

software Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 1, 2010

Y.Cui  
M. Xu  
S.Wang  
X.Li  
J.Wu  
Tsinghua

University

Chris Metz  
cisco systems  
July 2, 2009

PET-based framework for IPv4/IPv6 coexistence  
[draft-cui-software-pet-framework-00.txt](#)

#### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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 January 1, 2010.

#### Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

#### Abstract

IPv6 offers significant advantages over IPv4, however it will take a long time to replace IPv4 with IPv6. Therefore, these two protocols

are expected to coexist during the transition period. Currently,

there are many transition technologies, such as translation and tunneling. In some typical transition scenarios, both tunneling and translation are needed. However, either translation or tunneling has limitation and application scope. In addition, besides IP version of source, middle and destination network, the network property (a regular edge network or a backbone) has key impact on system performance. Therefore, we need to decide which transition method should be used in some typical transition scenarios and how transition and tunneling collaborate for solving transition problems. This draft presents an IPv4-IPv6 transition framework, which is a network side transition solution. It introduces a toolbox named PET (short for Prefixing, encapsulation and translation) to solve IPv4-IPv6 transition. PET includes fundamental elements needed in transition scenarios, which provides the flexibility for network to decide the proper transition methods. In addition, this draft also addresses how to deploy PETs and analyze the advantages and disadvantages of all transition methods that PET may adopt.

## Table of Contents

<a href="#">1. Introduction.....</a>	<a href="#">2</a>
<a href="#">2. Requirements Terminology.....</a>	<a href="#">4</a>
<a href="#">3. Fundamental requirements of IPv4-IPv6 transition methods....</a>	<a href="#">4</a>
<a href="#">4. Descriptions of PET.....</a>	<a href="#">5</a>
<a href="#">5. PET Framework.....</a>	<a href="#">6</a>
<a href="#">5.1. IPv4-IPv6-IPv4.....</a>	<a href="#">8</a>
<a href="#">5.2. IPv4-IPv6-IPv6.....</a>	<a href="#">9</a>
<a href="#">5.3. IPv6-IPv6-IPv4.....</a>	<a href="#">11</a>
<a href="#">5.4. IPv6-IPv4-IPv6.....</a>	<a href="#">11</a>
<a href="#">5.5. IPv4-IPv4-IPv6.....</a>	<a href="#">12</a>
<a href="#">5.6. IPv6-IPv4-IPv4.....</a>	<a href="#">12</a>
<a href="#">6. Implementation issues.....</a>	<a href="#">14</a>
<a href="#">7. Acknowledgment.....</a>	<a href="#">14</a>
<a href="#">8. References.....</a>	<a href="#">15</a>
<a href="#">8.1. Normative References.....</a>	<a href="#">15</a>
<a href="#">8.2. Informative References.....</a>	<a href="#">16</a>
Author's Addresses.....	<a href="#">16</a>

## [1. Introduction](#)

Recently more and more IPv6 networks have been deployed, especially IPv6 backbone networks, while the existing IPv4 networks still carry the major network traffic and hold the major network services and applications, though facing serious address space problem and other problems. It has been agreed that IPv4 and IPv6 networks will co-



exist for a long term. This leads to the need of IPv4-IPv6 transition methods.

There are many methods for IPv4-IPv6 transition, which can be roughly classified into two groups: translation and tunneling. Translation is a technology that translates semantic between IPv4 and IPv6. There are many translation methods, such as 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 interworking between IPv4 and IPv6 directly, however, it will lead to information loss.

Tunneling is a technology to encapsulate packets from a different protocol within the protocol of the route that delivers it to the target network. There are many tunneling methods, such as IP-in-IP tunnel [[RFC 2893](#), [RFC 4213](#)], GRE tunnel [[RFC 1702](#)], 6to4 tunnel [[RFC 2893](#)], 6over4 tunnel [[RFC 2529](#)], softwire transition technology [[RFC 5565](#)] and so on. Tunneling technology can not realize the interworking between IPv4 and IPv6 directly. It can only deal with the scenario where two IPv4 (IPv6) nodes want to communicate with each other through IPv6 (IPv4) network. However, tunneling technology has several advantages, besides no information loss, it can be realized easily by hardware, and does not introduce routing information into a network with different address family.

In some typical transition scenarios, both tunneling and translation are needed. However, as described above, either translation or tunneling has limitation and application scope. In addition, besides IP version of source, middle and destination network, the network property (a regular edge network or a backbone) has key impact on system performance. Therefore, we need to decide which transition method should be used in some typical transition scenarios and how transition and tunneling collaborate for solving transition problems. This draft presents an IPv4-IPv6 transition framework, which is a network side transition solution. It introduces a toolbox named PET (short for Prefixing, encapsulation and translation) to solve IPv4-IPv6 transition. PET includes fundamental elements needed in transition scenarios, which provides the flexibility for network to decide the proper transition methods. 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. Requirements 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 requirements of IPv4-IPv6 transition methods**

There are two main IPv4-IPv6 transition scenarios. One is to connect several edge networks with the same address family across a transit core network with another address family; the other scenario is to make hosts with one address family capable of directly communicating with hosts with the other address family. We call the first scenario heterogeneous crossing. The scenario where two IPv4 (IPv6) nodes want to communicate with each other through IPv6 (IPv4) network belongs to heterogeneous crossing. We call the second scenario heterogeneous direct-connection. The scenario where an IPv4 (IPv6) node wants to directly communicate with an IPv6 (IPv4) node belongs to heterogeneous direct-connection.

In fact, most IPv4-IPv6 transition scenarios can be viewed as the combination of heterogeneous crossing and direct-connection. Hence, the fundamental transition elements needed in heterogeneous crossing and direct-connection are those needed in most IPv4-IPv6 transition scenarios.

In heterogeneous crossing scenario, tunneling technology can be used to transmit IPv4 (IPv6) packets through IPv6 (IPv4) networks. In addition, through twice translations, IPv4 (IPv6) packets can also be transmitted through IPv6 (IPv4) networks in heterogeneous crossing scenario. In heterogeneous direct-connection scenario, when IPv4 (IPv6) nodes want to communicate with IPv6 (IPv4) nodes directly, it can only use translation technology.

In addition, when adopting tunneling for supporting IPv4/IPv6 interworking, some control operations involved with subnet prefix should be done beforehand. These operations include prefix announcement, tunnel endpoint discovery, the selection of tunnel endpoint and tunnel belonging to the same tunnel endpoint, tunnel configuration, and so on. Similarly, when adopting translation method for supporting IPv4/IPv6 interworking, some control operations involved with subnet prefix also should be done in advance. These operations include the establishment of address mapping mechanism, prefix configuration and so on. We call these control operations involved with subnet prefix prefixing.

In conclusion, there are three fundamental elements needed in IPv4-IPv6 transition scenarios, i.e. prefixing, encapsulation and translation.

To realize a framework in network side for IPv4/IPv6 translation, this draft introduces a toolbox named PET which includes the above fundamental elements. The detailed descriptions are given in [section 4](#).

#### **4. Descriptions of PET**

**PET is a smart transition toolbox supporting IPv4/IPv6 inter-working.** It can deal with the heterogeneous crossing and direct-connection scenarios. Because all IPv4-IPv6 transition scenarios can be viewed as the combination of the heterogeneous crossing and direct-connection, the PET-based transition method is a generic solution for IPv4/IPv6 transition. PET toolbox has the following functions:

P: representing prefixing. Prefixing includes all transition operations of control plane involved with subnet prefix.

In detail, in tunneling technology, prefixing includes prefix announcement, tunnel endpoint discovery, the selection of tunnel endpoint and tunnel belonging to the same tunnel endpoint, and so on. For example, in software transition technology, the IPv6 prefix and IPv4 next-hop mapping information should be announced out through extended MP-BGP (Multi-protocol BGP) signaling beforehand. Based on this prefixing operation, the automatic 4over6 tunnels can be established. In translation technology, prefixing includes the establishment of address mapping mechanism, prefix configuration and so on. For example, in IIVI-based translation scheme, the global IPv6 prefix should be configured in an autonomous domain (AS) beforehand, to form the global IIVI address, thus realizing the stateless translation.

E: representing encapsulation. E includes all tunneling operations of data plane, such as encapsulation, decapsulation and maximum transmission unit (MTU) processing and so on.

Through this operation, packets from IPv6 (IPv4) network are encapsulated on the PET toolbox and sent across IPv4 (IPv6) backbone to another IPv6 (IPv4) network according to the mappings stored on the PET box.

T: representing translation. It includes all translation operations of data plane, such as address mapping and protocol translation, MTU processing.

Address mapping is to map IPv4 addresses to IPv6 addresses, and vice versa. Based on address mapping, packets can be translated from one address family to another. IPv4 and IPv6 are not directly compatible, so programs and systems designed on one standard can not communicate with those designed to the other. Hence we need protocol translation. Here, protocol translation includes IP layer translation and application layer translation. Through protocol translation, the semantic of IP layer and application layer of an IPv4 packet is equivalent with that of the translated IPv6 packet and vice versa. In addition, to implement translation, PET may collaborate with the domain name system (DNS).

The basic idea of our solution is to deploy several PET toolboxes between backbone network and customer networks. The following section will discuss how PET deals with different IPv4/IPv6 translation scenarios in detail.

## **5. PET Framework**

**Figure 1 shows the overall topology of PET framework, which uses PET boxes between IPv6 backbone and IPv4 customer networks.** In this topology, an IPv6 backbone is connected with several customer networks including IPv4 backbone, IPv4 virtual private networks (VPNs), IPv6 network and dual stack networks.



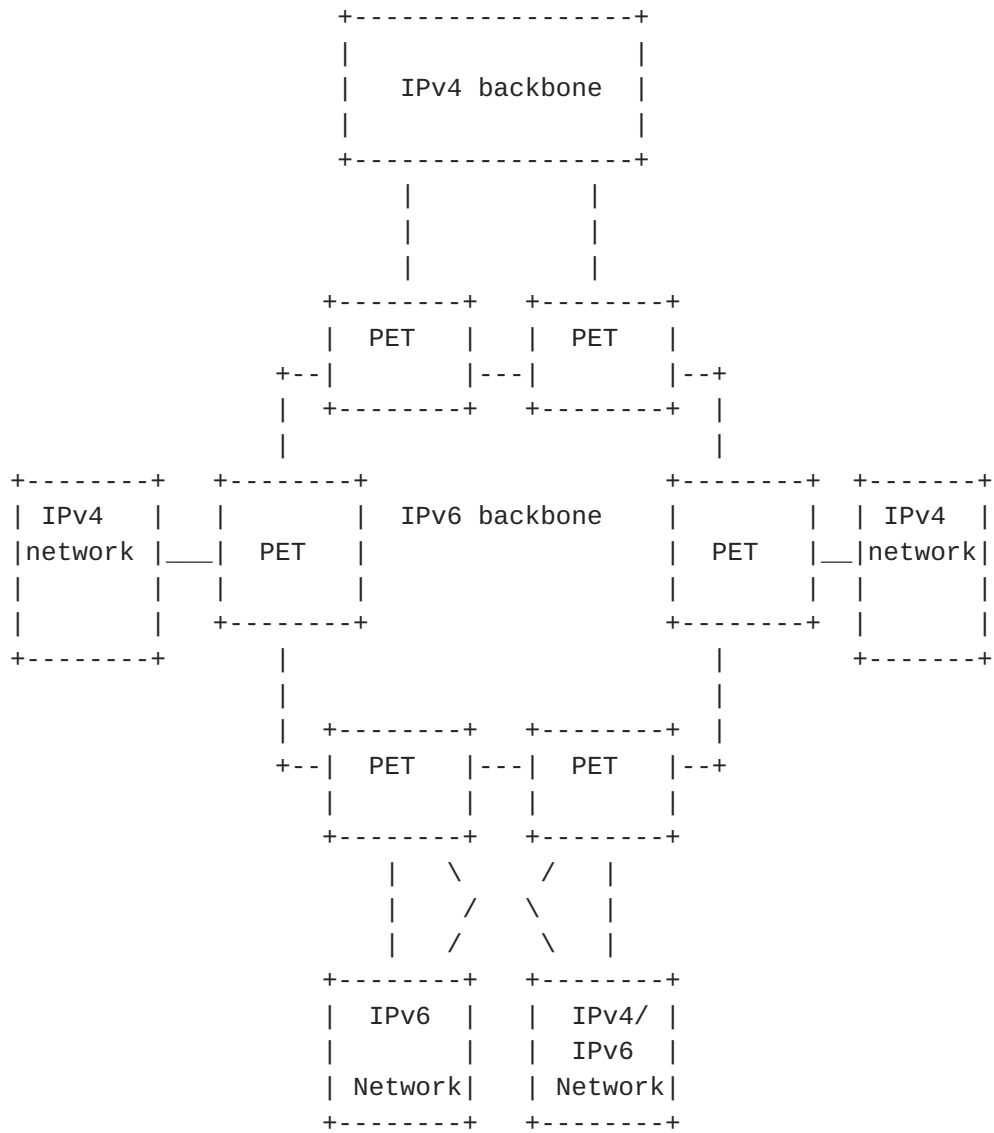


Figure 1: Topology of PET Framework

For different transition scenarios, PET can provide different functionalities to ensure the inter-working of IPv4/IPv6 network. We

will analyze how PET works in some typical scenarios in the following subsections.

### 5.1. IPv4-IPv6-IPv4

This is the scenario where an IPv4 network wants to talk with another IPv4 network across IPv6 backbone. There are two methods for PET to handle this scenario. One is translation and the other is tunneling. If PET adopts translation method, we need twice translations. In detail, an IPv4 packet need be translated by PET into an IPv6 packet for being delivered through IPv6 backbone. When this packet arrives at another PET, it will be translated into an IPv4 packet again for being delivered through IPv4 network.

The other method for IPv4-IPv6-IPv4 scenario is tunneling. This requires a PET to encapsulate the packets and send them to the tunnel endpoint PET across IPv6 backbone. When these packets arrive at the tunnel endpoint PET, they are de-capsulated and sent to IPv4 customer networks.

Because translation method will incur information loss, PET prefers to use tunneling technology to handle IPv4-IPv6-IPv4 scenario. Its operations are shown in Figure 2.

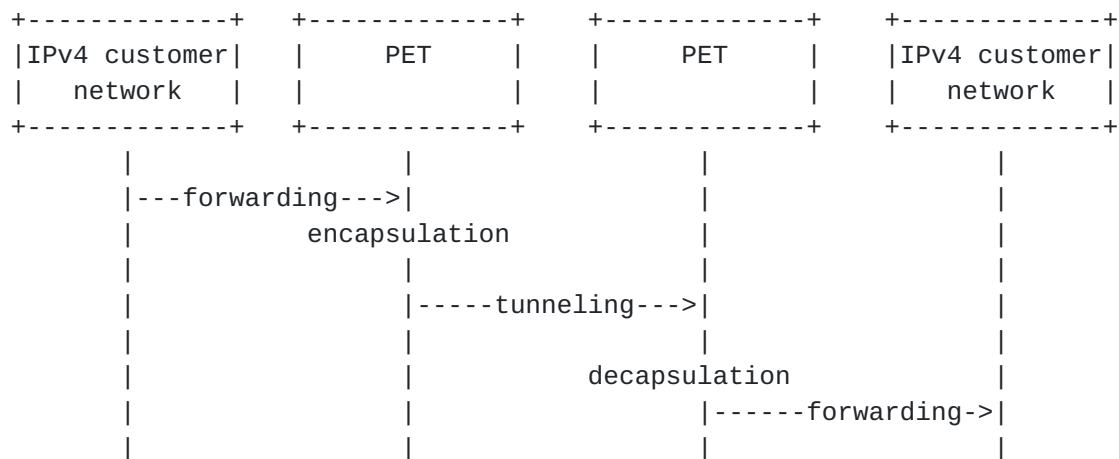


Figure 2: PET operations in IPv4-IPv6-IPv4 scenario

## 5.2. IPv4-IPv6-IPv6

This is the scenario where an IPV4 customer network wants to talk with an IPV6 customer network across IPV6 backbone. There are two methods to deal with this scenario. One is translation plus forwarding. The other is tunneling plus translation.

In the first method, when an IPv4 packet arrives at PET, it will be translated into an IPv6 packet and then sent to the IPv6 network through IPv6 backbone. In the second method, when an IPv4 packet arrives at PET, it will be encapsulated as an IPv6 packet for being delivered through IPv6 backbone. Once this packet arrives at the tunnel endpoint PET, it will be de-capsulated to the original IPv4 packet and then be translated as an IPv6 packet to be delivered to the IPv6 customer network.

If the IPv4 customer network is not an IPv4 backbone, PET prefers to adopt the first method because the complexity of second method is higher than that of the first method. Its operation is shown in Figure 3.

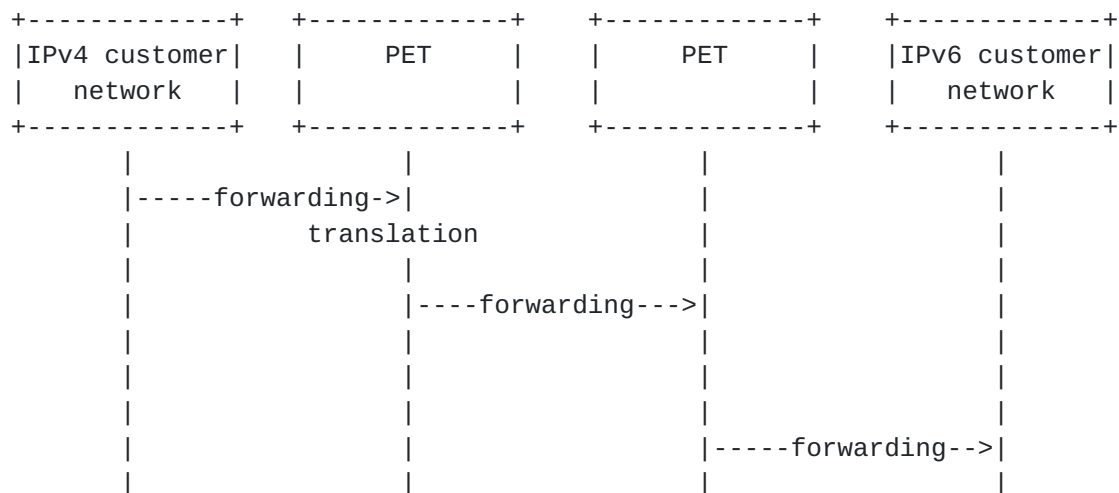


Figure 3 : PET operations in IPv4-IPv6-IPv6 scenario (IPv4 customer network is not a backbone)

If the IPv4 customer network is a backbone, PET prefers to adopt the second method for the following reasons:

i) Translation mechanism usually needs application level gateway (ALG), which is an application specific agent that allows an IPV6 node to communicate with an IPV4 node and vice versa. Backbone network requires hardware forwarding for high speed transmission. However, it is hard to use hardware to do the work of ALG.

ii) To avoid single point of failure, several PETs usually be deployed among networks. They improve performance and robustness using dynamic routing mechanism. However, translation is a static process. It is hard to use dynamic routing mechanism.

iii) At last, some translation mechanisms, such as IIVI-based scheme, require IPv4 routing information to be introduced into IPv6 backbone, which will increase the routing base size. Based on the above analyses, PET refers to adopt the second method to deal with this scenario. Its operations are shown in Figure 4.

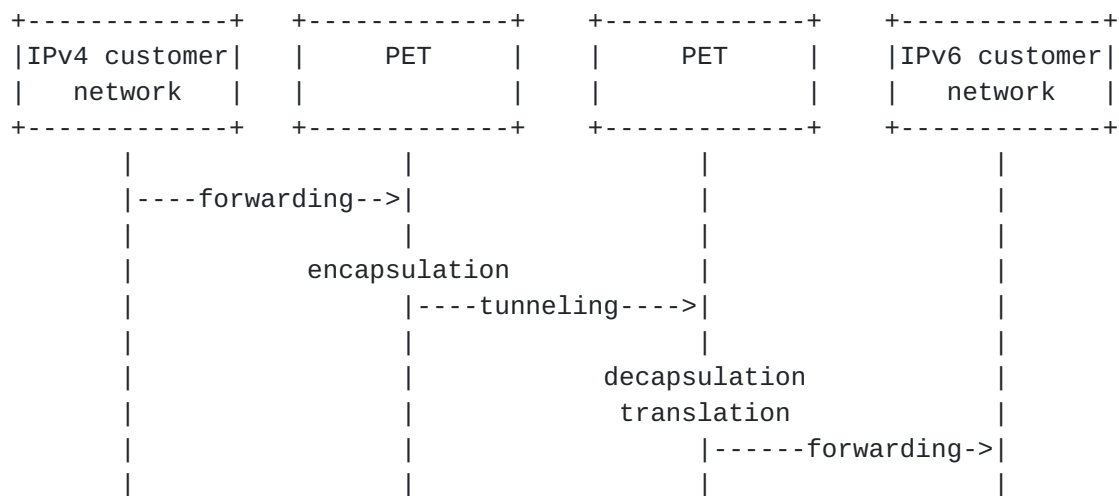


Figure 4: PET operations in IPv4-IPv6-IPv6 scenario (IPv4 customer network is a backbone)

### 5.3. IPv6-IPv6-IPv4

This is the scenario where an IPv6 customer network wants to talk with an IPv4 customer network across IPv6 backbone. In this scenario, when an IPv6 packet arrives at PET, it will be translated as an IPv4 packet and then the PET encapsulates it and sends it to the tunnel endpoint PET. When the translated IPv4 packet arrives at the tunnel endpoint PET, it will be de-capsulated and sent to the IPv4 customer network. The operations are shown in Figure 5.

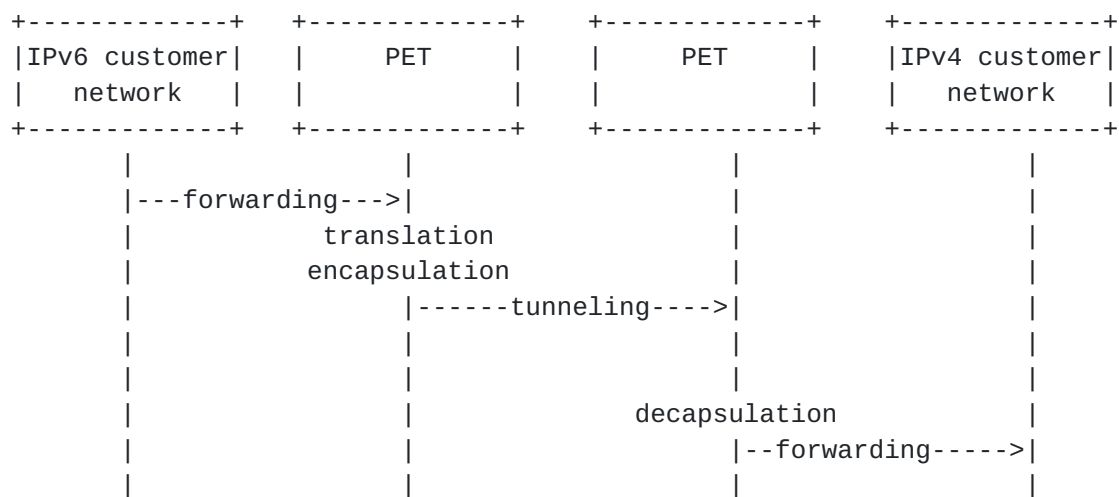


Figure 5 : PET operations in IPv6-IPv6-IPv4 scenario

### 5.4. IPv6-IPv4-IPv6

This is the scenario where an IPv6 network wants to talk with another IPv6 network across IPv4 backbone. This scenario is similar to IPv4-IPv6-IPv4 scenario. Hence, PET prefers to use tunneling technology to handle this scenario. Its operations are shown in Figure 6.

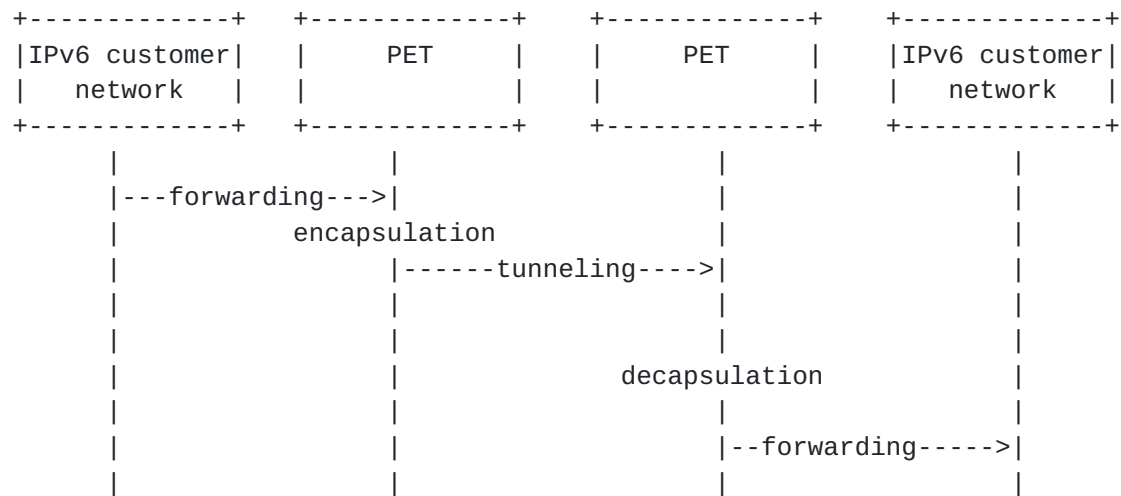


Figure 6: PET operations in IPv6-IPv4-IPv6 scenario

### 5.5. IPv4-IPv4-IPv6

This is the scenario where an IPv4 customer network wants to talk with an IPv6 customer network across IPv4 backbone. This scenario is similar to IPv6-IPv6-IPv4 scenario. Hence, PET adopts the similar method to deal with this scenario. Its operations are shown in Figure 7.

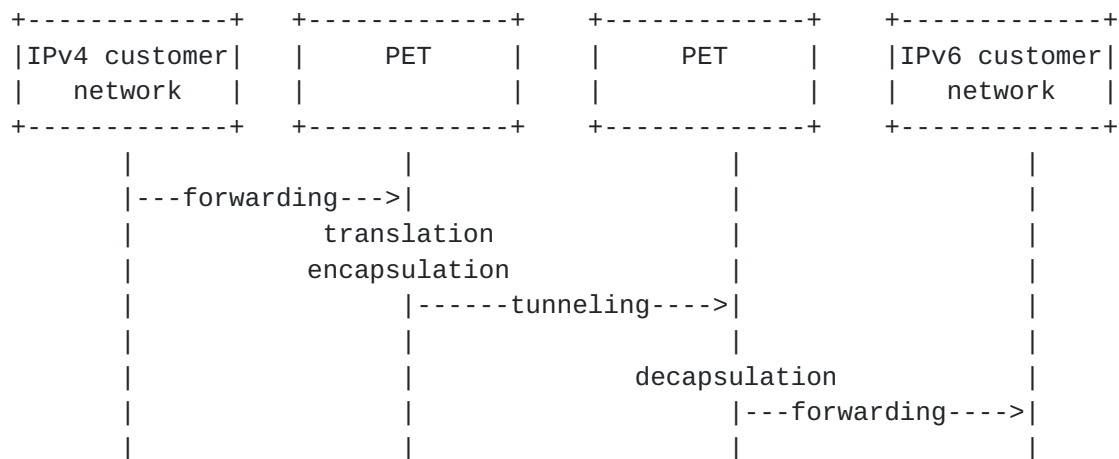


Figure 7: PET operations in IPv4-IPv4-IPv6 scenario

## 5.6. IPv6-IPv4-IPv4

This is the scenario where an IPv6 customer network wants to talk with an IPv4 customer network across IPv4 backbone. This scenario is

similar to IPv4-IPv6-IPv6 scenario. Hence, PET adopts the similar method to deal with this scenario. Its operations are shown in Figures 8 and 9.

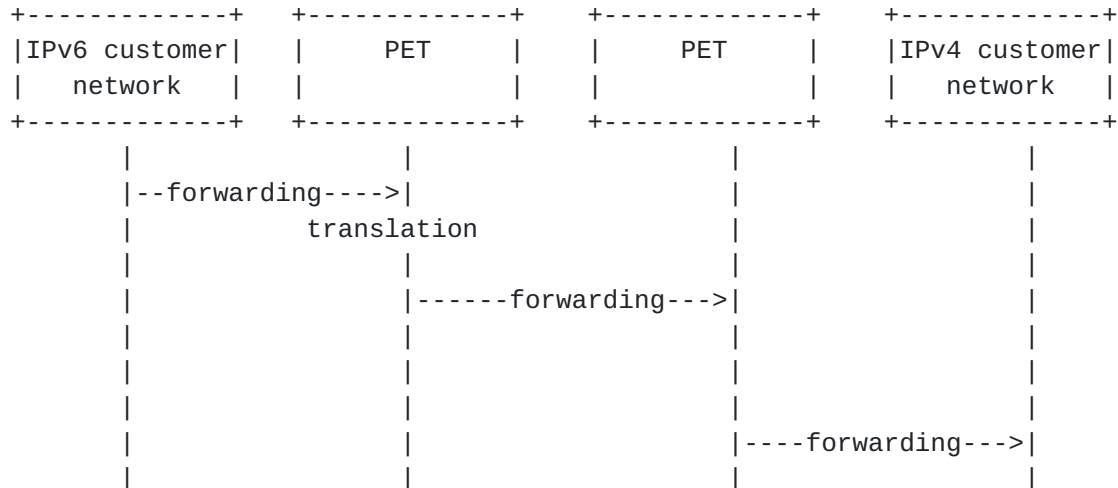


Figure 8: PET operations in IPv6-IPv4-IPv4 scenario (IPv6 customer network is not a backbone)

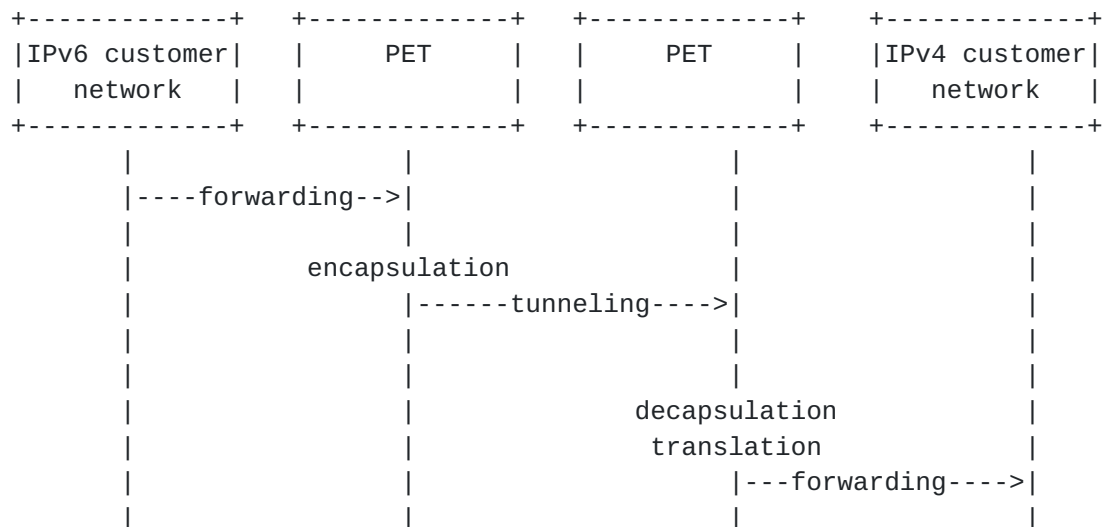


Figure 9: PET operations in IPv6-IPv4-IPv4 scenario (IPv6 customer network is a backbone)

## **6. Implementation issues**

In this draft, we recommend how to use tunneling and translation method in each scenario using PETs. 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](#)], IIVI[[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.

In addition, this draft does not address how PET collaborates with DNS, ALG and other transition devices, as well as inter domain transition problems, which will discuss in the next version.

## **7. Acknowledgment**

The authors would like to thank Lixia Zhang for her valuable comments on this draft.



## 8. References

### 8.1. Normative References

- [1] [[RFC2765](#)] E.Nordmark. "Stateless IP/ICMP Translation Algorithm
- [2] (SIIT)", [RFC 2765](#), February 2000
- [3] [[RFC2766](#)] G. Tsirtsis, P. Srisuresh." Network Address Translation - Protocol Translation (NAT-PT)", [RFC 2766](#), February 2000
- [4] [[RFC2767](#)] K. Tsuchiya, H. Higuchi, Y. Atarashi. "Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)", [RFC 2767](#), February 2000
- [5] [[RFC3089](#)] H. Kitamura." A SOCKS-based IPv6/IPv4 Gateway Mechanism" [RFC 3089](#), April 2001
- [6] [[RFC3338](#)] S. Lee, M-K. Shin, Y-J. Kim, E. Nordmark, A. Durand. "Dual Stack Hosts Using "Bump-in-the-API" (BIA)", [RFC 3338](#), October 2002
- [7] [[RFC2893](#)] R. Gilligan, E. Nordmark. "Transition Mechanisms for IPv6 Hosts and Routers", [RFC 2893](#), August 2000
- [8] [[RFC4213](#)] E. Nordmark, R. Gilligan. "Basic Transition Mechanisms for IPv6 Hosts and Routers", [RFC 4213](#), October 2005
- [9] [[RFC1702](#)] S. Hanks, T. Li, D. Farinacci, P. Traina." Generic Routing Encapsulation over IPv4 networks", [RFC 1702](#), October 1994
- [10] [[RFC2893](#)] R. Gilligan, E. Nordmark." Transition Mechanisms for IPv6 Hosts and Routers", [RFC 2893](#), August 2000
- [11] [[RFC2529](#)] B. Carpenter, C. Jung." Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", [RFC2529](#), March 1999
- [12] [[RFC5565](#)] J. Wu, Y. Cui, C. Metz, E. Rosen. " Software Mesh Framework", [RFC 5565](#), June 2009

## **8.2. Informative References**

[I-D. [draft-ietf-xli-behave-ivi-02](#)]

X. Li, C. Bao, M. Chen, H. Zhang, J. Wu. " The CERNET IVI  
Translation Design and Deployment for the IPv4/IPv6  
Coexistence and Transition", Internet Draft, June 13, 2009

### Author's Addresses

Yong Cui  
Tsinghua University  
Department of Computer Science, Tsinghua University  
Beijing 100084  
P.R.China

Phone: +86-10-6278-5822  
Email: cuiyong@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

Xing Li  
Tsinghua University  
Network Center, Tsinghua University  
Beijing 100084  
P.R.China

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

Jianping Wu  
Tsinghua University  
Network Center, Tsinghua University  
Beijing 100084  
P.R.China

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

Chris Metz  
cisvo systems  
170 West Tasman Drive  
San Jose 95134-1706  
CA

Email: chmetz@cisco.com