Active Network Supports for Mobile IP

LU Yueming (陆月明), QIAN Depei (钱德沛), XU Bin (徐 斌) and WANG Lei (王 磊)

Department of Computer Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, P.R. China

E-mail: yueming@xjtu.edu.cn

Received January 8, 2000; revised November 6, 2000.

Abstract The basic mobile IP protocol is difficult to implement on the traditional IP network and not flexible and efficient under certain conditions. For example, firewalls or boundary routers may drop packets sent by mobile nodes for security reasons. Traditional networking devices such as routers cannot dynamically load and unload extended services to achieve different qualities of services. In this paper, a new scheme of using the active network to support the mobile IP is presented. The Softnet, a prototype of active networks based on mobile agents, is introduced. The active network is characterized by the programmability of its intermediate nodes and therefore presents dynamic and flexible behaviors. Special services can be dynamically deployed onto the active nodes in the Softnet. This property is definitely required in implementing the mobile IP protocols. The Softnet supports not only the basic mobile IP protocol but also other extended mobile IP protocols. Virtual networks for mobile IP services are dynamically formed by mobile agents in the Softnet to provide different qualities of services.

Keywords mobile IP, Softnet, active network, mobile agent, virtual network

1 Introduction

The mobile IP service becomes more and more important with the ever increase of portable devices such as palm PCs and wireless communication equipment in the network. The mobile IP protocol allows the node to dynamically change its access points to the network while still communicating with other Internet nodes using its "permanent" IP address, the home address. Fig.1 illustrates the basic mobile IP protocol described in IETF RFC[1].

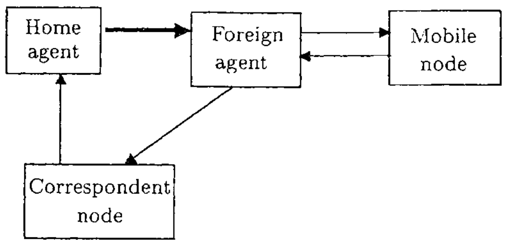


Fig.1. Mobile IP protocol descried in IETF RFC.

When a mobile node is away from its home, datagrams sent to that node arrive at its home network via traditional IP routing and are intercepted by the home agent. The home agent is responsible for tunneling the datagrams to the foreign agent with proper care-of address. The datagrams are finally recovered by the foreign agent and delivered to the mobile node. Datagrams from the mobile node are sent directly to the correspondent node via the foreign agent.

However, there are some limitations in deploying the above basic mobile IP protocol in traditional IP networks. In the next section, we will consider some aspects regarding the difficulties. Then a new type of network, the mobile agents-based active network, will be introduced for supporting efficient and flexible mobile IP protocols. The Softnet, a prototype of the active network[2], is used as an example throughout this paper.

1.1 Limitations to Implementation of the Mobile IP on Traditional IP Networks Routing

Supports to the mobile IP protocols require the extension of basic routing protocols. At least, the mobile node's default gateways on the home network must be upgraded to support the home agent

This work is supported by the National Natural Science Foundation of China (No.69973036), the '863' High-Tech Programme of China (No.863-317-01-10-99) and the NKBRSF (No.G1999032700).

functions and link layer protocols like the proxy ARP[3]. But the traditional IP routers are not able to extend its services dynamically. To support a special service that is seldom used, all devices must be upgraded. Services supported must run all the time and cannot be dynamically unloaded. When special services keep increasing, network devices become hard to maintain and the efficiency will be low.

Security

Consider the case in Fig.1 of transmitting a packet from the mobile node to the correspondent node. Though transmitting it through a direct path from the mobile node, via the foreign agent to the correspondent node seems to be natural and efficient, it may not be possible under certain circumstances. Remember that the source IP address of the packet is the mobile node's home IP address. For security reasons, firewalls or boundary routers existing in the foreign network may check the source IP address of the packet to verify if it is from an internal node. All packets with a source IP address that does not belong to the foreign network will be dropped[4]. The on-going efforts to solve this problem include route optimization[5], firewall traversal[6], reverse-tunneling[7] and bi-directional tunneling[8]. All these schemes allow packets sent by a mobile node to cross boundary routers and firewalls.

Performance

When the correspondent node and the mobile node are closely located, the bi-directional tunneling mobile protocol as well as the basic mobile IP protocol in Fig.1 are inefficient. Direct delivery between the mobile node and the correspondent node is desirable, especially when the correspondent node is also a mobile node. For performance consideration, we need to create a virtual path between the mobile node and the correspondent node to tunnel packets.

Flexibility

The mobile node registers its IP addresses to agents. Hence the basic mobile IP protocol can only provide identical quality of service to every mobile node regardless of the distance between the mobile node and the correspondent node.

Agents Intelligence

The foreign agent and the home agent in the basic mobile IP protocol are non-intelligent, fixed, and static, and can handle data streams only.

1.2 Active Network Supports for Mobile IP Protocols

The active network is an innovative network characterized by programmability of its intermediate network nodes such as routers. This feature means that special services can be dynamically deployed to relevant network nodes so that their external behaviors will be changed according to the deployed services. It is the feature that we need to implement the mobile IP protocols. We will discuss later how the active network dynamically forms virtual networks by loading and unloading services on its active nodes.

In an active network, the virtual path between the home agent and the foreign agent can be dynamically formed to support the bi-directional tunneling mobile IP protocol. Direct delivery between the mobile node and the correspondent node can be supported also by establishing a virtual path between them.

Because different virtual networks can be dynamically formed, services with different qualities can be supported according to the transmission distance and other criteria. The mobile agents used in implementing the active network prototype are intelligent components and can handle mobile code to support complex services.

2 The Architecture of the Softnet

The Softnet, an active network based on mobile agents, is a virtual network built on top of the traditional IP network. Fig.2 illustrates the architecture of Softnet. Physically, the Softnet (G_p)

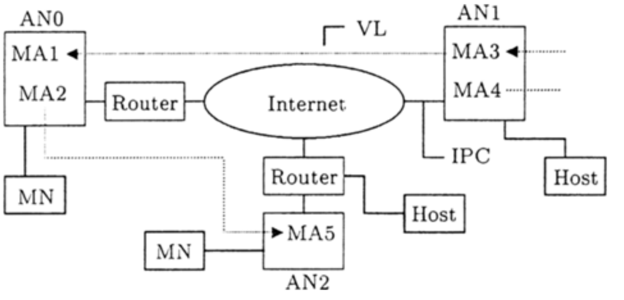
consists of active nodes (AN) and IP channels (IPC). It can be denoted as

$$G_P = (\text{AN}, \text{IPC}),\ \text{IPC} = \{(\text{AN}_i, \text{AN}_j) | \text{AN}_i, \text{AN}_j \in \text{AN}\}.$$

However, logically the Softnet (G_l) is a virtual network which consists of mobile agents (MA) and interconnecting virtual links (VL). It can be denoted as

$$G_l = (\text{MA}, \text{VL}), \text{VL} = \{(\text{MA}_i, \text{MA}_j) | \text{MA}_i, \text{MA}_j \in \text{MA}\}.$$

Mobile nodes (MN) or conventional hosts (Host) can be attached to routers or active nodes to obtain services of the Softnet.



MA: mobile agent; AN: active node; VL: virtual link; IPC: IP channel; MN: mobile node.

Fig.2. Architecture of Softnet.

2.1 Active Nodes

An active node consists of MAC drivers and the execution environment. Fig.3 illustrates the structure of the execution environment. It is composed of a de-multiplexer (demux), an execution unit, and a loader. The traditional packet consisting of a header and a payload is called the passive packet and can be denoted as *P*(*header, payload*). The packet consisting of a header, code, data, and a payload is called an active packet and can be denoted as *P*(*header, code, data, payload*)[9]. The function of demux is to identify different packets and to dispatch them to different processing units. Mobile agents, Java objects, native functions, and code are loaded by the loader and executed in the execution unit.

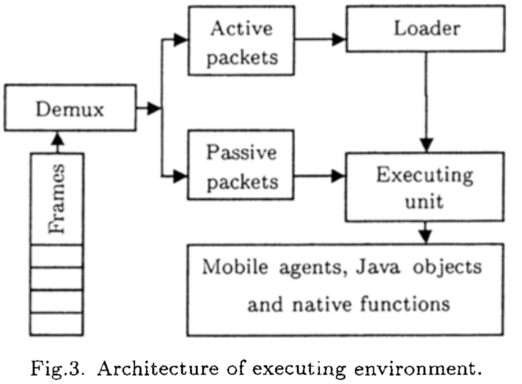


Fig.3. Architecture of executing environment.

3 Softnet Supports for Mobile IP

3.1 Implementing the Home and Foreign Agents

Some mobile agents in the Softnet are programmed to be mobile IP home and foreign agents, which are named mobile home agent and mobile foreign agent, respectively. The node where the mobile home agent resides is called home active node. The node where the mobile foreign agent resides is called foreign active node.

If the correspondent node is also an active node, it can load itself a mobile agent called mobile correspondent agent. The mobile home or foreign agent may reside in any node or in the default gateway as long as it is an active node.

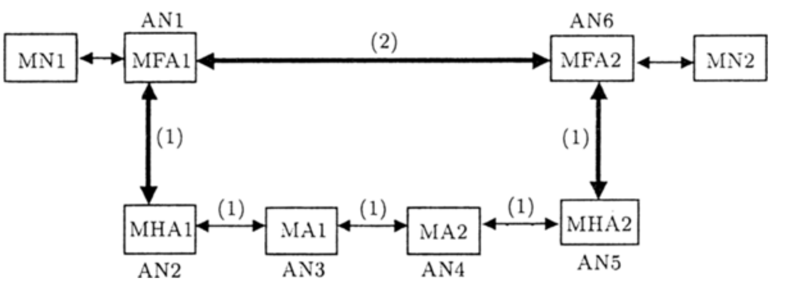
3.2 Linking Agents to Support the Basic Mobile IP Protocol

In the basic mobile IP protocol, a mobile node obtains a care-of address when it moves into a foreign network. The care-of address can either be determined from a foreign agent's advertisements (a foreign agent care-of address), or by some external assignment mechanism such as DHCP (a co-located care-of address). Then, the mobile node registers the care-of address to its home agent and its home address to the foreign agent.

To support the basic mobile IP protocol in a traditional IP network, agents must permanently exist on the home network and all accessed foreign networks. In Softnet, no mobile agent is dispatched to any active node as long as the mobile node stays in its home network. When the mobile node moves into a foreign network, it obtains a care-of address from the advertisement of the default active gateway and dispatches a mobile foreign agent to that gateway. Then, through the newly established mobile foreign agent, the mobile node dispatches a mobile home agent to the default active gateway or an active node in its home network. Finally, the mobile node links the mobile home agent and the mobile foreign agent by registering their addresses to each other. After the link is established, IP packets can be tunneled between the mobile home agent and the mobile foreign agent.

3.3 Linking Mobile Agents to Support Extended Mobile IP Protocols

Notice that in Softnet the addresses of the mobile agents instead of the IP addresses of the mobile nodes are registered. The mobile node can dispatch different mobile home agents or foreign agents to the same active node and therefore forms different virtual networks to support different qualities of services defined by the basic mobile protocol and the extended mobile protocols. Fig.4 shows a new virtual network supporting direct transmission between mobile nodes.



MN: mobile node; MFA: mobile foreign agent; MHA: mobile home agent;
MA: mobile agent; AN: active node.

Fig.4. Direct transmission between mobile nodes supported by Softnet.

For security reasons, the Softnet uses bi-directional tunneling between the mobile home agent and the mobile foreign agent. The path is identified as (1) in Fig.4. If the distance between mobile nodes (e.g. MN1 and MN2 in Fig.4) is very small and these mobile nodes have large amount of data to exchange, it is inefficient to use the bi-directional mobile IP protocol to pass data via MHA1 and MHA2.

The active node in the Softnet handles active packets with attached program code. In the path-building stage, program code executed on every active node will try to find and link mobile agents to form a virtual path or a virtual network for special services. To establish a virtual path or network means that every agent on the path or network must keep the addresses of other related agents in an address table. The address table in a mobile agent is similar to the routing table in a traditional router. Let us see how the direct path between MN1 and MN2 (in fact MFA1 and MFA2) can be established.

Before sending MN1's datagram, MFA1 sends an active packet via the basic routing protocol to MN2. The purpose to send this active packet is to find a better path, i.e., to see if there is a shorter path between MFA1 and MN2. Each active node (AN2, AN3, AN4, AN5 or AN6 in Fig.4) along the path (1) executes the program code in the active packet and reports the complete address table of its mobile agent (e.g., MA1, MA2, MHA2 or MFA2 in Fig.4) to MFA1. MFA1 keeps these address tables and returns its address table to those mobile agents. After analyzing the address tables, both MFA2 and MFA1 know that there is a direct path, path (2) in Fig.4, between them. Then they can use the shortcut to exchange data.

A mobile node may dispatch different mobile foreign agents to the same active node to form different virtual networks. Via different virtual networks, the mobile node can communicate to different correspondent nodes with different qualities of services. Fig.5 shows the basic condition that a single agent is used for communication with correspondent nodes CN1 and CN2, providing identical QoS.

In Fig.6 two pairs of mobile agents provide different communication paths for different correspondent nodes. Broken lines denote links of virtual networks, solid lines denote packet flows, and thick lines denote tunnels. MA is a mobile agent for mobile IP residing in AN3 which is on the path from AN2 to CN2. It can be configured in advance. Here MFA1 and MHA1 form one virtual network supporting the basic mobile IP protocol. MFA2, MHA2 and MA form another virtual network to support extended mobile IP protocols. MFA1 and MFA2 register flow types to demux on AN1 and the demux dispatches flows to MFA1 or MFA2 according to the registration, so that flows (solid line in Fig.6) sent by MN can travel through different virtual networks to CN.

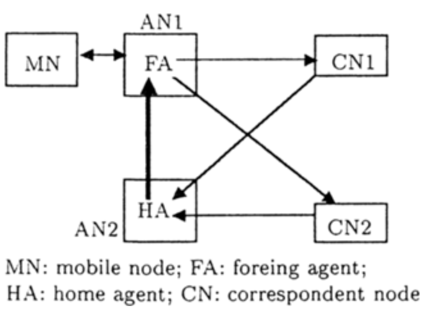


Fig.5. Basic mobile IP protocol.

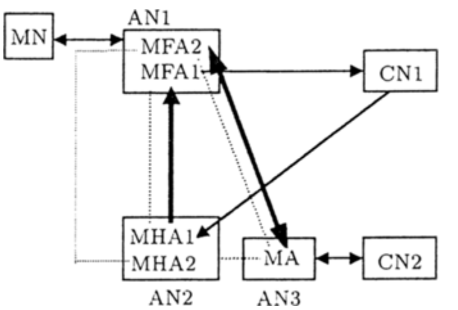


Fig.6. Mobile IP supported by Softnet.

3.4 Dispatching and Loading Mobile Agent Classes

Mobile agents implemented in the Softnet are instances of Java classes called "Magent". Let B represent Java byte code of "Magent", where

$$B = \{B_i | i \in N\},$$

B_i is the i-th fragment of B. B can be dispatched to active nodes with a series of active packets P,

$$P = \{P(\text{header, code, } B_i, \text{ payload}) | i \in N\}.$$

Fragments are assembled at the destination active node by the loader using function

$$\text{regroup}(P \text{ (header, code, } B_i, \text{ payload), packetlength)}$$

The following is the Java code for loading and creation of the mobile agent "Magent":

```
AgentClassLoader AgentLoader = new AgentClassLoader(" ",B);
Class AgentClass = AgentLoader.loadClass (classname, true);
Agent Ag =(Agent) AgentClass.newInstance();.
where code of AgentLoader. loadClass() is:
Class newClass = (Class) loadedClasses.get(classname);
if (newClass ! = null){
    if (resolve) resolveClass(newClass);
    return newClass;}
```

```
if (classname == "Magent") newClass = defineClass(B, 0,B.length);
else newClass = Class.forName(classname);.
```

The *AgentLoader* loads Java classes from the cache, buffer, or disk, but classes cannot be loaded twice. The function *loadClass*() firstly checks if the class has been loaded; if not, it loads the class from the cache or disk. If the class name is "Magent", the *AgentLoader* loads class from the buffer B.

All the mobile agent classes must be put into a dictionary. Since classes of mobile agents have the same name of "Magent", the loader cannot identify classes by their names. To facilitate the access of classes from the dictionary, the loader calculates Magents' MD5[10] value of the mobile agent as its index before putting it into the dictionary.

3.5 Linking and Executing

A mobile agent class consists of functions and variables. Function *passiveEntry* (*P*(*header, payload*), *packetlen*) handles passive packet streams. Function *activeEntry*(*P*(*header, code, data, payload*), *packetlen*) handles active packet streams. These two functions must authenticated by active nodes and registered in the execution environment. Function *Entry*() is called by an active packet to provide extended services. Function *callback*() is asynchronously called by another mobile agent to handle returned results.

Mobile agents are distributed among active nodes and their functions are called from one mobile agent to another. A program running in the Softnet needs to link functions of different mobile agents at different active nodes. Every mobile agent is assigned an ID by means of which the mobile agent can be accessed from the dictionary. The address of the mobile agent consists of the IP address of the active node where the mobile agent resides, and the MD5 value of the mobile agent class. The functions of a mobile agent are accessed by their names because each function within a mobile agent has a unique name. The type of active packet is used for finding and linking functions at different active nodes. Mobile agents exchange their addresses by sending active packets and store the addresses in tables called virtual links.

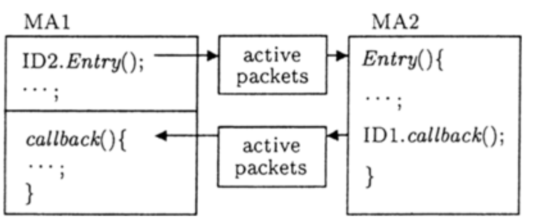


Fig.7. Asynchronous call between mobile agents.

The function *Entry*() called by the active packets leads to different service functions according to the parameters. The *Entry*() calls the caller's function *callback*() to return results by sending active packets. Fig.7 illustrates this procedure.

3.6 Optimizing Paths with Mobile Agents

The purpose of finding and linking agents described above is to build routing tables in mobile agents customized by the active packets. Because the routing table does not contain performance metrics for every path entry, the mobile agent cannot identify which is the best path to send or tunnel packets to the destination. If the mobile node or the correspondent node has a large amount of data to exchange, it is important to evaluate the performance of the path before transmission. Two factors affect the performance. First, when tunneling is performed, additional data are added to the original packets, hence reducing the overall performance of data transmission. Secondly, if the added data cause the length of the original packet to exceed the maximum length of the IP packet, fragmentation has to be performed before tunneling, increasing the total number of packets to be transmitted. In order to evaluate the effect of tunneling, an experiment is performed in the Softnet. 500 bytes are added to each original packet in tunneling. The result is shown in Fig.8.

The curves in Fig.8 illustrate the relationship between the packet size and the transmission latency in our experiment. The latency here is defined as the amount of time for a packet to be delivered from one peer to another. The latency of the direct transmission of original packets is specified with

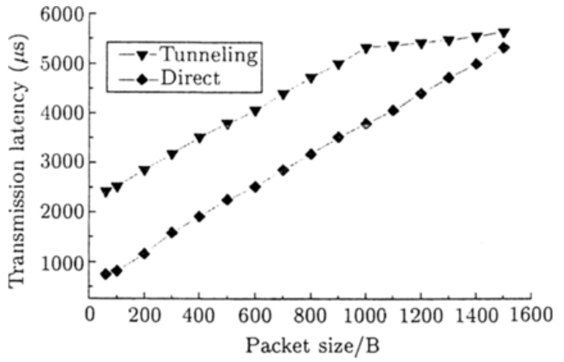


Fig.8. Relationship of packet size and transmission latency.

the lower curve in Fig.8. For comparison, the top curve of Fig.8 denotes the latency of tunneling. We can learn from the curves that the longer the packet, the higher the latency in both original and tunneled packet transmissions. The gap between these two curves reflects the effect of the added tunneling data and fragmentation. Although more data (and more packets in case of fragmentation) are transmitted, the increase of latency is not very big because of the pipelining effect of IP packet delivery in the Softnet.

We assume that the bandwidth for network segment i is U_i. T_i denotes the amount of time for router i to forward a packet, l_0 denotes the smallest packet size, and $T(l_0)$ denotes its transmission latency. The transmission latency of a packet with the size of l in bytes is

$$\begin{aligned}T(l) =&(T_1+T_2+\cdots+T_n)+(l/U_1+l/U_2+\cdots+l/U_n)\\=&(T_1+T_2+\cdots+T_n)+(l_0/U_1+l_0/U_2+\cdots+l_0/U_n)\\&+((l-l_0)/U_1+(l-l_0)/U_2+\cdots+(l-l_0)/U_n)\\=&T(l_0)+\lambda(l-l_0)\end{aligned} \tag{1}$$

where $n \in N$, $\lambda=(1/U_1+1/U_2+\cdots+1/U_n)$ is related to the transmission rate and denotes the slope of the curve in Fig.8. $T(l_0)$ is influenced by the routing and congestion conditions. λ is influenced by switching and buffering. $T(l_0)$ and λ are main parameters to evaluate a peer-to-peer path. An experiment is conducted to calculate parameters of a path between two agents.

We assume that $T_i(l)$ denotes the latency for packet with size l over path i. We compare two paths, path 1 and path 2, from agent A1 to A2 by sending active packets with a path label. When a packet arrives at A2, A2 returns the time $T_1(l)$ or $T_2(l)$ by sending an active packet calling A1's *callback*(). A1 calculates parameters $\Delta T_{12}(l_0)$, λ_1, and λ_2 according to

$$\begin{aligned}\Delta T_{12}(l_0)&=T_1(l_0)-T_2(l_0)\\ \lambda_1&=(T_1(l_1)-T_1(l_2))/(l_1-l_2)\\ \lambda_2&=(T_2(l_1)-T_2(l_2))/(l_1-l_2)\end{aligned}$$

Before forwarding a packet, A1 selects the path by the following criteria:

(1) If $\Delta T_{12}(l_0)=\infty$, it means the packet may be dropped on path 1 for security or other reasons. A1 chooses path 2 to send packets to A2.

(2) If $T_1(l) < T_2(l)$, that is, $\lambda_2(l-l_0)-\lambda_1(l-l_0) > \Delta T_{12}(l_0)$, A1 chooses path 1, else, A1 chooses path 2 to send packets to A2.

4 Conclusions

The active network provides a programmable network infrastructure with which the basic mobile IP protocol and other extended mobile IP protocols can be efficiently implemented. The dynamic and flexible nature of the active network paves the way for providing different qualities of services for the mobile IP. In our active network prototype, the Softnet, virtual networks supporting the transmission between mobile nodes can be formed dynamically by distributing and linking mobile agents in active nodes. Security and optimization issues can also be taken into consideration with the programmability of the active network.

References

[1] Perkins C. IP Mobility support. ftp://ftp.xanet.edu.cn/pub/internet/rfc/rfc0000/rfc2002.txt, October, 1996.
[2] David L Tennenhouse, Jonathan M Smith, W David Sincoskie *et al.* A survey of active network research. *IEEE Communication Magazine,* Jan., 1997, 35(1): 80–86.
[3] Smoot Carl-Mitchell, John S Quarterman. Using ARP to implement transparent subnet gateway. ftp://ftp.xanet.edu.cn/pub/internet/rfc/rfc0000/rfc1027.txt, October, 1987.
[4] Stuart Cheshire, Mary Baker. Internet mobility 4×4. In *SIGCOMM'96,* Stanford, California, USA, August, 1996.
[5] David B Johnson, Charles Perkins. Routing optimization in mobile IP. Internet Draft, November, 1995.
[6] Gupta V, Glass S. Firewall traversal for mobile IP: Guidelines for firewall and mobile IP entities. Internet Draft, March, 1997.
[7] Montenegro G. Reverse tunneling for mobile IP. Internet Draft, March, 1997.
[8] Xinhua Zhao, Mary Baker. Flexible connectivity management for mobile hosts. Technical Report CSL-TR-97-735, Stanford University, September, 1997.
[9] Active Networks Working Group. Architectural Framework for Active Networks, Version 0.9: http://www.cc.gatech.edu/projects/canes/arch/arch-0-9.ps, August, 1998.
[10] Rivest R. The MD5 message-digest algorithm. ftp://ftp.xanet.edu.cn/pub/internet/rfc/rfc0000/rfc1321.txt, April, 1992.

LU Yueming received the B.E. and M.E. degrees in computer application from Xi'an University of Architecture and Technology. He is a Ph.D. candidate of Xi'an Jiaotong University now. His current research interests include active networks, network security and network management.

QIAN Depei, born in 1952, is currently a professor in the Department of Computer Science and Technology, Xi'an Jiaotong University. His research interests include high performance computer architecture, active networks, and network management.

XU Bin, born in 1974, received his Ph.D. degree in computer architecture from Xi'an Jiaotong University in 2001. He is currently a research staff in the Bell Lab., China. His major research interests include active network and network management.

WANG Lei, born in 1976, received his B.A. and M.A. degrees from Xi'an Jiaotong University in 1998 and 2001, respectively, both in computer architecture. He is currently an engineer in the Microsoft Research Lab., China. His research interests include network performance measurement, firewall technology, and network QoS control.