An Approach for Building Scalable Proxy Mobile IPv6 Domains

Hongbin Luo, Hongke Zhang, Yajuan Qin, and Victor C. M. Leung, *Fellow, IEEE*

***Abstract*—As a promising network-based mobility management method that does not require active participation of mobile nodes (MNs), Proxy Mobile IPv6 (PMIPv6) is attracting considerable attention among the telecommunication and Internet communities. It remains an open issue how to build a scalable PMIPv6 domain that is able to support a large number of MNs while keeping handover delays low. In this paper, we propose an approach for building Scalable And Robust PMIPv6 (SARP) domains. We propose that every mobility access gateway (MAG) in a SARP domain also functions as a local mobility anchor (LMA), and is organized into a virtual ring with all other MAGs. Consistent hashing is used to efficiently distribute the mapping between each MN and its LMA to all MAGs. A MAG finds an MN's LMA by sending a query message to the virtual ring. Our analysis verifies the robustness and scalability of SARP. We also propose two handover procedures for SARP and show that they achieve low handover delays.**

***Index Terms*—Mobility management, Proxy Mobile IPv6 (PMIPv6), scalability, robustness, distributed hash table.**

I. Introduction

THE success of mobile wireless networks and the rapid growth in the number of mobile subscribers and mobile devices such as cellular phones, personal digital assistants, and laptop computers lead to an increasing demand for mobile wireless access to Internet applications [1], [2]. Different wireless access network technologies such as IEEE 802.11a/b/g Wireless-Fidelity (WiFi), 802.16 World Interoperability for Microwave Access (WiMAX), and General Packet Radio Service employ specific methods to support inter-technology handover with fairly low latency. Since these diverse wireless access networks are converging in terms of their universal support of the Internet protocol (IP) suite and the use of IP in their core infrastructures, it is increasingly important to enable mobile nodes (MNs) with multiple-technology access capabilities to seamlessly roam across heterogeneous networks while enjoying the plethora of "all-IP-based" services. To address the challenges arising from inter-technology "vertical" handovers, mobile Internet protocols (MIPs) [4] - [9] have been proposed and standardized by the Internet Engineering Task Force (IETF).

In general, these protocols can be classified into two categories: host-based protocols and network-based protocols. Host-based protocols require protocol stack modifications of MNs. This requirement not only causes increased complexity in MNs but also is considered as one of the primary reasons that host-based protocols have not been widely deployed in the past years [1], [10], [11]. On the contrary, in network-based approaches, the serving network performs the mobility management on behalf of MNs so that they are not required to participate in any mobility-related signaling. As a result, network-based approaches do not require any modification of MNs, which facilitates network service providers to offer services to as many customers as possible [11]. In addition, compared to host-based mobility management approaches, network-based approaches improve resource utilization and reduce handover latency since MNs do not participate in mobility-related signaling [1]. Because of these salient features, network-based mobility management approaches are attracting considerable attention among the telecommunication and Internet communities.

Proxy mobile IPv6 (PMIPv6) is a network-based mobility management approach that was standardized by the IETF network-based localized mobility management (NETLMM) working group and specified in Request for Comments (RFC) 5213 [9]. With PMIPv6, an unmodified IP node may change access router without changing the IP address on an interface, within a given administrative IP domain. While such an IP domain may be as small as a localized company/campus network, it may also cover a large area with plenty of users, such as a metropolis or even a country. For example, China Mobile has more than ten million users in Beijing and may wish to cover the whole Beijing city with one PMIPv6 domain.

To cover such a large area, one approach is to use a PMIPv6 domain that maintains a single local mobility anchor (LMA). Since this LMA needs to intercept packets to all MNs in the PMIPv6 domain, the computation load to process intercepted packets and proxy binding update messages limits the size of the PMIPv6 domain served by this single LMA. For example, it is unlikely that China Mobile can use a single LMA to effectively serve its huge number of MNs in Beijing. As a result, a PMIPv6 domain cannot scale well with this approach. Another approach is to cover such an area by using a PMIPv6 domain that maintains multiple LMAs and letting each LMA

Manuscript received July 21, 2010; revised March 22, 2011. The associate editor coordinating the review of this paper and approving it for publication was R. Campbell.

This work was supported in part by the National Basic Research Program of China ("973 Program") under Grant No. 2007CB307100; in part by the National Natural Science Foundation of China (NSFC) under Grant No. 60833002 and 60702008; in part by the "111 Project" of China under Grant No. B08002; and in part by the "863 Project" under Grant No. 2007AA01Z202. Part of this work appeared in the Proc. of IEEE MobiWorld'10.

H. Luo, H. Zhang, and Y. Qin are with the School of Electronic and Information Engineering, Beijing Jiaotong University (BJTU), Beijing 100044, China. They are also with the National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU (e-mail: hbluo@bjtu.edu.cn).

V. Leung is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4 (e-mail: vleung@ece.ubc.ca).

Digital Object Identifier 10.1109/TNSM.2011.071511.20100063

1932-4537/11/$25.00 © 2011 IEEE

manage a subset of all MNs in the domain [12]. This approach, however, entails careful assignment of MNs to LMAs so that an LMA will not be overloaded. A third approach is to cover such an area by using multiple smaller PMIPv6 domains that each uses a single LMA. In reality, however, we daily roam from place to place. For instance, many people in Beijing live in the outskirts but work in the city center, and most of them roam from home to office in the morning and go back home in the evening. Thus with the third approach, it is possible for an MN to move across different PMIPv6 domains during a movement. Indeed, in our analysis shown in Section IV.C, it is possible that, in big cites, an MN moves across more than ten smaller PMIPv6 domains during a trip. In practice, MIPv6 is generally used to deal with handovers of these inter-domain movements. As is well known, however, the handover delay of inter-domain movements is significantly longer than that of an intra-domain movement in PMIPv6 and is unacceptable for many applications. It is therefore preferable that a single PMIPv6 domain can cover the area over which most MNs roam on a daily basis, e.g., the metropolitan area of Beijing.

Therefore, it is of great importance to address the problem of building scalable PMIPv6 domains that can serve a large number of MNs over a large area, while keeping handover delays very small. While research on PMIPv6 [13] - [18] is gaining interest in recent years, to our knowledge, the above problem is still open. In this paper, we present a novel solution that solves this problem. We make two main contributions.

First, we propose an approach for building Scalable And Robust PMIPv6 (SARP) domains. SARP has four main merits. 1) It is scalable. For the first time in the literature, we propose that every mobile access gateway (MAG) behaves as both a MAG and an LMA. All MAGs in a PMIPv6 domain are then organized into a virtual ring by using Chord [19] and (key = the hash of an MN-identifier, value = the IPv6 address of MN's LMA) pairs are distributed to all MAGs using consistent hashing. In order to query the LMA of an MN, a MAG only needs to send query messages to the MAG that stores the MN's LMA address. Notice that, while Chord is a well-known distributed hash table (DHT) protocol, the application of Chord in the context of this paper is novel. 2) SARP provides a mechanism for enhancing the security of a PMIPv6 domain. 3) By using a fast handover procedure, SARP keeps the handover delay very low. 4) SARP makes it easier to realize load balancing. While the binding update messages and data packets are sent to one or more dedicated LMAs in existing approaches, a MAG in SARP only receives a small fraction of these messages and packets.

Second, we analyze the performance of SARP in depth and show that, with SARP, a single PMIPv6 domain is able to support more than 10^8 MNs, which is significantly larger than the population of most big cities. By contrast, if we use many small PMIPv6 domains to cover a big city, a single movement incurs multiple inter-domain handovers with handover delays that are significantly longer than the handover delays of intra-domain handovers in PMIPv6.

The rest of this paper is organized as follows. Section II presents related work. Section III describes the proposed SARP scheme in detail. Section IV presents analytical and numerical results. Section V concludes the paper.

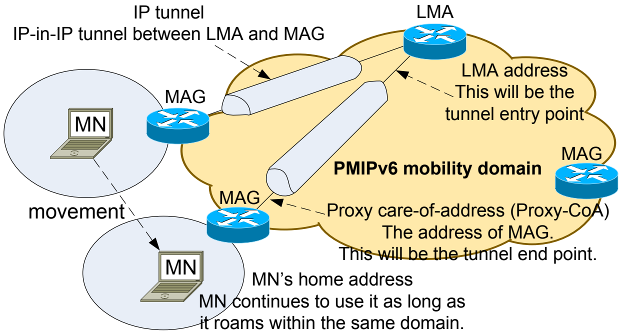

IP tunnel
IP-in-IP tunnel between LMA and MAG
LMA
LMA address
This will be the
tunnel entry point
MAG
MN
MAG
PMIPv6 mobility domain
movement
MAG
Proxy care-of-address (Proxy-CoA)
The address of MAG.
This will be the tunnel end point.
MN
MN's home address
MN continues to use it as long as
it roams within the same domain.

Fig. 1. Overview of PMIPv6.

II. Related Work

In this section, we first give a brief introduction on PMIPv6, focusing on its handover procedure. Since we use Chord [19] to logically organize MAGs in SARP, we then give an overview of Chord.

A. Proxy Mobile IPv6

PMIPv6 enables IP mobility for an MN without requiring its participation in any mobility-related signaling [9]. Instead, two network entities, *i.e.*, MAG and LMA, are used for managing IP mobility on behalf of the MN in a PMIPv6 domain. Each MN is served by a pre-configured LMA within the domain. The MAG detects the MN's movement and initiates signaling with the MN's LMA to update the route to the MN's home address (HoA), emulates the MN's home link on the access link, and sets up a tunnel between it and the MN's LMA so that the MN can use its HoA for communications over the access link. Behaving as a home agent as defined in [6], LMA processes Proxy Binding Updates (PBU) and Proxy Binding Acknowledgement (PBA) messages. The LMA also intercepts packets enroute to MNs it is serving, processes and tunnels intercepted packets to the appropriate MAGs, and supports additional capabilities required by PMIPv6 [9].

Figure 1 illustrates mobility support within a PMIPv6 domain. In particular, the PMIPv6 domain learns the MN's HoA and assigns a corresponding home network prefix that conceptually follows the MN wherever it moves within the PMIPv6 domain. Whenever a MAG detects the MN's attachment, it provides the MN's home network prefix on the access link. Thus from the perspective of the MN, the entire PMIPv6 domain appears as its home network and the MN does not need to configure care-of-address (CoA) when it roams within the PMIPv6 domain.

When an MN roams from the previous MAG (pMAG) to a new MAG (nMAG) in a PMIPv6 domain, the handover procedure illustrated in Figure 2 is executed.

Step 1) When the MN attaches to nMAG, an access authentication procedure is performed using the MN's identity (*i.e.*, MN-identifier). Upon successful access authentication, nMAG knows the MN's identity.

Step 2) The nMAG sends a query message to the policy store (*e.g.*, authentication, authorization, and accounting (AAA) server) to obtain the MN's configuration profile.

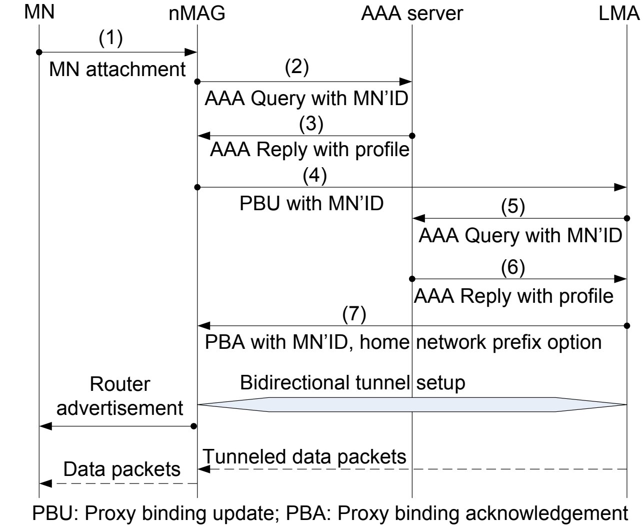
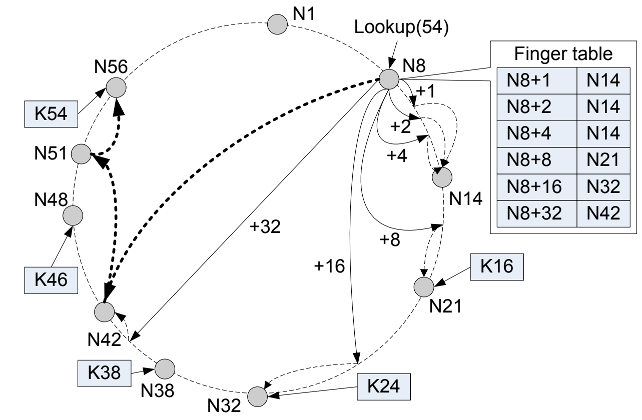


Fig. 2. Handover operations of PMIPv6.



N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

Fig. 3. Illustration for Chord when m = 6 [19].

Step 3) The policy store sends the MN's profile including the MN's MN-identifier, LMA address, and supported address configuration mode to nMAG.

Step 4) The nMAG then sends a PBU message including the MN-identifier to the MN's LMA on behalf of the MN.

Step 5) When it receives the PBU message, the LMA checks the policy store to ensure that the sender is authorized to send the PBU message.

Step 6) The policy store sends a reply to the LMA to indicate whether or not the sender is authorized.

Step 7) If the sender is authorized, the LMA sends a proxy binding acknowledgment (PBA) message including the MN's home network prefix option, and sets up a route for the MN's home network prefix over a tunnel to nMAG. The two tunnel end points include the LMA address and the nMAG's address, as illustrated in Figure 1. The LMA should also update its binding cache.

Once the nMAG receives the PBA message, it has obtained all the required information to emulate the MN's home network on the access network. It then sends to the MN a router advertisement (RA) message that contains the MN's home network prefix and sets up a tunnel to the LMA. Then the LMA tunnels subsequent packets from any corresponding node (CN) to the nMAG for delivery to the MN.

In this paper, we propose an approach for building SARP domains. Notice that our approach does not conflict with the proposals developed in the IETF NETLMM working group. On the other hand, it naturally complements those proposals.

B. Chord

Chord is a distributed protocol used for locating nodes based on keys in a scalable manner. In an N-node Chord system, each node only needs to maintain information about $O(\log N)$ other nodes, and resolves all lookups via $O(\log N)$ messages to other nodes. In addition, the average lookup time in an N-node Chord system is $(1/2)\log N$ [19].

For this purpose, Chord assigns keys to nodes with consistent hashing [20], which assigns an m-bit identifier using SHA-1 [21] to every node and key. In particular, the identifier of a node is obtained by hashing the node's IP address, and that of a key is obtained by hashing the key. In order to make the probability of two nodes or keys hashing to the same identifier negligible, the identifier length m must be large enough.

Identifiers are ordered on a Chord circle modulo 2^m [19], as illustrated in Figure 3. Let the successor of a key k be the node whose identifier is equal to or follows (the identifier of) k in the identifier space. For ease of presentation, we denote the successor of a key k by $successor(k)$. Accordingly, $successor(k)$ is the first node clockwise from k if identifiers are represented as a circle of numbers from 0 to $2^m - 1$. Given a key k, it is assigned to $successor(k)$. For example, key 46 in Figure 3 is assigned to node N48 since node N48 is its successor.

In Chord, nodes can enter and leave the network with minimal disruption. In particular, when a node n joins the network, some keys previously assigned to n's successor are assigned to n. On the other hand, when node n leaves the network, all of its assigned keys are reassigned to n's successor. Assignments of keys to other nodes need not be changed. In Figure 3, the node with identifier 26 obtains keys with identifiers between 22 and 26 from the node with identifier 32 when it joins.

For efficient lookup, every Chord node maintains a finger table that stores at most m entries. Each entry includes both the Chord identifier and the IP address (and port number) of the relevant node. In addition, for a node n, the ith entry in its finger table contains the identity of the first node s that succeeds n by at least 2^{i-1} on the Chord circle. That is, the ith finger of node n is s = $successor(n + 2^{i-1})$, where $1 \leq i \leq m$. The finger table of node 8 is shown in Figure 3. As shown, node 8's first finger points to node 14 since node 14 is the first node that succeeds $(8 + 2^0)$ mod 2^6 = 9, where *mod* denotes modulo. Similarly, node 8's last finger points to node 42 because node 42 is the first node that succeeds $(8 + 2^5)$ mod 2^6 = 40.

To look up the value for a key k, a node n only needs to look up its finger table and sends the lookup request to the Chord node whose identifier is the largest finger (of Chord node n) that precedes k. Suppose node N8 in Figure 3 wants to find the value for key 54. Since the largest finger of node N8 that precedes 54 is node N42, node N8 sends the lookup request to node N42. In turn, node N42 determines the largest finger in its finger table that precedes 54, *i.e.*, node N51. Finally, the lookup request will be forwarded to Chord node N56, which stores the value for key 54.

III. The Proposed Approach for SARP

In this section, we describe our proposed approach in detail. We begin with the basic design of SARP, which is followed by a description of security considerations, how SARP realizes robustness, and handover operations in SARP.

A. Basic Design

The goal of SARP is to provide scalable and robust PMIPv6. For this purpose, we assume that there are many MAGs in a PMIPv6 domain and that every MAG has a MAG identifier that is similar to an MN-identifier specified in [22]. In addition, all MAGs in a PMIPv6 domain are organized into a Chord circle using consistent hashing over MAG identifiers. Most importantly, every MAG in SARP functions as both an LMA for those MNs that it is pre-configured to serve, and a MAG for those MNs that are attached to it over some access links. For a given MN, its LMA in a PMIPv6 domain is configured by the network administrator of the domain or by some algorithm and does not change as long as the MN roams within the PMIPv6 domain. In practice, an MN is often located at specific areas since most of us work daily for about eight hours at the office during the day and go back home in the evening. That is, an MN is often attached to a MAG covering the location of its owner's home or work place. If we choose such a MAG as the LMA for the MN, with high probability, the LMA/MAG can directly send packets to the MN without tunneling them to another MAG; i.e., the MN's LMA needs to tunnel packets to another MAG only when the MN is not attached to its LMA. In contrast, packets sent to an MN in a typical PMIPv6 domain are always intercepted by the LMA serving the MN, which in turn tunnels the packets to the MAG that the MN attaches to, even if the MN rarely moves away from the MAG. As a result, our approach to collocate the LMA with the MAG most often covering the MN significantly improves resource efficiency since the tunneling overheads are reduced.

As stated above, the LMA of an MN may be configured by the network administrator and is not well-known. As a result, when a MAG detects the attachment of an MN, it may not know which MAG is the MN's LMA. In order to deal with this issue, we store *(key, value)* pairs at the MAGs via a simple hashing mechanism. Here the key is the hash value of an MN-identifier and the value is the IPv6 address of the LMA serving the corresponding MN. For an MN with a given MN-identifier, the IPv6 address of its LMA should be stored at the MAG that is the successor of the MN-identifier. In the example shown in Figure 3, if an MN's MN-identifier corresponds to key 24, the (key, value) pair for the MN would be stored at the MAG corresponding to node N32. For ease of presentation, we denote the MAG that stores the (key, value) pair for an MN as *QServer(MN)* since other MAGs sends query messages to it in order to find the MN's LMA when they detect the attachment of the MN. With this notation, the QServer for key 24 in Figure 3 would be node N32. That is, QServer(24) = 32.

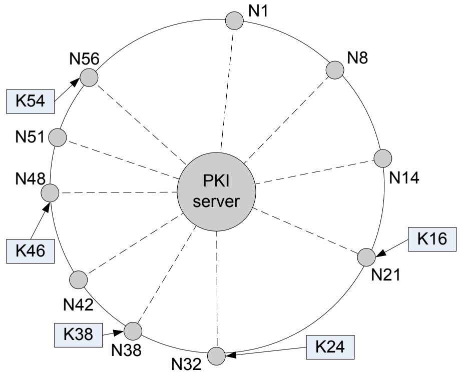


Fig. 4. Illustration for a Chord system with a PKI server in SARP.

Whenever a new MAG detects the attachment of an MN, it may not know the MN's LMA. In this case, it sends a query message to the Chord system in order to find the MN's LMA. The query message should include the MN-identifier, the new MAG's identifier and IPv6 address. The Chord nodes forward the query message until it reaches QServer(MN). For efficient forwarding, as stated above, every MAG should maintain a finger table that maps hash values of MN-identifiers onto IPv6 addresses. When the MAG that serves as the QServer of the MN receives the query message, it sends the IPv6 address of the LMA serving the MN to the new MAG. When the new MAG receives the IPv6 address of the MN's LMA, it stores the LMA's IPv6 address locally. For this purpose, the new MAG needs to maintain a table that records the IPv6 address of the LMA serving each attached MN.

Note that the query message may be forwarded over multiple hops in the Chord ring, leading to a long lookup delay. From this perspective, it may be better to organize MAGs into a one-hop DHT [23]. However, since our goal is to build scalable large PMIPv6 domains, a large PMIPv6 domain may be partitioned into multiple routing domains in order to simplify network management and troubleshooting. In that case, a MAG in a PMIPv6 domain may not know the routes to the other MAGs in the PMIPv6 domain and one-hop DHTs may not be applicable.

B. Security Considerations

An important issue in using DHT-based protocols such as Chord is that a node may not trust other nodes. Fortunately, the MAGs in the Chord system described above are located in a common PMIPv6 domain, which is typically managed by a single network provider. As a result, there is no need to protect a MAG from being attacked by other MAGs. However, it is still necessary to protect MAGs from being spoofed by mobile hosts.

For this purpose, we propose to utilize the public key infrastructure (PKI), by setting a PKI server in a SARP domain as illustrated in Figure 4. The PKI server is used to assign public/private key pairs to the MAGs in the same SARP domain. The public key assigned to a MAG is known by all MAGs in the same SARP domain. On the other hand, the private key assigned to a MAG is only known by the MAG and the PKI server. When a MAG sends messages to other MAGs in the same SARP domain, it signs them by using its private key. When other MAGs receives these messages, they verify the authenticity of the messages by using the public key of the MAG.

When a new MAG joins the Chord system, it first obtains a public/private key pair from the PKI server in the same SARP domain. The new MAG then uses the obtained public/private key pair to verify its identity to its successor in the Chord system. The successor can verify the new MAG's authenticity by communicating with the PKI server. Only when the new MAG's identity has been authenticated, would the successor send to the new MAG the (key, value) pairs of those MNs whose MN-identifiers hash to the new MAG.

Before a MAG sends a query message to the Chord system, it should sign the message using its private key. When other MAGs receive this query message, they can verify its authenticity using the MAG's public key. When the MN's QServer sends a reply message to the MAG, it should also sign the reply message so that the MAG can verify the authenticity of the reply message. Similarly, the MAGs should sign the PBU messages before they send these PBU messages to the corresponding LMAs, and the LMAs should sign the PBA messages that are returned to the requesting MAGs.

Note that the PKI server is only responsible for issuing public/private key pairs to MAGs and does not provide AAA services to MNs. Instead, if a MAG is chosen as the LMA of an MN, it would provide the AAA service for the MN. That is, AAA service in SARP is also distributed to all MAGs. Because of the use of PKI in the Chord system, the AAA service provided by MAGs could be trusted. By doing this, not only can we avoid using high performance servers to provide AAA service, but we can also reduce handover latency since there is no need for the MAG and the LMA to communicate with AAA servers in order to verify the identity of an MN and that of the MAG. Furthermore, it is reported that, on average more than 30% source-destination pairs in Global System for Mobile communications (GSM) networks are attached to common base station controllers [24]. In some areas, *e.g.*, some developing countries, this fraction could be as high as 60% [24]. With the fast development of data centers and nano data centers [25] - [27], it is anticipated that MNs would obtain most of their desired data from (nano) data centers that are attached to a common MAG with the MN. As a result, letting the MN's LMA (instead of a AAA server) store the MN's profile significantly reduces the number of AAA queries.

It is worth noting that the network operator of a SARP domain needs to maintain a subscriber information database. This database stores the complete subscription information of all MNs in the domain and can reside in the PKI server of the domain. In order to achieve the distributed AAA service described above, it is only required to distribute, for a given MN, the MN's subscription information to the MN's LMA. Once a new MN subscribes to the SARP domain and is assigned an LMA, the subscription information of the MN is sent to the MN's LMA. We notice that the number of new subscribers is very small when compared with the processing capability of modern routers. For example, on average, the number of new subscribers of China Mobile in Beijing is less than 5,000 per day. This way, it is not necessary to synchronize multiple databases and the AAA service is distributed to all MAGs in a SARP domain.

C. Robustness

Robustness is another issue worthy of particular attention. As stated above, the Chord system is robust since the failure of a MAG only affects a very limited number of MAGs in the Chord system. In addition, unlike general networking situations in which nodes may join and/or depart frequently, MAGs in a PMIPv6 domain rarely depart once they are put into use. While some new MAGs may join the system when the network size increases, the joining of new MAGs does not cause data loss, although this may lead to the change of some MNs' QServers.

However, it is possible that a MAG stops working, e.g., during maintenance or due to failure. Should this happen in a SARP domain, the (key, value) pairs stored at the MAG would be lost and the system cannot find the LMAs for the MNs whose QServer is the failed MAG. In order to address this issue, we assume that the MAGs periodically (*e.g.*, once several minutes) send their (key, value) pairs to the PKI server, which would store all (key, value) pairs in the SARP domain. Notice that there is no need for the MAGs to frequently send their (key, value) pairs to the PKI server since the LMA of an MN does not change frequently. While one may say that the MAG of an MN may change frequently, this change does not affect the (key, value) pairs stored at a MAG.

When a MAG fails, the PKI server sends all (key, value) pairs that should be stored at the failed MAG to its successor. Therefore, the successor of the failed MAG is able to answer the query messages for the MNs whose (key, value) pairs had been stored at the failed MAG. As a result, the failure of a MAG would not affect the operation of the PMIPv6 system since the living MAGs would cover for the failed MAGs.

Notice that, like the standard PMIPv6, the failure of a MAG would lead to the MNs attached to the MAG being unable to access the Internet. Since MNs communicate with their MAGs through access points (APs), this can be dealt with by letting an AP communicate with two MAGs: one for normal use and the other for use when the first one fails.

Finally, the failure of a MAG may cause the MNs whose LMA are the failed MAG to be unable to receive packets from their CNs. Given the fact that these packets have been routed to the SARP domain, they may be intercepted by the network routers and then directed to some other MAGs (*e.g.*, the successor of the failed MAG) or the PKI server, which would then tunnel them to the alternate MAGs replacing the failed MAGs.

While one may argue that it would be better to use a distributed mechanism to guarantee the robustness in the case that the PKI server fails, we note that both MAGs and the PKI server are highly reliable equipment with built-in redundancy deployed for carrier-grade service. Therefore, we feel that it is sufficient to use a single working PKI server protected by a backup PKI server.

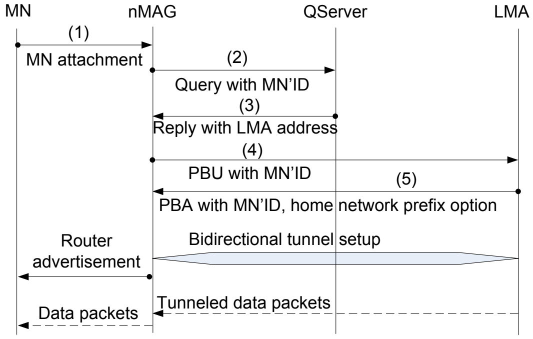


Fig. 5. Illustration for a basic handover procedure in SARP.

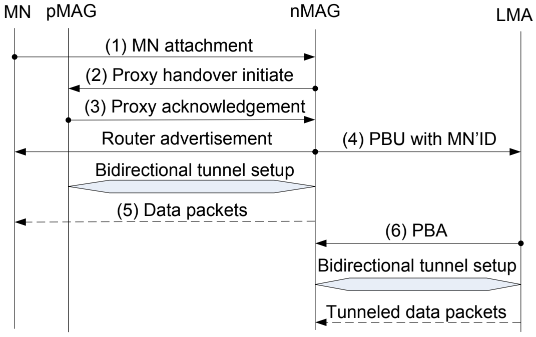


Fig. 6. Handover message flows for fast handover in SARP.

D. Handover in SARP

When an MN roams to a new MAG (nMAG) while communicating with a CN, nMAG would initiate a handover procedure in order to maintain communications with the CN. In this section, we first propose a basic handover procedure, which is illustrated in Figure 5 and comprises the following steps.

Step 1) When MN attaches to nMAG, an access authentication procedure is performed by using the MN's identity (*i.e.*, the MN-identifier). Upon successful access authentication, nMAG knows the MN's identity.

Step 2) The nMAG sends a query message to the Chord system in order to obtain the IPv6 address of the MN's LMA. The query message would be routed by the Chord nodes until it reaches the MN's QServer. As stated above, the query message should be signed by nMAG using its private key.

Step 3) The MN's QServer sends the IPv6 address of the MN's LMA to n MAG. For security, the QServer should also sign its message sent to nMAG.

Step 4) The nMAG then sends a PBU message including the MN's identifier to the MN's LMA on behalf of the MN. Again, nMAG should sign the PBU message so that the MN's LMA can verify its authenticity.

Step 5) The LMA checks the authenticity of the PBU message by using nMAG's public key when it receives the PBU message. If nMAG is authenticated, the LMA signs and returns a PBA message including the MN's identifier, the MN's supported address configuration mode, and the MN's home network prefix option to nMAG. It also sets up a route for the MN's home network prefix over the tunnel to nMAG.

Once nMAG receives the PBA message, the following steps are like those specified in standard PMIPv6 [9]. From the above procedure, one can see that the biggest difference is that nMAG and the LMA of the MN do not need to query the AAA server. On the other hand, nMAG should first send a message into the Chord system in order to obtain the IPv6 address of the MN's LMA. Furthermore, all messages are signed by the senders using their private keys. As a result, their authenticity would be guaranteed. Therefore, in contrast to standard PMIPv6, it is not necessary in SARP to query a AAA server since an MN's profile is stored in its LMA.

However, the above presented handover procedure may incur long handover delays since a query message sent by nMAG may be forwarded in the Chord system over several hops. In order to address this issue, we propose to use the fast handovers for PMIPv6 (F-PMIPv6) scheme proposed in [14]. Unfortunately, since nMAG does not know which MAG serves as the MN's LMA, we cannot directly use F-PMIPv6 in SARP and have to modify it.

Figure 6 shows the handover procedure of the modified fast handover for PMIPv6, which comprises the following steps.

Step 1) When MN attaches to nMAG, an access authentication procedure is performed by using the MN's identity (*i.e.*, the MN-identifier). Upon successful access authentication, nMAG knows the MN's identity. This step is like Step 1) in the basic handover procedure.

Step 2) The new MAG sends a proxy handover initiate (PHI) message to the previous MAG (pMAG) of the MN. The PHI message should include the MN-identifier.

Step 3) Upon receiving the PHI message, pMAG responses with a proxy acknowledgement (PA) message to nMAG, which include the IPv6 address of the MN's LMA and the MN's profile such as the MN's identifier, the MN's supported address configuration mode, and the MN's home network prefix option.

Step 4) When nMAG receives the MN's profile, it emulates the MN's home network and sends a router advertisement (RA) message to the MN. At the same time, it sends a PBU message to the MN's LMA.

Step 5) Data packets can be forwarded directly through the tunnel between nMAG and pMAG. When nMAG receives packets from pMAG, it sends them to the MN.

Step 6) After the MN's LMA receives the PBU message from nMAG, it updates the MN's location at its binding cache entry, and sends a PBA message to nMAG. In addition, it sets up a tunnel to nMAG. The subsequent packets are sent directly from the MN's LMA to nMAG.

Step 7) When nMAG receives the PBA message from the MN's LMA, it sets up a route for the MN's home network prefix over the tunnel to the MN's LMA. In addition, nMAG tears down the tunnel towards pMAG.

Notice that, for security, all messages should be signed by the senders using their private keys. In addition, nMAG may send the PHI message in a way similar to how a new access router sends a router solicitation for proxy (RtSolPr) to the previous access router as in [7]. As will be shown in the next section, the handover latency in the fast handover for SARP is significantly less than that in the basic handover for SARP.

For comparison, we show the handover message flows of fast PMIPv6 [14] in Figure 7. Comparing Figure 6 and

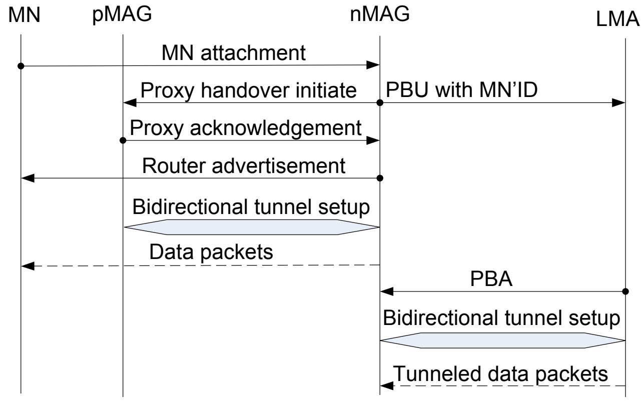


Fig. 7. Handover message flows in fast PMIPv6.

Figure 7, one can clearly observe that the difference between fast handover for PMIPv6 and fast handover for SARP is that nMAG sends the PBU message at a different time. In fast handover for PMIPv6, nMAG sends the PBU message to the LMA at the same time when it sends the PHI message to pMAG. On the other hand, in fast handover for SARP, nMAG sends the PBU message to the LMA after it receives the proxy acknowledgement message from pMAG. This is because, in fast handover for SARP, nMAG does not know which MAG serves as the MN's LMA. As a result, it cannot know where to send the PBU message. By contrast, nMAG in fast handover for PMIPv6 knows the MN's LMA. As a result, it can directly send the PBU message to the MN's LMA. Notice that this difference does not lead to differences in handover delays between fast handover for SARP and fast handover for PMIPv6. A side effect of this difference is that, in fast handover for SARP, more packets will be sent to nMAG through pMAG. This is because in fast handover for SARP nMAG sends the PBU message to the MN's LMA later than that in fast handover for PMIPv6. Comparing Figure 6 and Figure 7, we observe that this delay equals a round trip time between pMAG and nMAG. Since pMAG and nMAG are in the same PMIPv6 domain, a round trip time between them ranges from several to several tens of milliseconds in general. When compared with the residence time of an MN in the area covered by a MAG, however, this delay is fairly small since the residence time is in the order of tens of seconds, as will be shown in the next section.

IV. Numerical Results

In this section, we first analyze the feasibility of SARP. We then compare the handover latencies of standard PMIPv6, fast handover for PMIPv6, basic handover for SARP, and fast handover for SARP. Finally, we give results to show the merits of SARP over standard PMIPv6.

A. Feasibility Analysis

We let N and M be the numbers of MAGs and MNs in a PMIPv6 domain, respectively. We further let C be the capability of a MAG to process requests and p be the average number of MAG handovers per second caused by a single MN. We analyze the feasibility of SARP from two perspectives: storage requirements and processing capability.

1) Storage Requirements: A MAG needs to maintain four tables: a table that stores a binding update entry and a policy profile for every attached MN, a table that records a binding cache entry for each MN whose LMA is the MAG, a table that stores the profile of each MN whose LMA is the MAG, and a finger table for efficient lookup in the Chord system.

For every attached MN, a MAG needs to store a binding update entry and a policy profile for it. A binding update entry includes the MN-identifier (at most 128 bits), the link layer identifier of the MN's connected interface (16 bits), a list of IPv6 home network prefixes assigned to the MN's connected interface (each requires 128 bits), the link-local address of the MAG on the access link shared with the MN (128 bits), the IPv6 address of the MN's LMA (128 bits), the interface of the point-to-point link between the MAG and the MN (at most 128 bits), the tunnel interface identifier of the bi-directional tunnel between the MN's LMA and MAG (at most 128 bits), and other fields (about 500 bits) specified in Section 11.1 of [6]. A policy profile would include the MN-identifier (at most 128 bits), the IPv6 address of the MN's LMA (128 bits), and other optional fields (at most 200 bits) [9]. The sum of all the fields entails a storage space of about (1600 + 128 * s) bits in a MAG to store an MN's binding update entry and policy profile, where s is the number the IPv6 home network prefixes assigned to the MN's connected interface. To make an overestimation, we assume that a MAG needs a storage space of about 3000 bits to store an MN's binding update entry and policy profile.

Similarly, we assume that a MAG needs a storage space of about 3000 bits to store an MN's binding cache entry and policy profile for each MN whose LMA is the MAG, since the fields in a binding update entry and those in a binding cache entry are similar.

In a finger table, every entry only includes a key (assuming 128 bits) and an IPv6 address of the next hop (128 bits). Therefore, an entry in a finger table entails a 256 bits storage space. However, since the number of entries in a finger table is significantly less than that of MNs, and the storage space required to store an MN's binding cache entry is significantly larger than that required to store an entry in a finger table, we neglect the storage space requirement of a finger table.

Therefore, an MN needs a storage space of about 6,000 bits. With current technology, a single DRAM is able to provide a storage space of 2 gigabits. Thus, a MAG is able to store information for over 300,000 MNs.

2) Processing Capability: Another factor that limits the performance of SARP is the processing capability of the MAGs. Whenever an MN attaches to a new MAG, the new MAG would send a query message to the Chord system in order to learn the IPv6 address of the MN's LMA. For an N-node Chord system, it entails $\log N/2$ hops on average to route a query message. That is, a query message would be processed $\log N/2$ times on average by the Chord system. When the QServer of an MN receives a query message, it would look up its cache in order to find the IPv6 address of the MN's LMA. When the new MAG of the MN finds the LMA of the MN, it would send a PBU message to the MN's LMA, which would send a PBA message to the MN's MAG. Therefore, a handover of an MN would incur

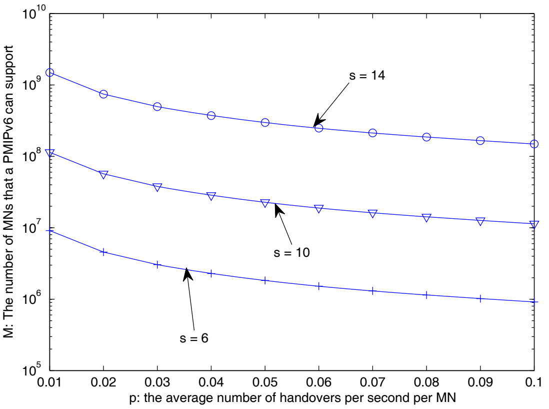


Fig. 8. The number of MNs that a SARP domain can support when p and s vary.

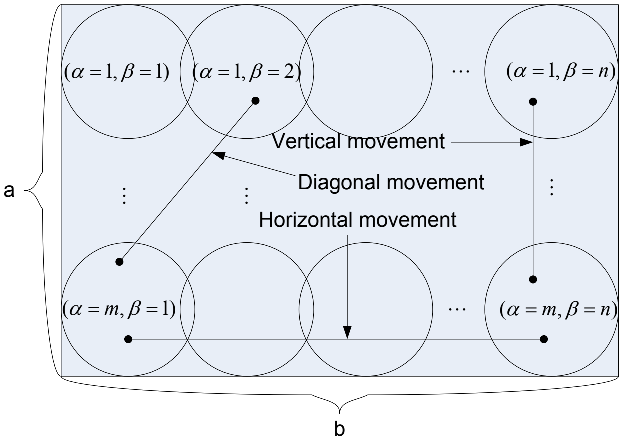


Fig. 9. The rectangular topology considered in the paper.

$(\log N/2+4)$ processing by the MAGs in the Chord system. If an MN performs p handovers between MAGs per second, the total number of handovers would be Mp per second. By multiplying the above two numbers, a Chord system needs to process $(\log N/2+4)Mp$ messages per second. Without loss of generality, we assume that these messages are processed by all MAGs that have the same processing capability. Therefore, a single MAG needs to process $(\log N/2+4)Mp/N$ messages. However, the processing capability of a MAG is bounded by C. That is, it is required that $(\log N/2+4)Mp/N \leq C$. By rewriting this inequality and letting $N=2^s$, we have

$$M \leq \frac{C*2^{s+1}}{p*(s+8)}. \tag{1}$$

From the above two aspects, the total number of MNs that a SARP domain can support would be

$$M \leq \min(\frac{C*2^{s+1}}{p*(s+8)},\ 300{,}000*2^s). \tag{2}$$

Since many messages are signed by MAGs using their private keys, MAGs need to create and verify signatures. It is reported that a 3 GHz processor can create and verify 2048-bit signatures in 150 and 100 microseconds [28], respectively, by using fast crypto-systems such as ESIGN [29]. This means that a 3 GHz processor can create and verify about 6,600 and 10,000 signatures per second, respectively. Furthermore, with the fast development of multi-core technologies, it is anticipated that, in the future, a MAG is able to process 10,000 messages per second (*i.e.*, C = 10,000).

Figure 8 shows the number of MNs that a SARP domain can support when p and s varies. Notice that, as will be explained later in this section, p is set to be moderate to large in Figure 8. From this figure, one can clearly observe that a SARP domain is able to support about 10^7 MNs if p = 0.01 and s = 6 (which corresponds to 64 MAGs). If we keep s unchanged and increase p, the number of MNs that a SARP domain can support is reduced. When p increases to 0.1, the number of MNs reduces to about 10^6. When s = 10 (which corresponds to 1024 MAGs), the number of MNs that a SARP domain can support ranges from 10^8 (for p = 0.01) to about 10^7 (for p = 0.1). If we further increase s to 14, *i.e.*, M = 2^{14} = 16384, the number of MNs that a SARP domain can support ranges from more than 10^9 (for p = 0.01) to more than 10^8 (for p = 0.1). These numbers imply that, if a SARP domain comprises of 1024 MAGs, it can support more than 10^7 MNs when p = 0.1.

With the above considerations, we can observe from Figure 8 that a single SARP domain is able to support $\times 10^8$ MNs when N = 16384 (*i.e.*, s = 14) but p = 0.1. This implies that, if every person in Beijing has five MNs, a single SARP domain is able to cover the whole Beijing city since the number of citizens in Beijing is less than twenty millions. Considering the fact that only a few cities have bigger populations than Beijing, almost all cities around the world can be covered by a single SARP domain. In addition, we can further increase the number of MNs that a SARP domain can support by placing more MAGs in the domain.

3) Estimation of p: Above we have analyzed the feasibility of SARP by letting p range from 0.01 to 0.1. In this section, we present the reason for this choice. Notice that p is the average number of handovers between MAGs that an MN entails per second. This is generally a lower handover rate compared to the rate of handovers between APs since many APs are attached to a MAG and the handovers between APs attached to the same MAG do not lead to exchange of PBU, PA and query messages. For this purpose, we analyze the average residence time of an MN within the coverage area of each MAG. In particular, we consider the random waypoint mobility model [30], which is the most frequently used model in mobile networking research. In this model, an MN randomly selects a destination point (waypoint) in the area of interest according to a uniform distribution, and moves at a constant speed on a straight line to this point. Here the constant speed is uniformly selected between (V_{min}, V_{max}). After waiting a pause time, it chooses a new destination and speed, moves at constant speed to this destination, and so on. This way, a movement is called a *transition*, and the elapsed time and the moved distance during a transition are called transition time denoted by T and transition length denoted by L, respectively. At a destination, the MN stays stationary for a period of time, which is assumed to be zero in the paper. After that, a new transition starts.

We need to calculate the average transition time (denoted by $E[T]$) and the average number of crossings between MAG coverage areas (denoted by $E[C]$) during a transition, since

$$p = E[C]/E[T] \tag{3}$$

To calculate $E[C]$ and $E[T]$, we assume that a single SARP domain covers a rectangular area with length b meters and width a meters, as illustrated in Figure 9. In addition, we assume that the rectangular area is covered by m rows and n columns of MAGs and the coverage area of each MAG is represented by a circle with a radius of R. Under the random waypoint model, the average transition length $E[L]$ is given by [30]

$$\begin{aligned} E(L) = & \frac{1}{15}[\frac{a^3}{b^2}+\frac{b^3}{a^2}+\sqrt{a^2+b^2}(3-\frac{b^2}{a^2}-\frac{a^2}{b^2})] \\ & +\frac{1}{6}[\frac{b^2}{a}\Phi(\frac{\sqrt{a^2+b^2}}{b})+\frac{a^2}{b}\Phi(\frac{\sqrt{a^2+b^2}}{a})] \end{aligned} \tag{4}$$

where $\Phi(x) = \ln(x+\sqrt{x^2-1})$.

Since the transition length L and the movement speed v are independent in the random waypoint model, the average transition time $E[T]$ can be calculated as:

$$E(T) = E(L/v) = E(L)E(1/v). \tag{5}$$

In addition, since the moving speed v is uniformly distributed between (V_{min}, V_{max}), we have

$$E(1/v) = \int_{V_{min}}^{V_{max}} \frac{1}{v} \times \frac{1}{V_{max}-V_{min}} dv = \frac{\ln(V_{max}/V_{min})}{V_{max}-V_{min}}. \tag{6}$$

Therefore, we can obtain the average transition time $E[T]$ by replacing $E[L]$ and $E[1/v]$ in (5) using (4) and (6), respectively.

In order to calculate $E[C]$, we consider three kinds of movements: horizontal, vertical, and diagonal, as shown in Figure 9. For a movement from the coverage area of a MAG $(\alpha_i,\ \beta_i)$ to that of another MAG $(\alpha_j,\ \beta_j)$, the number of MAG crossings $c(\alpha_i,\ \beta_i,\ \alpha_j,\ \beta_j)$ is given by the Manhattan distance between the MAGs [30],

$$c(\alpha_i,\ \beta_i,\ \alpha_j,\ \beta_j) = |\alpha_i-\alpha_j|+|\beta_i-\beta_j|. \tag{7}$$

In addition, the average number of MAG crossings $E[C]$ is computed by the average of $c(\alpha_i,\ \beta_i,\ \alpha_j,\ \beta_j)$ over all possible MAG pairs. That is,

$$E[C] = \frac{1}{m^2n^2}\sum_{\alpha_i=1}^{m}\sum_{\beta_i}^{n}\sum_{\alpha_j=1}^{m}\sum_{\beta_j=1}^{n} c(\alpha_i,\ \beta_i,\ \alpha_j,\ \beta_j). \tag{8}$$

Substituting $c(\alpha_i,\ \beta_i,\ \alpha_j,\ \beta_j)$ using (7), we have

$$E[C] = \frac{1}{m^2n^2}\sum_{\alpha_i=1}^{m}\sum_{\beta_i}^{n}\sum_{\alpha_j=1}^{m}\sum_{\beta_j=1}^{n} (|\alpha_i-\alpha_j|+|\beta_i-\beta_j|). \tag{9}$$

Notice that, given a, b, and R, we have

$$m = (a-L_0)/(2R-L_0), and\ n = (b-L_0)(2R-L_0), \tag{10}$$

where L_0 is the overlapping distance between the coverage areas of two neighboring MAGs. From (3), (5), (9) and (10), we obtain p.

In Figure 10, we plot for different R values the average value of p by varying V_{max} when V_{min} = 1, L_0 = 20 meters, a = 80,000 meters, and b = 60,000 meters. From Figure 10, one can clearly observe that p is 0.094 even if R = 100 meters and V_{max} = 50 meters per second (m/s), which is a very high speed. For example, the allowed highest driving speed at Beijing is limited to about 120 kilometers per hour (about 33.4 m/s). In addition, as can be observed from Figure 10, p decreases significantly if the radius of the area covered by a MAG increases. For example, when R = 200 meters, p reduces to be about 0.045 even if the maximum speed is 50 m/s. Furthermore, p is less than 0.03 when R is larger than 200 meters and V_{max} is less than 30 m/s.

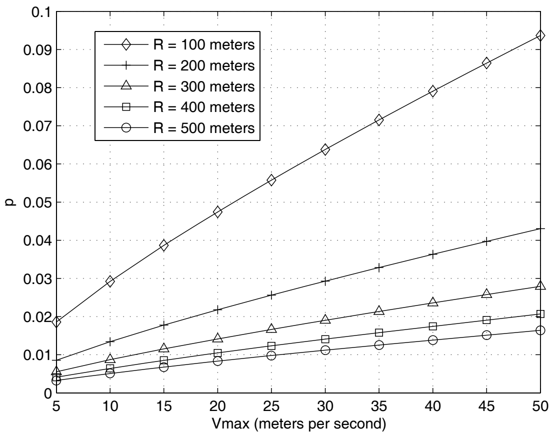


Fig. 10. The average value of p for different R when V_{min} = 1, L_0 = 20 meters, a = 80,000 meters, and b = 60,000 meters.

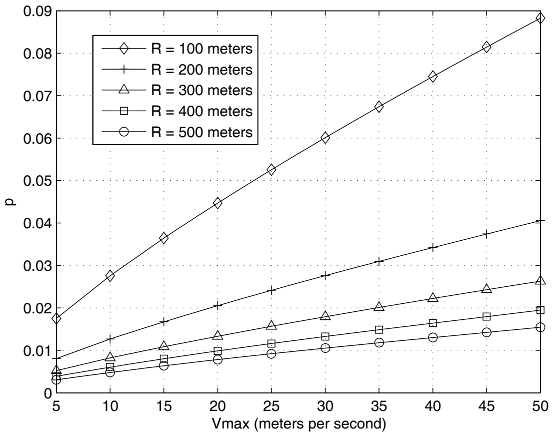


Fig. 11. The average value of p for different R when V_{min} = 1, L_0 = 20 meters, a = 30,000 meters, and b = 10,000 meters.

In Figure 11, we further plot for different R the average value of p by varying V_{max} when V_{min} = 1, L_0 = 20 meters, a = 30,000 meters, and b = 10,000 meters. From this figure, we obtain similar results to those from Figure 10. Notice that the curves for other values are very similar to those shown in Figure 10 and Figure 11, and we do not show them due to space limitation. In summary, these results indicate that it is reasonable to set p ranging from 0.01 to 0.1 when we analyze SARP's feasibility. Furthermore, these results also show that the average residence time that an MN resides at the area covered by a MAG is at least 10 seconds.

4) Bandwidth Overhead: In SARP, a MAG may need to send query messages to other MAGs in order to know an

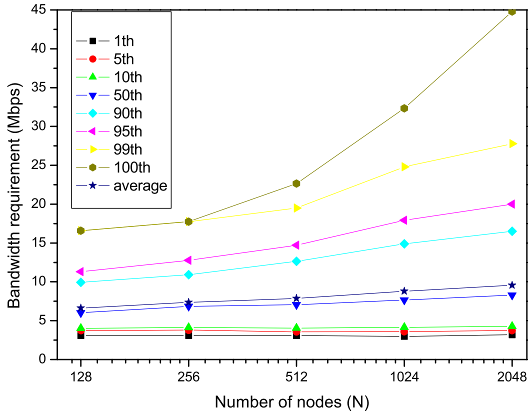


Fig. 12. Bandwidth requirement used for transmitting query messages.

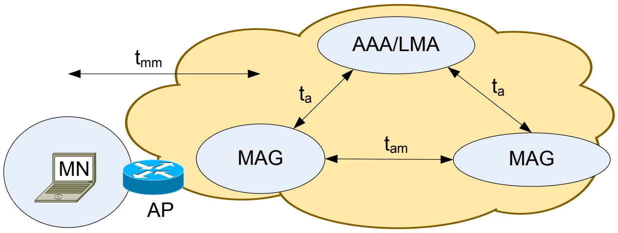


Fig. 13. An analytical model for handover latency analysis.

MN's LMA. Below we analyze the bandwidth overhead used for transmitting query messages and other messages needed to maintain the Chord system. For this purpose, we build a discrete-event packet level simulator using C++ and run simulations over five networks with 128, 256, 512, 1024, and 2048 nodes, respectively. In our simulations, a MAG sends a notification message to its predecessor in order to maintain the Chord system. In addition, we assume that a node sends out a query message with a randomly generated MN-identifier per millisecond, which corresponds to the case that there are 10,000 MNs attached to a MAG and 10% of the MNs roam to other MAGs on average. With this assumption, it implies that there are more than 10^7 MNs in a SARP domain if N = 1024. In order to evaluate the bandwidth requirement, we assume that the size of a query message is 100 bytes (including the 16-byte IPv6 address of the MAG sending the query message, the MN-identifier, a signature, and other packet overhead). In addition, we count, for each node, the amount of incoming traffic per second.

Figure 12 shows the percentile and average bandwidth requirements. The statistics presented in this figure are based on ten different simulations. In order to produce the percentile bandwidth requirements, we record the incoming traffic for every node per second and arrange the records in an increasing order. As a result, an $x-th$ percentile bandwidth requirement in Fig. 12 represents the bandwidth requirement that is larger than the bandwidth requirements of the first x% of the records. From this figure, we observe that the bandwidth requirement increases with the number of nodes in the Chord system, because a query message is forwarded over a larger number of hops in larger networks. While the highest bandwidth requirement (*i.e.*, 100th percentile) increases rapidly with the number of nodes, the average bandwidth requirement increases very slowly. More importantly, we observe that the highest bandwidth requirement for a 2048-node network is less than 45 megabits per second (Mbps). Notice that in modern networks, even an Ethernet link is able to provide a bandwidth of one Gbps. Thus the highest bandwidth requirement in a 2048-node network is only 4.5% of a one Gbps link. With the fast development of wavelength-division multiplexing technology, a link can provide significantly more bandwidth (e.g., one terabit per second), which further reduces the percentage of bandwidth used for carrying querying messages and maintaining Chord.

Notice that we use a large average handover rate in the above simulations. If the average handover rate is reduced, the required bandwidth is also reduced accordingly. Furthermore, fast handover for SARP does not send query messages into the Chord system since a new MAG knows an MN'S LMA from the previous MAG. As a result, the extra bandwidth used for transmitting querying messages is negligible.

B. Handover Latency

In this section, we compare the handover latency of the two proposed handover procedures by comparing them with basic handover for standard PMIPv6 [9] and fast handover for PMIPv6 [14]. Since the handover latency is affected by many factors that are difficult to model, we use the approach like the one used in [1] and [3] to analyze it with the help of the analytical model [1] shown in Figure 13. The notations below are also borrowed from [1].

- The average delay between the MN and the MAG is t_{mm}, which is the time required for a packet to be sent between the MN and the MAG.
- The average delay between the MAG and the LMA is t_{am}. Without loss of generality, we assume that the delay between two MAGs in a PMIPv6 domain is also t_{am}.
- The average delay between the MAG and the AAA is t_a.

For basic handovers in standard PMIPv6, the analysis in [1] shows that the average handover delay is

$$D_{standard} = 4t_a + 2t_{am} + t_{mm}. \qquad (11)$$

For fast handovers in standard PMIPv6, the analysis in [14] shows that the average handover delay is

$$D'_{fast} = 2t_{am} + t_{mm}. \qquad (12)$$

For fast handovers in SARP, the average handover latency (denoted by D_{fast}) comprises the following parts (please refer to Figure 6): the average packet transmission delay from the MAG to the MN (*i.e.*, t_{mm}), and the average delay between nMAG and pMAG (*i.e.*, $2t_{am}$). That is,

$$D_{fast} = 2t_{am} + t_{mm}. \qquad (13)$$

Notice that we do not consider the delay between the MN's nMAG and the MN's LMA when we calculate D_{fast}. This is because the MN's LMA can send packets to the MN's pMAG, which tunnels packets to the MN's nMAG so that the MN can receive packets from the nMAG.

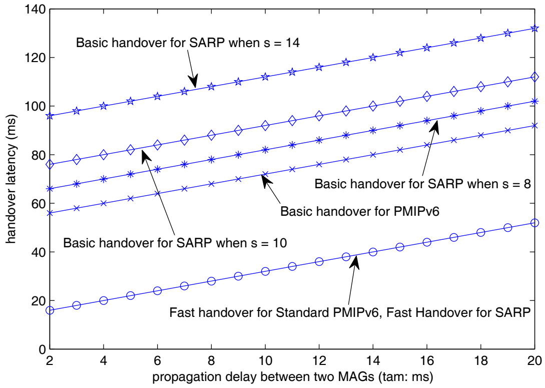


Fig. 14. Comparison of handover latency for the four handover approaches when t_{mm} = 12 ms and t_a = 10 ms.

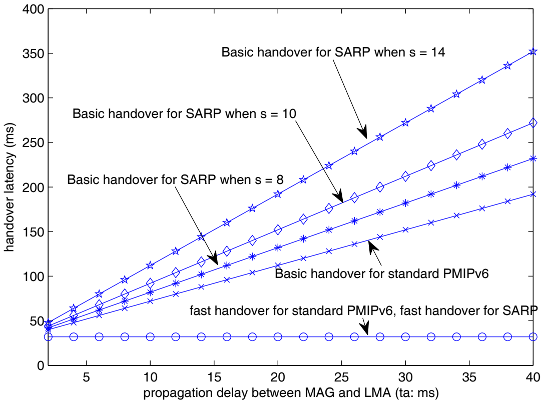


Fig. 15. Comparison of handover latency for the four handover approaches when t_{mm} = 12 ms and t_{am} = 10 ms.

For basic handovers in SARP (please refer to Figure 5), the average handover latency (denoted by D_{basic}) comprises: the average delay from the MAG to the MN (*i.e.*, t_{mm}), the average delay for sending a query message from the MAG to the QServer (T_{query}), the average delay between the QServer and the MAG (denoted by t_{qm}), and the average round-trip delay between the MAG and the LMA (*i.e.*, $2t_{am}$). Since the QServer and the MAG may not connect directly, we assume that $t_{qm} = t_a$. In addition, T_{query} depends on the number of hops (denoted by h) that a query message traversed in the Chord system and the average delay (denoted by t_{aa}) of each hop in the Chord system. Like t_{qm}, we assume that t_{aa} equals to t_a. We notice that the assumptions for t_{aa} and t_{qm} are reasonable since t_a is the average delay from a MAG to an AAA server that is not directly connected. From [19], we know that h belongs to $[0, \log N]$ (*i.e.*, $h \in [0, s]$) and its average is $s/2$. As a result, D_{basic} is given by

$$\begin{aligned} D_{basic} &= T_{query} + t_{qm} + 2t_{am} + t_{mm} = h \times t_a + t_a + 2t_{am} + t_{mm} \\ &= (h+1)t_a + 2t_{am} + t_{mm}. \end{aligned} \quad (14)$$

In Figure 14, we compare the handover latency of the above mentioned four cases by varying t_{am} when t_{mm} = 12 ms and t_a = 10 ms. For basic handover in SARP, we only plot the average handover latency (*i.e.*, $h = s/2$). From Figure 14, we clearly observe that the average handover latency of fast handover in standard PMIPv6 and that of fast handover in SARP are the lowest, because they depend only on the average round-trip delay of two neighboring MAGs. In addition, when compared with the average handover latencies of the basic handover approaches, the handover latencies of the fast handover approaches are significantly lower.

In Figure 15, we further compare the average handover latency of the four cases by varying t_a when t_{mm} = 12 ms and t_{am} = 10 ms. From this figure, we observe that the handover latencies of the basic handover approaches increase with t_a. In particular, when t_a is large (*e.g.*, 40 ms), the handover latencies of the basic handover approaches are too long for real-time applications that typically require a handover latency lower than 150 ms. On the other hand, fast handover in SARP has a very low average handover delay. When t_a varies from 2 ms to 40 ms, like the average handover delay of fast handover in standard PMIPv6, the average handover delay of fast handover in SARP stays unchanged, since it does not depend on t_a. As a result, with the fast handover approach, SARP has very low handover latency.

C. Comparison of SARP with Standard PMIPv6

In this section, we compare SARP with existing work. We first notice that, to our knowledge, there is no prior work on the scalability of PMIPv6. While peer-to-peer for Session Initiation Protocol (P2PSIP) also uses Chord to organize SIP proxies, the performance measure of P2PSIP and SARP are different. In P2PSIP, the main performance measure is the session setup delay, which is closely related to the query delay in the Chord system [33],[34]. On the other hand, in SARP, the main focus is handover delay. With the fast handover approach, as shown above, the handover delay in SARP does not depend on the query delay in the Chord system. As a result, we do not compare SARP with P2PSIP. Instead, we compare SARP with standard PMIPv6. We have shown above that, with SARP, a PMIPv6 domain is able to cover a metropolitan area with more than 10^8 MNs. Thus when an MN roams within such an area, there is no handover between PMIPv6 domains since only one PMIPv6 domain is able to cover such an area. However, for standard PMIPv6, we first notice that it is impossible for a standard PMIPv6 domain to cover the same area and support the same number of MNs. Thus we show below that using many standard PMIPv6 domains to cover the same area leads to inter-PMIPv6 domain handovers whose average delay is significantly longer than the average handover delay in a SARP domain.

For comparison, we also consider the random waypoint model and the rectangular service area used in the above feasibility analysis for SARP. In addition, we also assume that an LMA in standard PMIPv6 is able to deal with traffic for Q MNs and that there are S MNs per square kilometers (*i.e.*, km^2). Therefore, given a rectangular area with length a meters and width b meters, the total number of MNs is

$$M = \frac{a}{1000} \times \frac{b}{1000} \times S = 10^{-6} \times a \times b \times S. \quad (15)$$

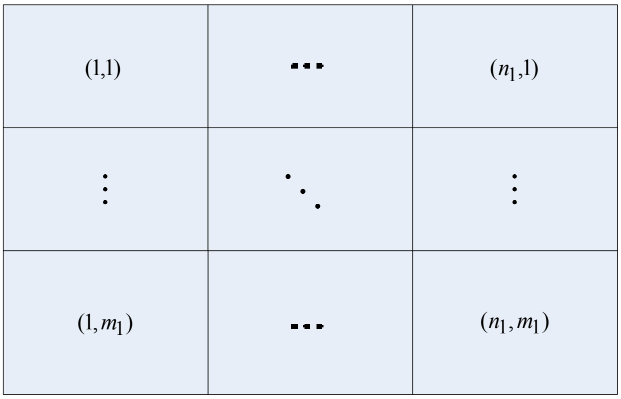


Fig. 16. Layout of PMIPv6 domains.

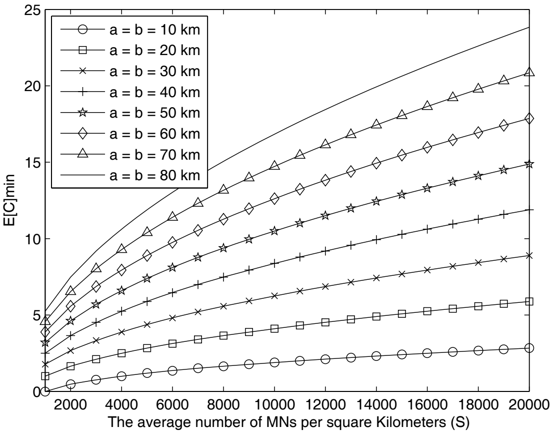


Fig. 17. The average number of handovers per transition when a standard PMIPv6 LMA is able to deal with 100,000 MNs.

Assuming that every PMIPv6 domain maintains only one LMA, the total number of PMIPv6 domains required to cover the rectangular topology is

$$N_D = M/Q = \frac{a \times b \times S}{10^6 \times Q} \tag{16}$$

Without loss of generality, we assume that the coverage areas of the PMIPv6 domains have a layout as illustrated in Figure 16 so that there are m_1 and n_1 vertical and horizontal domains, respectively. With this layout, the average number of handovers among PMIPv6 domains during a single transition is [30]

$$E[C] = \frac{1}{3}(m_1 + n_1 - \frac{1}{m_1} - \frac{1}{n_1}) \tag{17}$$

Since $m_1 \times n_1 = N_D$, it is evident that we can get the minimum value of $E[C]$ when $m_1 = n_1 = \sqrt{N_D}$. That is,

$$E[C]_{min} = \frac{1}{3}(\sqrt{N_D} + \sqrt{N_D} - \frac{1}{\sqrt{N_D}} - \frac{1}{\sqrt{N_D}})$$

$$= \frac{2}{3}(\sqrt{N_D} - \frac{1}{\sqrt{N_D}}) = \frac{2}{3}(\sqrt{\frac{a \times b \times S}{10^6 \times Q}} - \sqrt{\frac{10^6 \times Q}{a \times b \times S}}) \tag{18}$$

In Figure 17, we plot $E[C]_{min}$ for Q = 100,000 by varying a, b, and S. From this figure, we can clearly observe that the average number of handovers per movement is larger than zero, except in the case that $a = b = 10km$ and $S = 1000$. With the increase of S (*i.e.*, the density of MNs), the average number of handovers per movement also increases. Similarly, when the network service area becomes larger, the average number of handovers per movement increases. These observations indicate that standard PMIPv6 cannot scale well when the service area increases, or when the density of MNs increases. As can be seen from the figure, the average number of handovers between PMIPv6 domains increases to more than 5 if we simply increase the service area from 100 km^2 to 6400 km^2 while keeping S unchanged. Similarly, if we keep the service area unchanged, the average number of handovers between PMIPv6 domains increases from 0 to about 3. By comparison, a single SARP domain can cover the whole area of 6400 km^2 even if S = 20,000 since in this case, the total number of MNs is only 1.28×10^8, which is far below the limit of a single SARP domain.

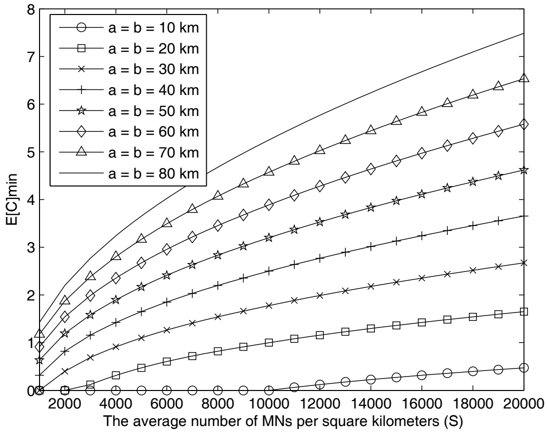


Fig. 18. The average number of handovers per transition when a standard PMIPv6 LMA is able to deal with 1,000,000 MNs.

While one may argue that an LMA can support more than 100,000 MNs, we also increase S to 1,000,000 and plot the corresponding $E[C]_{min}$, as shown in Figure 18. From this figure, we can also observe that the average number of handovers between PMIPv6 domains is larger than zero in most cases, although it is reduced when compared with the case S = 100,000.

Notice that in the above analysis we set moderate to large values for the number of MNs that a single LMA supports. In order to demonstrate this, we use the concept of oversubscription rate, which is defined as follows: if a given router can support N_p MNs simultaneously at a given peak access rate A_p but $N_a \geq N_p$ users are connected, the oversubscription rate is $(N_a - N_p)/N_p$. For example, an oversubscription rate of 1.0 means that only half of the users can simultaneously access the Internet at full speed, or all users at half speed. In the past when the Internet was used mainly for website browsing, Internet service providers often heavily oversubscribe the network with oversubscription rates of 24. However, as the Internet is increasing used for multimedia applications such as streaming videos, video conferencing and other peer-to-peer applications, the demand is much greater and is also relatively more constant [31]. Therefore, we assume that the

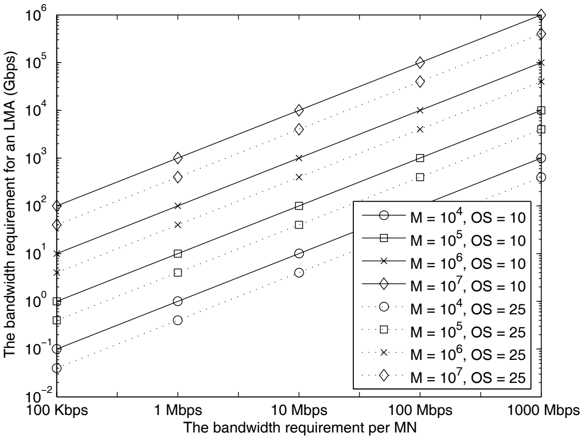


Fig. 19. The bandwidth requirement of an LMA.

oversubscription rates are 10 and 25, respectively, in our analysis.

Given an oversubscription rate (OS), the bandwidth requirement of an LMA is $M \times A_p/OS$. Figure 19 shows the bandwidth requirement of an LMA as a function of M and A_p when the oversubscription rates are 10 and 25 respectively. From this figure, we observe that when the bandwidth requirement per MN is low, a single LMA is able to attach more than 10^7 MNs since the bandwidth requirement is less than 100 Gbps. When the bandwidth requirement per MN increases to 100 Mbps, however, the bandwidth requirement of an LMA is about 10^4 Gbps (= 10 Tbps) if the number of MNs attaching to the LMA is 10^6 and the oversubscription rate is 10. Notice that only a few types of high-end routers can provide a throughput larger than 10 Tbps. Furthermore, it is unknown whether these routers is able to provide such a throughput when the number of routing entries is about 10^6 since in PMIPv6, an LMA needs to maintain an entry for every MN.

Some others may also argue that a standard PMIPv6 domain may maintain multiple LMAs instead of only one. However, the performance of this approach is the same as the case in which the average number of MNs per square kilometer is divided by the number of LMAs; i.e., the case with a significantly reduced number of MNs per square kilometers in a standard PMIPv6 domain with a single LMA. This is why we varies the number of MNs per square kilometers over a wide range in the above analysis. Despite this, as have been shown in Figure 17 - Figure 18, the average number of handovers between PMIPv6 domains is more than zero and, in many cases, larger than one even if the number of MNs per square kilometer is only 1000.

V. Conclusions

In this paper, we have addressed the problem of building scalable, secure, and robust PMIPv6 domains. In the proposed SARP domains, every MAG also functions as an LMA. Based on a well-known distributed hash table protocol, *i.e.*, Chord, we have proposed to organize all MAGs in a SARP domain into a Chord system. The binding between an MN and its LMA are distributed over the Chord system by using consistent hashing. In addition, every SARP domain maintains a PKI server that issues public/private key pairs for all MAGs in the domain. All handover-related messages in the SARP domain are signed so that security is guaranteed. We have proposed two handover procedures and analyzed their handover latencies. We have also analyzed the feasibility of the proposed approach. We have presented results to show that a single SARP domain is able to support more than 10^8 MNs while achieving a low average handover latency that is comparable to that achieved by a fast handover in PMIPv6. While we use the random waypoint model in our analysis, we are interested in investigating the actual mobility behavior of mobile users by using data collected from a cellular network in China.

Acknowledgment

The authors would like to thank Prof. Roy Campbell and the anonymous reviewers for their invaluable comments, which have helped to improve the paper.

References

[1] K.-S. Kong, W. Lee, Y.-H. Han, M.-K. Shin, and H. You, “Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6," *IEEE Wireless Commun.*, vol. 15, no. 2, Apr. 2008.

[2] R. Li, J. Li, K. Wu, Y. Xiao, and J. Xie, “An enhanced fast handover with low latency for mobile IPv6," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 334-342, Jan. 2008.

[3] Q. B. Mussabbir, W. Yao, Z. Niu, and X. Fu, “Optimized FMIPv6 using IEEE 802.21 MIH services in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3397-3407, Nov. 2007.

[4] C. E. Perkins, “IP mobility support," IETF RFC 3220, Jan. 2002.

[5] C. E. Perkins, “Mobile IP," *IEEE Commun. Mag.*, vol. 35, no. 5, pp. 84-99, May 1997.

[6] D. Johnson, C. Perkins, and J. Arkko, “Mobility support in IPv6," IETF RFC 3773, June 2004.

[7] R. Koodli, “Fast handovers for mobile IPv6," IETF RFC 4068, July 2005.

[8] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, “Hierarchical mobile IPv6 (HMIPv6) mobility management," IETF RFC 5380, Oct. 2008.

[9] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, “Proxy mobile IPv6," IETF RFC 5213, Aug. 2008.

[10] N. Banerjee, W. Wu, and S. K. Das, “Mobility support in wireless Internet," *IEEE Wireless Commun.*, vol. 10, no. 5, Oct. 2003.

[11] J. Kempf, “Problem statement for network-based localized mobility management (NETLMM)," IETF RFC 4830, Apr. 2007.

[12] J. Korhonen, S. Gundavelli, H. Yokota, and X. Cui, “Runtime LMA assignment support for proxy mobile IPv6," IETF Draft, draft-ietf-netext-redirect-07.txt (work in progress), Mar. 2011.

[13] V. Devarapalli, R. Koodli, H. Lim, N. Kant, S. Krishnan, and J. Laganier, “Heartbeat mechanism for proxy mobile IPv6," IETF RFC 5847, June 2010.

[14] J. Lei and X. Fu, “Evaluating the benefits of introducing PMIPv6 for localized mobility management," in *Proc. IWCMC*, Aug. 2008, pp. 74-80.

[15] G. Giaretta, “Interactions between PMIPv6 and MIPv6: scenarios and related issues," IETF Draft, draft-ietf-netlmm-mip-interactions-07.txt (work in progress), Oct. 2010.

[16] A. Muhanna, M. Khalil, S. Gundavelli, and K. Leung, “GRE key option for proxy mobile IPv6," IETF RFC 5845, June 2010.

[17] R. Wakikawa and S. Gundavelli, “IPv4 support for proxy mobile IPv6," IETF RFC 5844, May 2010.

[18] H.-N. Nguyen and C. Bonnet, “Proxy mobile IPv6 for cluster based heterogeneous wireless mesh networks," in *Proc. IEEE MASS*, Sept. 2008, pp. 617-622.

[19] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17-32, Feb. 2003.

[20] D. R. Karger, E. Lehman, F. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent hashing and random trees: distributed caching protocols for relieving hot spots on theWorldWideWeb," in *Proc. 29th Annu. ACM Symp. Theory Comput.*, May 1997, pp. 654-663.

[21] D. Eastlake and P. Jones, "US secure hash algorithm 1 (SHA1)," IETF RFC 3174, Sep. 2001.

[22] A. Patel, K. Leung, M. Khalil, H. Akhtar, and K. Chowdhury, "Mobile node identifier option for mobile IPv6 (MIPv6)," IETF RFC 4283, Nov. 2005.

[23] L. R. Monnerat and C. L. Amorim, "D1HT: a distributed one hop hash table," in *Proc. 20th IEEE International Parallel & Distributed Processing Symposium*, Apr. 2006.

[24] "Saving money in GSM," *Huawei Technology*, vol. 25, pp. 27-28, Dec. 2007 (in Chinese). Available: http://www.huawei.com/cn/publications/view.do?id=2915&cid=5503&pid=88.

[25] N. Laoutaris, P. Rodriguez, and L. Massoulie, "ECHOS: edge capacity hosting overlays of nano data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 1, pp. 51-54, Jan. 2008.

[26] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, "Push-to-peer video-on-demand system: design and evaluation," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1706-1716, Dec. 2007.

[27] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM*, Aug. 2008, pp. 63-74.

[28] J. Li, M. N. Krohn, D. Mazi'eres, and D. Shasha, "Secure untrusted data repository (SUNDR)," in *Proc. OSDI*, pp. 121-136, Dec. 2004.

[29] T. Okamoto and J. Stern, "Almost uniform density of power residues and the provable security of ESIGN," in *Proc. ASIACRYPT*, vol. 2894 of LNCS, pp. 287-301, Dec. 2003.

[30] C. Bettstetter, H. Hartenstein, and X. Perz-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Netw.*, vol. 10, no. 5, pp. 555-567, Sep. 2004.

[31] J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the Internet," in *Proc. IEEE COIN-ACOFT*, June 2007.

[32] H. Luo, H. Zhang, and V. C. M. Leung, "An approach for scalable proxy mobile IPv6," in *Proc. IEEE MobiWorld*, Jan. 2010.

[33] J. Maenpaa and G. Camarillo, "Analysis of delays in a peer-to-peer session initiation protocol overlay network," in *Proc. IEEE CCNC*, Jan. 2010.

[34] X. Zheng and V. Oleshchuk, "Improvement of Chord overlay for P2PSIP-based communication systems," *International J. Comput. Netw. Commun.*, vol. 1, no. 3, pp. 133-142, Oct. 2009.

Hongbin Luo received his M.S. (with honors) and Ph.D. degrees in Communications and Information Science from University of Electronic Science and Technology of China (UESTC), in June 2004 and March 2007, respectively.

In June 2007, he joined the School of Electronic and Information Engineering, Beijing Jiaotong University, where he is an associate professor. From Sep. 2009 to Sep. 2010, he was a visiting scholar at Purdue University. He is the first author of more than 30 peer-reviewed papers published in leading journals (such as IEEE/ACM TRANSACTIONS ON NETWORKING) and conference proceedings. He served on the technical program committee (TPC) of several international conferences such as IEEE GLOBECOM'08-11, IEEE ICC'08-12. He was a TPC vice-chair of the 2009 IEEE International Conference on Future Information Networks (ICFIN'09) in Beijing. His research interests are in the areas of Internet routing, Internet architecture, and optical networking.

Hongke Zhang received his M.S. and Ph.D. degrees in Electrical and Communication Systems from University of Electronic Science and Technology of China in 1988 and 1992, respectively.

From Sep. 1992 to June 1994, he was a post-doc research associate at Beijing Jiaotong University. In July 1994, he joined the School of Electronic and Information Engineering, Beijing Jiaotong University, where he is a professor. He has published more than 100 research papers in the areas of communications, computer networks and information theory. He is the holder of more than 50 Chinese patents and is the Chief Scientist of a National Basic Research Program of China ("973 Program").

Dr. Zhang is the recipient of various awards such as the Zhan Tianyou Technical Innovation Award and the Mao Yisheng Technical Innovation Award.

Yajuan Qin received her B.S. and M.S. degrees in Electrical Engineering from the University of Electronic Science and Technology of China (formerly known as Chengdu Institute of Radio Engineering) in 1985 and 1988, respectively, and Ph.D. degree in communication engineering from Beijing University of Posts and Telecommunications in 2003.

From January 2002 to April 2002, she was a research associate at CRL, Japan. In 2003, she joined the School of Electronic and Information Engineering, Beijing Jiaotong University, where she is a full professor. She has published more than 30 research papers and is the holder of more than ten patents. Her research interests are in the areas of computer networks and wireless communications.

Victor C. M. Leung (S'75-M'89-SM'97-F'03) received the B.A.Sc. (Hons.) and Ph.D. degrees, both in electrical engineering, from the University of British Columbia (U.B.C.), where he holds the positions of Professor and TELUS Mobility Research Chair. He has co-authored more than 500 technical papers in international journals and conference proceedings in the broad areas of wireless networks and mobile systems. Dr. Leung is a registered professional engineer in the Province of British Columbia, Canada. He is a Fellow of IEEE, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He has served on the editorial boards of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, TRANSACTIONS WIRELESS COMMUNICATIONS and TRANSACTIONS ON VEHICULAR TECHNOLOGY, and is serving on the editorial boards of the IEEE TRANSACTIONS ON COMPUTERS, the *Journal of Communications and Networks*, *Computer Communications*, as well as several other journals. He has guest-edited several journal special issues, and served on the organizing and technical program committees of numerous international conferences.