Transition from IPv4 to IPv6: A State-of-the-Art Survey

Peng Wu, Yong Cui, Jianping Wu, Jiangchuan Liu, and Chris Metz

***Abstract*—In the process of Internet evolution, the transition from IPv4 to IPv6 has become inevitable and fairly urgent. IANA (Internet Assigned Numbers Authority) has finally exhausted the global IPv4 address space, which leaves the community no choice but pushes forward the IPv6 transition process. IPv4 and IPv6 networks both will exist during the transition period, while the two are not compatible in nature. Therefore it is indispensable to maintain the availability, as well as to provide the inter-communication ability of IPv4 and IPv6. Years ago a series of transition techniques were actually proposed. However, because of their technical immatureness, they failed to cover the solution space well. Some of these techniques were even obsoleted by IETF due to their flaws. This paper reconsiders the basic problems and key difficulties in IPv4-IPv6 transition, and introduces the principles of tunneling and translation techniques. Then the paper surveys the mainstream tunneling and translation mechanisms raised since 1998, especially the new mechanisms proposed recently, capturing the aspects of technical principles, pros and cons, scenarios and applicability. Recommendations on mechanism selection for different scenarios are provided. Moreover, the paper looks into the characteristics and transition requirements of practical ISP networks, and proposes the usage and deployment strategy of the transition mechanisms in both backbone and edge networks.**

***Index Terms*—IPv6 transition, heterogeneous network connectivity, translation, tunneling, heterogeneous addressing.**

I. Introduction

IPv4 [1] has been the network layer protocol since the very early stage of the Internet. The scale of IPv4 Internet has become far larger than one could ever imagine when designing the protocol [2]. Currently IPv4 Internet is facing a series of problems including address exhaustion, routing scalability, and broken end-to-end property. IANA (Internet Assigned Numbers Authority) had run out of global IPv4 address pool in Feb 2011, while simulations show that within 3 years all the RIRs (Rigional Internet Registries) will exhaust their IPv4 address space [3]. On the other hand, the scale of Internet is still growing fast, especially on the user side where the number of Internet-enabled mobile devices increases rapidly. This leads to continuous demands for new IP address allocation, which seems impossible to satisfy with IPv4. ChinaTelecom, one of the largest telecom ISPs (Internet Service Providers) in the world claims that by the end of 2012, they will use up all the IPv4 addresses they have acquired or can acquire. Besides, the prefix de-aggregation caused by address block subdivision, multihoming and traffic engineering has caused a burst in Global IPv4 RIB (Routing Information Base) and FIB (Forwarding Information Base). The Internet is suffering from this routing scalability problem. Moreover, the wide use of NAT has broken down the fundamental end-to-end property all over the Internet.

IPv6 [4] is developed as the next-generation network layer protocol, overcoming the problems in IPv4. Its 128-bit address format significantly enlarges the address space and will satisfy the address demands for a fairly long time. The length of the address also makes prefix aggregation fairly flexible, and subsequently achieves global addressing and routing in a hierarchical pattern. Forwarding efficiency is improved by simplifying the protocol header, as well as moving fragmentation to end hosts. In IPv6, flow label based QoS can be supported; stateless auto-configuration is invented to support Plug and Play feature [5]. Besides, IPv6 has better mobility and security supports than IPv4 [6]. In general, IPv6 is a redesign of IPv4. It solves the problems in IPv4 and provides better IP service. It has been widely believed that IPv6 is the most mature and feasible solution for the next-generation Internet.

However, IPv6 has no built-in backwards compatibility with IPv4, which means IPv6 networks cannot communicate with IPv4 in nature. Essentially IPv6 has created a parallel, independent network that coexist with its counterpart IPv4. If an IPv4 network wants to further support IPv6 communication, it has to carry out dedicated addressing and routing for IPv6, and update the network devices to enable IPv6. Currently IPv6-capable applications and IPv6-accessible contents are still the minority [7]; the majority of network resources, services and applications still remain in IPv4. Therefore IPv4 network will probably last for a long time. On the other hand, the continuous demands for new IP addresses are driving IPv6 towards a large-scale deployment. Therefore, IPv4 and IPv6 will coexist for a long period, and the transition process will be gradual. During this period, we need to manage the availability of both IPv4 and IPv6 and solve the issues arising in DNS, QoS, security and other aspects under the dual-stack environment. Above all, we need a number of transition

Manuscript received 14 December 2011; revised 9 July 2012. This work is supported by National Natural Science Foundation of China (no. 61120106008,60911130511), National Major Basic Research Program of China (no. 2009CB320501, 2009CB320503).

P. Wu and Y. Cui are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China (e-mail: {weapon, cy}@csnet1.cs.tsinghua.edu.cn).

J. Wu is with the National Laboratory for Information Science and Technology, Tsinghua University, Beijing, China (e-mail: jianping@cernet.edu.cn).

J. Liu is with the School of Computing Science, Simon Fraser University, Metro-Vancouver, British Columbia, Canada (e-mail: jcliu@cs.sfu.ca).

C. Mertz is with Cisco Systems, Inc. San Jose, CA, USA (e-mail: chmetz@cisco.com).

Digital Object Identifier 10.1109/SURV.2012.110112.00200

1553-877X/13/$31.00 © 2013 IEEE

techniques to maintain the connectivity of both IPv4 and IPv6, to achieve inter-connection between IPv4 and IPv6, and to promote the adoption process of IPv6.

The IPv6 transition techniques have been the main bones of contention in Internet Engineering Task Force (IETF) for over ten years. Around the year 2000, researchers proposed a series of transition techniques, including 6to4 (Connection of IPv6 Domains via IPv4 Clouds) [8], 6over4 (Transmission of IPv6 over IPv4 Domains without Explicit Tunnels) [9], NAT-PT (Network Address Translation - Protocol Translation) [10], SIIT (Stateless IP/ICMP Translation Algorithm) [11], BIS (Bump-In-the-Stack) [12], etc. Analyses of these techniques were proposed [13], as well as technique usage guidelines and transition architectures based on these techniques [14]–[17]. However, most of these techniques did not come to a wide deployment, and the transition problem did not get solved by these techniques. The reason is that they did not put enough consideration on the factors of scalability, addressing, deployment model, IPv4 address shortage, etc. Some of these techniques were even obsoleted by the IETF because of these flaws. Yet these trials do become inputs for later researches. In 2004 and 2005, two dedicated working groups were established to carefully reconsider the transition problem and develop enhanced transition techniques. Researches were carried on in these two groups ever since, and new mechanisms have been coming out recently.

The demand for IPv6 transition techniques comes from multiple entities in the community. Network operators must find feasible transition mechanisms and subsequently make feasible transition plans, to cover all the potential communication demands of the customers. Vendors expect to invest on implementing well-developed transition techniques, so that their products can have good capability and bring high profits. As for ICPs (Internet Content Providers), they need to find a way to provide the existing services for both IPv4 and IPv6 users, and coordinate their services with a foreseeable deployment of transition techniques on the Internet. These different entities raise diverse requirements in techniques. Besides, to develop a proper transition technique, various critical issues should be studied, including routing and forwarding methods in heterogeneous networks, a feasible IPv4-IPv6 address mapping method despite of the asymmetry of address spaces, an end-to-end heterogeneous addressing method, scalability, end-to-end property and upper-layer transparency, etc. Both the diversity of requirements from different entities and the variety of relevant issues bring great challenges to the transition technique research.

This paper introduces heterogeneous traversing and heterogeneous inter-connection as the basic problems of IPv6 transition, with IPv4-IPv6 translation and IPv4-over-IPv6/IPv6-over-IPv4 tunneling as the basic solutions. Then the paper provides a comprehensive survey of the mainstream translation and tunneling mechanisms proposed since the birth of IPv6, capturing the aspects of technical principles, pros and cons, scenarios and applicability. In this survey, extra efforts are spent on the new mechanisms coming out in the recent 5 years, aiming to track the cause of the improvements and thereby prove the advantages against the pervious mechanisms. Recommendations on mechanism selection for different transition scenarios are proposed. Moreover, the paper looks into the characteristics and transition requirements of practical ISP networks, and customizes the usage and deployment strategy of the transition mechanisms in both backbone and edge networks. Examples of backbone networks and telecom access networks are provided as a case study.

II. PROTOCOL SPECIFICATION: IPV4 VS. IPV6

The basic protocol specification of IPv6 was proposed in 1998, and related standards have been developed ever since. IPv6 has a different address architecture from IPv4, as well as a series of new features [18].

A. Addressing

The most obvious advantage of IPv6 over IPv4 is its larger address space. The 128-bit IPv6 address length provides approximately $3.4 * 10^{38}$ available addresses, while IPv4 only provides $4.3 * 10^{9}$ addresses due to the 32-bit limit. The IPv6 address length is selected based on the lesson of IPv4 address exhaustion. The vast address space is believed to be enough for the foreseeable future.

A typical IPv6 unicast address is composed of two parts: a 64-bit network prefix and a 64-bit interface identifier. The interface identifier is unique within a subnet prefix and used to identify interfaces on a link. Unlike in IPv4, the subnet size in IPv6 is fixed to 2^{64}. The 64-bit network prefix length provides great flexibility in network management. By recommendation a /32 prefix is provided for an ISP, while a prefix between /56 and /64 is given to an end-consumer site [19]. This leaves the ISPs at least /24 space to organize their networks, and the global Internet /32 space to manage global routing. Therefore address allocation can be simplified and route aggregation can be achieved efficiently, under which circumstances it is feasible to build a hierarchical addressing and routing architecture. Besides, the vast address space along with the 64-bit subnet size also eliminates the major demands for NAT.

Another benefit of this address format is renumbering. In IPv4, renumbering an existing network is a major effort. In IPv6, however, with RA (Router Advertisement) [20] for changing network prefixes and SLAAC (Stateless Address Auto-configuration) [5] for self-configuring interface identifiers, renumbering an IPv6 network will be much easier. Moreover, this longer address format also allows the implementation of special address schemes, such as embedding an IPv4 address.

B. New features in IPv6

In order to inherit the merits of IPv4 smoothly, IPv6 improves some beneficial features of IPv4 up to its own standard, and goes further with introducing additional features that are not presented in IPv4:

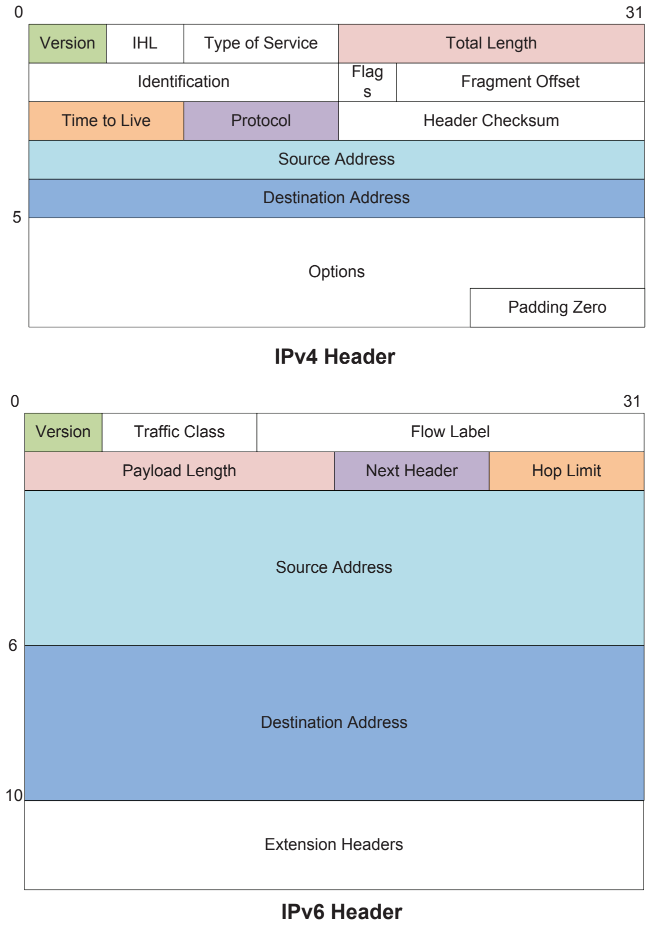
(1) Stateless address auto-configuration. Besides manual configuration and stateful configuration (DHCP), IPv6 provides a third, stateless configuration manner. IPv6 hosts can leverage ND (Neighbor Discovery) Protocol [20] to configure themselves automatically when connected to a network. In a standard procedure, the host generates a link-local address

by appending an interface identifier to the well-known link-local prefix, and then verifies the uniqueness of the address by sending out a Neighbor Solicitation message. When the verification is confirmed, the host assigns the link-local address to the interface, and then either send out a link-local Router Solicitation message to retrieve a corresponding Router Advertisement message from a router, or wait for periodical Router Advertisement that contains network-layer configuration parameters.
(2) Simplified protocol header. As is shown in Figure 1, some insignificant fields such as IHL and TOS, as well as the fragmentation-related fields are removed or moved to optional extension headers. Header Checksum is also removed and the responsibility is left to the link layer and the transport layer. The IPv6 header simplifies the processing on routers.
(3) Moving fragmentation from routers to end hosts. IPv6 hosts are required to either perform path MTU discovery and fragment packets before sending them out, or only send packets no bigger than the minimum MTU (1280 bytes). This feature also simplifies the processing on routers.
(4) Flow label. A flow label field is presented in the IPv6 header, which provides the flexibility for ISPs to perform traffic engineering, QoS service, etc.
(5) Mandatory network-layer security. Implementation of IPsec was mandated in original IPv6 specification. Recently IETF lowered the requirement a bit to “all IPv6 nodes SHOULD support IPsec architecture”.
(6) Better mobility support. Besides the bidirectional tunneling mode, Mobile IPv6 supports the route optimization mode which allows the packets to be sent directly between the mobile node and the correspondent node, with the aid of home agent only in the initial phase. In this mode the shortest communication path can be employed, and the congestion at home agent can be eliminated.

C. Issues with IPv4-IPv6 coexistence

Due to the significant differences in the protocol format and behavior, IPv4 and IPv6 are not inter-operable. To further support IPv6, an ISP has to create an essentially a parallel, independent network. As to end hosts, modern computer operating systems have already implemented dual-protocol stacks for access to both networks.

The coexistence of IPv4 and IPv6 networks raises several general issues in different aspects. Network devices like routers, firewalls and various servers have to upgrade their hardware and software to support IPv6 features. Extra resource dispatching mechanisms are needed in an overlapped environment, to allocate shared resources (link bandwidth, FIB entries, etc.) to each network and guarantee service for both. A new QoS strategy for IPv6 should be developed to leverage the flow label field explicitly. New protocols in IPv6 suite such as Neighbor Discovery and DHCPv6 may raise new security risks and thereby need to be evaluated. For end hosts, applications require intelligence to decide which protocol stack to use when the remote end may be reachable by both IPv4 and IPv6 (For example, DNS responses with both A and AAAA records). With the community getting to know IPv6 better, solutions to these issues have been developed or can be expected in the near future.



IPv4 Header

IPv6 Header

Fig. 1. Protocol header: IPv4 vs. IPv6

Beside all the issues mentioned above, there is a much more fundamental problem that has arisen since the very first day of the IPv4-IPv6 coexistence: network connectivity under the IPv4-IPv6 heterogeneous environment. Since IPv4 and IPv6 protocols are not compatible, they run their individual addressing and routing systems. Without additional mechanisms, the two types of networks cannot communicate. However, on the real Internet, IPv4 and IPv6 networks will be mixed together, because different ISPs, ICPs, and users would decide independently when to adapt themselves to IPv6. While a network operator may run an IPv4 network, or an IPv6 network, different networks still want to connect to and communicate to each other; a network user may have IPv4 access, or IPv6 access, yet different users want to communicate with each other. Therefore we have to enforce some artificial ”inter-operability” between IPv4 and IPv6, to enable the network connectivity in heterogeneous networks. A great series of efforts have been made on this problem, and the set of proposed solutions are called IPv6 transition techniques. The following of this paper will focus on the heterogeneous connectivity problem and the transition techniques.

III. Heterogeneous network connectivity problem

The problem of heterogeneous network connectivity is, how to build connectivity across networks that are mixed with IPv4

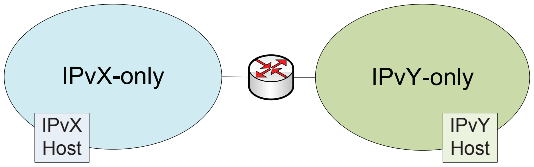


Fig. 2. inter-connection scenario

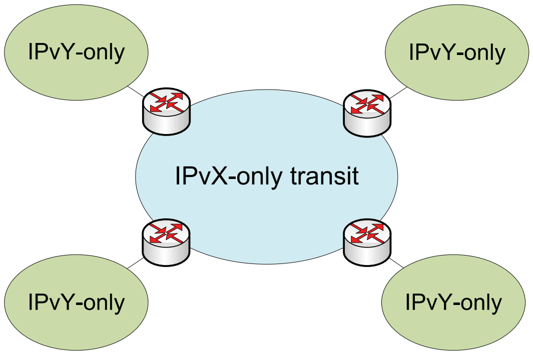


Fig. 3. Mesh traversing scenario

and IPv6 blocks. The key feature here is that the connectivity will cross one or more IPv4-IPv6 network borders, so we have no options but to enable artificial inter-operability between IPv4 and IPv6 on the borders. Since the ultimate goal is to realize a complete transfer to IPv6 networks, enabling the inter-operability also helps to overcome the influence of the IPv4's existing base [21].

Theoretically, the connectivity may cross several borders along the path. However, we can decompose it into two types of atomic problems: heterogeneous inter-connection (Figure 2) and heterogeneous traversing (Figure 3 and Figure 4). This is actually a practical way to study the original problem, because (1) a more complicated situation is always composed of multiple atomic problems, (2) the solutions have to be tightly coupled with local addressing and routing, and (3) since both the IPv4 Internet and theIPv6 Internet are quite well-organized already , the two atomic problems are actually the most common instances in the real world.

An inter-connection problem happens when networks or hosts using different address families are directly connected and want to communicate. Transition mechanisms are required to achieve the communication between IPv4 and IPv6 networks (or hosts). Since the source and the destination lie in different address families, protocol conversion is indispensable to achieve the inter-connection.

The Behave Working Group [22] of IETF is focusing on developing and standardizing solutions to inter-connection problems. Based on the diversity of network scales and from the perspective of the communication initiators, Behave proposes 8 different scenarios for the inter-connection problem: an IPv6 network to an IPv4 network, an IPv6 network to the IPv4 Internet, the IPv6 Internet to an IPv4 network, the IPv6 Internet to the IPv4 Internet, an IPv4 network to an IPv6 network, an IPv4 network to the IPv6 Internet, the IPv4 Internet to an IPv6 network, the IPv4 Internet to the IPv6 Internet. (Compared with the Internet, a network here has a clearly identifiable administrative domain, for example an enterprise campus network, a mobile operator's cellular network, a residential subscriber network, etc.) In different scenarios, the scalability requirements are different, which results in difficulties with different levels to overcome, so as to achieve inter-connection. Therefore we need to consider them according to their levels.

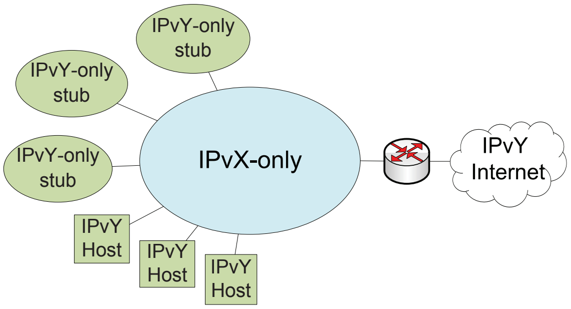


Fig. 4. Hub & Spokes traversing scenario

Besides the inter-connection problem between networks, there is this special scenario where an IPv4 application needs to communicate with IPv6 using the host's IPv6 connection, or an IPv6 application needs to communicate with IPv4 using the host's IPv4 connection. In this case the TCP/IP stack inside the host should provide the protocol conversion mechanism that enables the IPvX application to leverage the IPvY connection. (For description convenience, we use {IPvX, IPvY} to represent {IPv4, IPv6}.) This can be viewed as "inter-connection" between applications and heterogenous networks.

A traversing problem happens when two or more native IPv4/IPv6 networks (or hosts) are separated by a network which uses the other address family and thereby is not IPv4/IPv6-capable. Transition mechanisms are required for crossing the heterogeneous network. If IPv4 networks or hosts are separated by an IPv6 network in the middle, IPv4-over-IPv6 traversing is required; if IPv6 networks (or hosts) are separated by an IPv4 network, the problem then is IPv6-over-IPv4 traversing.

The Softwire Working Group [23] of IETF is focusing on developing and standardizing solutions to the traversing problem. Softwire divides the traversing problem into two typical scenarios: Mesh and Hub & Spokes [24]. In the Mesh scenario, network islands of one Address Family (IPvY) are separated by a network of the other Address Family (IPvX) from connecting with each other. In the Hub & Spokes scenario, a number of hosts or stub networks (IPvY) are separated by a network of the other address family (IPvX) from reaching a centralized native access. The former case often happens in transit networks while the latter one is usually seen in edge networks.

Due to the incompatibility of IPv4 and IPv6, it is not easy to achieve heterogeneous inter-connection and traversing. While we can compromise on the semantics of protocol fields, there is still a critical gap where the address format of IPv6 is 96-bits longer than that of IPv4, which makes it impossible to build one-to-one address mapping between IPv4 and IPv6. More specifically, although mapping from IPv4 address space to IPv6 is easy, IPv6 address space can not be mapped effectively into IPv4. Moreover, when solving these problems, we should always promote IPv6 usage and ease IPv6 adoption; the proposed transition mechanisms should not hinder the transition process.

The alternative is to compromise on a dual-stack Internet, which means every node on the Internet supports both IPv4 and IPv6. Then any two nodes can use both IPv4 and IPv6 for communication; IPv4-IPv6 inter-connection or traversing will become unnecessary. In that case, the whole Internet becomes two logically-separated networks based on the same infrastructure. However, dual-stack Internet is neither practical nor continuable, because large-scale expansion of IPv4 Internet is unrealistic considering the address space exhaustion, and the cost of fully supporting both IPv4 and IPv6 is unacceptable.

In spite of that, it is still feasible to enable dual-stack for a small portion of the Internet nodes. Actually, as a result of protocol incompatibility, dual-stack node will be an indispensable element to achieve IPv4-IPv6 inter-operability. To be more specific, the dual-stack nodes located on the IPv4-IPv6 network border can communicate with both IPv4 and IPv6, and perform the IPv4-IPv6 inter-operation in between. Based on this, the community proposed two categories of transition techniques: translation and tunneling, which will be covered in the next two sections.

IV. Translation mechanisms

A. Basic Principle of Translation

IPv4-IPv6 translation is used to achieve direct communication between IPv4 and IPv6. The basic principle of translation is shown in Figure 5. The idea is to convert the semantics between IPvX and IPvY, turning IPvX packet into IPvY if the packet is destined to IPvY network, or turning IPvY packet into IPvX if the packet is destined to IPvX network. Usually, translation happens on the IPvX-IPvY border, so the translator would be an AFBR (Address Family Border Router). Suppose Host1 (H1) in IPvX network is the communication initiator, and Host2 (H2) in IPvY network is the remote end. H1 has to learn the in-protocol address (IPvX) used by H2 before the communication starts. Later the data packets with this address as destination will be forwarded to the translator, translated into IPvY and forwarded to H2. On the other hand, the IPvY source address for these packets, i.e. the IPvY address used by H1, is assigned or calculated by the translator during the translation. Along with these addressing operations, routing support should guarantee that the IPvX packets destined to H2 IPvX address and the IPvY packets destined to H1 IPvY address are forwarded through the translator. IPv4-IPv6 translation is similar to IPv4 NAT on some certain level. However, applying translation to large-scale networks and asymmetric IPv4-IPv6 address space is much more challenging than that to the scenario of ordinary IPv4 NAT.

The basic data plane operation of translation is IPv4-IPv6 packet translation, which involves network, transport, and application layer. It includes address and port conversion, IP/TCP/UDP protocol field translation, and application layer translation (address and port conversion when they appear in application protocol [25]). What is more, to overcome further diversities in the protocol definition between IPv4 and IPv6, translation has to take care of issues like fragmentation and reassembling, path MTU discovery, ICMP, etc. [26]. As to the control plane, translation should follow the address conversion rule: either some special address scheme needs to be deployed in advance, or dynamic address bindings have to be built during the translation. Heterogeneous addressing (learning the in-protocol address of the remote end) and corresponding routing should be performed based on the address conversion rule.

According to the address conversion manner, we can divide network-side translation mechanisms into stateless translation and stateful translation. There is also host-side translation, which happens in the TCP/IP stack of the end host.

B. Stateless Translation

SIIT (Stateless IP/ICMP Translation Algorithm) [11] is an early stateless translation mechanism. It proposes the basic principle of IPv4-IPv6 stateless translation and the algorithm for IP/ICMP semantic conversion (Figure 6). The SIIT address scheme is based on the assumption that every IPv6 host in a network possesses an IPv4 address. The IPv6 address of each IPv6 host is generated by adding the IPv6 prefix 0:ffff:0:0:0/96 before the IPv4 address. This type of IPv6 addresses is called IPv4-translated address, which is assigned to an IPv6 host and potentially matches an IPv4 address. On the other hand, the IPv6 address of an actual IPv4 host is generated by adding a different IPv6 prefix ::ffff:0:0/96 before the IPv4 address. This type of IPv6 addresses is called IPv4-mapped address, which is mapped from an IPv4 address to represent an IPv4 host in an IPv6 network. However, SIIT specifies neither how an IPv6 host retrieves an IPv4-translated address, nor how an IPv6/IPv4 host learns the IPv4-mapped/IPv4-translated address of the remote end. Routing support for the address mapping rules is not specified either. Following the two address mapping rules, the address translation can be performed by algorithmic mapping. When translating an IPv4 packet into IPv6, SIIT translator adds the prefixes of ::ffff:0:0/96 and 0:ffff:0:0:0/96 to the source and destination addresses respectively; when translating an IPv6 packet into IPv4, the translator removes the corresponding prefixes from the source and destination addresses.

SIIT keeps the translator stateless according to the address conversion rule. It takes a unified processing for all packets; the data plane performance is not bound by the number of users, and line-speed processing is expected. As long as heterogeneous addressing is realized, SIIT can provide bi-directional communication. SIIT promotes IPv6 development by providing IPv6 networks with the bi-directional connectivity to legacy IPv4. SIIT does not introduce new security issues to the network. However, the usage of fixed prefix for IPv4-translated address brings significant routing scalability problem, because different prefixes composed of the /96 prefix + the IPv4 prefix of the SIIT host addresses would be injected into IPv6 global RIB and FIB and they are impossible to aggregate. Because of the per-host IPv4 address consumption requirement, the IPv6 side of SIIT cannot be huge. Therefore its application scenarios are IPv6 network ⇌ IPv4 Internet and IPv6 network ⇌ IPv4 network.

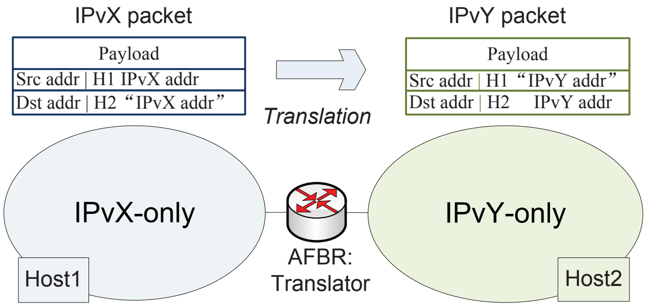


Fig. 5. Basic principle of translation

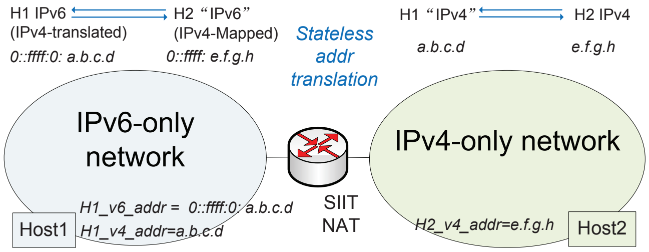


Fig. 6. SIIT: principle

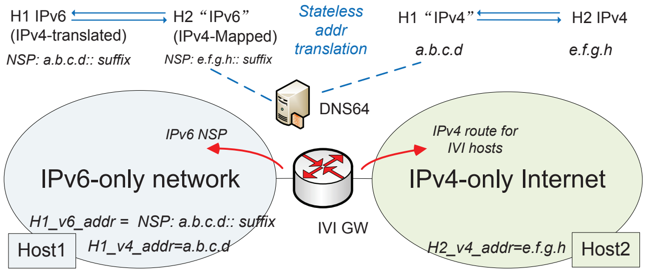


Fig. 7. IVI: principle

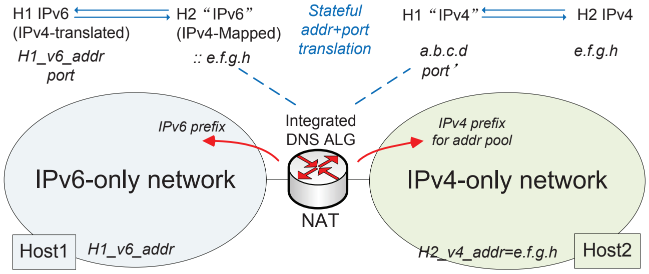


Fig. 8. NAT-PT: principle

IVI [27] follows the principle of stateless translation and improves SIIT. IVI uses a network-specific, variable prefix (NSP) to replace the two fixed /96 prefixes in SIIT. Both the IPv4-translated addresses and IPv4-mapped addresses are presented as NSP + IPv4 address + suffix [28] (Figure 7). In this way, these IPv6 addresses can naturally aggregate as the NSP within the network; therefore routing scalability is no longer a concern. The IPv4-translated addresses can be assigned to IPv6 hosts through DHCPv6 or SLAAC (With SLACC IPv4 address embedded in the first 64-bits). The IPv6 hosts learn IPv4-mapped addresses of IPv4 hosts by querying a local DNS server, DNS64 [29]. DNS64 turns the A records of IPv4 hosts into AAAA records following the address mapping rule. On the other hand, the IPv6 hosts register their IPv4 addresses in DNS server as A records, which are used to answer the heterogeneous addressing query from the IPv4 side. As for routing, The IVI translator (i.e., IVI gateway) is responsible for advertising the prefix of IPv4 addresses possessed by IPv6 hosts to the IPv4 side, as well as the NSP route to the IPv6 side.

IVI is an improvement of SIIT: routing scalability problem and addressing issues that exist in SIIT are no longer in IVI. Meanwhile, IVI inherits the advantages of SIIT, including high performance, bi-directional connectivity, IPv6 promotion ability and the guarantee of security. However, the per-host IPv4 address consumption is still required in IVI. Therefore its application scenario is also IPv6 network ⇌ IPv4 Internet and IPv6 network ⇌ IPv4 network. An extension of IVI is proposed to support address multiplexing [30], in which one IPv4 address is shared by multiple IPv6 hosts through port space division.

C. Stateful Translation

Unlike stateless translation which assigns IPv4 address ownership to IPv6 hosts, stateful translation maintains the IPv4 address resource as a pool on the translator, and uses the resource at per-port granularity. The number of IPv4 addresses could be much smaller than the number of served IPv6 hosts. NAT-PT (Network Address Translation - Protocol Translation) [10] is an previous stateful translation mechanism which claimed to support both IPv6→IPv4 and IPv4 → IPv6. For presenting an IPv4 host in IPv6, NAT-PT follows the addressing manner of SIIT and IVI, which generates an IPv4-mapped address by adding an IPv6 prefix (::/96) to the IPv4 address of the host. On the contrary, for presenting the IPv6 hosts in IPv4, NAT-PT learns from traditional IPv4 NAT and leverages the manner of stateful address + port binding (Figure 8). More precisely speaking, based on realtime flows, the NAT-PT translator dynamically binds the IPv6 host address and the transport layer ID (TCP/UDP port or ICMP ID) with one IPv4 address and one transport layer ID from the IPv4 address pool. Following the two addressing schemes, NAT-PT translator should advertise the prefix of IPv4 address pool to the IPv4 side, as well as the IPv6 prefix of ::/96 to the IPv6 side. As for the data plane, when translating an IPv6 packet into IPv4, the translator uses the source IPv6 address and port to look up the NAT binding table and find the source IPv4 address and port (creates a new binding if no former binding is found); the destination IPv4 address is generated by removing the IPv6 prefix. When translating an IPv4 packet into an IPv6 packet, the translator adds the prefix to the source IPv4 address to form the source IPv6 address, and uses the destination IPv4 address and port to look up the NAT binding table and find the destination IPv6 address and port (drop the packet when no binding is found).

The difficulties of initiating the communication from the IPv4 side and the IPv6 side are quite different. If the communication is initiated from the IPv6 side, the IPv6 destination address for the source host is an IPv4-mapped address generated statelessly from the IPv4 destination address. If the communication is initiated from the IPv4 side, then the stateful binding to map the IPv6 destination into IPv4 has to be built on the translator first and informed to the source host; otherwise the source host cannot figure out the IPv4 destination address and port at all. Unfortunately, this is still unpractical due to issues like potential useless state maintenance, state inconsistency, etc. Nevertheless, NAT-PT suggests to achieve the heterogeneous addressing procedure for both sides with DNS ALG (Application-Layer Gateway) on the translator. To cooperate, the DNS server in IPv6 should have the heterogeneous, IPv6 address of the DNS server in IPv4, while the DNS server in IPv4 network should have the heterogeneous, IPv4 address of the DNS server in IPv6 in advance, so that the DNS messages could traverse the translator. For an AAAA query from IPv6, the translator converts it into an A query, and converts the A response into AAAA record with an IPv4-mapped address. For an A query from IPv4, the translator converts it into AAAA query, extracts the IPv6 address from the AAAA response, creates a stateful binding between the IPv6 address and an IPv4 address from the pool, and converts the AAAA response into A response with the IPv4 address. In the more general case of per-flow binding, the DNS protocol has to be extended to include port information, which brings significant changes to today's DNS model.

The binding table lookup operation may become the performance bottleneck of NAT-PT. If the NAT-PT translator is implemented in software, the processing speed is negatively correlated to the size of the table. If it is implemented by hardware, the cost and capacity would be positively correlated to the size of the table. Besides, the time-delay of creating NAT binding for new flows could still lower the processing speed. The gain is the better IPv4 address utilization than that of stateless translation. NAT-PT promotes the IPv6 development by providing the connectivity with legacy IPv4. However, NAT-PT has a series of issues including the heterogeneous addressing difficulty [31], which drove IETF to discard the NAT-PT protocol standard.

Nevertheless, the IPv6→IPv4 direction of NAT-PT is still feasible in general. It was later enhanced and proposed as a dedicated mechanism–NAT64 (Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers) [32]. NAT64 only specifies the communication initiated from the IPv6 side. It prescribes the IPv6 prefix used for IPv4-mapped address to be 64:FF9B::/96. Accordingly, NAT64 translator should advertise the prefix of 64:FF9B::/96 to the IPv6 side, and the prefix of the IPv4 address pool to the IPv4 side. As to the heterogeneous addressing manner, NAT64 refines the DNS-based method. The DNS ALG function is extracted from the translator and becomes a dedicated DNS64 server, which is actually cascaded in the hierarchical DNS system. It translates the AAAA query from IPv6 hosts into A query when receiving one, and translates the A response for IPv6 hosts into AAAA response following the IPv4-mapped address rule before sending one out. The data plane processing is the same with IPv6→IPv4 NAT-PT.

The performance of NAT64 is similar to NAT-PT. The main security issue of NAT64 is DoS(Deny of Service) attack on the binding table, with ingress filtering on the IPv6 side as the solution. The per-flow stateful nature of NAT64 excludes the scenario of IPv6 Internet→IPv4 Internet, for uncontrollable network sizes on both sides could lead to the unmanageable state number. So the application scenarios of NAT64 are: IPv6 network→IPv4 Internet, IPv6 network→IPv4 network, and IPv6 Internet→IPv4 network.

D. Host side translation

In addition to the network-side translation, there are also some host-side translation mechanisms, including BIS (Bump-In-the-Stack) [12] and BIA (Bump-in-the-API) [33]. These two mechanisms are used for the scenario where an IPv4 application on the end host needs to communicate with an IPv6 remote end through the IPv6 network. Here the host is only provided with IPv6 access, and the remote end is also in IPv6, while the upper-layer application uses the IPv4 stack. So what we need is an IPv4-IPv6 translation inside the TCP/IP stack of the host, to simulate an IPv4 "environment" and "trick" the application into believing the remote end is also IPv4. The purpose of the host-side translation is to preserve IPv4-only applications in the IPv6 environment and avoid application upgrades. Therefore, the reversed scenario, i.e., translation between an IPv6 application and an IPv4 network is not demanded. We can simply use an IPv4 application in that case.

BIS and BIA take different manners to achieve this translation. BIS processes a per-packet translation, while BIA sets the translation on the socket level. They both maintain the stateful IPv4-IPv6 address binding (port number not included) for each remote end, which is typically triggered by DNS ALG during the heterogeneous addressing phase. Any unassigned IPv4 addresses can be used to create binding, and assigned to the host for BIS/BIA usage, for they will not flow out the host. BIS performs packet translation based on this binding table. When upper-layer application passes down an IPv4 data packet, BIS translates the packet into IPv6, using the host's IPv6 address as source address, and looking up the IPv6 destination address in the binding table with the IPv4 destination address. When an IPv6 packet is received from the network, BIS translates it into IPv4, uses the host's IPv4 address as destination address, and look up the IPv4 source address in the binding table with the IPv6 source address. BIA uses the binding table for socket translation. It accepts the IPv4 socket invocation from the application, translates it into an IPv6 socket invocation which is handled by the IPv6 stack, and sends the return values of the IPv6 invocation back to IPv4. Data is also handed over between IPv4 socket and IPv6 socket. Since the translation happens on the socket API level, in BIA there are no IPv4 packets.

BIS and BIA are implemented in software and planted inside end hosts. They are only responsible for the IPv4 traffic inside their own host, so performance will not be an issue. The main security risk is also inside the host: malicious applications may launch a DoS attack by sending a large number of DNS queries and exhausting the IPv4 address pool and the space of the binding table. The two mechanisms obviously promote IPv6 by easing the transition of upper-layer applications.

TABLE I
SUMMARY OF TRANSLATION MECHANISMS

Mechanism	Scenario	Mechanism status	Address mapping	IPv6 prefix usage	IPv4 address usage	Issues
SIIT	IPv6 network ⇌ IPv4 network; IPv6 network ⇌ IPv4 Internet	Replaced by IVI	Stateless address mapping	Separated, fixed prefixes for IPv6 hosts and IPv4 hosts	One IPv4 address for each IPv6 host	Voracious IPv4 address consumption; routing scalability problem; common translation issues
IVI	IPv6 network ⇌ IPv4 network; IPv6 network ⇌ IPv4 Internet	Replace SIIT	Stateless address mapping	Same network-specific prefix for IPv6 hosts and IPv4 hosts	One IPv4 address for each IPv6 host	Voracious IPv4 address consumption; common translation issues
NAT-PT	IPv6 network → IPv4 Internet; IPv6 network → IPv4 network; IPv6 Internet → IPv4 network; IPv4 → IPv6(low feasibility)	Replaced by NAT64 on IPv6 → IPv4 direction	Stateful address + port binding	Fixed IPv6 prefix for IPv4 hosts	Address pool maintained by the translator	Low feasibility on IPv4 → IPv6 direction; per-flow state maintenance; DoS risk; common translation issues
NAT64	IPv6 network → IPv4 network; IPv6 Internet → IPv4 network; IPv6 network → IPv4 Internet	Replace NAT-PT on IPv6 → IPv4 direction	Stateful address + port binding	Fixed IPv6 prefix for IPv4 hosts	Address pool maintained by the translator	Per-flow state maintenance; DoS attack risk; common translation issues
BIS/BIA	IPv4 application⇌ IPv6 Internet, preserve IPv4-only applications in IPv6 network	Active host side translation mechanism	Stateful address binding	None	Any unsigned IPv4 addresses	DoS attack risk; ALG issue

E. Summary of translation mechanisms

The translation mechanisms are summarized in Table I. The common translation issues in the table include:
(1) Scalability. Stateless translation requires voracious consumption of IPv4 addresses, while stateful translation requires per-flow state maintenance. Both are not suitable for large-scale networks.
(2) Heterogeneous Addressing. To start the communication, the initiator should learn the in-protocol address of the other end. Therefore, at least one end of the communication has to be aware of the translation: either the initiator constructs the in-protocol address itself, or the other end informs the initiator of the address, for example by means of DNS ALG/DNS64. When DNS ALG/DNS64 is not available (it is not likely that every host will register on DNS servers), this would become a problem: it will bring behavior modifications to the applications, not to mention the infeasibility in IPv4 → IPv6 stateful translation.
(3) Application layer translation. Theoretically translators should support application layer translation, but in reality it is impossible to satisfy this requirement in real time. The difficulties are caused by the cost of implementing application layer operations on network devices, and the variety of applications.

The fundamental causes of these issues are the asymmetry of IPv4 and IPv6 address spaces and the broken end-to-end property. Currently the researchers and engineers admit these issues, yet they still want to utilize the translation mechanisms. There are some ongoing efforts aiming to lighten the problem. For example, PCP protocol [34] offers a method for end hosts to apply for address+port mapping from a translator, and PET [35], [36] proposes the idea of transferring translation spot with a tunnel.

The mainstream of translation techniques is network translation. Among the network translation mechanisms, IVI is a feasible stateless translation mechanism, and NAT64 is a feasible stateful translation mechanism. Stateless translation achieves bidirectional communication at the cost of voracious IPv4 address consumption; stateful translation achieves better IPv4 address utilization, yet it requires per-flow state maintenance. So far there is no feasible stateful solution to IPv4 → IPv6 scenarios or stateless solution to IPv4 → IPv6 Internet scenarios. Host side translation provides the approach to preserving IPv4-only applications in the IPv6 environment and avoid application upgrades.

V. TUNNELING MECHANISMS

A. Basic Principle of Tunneling

Tunneling is used to achieve heterogeneous traversing. The basic principle of tunneling is shown in Figure 9. To deliver IPvY packets across the IPvX network in the middle, we deploy two tunnel endpoints on the border of the IPvX network. When the ingress endpoint (Tunnel endpoint 1) receives an IPvY packet from the IPvY network, it encapsulates the IPvY packet with IPvX protocol header and puts the whole IPvY packet into the payload of the new IPvX packet. Then the IPvX packet is forwarded through the IPvX network. When the egress endpoint (Tunnel endpoint 2) receives the encapsulated IPvX packet, it decapsulates the packet, extracts the original IPvY packet and forwards it to the IPvY network. When performing the encapsulation, Endpoint 1 should fill in the IPvX destination address in the encapsulation header properly, which guarantees that the encapsulated packet will be forwarded to endpoint 2. Usually the IPvX address of endpoint 2 is figured out and used as the encapsulation destination address. Tunneling is actually a generic technology; under the scope of IPv6 transition, tunneling can achieve communications between IPv4 networks/hosts across an IPv6 network (IPv4-over-IPv6), and communications between IPv6 networks/hosts across an IPv4 network (IPv6-over-IPv4).

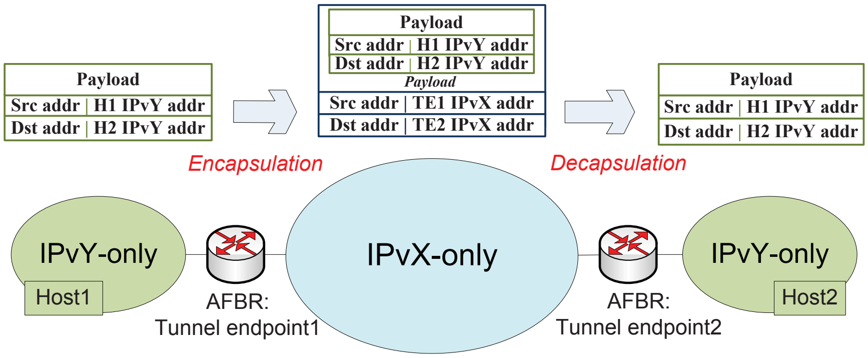


Fig. 9. Basic principle of tunneling

The basic data plane operation of tunneling is encapsulation and decapsulation. For IPv6 transition usage, the encapsulation manners such as IP-IP [37], GRE (Generic Routing Encapsulation) [38], [39], L2TP (Layer Two Tunneling Protocol) [40], MPLS (Multiple protocol Label Switching) [41], [42], IPsec (Internet Protocol Security) [43] can all be adopted. To support correct encapsulation on data plane, the control plane has to support encapsulation address mapping by a particular address scheme or address/prefix binding. IPvY routing across the IPvX network is required for IPvY forwarding.

For static, simple traversing demand, traditional configured tunnels can be a satisfactory solution. As for the general case, we need more flexible tunneling mechanisms. Based on the diversity in traversing scenarios, there are three types of tunneling mechanisms: Tunnel Mesh, Host-to-Host Tunnel and Hub & Spokes Tunnel [44].

B. Tunnel Mesh Mechanisms

6to4 (Connection of IPv6 Domains via IPv4 Clouds) [8] aims to solve the problem that isolated IPv6 networks communicate with each other across an arbitrary IPv4 network, as well as the problem that an isolated IPv6 network communicates with IPv6 Internet across the same IPv4 network. As is shown in Figure 10, 6to4 leverages the address scheme of algorithmic mapping to achieve an automatic tunnel. It links the addresses of each isolated IPv6 network to the IPv4 address of its 6to4 Router with a fixed IPv6 prefix: the addresses in the IPv6 network must be within the prefix of 2002:IPv4ADDR::/48, in which the 32-bit IPv4ADDR is the address of 6to4 Router. Thus for communication between isolated IPv6 networks, when a 6to4 Router performs encapsulation, the encapsulation destination address which is the address of the egress 6to4 Router can be extracted directly from the IPv6 destination addresses. As for the communication between an isolated IPv6 network and the IPv6 Internet, 6to4 builds a tunnel between 6to4 Router and 6to4 Relay. In this case the 6to4 Router should learn in advance the Relay's IPv4 address as encapsulation destination. Meanwhile, the 6to4 Relay should advertise the prefixes of 2002:IPv4ADDR::/48 for each isolated network. 6to4 does not impose extra routing requirements on the IPv4 network in the middle.

6to4 keeps the 6to4 Router and Relay stateless. Similar to SIIT, it processes all the packets in a unified way, so the data plane performance is not bound by the number of users and can easily reach line-speed. 6to4 promotes IPv6 development by providing IPv6 connectivity despite of the separation caused by IPv4. However, the use of 6to4 Relay brings a routing scalability problem. 6to4 splits the well-known 2002::/16 prefix into scattered IPv6 network prefixes which cannot aggregate on a 6to4 Relay at all. Every 6to4 Relay has to advertise all the /48 prefixes from its customer IPv6 networks into the global IPv6 RIB and FIB. Otherwise the relay may lose some customers' traffic or receive unwanted traffic. With the IPv6 Internet getting larger and larger, the global IPv6 FIB and RIB will suffer from the large number of such /48 prefixes. Therefore 6to4 will not be a continuable IPv6-over-IPv4 mesh solution. 6to4 Routers and Relays also open a hole for spoofing attack on IPv6 from IPv4, which exists in most tunnel mechanisms.

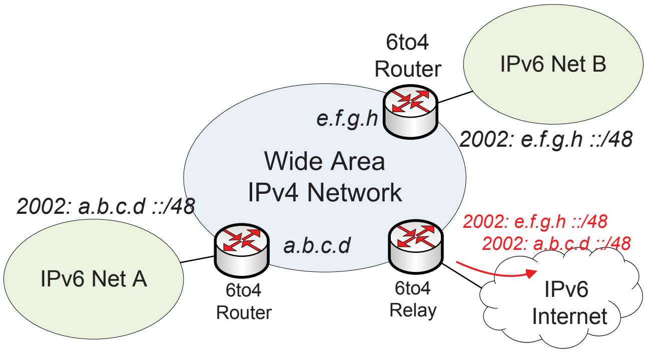
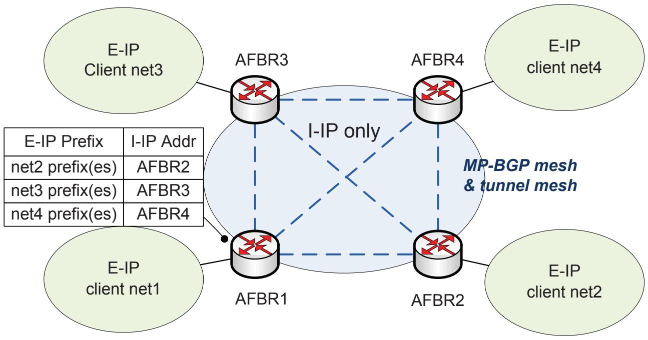


Fig. 10. 6to4 Principle

Softwire Mesh [45]–[47] is another router-to-router tunnel mechanism in the mesh scenario. It is used to connect E-IP (External-IP) client networks under the same I-IP (Internel-IP) backbone (Figure 11). The mechanism is applicable to both IPv4-over-IPv6 and IPv6-over-IPv4 traversing. Unlike 6to4, Softwire Mesh does not have requirements on the E-IP address scheme: the addressing of E-IP and I-IP networks remains independent. As tunnel endpoints, AFBRs on the border of the I-IP network and E-IP client networks form an iBGP mesh to create prefix bindings. By extending the MP-BGP protocol, the AFBRs advertise and receive E-IP routes of client networks across the I-IP backbone. These routes turn into the address bindings between E-IP prefixes and I-IP AFBR addresses on the recipient AFBR. When the AFBR performs I-IP encapsulation, it chooses from the binding table the entry



E-IP Prefix	I-IP Addr
net2 prefix(es)	AFBR2
net3 prefix(es)	AFBR3
net4 prefix(es)	AFBR4

Fig. 11. Softwire Mesh Principle

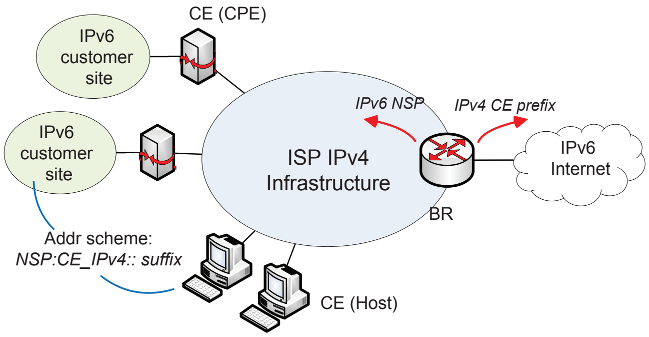


Fig. 12. 6RD principle

that has the longest match with the E-IP destination address, and uses the I-IP address of the entry as the encapsulation destination. The E-IP over I-IP routing is also realized by the iBGP mesh in the identical process.

In Softwire Mesh, each AFBR has to maintain the bindings and is thereby per-prefix stateful. Yet the size of the binding table will be no larger than the size of E-IP forwarding table. Therefore the cost of the binding lookup during encapsulation is acceptable, and the mechanism has good performance and scalability. Besides, Softwire Mesh supports multiple tunnel types which includes IP-IP, GRE, MPLS, L2TP, IPsec, by offering a generic tunnel signaling method in MP-BGP. The mechanism promotes IPv6 development because it enhances IPv6 inter-connectivity over IPv4, and adds the function of IPv4 transport to IPv6. Since iBGP has mature security solutions, the control plane of the mechanism is secure. Spoofing attack could still happen on the data plane, yet it can be significantly reduced by applying E-IP destination verification.

6PE (Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers) [48] adopts the similar mechanism of Softwire Mesh to solve the problem of IPv6 networks traversing IPv4 MPLS transit. The differences between 6PE and IPv6-over-IPv4 Softwire Mesh are: 6PE uses standard MP-BGP to advertise IPv6 routes with IPv6 next hop mapped from AFBR's IPv4 address, and 6PE AFBR uses IPv6-over-MPLS tunnel for forwarding, in which encapsulation is based on the MPLS label of the mapped IPv6 address of remote AFBR. In general this mechanism can be viewed as a customized Softwire Mesh solution to MPLS transit.

C. Host-to-host Tunnel Mechanisms

6over4 (Transmission of IPv6 over IPv4 Domains without Explicit Tunnels) [9] is a tunnel mechanism between hosts, used to achieve IPv6 communication between isolated IPv6-capable hosts across an IPv4-only network. The idea of 6over4 is leveraging IPv4 multicast to build a virtual "LAN" among IPv6-capable hosts. In other words, it is an IPv6 Ethernet-over-IPv4 multicast tunnel. While it does not require any address scheme or binding, the control plane complexity is actually quite high: the host and the network infrastructure have to fully support IPv4 multicast; special efforts are required to enable IPv6 LAN protocols such as SLAAC and ND to work on the virtual LAN. Subsequently the failure mode would be quite complex. On data plane, the multicast forwarding manner causes redundant transmission. A node may receive traffic which is not intended for them. Other than that, data plane performance is not a concern because the tunnel endpoints are on the hosts. The main security risk is the attack on the ND protocol. Attackers from IPv4 may inject unicast ND messages to break the ND process or fake as a 6over4 endpoint. Due to these issues and the limited multicast support in today's ISP network, 6over4 does not seem to have much application prospect.

ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) [49] is another tunnel mechanism for IPv6-capable hosts to communicate across IPv4 networks. Unlike 6over4, ISATAP treats IPv4 as a virtual NBMA (nonbroadcast multiple-access network) data link layer. An ISATAP host uses a link-local IPv6 address which has the fixed prefix of fe80::5efe/96 followed by the 32-bit IPv4 address of the host. The data plane procedure follows typical stateless manner. When encapsulating IPv6 packets between ISATAP hosts, the IPv4 source and destination addresses can be extracted from the IPv6 source and destination. Besides the regular hosts, ISATAP can also have one or more IPv6 gateway routers as tunnel endpoints. These gateway routers can provide tunneled IPv6 access to ISATAP hosts leveraging ND protocol. However, the nonbroadcast nature of the virtual link makes automatic router discovery impossible. Therefore the IPv4 addresses of the gateways have to be configured on the hosts. Then the gateway routers can provide on-link global IPv6 prefixes and subsequently a global IPv6 access to ISATAP hosts. In this case ISATAP turns into a Hub & Spokes tunnel, which is actually the main usage of ISATAP in current deployment.

The performance of ISATAP data plane is good in general, because it is stateless. However, as a Hub & Spokes tunnel mechanism, the control plane complexity is still higher than the new 6RD mechanism which uses the pure layer-3 stateless IP-IP tunnel. ISATAP promotes IPv6 adoption by providing a rapid end user IPv6 deployment in the IPv4 environment. The security risk of ISATAP is similar to 6over4. A malicious IPv4 host can pretend to be part of the ISATAP link and launch attacks.

D. Hub & Spokes Tunnel Mechanisms

Hub & Spokes Softwire [50] adopts the L2TP-over-UDP-over-IP tunnel, to provide an IPv4/IPv6 Internet access to hosts/home networks across an IPv6/IPv4 network. This mechanism can apply to both IPv4-over-IPv6 and IPv6-over-IPv4

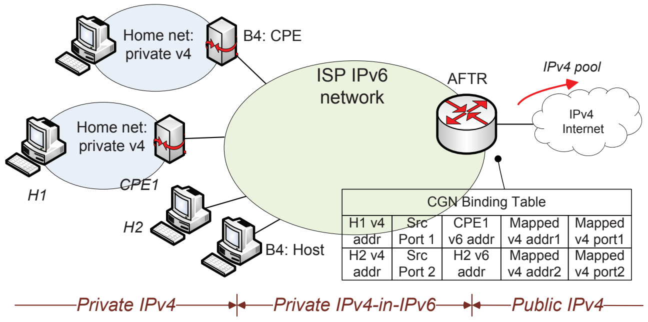


Fig. 13. Dual-stack Lite principle

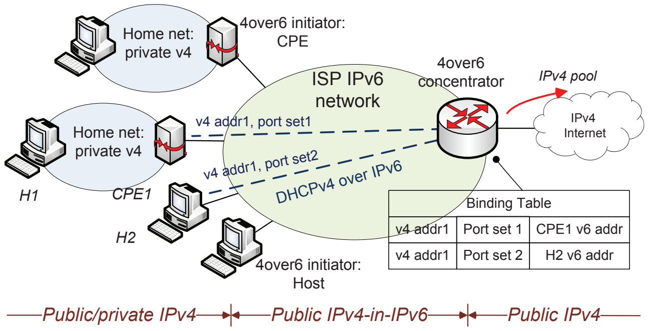


Fig. 14. Lightweight 4over6 principle

scenarios. It uses L2TP on the inner layer of the tunnel to build a virtual layer-2 environment over the tunnel, and subsequently forms a point-to-point IP connection upon it. L2TP/PPP session state maintenance is required, as well as the IPv4-IPv6 address binding which maps outer layer destination addresses into different tunnels. On data plane, UDP is included in the encapsulation header. The initiators would always encapsulate the packet to the concentrator, while the concentrator has to decide the correct initiator as the encapsulation destination, by looking up the stateful address binding. Therefore the data plane of Hub & Spokes Softwire is per-user stateful. However, the control plane is too complex. L2TP and PPP require session state maintenance as well as several rounds of signaling, which may affect the performance in large-scale deployment. It also exposes more vulnerability to attackers. The L2TP tunnel and the PPP sessions may be hijacked, and the PPP signaling may be disrupted. DoS attacks could happen on multiple levels. As a result, recent researches focus on developing mechanisms with pure IP-IP tunnels.

The IP-IP solution to IPv6-over-IPv4 Hub & Spokes scenario is 6RD (IPv6 Rapid Deployment on IPv4 Infrastructures) [51], [52]. 6RD mechanism applies 6to4 tunnel into the Hub & Spokes scenario: it concentrates one endpoint of each tunnel to be the 6RD Border Router (BR) on the IPv4-IPv6 Border, and distributes the other endpoint of each tunnel to the customer CPEs or hosts which are called 6RD Customer Edge routers (CEs) in the context. 6RD uses network specific IPv6 prefix instead of the fixed prefix 2002:IPv4ADDR::/48 in 6to4. The IPv6 customer networks under 6RD CEs must use the prefixes of NSP:CE_IPv4ADDR/, which can be aggregated at the 6RD BR as the NSP, and advertised to the global IPv6 Internet. By doing this, 6RD removes the routing scalability concern in 6to4 and guarantees unique global reachability for customers. Based on the address scheme, the data plane encapsulation on BR is stateless; it does not keep any IPv4-IPv6 address binding. Another benefit of the stateless encapsulation is that, when the communication happens between two CEs, they can build direct tunnels between themselves and thus avoid hairpin on BR (i.e., tunnel from CE to BR and then from BR to the other CE).

The performance of 6RD is quite good because of its statelessness and simplicity. Line-speed data processing is expected on the BR. 6RD promotes IPv6 adoption by providing a rapid end site IPv6 deployment in the IPv4 environment. The security issue of 6RD is spoofing attack on IPv6 from IPv4. Packet amplification attack may arise, which generates traffic of endless loop inside a 6RD domain. The solution is ingress filtering based on the 6RD address scheme.

There is still a particular problem in IPv6-over-IPv4 Hub & Spokes scenario: NAT traversal in the IPv4 network. One or more IPv4 NATs may exist in the middle network. Since 6to4 and 6RD use IP-in-IP encapsulation while typical NAT only allows TCP/UDP/ICMP packets, the encapsulated packets cannot traverse these NATs. Teredo (Tunneling IPv6 over UDP through Network Address Translations) [53] builds a solution to address this problem. It uses IPv6-over-UDP-over-IPv4 encapsulation, with a chosen source port and 3544 as the destination port. A Teredo client (tunnel initiator) adopts a stateless-style address scheme: the 64-bit prefix consists of 2001:0000/32 and the IPv4 Teredo server address, while the interface ID contains the UDP port and IPv4 address mapped by the outermost NAT. A Teredo client learns this IPv4 port and address by interacting with the Teredo server and simultaneously installing the address + port binding(s) on the NAT(s) along the path. The client must refresh the binding(s) by periodically interacting with the server as well. Then two Teredo clients can communicate with each other directly using this type of addresses and the IPv6-over-UDP-over-IPv4 tunnel. During encapsulating, the IPv4 destination address and UDP destination port can be extracted from the IPv6 destination address. As for the communication between Teredo clients and plain IPv6 nodes, Teredo relay is used as the tunnel concentrator. A Teredo relay must advertise reachability of the Teredo prefix to IPv6 Internet. A Teredo client discovers the relay implicitly. It sends an IPv6 ICMP echo request to the IPv6 node, which is tunneled to the server and then the destination. The echo reply will be forwarded to the Teredo relay following IPv6 routing, and tunneled back to the Teredo client. By examining the IPv4 source address and port of the encapsulation packet, the client discovers the relay, and uses it as the encapsulation destination in further forwarding. Therefore, Teredo server only participates in control plane so the load is reduced.

The data plane performance of Teredo is good due to the statelessness. However, the control plane functions for traversing NAT brings too much complexity. They also introduce various failure modes and several security risks including man-in-the-middle attacks at the server or relay, DoS attack, etc.

Therefore, Teredo has much less deployment compared with 6to4 and ISATAP. But it is so far the only IPv6-over-IPv4 solution that survives IPv4 NAT well.

The IPv4-over-IPv6 scenario is a little more complicated. In consideration of the IPv4 address shortage, researchers add the address sharing feature to IPv4-over-IPv6 Hub & Spokes mechanisms. One manner to share IPv4 addresses is CGN (Carrier-Grade NAT) which gathers IPv4 address resources and allocates them dynamically in a per-port style based on realtime flows; the other manner is port-set provisioning which divides every IPv4 address into multiple port-sets and provisions the addresses with restricted port-set to end users. Another bifurcation is stateful encapsulation vs. stateless encapsulation. In stateful encapsulation the concentrator keeps the bindings between the IPv4 and IPv6 addresses of the initiators, while in stateless encapsulation, by leveraging the address scheme of embedding the initiator's IPv4 address into an IPv6 address, binding maintenance is not necessary. Based on these two bifurcation elements, three types of IPv4-over-IPv6 Hub & Spokes solutions are proposed [54].

Dual-stack Lite [55] takes the CGN manner and stateful encapsulation to solve the IPv4-over-IPv6 Hub & Spokes problem. In Dual-stack Lite, the home networks/hosts are required to use private IPv4 addresses, while the AFTR (Address Family Transition Router, the tunnel concentrator) functions as a CGN and performs unified IPv4 private-public translation. Neither IPv4 provisioning nor particular IPv6 address scheme is needed in Dual-stack lite. The AFTR should advertise the prefix of the CGN pool to the IPv4 Internet. In the outbound direction of the data plane, Basic Bridging BroadBand element (B4, IPv6-faced host or Customer Premises Equipment of home network) sends every IPv4 packet to AFTR through the IPv4-in-IPv6 tunnel. AFTR decapsulates the packet, performs the translation, records the binding between the original (private IPv4 address, source port, B4 IPv6 address) and the translated (mapped public IPv4 address, mapped source port) in the CGN table, and then forwards the packet to the IPv4 Internet. In the inbound direction, an IPv4 packet arriving on the AFTR will be translated and encapsulated simultaneously according to the binding in the CGN table, and then forwarded to a B4.

Similar to NAT64, the CGN lookup operation is the performance bottleneck of Dual-stack Lite. In case of software implementation, the processing speed is negatively correlated to the size of the CGN table. As for hardware implementation, the cost and capacity would be positively correlated to the size of the table. The time-delay of creating new bindings could lower the processing speed. Besides, CGN introduces issues like application layer translation and inbound access disability. The gain is a high multiplexing rate of the address resource. Dual-stack Lite preserves IPv4 availability in future IPv6 infrastructure and therefore promotes IPv6 adoption (So is 4over6 and MAP-E). The main security issue is DoS attack on CGN. Considering the address-sharing mode in Dual-stack lite, ISPs have to log 5-tuple binding with timestamp for user traceability, which could become a burden.

4over6 (IPv4 over IPv6 Access Network) chooses the combination of port-set provision and stateful encapsulation. As a basis, Public 4over6 [56] describes the case of full IPv4 address provision. Public 4over6 has no requirement on the IPv6 address scheme. In IPv4 scope, DHCPv4 is modified to run properly in the IPv6 environment, so as to achieve the dynamic IPv4 address allocation from the concentrator to the initiators (IPv6-faced host or home network CPE). When one such allocation is done, the initiator assigns the allocated address as the tunnel inner layer address, while the concentrator records the binding between the allocated IPv4 address and the initiator's IPv6 address. The concentrator should advertise the prefix of the DHCP address pool to the IPv4 Internet. In the outbound direction of the data plane, the concentrator simply decapsulates the packet tunneled from an initiator and forwards it. As for inbound packet encapsulation, the concentrator should look up the IPv6 destination in the address binding table using the IPv4 destination address. Lightweight 4over6 [57] extends Public 4over6 to serve the case of address sharing. The differences from Public 4over6 lie in port-set support: the DHCPv4 process is further extended to support port-set provision, and the binding table on the concentrator includes the port-set information besides the IPv4 address. The two are combined to form an index for encapsulation destination lookup.

The state amount of 4over6 is reduced to a per-user scale rather than a per-flow scale. Therefore the performance would be much better. Besides, 4over6 does not have the ALG issue, and the communication can be fully bidirectional. Although, the multiplexing rate will not be as high as that in Dual-stack lite, and the amount of port for each user is limited. The main security issue of 4over6 is man-in-the-middle attack on DHCP. As a result, ingress filtering and DHCP security solutions should be applied.

As the third type solution, MAP-E (Mapping of Address and Port using Encapsulation) [58] and 4RD (IPv4 Residual Deployment via IPv6) [59] take port-set provision and stateless encapsulation. The principles of the two mechanisms are very similar in essence; we use MAP-E to represent them in the following. MAP-E can be viewed as the mirror image of 6RD in IPv4-over-IPv6 scenario, with the port-set style address sharing. In MAP-E, the Customer Edge routers (CE, IPv6-faced host or CPE of home network) is provisioned with IPv4 address of restricted port-set. Instead of explicit IPv4 provision, CE gets the IPv4 address and port-set from its IPv6 address. This is achieved by encoding the information of IPv4 address and port-set into the CE's IPv6 address and thus coupling IPv6 and IPv4 provision. To be more specific, the CE IPv6 address is composed of a Rule IPv6 prefix, the EA bits which contain IPv4 address suffix and port set index, a subnet ID and a 64-bit interface ID. There can be multiple instances of the address rules, with different IPv6 prefixes, IPv4 prefixes and address sharing ratios. By maintaining all these rules on the BR (Border Relay, tunnel concentrator) and provisioning some rules to CEs, stateless encapsulation can be achieved. Both the CEs and BR are able to calculate the IPv6 encapsulation destination address, based on the rules and the IPv4 destination address and port. When a CE maintains the address rule for a remote CE, it can calculate the encapsulation destination, send packets to the remote CE directly and avoid hairpin on the BR. When the CE does not have this rule, it sends the packet to BR. The BR should advertise the IPv4 prefixes for all the address rules it maintains.

Compared to 4over6 and Dual-stack lite, MAP-E achieves the great benefit of statelessness, so the performance will be good. Line-speed data processing is expected on the BR. However, since the mechanism deeply couples IPv4 and IPv6 addressing, it becomes a little less flexible to deploy: the deployment has to be entire-network style rather than on-demand style, otherwise some of the coupled IPv4 addresses will be wasted. With the combination of ingress filtering and consistency checks on IPv4 and IPv6 address, spoofing attacks and routing-loop attacks will not happen to MAP-E. Traffic hijacking could happen by man-in-the-middle attack on DHCPv6 which provisions the rules. DHCP Security solution should be deployed.

E. Double translation vs. tunneling

Theoretically, IPvY-IPvX-IPvY traversing may also be achieved by one-time IPvY-IPvX translation on ingress tunnel endpoint and one-time IPvX-IPvY translation on egress tunnel endpoint. However, this is infeasible in IPv6-IPv4-IPv6 scenario. While we can turn IPv6 addresses into IPv4 addresses during the first translation, it is impossible to recover these 128-bit IPv6 addresses based on the 32-bit IPv4 addresses in the second translation. As for the IPv4-IPv6-IPv4 scenario, there have been two mechanisms proposed based on double translation.

For the stateless solution to IPv4-IPv6-IPv4 Hub & Spokes, MAP-T (Mapping of Address and Port using Translation) [58] is proposed along with MAP-E. In MAP-T the two translations happen on CE and BR. MAP-T follows the exact address scheme of CEs in MAP-E. There is a tiny difference, however. In MAP-E, for encapsulated IPv6 packets between CE and BR, BR's address is used as IPv6 source or destination. But In MAP-T, the IPv6 packets have to use the IPv4-mapped addresses as source addresses in the inbound direction and destination addresses in the outbound direction. Otherwise the IPv4 address of the remote end will be lost.

For the stateful solution of IPv4-IPv6-IPv4 Hub & Spokes, 464XLAT [60] is proposed, which is similar to Dual-stack lite to some extent. 464XLAT is actually a combination of SIIT on the user side and NAT64 on the carrier side. Hosts or home networks on the user side use private addresses. Take the data plane of the outbound direction as an example, an IPv4 packet is translated into IPv6 on the ”B4” following SIIT mechanism (with a private IPv4 address embedded in an IPv4-translated address), forwarded natively to the ”AFTR”, and translated into IPv4 packet which has a public source address. From end-to-end perspective, 464XLATE achieves the exact effect of Dual-stack Lite.

In these two examples, it seems that tunneling and double translation can replace each other in their respective scenarios. However, detailed differences still exist. While tunneling keeps full transparency of the inner IPv4 by preserving the original packets, translation cannot achieve that. Due to the protocol diversity between IPv4 and IPv6, translation cannot keep full set information in the protocol header. Most of the fields are preserved, whereas the rest are lost, such as ToS, Flags and Identification. On the other hand, double translation exposes a little more information of the inner IPv4 addresses than tunneling, so theoretically it may provide some convenience for operators to perform tasks like traffic engineering.

F. Summary of tunneling mechanisms

The tunneling mechanisms are summarized in Table II. We can see from this table that Softwire Mesh, 6RD, Dual-stack lite, 4over6 and MAP-E are able to form the set of enhanced tunneling mechanisms, which covers most cases of the heterogeneous traversing problem. In particular, for IPv4-over-IPv6 Hub & Spokes scenario, Dual-stack Lite, 4over6 and MAP-E have respective pros and cons. The three mechanisms together fulfill different demands under that scenario. As an exception, Teredo is the only solution to the IPv6-over-IPv4 Hub & Spokes problem with NAT traversal capability.

VI. Mechanism usage & deployment strategy

The former two sections provide an overview of the mainstream IPv6 transition mechanisms that have been proposed. Table III maps the proper mechanisms into all the scenarios of heterogeneous inter-connection and heterogeneous traversing. However, these mechanisms are still not designed in a very practical environment of ISP networks. It still causes confusions for operators when selecting among so many mechanisms and making deployment plans. This section will discuss how to choose and deploy the transition mechanisms.

A. Transition Requirements in ISP networks

A practical ISP network contains ISP backbone and edge networks. The backbone network is usually connected with provider ISPs, customer ISPs, peer ISPs, and edge networks inside the ISP, typically all through BGP. The border routers of the backbone form an iBGP mesh. The scale of the backbone network is usually limited, therefore IPv4 address shortage is not a concern. The routers in the backbone usually have the highest upgrading priority.

An edge network (regional and access network, campus network, etc.) is attached to the backbone in the upward direction and faces end users in the downward direction. The edge network is relatively independent from the backbone and provides the infrastructure services by itself. The edge network has the aggregating characteristic, all along from end users to the backbone entrance. Due to the large population of end users, most often the edge network will not provision public IPv4 addresses freely in the recent future. Typically, a large number of routers, access devices and servers in the existing edge network cannot support IPv6 well. A considerable, costly upgrade is needed to support native IPv6. Besides, although the mainstream operating systems on end user devices are IPv6-ready, not a lot of applications are actually IPv6-capable.

Based on these facts, we summarize the ISP transition requirements as follows:
(1) Provisions of both IPv4 and IPv6 services. The ISP should guarantee that users can reach and be reached by both IPv4 and IPv6 Internet.

TABLE II
SUMMARY OF TUNNELING MECHANISMS

Mechanism	Scenario	Encapsulation manner	Address mapping for Encapsulation	Heterogeneous routing	Issues
6to4	IPv6-over-IPv4 mesh	IP-in-IP	Stateless mapping,IPv4 address embedded in IPv6	BR advertises IPv6 routes for isolated IPv6 islands to IPv6	Routing scalability issue: prefix unable to aggragate
Softwire Mesh	IPv6-over-IPv4/ IPv4-over-IPv6 mesh	IP-in-IP/ GRE/ L2TP/ MPLS/ IPsec	(E-IP prefix, I-IP address) mapping, per prefix	Extend MP-BGP to route E-IP prefixes across I-IP	None
6PE	IPv6-over-IPv4 mesh	IP-in-MPLS	E-IP prefix with I-IP address mapped to MPLS label	Extend MP-BGP to route E-IP prefixes across I-IP	Only apply to MPLS infrastructure
6over4	IPv6-over-IPv4 host-to-host	Ethernet over IPv4-multicast	none	none	need multicast support in IPv4 infrastructure
ISATAP	IPv6-over-IPv4 host-to-host/ Hub & Spokes	NBMA-over-IPv4	stateless mapping with link-local or global prefix	Prefix configuration by ND	More complicated control plane than layer-3 IP-IP
Hub & Spokes Softwire	IPv6-over-IPv4/ IPv4-over-IPv6 Hub & Spokes	L2TP-over-UDP	(E-IP, I-IP) address binding, per user	concentrator advertises E-IP routes for users	Too complicated to manage L2TP-over-UDP tunnel
6RD	IPv6-over-IPv4 Hub & Spokes	IP-in-IP	stateless automatic mapping, IPv4 address embedded in IPv6	BR advertises IPv6 routes for users	none
Teredo	IPv6-over-IPv4 Hub & Spokes with IPv4 NAT traversal	IP-in-UDP-in-IP	stateless mapping, IPv4 address +port embedded in IPv6	Relay advertises IPv6 routes for users	Too complex control plane signaling
Dual-stack Lite	IPv4-over-IPv6 Hub & Spokes (private IPv4 on user side)	IP-in-IP	(IPv6 address, private IPv4 address, port)-(public IPv4 address, port) binding, per flow	AFTR advertises IPv4 routes for users	Application layer translation, per-flow state maintenance, disability of inbound access
4over6	IPv4-over-IPv6 Hub & Spokes (public IPv4 for users)	IP-in-IP	IPv6 address-IPv4 address(+port-set) binding, per user	Concentrator advertises IPv4 routes for users	Address sharing rate not as high as Dual-stack lite, per-user state maintenance
MAP-E	IPv4-over-IPv6 Hub & Spokes (public IPv4 for users)	IP-in-IP	stateless mapping, IPv4 address + port-set embedded in IPv6	IPv6 routing to support coupled addressing; BR advertises IPv4 routes for users	less flexible than 4over6, sharing rate not as high as Dual-stack Lite

(2) Sustainable development. The ISPs are eager for mechanisms which can ease and promote IPv6 adoption, rather than those that will restrain IPv6 development.
(3) Incremental deployment and minimum upgrade. The existing infrastructure is already a huge investment for the ISPs. Incremental deployment and minimum upgrade could significantly save the cost and reduce the operation burden.
(4) Mechanisms should be simple, robust, and easy to deploy.
(5) Mechanisms should have high performance and scalability.
(6) Best-effort transparency to end users and applications. The less the transition mechanism affects end users and applications, the easier it is for end users and ICPs to accept the transition and subsequently cooperate with the transition process.

B. Transition Strategy of Backbone Network

The backbone network is responsible for high-speed IP forwarding, while the specific networks services and subscriber management are charged by the edge network. This is also reflected in transition deployment: the backbone focuses on providing both IPv4 and IPv6 transports to edge networks.

The straightforward approach here is upgrading the backbone to dual stack, or building a standalone IPv6 backbone beside IPv4. The drawback is also obvious: the cost in both hardware upgrade and operation & management is too high.

TABLE III
TRANSITION TECHNIQUES: SOLUTION SPACE

Transition Scenario	Transition Proposal
IPv6 network → IPv4 network	IVI/NAT64
IPv6 network → IPv4 Internet	IVI/NAT64
IPv6 Internet → IPv4 network	NAT64
IPv6 Internet → IPv4 Internet	None
IPv4 network → IPv6 network	IVI
IPv4 network → IPv6 Internet	None
IPv4 Internet → IPv6 network	IVI
IPv4 Internet → IPv6 Internet	None
IPv4 application ⇌ IPv6 Internet	BIS/BIA
IPv6 application ⇌ IPv4 Internet	None
IPv6-over-IPv4 mesh	Softwire Mesh
IPv4-over-IPv6 mesh	Softwire Mesh
IPv6-over-IPv4 Hub &Spokes	6RD, Teredo(NAT traversal only)
IPv4-over-IPv6 Hub &Spokes	Dual-stack Lite, 4over6, MAP-E

The other solution is to deploy Softwire Mesh and provide dual-stack transport on top of the single-stack backbone. For the existing IPv4 backbone, we can deploy IPv6-over-IPv4 mesh. To achieve that we should upgrade the border routers of the backbone to support dual-stack, as well as IPv6-over-IPv4 mesh AFBR functions. IPv6 traffic can be forwarded in the backbone by IPv6-over-IPv4 tunnel between AFBRs. The routers inside the backbone can remain IPv4-only and avoid upgrading. As for IPv6 backbone which will be widely built in the future, we can deploy IPv4-over-IPv6 mesh to preserve IPv4 transport. In case the number of AFBRs gets large, a Softwire-Mesh-friendly router reflector can be set up inside the backbone, to reduce the control plane complexity. The Softwire Mesh solution significantly saves the hardware and operation cost. Only one physical backbone is used while IPv4 and IPv6 transports are achieved simultaneously.

C. Transition Strategy of Edge Network

On the foundation that the backbone provides both IPv4 and IPv6 transports, we then discuss the transition strategy in edge networks. It is known that dual-stack edge network is unpractical due to the cost and user population. Besides, because of the complexity and scalability issues, translation mechanisms should be deployed inside the edge network rather than in the backbone. In the following we provide different transition strategies according to different communication requirements and edge network types.

The IPv4 edge network provides a native IPv4 access to end users. If end users further want IPv6 access, 6RD can be used. End host or CPE should be dual-stack and support 6RD CE functions. On the ISP side, one or more 6RD BRs should be deployed as tunnel concentrators. The IPv6 forwarding path between BRs and backbone entrance can be built by dedicated IPv6 links or stateless tunnel, or even a softwire mesh. BR discovery on CE can be implemented by DHCPv4 extension or out-of-band configuration.

Users in IPv4 edge networks may also want to visit IPv6 using IPv4. Among the transition mechanisms, we only find NAT-PT to satisfy the demand. However, the IPv4→IPv6 direction of NAT-PT is not so feasible. Therefore, we suggest the ISPs not to support this demand, or only provide it in a small-scale, controllable environment.

Unlike existing IPv4 edge networks, new edge networks built in the following years will probably be native IPv6. Then for IPv4 access demands, the ISP could deploy dual-stack lite, 4over6 or MAP-E. This requires end hosts or CPEs to become dual-stack and support these tunnel initiator functions. On the ISP side, one or more tunnel concentrators should be deployed. The IPv4 forwarding path between tunnel concentrators and backbone entrance can be built by dedicated IPv4 links, stateless tunnels, or softwire mesh. Tunnel concentrator discovery can be implemented by DHCPv6 extension or out-of-band configuration. The ISP should choose among the three candidate mechanisms based on its IPv4 address surplus situation and network condition. If high-rate address sharing is the top requirement, or CGN issue is not a big concern for users (e.g., mobile users), then Dual-stack Lite would be a suitable solution. On the other hand, 4over6 and MAP-E are the choices when the ISP has enough IPv4 addresses for port set provisioning. If IPv4 access is a common requirement from subscribers and the ISP is able to renumber the IPv6 network, then MAP-E would be a better choice with the great advantage of statelessness. If the ISP wants to keep IPv6 and IPv4 uncoupled and bring no modifications to the IPv6 network, or provide the IPv4 access in an on-demand style, then 4over6 becomes the preferred solution.

Users in IPv6 edge networks may also want to visit IPv4 with IPv6. This requires IPv6-IPv4 translation, which could be realized by IVI or NAT64. To support these two mechanisms, the ISP should deploy one or more translators in the network, and add DNS64 functions to its DNS system. When choosing between IVI and NAT64, the criteria also lies in IPv4 address surplus situation. IVI provides bidirectional communication at the cost of per-host address consumption, while NAT64 only guarantees communication initiated from the IPv6 side but achieves dynamic IPv4 address sharing.

Besides the common users, demands from ICP servers should also be considered. The difference between a server and a common user is that connections of servers are usually initiated from the remote end, so it is of great convenience if the server possesses a full IPv4 address. The sever should provide services for both IPv4 and IPv6 clients. The IP layer support to that is dual-stack access provided by the Public 4over6 or MAP-E. If the server supports both IPv4 and IPv6 on the application layer, then no further mechanism is required; otherwise application level transition support should be provided. If the server runs on IPv4, we can deploy a NAT64 translator in front of the ICP network with its IPv4 side facing the server, to translate connections from IPv6 clients into IPv4; otherwise the server runs on IPv6, and we can deploy an IVI translator in front of the ICP network, with its IPv6 side facing the server, to translate connections from IPv4 clients into IPv6. ALG should be carefully customized in these cases.

D. Typical Case Study

1) CERNET2 backbone: CERNET (China Education and Research Network) is one of the earliest ISPs which have activated IPv6. The IPv6 project of CERNET was launched in Sept 2003 and has been providing IPv6 transport service for campus networks since Jun 2004. Currently, the CERNET2 backbone is a pure IPv6 network containing 20 PoPs and 1 IXP. The clients include over 100 campus networks.

A CERNET campus network provides native dual-stack access for end users, with the IPv4 and IPv6 campus gateways separated. The IPv4 gateway connects to CERNET IPv4 backbone, while the IPv6 gateway connects to CERNET2 IPv6 backbone. It is of great significance that CERNET2 can provide IPv4 transport besides IPv6 transport. On one hand, the operator can transfer a portion of IPv4 traffic from CERNET to CERNET2, reducing the load of CERNET IPv4 backbone and leveraging the CERNET2 infrastructure. On the other hand, in a long term CERNET2 backbone will replace CERNET backbone and become the major backbone eventually. At that time, it will be a basic requirement of CERNET2 to support IPv4 transport.

CERNET2 adopts Softwire Mesh to satisfy this demand. It builds an IPv4-over-IPv6 mesh on the IPv6 backbone to achieve connectivity between IPv4 campus networks. To be more specific, the operator assigns the border routers of the IPv6 backbone as the mesh AFBR routers, and attaches each IPv4 campus gateway to one AFBR router. Also, the operator upgrades a router in CERNET2 to be an AFBR and connects it with a CERNET ISP border router, which acts as Internet traffic exit. Then, by the additional plan of IPv4 routing, IPv4 traffic between campus networks can be forwarded through tunnels between campus-faced AFBRs, and IPv4 traffic between campus networks and the Internet can be forwarded through tunnels between campus-faced AFBRs and CERNET-faced AFBR.

2) dual-stack access in IPv6 telecom networks: Due to the shortage on IPv4 address resource, IPv6 edge network seems to be the inevitable solution to telecom ISPs. Chinese ISPs including China Telecom, China Mobile and China Unicom have claimed to fully deploy IPv6 in one or two years, and China Telecom has already upgraded the network to support IPv6 in two provinces. When deploying IPv6, they still have to provide IPv4 reachability for the subscribers, to guarantee connectivity with IPv4 remote ends. In particular, a lot of application services will remain IPv4 for a while.

There are two directions to solve the problem here: one is to provide IPv6-to-IPv4 translation, and have the subscriber applications use IPv6 for communication with IPv4; the other is to provide IPv4 access for subscribers through IPv4-in-IPv6 Hub & Spokes tunnel, and preserve the applications to use IPv4 for this type of communication. Amongst all the applications, not a large portion of them are IPv6-capable while the majority of them support IPv4 well. That is to say, currently IPv4 is much more universal than IPv6 in the application field. Besides, translation still has the complexities of heterogeneous addressing and application layer gateways. Therefore, IPv4-over-IPv6 tunnel is preferred as the basic solution. By deploying tunneled IPv4 access, ISPs can actually develop IPv6 edge networks while preserve subscribers' IPv4 demands, and hence promote IPv6 development.

As is described in the former section, Dual-stack Lite, 4over6 and MAP-E can be adopted to provide IPv4-over-IPv6 access. ISPs can select the suitable mechanism based on their IPv4 address surplus situation and network condition. The AFTR/concentrator/BR devices have multiple candidate deployment locations, including BRAS which is a low-position access device, P router in the regional network, and backbone entrance which connects directly to the backbone PE router. ISPs can decide the location based on management policy and device capacity.

VII. Conclusion & forecast

Given that IANA has eventually run out IPv4 address space, the Internet is bound to enter the IPv6 era. Nevertheless, IPv4 networks will coexist with IPv6 networks for a long time during the transition. The IPv6 transition process should be steady and smooth. Therefore, the IPv4-IPv6 coexisting networks should sustain the availability of both IPv4 and IPv6, and support IPv4-IPv6 interconnection as well.

This paper analyzes the basic problem of heterogeneous traversing and heterogeneous interconnection in IPv6 transition, introduces the principle of tunneling and translation techniques, and reviews the mainstream tunneling and translation mechanisms. The aspects of address scheme and routing, heterogeneous addressing, data forwarding, performance, security and scalability are studied for these mechanisms. The paper also summarizes the pros and cons, and subsequently application scenarios of every mechanism.

A series of mechanisms including Softwire Mesh, 6RD, DS-Lite, 4over6, MAP, IVI and NAT64 are recommended as feasible solutions to filling in their respective application scenarios. Based on these recommendations, this paper studies the characteristics and transition requirements of practical ISP networks, and proposes the transition strategies for both backbone and edge networks by selecting and deploying the recommended mechanisms.

The transition techniques are still facing challenges and require further research efforts. For translation techniques, the most critical issue is the lack of feasible, stateful IPv4→IPv6 translation mechanisms. Unfortunately, based on the current understanding of IPv4-IPv6 translation, this problem seems unlikely to be solved. We need to find a new angle to develop a solution. As for the existing translation mechanisms, there are still the issues of scalability, heterogeneous addressing and application layer translation. We probably could avoid the scalability issue by choosing rational positions to deploy translation. Solving the heterogeneous addressing problem requires the end users to perceive the existence of the translation and interact with the translator. For communication initiated from the IPv6 side, the IPv6 end should learn the prefix for the IPv4-mapped address, and construct the IPv6 destination address itself when only the IPv4 address of the remote end is informed. For communication initiated from the IPv4 side, in NAT64, the IPv6 end should interact with the translator to create NAT binding(s) and inform of the IPv4 end the information. However, both proposals require extra intelligence in the protocol stack and the applications. Better solutions are expected. Application layer translation is also a problem that seriously affects the performance of the translation. The existing proposal is to transfer the responsibility to end user applications, which also relies on the intelligence of the application. For tunneling techniques, mechanisms like 4over6 and MAP-E change the provisioning granularity from a full address into a port set. A New address resource management model are required to achieve good address resource utilization.

Before the transition techniques can be applied to large-scale deployment, systematic and quantitative performance analyses should be carried out first. The important tasks include estimation on the hardware cost of implementing tunneling and translation, evaluation on the performance reduction caused by frequent fragmentation and reassembling, capacity analysis of CGN and concentrator devices regarding the number of users and the traffic volume, etc. Redundancy backup schemes and security schemes for different transition mechanisms also need to be further explored. New log solutions are demanded for CGN-based mechanisms like NAT64 and Dual-stack lite.

The transition techniques also have impacts on end hosts and ICPs. When both IPv4 and IPv6 are available, IP stack selection, switching method and end-to-end negotiation method need to be proposed. The appearance of CGN in NAT64 and Dual-stack lite would widely break the end-to-end property for network users, which requires evaluation on the influence and guidelines for developing applications in such an environment. With the transition mechanisms deployed in the network, ICPs should adjust their services to various levels to cooperate with the transition mechanisms. For example, they may need to coordinate their CDNs based on the deployment of transition mechanisms; they need to find a solution to construct cross-IP overlay networks.

From a more general view, IPv6 itself also requires significant efforts to achieve global adoption in the near future. The IPv6 support of the network infrastructures needs to be strengthened, to survive large-scale commercial deployment. Various network elements require further efforts from the community, such as IPv6 access provisioning, IPv6 AAA/DNS/DHCP solution, network management solution, IPv6 VPN support, IPv6-specific security protection, etc. At the moment, a lot of the IPv6 protocol implementations in network devices and end host OS are still of low quality. Comparing with IPv4, some existing IPv6 networks even suffer in QoS regarding factors like end-to-end delay and bandwidth. Besides, the IPv6 support in the network application world is far from mature and universal. Only a small portion of mainstream applications, such as YouTube, part of Google services can actually support IPv6; most applications are not IPv6-ready. Although the use of IPv4-over-IPv6 mechanisms can temporarily relieve the upgrading pressure, fundamentally we still need the applications to accept IPv6, the sooner the better.

During the IPv6 transition process, the above problems are the essential challenges that need to be overcome. They are all non-neglectable problems in promoting IPv6, and hopefully they are solvable with the combination of techniques and business means. With the continuous development of IPv6 techniques, IPv6 transition techniques, and the step-by-step follow-ups of vendors, ISPs, ICPs and end users, IPv6 will finally accomplish the transition process and take charge of the future Internet.

REFERENCES

[1] I. S. I. at University of Southern California, “Internet Protocol, DARPA Internet Program, Protocol Specification,” 1981, IETF RFC 791.

[2] “Internet Usage Statistics,” Miniwatts Marketing Group, Tech. Rep., Jun. 2011. [Online]. Available: http://www.internetworldstats.com

[3] G. Huston, “IPv4 Address Report,” Tech. Rep., Sep. 2010. [Online]. Available: http://www.potaroo.net/tools/ipv4

[4] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” 1998, IETF RFC 2460.

[5] S. Thomson, T. Narten, and T. Jinmei, “IPv6 Stateless Address Autoconfiguration,” 2007, IETF RFC 4862.

[6] C. Perkins, D. Johnson, and J. Arkko, “Mobility Support in IPv6 ,” 2011, IETF RFC 6275.

[7] “IPv6 Adoption Monitor,” University of Pennsylvania, Comcast and Tsinghua University, Tech. Rep., 2010. [Online]. Available: http://mnlab-ipv6.seas.upenn.edu/monitor

[8] B. Carpenter and K. Moore, “Connection of IPv6 Domains via IPv4 Clouds,” 2001, IETF RFC 3056.

[9] B. Carpenter and C. Jung, “Transmission of IPv6 over IPv4 Domains without Explicit Tunnels,” 1999, IETF RFC 2529.

[10] G. Tsirtsis and P. Srisuresh, “Network Address Translation - Protocol Translation (NAT-PT),” 2000, IETF RFC 2766.

[11] E. Nordmark, “Stateless IP/ICMP Translation Algorithm(SIIT),” 2000, IETF RFC 2765.

[12] K. Tsuchiya, H. Higuchi, and Y. Atarashi, “Dual Stack Hosts using the ”Bump-In-the-Stack” Technique (BIS),” 2000, IETF RFC 2767.

[13] M. Tatipamula, P. Grossetete, and H. Esaki, “IPv6 integration and coexistence strategies for next-generation networks,” *IEEE Commun. Mag.*, vol. 42, no. 1, pp. 88 – 96, jan 2004.

[14] D. Waddington and F. Chang, “Realizing the transition to IPv6,” *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 138 –147, jun 2002.

[15] A. Durand, “Deploying IPv6,” *IEEE Internet Computing*, vol. 5, no. 1, pp. 79–81, jan/feb 2001.

[16] J. J. Amoss and D. Minoli, *Handbook of IPv4 to IPv6 Transition: Methodologies for Institutional and Corporate Networks*. Auerbach Publications, 2007.

[17] M. Mackay and C. Edwards, “A Managed IPv6 Transitioning Architecture for Large Network Deployments,” *IEEE Internet Computing*, vol. 13, no. 4, pp. 42 –51, july-aug. 2009.

[18] Silva Hagen, *IPv6 Essentials*, 2nd Edition ed. O’Reilly Media, 2006.

[19] R. Hinden and S. Deering, “IP Version 6 Addressing Architecture,” 2006, IETF RFC 4291.

[20] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” 2007, IETF RFC 4861.

[21] S. Sen, Y. Jiny, R. A. Guerin, and K. Hosanagar, “Modeling the Dynamics of Network Technology Adoption and the Role of Converters,” *IEEE/ACM Trans. Netw.*, vol. 18, pp. 1793 – 1805, 2010.

[22] “IETF Behave WG charter,” http://datatracker.ietf.org/wg/behave/charter/.

[23] “IETF Softwire WG charter,” http://datatracker.ietf.org/wg/softwire/charter/.

[24] X. Li, S. Dawkins, D. Ward, and A. Durand, “Softwire Problem Statement,” 2007, IETF RFC 4925.

[25] F. Baker, X. Li, C. Bao, and K. Yin, “Framework for IPv4/IPv6 Translation,” 2011, IETF RFC 6144.

[26] X. Li, C. Bao, and F. Baker, “IP/ICMP Translation Algorithm,” 2011, IETF RFC 6145.

[27] X. Li, C. Bao, M. Chen, H. Zhang, and J. Wu, “The CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition,” 2011, IETF RFC 6219.

[28] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li, “IPv6 Addressing of IPv4/IPv6 Translators,” 2010, IETF RFC 6052.

[29] M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum, “DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers,” 2011, IETF RFC 6147.

[30] X. Li, C. Bao, and H. Zhang, “Address-sharing stateless double IVI,” 2011, IETF draft.

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

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

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

[34] D. Wing, S. Cheshire, M. Boucadair, R. Penno, and F. Dupont, “Port Control Protocol (PCP),” 2011, IETF draft.

[35] P. Wu and Y. Cui and M. Xu and J. Wu and X. Li and C. Metz and S. Wang, “PET: Prefixing, Encapsulation and Translation for IPv4-IPv6 Coexistence,” in *GLOBECOM*, 2010.

[36] P. Wu, Y. Cui, M. Xu, J. Dong, and C. Metz, “Flexible Integration of Tunneling and Translation for IPv6 Transition,” *Networking Science*, vol. 1, pp. 23 – 33, 2012.

[37] C. Perkins, “IP Encapsulation within IP,” 1996, IETF RFC 2003.

[38] S. Hanks, T. Li, D. Farinacci, and P. Traina, “Generic Routing Encapsulation (GRE),” 1994, IETF RFC 1701.

[39] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Generic Routing Encapsulation (GRE),” 2000, IETF RFC 2784.

[40] J. Lau, M. Townsley, and I. Goyret, “Layer Two Tunneling Protocol - Version 3 (L2TPv3),” 2005, IETF RFC 3931.

[41] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture,” 2001, IETF RFC 3031.

[42] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta, “MPLS Label Stack Encoding,” 2001, IETF RFC 3032.

[43] S. Kent and R. Atkinson, “Security Architecture for the Internet Protocol,” 1998, IETF RFC 2401.

[44] Y. Cui, J. Dong, P. Wu, J. Wu, C. Metz, Y. Lee, and A. Durand, “Tunnel-based IPv6 Transition,” *IEEE Internet Computing(accepted)*.

[45] Y. Cui, J. Wu, X. Li, M. Xu, and C. Metz, “The Transition to IPv6, Part II: The Softwire Mesh Framework Solution,” *IEEE Internet Computing*, vol. 10, pp. 76 – 80, 2006.

[46] J. Wu, Y. Cui, C. Metz, and E. Rosen, “Softwire Mesh Framework,” 2009, IETF RFC 5565.

[47] J. Wu, Y. Cui, X. Li, and C. Metz, “The Transition to IPv6, Part I: 4over6 for the China Education and Research Network,” *IEEE Internet Computing*, vol. 10, pp. 80 – 85, 2006.

[48] J. D. Clercq, D. Ooms, S. Prevost, and F. L. Faucheur, “Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE),” 2007, IETF RFC 4798.

[49] F. Templin, T. Gleeson, and D. Thaler, “Intra-Site Automatic Tunnel Addressing Protocol (ISATAP),” 2008, IETF RFC 5214.

[50] B. Storer, C. Pignataro, M. D. Santos, B. Stevant, L. Toutain, and J. Tremblay, "Softwire Hub and Spoke Deployment Framework," 2009, IETF RFC 5571.

[51] R. Despres, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)," 2010, IETF RFC 5569.

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

[53] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," 2006, IETF RFC 4380.

[54] Y. Cui, P. Wu, M. Xu, J. Wu, Y. Lee, A. Durand, and C. Metz, "4over6: network layer virtualization for IPv4-IPv6 coexistence," *IEEE Network*, vol. 26, pp. 44 – 48, 2012.

[55] A. Durand, R. Droms, J. Woodyatt, and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion," 2011, IETF RFC6333.

[56] Y. Cui, J. Wu, P. Wu, C. Metz, O. Vautrin, and Y. Lee, "Public IPv4 over Access IPv6 Network," 2011, IETF draft.

[57] Y. Cui, J. Wu, P. Wu, Q. Sun, C. Xie, C. Zhou, Y. Lee, and T. Zhou, "Lightweight 4over6 in access network," 2011, IETF draft.

[58] O. Troan and W. Dec and X. Li and C. Bao and Y. Zhai and S. Matsushima and T. Murakami, "Mapping of Address and Port (MAP)," 2012, IETF draft.

[59] R. Despres and R. Penno and Y. Lee and G. Chen and S. Jiang , "IPv4 Residual Deployment via IPv6 - a unified Stateless Solution (4rd)," 2012, IETF draft.

[60] M. Mawatari and M. Kawashima and C. Byrne , "464XLAT: Combination of Stateful and Stateless Translation," 2012, IETF draft.

Peng Wu received the BE degree from Tsinghua University, China in 2008. He is currently a Ph.D. candidate in the Department of Computer Science and Technology, Tsinghua University. His current research interests include IPv4-IPv6 transition and next-generation Internet.

Yong Cui received the BE degree and the Ph.D. degree from Tsinghua University, China in 1999 and 2004, respectively. He is currently an associate professor in Tsinghua University, Council Member in China Communication Standards Association, Co-Chair of IETF IPv6 Transition WG Softwire. Having published over 80 papers in refereed journals and conferences, he received the National Science and Technology Progress Award (Second Class) in 2005, the Influential Invention Award of China Information Industry in 2004, and two best paper awards in ACM ICUIMC 2011 and WASA 2010. Holding more than 40 patents, he is one of the authors in RFC 5747 and RFC 5565 for his proposal on IPv6 transition technologies. His major research interests include mobile wireless Internet and computer network architecture.

Jianping Wu received his B.S., M.S., and Ph.D. from Tsinghua University, China. He is currently a Full Professor in Tsinghua University, Vice Chairman of the Information Committee, Director of the Information Office, Dean of the CS Department and Director of the Network Research Center, Dean of Institute for Network Sciences and Cyberspace, Director of Information Technology Center, Tsinghua University. He is director of Network Center and Technic Committee of China Education and Research Network CERNET, director of the National Engineering Laboratory for Next Generation Internet, a member of Advisory Committee of National Information Infrastructure for Secretariat of State Council of China, and Vice President of Internet Society of China (ISC). He is the IEEE Fellow and was also the Chairman of Asia Pacific Advanced Network from 2007-2011. He received the Jonathan B. Postel Award from the Internet Society in 2010. His research interests include next-generation Internet, IPv6 deployment and technologies, Internet protocol design and engineering.

Jiangchuan Liu (S'01-M'03-SM'08) received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is a recipient of Microsoft Research Fellowship (2000), Hong Kong Young Scientist Award (2003), and Canada NSERC DAS Award (2009). He is a co-recipient of the Best Student Paper Awards of IWQoS'2008 and 2012, the Best Paper Award (2009) of IEEE ComSoc Multimedia Communications Technical Committee, and the Best Paper Award of IEEE Globecom 2011. He is currently an Associate Professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada, and was an Assistant Professor in the Department of Computer Science and Engineering at The Chinese University of Hong Kong from 2003 to 2004. His research interests include multimedia systems and networks, wireless ad hoc and sensor networks, cloud computing, and peer-to-peer and overlay networks. He is a Senior Member of IEEE and a member of Sigma Xi. He is an Associate Editor of IEEE Transactions on Multimedia, and an editor of IEEE Communications Surveys and Tutorials. He is TPC Vice Chair for Information Systems of IEEE INFOCOM'2011.

Chris Metz is a Principal Engineer in the Service Provider CTO Architecture Group for Cisco Systems based in San Jose, California. His current areas of interest include Internet architectures and services, IP/MPLS, IPv6 transition, Software Defined Networking and network analytics. He has spoken at industry conferences worldwide on topics involving IP protocols and architectures. He is actively involved in various standards bodies including the IETF where his recent efforts have been devoted to Softwires, BEHAVE and v6ops. He helped architect and deliver a Cisco solution for SP-class CGN and IPv6 Transition features on the CRS-1 router. More recently he has collaborated in architecting and prototyping SDN and analytics solutions for service providers. He is the coauthor of "ATM and Multiprotocol Networking," McGraw-Hill, 1997 and the author of "IP Switching: Protocols and Architectures," McGraw-Hill, 1999. He is a member of ACM/Sigcomm, IEEE and was a contributing editor to IEEE Internet Computing between 1999 and 2010. He also served as the Internet area editor for IEEE Communcations Surveys and Tutorials between 2004 and 2010. He hold 6 patents and prior to joining Cisco in 1998, spent 14 years with IBM Corp.