Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 23, 2009

Y. Cui
S. Wang
M. Xu
J. Wu
X. Li
Tsinghua University
May 22, 2009

VA-Based IPv6 Transition
draft-cui-softwire-va-based-transition-00

Status of this Memo

Copyright Notice

Abstract

With the increasing deployment of IPv6 networks, IPv6 transition has become one of the key problems in developing IPv6 networks. Among various transition scenarios, one is common where connectivity between IPv4 networks is desired across IPv6-only backbone network. In such case, ISP operating the IPv6 backbone will accommodate connectivity and offer transit services for attached IPv4 networks. Softwire WG defined softwire mesh mechanism for both of the IPv4-over-IPv6 scenario and the opposite scenario of IPv6-over-IPv4. Softwire mesh uses automatic softwire tunnels employing multi-protocol BGP extensions for distributing IPv4 routes, where BGP and tunnuls should be configured or setup as a full-mesh architecture. This draft, however, proposed an aggregated, centralized mechanism similar to Virtual Aggregation (VA) mechanism, which can significantly shrink the forwarding information base (FIB) size of Address Family Border Routers (AFBRs), reduce the total amount of routing activity, and provide the IPv6 ISP with an easy way to manage the transit service.

Table of Contents

1. Introduction

Recently more and more IPv6 networks have be deployed, especially IPv6 backbone networks, while the existing IPv4 networks still carry the major network traffic and hold the major network services and applications, though facing serious address space problem and other problems. It has been agreed that IPv4 and IPv6 networks will co-exist for a long term. There's been basically two aspect of IPv6 transition research: connection between IPv4 and IPv6 nodes and connection between networks of one address family traversing network of the other address family. Basic solution for the former one is address translation and for the latter one is tunneling.

For the traversing transition of IPv4-over-IPv6, softwire mesh framework gives a way that IPv4 client networks can be connected through IPv6 backbone. It's done by extending MP-BGP to exchange IPv4 routes between dual-stack Provider Edge routers (PE), and using softwire tunnel to forward IPv4 packets through encapsulation and decapsulation. In both data plane and control plane, all PEs form a full mesh.

Virtual Aggregation (VA) mechanism shrinks the FIBs of any and all routers easily by an order of magnitude with negligible increase in path length and load [I-D. draft-francis-intra-va]. It can be deployed autonomously by an ISP, and co-exist with legacy routers in the ISP. VA divides the IP address space into Virtual Prefixes (VPs), and uses tunnels to aggregate the regular sub-prefixes within each VP. For each sub-prefix within a VP, Aggregation Point Routers (APRs) have a tunnel from themselves to the remote ASBR (Autonomous System Border Router) where packets for that prefix should be delivered. Since APRs may not be on the shortest path between the ingress and egress routers, the packets may take a few more hops and experience additional latency. VA can avoid traversing the APR for selected routes by installing these routes in non-APR routers. In other words, even if a router is not an APR for a given sub-prefix, it may still install that sub prefix into its FIB. Packets in this case are tunneled directly to the BGP NEXT_HOP.

This draft proposes an approach that learns the basic idea from VA to solve the IPv4-over-IPv6 traversing problem, called VA-based IPv6 transition. In control plane, it organizes the IPv4 address space into VPs and aggregates the IPv4 routes from the client network; regular IPv4 prefixes are collected by APRs in IPv6 backbone, while PEs only have to maintain VPs in the FIB. In data plane, VA-based IPv6 transition uses APRs in IPv6 backbone to be intermediate tunneling forwarders between PEs; IPv4 packets are tunneled to APRs from IPv4 client network, and then tunneled to the destination IPv4 network.

VA-based IPv6 transition can significantly shrink the transition FIB size of PEs, reduce the total amount of transition routing activity (routing protocol process in transition-related routers and transition-related routing packets delivered), and provide the IPv6 ISP with a better way to manage the IPv4-over-IPv6 transit service. This mechanism has good scalability and works well when the number of IPv4 client networks increases.

2. Terminology

Aggregation Point Router (APR): This draft adopts the name of APR from VA. An Aggregation Point Router (APR) is a router that aggregates a Virtual Prefix (VP) by installing routes (into the FIB) for all of the sub-prefixes within the VP. APRs advertise the VP to other routers. In this draft, all VPs are IPv4 prefixes and APR is deployed in IPv6 backbone; for each sub-prefix within the VP, APRs have a tunnel to the IPv4 client network where IPv4 packets reach their destinations.

Provider Edge router (PE): The dual-stack edge routers of the IPv6 backbone, where IPv4 packets enter and leave the backbone. PE is often referred to as AFBR (Address Family Border Router). Interior nodes of the backbone are often known as "P routers".

Popular Prefix: This draft adopts the name of popular prefix from VA. In VA, a popular prefix is a sub-prefix that is installed in a router in addition to the sub-prefixes it holds by virtue of being an Aggregation Point Router. The popular prefix allows packets to follow the shortest path. In VA-based IPv6 transition, a popular prefix is an IPv4-prefix installed in a PE router in addition to VPs. IPv4 packets whose destination falls within popular prefix can traverse IPv6 backbone in softwire tunnels.

Virtual Prefix (VP): This draft adopts the name of VP from VA. A Virtual Prefix (VP) is a prefix used to aggregate its contained regular prefixes (sub-prefixes). A VP is not physically aggregatable, and so it is aggregated at APRs through the use of tunnels.

3. VA-Based Transition framework

3.1. Scenario

The scenario of VA-Based IPv6 Transition is illustrated in figure1. A number of P routers compose an IPv6 only backbone, in which a few APRs are deployed. The PE routers are dual-stack and connect IPv4 client networks. Every PE builds tunnels with every APR. The IPv6 backbone acts as a transit core to transport IPv4 packets across the IPv6 backbone. This enables each of IPv4 client network to communicate with each other via tunnels.

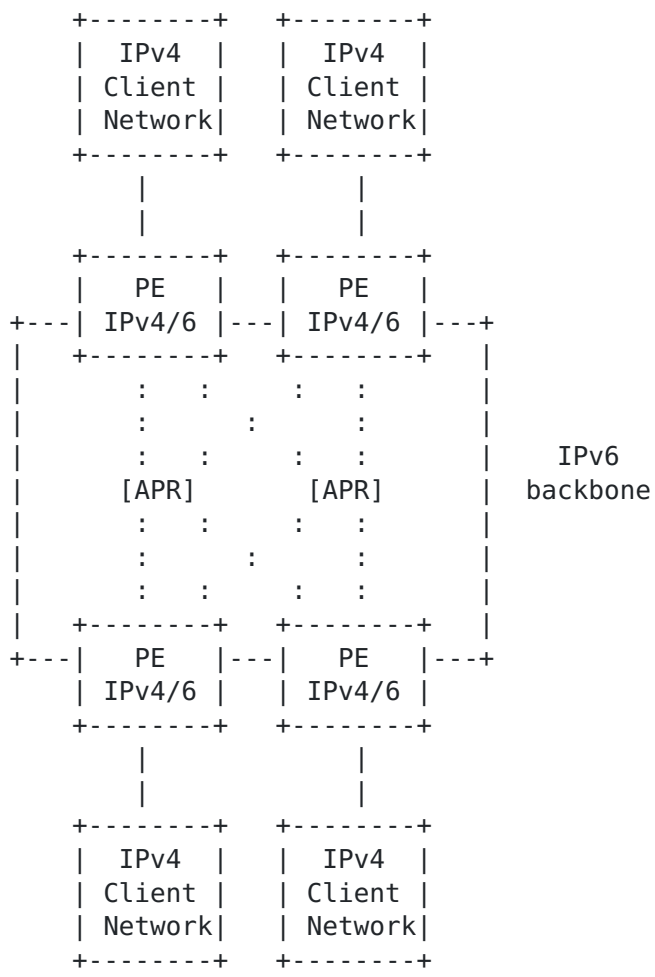

Figure 1 VA-Based IPv6 Transition Scenario

In this scenario, VA-Based IPv6 Transition provides IPv4 connectivity in three steps: downlink routing, uplink routing and tunneled forwarding.

3.2. Downlink routing

In downlink routing, every selected APR must distribute its VP information to every PE. VP can be configured in advance in every APR, that is, Every APR is configured with which VPs it is responsible for. Then ARP must advertise its VPs to all PEs, using some intra-domain routing protocol.

In VA it is said that initial VPs can be statically configured in every VA router as VP-list, and a VP can be added by some APR originating routes for it and advertising these routes, also a VP can be deleted by first removing it from the VP-Lists of non-APRs, waiting for them to install sub-prefixes and then removing it from the APRs. This draft, instead, uses the uniform routing method that initial VPs and all VP changes are first configured in APRs and then advertised to all PEs for automaticity and simplicity. How to process this routing behavior is still a concern, since IPv4 routes need to be advertised in or through IPv6 network. We'll discuss this in section 4.1.

After this step, every PE has all the VPs in its FIB table so it knows which APR to forward an IPv4 packet to, even there is no regular prefix match for the destination.

3.3. Uplink routing

This step is opposite to downlink routing. In this step, every PE must advertise the prefixes of the IPv4 client network behind it to corresponding APRs. For example, if the IPv4 network behind PE1 contains two subsets, 192.168.1.0/24 and 10.0.0.0/16, and APR1 in IPv6 backbone is responsible for VP 0.0.0.0/2 while APR2 is responsible for VP 192.0.0.0/2, then PE1 must advertise 192.168.1.0/24 to APR2 and 10.0.0.0/16 to APR1, since sub-prefix 10.0.0.0/16 falls within VP 0.0.0.0/2 and 192.168.1.0/24 falls within 192.0.0.0/2.

After this step, every APR has all the sub-prefix that is from the IPv4 client networks and falls within one of the VPs the APR is responsible for. Therefore, every APR knows which PE to forward an IPv4 packet received from another PE earlier to.

3.4. Tunneled forwarding

In VA-based transition, forwarding an IPv4 packet through the IPv6 backbone includes the following steps: the ingress PE encapsulates the incoming IPv4 packet with the IPv6 tunnel header; transmits the encapsulated packet through the IPv6 backbone to an APR; the APR decapsulates the packet and encapsulates the packet with another IPv6

tunnel header; transmits the encapsulated packet through IPv6 backbone to the egress PE; the egress PE decapsulates the IPv6 header and forwards the original IPv4 packet. All the encapsulations and decapsulations are performed on PEs and APRs, other routers in IPv6 backbone take encapsulated packets just as native IPv6 packets. The forwarding procedure is illustrated in figure2.

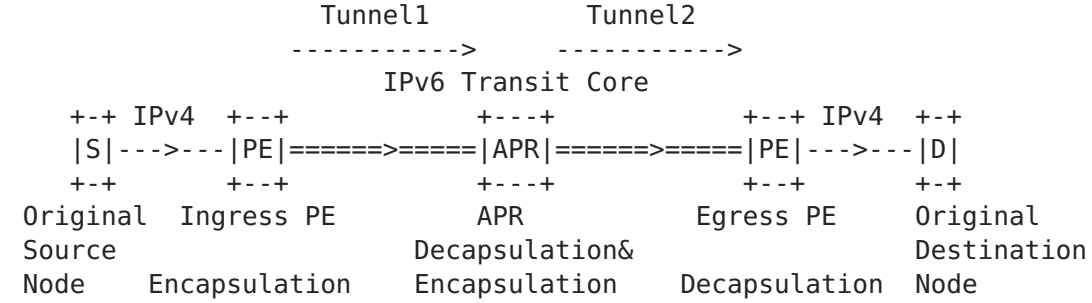

Figure 2 VA-based transition: tunneled forwarding

When an ingress PE receives an IPv4 packet from IPv4 network, it looks up the destination IP address of the packet. In this case, the best match for that address will be a VP route whose next hop is an APR. The ingress PE must forward the packet through a tunnel to the APR. This is done by encapsulating the packet, using an IPv6-encapsulated header with the destination address of the APR, and then forwarding the packet to IPv6 network.

When the APR receives the encapsulated IPv6 packet, it extracts the payload, i.e., the original IPv4 packet, and looks up the packet's destination address. In this case, the best match for that address will be a regular IPv4 route whose next hop is a PE (the egress PE). The APR will encapsulate the packet again, this time with the destination IPv6 address of the egress PE, and then forward the packet to IPv6 network again.

When the egress PE receives the encapsulated IPv6 packet, it extracts the payload, gets the original IPv4 packet and forwards it further by looking up its IP destination address.

4. Design detail discussion

4.1. Routing

4.1.1. BGP-based routing scheme

One way to achieve downlink and uplink routing is using BGP. For routing exchange, Every APR must build an IPv6 IBGP peer with every PE.

In downlink direction, VPs will be advertised through BGP process, from every APR to every PE. Note here the prefix is IPv4 format and next hop is IPv6 format. RFC5549 (Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop) has made an extension of MP-BGP, modifying MP_REACH_NLRI format to convey IPv4 NLRI prefix information with an IPv6 next hop address. So we can just use this v4nlri-v6nh extension here. BGP process in APRs and PEs need to be modified to fit the extension.

In uplink direction, BGP routing is similar to downlink routing, except this time every PE advertise the prefixes of the IPv4 client network behind it to corresponding APRs. In this part, because only the APRs whose VPs the IPv4 prefixes fall within need to receive the routes, ideally we can build BGP peers only between such APRs and PEs. However, we've already built BGP peers between every APR and every PE in downlink routing, so we can just add some filters in every APR so that it only accept what it actually needs, that is, the IPv4 prefixes fall within its VPs.

Since BGP is run on end to end TCP connection, we won't introduce IPv4 routes into IPv6 networks, but we do need a static configuration for APRs and PEs.

4.1.2. OSPF & registration-based routing scheme

It's mentioned in section3.2 that VA can add a VP for every router by originating route for it and advertising the route. Obviously it's done through intra-domain routing. We can also achieve routing exploiting intra-domain protocol.

For downlink routing, we can extend IPv6 OSPF to distribute IPv4 VPs. What we need to do is to add a particular prefix for every VP to form a 'fake' IPv6 prefix and modify the OSPF process in APRs and PEs, so that APRs can generate routes with such 'fake' IPv6 prefixes while PEs can recognize these routes and extract the IPv4 VPs to update their FIBs. No change is required for other OSPF routers. Using OSPF we'll introduce the IPv4 routes into IPv6 network, but since the total number of VPs is small, it won't become a serious problem.

Meanwhile, since OSPF will flood these routes, all PE can learn VPs atomically without configuration. Also, if the other routers can cooperate by recognizing the fake prefixes as regular prefixes and forward packets for those destinations, then they can participate in the forwarding process from PEs to APRs and tunnel won't be actually needed in this step.

Apparently uplink routing can't be executed by OSPF, or we'll pour a lot of client IPv4 routes into the IPv6 backbone. However, since the uplink routing activity is really simple, we can consider defining a new, registration routing protocol to achieve it. The protocol is a simple point to point, one hop routing protocol, and deployed in APRs and PEs. On a PE, for every prefix change in its IPv4 FIB (all the prefixes in initial case), send a routing message containing the prefix to corresponding APR (PE can figure out which APR to send message to by checking its FIB: the next hop for corresponding VP is the right APR); On an APR, for every routing message received, normally the prefix will fall into the APR's VP, so the APR updates its IPv4 FIB. This procedure is similar to DNS registration, except that DNS servers are hierarchical while APRs' architecture is flat. Note here this protocol should be triggered when the downlink routes or we say the VPs reaches PEs, since in this scheme we don't configure the APRs for PEs in advance and let PEs learn the APR and VP information through OSPF.

4.2. Tunnel

In VA, a variety of tunnel types can be used: MPLS LSPs, IP-in-IP, GRE, L2TP, and so on. VA-based transition doesn't restrict tunnel types, either. Actually since remote ASBR information for VA isn't needed in VA-based transition, standard softwire tunnel can be used in the scenario. Note that signaling is needed for some types of tunnels using BGP. Refer to [I-D. draft-wu-softwire-mesh-framework] and RFC5512(The BGP Encapsulation SAFI and the BGP Tunnel Encapsulation Attribute) for details.

4.3. Cooperate with softwire mesh

VA-base transition can cooperate well with softwire mesh, just similar to popular prefix can be used in VA. Since the two mechanisms both influent data forwarding by inject entries to PE's FIB, there will be no confliction in them. Actually, if both mechanisms are used, because VPs' masks are usually shorter than regular prefixes, softwire entries will have higher priority. Here we also call the prefixes of softwire entries popular prefixes.

For some common, high-traffic IPv4 connections, PEs can choose to follow popular prefixes, so only one time encapsulation and

decapsulation is needed and encapsulated packets can follow the shortest path in IPv6 backbone; for other IPv4 connections, PEs can refer to VA-based transition so the FIB size won't be large and PEs don't have to build a lot of BGP peers and tunnels with other PEs.

4.4. Inter-domain situation

The situation will be different if two IPv4 network from two IPv6 domains which run VA-based IPv6 transition want to get connected. In this inter-domain scenario, APRs in one ISP don't have the regular prefixes of the IPv4 network behind the other ISP, though it may fall within its VPs. So if a PE receives an IPv4 packet whose destination is in the other ISP, it will still encapsulate and send the packet to the APR whose VP matches the destination in its own ISP; After the corresponding APR receive and decapsulate the packet, it has to drop the packet since there is no regular prefix match in its IPv4 FIB table for the destination. Apparently APRs need to learn IPv4 routes of the other ISP.

The scenario is illustrated in figure3. Our basic thought is to get APRs from different domains to exchange IPv4 sub-prefixes if their VPs have overlaps, through EBGP process. We can either build EBGP peers directly between APRs, or use an extra BGP router to run EBGP process and collect and distribute inter-domain IPv4 sub-prefixes. The former way provides less number of tunnels in forwarding procedure while the latter one provider simpler routing scheme. However, both the two methods described here are incipient and require further consideration.

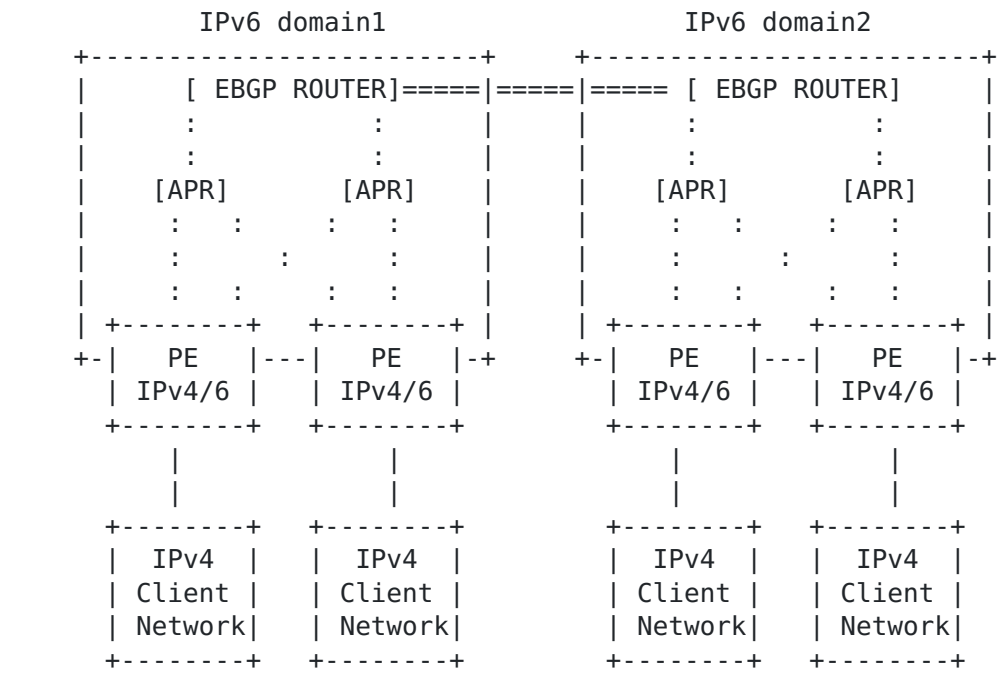

Figure 3 inter-domain scenario

5. Benefits of VA-Based IPv6 Transition

There are mainly three benefits using VA-Based IPv6 Transition in IPv4-over-IPv6 traversing. The first one is that it can significantly shrink the FIB size of the PEs. Every PE needs to store only the IPv4 VPs of all APRs, while the whole IPv4 regular sub-prefixes are distributed in the APRs' FIBs. PE can also keep a few regular prefixes in its FIB for softwire use to reach better performance. So we achieve better scalability than pure softwire mesh.

Secondly, it can reduce the total amount of routing activity for transition. In this mechanism routing is executed between every APR and every PE. Since there are only a few APRs in the domain, the total amount of routing activity is in proportion to the number of PEs. In softwire mesh, every two PEs form a BGP peer, so the amount of routing activity is in proportion to the square of the number of PEs. It's obvious that we can carry out less routing activity than softwire simply implementing uplink and downlink routing in BGP.

Moreover, VA-Based IPv6 Transition can provide the IPv6 ISP with a better way to manage the IPv4-over-IPv6 traversing service. In this mechanism, IPv4 routes are collected in APRs maintained by the ISP. PEs don't know the detailed routes: they just learn a few VPs for IPv4 forwarding. If a new client IPv4 network wants to jump in and get connected with other IPv4 networks, the ISP just needs to tell the access PE the addresses of the APRs. It's more transparent than softwire mesh where PE needs to know the addresses of all other PEs and Fewer configurations are needed.

6. IPv6-over-IPv4 scenario

So far we've only discussed the IPv4-over-IPv6 scenario, but we believe the opposite scenario is similar: IPv6-over-IPv4 doesn't bring extra difficulty. For tunneled forwarding, standard softwire tunnels don't restrict the IP protocol type; in fact, the transit network protocol and the network protocol inside the tunnel are referred as E-IP and I-IP for general purpose in [I-D. draft-wu-softwire-mesh-framework]. As for routing, when BGP is used, RFC4798 (Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers (6PE)) defines the extension of MP-BGP to support V6NLRI-V4NH; If we choose to use OSPF and the new registration protocol, then we need to do extension and definition in either scenario. VA-based IPv6 transition fits both IPv6-over-IPv4 scenario and IPv4-over-IPv6 scenario naturally.

7. Security considerations

Since our mechanism is based on VA, we refer to VA for security concerns. Our mechanism doesn't introduce security problems other than the ones of VA's.

If VA is configured properly, or we say if all APRs and PEs are configured properly, then any new concerns for intra-domain security appear to be relatively minor. In particular, DoS attack to APR won't significantly worsen the DoS problem, and VA won't limit the deployment of DoS defense systems.

As to the situation of Mis-configured VA, VA introduces the possibility that a VP is advertised outside of an AS. Usually a VP is large(i.e. larger than any real prefixes), and the impact is minimal. Smaller prefixes will be preferred because of best-match semantics, and so the only impact is that packets that otherwise have no matching routes will be sent to the misbehaving AS and dropped there. If the VP is small, then it may cause a traffic hijack which can happen with or without VA, so VA doesn't introduce a new security problem.

8. Acknowledgements

This draft gets the very original idea from VA, and extends the idea to solve a different problem: IPv4-over-IPv6 transiton. The authors would like to thank P.Francis, X.Xu, H.Ballani everyone else contributed to VA.

9. References

9.1. Normative References

[RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, February 2007.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, April 2009.

[RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, May 2009.

9.2. Informative References

[I-D.francis-intra-va] Francis, P., Xu, X., Ballani, H., Jen, D., Raszuk, R., and L. Zhang, "FIB Suppression with Virtual Aggregation", draft-francis-intra-va-01 (work in progress), April 2009.

[I-D.ietf-softwire-mesh-framework] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", draft-ietf-softwire-mesh-framework-06 (work in progress), February 2009.

Authors' Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: cy@csnet1.cs.tsinghua.edu.cn

Shengling Wang
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: slwang@csnet1.cs.tsinghua.edu.cn

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: xmw@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

Xing Li
Tsinghua University
Department of Electronic Engineering, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
Email: xing@cernet.edu.cn