Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2011

Y. Cui
M. Xu
P. Wu
S. Wang
J. Wu
X. Li
Tsinghua University
C. Metz
Cisco Systems, Inc.
October 25, 2010

Translation Spot Negotiation in IPv4/IPv6-Coexist Mesh draft-cui-softwire-pet-03

Abstract

IPv4 and IPv6 are expected to coexist for a long period. Currently, there are many IPv4/IPv6 transition/coexistence techniques, roughly divided into the categories of tunneling and translation. Tunneling and translation have respective application scopes, and translation has some technical limitations, including scalability issue, application layer translation, operation complexity, etc. To improve the availability of translation, this draft proposes the method of selecting appropriate translation spot to execute translation. When the translation spot is not on IPv4-IPv6 border, tunnel is used to achieve the traversing between translation spot and IP border. This method applies well in mesh scenario where both IPv4 and IPv6 client network exists, and BGP can be extended to achieve a translation spot signaling.

Table of Contents

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 widely believed that IPv4 and IPv6 networks will coexist for a long term. This leads to the demand for IPv4-IPv6 coexistence technology.

Till now there are two types of IPv4-IPv6 coexistence techniques: tunneling and translation. Tunneling can achieve IPv4-over-IPv6/ IPv6-over-IPv4 traversing, by means of encapsulation and decapsulation. Examples of tunneling methods include IP-in-IP tunnel [RFC2893][RFC4213], GRE tunnel [RFC1702], 6to4 tunnel [RFC3056], 6over4 tunnel [RFC2529], softwire mesh technique [RFC5565], etc. Tunneling is transparent and light-weighted. It can be implemented fully by hardware.

On the other hand, translation is used to achieve IPv4-IPv6 inter-communication, by means of converting the semantic between IPv4 and IPv6. Examples of translation methods include SIIT [RFC2765], NAT-PT [RFC2766], BIS [RFC2767], BIA [RFC3338], IVI [I-D.xli-behave-ivi], NAT64 [I-D.ietf-behave-v6v4-xlate-stateful] and so on. Translation can achieve IPv4-IPv6 interworking which tunneling cannot do, but it has several technical limitations:

- Scalability. In stateful translation, the dynamic mapping of (address, port) tuple should be maintained on the translation device. The total number of mapping entries is up to the order of flow number. As to stateless translation, it has to consume IPv4 addresses to satisfy IPv6 hosts. This is also not scalable since IPv6 address space is much larger than IPv4 address.

- Application layer translation. Since translation will modify the address of an IP packet, or we say an end host, an application protocol that contains IP addresses in its payload won’t work if we don’t convert the addresses. However, due to the variety of applications protocols, it’s unrealistic for the translation device to support all of them.

- Operation complexity. To accomplish correct translation, the following operations are required: address or (address, port) tuple conversion, IP and ICMP fields translation, TCP/UDP checksum re-computing, application layer detection and translation, fragmentation when necessary. It’s rather complex for a per-packet process and probably unacceptable when the volume is high.

- Lack of efficient NAT46 translation mechanism. No efficient IPv4 to IPv6 communication mechanism has been proposed since NAT-PT. A fundamental difficulty here is that IPv6 address space is much larger than IPv4 so the translation mechanism has to make DNS or other addressing method stateful. Obviously this is not scalable.

Though facing all these issues, translation is irreplaceable in its application scope, so it’s necessary to find a way to improve its availability. To solve this problem, this draft proposes the method of finding the appropriate translation spot to execute translation. The method adopts tunnel when necessary, to achieve traversing between translation spot and IP border. As an attempt, this draft applies the method in IPv4/IPv6-coexist mesh scenario, and extends BGP to achieve translation spot signaling in the scenario.

2. Translation Spot Selection

The issues of translation listed in section 1 are inherent disadvantages due to the principle of translation. Hence it's difficult to solve these problems by improving the mechanism. However, by choosing the appropriate location to perform translation, these problems can be solved or lightened, and translation can be more available. This draft calls the location to perform translation as "translation spot".

The basic idea of translation spot selection is to choose the place where the scalability and complexity is not a concern, i.e., the place where the translator is capable for its own translation traffic. Following this thought, a straightforward principle is to push translation down to edge networks. Since the volume of translation traffic in edge networks is relatively low, it's possible to achieve a real-time per-flow mapping and per-packet modification there. On the contrary, traffic in backbone is aggregated and hence much higher in volume. So routers in backbone would rather only support routing and forwarding than take charge of high-speed translation. However, when the total translation volume is low, it's easier to perform a unified translation in backbone than to distribute the job to many edge networks.

To achieve flexible translation spot selection, there's still a difficulty in packet forwarding: in a given topology, the IPv4-IPv6 border spot is fixed; If the translation spot isn't identical to the IP border spot, the packets can't be forwarded between the two spot due to IP diversity. See the example in Figure 1. The IP border is on spot 2 between IPvY backbone and IPvX Internet while the translation spot can be spot 1 or spot 2. If spot 1 is chosen, then packets from IPvY edge network are translated into IPvX on spot 1; they have to traverse to IPvY backbone to reach IPvX Internet. , and packets from IPvX Internet have to traverse the IPvY backbone to reach spot 1 for translation. Similar thing happens when spot 2 is chosen in Figure 2.

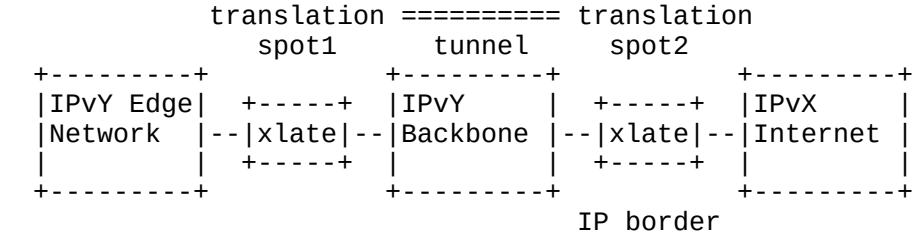


Figure 1 Translation Spot selection

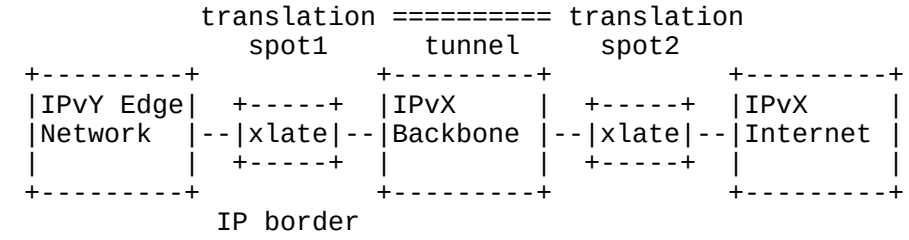


Figure 2 Translation Spot selection

This is actually a traversing problem and the typical solution is tunneling. By building a tunnel to connect IP border and the translation spot, the forwarding path can be achieved. In the example of Figure 1, an IPvX-over-IPvY tunnel between spot 1 and spot 2 can be used to forward translated-to-IPvX packets from spot 1 to spot 2, and to-be-translated IPvX packets from spot 2 to spot 1. In Figure 2, an IPvY-over-IPvx tunnel between spot 1 and spot 2 can be used to forward to-be-translated IPvY packets from spot 1 to spot 2, and translated-to-IPvY packet from spot 2 to spot 1. Although the flexible translation spot selection may require an extra tunnel, its cost is much lower than translation, and hence acceptable.

3. Translation Spot Selection in IPv4/IPv6-coexist Mesh

3.1. Scenario description

Translation spot selection can be used in many scenarios. As a demonstration this draft applies it to the mesh scenario described in Figure 3. In this scenario, an IPvX-only backbone is connected to both IPvX networks and IPvY networks. The backbone may also have entrance to IPvX and IPvY Internet. Besides native traffic and IPvY-over-IPvX softwire traffic described in [RFC4925], there're also traffics between IPvX and IPvY networks, between IPvX network and IPvY Internet, and between IPvY network and IPvX Internet. All these three types of traffics require translation, which should be performed on AFBRs (Address Family Border Router) or BRs (Border Router) on the border of the backbone.

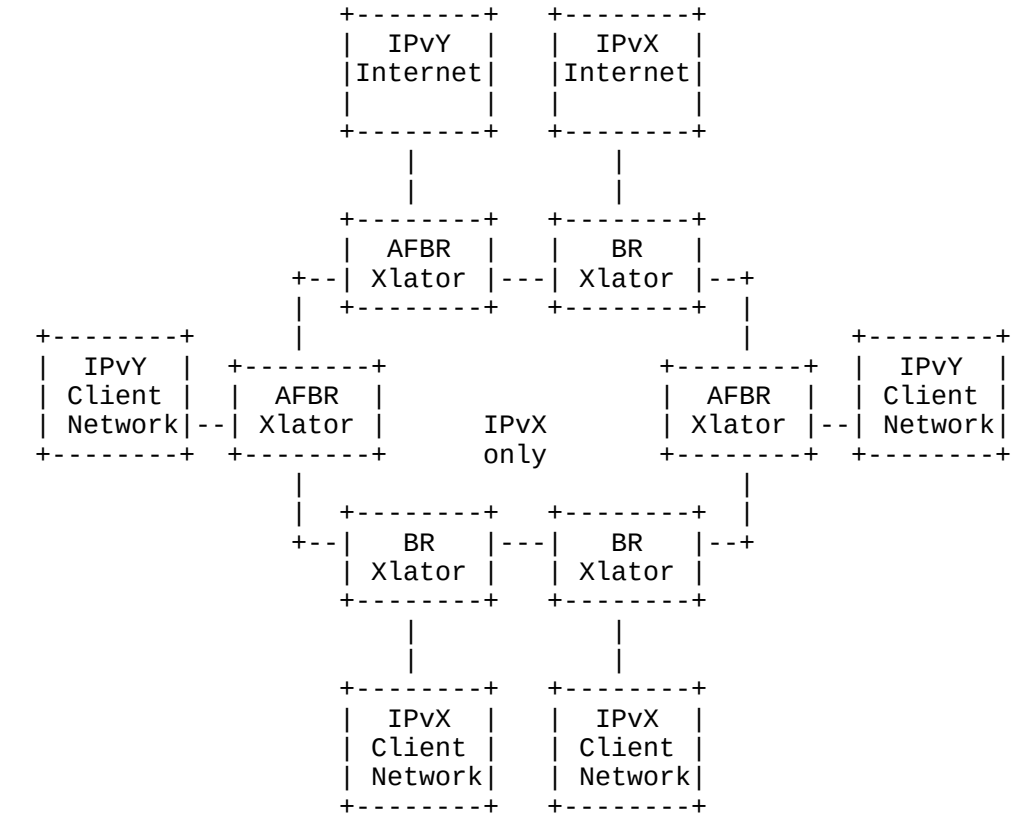


Figure 3 Translation Spot Selection in IPv4/IPv6-coexist Mesh

3.2. Translation between IPvX and IPvY networks

The communication between an IPvX network and an IPvY network follows the path "IPvX network - BR - IPvX backbone - AFBR - IPvY network". The translation can be performed either on the BR between IPvX network and backbone, or on the AFBR between IPvX backbone and IPvY network.

If the BR is chosen to be translation spot, a tunnel should be established for packet forwarding between the BR and the AFBR. Naturally it could be a softwire tunnel since it's a mesh scenario. Besides, to perform correct translation, BR needs the translation context delivered from the AFBR. This will be discussed in the next section.

3.3. Translation between IPvX network and IPvY Internet

The communication between an IPvX network and IPvY Internet follows the path "IPvX network - BR - IPvX backbone - AFBR - IPvY Internet". The translation spot can be either the BR between IPvX network and backbone, or the AFBR between IPvX backbone and IPvY Internet. BR can be chosen to avoid scalability and operation complexity issues, and AFBR can be chosen for unified translation purpose.

If the BR is chosen to be translation spot, a softwire tunnel should be established between the BR and the AFBR. Also BR needs the translation context delivered from the AFBR.

3.4. Translation between IPvY network and IPvX Internet

The communication between an IPvY network and IPvX Internet follows the path "IPvY network - AFBR - IPvX backbone - BR - IPvX Internet". The translation spot can be either the AFBR between IPvY network and IPvX backbone, or the BR between IPvX backbone and IPvX Internet. Usually the AFBR is preferred in this case, since it's the IP border and traffic is not so aggregated as in BR. However, BR can be chosen for unified translation purpose.

If the BR is chosen to be translation spot, a softwire tunnel should be established between the BR and the AFBR. Also BR needs the translation context delivered from the AFBR.

In all three types of translation-involved communication, translation spot selection is feasible. Yet an auto negotiation method is required to make the translation spot selection and translation context advertisement process more practical in the mesh scenario. This will be discussed in the next section.

4. Translation Spot Signaling

In the IPv4/IPv6-coexist mesh, the total number of client networks, and hence the total number of AFBRs and BRs could be quite high, so an auto negotiation method is required to select the translation spot for all translation-involved communications, rather than manual configuration on every AFBR and BR. This negotiation method is called translation spot signaling.

4.1. Signaling content

It’s clear that translation should be performed on an appropriate translator, or as in this scenario, an AFBR or BR device. Here the concept of Translation Preference (TP) is defined to represent the appropriateness of a device to perform translation. TP is a quantified value set by the administrator of the corresponding AFBR or BR device. By exchanging and comparing TP values, two translators can decided which one to be the translation spot.

The TP value should be decided by the administrator. The general criterion here is, the translator whose performance is better, whose traffic volume is lower, and the size of network behind which is smaller (thus the translation traffic is less aggregated), is preferred to do translation and should have a high value. TP can also be configured based on administrator’s policy, such as unified translation.

TPs for stateless and stateful translation are separated because they have different foundations (stateless translation requires IPv6 host to possess IPv4 address). In a mixed scenario, some translators can’t perform stateless translation like others because IPv6 hosts in its network don’t own IPv4 addresses.

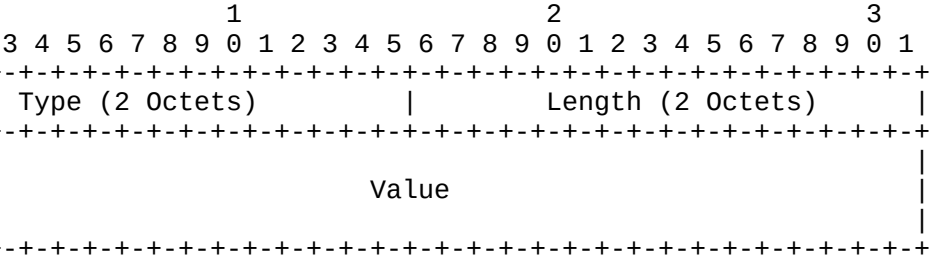
Besides TP, translation context should also be advertised through signaling. The translation context is the necessary knowledge to perform a translation. For stateless translation it’s the mapping prefix, and for stateful translation it’s the address pool used for address mapping. For example, in the type of "IPv6 network - BR - IPv6 Backbone - AFBR - IPv4 Internet" communication, if stateless translation is adopted, then AFBR should tell BR the prefix for IPv4-IPv6 address mapping when BR performs the translation; if stateful translation is adopted, then AFBR should tell BR the IPv4 addresses BR can use for address mapping when BR performs the translation.

4.2. Extensions in MP-BGP

MP-BGP is adopted to carry the translation spot signaling process since it fits the mesh scenario and is already used in softwire

mesh[RFC5565].

We define a new a new BGP Attribute, "Translation Information Attribute" to carry the TP and translation context information. It's an optional transitive attribute, and the attribute type code is TBD by IANA. The value field of this attribute is composed of a set of Type-Length-Value (TLV) encodings. The TLV is structured as follows. The Length field stands for the total number of octets in the Value field.



We define 4 TLVs here: Stateless_TP TLV, Stateful_TP TLV, IPv6_Prefix TLV and IPv4_pool TLV. More TLVs may be defined in the future when necessary.

- Stateless_TP TLV has the type field assigned to 1 and length field assigned to 2. The value field is filled with the 16bit TP value for stateless translation. High the TP value means high preference to perform translation.

- Stateful_TP TLV has the type field 2 and length field 2. The value field is filled with the 16bit TP value for stateful translation. High the TP value means high preference to perform translation.

- IPv6_Prefix TLV has the type field assigned to 3. The length field is variable. The value field is filled with the IPv6 prefix for address mapping in stateless translation, encoding in NLRI format[RFC4760].

- IPv4_pool TLV has the type field assigned to 4. The length field is variable. The value field is filled with the IPv4 pool for address mapping in stateful translation, encoding in NLRI format.

The AFBRs and BRs in the mesh should run MP-BGP process and peer with each other. When a new BGP session is established, AFBR and BR send a update containing the Translation Information Attribute to each other, which contains the Stateless_TP TLV or Stateful_TP TLV. Each router independently decides translation spot based on received TP

value. When the selected translation spot isn't the AFBR, then the AFBR should send another update with the Translation Information Attribute containing the IPv6_Prefix TLV or the IPv4_pool TLV to the BR. The tunnel-related routing should be triggered too, if there's any.

5. Further discussion

5.1. Achievement of translation spot selection

To be precise, through translation spot selection, we can solve the scalability problem of stateful translation and the operation complexity problem for both stateless and stateful translation. Also we make it more possible to perform application layer translation and adopt NAT46 mechanisms (NAT-PT) by pushing the translation spot down to the edge.

5.2. Cooperate with softwire

In the mesh scenario, softwire[RFC5565] is usually adopted as the tunnel mechanism. If it’s used to support forwarding between the BR and the AFBR, then after translation spot signaling, BR and AFBR should trigger the softwire routing process, in which AFBR should advertise the actual IPv4 prefixes, while BR should advertise to AFBR either the address pool assigned from the AFBR (stateful case), or the IPv4 address prefix containing the IPv4 address possessed by the IPv6 hosts (stateless case).

5.3. Using NAT64 or IVI as translation mechanism

NAT64[I-D.ietf-behave-v6v4-xlate-stateful] is a typical stateful translation mechanism. It can be used in the IPv4/IPv6-coexist mesh for translation-involved communications across the backbone. If AFBR is chosen to be the translation spot, then the traffic will follow a traditional NAT64 process; else BR is chosen to be the translation spot, then AFBR should divided its public IPv4 address pool and assigned one block to the BR through translation spot signaling. BR will perform the NAT64 translation using the assigned IPv4 address block. In softwire routing, BR should advertise this block to AFBR.

IVI[I-D.xli-behave-ivi] is a typical stateless translation mechanism. It can be used in the IPv4/IPv6-coexist mesh for translation-involved communications across the backbone. If AFBR is chosen to be the translation spot, then the traffic will follow a traditional IVI process; else BR is chosen to be the translation spot, then AFBR should inform BR the IVI prefix, then BR can learn the address mapping role and the IPv4 prefix possessed by its network. In softwire routing, BR should advertise this IPv4 prefix to AFBR.

6. IANA considerations

IANA is requested to assign a value from the "BGP Path Attributes" Registry, to be called "Translation Information Attribute", with this document as the reference.

7. Acknowledgements

The authors would like to thank Lixia Zhang, Eric Nordmark, Jari Arkko, Alain Durand and David Ward for their valuable comments on this draft.

8. References

8.1. Normative References

[RFC1702] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation over IPv4 networks", RFC 1702, October 1994.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.

[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.

[RFC2767] Tsuchiya, K., HIGUCHI, H., and Y. Atarashi, "Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)", RFC 2767, February 2000.

[RFC2893] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 2893, August 2000.

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

[RFC3338] Lee, S., Shin, M-K., Kim, Y-J., Nordmark, E., and A. Durand, "Dual Stack Hosts Using "Bump-in-the-API" (BIA)", RFC 3338, October 2002.

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

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

[RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

8.2. Informative References

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

[I-D.xli-behave-ivi]
Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", draft-xli-behave-ivi-07 (work in progress), January 2010.

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

Peng Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P. R. China

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