Science in China Series F: Information Sciences

www.scichina.com
info.scichina.com
www.springerlink.com

IETF softwire unicast and multicast framework for IPv6 transition

CUI Yong[1†], XU MingWei[1], WU JianPing[1] & LI Xing[2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;
[2] Networking Center, Tsinghua University, Beijing 100084, China

IPv6 protocol plays an important role in the next generation of Internet (NGI). It is expected that the elegant coexistence of IPv4 and IPv6 is the key point of IPv6 transition. To solve the transition problem, we propose a mesh unicast framework and a multicast framework in this paper. We describe two reference models for the mesh unicast framework, and put forward two potential solutions for the multicast framework. A Linux-based prototype is implemented for IPv4 over IPv6 scenario and a test bed is deployed with 8 nodes on CERNET2. The deployment demonstrates the advantages of the framework.

IPv6, transition, mesh, routing, BGP

1 Introduction

Due to the shortcomings of IPv4 protocol[1], such as the shortage of address space[2], it is expected that IPv4 protocol will be replaced by IPv6 protocol[3]. Service providers have done some trials to build IPv6 network and thus to provide IPv6 access (such as CERNET2[4]). However, the IPv6 transition still has a long way to go, since there are huge numbers of end users and network devices that rely on IPv4 and the switch from IPv4 to IPv6 cannot be completed overnight[5,6]. It is commonly believed that the transition will take 3 steps: IPv6 islands with IPv4 backbone, IPv4 islands with IPv6 backbone and native IPv6 networks. The first two steps will face the IPv4 and IPv6 coexistence situation.

By far, the transition problem for IPv6-over-IPv4 has been studied a lot. There are several schemes to solve this problem, including the using of multicast in core IPv4 network and automatic tunneling using compatible addresses. However, the IPv6 addresses cannot be embedded in IPv4 addresses, so the automatic tunneling scheme for IPv6-over-IPv4 cannot be used in IPv4-over-IPv6. Moreover, the schemes are used to deal with such cases where only small edge IPv6 networks connect to the core IPv4 network, so there are no scalability considerations. The

Received June 3, 2008; accepted August 21, 2008
doi: 10.1007/s11432-008-0152-8
†Corresponding author (email: cy@csnet1.cs.tsinghua.edu.cn)
Supported by the National Basic Research Program of China (Grant No. 2003CB314801), the National High-Tech Research & Development Program of China (Grant No. 2006AA01Z205), and the National Natural Science Foundation of China (Grant Nos. 90604024 and 90704001)

framework and solutions we proposed are aimed at solving both 6over4 and 4over6 problems with high scalability, which means that large domains can connect with a lot of other domains across a backbone with a different address family and the routing information from client networks should not influence the core network.

As for the development of IPv6 network, Tsinghua University promoted the foundation of a new IETF Working Group (WG) called Softwire[7], which was dedicated to handle the IPv4/v6 transition problem. Tsinghua University summarized the Softwire Problem Statement as RFC 4925[8], dividing the problem into "Hubs and Spokes" problem and "Mesh" problem. More specifically, for the "Mesh" problem, we are also specifying the standardization of discovery, control and encapsulation methods for connecting IPv4 networks across IPv6-only networks and IPv6 networks across IPv4-only networks in a way that will encourage multiple, inter-operable vendor implementations. The problem we focus on is how to support a general routed topology, in which a backbone network not supporting a particular address family can be used as a part of the path for packets belonging to that address family.

The mesh framework is divided into two parts: the unicast framework and the multicast framework. As for the unicast framework, we propose two reference models: Softwire mesh reference model (SMRM) and AFBR reference model (ARM). SMRM defines the basic entities of the Softwire and the transmission of the control message and data. The ARM defines the functional structure of AFBR that performs routing process and encapsulation/decapsulation processes. In addition, we design a light-weighted and highly scalable full-mesh transition mechanism called 4over6 based on BGP to handle the IPv4/6 transition problem. Also, we propose two solutions for multicast framework. One is for the case where the transit core does not support multicast, so we have to do it in a unicast way; and the other is for the case where the transit core is multicast-enabled.

In order to evaluate the framework we propose, we implement a Linux-based prototype system for 4over6 transition problem, which uses standard IP-in-IP tunneling technique in transferring data and MP-BGP to announce route messages.

The rest of this paper is organized as follows. The related work is summarized about IPv4/IPv6 transition in section 2. We propose the softwire mesh unicast framework and multicast framework in section 3 and section 4, respectively. We describe the implementation and deployment in section 5, and finally conclusions are drawn in section 6.

2 Related work

During the past years, many transition techniques have been proposed. According to different deploying scenarios, they can fall into 3 parts, namely IPv6 interconnection based on IPv4, IPv4 interconnection based on IPv6 and the interconnection between IPv4 and IPv6. From the technical view, these transition techniques can be dual stack, tunneling and protocol translation.

As the simplest transition method, the dual-stack technique is also the fundamental technique of all the transition mechanisms. It is suitable for the communication among IPv4 or IPv6 nodes at the same time. Since all the equipments in the dual-stack network employ both IPv4 and IPv6, it is impossible to update the enormous IPv4-only network all around the world.

Tunneling transition technique makes advantages of tunneling technology by encapsulating one kind of packet into another kind. The former is the payload of the transmission and the latter is the media of the transmission. There are different tunneling transition techniques such as con-

figured tunneling, automatic tunneling and MPLS tunneling.

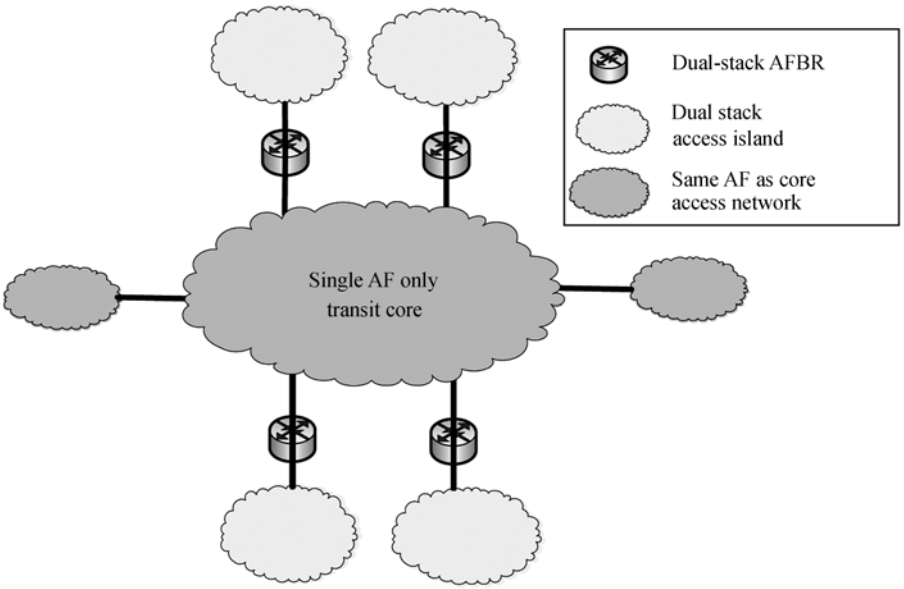

Figure 1 Mesh problem topology.

Configured tunneling is one of the IPv6-over-IPv4 tunneling technologies where the IPv4 tunnel endpoint address is set by configuration information on the encapsulating node. The tunnel can be either unidirectional or bidirectional. Bidirectional configured tunnels behave as virtual point-to-point links, such as IPv6 in IPv4 tunneling[9], GRE over IPv4 tunneling[10]. In automatic IPv6-over-IPv4 tunneling technology, the IPv4 tunnel endpoint address is determined from the IPv4 address embedded in the IPv4-compatible destination address of the IPv6 packet being tunneled such as Automatic 6to4[11], ISATAP[12]. Another scheme for IPv6 over IPv4 tunneling is using multicast in IPv4 core network[13], so the attached IPv6 networks can treat the IPv4 transit network as a LAN. The shortcomings of this scheme are obvious: the infrastructure in IPv4 network has to support multicast, which is not the situation for most of today's networks. In MPLS tunneling, L2/L3 VPN[14,15] is employed to connect IPv6 islands. They can be based on CE (customer edge), PE (provider edge), MPLS circuit and IPv6 MPLS.

Protocol translation techniques such as NAT-PT[16] and transport relay translator[17] are designed for the communication between the nodes with different address families. These transition techniques are mainly divided into network layer translation, transport layer translation, application layer translation and application layer interface translation. Just like NAT in IPv4, protocol translation technique is of low performance and reliability.

Among the translation techniques, some are proposed for multicast applications[18−20]. The basic requirement is that there should be at least one protocol conversion device in the path of communication that connects nodes with different address families. There are several typical mechanisms such as IPv4/v6 multicast reflector, multicast translator proxying, IPv4-IPv6 multicast gateway, 6to4 multicast extension and transition multicast solution (TMS). Some are based on manual configuration without scalability and manageability, and some are of low performance and efficiency. Most importantly, these schemes are designed only for IPv6-over-IPv4 situation that belongs to the first step of transition process.

3 Softwire mesh unicast framework

We will define the mesh problem and illustrate the softwire mesh and AFBR reference models in this section and describe the entities of each in section 4. These models are fundamental for our 4over6 architecture.

3.1 Mesh problem

The "mesh" problem is named in reference to typical routing problems, in which there are multiple paths to a destination and a routing protocol is needed to select the best path. In this problem, carriers may be required to establish connectivity to "islands" of networks of one address family type across a transit core of a different address family type.

Taking Figure 1 as an example, to provide reachability across the transit core, dual-stack devices are installed as "address family border routers (AFBRs)". These AFBRs can perform peering across autonomous systems or perform as Provider Edge routers (PE) in VPN parlance within an autonomous system. With respect to deployment considerations, the islands do not have to be upgraded at the time of deploying the transit core and interwork as if there was no awareness of the AFBR.

3.2 Softwire mesh reference model

Softwires are established between dual-stack AFBRs using softwire signaling. Client network reachability and BGP next-hop information is exchanged between AFBRs.

Note that the remote endpoint of the softwire used to carry a particular packet will be the BGP next hop for that packet. AFBRs will also peer with routers in the client networks to exchange AF(i) routing information. Packets composed of a payload and an IP header are termed the SPH flow across the single AF transit core in softwires encapsulated with an AF(j)-based STH. Note that this reference model is not different from the reference model one could give for any case of tunneling through a "BGP-free core".

The entities of the reference model are:

● AF(i), AF(j) and AF(i,j). Address families c and b are generally used when i=4 and j=6 or when i=6 and j=4, but it is not important to distinguish them. AF(i,j) is the notation used to indicate that a node is dual-stack or that a network is composed of dual-stack nodes.

● AF(i)-only transit core (backbone). This is an IPv4 or IPv6 backbone network surrounded by a periphery of dual-stack AFBR routers.

● AF(j)-only or AF(i,j) client networks. Client networks can be AF(i)-only or dual-stack AF(i,j). In either case, they rely on the transit core for connectivity to other client networks with which they have an AF in common. Each client network must have at least one router that peers with an AFBR to exchange routing information.

● Address family border routers (AFBR). AFBR nodes exchange routing information with each other (including the AF(i) routing information) and engage in any signaling necessary in setting up the needed set of softwires.

● Softwire signaling. This is the signaling needed in setting up the softwires, if any. Whether signaling is needed depends on the particular type of tunneling technology used to instantiate the softwires.

3.3 ABFR reference model

As shown in Figure 2, the reference model for a dual-stack softwire AFBR includes the following:

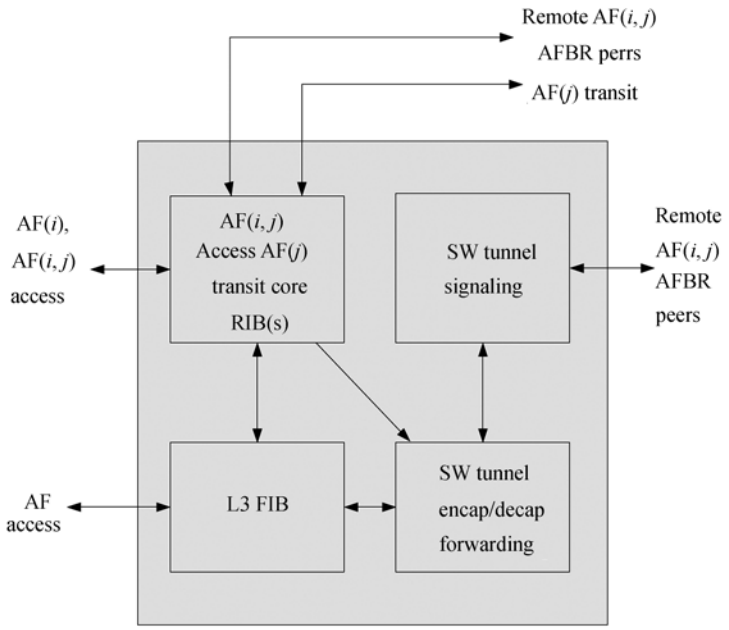

Figure 2 Softwire AFBR reference model.

● SW signaling module. This module is responsible for engaging in whatever signaling, if any, necessary in setting up and maintaining the inter-AFBR softwires.

● AF routing information base (RIB). This entity represents one or more routing information bases (RIB) needed to store AF reachability information received over the AFBR's multiple peering relationships.

● AF forwarding information base (FIB). This entity represents one or more forwarding information bases (FIB) computed from the RIB(s) and needed to forward the packets to and from the client networks and out of the softwire tunnels.

● SW tunnel encap/decap and forwarding. This entity represents the softwire encapsulation and decapsulation processes performed at the ingress and egress AFBR respectively as well as the lookup and forwarding of the packet based on the STH. This does NOT specify the implementation of a softwire but rather illustrates the basic function blocks that run in the dual-stack, softwire-capable AFBR.

3.4 Mesh 4over6 architecture

To address the transition problem described in the introduction, we propose an IPv4 networking-interconnection mechanism called 4over6. Generally, our 4over6 solution is mainly composed of two parts: control plane and data plane. The control plane is based on border-gateway protocol and designed to advertise 4over6 tunnels and IPv4 network prefixes. It is in charge of maintaining routing and encapsulation information. The stateless softwire tunnels are built between PEs that exchange the IPv4 network layer reachability information (NLRI) by MP-BGP. The data plane uses standard IP encapsulation and decapsulation performed at IPv4-IPv6 dual-stack routers.

We realize IPv4 network interconnection by using routing transport on the control plane and packet transport on the data plane. Because it avoids explicit tunnels and manual configuration, the 4over6 mechanism is light-weighted, adaptive to dynamic routes, and transparent for network

end systems. It is designed for the purpose of interconnecting large-scale networks because of its high scalability and easiness of deployment.

4 Softwire mesh multicast framework

In this section we explain the softwire multicast in detail. The 4over6 topology means that some IPv4 customer networks are interconnected by IPv6 backbone. Figure 3 shows the basic components of a 4over6 multicast solution.

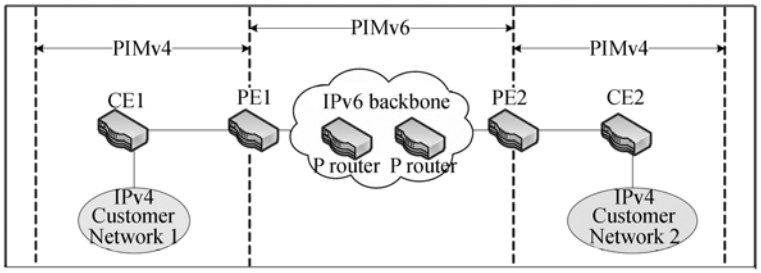

Figure 3 4over6 multicast components.

- CE, PE and P routers perform similar roles to those defined in 4over6 unicast. PE is a router at the edge with IPv4/IPv6 dual stacks, CE is a pure IPv4 router and P is a pure IPv6 router.
- PIMv4 runs inside the customer networks and between CEs and PEs, while PIMv6 runs in the IPv6 backbone network.
- The encapsulation table in PEs is extended to multicast encapsulation table.

Two situations are to be considered:

- The IPv6 transit core supports protocol-independent multicast for IPv6 (PIMv6).
- The IPv6 transit core does not support PIMv6.

4.1 Multicast solution for unicast transit core

If the transit core does not support multicast protocol, we have to transmit the multicast data packets in a unicast manner through transit core.

For the control plain, we need to handle the join and prune messages from IPv4 networks in order to help build multicast tree in customer networks. The way to do this is using the 4over6 mesh architecture to encapsulate and transmit IPv4 join/prune messages between customer networks.

As shown in Figure 4, an IPv4 host *A* from network 1 is going to join a multicast group, where the multicast source is host *B* located in network 2, and the address of the group is g. In order to join the group (s,g), *A* will send the join (s,g) message. As we have discussed before, the message will arrive at PE1 first. PE1 then encapsulates the packet and sends it to PE2, which decapsulates the packet and forwards it to network 2. When the join message travels through network 1 and network 2, the path of the multicast tree in customer networks will be constructed.

Meanwhile, PE2 keeps a record of the mapping between (s,g) and PE1, which means that PE1 is one of the receivers of the multicast group (s,g).

When a prune message from PE1 for (s,g) arrives at PE2, it simply removes the mapping between (s,g) and PE1 from its mapping table.

On the data plane, when a multicast data packet for group (s,g) arrives at PE2, it looks up the mapping table, and then encapsulates and sends the packet to every PE router in the mapping ta-

ble, which will decapsulate the packet and transmit it along the IPv4 network's multicast tree.

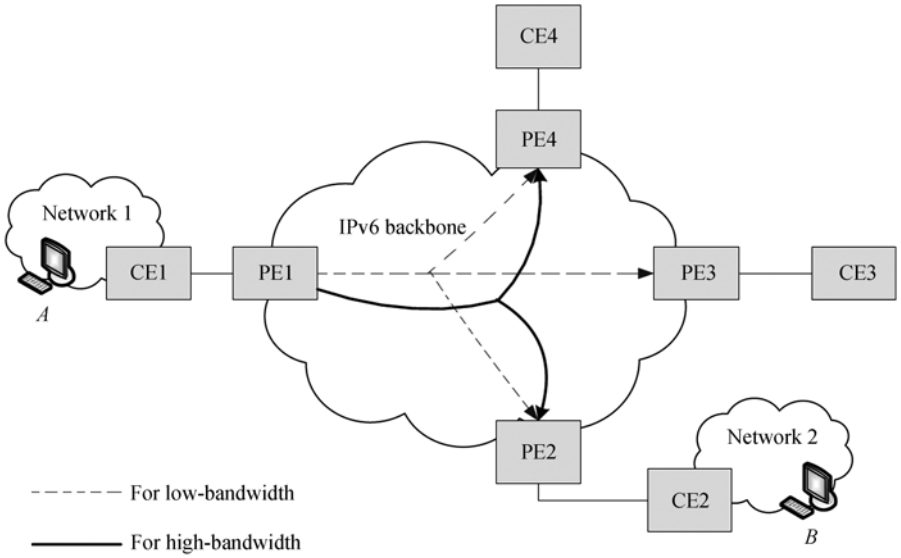

Figure 4 Multicast distribution trees.

4.2 Multicast solution for multicast transit core

The provider runs an instance of protocol-independent multicast for IPv6 (PIMv6) among PE routers and provider (P) routers, whereas the customer runs PIMv4 inside its network and between CEs and PEs. The PE performs 4over6 multicast functions, including connecting the PIMv4 trees and PIMv6 trees together and encapsulating multicast traffic for transport across the provider backbone.

The 4over6 multicast solution for this situation can be described like this: the IPv4 multicast source is in one IPv4 network, while its receivers may be in the same or different IPv4 networks. When constructing the multicast tree, it should cover the entire network (including IPv4 and IPv6), but has different packet formats in different networks. Now we will discuss the process in detail.

When an IPv4 multicast join packet arrives at the PE router, it translates its IPv4 source and group addresses into IPv6, and uses the translated addresses to continue the tree-constructing process in IPv6 network. But since the addresses are directly translated from IPv4, they do not actually exist in the IPv6 network. So here we use the RPF vector extension for PIMv6 to carry the egress PE router's IPv6 address. All the P routers inside the IPv6 network will forward the join packet according to the RPF vector. When the packet finally arrives at the egress PE router connecting the IPv4 network in which the source of the multicast is, the addresses of the packet will be translated back to IPv4 and the rest of the multicast tree will be constructed in the IPv4 network.

When a multicast prune message or data packet arrives at a PE router, their addresses will also be translated to IPv4, and RPF vectors will be used in the same way. So when the packet travels into IPv6 network, it will be treated as an IPv6 multicast packet, and all the translating jobs are done at the PE routers.

The 4over6 multicast solution has been submitted to IETF Softwire working group as an

Internet draft, and a prototype of the solution to be deployed on CERNET2 has been developed at Tsinghua University. The 4over6 multicast solution extends the functions of the 4over6 mechanism, and will promote the transition to IPv6 as well as the development of multicast applications in the future.

5 Implementation and deployment

We implemented a Linux-based prototype for 4over6 transition mechanism at Tsinghua University Lab in Beijing, and deployed it on CERNET2. As shown in Figure 5, 4over6 implementation on the routers has four functional modules: OAM module, MP-BGP module, EnCap table module and Forwarding module.

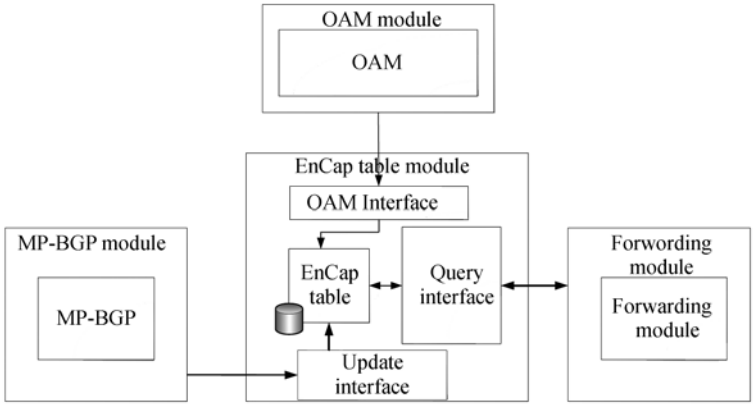

Figure 5 The 4over6 prototype implemented at the Tsinghua University Lab.

The 4over6 OAM module processes configuration commands that define the PE routers' tunnel attributes (IP version number, interface address, tunnel identifier, tunnel type, and so on).

The MP-BGP module supports the distribution of the 4over6 attribute, which in turn populates the PE encapsulation table containing the destination IPv4 prefix and IPv6 header address entries. The EnCap table module constructs and maintains the encapsulation table that stores IPv4 and IPv6 address mappings. In forwarding module, the virtual interface encapsulates and decapsulates packets received from IPv4 or IPv6 networks.

CERNET and CERNET2 have deployed both IPv4 and IPv6 networks and will continue to do so, thus introducing a common and growing requirement for coexistence between the two IP address families. So we deploy our 4over6 mechanism in CERNET2 and connect some IPv4 islands in CERNET over the IPv6 backbone as shown in Figure 6.

There are seven universities with totally 8 nodes joining the deployment and making cooperation to finish the test. We deployed 4over6 gateway routers (AFBR) on the edge of CERNET2 in order to offer the IPv6 access. All the clients in CERNET can not only communicate with each other, but also get access to the Internet all around the world. The deployment demonstrates the scalability of our 4over6 mechanism.

In order to validate the performance of the proposed framework, we did some tests on the 4over6 gateway boxes. In the tests, we first configured two 4over6 boxes to connect with each other across CERNET2, and then used a tester to generate IPv4 packets and sent them through the established 4over6 tunnel. The hardware configurations of the 4over6 boxes we used are

listed in Table 1.

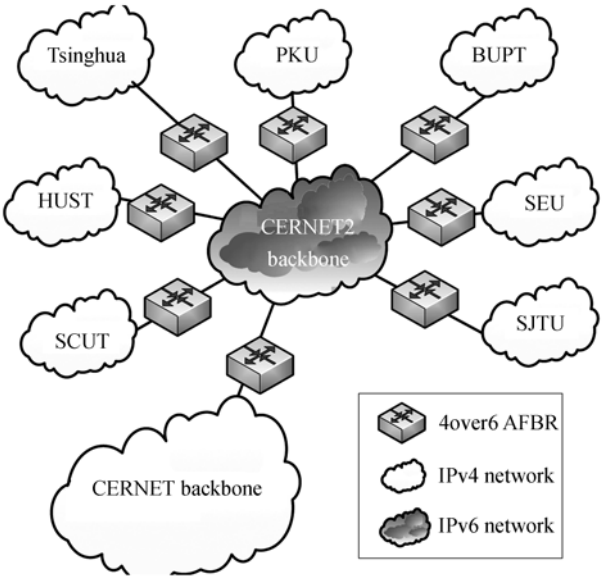

Figure 6 4over6 deployment.

Table 1 Hardware configurations for 4over6 boxes

CPU	Intel Xeon 2.8 GHz×4
Memory	1 GB
OS	RedHat Linux 9
Network Adaptor	1 Gbps

Figures 7—10 show the results of the tests for different packet sizes, i.e. 64, 256, 512 and 1400 bytes. In these figures, the *x*-coordinate shows the speed at which we generate packets, and the *y*-coordinate shows the percentage of packet loss and CPU time for packet processing. In each figure there is a key point (marked in a small circle) where the packet loss becomes higher. Therefore, the *x*-coordinate of this key point can be taken as the best throughput for that packet size. We conclude these throughputs in Table 2.

From the test results we can see that on medium-size packets (512 bytes), we can achieve a high throughput (600 Mbps) with a low loss rate (0.001%). Since the prototype is implemented to forward packets by software, its performance will be much better if implemented by hardware.

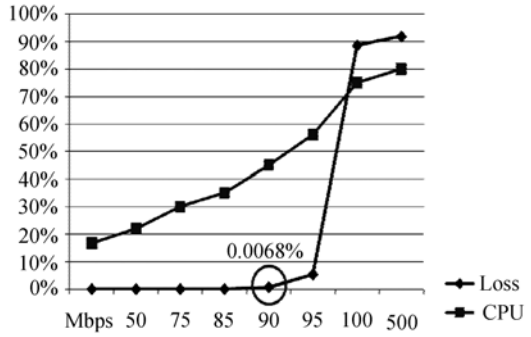

Figure 7 Test results of 64 bytes packets.

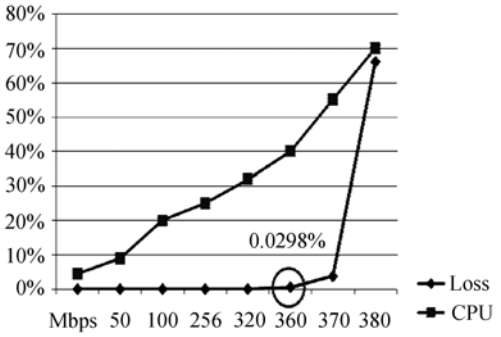

Figure 8 Test results of 256 bytes packets.

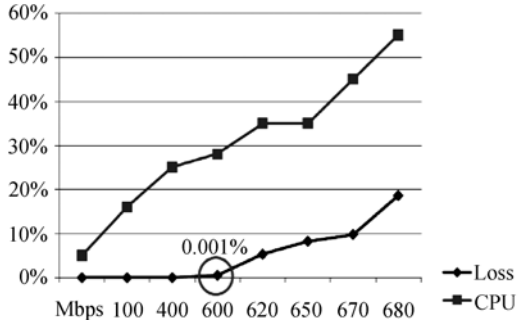

Figure 9 Test results of 512 bytes packets.

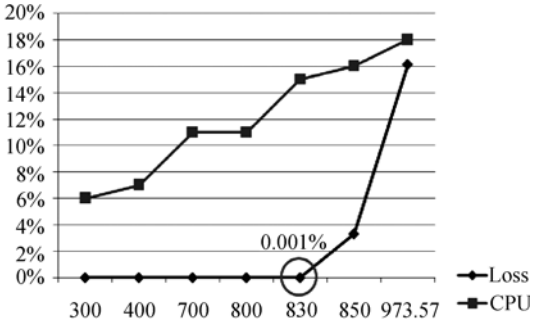

Figure 10 Test results of 1400 bytes packets.

Table 2 Test results of 4over6 boxes

Packet size (B)	Throughput (Mbps)	Loss (%)
64	90	0.0068
256	360	0.03
512	600	0.001
1500	830	0.001

Although there are several other transition solutions as discussed in section 2, the proposed IPv4-over-IPv6 scheme achieves high scalability and low management overhead. For example, NAT-PT needs to be implemented on a special box that connects two different types of networks, but it requires the box to hold states for each connection like NAT does, thus causing scalability problems. Another widely used scheme is manual-configured tunneling, but it only supports IPv60-over-IPv4 and can obviously increase management overhead when the nodes in the network increases in number.

6 Conclusion and future work

In this paper, we aim to find a solution to the IPv4/IPv6 transition problem, especially the transition based on IPv6 backbone. Two frameworks are proposed and several detailed reference models and solutions are designed. Furthermore, we implemented our idea for 4over6 transition mechanism in a Linux-based prototype and deployed the 4over6 devices on CERNET2 to validate our idea.

There are lots of problems in the field of IPv4/IPv6 transition, such as the problem of how to enhance the efficiency of multicast and how to accommodate mobility management in 4over6 scenario. In addition, more experiments and large deployment need to verify the reliability and scalability.

1 Information Sciences Institute. Internet Protocol, RFC0791, September 1981
2 Huston G. IPv4 —— How long have we got? http://www.potaroo.net/ispcolumn/2003-07-v4-address-lifetime/ale.html, July 2003
3 Hinden R, Deering S. Internet Protocol Version 6 (IPv6) Addressing Architecture, IETF RFC 3513, April 2003
4 Wu J P, Cui Y, Li X, et al. The transition to IPv6, Part I. 4over6 for the China education and research network. IEEE Internet Computing, May-June 2006. 80－85
5 Tatipamula M, Grossetete P, Esaki H. IPv6 integration and coexistence strategies for next-generation networks. IEEE Commun Mag, 2004, 42(1)
6 Park E -Y, Lee J -H, Choe B -G. An IPv4-to-IPv6 dual stack transition mechanism supporting transparent connections be-

tween IPv6 hosts and IPv4 hosts in integrated IPv6/IPv4 Network. In: Communications, 2004 IEEE International Conference, 2004, 2: 1024－1027
7 Durand A, Ward D. IETF Softwares WG Charter, http://ietf.org/html.charters/softwire-charter.html
8 Li X, Dawking S, Ward D, et al. Softwire Problem statement. IETF RFC 4925, September 2007
9 Gilligan R, Nordmark E. Transition mechanisms for IPv6 hosts and routers. RFC2893, August 2000
10 Hanks S, Li T, Traina P. Generic routing encapsulation over IPv4 networks. IETF RFC 1702, October 1994
11 Carpenter B, Moore K. Connection of IPv6 domains via IPv4 clouds. RFC 3056.
12 Savola P, Patel C. Security considerations for 6to4. IETF RFC3964, December 2004
13 Carpenter B, Jung C. Transmission of IPv6 over IPv4 Domains without explicit tuunnels. RFC2529, March 1999
14 De Clercq J, Ooms D, et al. Connecting IPv6 islands over IPv4 MPLS using IPv6 provider edge routers (6PE). IETF Draft "draft-ooms-v6ops-bgp-tunnel-03.txt", April, 2004
15 Cisco Press, IPv6 deployment Strategies. www.cisco.com/univercd/cc/td/ doc/cisintwk/intsolns/ipv6_sol/ipv6dswp.pdf
16 Tsirtsis G, Srisuresh P. Network address translation - protocol translation (NAT-PT). RFC2766,February 2000
17 Hagino J, Yamamoto K. An IPv6-to-IPv4 transport relay translator. IETF RFC 3142, June 2001
18 Fenner B, Handley M, Holbrook H, et al. Protocol independent multicast - sparse mode (PIM-SM): Protocol specification (revised). Internet Engineering Task Force (IETF), Draft, draft-ietf-pim-sm-v2-new-11.txt, October 2004
19 Bhaskar N, Gall A, Lingard J. Bootstrap router (BSR) mechanism for PIM. Internet Engineering Task Force (IETF) Draft, draft-ietf-pim-sm-bsr-06.txt, October 2005
20 Moy J. Multicast extensions to OSPF. IETF Request for Comments 1584, March 1994