IPv4/IPv6 Coexistence Framework (PET)
draft-cui-softwire-pet-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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

This Internet-Draft will expire on April 29, 2010.

Copyright Notice
IPv4 and IPv6 are expected to coexist for a long period. Currently, there are many IPv4/IPv6 transition/coexistence technologies, which can be generally divided into two categories: translation and tunneling. Both translation and tunneling have limitations and application scopes. In some typical transition scenarios, tunneling and translation are needed at the same time. In addition, there may be multiple network devices capable of doing translation or tunneling along the end-to-end path. It’s important to choose particular device(devices) for doing translation or tunneling.

This draft presents an IPv4-IPv6 transition/coexistence framework named PET (short for Prefixing, Encapsulation and Translation). PET is a network side solution which includes fundamental elements needed in various transition scenarios. In PET framework, signaling is required for transition devices to exchange necessary information and negotiate how to do combine transition and tunneling cooperatively. This draft also addresses how to deploy PETs and analyze the advantages and disadvantages of typical transition technologies that PET may adopt.
Table of Contents

1. Introduction .................................................. 5
2. Terminology ............................................... 7
3. Fundamental elements for IPv4-IPv6 coexistence ............ 8
4. PET Framework ........................................... 9
   4.1. PET operations on E-IP->I-IP->E-IP .................... 10
   4.2. PET operations on E-IP->I-IP->I-IP .................... 11
   4.3. PET operations on E-IP->E-IP->I-IP .................... 12
5. PET signaling .............................................. 14
6. Implementation issues ...................................... 15
   6.1. DNS consideration ................................... 15
7. Acknowledgements .......................................... 16
8. References ................................................. 17
   8.1. Normative References ................................. 17
   8.2. Informative References ............................... 17
Authors’ Addresses ............................................ 18
1. Introduction

Recently more and more IPv6 networks have been deployed, especially IPv6 backbone networks. However, the existing IPv4 networks still carry the major network traffic and hold the major network services and applications. It has been a common sense that IPv4 and IPv6 networks will coexist for a long term. This leads to the need of IPv4-IPv6 coexistence technologies.

There are many methods proposed for IPv4-IPv6 coexistence, which can be roughly classified into two categories: translation and tunneling. Translation is a technology that translates the semantic between IPv4 and IPv6. Examples of translation methods include SIIT [RFC 2765], NAT-PT [RFC 2766], BIS [RFC 2767], SOCKS64 [RFC 3089], BIA [RFC 3338], IVI [draft-ietf-xli-behave-ivi-02] and so on. Translation technology can realize IPv4 and IPv6 interworking directly; however, along with information loss, translation has performance issues. The address mapping in translation mechanism requires either NAT to maintain per-mapping state, or IPv6 and IPv4 address space that may be translated to be 1:1 in size. This becomes a problem when the network sizes on the side of NAT become large, which is the reason Behave WG restrict the scenario with the condition that at least one side of the NAT for IPv4-IPv6 translation has to be "a network with a clearly identifiable administrative domain" rather than Internet. Besides, from the perspective of application layer, per-application ALG is required for NAT-traversing. However, it is hard to implement ALG in hardware because of the variety of application protocol.

Tunneling is the technology that encapsulates packets of a different protocol within the protocol of the network that delivers it, so as to realize forwarding across Incompatible network. Examples of tunneling methods include IP-in-IP tunnel [RFC 2893, RFC 4213], GRE tunnel [RFC 1702], 6to4 tunnel [RFC 2893], 6over4 tunnel [RFC 2529], softwire transition technology [RFC 5655] and so on. Tunneling technology can not realize the IPv4 and IPv6 interworking; it can only deal with the scenario where two IPv4 (IPv6) nodes/networks want to communicate with each other through IPv6 (IPv4) network. However, tunneling technology has several advantages. Besides high transparency, it can easily be implemented by hardware, and only routing across the transit network is possibly required rather than introducing routing information into the network of different address family.

As described above, both translation and tunneling have limitations and application scopes. It’s probable that both tunneling and translation are required in some scenario. In addition, when there’re multiple device capable of doing translation along the network path, it’s necessary to choose which device/devices to do
translation and thus which devices to do tunneling. Therefore, we need to decide proper method for various transition scenarios and how translation and tunneling collaborate to solve transition problems. This draft presents PET framework for IPv4-IPv6 transition and coexistence. It includes fundamental elements needed in various transition scenarios and can work differently on demand. PET requires signaling process to negotiate how to do the transition along the end-to-end path and distribute necessary information to execute transition technologies. This procedure is essential for compatibility of different technologies and diversity of scenarios. In addition, this draft also addresses how to deploy PETs and analyze the advantages and disadvantages of all transition methods that PET may adopt.
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. Fundamental elements for IPv4-IPv6 coexistence

There are mainly two basic IPv4-IPv6 transition scenarios. One is to connect several edge networks of the same address family across a transit network of another address family, while the other scenario is to enable hosts/network of one address family to directly communicate with hosts/network of the other address family. We call the first scenario as heterogeneous crossing and the second as heterogeneous direct-connection. Most IPv4-IPv6 transition scenarios can be viewed as the combination of heterogeneous crossing and direct-connection. Generally, heterogeneous crossing can be achieved by tunneling or double translation, while heterogeneous direct-connection can be achieved by translation. In order to support either tunneling or translation, control plane operations always need prefix mappings or prefix announcement. So there are three fundamental elements needed in IPv4-IPv6 coexistence: prefixing, encapsulation and translation.

P: prefixing, which includes all transition operations on control plane related to subnet prefix.

For tunneling technology, prefixing includes prefix announcement, tunnel endpoint discovery, tunnel signaling and configuration, et al. For example, in Softwire technology, MP-BGP (Multi-protocol BGP) is extended to advertise the routing information of E-IP prefix with I-IP next-hop, and the parameters of tunnel endpoint before data plane forwarding. Based on this prefixing operation, the auto softwire tunnels can be established. For translation technology, prefixing includes address mappings of IPv4/IPv6, prefix configuration and announcement.

E: encapsulation, which includes all tunneling operations on data plane, such as encapsulation, decapsulation, maximum transmission unit (MTU) processing and so on. Based on this operation, packets from E-IP network are encapsulated and sent across I-IP transit network to another E-IP network according to the prefix advertisement on control plane.

T: translation, which includes all translation operations of data plane, such as address mapping and protocol translation, MTU processing. Based on address mapping, packets can be translated from one address family to another. Protocol translation is needed since IPv4 and IPv6 are not directly compatible. Here, protocol translation includes IP layer translation and application layer translation. To implement translation, PET may need to collaborate with the domain name system (DNS), since applications use domain name rather than IP address to visit other hosts a lot.
4. PET Framework

Based on the above three fundamental elements of IPv4/IPv6 coexistence, PET is a generic solution for IPv4/IPv6 transition for the scenarios of both heterogeneous crossing and direct-connection. Figure 1 shows the PET framework. We use E-IP and I-IP instead of the exact IPv4 and IPv6. Here I-IP is either IPv4 or IPv6, while E-IP is the other address family. In this framework, the middle I-IP transit network is connected with various types of networks including E-IP transit/client networks, I-IP transit/client networks and dual stack networks.

Figure 1 PET Framework

The basic idea of PET framework is to deploy PET boxes between middle I-IP transit networks and customer networks (E-IP or I-IP). PET box should support the above basic functions for IPv4/IPv6 coexistence, i.e. prefixing, encapsulation and translation. PET signaling is also an essential part of the framework since multiple PET boxes need to cooperate with each other. Through signaling, PET boxes can distribute necessary information and negotiate the particular functionalities executed on different boxes. PET signaling will be
analyzed in the next section.

We can extract three typical scenarios from Figure 1, including "E-IP->I-IP->E-IP", "E-IP->I-IP->I-IP" and "E-IP->E-IP->I-IP" (without the loss in generality, we assume that host in the first network initiate the connection). For different scenarios, PET can provide different functionalities to ensure the inter-working of E-IP/I-IP network. We will analyze how PET works in the three scenarios in the following subsections.

4.1. PET operations on E-IP->I-IP->E-IP

This is the scenario where an E-IP network wants to talk with another E-IP network across I-IP transit. There are two methods PET can adopt to handle this scenario. One is translation and the other is tunneling. If PET uses translation method, it’ll need twice translations. An E-IP packet need to be translated by the first PET which is I-IP ingress, into an I-IP packet for being delivered through I-IP transit. When this packet arrives at the second PET which is I-IP egress, it will be translated back into an E-IP packet for being delivered in destination E-IP network.

The other method for E-IP-I-IP-E-IP scenario is tunneling. This requires a PET to encapsulate the packets and send them to the other tunnel endpoint PET across I-IP transit. When these packets arrive at the second PET, they are decapsulated and sent to E-IP customer network.

Because translation method will bring heavier load because of address mapping and protocol translation, PET prefers to use tunneling technology to handle E-IP-I-IP-E-IP scenario. Its operations are shown in Figure 2.

```
+---------------+   +---------------+      +---------------+    +---------------+
|E-IP customer|   |     PET      |      |     PET      |    |E-IP customer|
|   network   |   |             |      |             |    |   network   |
+---------------+   +---------------+      +---------------+    +---------------+
                      |                  |
---forwarding--->    |              ---forwarding--->
                      |                  |
              |                  |
              |                  |
              ---encapsulation---
              |                  |
              |                  |
              ---tunneling--->
              |                  |
              |                  |
              ---decapsulation---
              |                  |
              |                  |
              ---forwarding--->
```

Figure 2 PET operations in E-IP->I-IP->E-IP scenario
4.2. PET operations on E-IP->I-IP->I-IP

This is the scenario where an E-IP customer network wants to talk with an I-IP customer network across I-IP transit. There are two methods to deal with this scenario. One is translation plus forwarding (Figure 3). The other is tunneling plus translation (Figure 4).

In the first method, when an E-IP packet arrives at the first PET(IPv4-IPv6 edge), it will be translated into an I-IP packet and then sent to the I-IP client network through I-IP transit. In the second method, when an E-IP packet arrives at the first PET, it will be encapsulated as an I-IP packet for being delivered through I-IP transit. Once this packet arrives at the other tunnel endpoint PET, it will be decapsulated to the original E-IP packet and then be translated into an I-IP packet to deliver to the I-IP customer network.

Both method works in theory, though the second method seems more complicated; however the key factor here is that translation brings heavy load, so the two related PET box need a signaling process to negotiate which one of them to do the translation, i.e. which method to use. In principle, it’s better that translation is done by the PET whose performance is better and the sizes of the networks on the side of which is smaller. We’ll discuss PET signaling in the next section.

Figure 3 PET operations (A) in E-IP->I-IP->I-IP scenario (Translation + Forwarding)
4.3. PET operations on E-IP->E-IP->I-IP

This is the scenario where an E-IP customer network wants to talk with an I-IP customer network across E-IP transit. There are two methods to deal with this scenario. One is forwarding plus translation. The other is translation plus tunneling.

In the first method, when an E-IP packet arrives at the first PET, it will be directly delivered to E-IP transit. At the edge of E-IP-I-IP, i.e. the second PET, the packet will be translated into an I-IP packet for delivering in I-IP customer network. In the second method, when an E-IP packet arrives at the first PET, it will be translated into an I-IP packet, and then PET encapsulates it and sends it to the second PET. When the E-IP packet encapsulating the translated I-IP packet arrives at the second PET, the I-IP packet will be decapsulated and sent to the I-IP customer network.

Here both methods work, too. We also require PET signaling to negotiate which PET to do translation.
Figure 5 PET operations (A) in E-IP→E-IP→I-IP scenario (Forwarding + Translation)

Figure 6 PET operations (B) in E-IP→E-IP→I-IP scenario (Tunneling + Translation)
5. PET signaling

We can see from section 4 that translation is inevitably adopted in a lot typical scenarios while it’s clear that translation technology has performance and scalability issues. So it’s important that translation is done on a proper PET when necessary. The principle here is that PET box whose performance is better and the sizes of the networks on the side of which is smaller, is preferred to do translation.

Here we propose a new concept called Translation Preference (TP) to express the PET’s willing of performing translation according to some policies the network administrators constitute. Through exchanging the Translation preference among PETs, PET framework can choose which PET to do translation and decide the communication mode for corresponding scenarios. TP can be decided by the performance of PET box, the size of networks which PET is connected to, and the administrators’ policy. We define the value of TP as the preference degree that the PET would like to translate a packet; the higher the TP value is, the higher probability the PET translates packets.

Besides TP, there is another type of information that PETs would want to exchange when translation is used, we call it PET prefix. PET prefix represents the prefix of network the PET box charges. Based on that other PETs can differentiate an IPv6 mapped address from a regular IPv6 address, thus learn about the source or destination of a packet. For example, in a IPv6-IPv6-IPv4 scenario, PET can announce a /96 prefix in signaling to express that all the IPv6 address within the prefix is a mapped IPv6 address for a IPv4 host; if the address mapping is done through add/remove IPv6 prefix, then PET learns how to do address mapping based on this information.

PET may need to exchange more information or do other negotiations in the future, we can use the signaling process to achieve it as long as it’s necessary. We won’t define how to do PET signaling for TP and PET prefix exactly and how the message formats should be in this draft, but we believe that this signaling process should be done by softwire signaling process through a slight extension of MP-BGP.
6. Implementation issues

In this draft, we recommend how to use tunneling and translation method in each scenario using PET. However, we do not restrict the specific tunneling and translation technology that PET adopts. It can be any transition technology, such as SIIT [RFC 2765], NAT-PT [RFC2766], BIS [RFC 2767], SOCKS64 [RFC 3089], BIA [RFC 3338], IVI [draft-ietf-xli-behave-ivi-02], IP-in-IP tunnel [RFC 2893, RFC 4213], GRE tunnel [RFC 1702], 6to4 tunnel [RFC 2893], 6over4 tunnel [RFC2529], softwire transition technology [RFC 5565] and so on.

6.1. DNS consideration

It’s a concern how to make PET collaborate with DNS system. In the translation schemes proposed lately, DNS is usually considered because hosts use domain names rather than IP address directly to visit other hosts a lot. Generally the DNS query strategy of these schemes can be divided into two types. The first method is to build a DNS ALG on the NAT and make all the DNS query and reply go through the NAT; the second method is to use DNS64 as a extended DNS server which collaborate with NAT to do the DNS translating. PET which may use these translation mechanisms introduces a slight change in that the PET box doing the translation may be not the PET box that is in the IPv4-IPv6 edge, so it’s a little complicated. PET boxes may need to remember the DNS server address behind the other PET boxes. However, the influence of PET using these translation schemes on DNS varies among different schemes, so we won’t discuss it one by one in this framework.
7. Acknowledgements

The authors would like to thank Lixia Zhang and Eric Nordmark for their valuable comments on this draft.
8. References

8.1. Normative References


8.2. Informative References

Authors' Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-6278-5822
Email: cy@csnet1.cs.tsinghua.edu.cn

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-6278-5822
Email: xmw@csnet1.cs.tsinghua.edu.cn

Shengling Wang
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-6278-5822
Email: slwang@csnet1.cs.tsinghua.edu.cn

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn
Xing Li
Tsinghua University
Department of Electronic Engineering, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
Email: xing@cernet.edu.cn

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, Ca. 95134
USA

Email: chmetz@cisco.com