State Management in IPv4 to IPv6 Transition

Yong Cui, Yuchi Chen, Jiangchuan Liu, Yiu-leung Lee, Jianping Wu, and Xingwei Wang

Abstract

The transition of the Internet from IPv4 to IPv6 is urgent and inevitable. A series of IPv6 transition solutions have been proposed by the Internet Engineering Task Force; yet most of them have not seen success in real world, and some were even obsoleted. Nowadays IPv6 transition solutions are still continuously being worked out. The major difference among these solutions is the state management, which is an essential function and has a profound influence on many aspects of networking, such as addressing, provisioning, and network performance. In this article, we present a comprehensive survey on IPv6 transition solutions from the perspective of state management. We first briefly review the basic rationale of IPv6 transition solutions, highlighting the necessity of state management. Then we study various types of state management adopted by typical IPv6 transition solutions, focusing on their impacts on aspects of networks. Based on the above analyses, we summarize the applicability of different types of state management, and discuss their state-of-the-art application directions that may lead to potential future research directions in the IPv6 transition process.

In February 2011, the Internet Assigned Numbers Authority (IANA) announced that global the IPv4 address pool had run out. The Internet community acknowledges that IPv6, the next generation of IP standardized by the Internet Engineering Task Force (IETF), is the most promising solution to IP address exhaustion. Since the late 1990s, IETF has published a series of solutions to promote the Internet transition from IPv4 to IPv6. Few of them, however, have seen success in the real world, and some have even been obsoleted by IETF [1].

Nowadays IPv6 transition is still one of the most important topics in IETF with solutions continuously being proposed to promote the IPv6 transition process. State management is the major difference among these solutions.

The *state* in IPv6 transition is the binding between IPv4 and IPv6 protocol semantics. State management has an important influence on the development of IPv6 transition solutions. The well-known NAT-PT [2] was one of the best candidate solutions, but it was finally obsoleted mainly because of its complicated cross-layer state management [1]. The application of another well-known solution named 6to4 was also restricted due to routing issues caused by improper state management [3]. In early years, Per-prefix State was adopted in solutions for the backbone network. Recently the research focus has shifted to the delicate management in access network. The solution with per-flow state was published [4], but its heavy cost pushed people to work on more lightweight state management such as per-subscriber state [5, 6] and stateless mapping [7]. Their protocol designs have been discussed a lot in IETF [4–7], but there is no systematic study on state management yet.

In this article, we present a comprehensive survey on IPv6 transition solutions from the perspective of state management. We first give a brief review on the basic rationale of IPv6 transition solutions, highlighting the necessity of state management. Then we study various types of state management adopted by several typical IPv6 transition solutions. By discussing the impacts of various types of state management on aspects of a network, we point out their causal relationships with major advantages and flaws of solutions. Based on the above analyses, we summarize the applicability of state management in various solutions, and discuss the state-of-the-art directions of their application, which may lead to potential future research directions in the IPv6 transition process.

The remainder of the article is organized as follows. The following section introduces the basic rationales of IPv6 transition solutions as a background. The third section studies various types of state management adopted by typical IPv6 transition solutions. Then we summarize the application directions of state management. The final section concludes the article.

States in IPV6 Transition

As a straightforward IPv6 transition solution, deploying a dual-stack network is popular among many Internet service providers (ISPs). However, this solution cannot really support interoperation between IPv4 and IPv6 networks [8]. In addition, managing a dual-stack network is complicated and expensive. Hence, the research community is focusing on other solutions that can support the interoperation between two heterogeneous protocols. Such solutions can be broadly classified into two categories, *translation* and *tunneling*. In both of

Yong Cui and Jianping Wu are with Tsinghua University.

Yuchi Chen is with Tsinghua University and Beijing University of Posts and Telecommunications.

Jiangchuan Liu is with Simon Fraser University.

Yiu-leung Lee is with Comcast Corporation.

Xingwei Wang is with Northeastern University.

0890-8044/15/$25.00 © 2015 IEEE

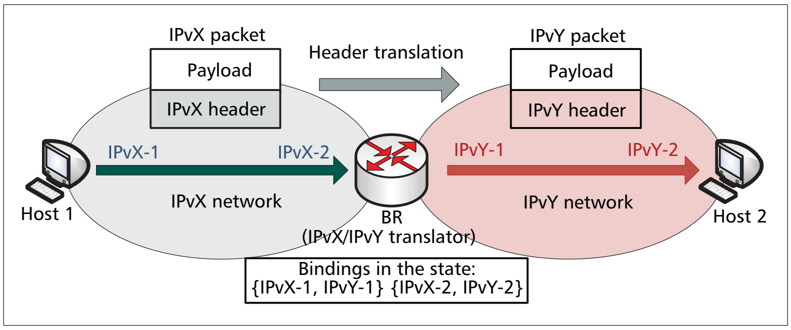

Figure 1. The BR replaces the IPvX header with an IPvY header before forwarding the packet to the IPvY network, according to the bindings of IPvX and IPvY addresses kept in the state.

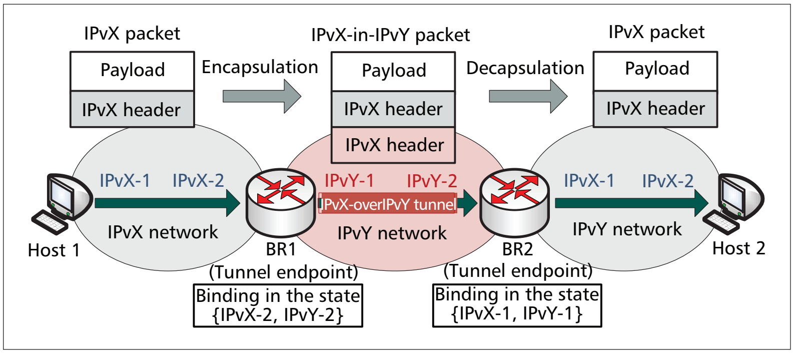

Figure 2. When BR1 receives an IPvX packet, it encapsulates the packet with an IPvY header and delivers it to BR2 through an IPvX-over-IPvY tunnel. When BR2 receives the packet from the tunnel, it decapsulates the outer IPvY header and forwards the packet to Host2.

them, state is essential because it keeps necessary information including binding between IPv4 and IPv6 addresses, transport-layer protocol and IDs (i.e., TCP/UDP ports or Internet Configuration Message Protocol IDs), and so on.

States in Translation

Translation, which converts between IPv4 and IPv6 protocol semantics, allows hosts of different IP versions to communicate with each other [9]. As illustrated in Fig. 1, the translation process converts the IPvX header into an IPvY header when it receives an IPvX packet destined to an IPvY network, where X or Y represents the respective IP version. The translation is generally performed in a *border router* (BR), also called an IPvX/IPvY *translator*, located between the two networks.

Suppose that Host1 initiates the communication with Host2. Since Host1 does not "speak" IPvY, the BR must translate the IPvY-2 to an IPvX address (IPvX-2) that Host1 can understand and process. Similarly, the BR must also translate the IPvX-1 to an IPvY address (IPvY-1) that Host2 can use to reply back to IPvX-1.

As such, the BR must create and maintain a state that keeps the binding between the original addresses (IPvX-1 and IPvY-2) and the translated addresses (IPvY-1 and IPvX-2). Additional information such as the type of transport layer protocol and the transport layer IDs of the packet may need to be kept for the translation. All these states are configured statically or generated algorithmically within a *translation table*, an information base that is associated with the *routing information base* (RIB) of the BR.

States in Tunneling

Tunneling allows communications between two IPvX hosts traversing IPvY networks, as shown in Fig. 2. Tunneling encapsulates the entire IPvX packet into an IPvY packet and delivers the packet over an IPvY network [9]. Such a solution is called IPvX-over-IPvY tunneling. BRs are deployed in the edges of the networks that run different IP versions in order to establish a tunnel for forwarding packets. For this reason, they are also referred to as *tunnel endpoints*.

Suppose that Host1 initiates a communication with Host2. As BR1 is both the gateway of Host1 and the tunnel endpoint, it should deliver the IPvX packet through an IPvX-over-IPvY tunnel to BR2. Hence, BR1 must maintain the state expressing that the next hop of IPvX-2 is BR2's IPvY address (IPvY-2). Similarly, the state that expresses that the next hop of IPvX-1 is BR1's IPvY address (IPvY-1) must be managed by BR2. These states are generally maintained as entries in a binding table, which is (or is combined with) the RIB in the BR.

Influences of States

State management plays an important role in both translation and tunneling. It has significant influence on many aspects of a network. The binding of heterogeneous addresses in states determines the overall addressing, routing, and provisioning methods. The amount of entries determines the spatial cost of state storage and logging. This amount is also proportionally relative to the temporal overhead of state lookup, which in turn affects the performance of packet processing and forwarding. In addition, if states are dynamically established and managed by different entities, the periodic synchronization between states is needed for failover. The amount of entries in states will proportionally determine the temporal and spatial cost of such synchronization.

State Management in IPv6 Transition Solutions

In IPv6 transition, state can be classified into four categories, including *per-flow state*, *per-subscriber state*, *per-prefix state*, and *stateless mapping*. In this section, we introduce their basic rationales and technical features, discuss their influence on diverse aspects of networking, and highlight their pros and cons in the context of typical IPv6 transition solutions.

Per-Flow State

A *flow* is a series of packets that have the same pairs of source and destination addresses, as well as the same transport layer IDs and protocol. It is typically represented by a 5-tuple binding: {*source address*, *destination address*, *source port*, *destination port*, *protocol*}. A *per-flow state* keeps this binding information of flows.

Per-flow state management is usually performed by a *network address and port translator* (NAPT). By keeping bindings between addresses and transport-layer IDs of flows, NAPT can dynamically provision IPv4 address to outbound flows, and share one IPv4 address among multiple flows.

Per-Flow State in Translation — Network Address Translation-Protocol Translation (NAT-PT) [2] is an early translation solution, which was intended to be an integrated solution to support bidirectional communication between IPv4 and IPv6 hosts.

A NAT-PT translator resides between the IPv4 and IPv6 hosts, acting as their common gateway. In order to translate between IPv4 and IPv6 sessions, the NAT-PT translator adopts per-flow state to keep the binding between IPv4 and IPv6 flows. A DNS application layer gateway (DNS-ALG) is co-located with the NAT-PT translator, responding to DNS queries from both IPv4 and IPv6 hosts. The DNS-ALG manages a DNS state that keeps the binding between the *fully qualified domain name* (FQDN) and the IPv4 or IPv6 addresses of the hosts.

NAT-PT requires the per-flow state and the DNS state to be combined. As such, the application-layer DNS state should synchronize to the network layer per-flow state, involving complicated cross-layer state management. This synchronization becomes a significant bottleneck to the performance of packet processing, especially when the traffic comes from the IPv4 side. After weighing the overheads and benefits, IETF eventually decided to abandon this solution [1]. Nevertheless, it was realized that the solution is still feasible for communications initiated from the IPv6 side. With the improvement in state management, NAT64 [10] was proposed as the solution supporting IPv6 hosts visiting IPv4 sites.

In order to translate between IPv4 and IPv6 flows, a NAT64 translator still applies per-flow state management. The difference is that the DNS-ALG is removed from the NAT64 translator and becomes a dedicated DNS server called DNS64. DNS64 only handles DNS queries from the IPv6 host. When receiving a DNS query, DNS64 delivers it to a remote DNS server. When a response from a remote server arrives, DNS64 simply composes the IPv6 address by adding a specific prefix in front of the IPv4 address carried in the response, and returns the IPv6 address as the final response to the DNS query. Using this method, DNS64 does not have to maintain any state. Since the DNS state is unloaded from the per-flow state, the state synchronization issue will not occur.

Per-Flow State in Tunneling — Dual-Stack Lite (DS-Lite) [4] is a stateful IPv4-over-IPv6 tunneling solution. It is proposed as a straightforward solution for rapid IPv6 network deployment by leveraging *carrier grade NAT* (CGN), which is widely used in existing ISP networks.

DS-Lite adopts per-flow state, which is compatible with the flow state management in CGN. The flow state of CGN is extended to include the IPv6 address in the form of {*private IPv4 address*, *public IPv4 address*, *internal port*, *external port*, *IPv6 address*}. As Per-flow State manages IPv4 addresses intensively, no public IPv4 address will be provisioned to the hosts or *Customer Premises Equipments* (CPEs). Hence no IPv4 provisioning protocol like *Dynamic Host Configuration Protocol* (DHCP) is required by DS-Lite.

In general, per-flow state introduces intensive address management. When applying per-flow state, private IPv4 addresses can be provisioned to the customer side, which will significantly save IPv4 addresses. Unfortunately, the number of entries in per-flow state is proportional to the amount of transiting flows, bringing heavy cost of state storage, packet processing, and logging. In addition, per-flow state cannot help preserve the end-to-end transparency because it hides the address of internal endpoint and restricts its reachability from outside the domain. Furthermore, flows belonging to the same session may create redundant bindings if they appear in different time periods. This issue can increase the burden on state management and logging.

Per-Subscriber State

Per-subscriber state is an extension of the IPv4 address binding state maintained by a traditional *Network Address Translator* (NAT). Unlike what per-flow state does, *per-subscriber state* keeps a binding between the IPv4 and IPv6 addresses of the endpoint (i.e., the *subscriber*). Per-subscriber state usually requires provisioning both global IPv4 and IPv6 addresses to subscribers before building up the binding states. Therefore, solutions adopting per-subscriber state should also apply provisioning protocols, such as Dynamic Host Configuration Protocol (DHCP) and DHCPv6.

Per-subscriber state is dynamic in nature. Furthermore, per-subscriber state can significantly reduce the number of necessary bindings compared to per-flow state, thus subsequently reducing the cost of management. It can also preserve the end-to-end transparency: the subscriber is reachable from outside the domain as long as the binding of its addresses lives in the per-subscriber state. Since per-subscriber state requires provisioning addresses to subscribers, the address utilization efficiency may be lower than that with per-flow state.

Per-subscriber states are used in a series of tunneling solutions. Public 4over6 [5] describes the framework of provisioning a full IPv4 address to the subscriber. Lightweight 4over6 [6] further extends Public 4over6 by allocating *port-restricted* IPv4 addresses with available port sets to subscribers. In Lightweight 4over6, the binding in per-subscriber state is {*IPv4 address*, *available port sets*, *IPv6 address*}. By including port sets in the binding, per-subscriber state can offer higher address utilization efficiency.

Per-Prefix State

The *per-prefix state* is mainly proposed for routers in the core network. These routers, including the core routers within the core network and border routers located at the edge of the core network, establish their RIB by exchanging routing information with each other using route protocols like *Border Gateway Protocol* (BGP). The idea of per-prefix state is to leverage the RIB state of border routers. Per-prefix state manages the binding between the address of the border router, and the network prefix of the customer network that takes the border router as its gateway. These bindings are no more than the entries in the RIB of the border router. As such, the number of bindings in per-prefix state is even smaller than that of per-subscriber state.

Softwire mesh [11] is a tunneling solution that adopts per-prefix state management. In softwire mesh, each customer network that runs *external IP* (E-IP) connects to the common core network that runs internal IP (I-IP) by a border router called the *address family border router* (AFBR). Every AFBR must manage the state that keeps the binding between each other's E-IP prefix and I-IP address.

Per-prefix state brings up no inherent dependence between E-IP and I-IP prefixes. Therefore, softwire mesh is very scalable and particularly suitable for backbone networks. Nevertheless, per-prefix state requires that the I-IP route protocol used by AFBRs must support carrying E-IP routing information. Due to this requirement, the network must apply an extendable route protocol, such as *Multiprotocol Extension for BGP* (MP-BGP) [12]. In addition, since per-prefix state maintains no information of any individual E-IP, it does not support E-IP provisioning to subscribers. In this case, other types of state management such as per-subscriber state should be applied.

Stateless Mapping

In *stateless mapping*, the IPv4 part of a state is algorithmically embedded in the IPv6 part. The IPv4 part can be extracted from the IPv6 part of state when needed. In most cases, this means that the IPv4 address or prefix, as well as transport layer IDs, are encoded in the IPv6 address or prefix. Figure 3 shows some examples of stateless mapping rules that are applied to perform such encoding.

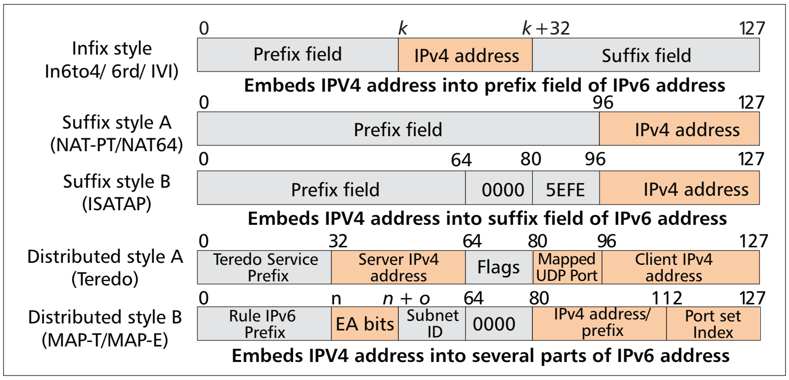

Figure 3. Examples of stateless mapping rules.

In per-flow state, per-subscriber state, and per-prefix state, the IPv4 and IPv6 part of state are independent from each other in nature. In stateless mapping, the two parts of state are coupled together. This makes the major difference between stateless mapping and the other three types of states.

Stateless Mapping in Translation — The Stateless IP/ICMP Translation Algorithm (SIIT) was early proposed as a straightforward algorithm to derive an IPv6 address from an IPv4 address. The basic idea of SIIT is to syndicate an IPv6 address by adding two dedicated IPv6 prefixes, that is, *::FFFF:0:0:0/96* and *::FFFF:0:0/96*, in front of the IPv4 addresses of internal and external nodes. SIIT lays the basic principle of addressing in stateless mapping, that is, an IPv4 address should be encoded into an IPv6 address. The two specified /96 prefixes, however, are too long to compose the IPv6 prefix of a SIIT domain. Such long IPv6 prefixes can significantly impact the scalability of global IPv6 RIB, making it too large to handle.

IVI is a translation solution that improves SIIT by using the variable *network-specific prefix* (NSP) to take the place of the two well-known prefixes. Since stateless mapping in IVI inherently requires one-to-one coupling of IPv4 and IPv6 addresses, the IPv6 addressing is quite limited by that of IPv4. Nevertheless, stateless mapping makes stateless double translation solutions possible. Double translation performs two translation processes to make IPvX packets traverse an IPvY network. Take mapping of address and port-translation (MAP-T) [13] as an example. Similar to IPv4-over-IPv6 tunneling, MAP-T supports IPv4 traffic traversing an IPv6 network. An IPv4 packet that is going to traverse an IPv6 network is converted to an IPv6 packet when it enters the IPv6 network. When it leaves the IPv6 network, this packet will be converted back to the original IPv4 packet. Because MAP-T applies stateless mapping to keep the binding between IPv4 and IPv6 addresses, both of these two translation processes can be performed straightforwardly without any information exchange between translators.

Similar to Lightweight 4over6, MAP-T adopts the port-restricted address provisioning method. In other words, IPv4 addresses are allocated to the customer side, each of which is bounded with an available port set. MAP-T includes the port set into stateless mapping. This port set is presented as a *Port Set ID* (PSID) field in the IPv6 address. As such, the one-to-one address coupling issue faced by IVI is mitigated.

Stateless Mapping in Tunneling — 6to4 is designed to support communications between isolated IPv6 networks across the IPv4 Internet in between. A *6to4 router* is deployed at the entrance of the IPv6 network and act as a tunnel endpoint. In order to achieve high transport efficiency, 6to4 adopts stateless mapping to keep the binding between the IPv4 and IPv6 addresses of 6to4 routers. The IPv4 address of 6to4 router *IPv4ADDR* is combined with a well-known prefix 2002::/16, forming an IPv6 prefix like *2002:IPv4ADDR::/48*. Since the IPv4 address is included in the IPv6 prefix, the IPv6 routing information is coupled with IPv4 routing information. In addition, the IPv6 addresses of 6to4 routers are unable to aggregate, subsequently challenging the scalability of global IPv6 RIB.

With a slight improvement of 6to4, 6rd [14] is proposed, which also adopts stateless mapping, but specifies its application within a single ISP domain. Instead of the well-known prefix, 6rd uses a *Network-Specific Prefix* (NSP) to compose the IPv6 address. This design avoids the routing scalability issue brought by 6to4.

Besides IPv6-over-IPv4 tunneling, there is also an IPv4-over-IPv6 tunneling solution called mapping of address and port encapsulation (MAP-E) [7]. MAP-E applies the same stateless mapping as in MAP-T. The IPv4 address and available port set are encoded in the IPv6 address. MAP-E supports sharing IPv4 addresses among multiple units of customer premises equipment (CPE), but still faces the challenge of addressing flexibility brought by stateless mapping.

Generally speaking, there is barely any explicit cost of state management in stateless mapping. It can help achieve high packet processing performance because no cost of state lookup occurs. Stateless mapping requires coupling IPv4 and IPv6 addresses, however, which brings up a significant challenge to addressing. This coupling may also require coupling IPv4 and IPv6 routing information, which is commonly considered to be a suboptimal design.

Application Directions of State Management

Although each type of state management is applicable in both translation and tunneling in theory, in reality they are adopted only in certain scenarios and solutions. Table 1 summarizes the features and applicability of different types of state management. From our observation, two directions are the mainstreams of application of state management in the research community: per-subscriber state and stateless mapping in the access network, and per-prefix state in the backbone network.

State Management in the Access Network

The access network provides Internet access to subscribers [15]. Due to the IPv4 address shortage, many ISPs prefer to allocate private rather than public IPv4 addresses to subscribers, making carrier grade NAT (CGN) an essential part of the IPv4 access network. It seems reasonable to leverage this CGN when deploying IPv6 networks. DS-Lite, as an example, follows this idea and leads to the application of per-flow state. Since the number of flows may grow quite large in the access network, the cost of logging and state management ise fairly prohibitive in this case.

The research community acknowledges the challenge brought by per-flow state, and accepts that leveraging the subscriber state, which is maintained in the provisioning system and manages the subscribers' addresses, seems to be more reasonable. Both per-subscriber state and stateless mapping are compatible with the subscriber state by nature. Hence, they are the mainstream in application of state management in the access network.

IETF is currently standardizing solutions that apply these types of state management, such as Lightweight 4over6, MAP-T, and MAP-E. Many ISPs and vendors are promoting the commercial use of these solutions. China Telecom has been

Types of state management	Per-flow	Per-subscriber	Per-prefix	Stateless mapping
Applicable mechanism	Translation or tunneling	Tunneling	Tunneling	Translation or tunneling
Applicable scenario	Access network	Access network	Backbone network	Access network
Typical applicable solutions	NAT64 [10] DS-Lite [4]	Public 4over6 [5] Lightweight 4over6 [6]	Softwire mesh [11]	MAP-T [13] 6to4/ 6rd [14] MAP-E [7]
State management location	Border relay (BR)	BR	Provider edge (PE)	BR, CPE, end host
Provisioning method	Dynamic address and port allocation per flow	Dynamic address (and port set) allocation per subscriber	Dynamic prefix learning through route protocols	Static parameters assignment for computing address (and port set)
Advantages	Centralized address management helps achieve higher utilization ratio of addresses.	Smaller in scale than per-flow state. Raises lower cost of management.	Even lower cost of state management than that of the per-subscriber state.	No explicit cost of state management occurs. Can provide high packet processing performance.
Flaws	Heavy cost of management. Cannot help preserve end-to-end communication transparency.	Efficiency of address utilization is not as high as in per-flow state.	Extensible routing protocol must be applied. Cannot support provisioning addresses to subscribers.	Addressing flexibility is not as high as in other types of state management. Raises potential routing scalability issue.

Table 1. Features and applicability of state management.

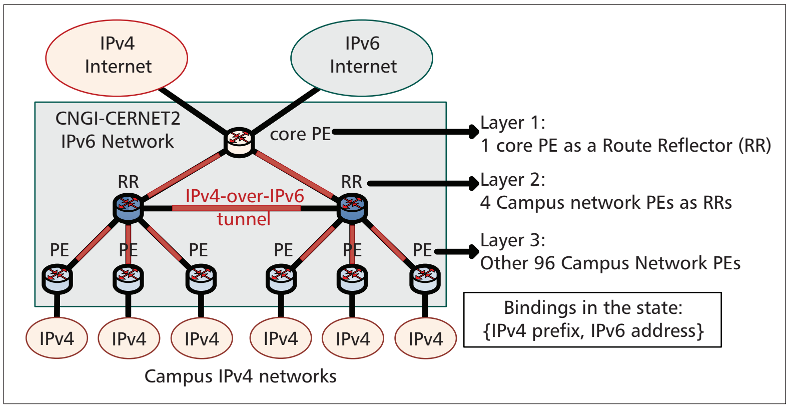

Figure 4. 4over6 Mesh Deployment on CNGI-CERNET2.

running a Lightweight 4over6 field trial in Hunan Province since late 2011. Deutsche Telekom and Hrvatski Telekom also started a Lightweight 4over6 field test within their TeraStream Project in Croatia in April 2013. In Canada and Brazil, testing of MAP-T has also been performed by Rogers Communications Inc. and Network Information Center (NIC). In Japan, an interoperability test on MAP-E was carried out by the Japan Network Operators Group (JANOG) in 2012.

State Management in the Backbone Network

The backbone network connects ISPs' access networks as well as the Internet [15]. The transport performance and robustness, which are based on efficient routing convergence, are the most mission-critical properties of the backbone network. 6to4 was proposed as an IPv6 transition solution for backbone networks. Stateless mapping can help improve the transport performance, but requires coupling of IPv4 and IPv6 routing information, which is fairly harmful to network routing convergence.

The research community has acknowledged the issues with stateless mapping, and decided to leverage the existing RIB state. Per-prefix state was proposed as an extension to such a RIB state. It can keep the independent nature of IPv4 and IPv6 routing information while bringing up no scalability issues. The usage of per-prefix state is performed in the field trial of softwire mesh for an IPv4-over-IPv6 scenario (4over6 Mesh) [12] on the China Education and Research Network II of the China Next Generation Internet (CNGI-CERNET2), one of the largest IPv6 backbone networks in the world.

There are three layers in the architecture of this 4over6 Mesh deployment, as shown in Fig. 4. The first layer includes a core *provider edge* (PE) router, which is the gateway of the CNGI-CERNET2, and a *route reflector* (RR), which learns and advertises the routing information of all the IPv4 campus networks to other PEs through the MP-BGP. The second layer consists of four campus network PEs, which also act as RRs. The third layer includes the other 96 campus network PEs, which connect the campus IPv4 network and the CNGI-CERNET2. Each PE manages a per-prefix state that keeps the binding between the IPv4 prefixes of other campus networks and the IPv6 addresses of other PEs. The trial result shows this per-prefix state works well in this deployment. The campus networks form a mesh model, by which any two IPv4 campus networks can directly communicate with each other through an IPv4-over-IPv6 tunnel.

Conclusion

In this article, we survey the IPv6 transition solutions from the perspective of state management. We discuss how state impacts on aspects of network, such as addressing, provisioning, and performance. Based on the above studies, we give a summary on applicability of different types of state management, and reveal the mainstream directions of their applications, which may provide a general reference for research on IPv6 transition solutions in near future.

Acknowledgment

This article is supported by National Natural Science Foundation of China (NSFC) projects (no. 61422206, 61120106008, 61225012) and 863 project no. 2013AA010401.

References

[1] C. Aoun and E. Davies, "Reasons to Move the Network Address Translator-Protocol Translator (NAT-PT) to Historic Status," IETF RFC 4966, July 2007.

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

[3] B. Carpenter, "Advisory Guidelines for 6to4 Deployment," IETF RFC 6343, Aug. 2011.

[4] A. Durand *et al.*, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion," IETF RFC 6333, Aug. 2011.

[5] Y. Cui *et al.*, "Public IPv4-over-IPv6 Access Network," IETF RFC 7040, Nov. 2013.

[6] Y. Cui *et al.*, "Lightweight 4over6: An Extension to the DS-Lite Architecture," IETF draft, work in progress, Nov. 2013.

[7] O. Troan *et al.*, "Mapping of Address and Port with Encapsulation (MAP)," IETF draft, work in progress, Aug. 2013.

[8] J. Czyz *et al.*, "Measuring IPv6 Adoption," accepted by ACM SIGCOMM 2014.

[9] P. Wu *et al.*, "Transition from IPv4 to IPv6: A State-of-the-Art Survey," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 3, Dec. 2012, pp. 1407–24.

[10] M. Bagnulo, P. Matthews, and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers," IETF RFC 6146, Apr. 2011.

[11] J. Wu *et al.*, "Softwire Mesh Framework," IETF RFC 5565, June 2009.

[12] J. Wu *et al.*, "4over6 Transit Solution Using IP Encapsulation and MP-BGP Extensions," IETF RFC 5747, Mar. 2010.

[13] X. Li *et al.*, "Mapping of Address and Port using Translation (MAP-T)," IETF draft, work in progress, Dec. 2013.

[14] M. Townsley and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) — Protocol Specification," IETF RFC 5969, Aug. 2010.

[15] Y. Cui *et al.*, "Tunnel-Based IPv6 Transition," *IEEE Internet Computing*, vol. 17, no. 2, Mar. 2013, pp. 62–68.

Biographies

Yong Cui (cuiyong@tsinghua.edu.cn) is a full professor at Tsinghua University. He has co-authored three IETF RFCs for his proposals on IPv4/IPv6 transition technologies. He is the Co-Chair of the IETF Softwire Working Group, which focuses on global IPv6 transition technologies. He is also an Editor of IEEE TPDS and IEEE TCC.

Yuchi Chen is a graduate student at Beijing University of Posts and Telecommunications. His research interests include IPv6 transition and next-generation Internet.

Jiangchuan Liu is a university professor at Simon Fraser University. He is an Associate Editor of *IEEE Transactions on Big Data*, *IEEE Transactions on Multimedia*, and *IEEE Communications Surveys & Tutorials*. He is a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM (2015).

Yiu-leung Lee is a Distinguished Engineer at Comcast. He is the co-author of five RFCs and an active participant in IETF. His research interests include large-scale IP network design, IPv4-v6 transitioning, and network virtualization.

Jianping Wu [F] is a full professor in Tsinghua University. He was Chairman of Asia Pacific Advanced Network from 2007 to 2011. He received the Jonathan B. Postel Award from the Internet Society in 2010. He has authored five IETF RFCs for his proposal on IPv6 technologies. His research interests include next-generation Internet and IPv6.

Xingwei Wang is a full professor at Northeastern University, Shenyang, China. His research interests include future Internet, network security, and cloud computing.