to it. When network and economic conditions warrant, remove the old, which is now no longer necessary.

The question arises: what if that is infeasible due to the time available to deploy or other considerations? What if the process of moving a network and its components or customers is starting too late for contract cycles to effect IPv6 turn-up on important parts at a point where it becomes uneconomical to deploy global IPv4 addresses in new services? How does one continue to deploy new services without balkanizing the network?

This document describes translation as one of the tools networks might use to facilitate coexistence and ultimate transition.

1.1. Why Translation?

Besides dual-stack deployment, there are two fundamental approaches one could take to interworking between IPv4 and IPv6: tunneling and translation. One could -- and in the [6NET] we did -- build an overlay network that tunnels one protocol over the other. Various proposals take that model, including 6to4 [RFC3056], Teredo [RFC4380], Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [RFC5214], and Dual-Stack Lite [DS-LITE]. The advantage of doing so is that the new protocol is enabled to work without disturbing the old protocol, providing connectivity between users of the new protocol. There are two disadvantages to tunneling:

- Users of the new architecture are unable to use the services of the underlying infrastructure -- it is just bandwidth, and
- It doesn’t enable new protocol users to communicate with old protocol users without dual-stack hosts.

As noted, in this work, we look at Internet Protocol translation as a transition strategy. [RFC4864] forcefully makes the point that people use Network Address Translators to meet various needs, many of which are met as well by routing or protocol mechanisms that preserve the end-to-end addressability of the Internet. What it did not consider is the case in which there is an ongoing requirement to communicate with IPv4 systems, but, for example, configuring IPv4 routing is not the most desirable strategy in the network operator’s view, or is infeasible due to a shortage of global address space. Translation enables the client of a network, whether a transit network, an access network, or an edge network, to access the services of the network and communicate with other network users regardless of their protocol usage -- within limits. Like NAT-PT, IPv4/IPv6 translation under this rubric is not a long-term support strategy, but it is a medium-term coexistence strategy that can be used to facilitate a long-term program of transition.

1.2. Terminology

The following terminology is used in this document and other documents related to it.

An IPv4 network: A specific network that has an IPv4-only deployment. This could be an enterprise’s IPv4-only network, an ISP’s IPv4-only network, or even an IPv4-only host. The IPv4 Internet is the set of all interconnected IPv4 networks.

An IPv6 network: A specific network that has an IPv6-only deployment. This could be an enterprise's IPv6-only network, an ISP's IPv6-only network, or even an IPv6-only host. The IPv6 Internet is the set of all interconnected IPv6 networks.

DNS46: A DNS translator that translates AAAA record to A record.

DNS64: A DNS translator that translates A record to AAAA record.

Dual-Stack implementation: A dual-stack implementation, in this context, comprises an IPv4/IPv6-enabled end system stack, applications plus routing in the network. It implies that two application instances are capable of communicating using either IPv4 or IPv6 -- they have stacks, they have addresses, and they have any necessary network support including routing.

IPv4-converted addresses: IPv6 addresses used to represent IPv4 nodes in an IPv6 network. They have an explicit mapping relationship to IPv4 addresses. Both stateless and stateful translators use IPv4-converted addresses to represent IPv4 addresses.

IPv4-only: An IPv4-only implementation, in this context, comprises an IPv4-enabled end system stack, applications directly or indirectly using that IPv4 stack, plus routing in the network. It implies that two application instances are capable of communicating using IPv4, but not IPv6 -- they have an IPv4 stack, addresses, and network support including IPv4 routing and potentially IPv4/IPv4 translation, but some element is missing that prevents communication with IPv6 hosts.

IPv4-translatable addresses: IPv6 addresses to be assigned to IPv6 nodes for use with stateless translation. They have an explicit mapping relationship to IPv4 addresses. A stateless translator uses the corresponding IPv4 addresses to represent the IPv6 addresses. A stateful translator does not use this kind of addresses, since IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic state.

IPv6-only: An IPv6-only implementation, in this context, comprises an IPv6-enabled end system stack, applications directly or indirectly using that IPv6 stack, plus routing in the network. It implies that two application instances are capable of communicating using IPv6, but not IPv4 -- they have an IPv6 stack, addresses, and network support including routing in IPv6, but some element is missing that prevents communication with IPv4 hosts.

Network-Specific Prefix (NSP): From an IPv6 prefix assigned to a network operator, the operator chooses a longer prefix for use by the operator's translator(s). Hence, a given IPv4 address would have different IPv6 representations in different networks that use different network-specific prefixes. A network-specific prefix is also known as a Local Internet Registry (LIR) prefix.

State: "State" refers to dynamic information that is stored in a network element. For example, if two systems are communicating using a TCP connection, each stores information about the connection, which is called "connection state". In this context, the term refers to dynamic correlations between IP addresses on either side of a translator, or {IP address, transport protocol, transport port number} tuples on either side of the translator. Of stateful algorithms, there are at least two major flavors depending on the kind of state they maintain:

Hidden state: the existence of this state is unknown outside the network element that contains it.

Known state: the existence of this state is known by other network elements.

Stateful Translation: A translation algorithm may be said to "require state in a network element" or be "stateful" if the transmission or reception of a packet creates or modifies a data structure in the relevant network element.

Stateful Translator: A translator that uses stateful translation for either the source or destination address. A stateful translator typically also uses a stateless translation algorithm for the other type of address.

Stateless Translation: A translation algorithm that is not "stateful" is "stateless". It derives its needed information algorithmically from the messages it is translating and from pre-configured information.

Stateless Translator: A translator that uses only stateless translation for both destination address and source address.

Well-Known Prefix (WKP): The IPv6 prefix defined in [RFC6052] for use in an algorithmic mapping.

1.3. Translation Objectives

In any translation model, there is a question of objectives. Ideally, one would like to make any system and any application running on it able to "talk with" -- exchange datagrams supporting applications with -- any other system running the same application regardless of whether they have an IPv4 stack and connectivity or IPv6 stack and connectivity. That was the model for NAT-PT, and the things it necessitated led to scaling and operational difficulties [RFC4966].

So the question comes back to what different kinds of connectivity can be easily supported, what kinds are harder, and what technologies are needed to at least pick the low-hanging fruit. We observe that applications today fall into two main categories:

Client/Server Application: Per whatis.com, "'Client/server' describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request." In networking, the behavior of the applications is that connections are initiated from client software and systems to server software and systems. Examples include mail handling between an end user and his mail system (POP3, IMAP, and MUA->MTA SMTP), FTP, the web, and DNS name resolution.

Peer-to-Peer (P2P) Application: A P2P application is an application that uses the same endpoint to initiate outgoing sessions to peering hosts as well as accept incoming sessions from peering hosts. These in turn fall broadly into two categories:

Peer-to-peer infrastructure applications: Examples of "infrastructure applications" include SMTP between MTAs, Network News, and SIP. Any MTA might open an SMTP session with any other at any time; any SIP Proxy might similarly connect with any other SIP Proxy. An important characteristic of these applications is that they use ephemeral sessions -- they open sessions when they are needed and close them when they are done.

Peer-to-peer file exchange applications: Examples of these include Limewire, BitTorrent, and UTorrent. These are applications that open some sessions between systems and leave them open for long periods of time, and where ephemeral sessions are important, these applications are able to learn about the reliability of peers from history or by reputation. They use the long-term sessions to map content availability.

Short-term sessions are used to exchange content. They tend to prefer to ask for content from peers that they find reliable and available.

If the goal is the ability to open connections between systems, then one must ask who opens connections.

- We need a technology that will enable systems that act as clients to be able to open sessions with other systems that act as servers, whether in the IPv6->IPv4 direction or the IPv4->IPv6 direction. Ideally, this is stateless; especially in a carrier infrastructure, the preponderance of accesses will be to servers, and this optimizes access to them. However, a stateful algorithm is acceptable if the complexity is minimized and a stateless algorithm cannot be constructed.

- We also need a technology that will allow peers to connect with each other, whether in the IPv6->IPv4 direction or the IPv4->IPv6 direction. Again, it would be ideal if this was stateless, but a stateful algorithm is acceptable if the complexity is minimized and a stateless algorithm cannot be constructed.

- In some situations, hosts are purely clients. In those situations, we do not need an algorithm to enable connections to those hosts.

The complexity arguments bring us in the direction of hidden state: if state must be shared between the application and the translator or between translation components, complexity and deployment issues are greatly magnified. The objective of the translators is to avoid, as much as possible, any software changes in hosts or applications necessary to support translation.

NAT-PT is an example of a facility with known state -- at least two software components (the data-plane translator and the DNS Application Layer Gateway, which may be implemented in the same or different systems) share and must coordinate translation state. A typical IPv4/IPv4 NAT implements an algorithm with hidden state. Obviously, stateless translation requires less computational overhead than stateful translation, and less memory to maintain the state, because the translation tables and their associated methods and processes exist in a stateful algorithm and don’t exist in a stateless one.

1.4. Transition Plan

While the design of IPv4 made it impossible for IPv6 to be compatible on the wire, the designers intended that it would coexist with IPv4 during a period of transition. The primary mode of coexistence was dual-stack operation -- routers would be dual-stacked so that the network could carry both address families, and IPv6-capable hosts could be dual-stack to maintain access to IPv4-only partners. The goal was that the preponderance of hosts and routers in the Internet would be IPv6-capable long before IPv4 address space allocation was completed. At this time, it appears the exhaustion of IPv4 address space will occur before significant IPv6 adoption.

Curran's "A Transition Plan" [RFC5211] proposes a three-phase progression:

Preparation Phase (current): characterized by pilot use of IPv6, primarily through transition mechanisms defined in [RFC4213], and planning activities.

Transition Phase (2010 through 2011): characterized by general availability of IPv6 in provider networks, which should be native IPv6; organizations should provide IPv6 connectivity for their Internet-facing servers, but should still provide IPv4-based services via a separate service name.

Post-Transition Phase (2012 and beyond): characterized by a preponderance of IPv6-based services and diminishing support for IPv4-based services.

Various timelines have been discussed, but most will agree with the pattern of the above three transition phases, also known as an "S" curve transition pattern.

In each of these phases, the coexistence problem and solution space have a different focus:

Preparation Phase: Coexistence tools are needed to facilitate early adopters by removing impediments to IPv6 deployment and to assure that nothing is lost by adopting IPv6 -- in particular, that the IPv6 adopter has unfettered access to the global IPv4 Internet regardless of whether they have a global IPv4 address (or any IPv4 address or stack at all). While it might appear reasonable for the cost and operational burden to be borne by the early adopter, the shared goal of promoting IPv6 adoption would argue against that model. Additionally, current IPv4 users should not be forced

to retire or upgrade their equipment, and the burden remains on service providers to carry and route native IPv4. This is known as the early stage of the "S" curve.

Transition Phase: During the middle stage of the "S" curve, while IPv6 adoption can be expected to accelerate, there will still be a significant portion of the Internet operating IPv4-only or preferring IPv4. During this phase, the norm shifts from IPv4 to IPv6, and coexistence tools evolve to ensure interoperability between domains that may be restricted to IPv4 or IPv6.

Post-Transition Phase: This is the last stage of the "S" curve. In this phase, IPv6 is ubiquitous and the burden of maintaining interoperability shifts to those who choose to maintain IPv4-only systems. While these systems should be allowed to live out their economic life cycles, the IPv4-only legacy users at the edges should bear the cost of coexistence tools, and at some point service provider networks should not be expected to carry and route native IPv4 traffic.

The choice between the terms "transition" versus "coexistence" has engendered long philosophical debate. "Transition" carries the sense that one is going somewhere, while "coexistence" seems more like one is sitting somewhere. Historically with the IETF, "transition" has been the term of choice [RFC4213] [RFC5211], and the tools for interoperability have been called "transition mechanisms". There is some perception or conventional wisdom that adoption of IPv6 is being impeded by the deficiency of tools to facilitate interoperability of nodes or networks that are constrained (in some way, fully or partially) from full operation in one of the address families. In addition, it is apparent that transition will involve a period of coexistence; the only real question is how long that will last.

Thus, coexistence is an integral part of the transition plan, not in conflict with it, but there will be a balancing act. It starts out being a way for early IPv6 adopters to easily exploit the bigger IPv4 Internet, and ends up being a way for late/never adopters to hang on with IPv4 (at their own expense, with minimal impact or visibility to the Internet). One way to look at solutions is that cost incentives (both monetary cost and the operational overhead for the end user) should encourage IPv6 and discourage IPv4. That way natural market forces will keep the transition moving -- especially as the legacy IPv4-only stuff ages out of use. The end goal should not be to eliminate IPv4 by fiat, but rather render it redundant through ubiquitous IPv6 deployment. IPv4 may never go away completely, but rational plans should move the costs of maintaining IPv4 to those who insist on using it after wide adoption of IPv6.

2. Scenarios for IPv4/IPv6 Translation

It is important to note that the choice of translation solution and the assumptions about the network where they are used impact the consequences. A translator for the general case has a number of issues that a translator for a more specific situation may not have at all.

The intention of this document is to focus on translation solutions under all kinds of situations. All IPv4/IPv6 translation cases can be easily described in terms of "interoperation between a set of systems (applications) that only communicate using IPv4 and a set of systems that only communicate using IPv6", but the differences at a detailed level make them interesting.

Based on the transition plan described in Section 1.4, there are four types of IPv4/IPv6 translation cases:

a. Interoperation between an IPv6 network and the IPv4 Internet

b. Interoperation between an IPv4 network and the IPv6 Internet

c. Interoperation between an IPv6 network and an IPv4 network

d. Interoperation between the IPv6 Internet and the IPv4 Internet

Each one of the above can be divided into two scenarios, depending on whether the IPv6 side or the IPv4 side initiates communication, so there are a total of eight scenarios.

Scenario 1: an IPv6 network to the IPv4 Internet

Scenario 2: the IPv4 Internet to an IPv6 network

Scenario 3: the IPv6 Internet to an IPv4 network

Scenario 4: an IPv4 network to the IPv6 Internet

Scenario 5: an IPv6 network to an IPv4 network

Scenario 6: an IPv4 network to an IPv6 network

Scenario 7: the IPv6 Internet to the IPv4 Internet

Scenario 8: the IPv4 Internet to the IPv6 Internet

We will discuss each scenario in detail in the next section.

2.1. Scenario 1: An IPv6 Network to the IPv4 Internet

Due to the lack of IPv4 addresses or due to other technical or economical constraints, the network is IPv6-only, but the hosts in the network require communicating with the global IPv4 Internet.

This is the typical scenario for what we sometimes call "green-field" deployments. One example is an enterprise network that wishes to operate only IPv6 for operational simplicity, but still wishes to reach the content in the IPv4 Internet. The green-field enterprise scenario is different from an ISP's network in the sense that there is only one place that the enterprise can easily modify: the border between its network and the IPv4 Internet. Obviously, the IPv4 Internet operates the way it already does. But, in addition, the hosts in the enterprise network are commercially available devices, personal computers with existing operating systems. This restriction drives us to a "one box" type of solution, where IPv6 can be translated into IPv4 to reach the public Internet.

Other cases that have been mentioned include wireless ISP networks and sensor networks. These bear a striking resemblance to this scenario as well, if one considers the ISP network to simply be a very special kind of enterprise network.

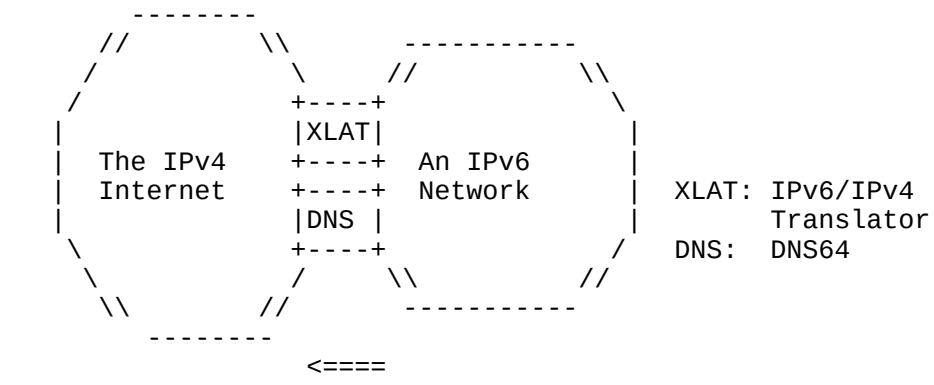

Figure 1: Scenario 1

Both stateless and stateful solutions can support Scenario 1.

2.2. Scenario 2: The IPv4 Internet to an IPv6 Network

When the enterprise networks or ISP networks adopt Scenario 1, the IPv6-only users will not only want to access servers on the IPv4 Internet but also will want to setup their own servers in the network that are accessible by the users on the IPv4 Internet, since the majority of the Internet users are still in the IPv4 Internet. Thus, with a translation solution for this scenario, the benefits would be clear. Not only could servers move directly to IPv6 without trudging through a difficult transition period, but they could do so without risk of losing connectivity with the IPv4-only Internet.

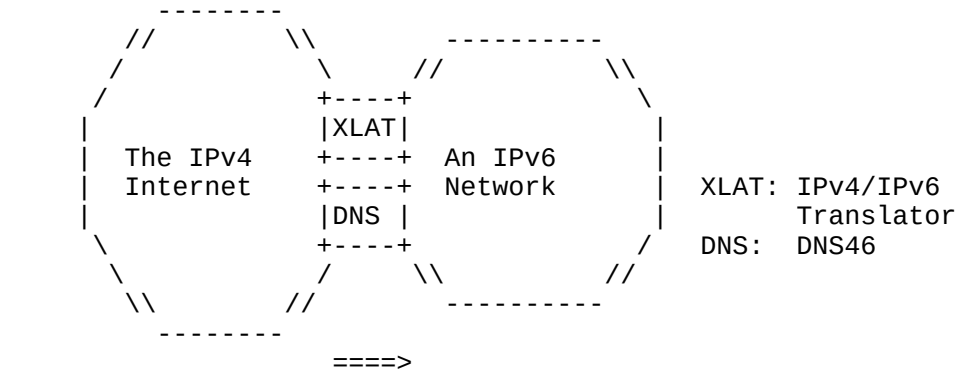

Figure 2: Scenario 2

In general, this scenario presents a hard case for translation. Stateful translation such as NAT-PT [RFC2766] can be used in this scenario, but it requires a tightly coupled DNS Application Level Gateway (ALG) in the translator, and this technique was deprecated by the IETF [RFC4966].

The stateless translation solution in Scenario 1 can also work in Scenario 2, since it can support IPv4-initiated communications with a subset of the IPv6 addresses (IPv4-translatable addresses) in an IPv6 network.

2.3. Scenario 3: The IPv6 Internet to an IPv4 Network

There is a requirement for a legacy IPv4 network to provide services to IPv6 hosts.

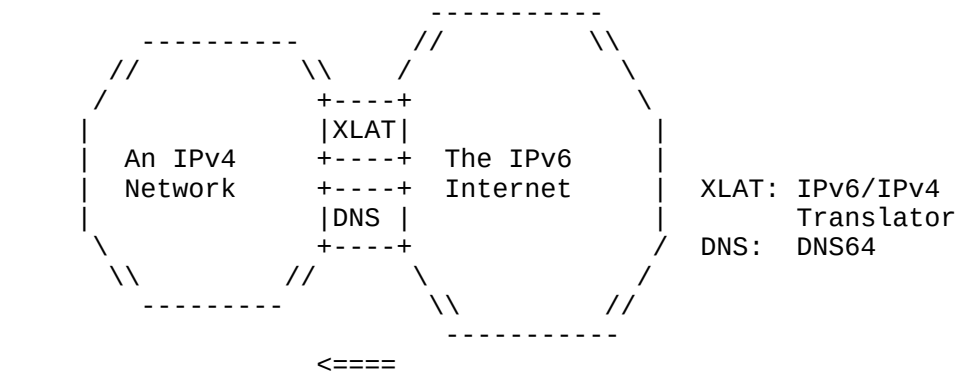

Figure 3: Scenario 3

Stateless translation will not work for this scenario, because an IPv4 network needs to communicate with all of the IPv6 Internet, not just a small subset, and stateless can only support a subset of the IPv6 addresses. However, IPv6-initiated communication can be achieved through stateful translation. In this case, a Network Specific Prefix assigned to the translator will give the hosts unique IPv4-converted IPv6 addresses, no matter whether the legacy IPv4 network uses public IPv4 addresses or [RFC1918] addresses. Also there is no need to synthesize AAAA from A records, since static AAAA records can be put in the regular DNS to represent these IPv4-only hosts as discussed in Section 7.3 of [RFC6147].

2.4. Scenario 4: An IPv4 Network to the IPv6 Internet

Due to technical or economical constraints, the network is IPv4-only, and IPv4-only hosts (applications) may require communicating with the global IPv6 Internet.

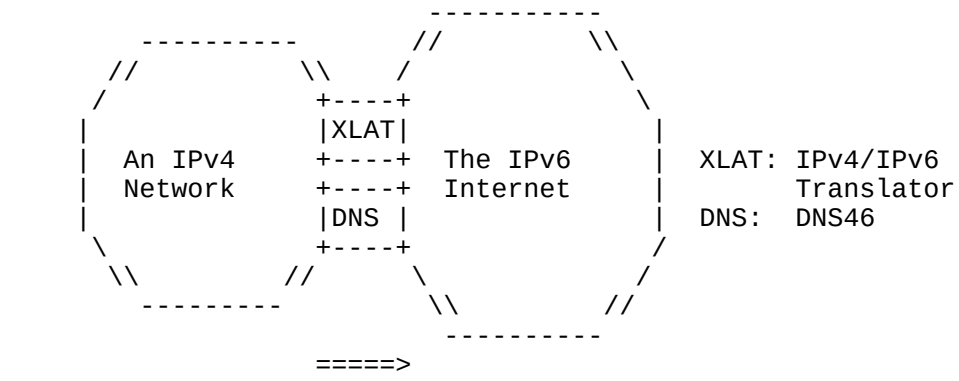

Figure 4: Scenario 4

In general, this scenario presents a hard case for translation. Stateful translation such as NAT-PT [RFC2766] can be used in this scenario, but it requires a tightly coupled DNS ALG in the translator, and this technique was deprecated by the IETF [RFC4966].

From the transition phase discussion in Section 1.4, this scenario will probably only occur when we are well past the early stage of the "S" curve, and the IPv4/IPv6 transition has already moved to the right direction. Therefore, in-network translation is not considered viable for this scenario, and other techniques should be considered.

2.5. Scenario 5: An IPv6 Network to an IPv4 Network

In this scenario, both an IPv4 network and an IPv6 network are within the same organization.

The IPv4 addresses used are either public IPv4 addresses or [RFC1918] addresses. The IPv6 addresses used are either public IPv6 addresses or ULAs (Unique Local Addresses) [RFC4193].

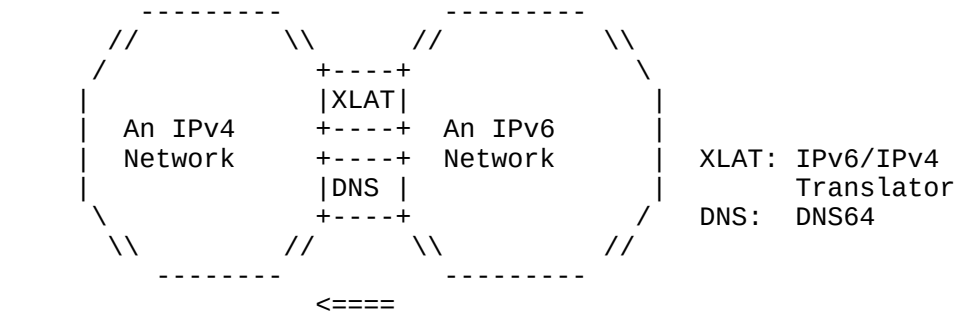

Figure 5: Scenario 5

The translation requirement from this scenario has no significant difference from Scenario 1, so both the stateful and stateless translation schemes discussed in Section 2.1 apply here.

2.6. Scenario 6: An IPv4 Network to an IPv6 Network

This is another scenario when both an IPv4 network and an IPv6 network are within the same organization.

The IPv4 addresses used are either public IPv4 addresses or [RFC1918] addresses. The IPv6 addresses used are either public IPv6 addresses or ULAs (Unique Local Addresses) [RFC4193].

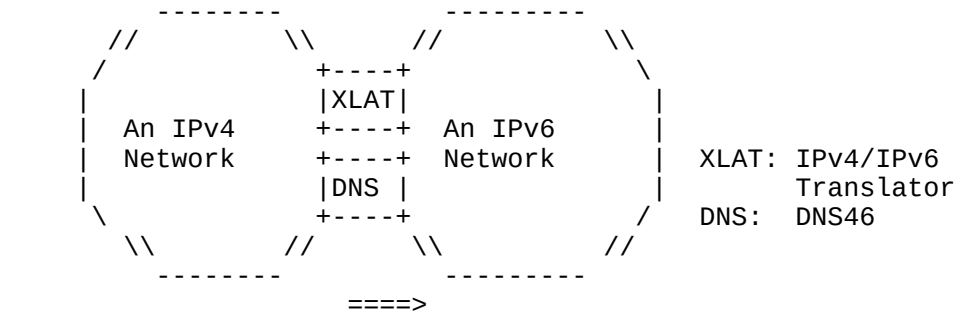

Figure 6: Scenario 6

The translation requirement from this scenario has no significant difference from Scenario 2, so the translation scheme discussed in Section 2.2 applies here.

2.7. Scenario 7: The IPv6 Internet to the IPv4 Internet

This seems the ideal case for in-network translation technology, where any IPv6-only host or application on the global Internet can initiate communication with any IPv4-only host or application on the global Internet.

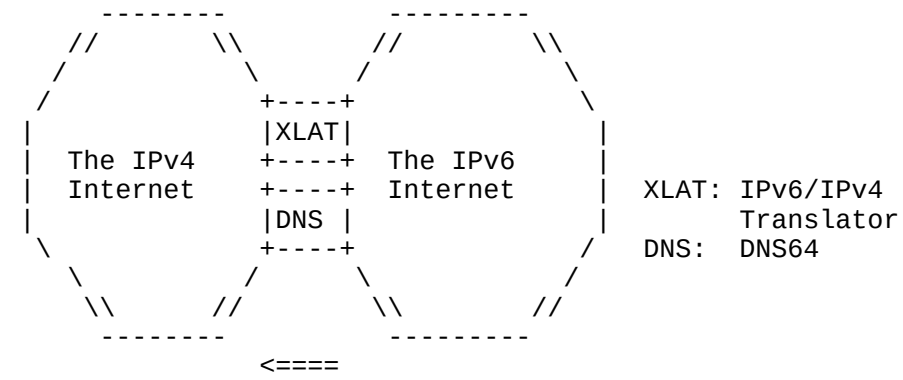

Figure 7: Scenario 7

Due to the huge difference in size between the address spaces of the IPv4 Internet and the IPv6 Internet, there is no viable translation technique to handle unlimited IPv6 address translation.

If we ever run into this scenario, fortunately, the IPv4/IPv6 transition has already passed the early stage of the "S" curve. Therefore, there is no obvious business reason to demand a translation solution as the only transition strategy.

2.8. Scenario 8: The IPv4 Internet to the IPv6 Internet

This case is very similar to Scenario 7. The analysis and conclusions for Scenario 7 also apply for this scenario.

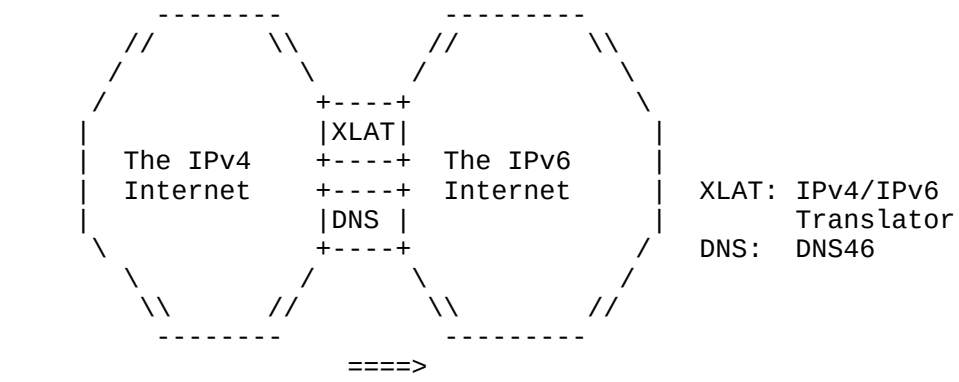

Figure 8: Scenario 8

3. Framework

Having laid out the preferred transition model and the options for implementing it (Section 1.1), defined terms (Section 1.2), considered the requirements (Section 1.3), considered the transition model (Section 1.4), and considered the kinds of scenarios the facility would support (Section 2), we now turn to a framework for IPv4/IPv6 translation. The framework contains the following components:

- Address translation
- IP and ICMP translation
- Maintaining translation state
- DNS64 and DNS46
- ALGs for other application-layer protocols (e.g., FTP)

3.1. Translation Components

3.1.1. Address Translation

When IPv6/IPv4 translation is performed, we should specify how an individual IPv6 address is translated to a corresponding IPv4 address, and vice versa, in cases where an algorithmic mapping is

used. This includes the choice of IPv6 prefix and the choice of method by which the remainder of the IPv6 address is derived from an IPv4 address [RFC6052]. The usages of the IPv6 addresses are shown in the following figures.

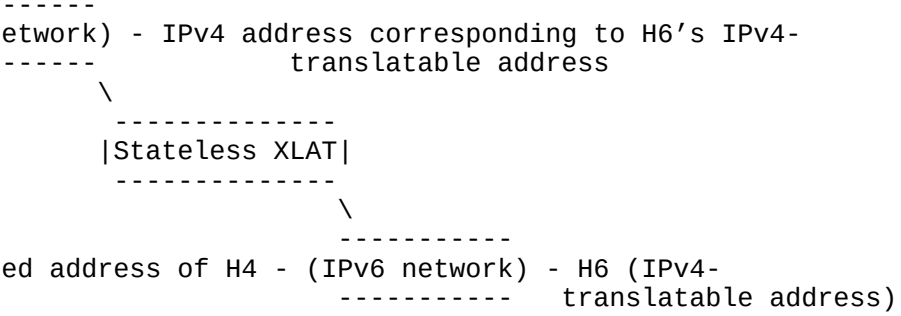

Figure 9: IPv6 Address Representation for Stateless Translation

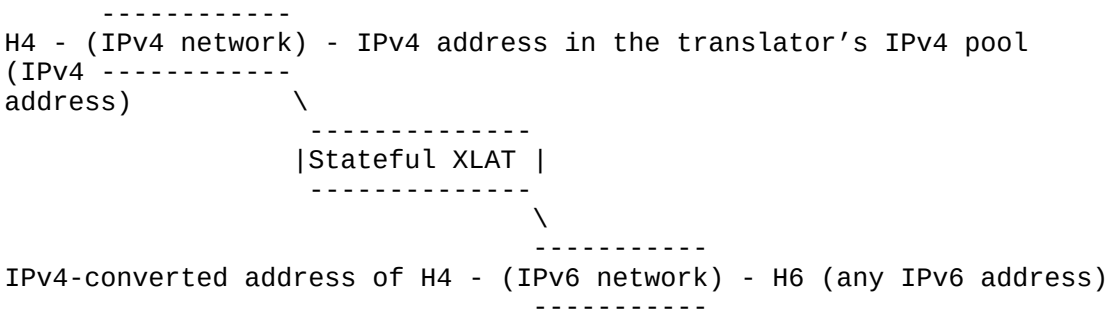

Figure 10: IPv6 Address Representation for Stateful Translation

For both stateless and stateful translation, an algorithmic mapping table is used to translate IPv6 destination addresses (IPv4-converted addresses) to IPv4 destination addresses in the IPv6-to-IPv4 direction and translate IPv4 source addresses to IPv6 source addresses (IPv4-converted addresses) in the IPv4-to-IPv6 direction. Note that translating IPv6 source addresses to IPv4 source addresses in the IPv6-to-IPv4 direction and translating IPv4 destination addresses to IPv6 destination addresses in the IPv4-to-IPv6 direction will be different for stateless translation and stateful translation.

o For stateless translation, the same algorithmic mapping table is used to translate IPv6 source addresses (IPv4-translatable addresses) to IPv4 source addresses in the IPv6-to-IPv4 direction and translate IPv4 destination addresses to IPv6 destination

addresses (IPv4-translatable addresses) in the IPv4-to-IPv6 direction. In this case, blocks of the service provider's IPv4 addresses are mapped into IPv6 and used by physical IPv6 nodes. The original IPv4 form of these blocks of the service provider's IPv4 addresses are used to represent the physical IPv6 nodes in IPv4. Note that stateless translation supports both IPv6 initiated as well as IPv4 initiated communications.

- o For stateful translation, the algorithmic mapping table is not used to translate source addresses in the IPv6-to-IPv4 direction and destination addresses in the IPv4-to-IPv6 direction. Instead, a state table is used to translate IPv6 source addresses to IPv4 source addresses in the IPv6-to-IPv4 direction and translate IPv4 destination addresses to IPv6 destination addresses in the IPv4-to-IPv6 direction. In this case, blocks of the service provider's IPv4 addresses are maintained in the translator as the IPv4 address pools and are dynamically bound to specific IPv6 addresses. The original IPv4 form of these blocks of the service provider's IPv4 addresses is used to represent the IPv6 address in IPv4. However, due to the dynamic binding, stateful translation in general only supports IPv6-initiated communication.

3.1.2. IP and ICMP Translation

The IPv4/IPv6 translator is based on the update to the Stateless IP/ICMP Translation Algorithm (SIIT) described in [RFC2765]. The algorithm translates between IPv4 and IPv6 packet headers (including ICMP headers).

The IP and ICMP translation document [RFC6145] discusses header translation for both stateless and stateful modes, but does not cover maintaining state in the stateful mode. In the stateless mode, translation is performed using a combination of information carried in the address and information configured in the translator. This permits both IPv4->IPv6 and IPv6->IPv4 session establishment. In the stateful mode, translation state is maintained between IPv4 address/transport port tuples and IPv6 address/transport port tuples, enabling IPv6 systems to open sessions with IPv4 systems. The choice of operational mode is made by the operator deploying the network and is critical to the operation of the applications using it.

3.1.3. Maintaining Translation State

For the stateful translator, besides IP and ICMP translation, special action must be taken to maintain the translation states. [RFC6146] describes a mechanism for maintaining state.

3.1.4. DNS64 and DNS46

DNS64 [RFC6147] and possible future DNS46 documents describe the mechanisms by which a DNS translator is intended to operate. It is designed to operate on the basis of known address translation algorithms defined in [RFC6052].

There are at least two possible implementations of a DNS64 and DNS46:

Static records: One could literally populate DNS with corresponding A and AAAA records. This mechanism works for scenarios 2, 3, 5, and 6.

Dynamic Translation of static records: In more general operation, the preferred behavior is an A record to be (retrieved and) translated to a AAAA record by the DNS64 if and only if no reachable AAAA record exists, or for a AAAA record to be (retrieved and) translated to an A record by the DNS46 if and only if no reachable A record exists.

3.1.5. ALGs for Other Applications Layer Protocols

In addition, some applications require special support. An example is FTP. FTP's active mode doesn't work well across NATs without extra support such as SOCKS [RFC1928] [RFC3089]. Across NATs, it generally uses passive mode. However, the designers of FTP wrote different and incompatible passive-mode implementations for IPv4 and IPv6 networks. Hence, either they need to fix FTP, or a translator must be written for the application. Other applications may be similarly broken.

As a general rule, a simple operational recommendation will work around many application issues: there should be a server in each domain, or an instance of the server should have an interface in each domain. For example, an SMTP MTA may be confused by finding an IPv6 address in its HELO when it is connected to using IPv4 (or vice versa), but would work perfectly well if it had an interface in both the IPv4 and IPv6 domains and was used as an application-layer bridge between them.

3.2. Operation Mode for Specific Scenarios

Currently, the proposed solutions for IPv6/IPv4 translation are classified into stateless translation and stateful translation.

3.2.1. Stateless Translation

For stateless translation, the translation information is carried in the address itself plus configuration in the translators, permitting both IPv4->IPv6 and IPv6->IPv4 session initiation. Stateless translation supports end-to-end address transparency and has better scalability compared with stateful translation [RFC6145].

The stateless translation mechanisms typically put constraints on what IPv6 addresses can be assigned to IPv6 nodes that want to communicate with IPv4 destinations using an algorithmic mapping. For Scenario 1 ("an IPv6 network to the IPv4 Internet"), it is not a serious drawback, since the address assignment policy can be applied to satisfy this requirement for the IPv6 nodes that need to communicate with the IPv4 Internet. In addition, stateless translation supports Scenario 2 ("the IPv4 Internet to an IPv6 network"), which means that not only could servers move directly to IPv6 without trudging through a difficult transition period, but also they could do so without risk of losing connectivity with the IPv4-only Internet.

Stateless translation can be used for Scenarios 1, 2, 5, and 6, i.e., it supports "an IPv6 network to the IPv4 Internet", "the IPv4 Internet to an IPv6 network", "an IPv6 network to an IPv4 network", and "an IPv4 network to an IPv6 network".

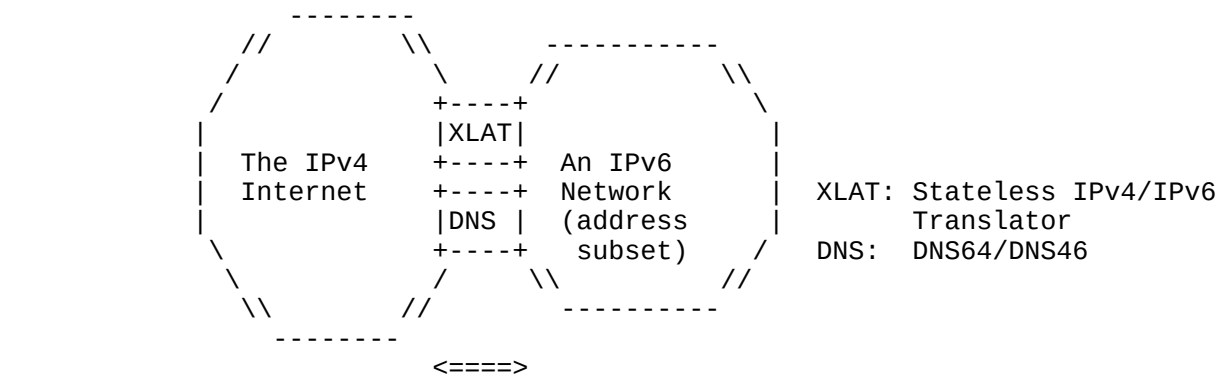

Figure 11: Stateless Translation for Scenarios 1 and 2

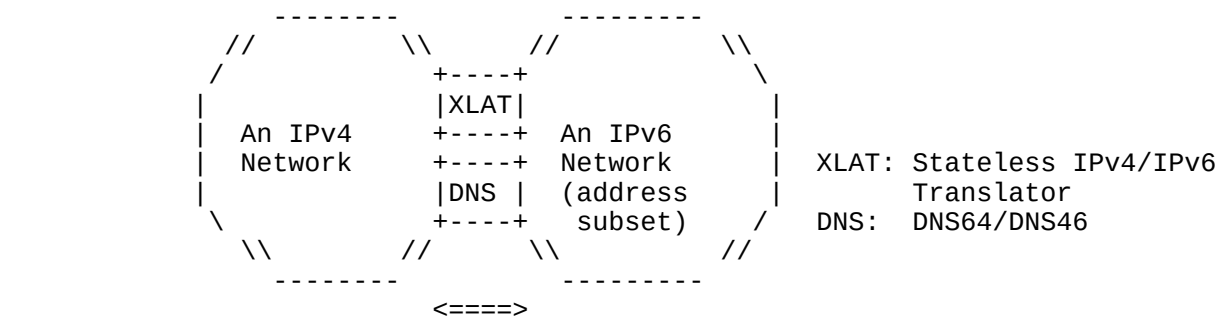

Figure 12: Stateless Translation for Scenarios 5 and 6

The implementation of the stateless translator needs to refer to [RFC6145] and [RFC6052].

3.2.2. Stateful Translation

For stateful translation, the translation state is maintained between IPv4 address/port pairs and IPv6 address/port pairs, enabling IPv6 systems to open sessions with IPv4 systems [RFC6145] [RFC6146].

Stateful translation can be used for Scenarios 1, 3, and 5, i.e., it supports "an IPv6 network to the IPv4 Internet", "the IPv6 Internet to an IPv4 network", and "an IPv6 network to an IPv4 network".

For Scenario 1, any IPv6 addresses in an IPv6 network can use stateful translation; however, it typically only supports initiation from the IPv6 side. In addition, it does not result in stable addresses of IPv6 nodes that can be used in DNS, which may cause

problems for the protocols and applications that do not deal well with highly dynamic addresses.

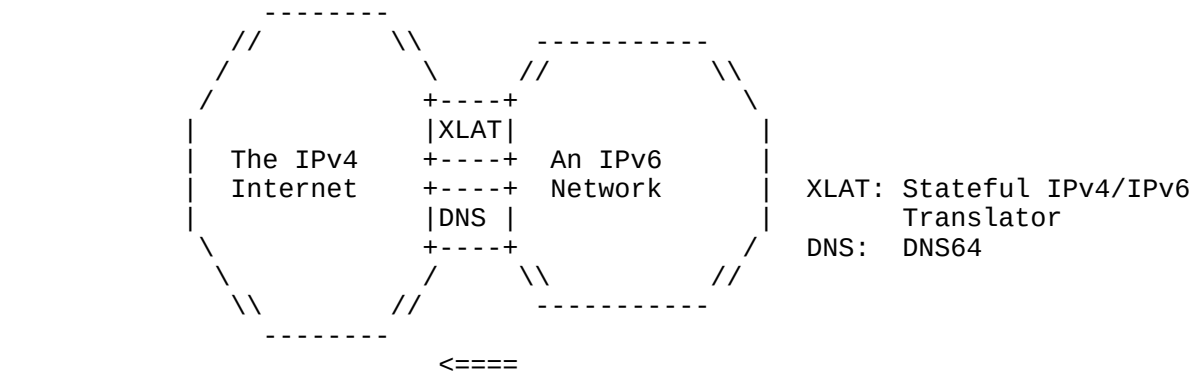

Figure 13: Stateful Translation for Scenario 1

Scenario 3 handles servers using IPv4 private addresses [RFC1918] and being reached from the IPv6 Internet. This includes cases of servers that for some reason cannot be upgraded to IPv6 and don't have public IPv4 addresses, and yet need to be reached by IPv6 nodes in the IPv6 Internet.

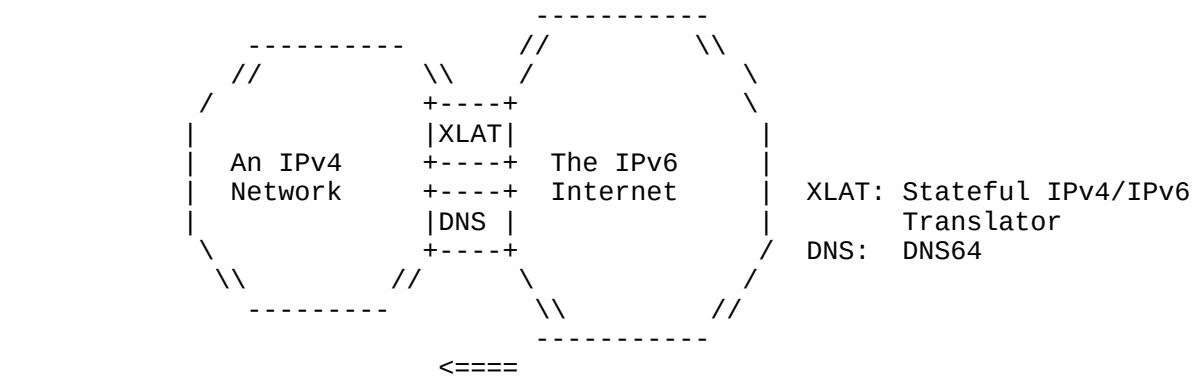

Figure 14: Stateful Translation for Scenario 3

Similarly, stateful translation can also be used for Scenario 5.

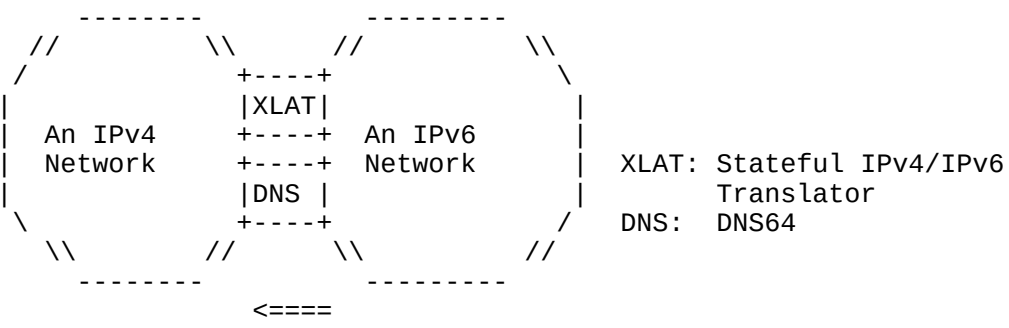

Figure 15: Stateful Translation for Scenario 5

The implementation of the stateful translator needs to refer to [RFC6145], [RFC6146], and [RFC6052].

3.3. Layout of the Related Documents

Based on the above analysis, the IPv4/IPv6 translation series consists of the following documents.

- Framework for IPv4/IPv6 Translation (this document).
- Address translation (the choice of IPv6 prefix and the choice of method by which the remainder of the IPv6 address is derived from an IPv4 address, part of the SIIT update) [RFC6052].
- IP and ICMP Translation (header translation and ICMP handling, part of the SIIT update) [RFC6145].
- Table maintenance (stateful translation including session database and mapping table handling) [RFC6146].
- DNS64 (DNS64: A to AAAA mapping and DNSSEC discussion) [RFC6147].
- FTP ALG [FTP64].
- Others (DNS46, Multicast, etc.).

The relationship among these documents is shown in the following figure.

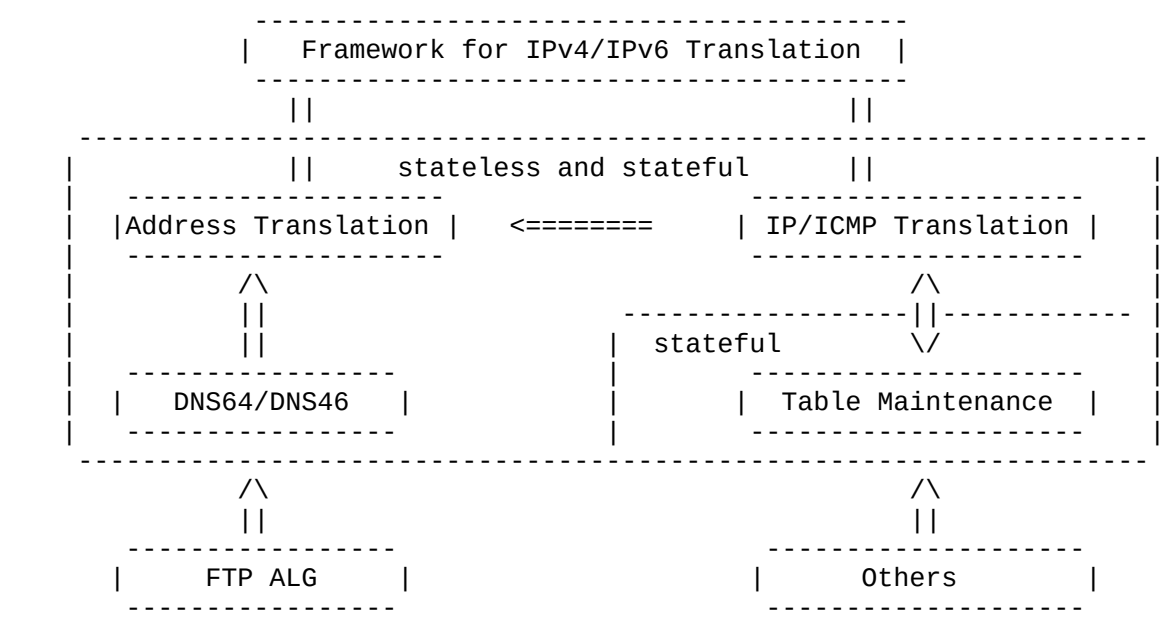

Figure 16: Document Layout

In the document layout, the IP/ICMP Translation and DNS64/DNS46 normatively refer to Address Translation. The Table Maintenance and IP/ICMP Translation normatively refer to each other.

The FTP ALG and other documents normatively refer to the Address Format, IP/ICMP Translation, and Table Maintenance documents.

4. Translation in Operation

Operationally, there are two ways that translation could be used -- as a permanent solution thereby making transition "the other guy's problem", and as a temporary solution for a new part of one's network while bringing up IPv6 services in the remaining parts of one's network. We obviously recommend the latter at the present stage. For the IPv4 parts of the network, [RFC4213]'s recommendation holds. Bring IPv6 up in those domains, move production to it, and then take down the now-unnecessary IPv4 service when economics warrant. This approach to transition entails the least risk.

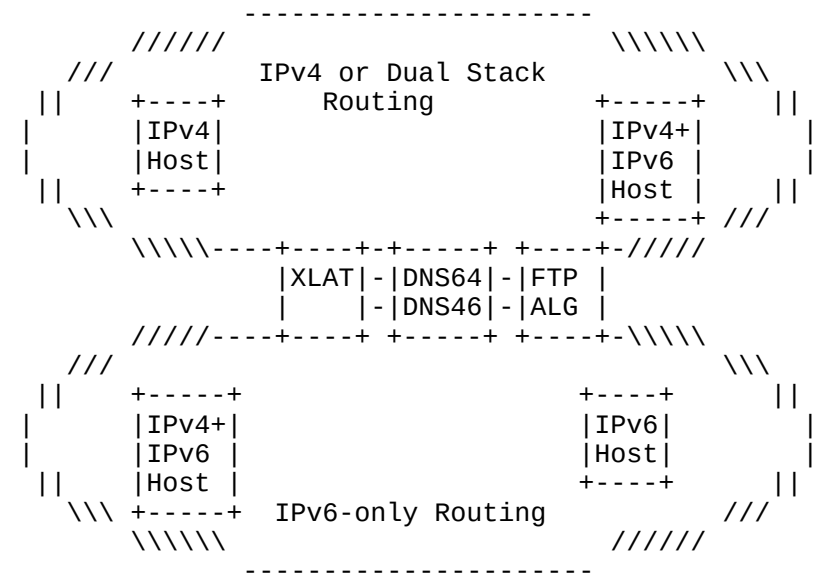

Figure 17: Translation Operational Model

Figure 17 shows that, during the coexistence phase, one expects a combination of hosts, applications, and networks. Hosts might include IPv6-only gaming devices and handsets, older computer operating systems that are IPv4-only, and modern mainline operating systems that support both. Applications might include ones that are IPv4-only and modern applications that support both IPv4 and IPv6. Networks might include dual-stack devices operating in single-stack networks (whether that stack is IPv4 or IPv6) and fully functional dual-stack networks.

5. Unsolved Problems

The framework does not cover all possible scenarios, and it may be extended in the future to address them.

6. Security Considerations

This document is the framework of IPv4/IPv6 translation. The security issues are addressed in individual IPv4/IPv6 translation documents, i.e., [RFC6052], [RFC6145], [RFC6146], [RFC6147], and [FTP64].

7. Acknowledgements

This is under development by a large group of people. Those who have posted to the list during the discussion include Andrew Sullivan, Andrew Yourtchenko, Bo Zhou, Brian Carpenter, Dan Wing, Dave Thaler, David Harrington, Ed Jankiewicz, Gang Chen, Hui Deng, Hiroshi Miyata, Iljitsch van Beijnum, John Schnizlein, Magnus Westerlund, Marcelo Bagnulo Braun, Margaret Wasserman, Masahito Endo, Phil Roberts, Philip Matthews, Remi Denis-Courmont, and Remi Despres.

Ed Jankiewicz described the transition plan.

8. References

8.1. Normative References

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

[RFC6146] Bagnulo, M., Matthews, P., and I. 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. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

8.2. Informative References

[6NET] 6NET Consortium, "6NET", <http://www.6net.org/>.

[DS-LITE] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", Work in Progress, March 2011.

[FTP64] Beijnum, I., "An FTP ALG for IPv6-to-IPv4 translation", Work in Progress, March 2011.

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

[RFC1923] Halpern, J. and S. Bradner, "RIPv1 Applicability Statement for Historic Status", RFC 1923, March 1996.

[RFC1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, "SOCKS Protocol Version 5", RFC 1928, March 1996.

[RFC2765] Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC 2765, February 2000.

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

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

[RFC3089] Kitamura, H., "A SOCKS-based IPv6/IPv4 Gateway Mechanism", RFC 3089, April 2001.

[RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.

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

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

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

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

[RFC5211] Curran, J., "An Internet Transition Plan", RFC 5211, July 2008.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

Authors' Addresses

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA

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

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing, 100084
China

Phone: +86 10-62785983
EMail: xing@cernet.edu.cn

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing, 100084
China

Phone: +86 10-62785983
EMail: congxiao@cernet.edu.cn

Kevin Yin
Cisco Systems
No. 2 Jianguomenwai Ave, Chaoyang District
Beijing, 100022
China

Phone: +86-10-8515-5094
EMail: kyin@cisco.com