This article has been accepted for publication in IEEE Internet Computing but has not yet been fully edited.
Some content may change prior to final publication.

Tunnel-based IPv6 Transition

Yong Cui[#], Jiang Dong[*], Peng Wu[#], Jianping Wu[#], Chris Metz[&], Yiu L. Lee[§], Alain Durand[†]

[#]*Department of Computer Science and Technology, Tsinghua University, Beijing, China*
{cy, weapon}@csnet1.cs.tsinghua.edu.cn, jianping@cernet.edu.cn

[*] *State key Laboratory of Networking and Switching Technology, BUPT, Beijing, China*
bupt_dongjiang@bupt.edu.cn

[&]*Cisco Systems, Inc. San Jose, CA, USA*
chmetz@cisco.com

[§]*Comcast Cable Communications, Philadelphia, PA, USA*
Yiu_Lee@Cable.Comcast.com

[†]*Juniper Networks, Sunnyvale, CA, USA*
adurand@juniper.net

***Abstract*—IPv6 transition presents many challenges to the Internet community, and various solutions including dual stack, tunneling and translation have been proposed. Tunneling supports "like-to-like" IP connectivity across an "unlike" network while translation supports "like-to-unlike" IP inter-connectivity. But to date there is still not any overarching strategy that addresses all the scenarios. As tunneling can keep the end-to-end model that the Internet is built on, we develop a tunnel-based framework which solves the transition problems in the backbone and access networks with different tunneling mechanisms. We choose the softwire mesh mechanism to support dual-stack transport in a single-stack backbone, as well as the dual-stack lite and public 4over6 to provide IPv4 services in IPv6 access networks. 4rd is also introduced in IPv6 access networks as a stateless complement, while 6rd is selected to cover the case of IPv6 over IPv4 access networks.**

***Keywords*—IPv6 transition; next generation Internet; softwire; tunneling; 4over6**

I. Introduction

The transition from IPv4 to IPv6 has become an increasingly urgent problem. Many transition techniques have been proposed these years, such as NAT-PT [1], 6to4 [2]. However, vendors and ISPs found several serious problems when they tried to deploy these techniques, including routing scalability, weak performance and low transparency for upper layer applications. As a result, the Internet community has to obsolete some of these technologies (e.g., NAT-PT [3], 6to4 [4]) and reconsider more practical solutions to the transition problem.

The transition mechanisms can be generally divided into two categories: translation and tunneling. Translation is used to interconnect IPv4-only hosts with IPv6-only hosts by means of a translation vehicle converting IPv4 packets to IPv6 and vice-versa. However, the use of translation comes with several challenges. Translation introduces an intermediate element between IP end-points and thus breaks the end-to-end model. Moreover, proper translation between IP address families requires IP header rewriting, address (and port) converting, TCP/UDP checksum re-computing, and even application layer translation, etc. A translation vehicle that lacks bandwidth capacity will become a performance bottleneck when at high speeds.

Tunneling enables IPv6 connectivity across an IPv4 network, and vice-versa. The tunneling operations include encapsulation, decapsulation and tunnel endpoint signalling, while no upper layer operation is required. The network layer data forwarding can be implemented by the hardware with line speed capacity. Although tunneling cannot achieve the direct interworking between IPv4 and IPv6, we believe that broadly adopting tunneling as the foundation for IPv6 transition will accelerate IPv6 adoption, retain legacy IPv4 connectivity and enable operators to leverage their existing IPv4 assets during the transition period. The key notion is that tunneling retains the end-to-end notion and IP like-to-like affinity that the Internet is built on.

This paper proposes a tunnel-based IPv6 transition framework. Considering the diversity in structure and functionality, the framework applies different tunneling mechanisms to backbone and access networks. The backbone network needs to support both IPv4 and IPv6 connectivity for client networks even when the backbone is homogenous IPv4 or IPv6. We choose the softwire mesh mechanism to satisfy this requirement. Independent of the IP affiliation, the access networks must support a mix of legacy IPv4, dual stack and emerging IPv6-only hosts connecting to the Internet. 6rd [5] will be applied to provide IPv6 connectivity in IPv4 access networks, while public 4over6 [6], dual-stack lite [7] and 4rd [8] are selected to support IPv4 connectivity when the access network is IPv6.

II. Tunnel-based IPv6 Transition Framework

Portions of the Internet, including both network infrastructures and application services, will remain IPv4 for a long period during IPv6 transition. From the perspective of end users, the network should guarantee both IPv4 and IPv6 reachability regardless of the IP affiliation of the ISP network.

Digital Object Indentifier 10.1109/MIC.2012.63 1089-7801/$26.00 2011 IEEE

This article has been accepted for publication in IEEE Internet Computing but has not yet been fully edited.
Some content may change prior to final publication.

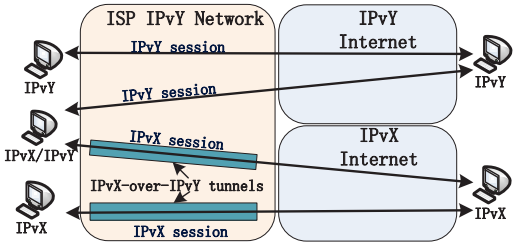

Fig.1 Communication modes of applications

For applications, transparency is mandatory for different IP protocol stacks.

As is shown in Figure 1, in an IPvY single-stack network (IPvX and IPvY represent the two different IP protocols), end users are usually dual-stack capable. They own native IPvY access, as well as IPvX access provided by the IPvX-in-IPvY tunnel. By combining the IPvX and IPvY access, the end hosts get their dual-stack reachability. Applications run on dual-stack capable hosts can employ IPvX or IPvY or both. For IPvY applications, they can communicate with IPvY peers with IPvY sessions directly. For IPvX applications, they can use IPvX sessions to interact with IPvX peers by IPvX-over-IPvY tunnels. As for dual-stack capable applications, they can choose between IPvX and IPvY stack freely, based on the peer's IP protocol. In brief, in single-stack networks, tunneling can satisfy the communication requirements during the transition. Tunneling techniques can be used in the Internet as much as possible, with the benefit of keeping the end-to-end model.

Figure 2 illustrates the tunnel-based IPv6 transition framework. In practice, the structure of an ISP network usually consists of two parts: backbone and access networks. The backbone networks are used to provide basic IP transport services for client networks, while the access networks are used to provide IP accesses to end users. The diversity in functionality naturally leads to the physical and logical separation, and subsequently makes each part relatively independent and easy to operate.

During the transition period, the backbone network should provide both IPv4 and IPv6 transport services to its clients, IPv4 and IPv6 networks alike. To achieve this goal with reasonable cost, an ISP can actually keep the internal routers of the backbone single-stack and only upgrade the edge routers of the backbone to dual stack. The magic here is to build tunnel mesh [9] between these edge routers to provide IPv4-over-IPv6 or IPv6-over-IPv4 transport services across the internal routers.

The access network is connected with the backbone on one side, and the end users on the other. It provides IP access to end users which could be either hosts or local networks behind CPEs. The access network has an aggregating feature: the data flows are concentrated on the user-side edge router from different subscribers, as well as on the backbone-side aggregation router from different edge routers. During the transition period, the access network should serve subscribers with both IPv4 and IPv6 accesses. However, for reasons of investment and management, access networks are usually single stack. So the ISP should upgrade the aggregation

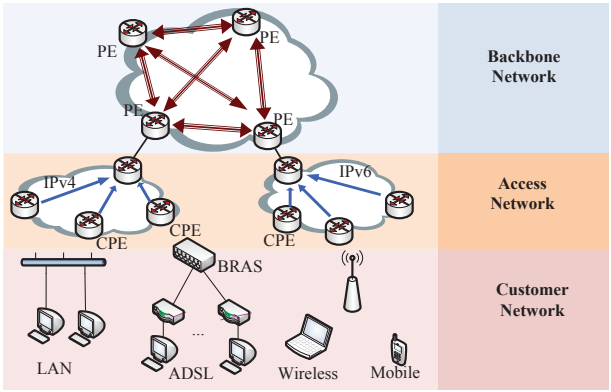

Fig.2 Tunnel-based IPv6 transition framework

routers to dual stack and provide heterogeneous accesses to end users by building hub and spoke tunnels [9].

There are several critical issues to be considered in tunnel-based IPv6 transition. The configuration procedure should be automatic and dynamic, so as to avoid management and provisioning complexity. In addition, to achieve correct encapsulation, a tunnel endpoint needs to keep track of the peer endpoint's address. Furthermore, to guarantee correct forwarding, the tunnel endpoint should learn the routing information from the other side of the tunnel.

Many existing tunneling techniques can be candidates to implement the transition framework. There are some basic principles to choose amongst them. The chosen techniques should have good scalability property, be simple in mechanism, and be easy for deployment and management.

III. Tunnel-based Transition in Backbone Network

An ISP is likely to run IPvX-only backbone network during the transition period. Nevertheless, there will be many IPvY networks connected to the backbone. With tunnel mesh between AFBRs, the backbone can forward IPvX packets directly, or encapsulate IPvY packets into IPvX on ingress AFBR and decapsulate them on egress AFBR. Ideally, the roadmap of the backbone migration would be as follow: pure IPv4, IPv6-over-IPv4, dual stack, IPv4-over-IPv6, and pure IPv6 at last. The tunnel mesh mechanism should not only function in both IPv6-over-IPv4 and IPv4-over-IPv6 phases, but also support automatic AFBR discovery and configuration. IPvY routing should be supported between AFBRs to achieve correct forwarding in IPvY networks and correct encapsulation on AFBRs.

Currently, several tunneling mechanisms can fit the backbone scenario, like manual configured tunnels, 6to4, 6PE, softwire mesh, etc. In manual configured tunnels, AFBRs are configured with encapsulation information in a static way. The main drawback is the low flexibility and overwhelming management load for ISPs, which have to frequently perform extensive manual configurations for each tunnel. Contrarily, 6to4 performs IPv6-over-IPv4 tunneling in an automatic way. However, in 6to4, each ISP has to use 2002::/16 as its network prefix, which is obviously against native IPv6 addressing and introduces routing scalability problems into the IPv6 Internet. 6PE [10] is customized to implement IPv6-over-IPv4 tunnel in MPLS-based IPv4 backbone network.

Digital Object Indentifier 10.1109/MIC.2012.63 1089-7801/$26.00 2011 IEEE

This article has been accepted for publication in IEEE Internet Computing but has not yet been fully edited.
Some content may change prior to final publication.

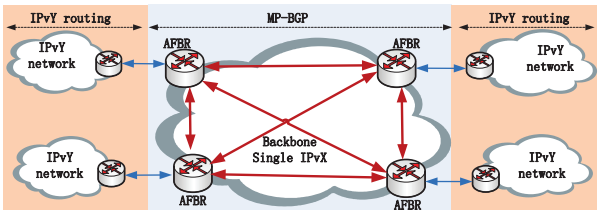

Fig.3 Softwire mesh framework

Softwire mesh [11]-[13] is a tunnel mesh mechanism which enables connectivity between IPvY islands across single-stack IPvX network. As shown in Figure 3, in the softwire mesh mechanism, AFBRs run extended multiprotocol-BGP (MP-BGP) to distribute tunnel parameters and advertise IPvY prefixes. All the AFBRs form a BGP mesh and a tunnel mesh with identical topology. By collecting the routes of IPvY prefix with IPvX next hop (remote AFBR IPvX address) each AFBR can build a prefix-level encapsulation table used for destination address lookup.

Softwire mesh builds and manages the tunnels automatically and dynamically, with IPvY routes as encapsulation basis. It is highly scalable in both control plane and data plane, and it is designed to interconnect large-scale networks. The control plane is built on BGP protocol, which is already activated on AFBRs of the backbone. The BGP mesh keeps the backbone free from all the IPvY routes, and the extra routing burden introduced to the AFBRs is not much different from the burden of a normal IPvY BGP router. To deploy softwire mesh, the upgrade happens only in AFBRs, which should support dual stack, along with extended MP-BGP and encapsulation/decapsulation. The rest of the backbone can stay unchanged, so the cost is quite acceptable. Comparing with other backbone tunneling techniques, softwire mesh is a more proper solution for backbone network transition.

IV. Tunnel-based Transition in IPv4 Access Network

Currently, most access networks are IPv4-only. Nevertheless, users of these networks desire to get connected with the IPv6 Internet. Therefore, ISPs should provide IPv6 access over IPv4-only networks, which could be achieved through IPv6-over-IPv4 tunnel. Since IPv6 address space is much larger than IPv4 address space, statelessness will be a natural property because IPv6 addresses can be generated by embedding IPv4 addresses.

There are several potential tunneling mechanisms in this scenario. 6over4 is a site local transition mechanism that builds IPv6 virtual link layer upon IPv4 multicast. As multicast is not universally well-supported, 6over4 cannot be widely used. ISATAP uses IPv4 as a virtual non-broadcast multi-access data link layer to connect dual-stack nodes over IPv4 networks. ISATAP has to take care of IPv6 LAN protocols, such as neighbour discovery beside the tunnel, which brings extra complexity. Teredo [14] can provide IPv6 connectivity to nodes located behind IPv4 NATs by tunneling IPv6 packets over IPv4 UDP. But Teredo cannot support users under symmetric NATs, and it cannot provide a fixed IPv6 address to its client, resulting in the fact that inbound access would fail if the IPv6 address is changed. IPv6 Tunnel Broker

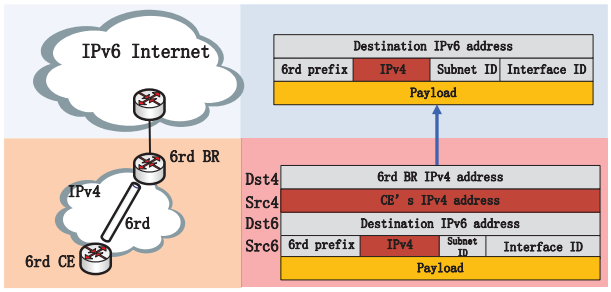

Fig.4 6rd

is another approach for isolated IPv6 hosts on IPv4 networks to connect to the IPv6 Internet. A tunnel broker client needs to negotiate with a tunnel broker first, which introduces extra procedures in the initial phase. The L2TP softwire [15] can support both IPv4-over-IPv6 and IPv6-over-IPv4 by using L2TPv2. Because the ISP needs to maintain L2TP session state for each tunnel, the load is increased in large scale.

6rd utilizes stateless IPv6-in-IPv4 encapsulation to provide IPv6 over IPv4-only access networks. A 6rd provider uses an ISP specific IPv6 prefix for stateless address mapping. All users' IPv6 addresses should be generated with the ISP's 6rd prefix followed by the IPv4 addresses of the Customer Edge (CE) devices (e.g., CPE or host). The users' addresses can aggregate at the 6rd Border Relay (BR), and be globally reachable through the 6rd BR. As shown in Figure 4, when an IPv6 packet from an end host arrives at the 6rd CE, it will be encapsulated by using the CE's IPv4 address as its source address and the BR's IPv4 address as its destination address. Arriving at the 6rd BR, this packet will be decapsulated and forwarded to the IPv6 Internet. When an IPv6 packet bound for the 6rd user reaches the BR, BR will then perform the IPv4 encapsulation, extracting the IPv4 address embedded in IPv6 destination address as the IPv4 encapsulation destination, and using the IPv4 address of the BR as the encapsulation source address. Then the packet will be forwarded to the correct CE in the IPv6-over-IPv4 tunnel and get decapsulated on the CE. In this way, 6rd does not require extra IPv4 routing.

Comparing with other IPv6-over-IPv4 mechanisms, 6rd does not require extra infrastructure support and extra negotiation, and no complexity is included except the tunnel itself. It is entirely stateless in both tunnel type and encapsulation destination mapping. As a result, the mechanism is very efficient and convenient to manage. We choose 6rd as the solution to the IPv4 access network scenario in the framework. However, 6rd cannot cover the corner case of legacy NAT CPE. In that case, we have to adopt Teredo or L2TP softwire to build tunnels with NAT traversing capability.

V. Tunnel-based Transition in New IPv6 Access Network

New access networks in the next few years will mainly be IPv6-only. To provide dual-stack access in IPv6-only access networks, we need an IPv4-over-IPv6 tunnel mechanism which can sustain the IPv4 availability when access networks switch to IPv6, and hence significantly advance the transition to IPv6. Early years of tunneling research in the Internet community focused on developing IPv6-over-IPv4 techniques.

Digital Object Indentifier 10.1109/MIC.2012.63 1089-7801/$26.00 2011 IEEE

This article has been accepted for publication in IEEE Internet Computing but has not yet been fully edited.
Some content may change prior to final publication.

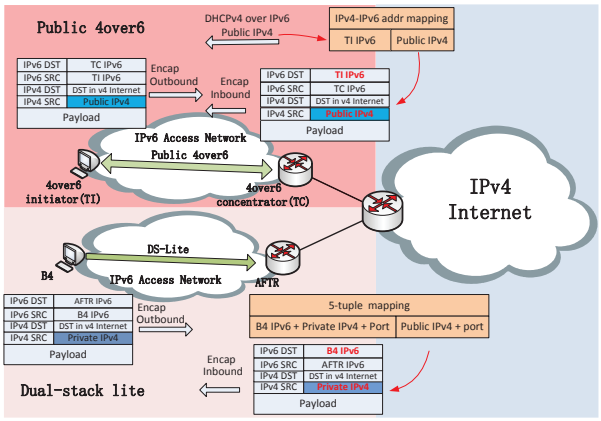

Fig.5 Public 4over6 and dual-stack lite

Recently, with the growing of IPv6 access networks, a series of IPv4-over-IPv6 tunneling techniques, including public 4over6, dual-stack lite and 4rd, are proposed to satisfy various transition demands.

The ISP can allocate IPv4 addresses to end hosts over IPv6 access networks. In this way, the ISP keeps the operating flexibility of IPv6 at the cost of AFBR maintaining per-host address mappings for encapsulation. Public 4over6, which provides IPv4 access to users over IPv6 networks with public IPv4 addresses, fits in this case. As shown on the top of Figure 5, when a subscriber needs IPv4 access, it will request a public IPv4 address from the 4over6 concentrator, and the concentrator will allocate a public IPv4 address to the subscriber, both via DHCPv4 over IPv6. When allocating an IPv4 address, the 4over6 concentrator keeps track of the mapping between the allocated IPv4 address and the subscriber's IPv6 address. The data forwarding procedure is a standard IPv4-in-IPv6 tunnel procedure, and the mapping will be used for destination address lookup during the encapsulation process on the concentrator. The mapping maintained in 4over6 concentrator is a per-subsriber address mappping. Therefore, the IPv4 service provided by public 4over6 is fully bi-directional.

The ISP can also maintain IPv4 addresses in AFBR in a centralized way. To achieve that, the AFBR will perform a NAT and maintain per-session states. Dual-Stack Lite introduces the carrier grade NAT (CGN) function on the Address Family Transition Router (AFTR). As shown at the bottom of Figure 5, hosts and CPEs (B4s) in IPv6 access networks use private IPv4 addresses for their IPv4 access with IPv4-over-IPv6 tunnels. The AFTR performs decapsulation and NAT translation. The AFTR maintains a 5-tuple mapping (i.e. B4 IPv6 address, private IPv4 address, port, public IPv4 address, port) for the translation of data flow. Note that Dual-Stack Lite introduces CGN to achieve address sharing for the issue of IPv4 address exhaustion. Anyhow, IPv4 public-private translation is still more lightweighted than IPv6-IPv4 translation because there is no inter-protocol behavior involved.

IPv4-over-IPv6 tunneling also can be implemented in a stateless way, as described in 4rd. The ISP can encode IPv4 address, or even port ranges, into IPv6 address for allocation. Then the encapsulation on AFBR can follow the encoding algorithm and turn stateless. The cost is coupling IPv4 and IPv6 addressing, which will increase the operating difficulty.

4over6 tunneling techniques	State maintenance	IPv4 address Multiplexing	IPv4 Address allocation	IPv4/IPv6 Address Coupling
Public 4over6	Per user state	No multiplex	Address allocation to user side	No
DS-lite	Per session state	NAT	Address management in CGN	No
4rd	stateless	Port range	Address allocation to user side	Yes

Table 1 Comparisons of IPv4-over-IPv6 techniques in access network

Comparisons among these IPv4-over-IPv6 techniques are listed in Table 1. ISPs can choose one or several proper mechanisms based on their requirements. If an ISP has enough IPv4 address resources, it can adopt either public 4over6 or 4rd to provide IPv4 service in IPv6 networks, while public 4over6 is implemented in a stateful way and 4rd is managed in stateless one at the cost of coupling the IPv4 and IPv6 addressing and routing. If ISPs want to multiplex IPv4 addresses to improve the utilization of IPv4 resources, then they can adopt either 4rd or dual-stack lite. In the latter case, public 4over6 can be integrated to provide bi-directional IPv4 service to advanced users like application servers. These mechanisms can preserve the IPv4 availability when access networks switch to IPv6, and hence significantly prompt the transition to IPv6.

VI. Implementation Consideration

The transition mechanisms used to compose the tunnel-based IPv6 transition framework have been widely supported by IETF, vendors and ISPs. In IETF, the documentations for softwire mesh, 6rd and dual-stack lite are already standard track RFCs; Public 4over6 has been accepted as a Softwire working group draft; 4rd draft has also been widely supported in the working group.

Vendors are active in implementing these transition mechanisms. To our knowledge, currently Cisco already has routers with 6rd functionality; Juniper already has dual-stack lite productions; Huawei and Bitway have developed routers that are capable of executing softwire mesh.

As to deployment, China's Next Generation Internet (CNGI) has begun a commercial trial of softwire mesh deployment in 100 campus networks since 2008. Comcast announced a trial in dual-stack lite, and China Telecom is active in testing and deploying dual-stack lite and public 4over6. Softbank in Japan has deployed 4rd recently.

Acknowledgments

This work is supported by National Major Basic Research Program of China (no. 2009CB320500) and NSFC project (no. 61120106008, 61140454).

Digital Object Indentifier 10.1109/MIC.2012.63 1089-7801/$26.00 2011 IEEE

This article has been accepted for publication in IEEE Internet Computing but has not yet been fully edited.
Some content may change prior to final publication.

REFERENCES

[1] G. Tsirtsis, P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)," IETF RFC2766. February 2000.
[2] B. Carpenter, K. Moore, "Connection of IPv6 Domains via IPv4 Clouds," IETF RFC3056, February 2001.
[3] C. Aoun, E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status," IETF RFC4966, July 2007.
[4] O. Troan, "Request to move Connection of IPv6 Domains via IPv4 Clouds (6to4) to Historic status", IETF draft, June 2011.
[5] W. Townsley et al., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)," IETF RFC5969, August 2010.
[6] Y. Cui et al., "Public IPv4 over Access IPv6 network," IETF draft, September 2011.
[7] A. Durand et al., "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion," IETF RFC6333, August 2011.
[8] T. Murakami, et al., "IPv4 Residual Deployment on IPv6 infrastructure", IETF draft, September, 2011.
[9] X. Li et al., "Softwire Problem Statement," IETF RFC4925, July 2007.
[10] J. De Clercq, et al., "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)," IETF RFC4798, February 2007.
[11] J. Wu et al., "Softwire Mesh Framework," IETF RFC5565, June 2009.
[12] J. Wu et al., "The Transition to IPv6, Part I: 4over6 for the China Education and Research Network," IEEE Internet Computing, May 2006.
[13] Y. Cui et al., "The Transition to IPv6, Part II: The Softwire Mesh Framework Solution," IEEE Internet Computing, Sept/Oct 2006.
[14] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", IETF RFC4380, February 2006.
[15] B. Storer et al., "Softwire Hub and Spoke Deployment Framework with L2TPv2", IETF RFC5571, June 2009.

Yong Cui is an associate professor at Tsinghua University, Beijing. He has published 2 IETF RFCs on IPv6 transition technologies, and he co-chairs the IETF Softwire Working Group which focuses on tunneling technology for IPv6 transition.

Jianping Wu is a full professor at Tsinghua University. As the pioneer of IPv6, he built the largest native IPv6 backbone in the world as the China Education and Research Network II and received the Jonathan B. Postel Award from the Internet Society in 2010. He co-authored 4 IETF RFCs on IPv6.

Chris Metz is a technical leader in the Routing Technology Group for Cisco Systems. He is a specialist in IPv6 transition and co-authored 4 IETF RFCs.

Yiu L. Lee works in the CTO office at Comcast. He contributes 8 IETF drafts on IPv6 transition technology.

Alain Durand is a director in the IPG/CTO group at Juniper Networks. He is also the co-chair of IETF IPv6 transition WG, Softwire, and has published 14 RFCs.

Digital Object Indentifier 10.1109/MIC.2012.63 1089-7801/$26.00 2011 IEEE