INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2012)

Distributed mapping management of identifiers and locators in mobile-oriented Internet environment

Moneeb Gohar[1], Heeyoung Jung[2] and Seok-Joo Koh[1,*,†]

[1]*Kyungpook National University, Daegu, Korea*
[2]*Electronics Telecommunications Research Institute, Daejeon, Korea*

SUMMARY

Many schemes have recently been proposed for the separation of identifier (ID) and locator (LOC), which include the Host Identity Protocol, the Identifier-Locator Network Protocol, and the Locator Identifier Separation Protocol. However, all of these schemes were originally designed in fixed network environment, rather than mobile network environment. In particular, these schemes are based on a centralized map server that is used as an anchor point for mobile nodes, and thus intrinsically subject to some limitations in a mobile environment. In this paper, we propose a distributed ID-LOC mapping management scheme in a mobile-oriented Internet environment. In the proposed scheme, we assume that a host has a globally unique and hierarchical Host ID (HID) that contains the information of its home network domain. Each network domain has a distributed map server for distributed management of ID-LOC mappings. For roaming support, each distributed map server maintains its own home HID register and visiting HID register, which are used to keep the mappings of HID and LOCs for mobile nodes in the distributed manner. By performance analysis, it is shown that the proposed distributed scheme can give better performance than the existing centralized schemes in terms of ID-LOC binding update and data delivery costs. Copyright © 2012 John Wiley & Sons, Ltd.

Received 12 October 2011; Revised 3 January 2012; Accepted 22 February 2012

KEY WORDS: mobile-oriented; identifier; locator; mapping; distributed management

1. INTRODUCTION

With the advent of smart phones and various mobile/wireless access networks, the number of mobile Internet users has been rapidly increasing [1–10]. The ever-lasting growth of Internet services and users has introduced the routing scalability problem to the network engineers [11, 12]. The IRTF (Internet Research Task Force) has investigated many proposals to design scalable Internet routing architecture [13], and reached a rough consensus that the separation of the identifier (ID) and locator (LOC) is required for enhancement of routing scalability in the Internet.

Although the ID-LOC separation was devised to address the routing scalability issue, it ironically brings up another scalability issue to the ID-LOC mapping management. In particular, the scalability of ID-LOC mapping management becomes more critical in the mobile network environment, because the aggregation of IDs for mobile hosts may not be achieved because of movement, and more frequent updates of ID-LOC mapping tables may be required. We note that the number of mobile users has been rapidly increasing with the advent of smart phones. This *mobile* trend has enforced us to effectively design the ID-LOC mapping management system on top of the ID-LOC separation principle.

---

[*] Correspondence to: Seok-Joo Koh, School of Computer Science and Engineering, Kyungpook National University.
[†] E-mail: sjkoh@knu.ac.kr

Until now, several schemes have been proposed to address the mobility issue with the ID-LOC separation and mapping management, which include the Host Identity Protocol (HIP) Rendezvous extension [14, 15], the mobility support of the Identifier-Locator Network Protocol (ILNP) [16, 17], and the mobility architecture based on the Locator Identifier Separation Protocol (LISP) [18–20]. It is noted that all of these schemes have inefficiency because they were originally designed for a fixed network environment, rather than a mobile network environment. For instance, these schemes rely on a centralized map server, which results in some limitations in a mobile network environment in terms of scalability and performance. As the number of mobile nodes (MNs) increases, the control overhead of the centralized server will get larger, because the IDs of MNs cannot be aggregated by movement. Moreover, the centralized schemes tend to increase the operational costs for ID-LOC mapping management and the services degradation by a single point of failure of a centralized mapping server [21]. In the mobile-oriented future Internet environment, these problems become much severe, and thus we need to design a scalable and efficient ID-LOC mapping management scheme for a mobile environment.

In this paper, we propose a distributed mapping management scheme for IDs and LOCs to enhance scalability and performance in mobile-oriented Internet environment. In the proposed scheme, it is assumed that a host has a globally unique and hierarchical Host ID (HID) with the information of its home domain. Each network domain has a distributed map server (DMS). To support the roaming case of mobile nodes, each DMS maintains its own home HID register (HHR) and visiting HID register (VHR), which are used to keep the mapping of HID and LOCs for MNs in the distributed manner. The proposed distributed ID-LOC mapping management can reduce the total costs associated with binding update and data delivery cost, and can also mitigate the problem of a single point of failure of a central server to a local network, compared with the existing centralized ID-LOC separation schemes.

The remainder of this paper is organized as follows. In Section 2, we review the existing ID-LOC mapping management schemes in the mobility perspective. In Section 3, we describe the proposed distributed ID-LOC mapping management scheme. Section 4 analyzes and compares the existing and proposed schemes in terms of the total costs for the binding update and data delivery operations. Section 5 concludes this paper.

2. RELATED WORKS

In this section, we review some of the existing schemes for ID-LOC separation that have so far been proposed: HIP [14, 15], ILNP [16, 17], and LISP [18–20]. It is noted that these schemes were originally designed for fixed network environment, but they have recently been extended to support the mobile network environment.

2.1. *Host identity protocol*

In HIP [14], a locator and a host identifier are separated, in which a 128-bit host identity tag (HIT) is used as a host ID, and an Internet Protocol (IP) address of the host is used as a LOC for packet routing in the network. To support the mobile environment, HIP uses the rendezvous server (RVS) [15]. The ID-LOC management operations of HIP in a mobile environment are illustrated in Figures 1 and 2.

In HIP, an RVS is used for LOC update operation as indicated in Figure 1. When MN is connected to an access router (AR), it configures its LOC. Then, MN sends a *HIP Update* message to RVS for HIT-LOC binding update (Step 1). The RVS will register the HIT-LOC mapping and respond with a *HIP Update ACK* message to MN (Step 2). After handover, MN is connected to a new AR and it configures its LOC (Step 3). Then, MN sends a *HIP Update* message to RVS for HIT-LOC binding update (Step 4). The RVS will update its database and responds with a *HIP Update ACK* message to MN (Step 5). Then, MN sends a *Binding Update* message to the correspondent node (CN) to update its cache (Step 6). After CN updates its cache, it responds with a *Binding Update ACK* to MN (Step 7).

DOI: 10.1002/dac

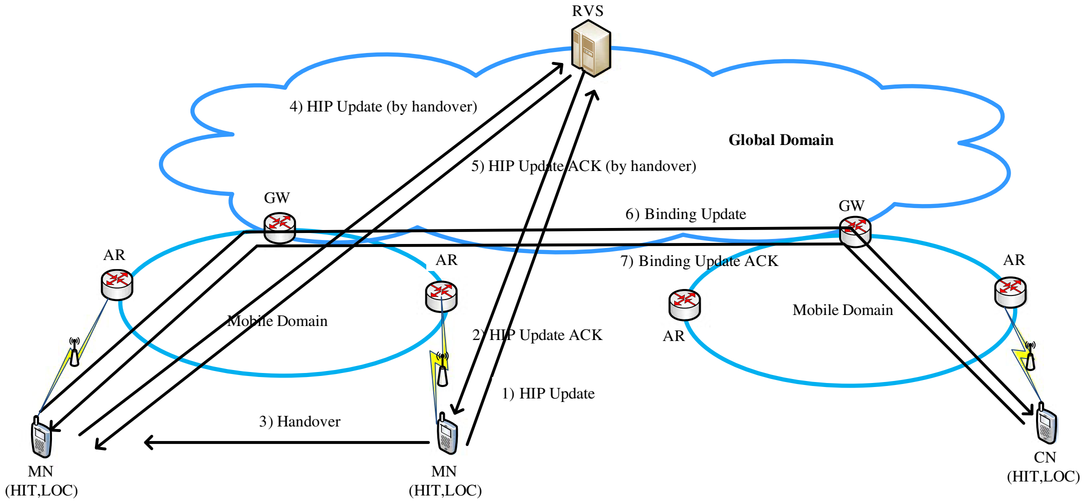

Figure 1. Binding update operation of HIP in a mobile environment [15].

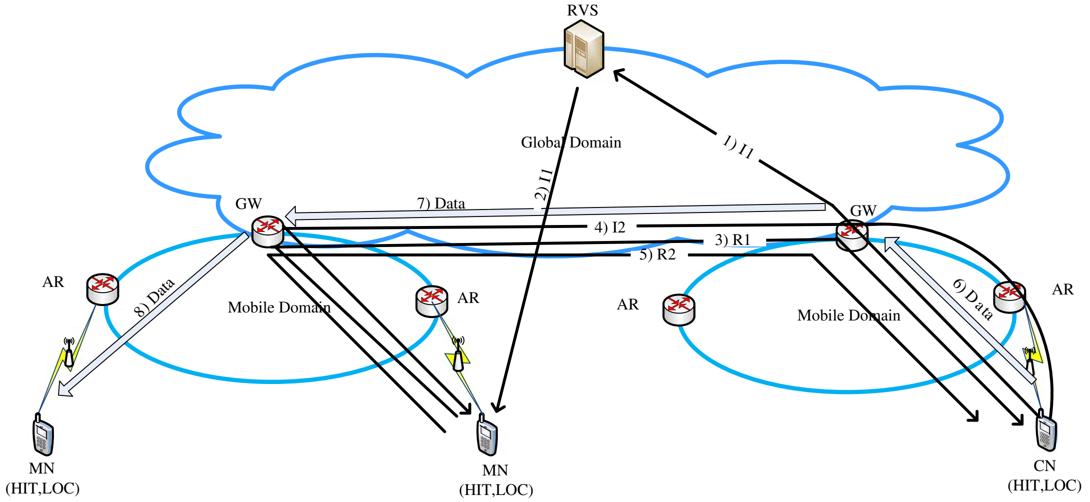

Figure 2. Data delivery operation of HIP in mobile environment [15].

In the data delivery, as indicated in Figure 2, CN sends a data packet to MN that is located in a different domain from CN. To do this, CN will initiate the HIP four-way handshaking operations with MN for connection setup. The first $I1$ packet is sent to RVS (Step 1), and RVS will forward the $I1$ packet to MN (Step 2). After receiving the $I1$ packet, MN responds with an $R1$ message directly to CN (Step 3). The other two messages, $I2$ and $R2$, are exchanged between CN and MN for completion of security association (Steps 4 and 5). Now, CN can send the data packets directly to MN (Steps 6, 7 and 8).

2.2. Identifier-locator network protocol

The ILNP [16, 17] was also proposed for ID-LOC separation, which is based on the address rewriting scheme, in which a 128-bit IPv6 address is divided into the upper 64 bits for LOC and the lower 64 bits for ID. The dynamic domain name system (DDNS) server is used for mapping between ID and LOC for mobile hosts. The ID-LOC mapping management operations of ILNP in mobile environment are illustrated in Figures 3 and 4.

In the LOC update operation, as indicated in Figure 3, when MN is connected to AR, it configures its LOC. Then, MN sends a *LOC Binding Update* message to DDNS for ID-LOC binding update

DOI: 10.1002/dac

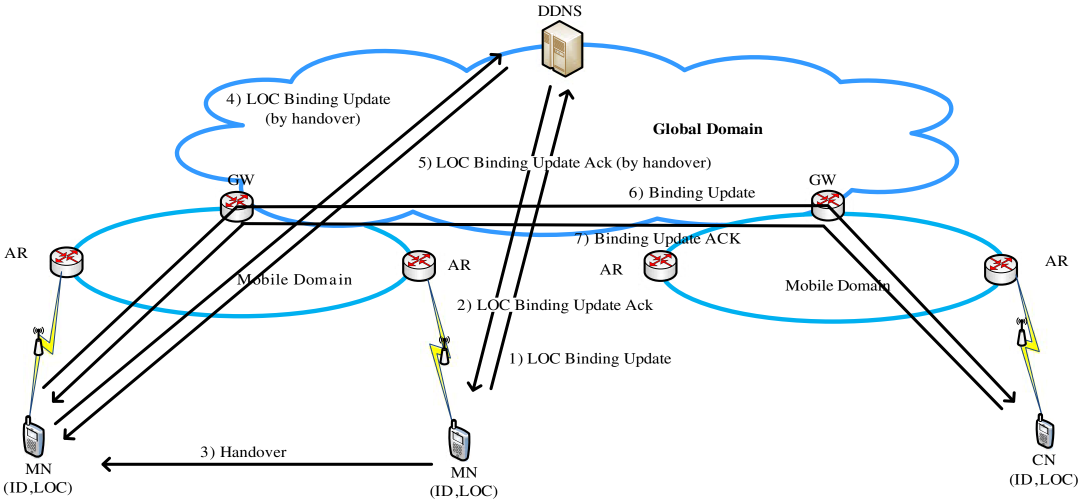

Figure 3. Binding update operations of ILNP in mobile environment [17].

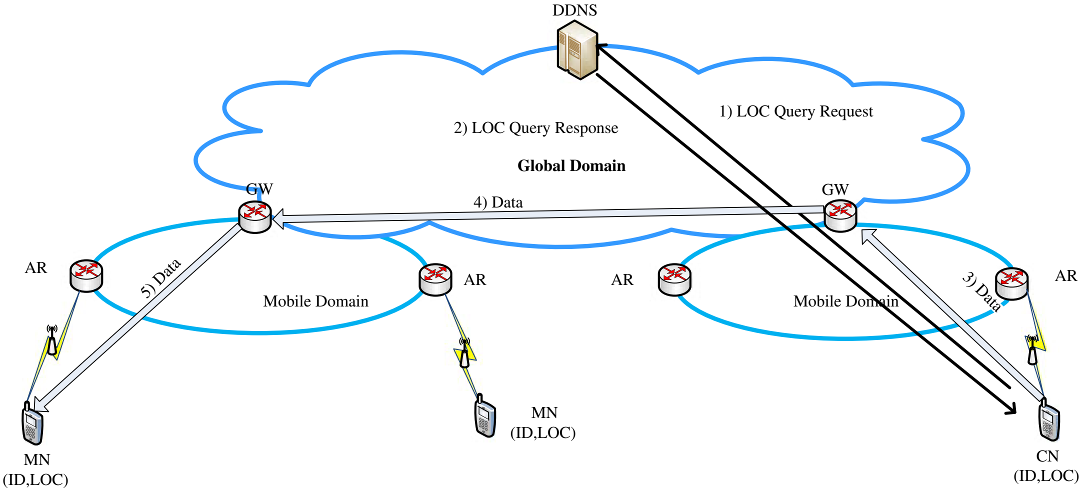

Figure 4. Data delivery operations of ILNP in mobile environment [17].

(Step 1). This DDNS will register the ID-LOC mapping for MN and respond with a *LOC Binding Update ACK* message to MN (Step 2). After handover to another AR (Step 3), MN is connected to the AR and configures its LOC. Then, MN will send a *LOC Binding Update* message to DDNS for ID-LOC binding update (Step 4). The DDNS will update its database and respond with *LOC Binding Update ACK* message to MN (Step 5). Then, MN sends a *Binding Update* message to CN to update its cache. After CN updates its cache, it responds with a *Binding Update ACK* to MN (Steps 6 and 7).

In the data delivery operation of ILNP, as indicated in Figure 4, CN sends a data packet to MN. CN will first send a *LOC Query Request* to DDNS to find the LOC of MN (Step1). Then, DDNS will respond with a *LOC Query Response,* after lookup of its ID-LOC database (Step 2). Now, CN sends the data packets directly to MN (Steps 3, 4 and 5).

2.3. *Locator identifier separation protocol–mobile node–GLAB*

The LISP has recently been proposed in the IETF (Internet Engineering Task Force) [18], which splits the current IP address space into endpoint identifier (EID) and routing locator (RLOC). To support the mobility, the LISP is extended to the LISP-MN architecture in [19], in which it is

DOI: 10.1002/dac

assumed that an MN implements the light-weight tunnel router functionality. In this architecture, a map server (MS) [22, 23] is used as an anchor point for MNs. That is, an MN will maintain the map cache and directly communicate with the MS.

It is noted that LISP-MN depends on a centralized MS for LOC binding update, which may incur significant overhead of control messages at MS in the global scale. To deal with this problem, the work in [20] proposed an enhanced scheme for LISP-MN, which is denoted by LISP-MN-GLAB (German-Lab) in this paper. In this scheme, the main idea is the same with LISP-MN. However, a local MS (LMS) is employed at the gateway of the mobile network to provide a localized mobility control. The mobility control operations of LISP-MN-GLAB are described in Figures 5 and 6.

In the LISP-MN-GLAB scheme, a global MS is used for interdomain communication, and an LMS is employed to support intradomain communication. LMS may possibly be located with the gateway (GW) of the mobile domain. It is assumed that a GW has its RLOCs and each MN uses its local LOC (LLOC) in the local domain, which will be configured with a dynamic IP address configuration scheme.

In the map register (or binding update) operation, as indicated in Figure 5, when MN is connected to AR, it configures its LLOC. Then, MN will send a *Map Register* message to LMS for binding

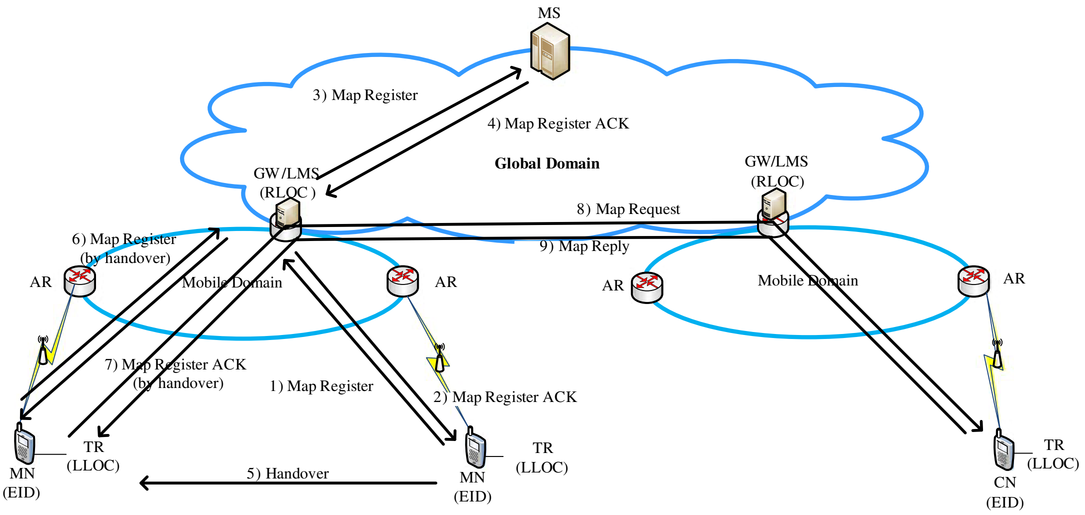

Figure 5. Binding update (map register) operations of LISP-MN-GLAB [20].

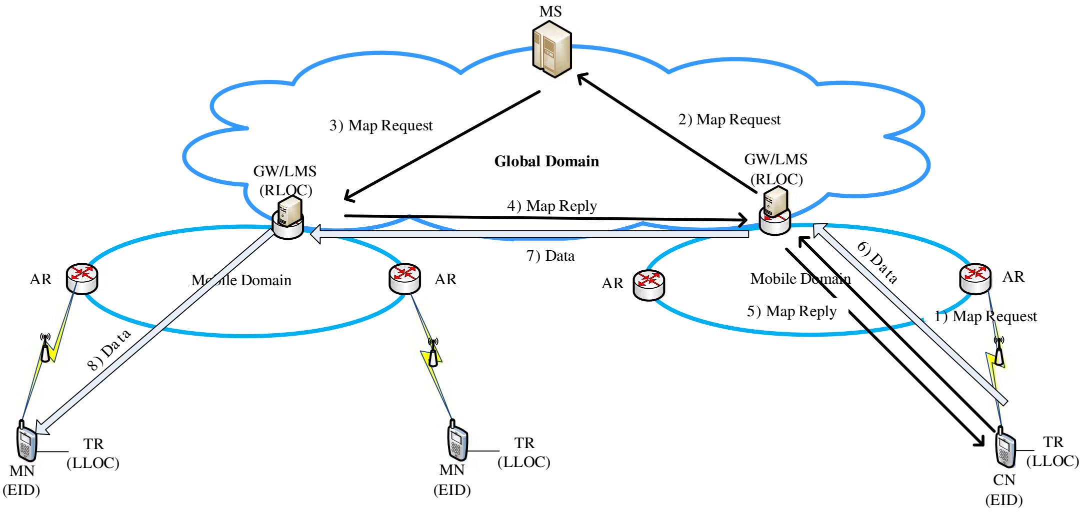

Figure 6. Data delivery operations of LISP-MN-GLAB [20].

DOI: 10.1002/dac

update (Step 1). Then, LMS will register the EID-LLOC mapping for MN and respond with a *Map Register ACK* message to MN (Step 2). In addition, LMS will also exchange the *Map Register* and *Map Register ACK* messages with the global MS to register the mapping between EID of MN and RLOC of LMS (Steps 3 and 4). After handover to another AR, the MN is connected to the AR (Step 5) and it configures its LOC. Then, MN will send a *Map Register* message to LMS for EID-LLOC binding update (Step 6). LMS will update its database and responds with a *Map Register ACK* message to MN (Step 7). Then, MN sends a *Map Request* message to CN to update its cache. When CN updates its cache, it responds with a *Map Reply* to MN (Steps 8 and 9).

In the data delivery operation, as indicated in Figure 6, CN sends a data packet to MN. CN will first send a *Map Request* message to LMS to find the LLOC of MN (Step 1). If there is no mapping information of MN, LMS sends a *Map Request* message to MS to find the RLOC of MN (Step 2). The MS will forward the *Map Request* message to LMS of MN (Step 3). Then, LMS of MN will directly respond to LMS of CN with a *Map Reply* message after lookup of its database (Step 4). In turn, LMS will respond to CN with a *Map Reply* message (Step 5). Now, CN can send the data packet to MN via LMS of MN (Steps 6, 7 and 8).

Until now, we have reviewed the existing schemes for ID-LOC separation and mapping management: HIP, ILNP, and LISP-MN-GLAB. All of these schemes were initially designed for fixed network environment, and they have been extended to support the mobile network environment. However, they are still based on a *centralized* map server. This results in some limitations in mobile network environment in terms of scalability and performance. Accordingly, in this paper, we propose a new distributed scheme for ID-LOC mapping management to provide a scalable ID-LOC mapping management in mobile-oriented Internet environment.

3. PROPOSED SCHEME

In this section, we describe a new distributed ID-LOC mapping management scheme that can effectively support the mobile environment, which has been studied as part of a project sponsored by the Korean government [24].

3.1. Overview

In the proposed scheme, we consider a hierarchical 128-bit HID structure that contains the information of a home network domain that the host is subscribed to, such as the autonomous system (AS) number of a domain. It is noted that the AS number of a domain is unique in the global domain, and the service provider may assign a unique subscriber ID to each of its subscribers within its domain. Accordingly, with the combination of the unique AS number and the unique subscribe ID, an HID will be unique in the global domain. Note that this hierarchical HID has also been discussed in the Routing Architecture for Next Generation Internet, as described in [13]. This type of HID is helpful for interdomain communication, because the network can easily determine whether MN is in the same domain with CN or not.

Figure 7 shows a network model for distributed ID-LOC mapping management, in which each mobile domain is interconnected with the other mobile domains via global Internet domain. For interdomain mapping management, the GW of a mobile domain is used to provide an interface with the global Internet domain. Each GW is connected to its own DMS via an internal interface. To support the roaming case, each DMS maintains its HHR and VHR.

Before going into the detailed description of the proposed scheme, let us compare the existing and proposed mapping management schemes in the architectural perspective, as described in Table I.

In HIP, both ID and LOC are assigned to a host. A 128-bit HIT is used as ID, and an IP address of the host is used as LOC. HIP uses a centralized and global RVS for ID-LOC mapping management. The LOC binding update operation is performed between hosts and RVS.

In ILNP, the lower 64 bits of the IPv6 address is used as an ID and the upper 64 bits of the IPv6 address is used as a LOC. A centralized and global DDNS is used for ID-LOC mapping management. The LOC binding and query operations are performed between hosts and DDNS.

LISP-MN-GLAB (Locator Identifier Separation Protocol-Mobile Node-GermanLab) uses the two types of LOCs: RLOC for network (e.g., GW) and LLOC for host. In LISP-MN-GLAB, each LMS

DOI: 10.1002/dac

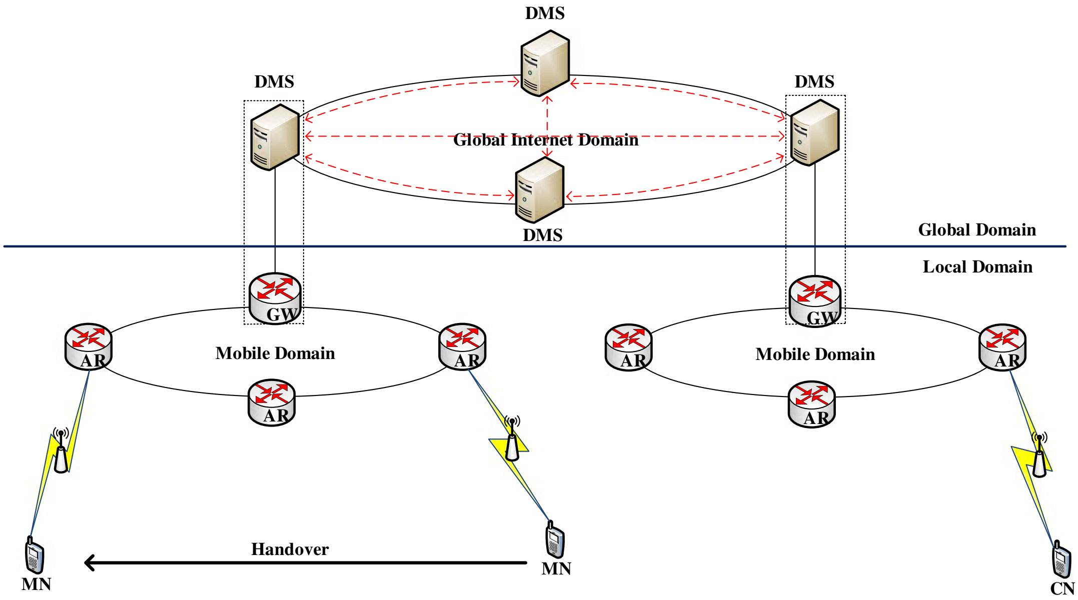

Figure 7. Network model for distributed ID-LOC mapping management.

Table I. Comparison of the existing and proposed schemes.

Scheme	ID	LOC	Mapping architecture	Mapping server
HIP	HIT (128 bits)	IPaddress	Centralized (global)	RVS
ILNP	IPv6 address (lower 64 bits)	IPv6 address (upper 64 bits)	Centralized (global)	DDNS
LISP-MN-GLAB	EID (IP address)	RLOC:network, LLOC:host (IP address)	Centralized (global/local)	MS, LMS
DMS (proposed)	HID (128 bits)	LLOC:network, GLOC:network (IP address)	Distributed (local)	DMS

performs the localized management of EID-LLOC mappings for hosts within a mobile domain, and the global MS is used for global management of EID-RLOC mappings.

In the proposed DMS scheme, a 128-bit HID is used as an ID for the host. For LOC, the IP address of AR is used as LLOC and the IP address of GW is used as Global LOC (GLOC). Each mobile domain has a DMS that performs the local HID-LLOC mapping management. DMS is also used for global HID-GLOC mapping management across different domains. The detailed operations of mapping management will be described in the subsequent sections.

3.2. Host ID-locator binding operations

It is assumed that a host is given by the service provider (of its home domain) on a subscription basis. When a host is attached to the visited network that is located in the other domain, it will establish the network connection with the concerned AR. With this network attachment, the initial HID-LOC binding operations are performed, as shown in Figure 8.

First, a host sends a *HID Binding Request* message to the connected AR. Then, AR constructs or updates its local HID cache (LHC), which contains the list of HIDs for all of the attached hosts. An example format of LHC is given in Table II.

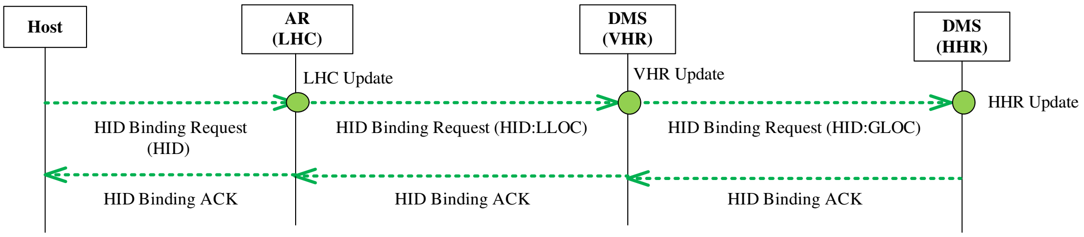

Figure 8. Initial HID-LOC binding operations.

Table II. Local HID cache.

No.	HID	Link status information	Type
1	HID1	—	Home
2	HID2	—	Visited
3	HID3	—	Home
—	—	—	—

In the table, the 'Link Status Information' field may include the link-layer information, such as Media Access Control address, and the timer associated with expiration of the cache entry. In addition, the 'Type' field indicates that the host is now in the *home* or *visited* domain. The LHC information will be referred to by the AR to deliver the data packets that are destined to the local hosts in the data delivery operation.

The AR will check whether a HID belongs to its domain (nonroaming case) or not (roaming case). Note that AR can determine this, based on the HID, because a HID contains the information of the home domain. In the nonroaming case, AR will deliver the *HID Binding Request* to the DMS with HHR. Then, DMS will respond with a *HID Binding ACK* message to AR and further to the host. However, in the roaming case, AR will send a *HID Binding Request* message to the visited DMS in the domain. On receipt of this message, the visited DMS will update its VHR, which maintains the list of HID-LLOC mappings for the visited hosts in the domain. An example format of VHR is given in Table III.

In the roaming case, after the VHR update, the visited DMS sends a *HID Binding Request* message to the home DMS that is in the home domain of the roaming host. The home DMS will update its HHR, which maintains the list of HID-GLOC mapping for the visited host, as shown in Table IV. After the HHR update, the home DMS responds with a *HID Binding ACK* message to the visited DMS, which will further be delivered to the roaming host, as described in Figure 6. The detailed formats of *HID Binding Request* and *ACK* messages are for further study.

Table III. VHR database.

No.	HID	LLOC	Others
1	HID1	LLOC1	Related data
2	HID2	LLOC2	Related data
3	—	—	—

Table IV. Home HID Register (HHR) database.

No.	HID	GLOC	Others
1	HID1	GLOC1	Related data
2	HID2	GLOC2	Related data
3	—	—	—

DOI: 10.1002/dac

After handover to another AR, MN will establish a network connection with the concerned AR. With this network attachment, the HID-LOC binding operations are performed, as shown in Figure 9. In the figure, MN sends a *HID Binding Request* to ARnew. After ARnew updates its LHC, it sends a *HID Binding Request* message to the visited DMS (V-DMS). When V-DMS updates its VHR table, V-DMS will respond with a *HID Binding ACK* to ARnew. Then, ARnew delivers this *HID Binding ACK* to MN.

3.3. Data delivery operations

Now, let us assume that MN is attached to the network, and the HID binding operation was completed between host and AR, and also between AR and DMS. Figure 10 shows the LOC query and the packet delivery operations in the proposed DMS scheme.

When CN sends a data packet to MN, it first sends a data packet to AR. AR then sends a *LOC Query Request* to its DMS. Now, DMS will forward this *LOC Query Request* to the home DMS (H-DMS) of MN. By referring to its HHR, the H-DMS will forward the *LOC Query Request* to the V-DMS, where MN stays at that time. Then, V-DMS will respond with a *LOC Query ACK* message to AR of CN. Now, AR of CN can deliver the data packet to MN via V-DMS. The detailed formats of *LOC Query Request* and *ACK* messages are for further study. In the handover case, V-DMS will forward the data packet to a new AR, which will be delivered to MN.

In the LOC query operation, each DMS or AR shall maintain its remote LOC cache (RLC), which is shown in Table V. The RLC contains the mapping of HID-LOC for the remote hosts that are in active communication with a certain local host. It is noted that each RLC will be referred to by DMS and by AR for data forwarding. In the table, LOC represents LLOC in the mobile domain and GLOC in the global domain.

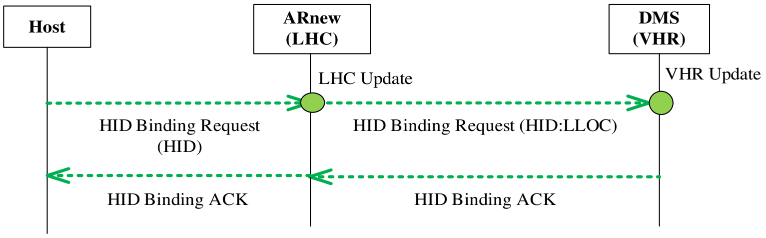

Figure 9. HID-LOC binding operations after handover.

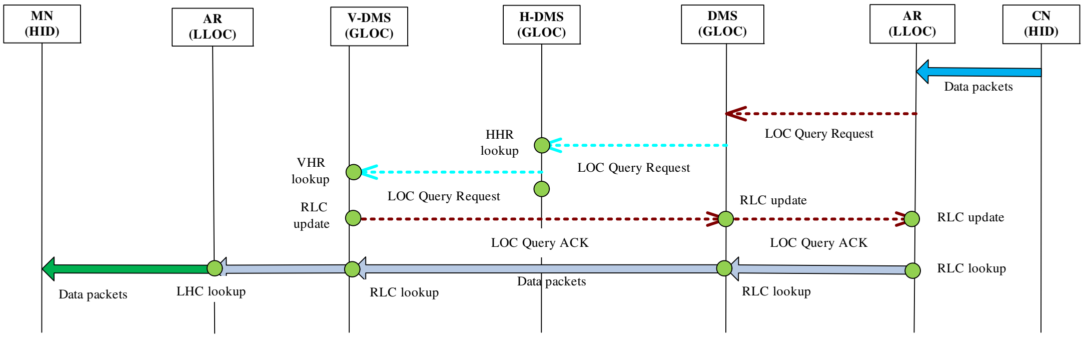

Figure 10. Data delivery operation.

Table V. Remote LOC Cache (RLC).

No.	HID	LOC	Status information
1	HID1	LOC1	—
2	HID2	LOC2	—
3	—	—	—

DOI: 10.1002/dac

4. PERFORMANCE ANALYSIS

To evaluate the performance of the proposed scheme, we analyze the total costs associated with the binding update with ID-LOC mapping server and the data delivery from CN to MN. We will compare the total costs for the existing schemes and the proposed DMS scheme.

4.1. *Analysis model*

We first describe the mobility and network models used for cost analysis. First, for mobility model, the well-known fluid flow model [25] is used, in which it is assumed that the location of MN is uniformly distributed with the range $(0, 2\pi)$. It is noted that the fluid flow model is suitable for mobile users with high mobility and frequent changes of movement direction. Then, the number of subnet crossing in the given mobile domain (λ_s) is as follows.

$$\boldsymbol{\lambda}_{\mathrm{s}} = \left[\mathbf{2} \cdot \frac{\boldsymbol{v}}{\sqrt{\boldsymbol{\pi} \cdot \boldsymbol{A}}} \right]. \tag{1}$$

In the equation, v is the average speed of MN and A is the area of the concerned subnet. It is assumed that all the subnets have the same circular shape and size. For simplicity, we assume that CN and MN are located in different mobile domains, as illustrated in Figure 11.

We define the parameters used for the analysis in Table VI.

4.2. *Cost analysis*

The total cost (TC) is defined as the sum of the binding update cost (BUC) and the data delivery cost (DDC), that is, $TC = BUC + DDC$

4.2.1. *Host identity protocol.*

The BUC of HIP is divided into two parts: the initial binding update cost with RVS and the binding update cost by handover to another AR. Then, the BUC of HIP can be expressed as

$$\boldsymbol{BUC}_{\mathrm{HIP}} = BUC_{\mathrm{RVS}} + \boldsymbol{\lambda}_{\mathrm{s}} \times (BUC_{\mathrm{RVS}} + BUC_{\mathrm{CN}}). \tag{2}$$

In HIP, the initial binding update operation with RVS is performed when MN enters a mobile domain for the first time and it configures its LOC by using the IP address configuration (e.g., DHCP). We assume that this operation takes roughly T_{AC}. After that, MN performs the HIP update operation with RVS by exchanging *HIP Update* and *HIP Update ACK* messages, and RVS will update the database. This operation takes $2T_{\text{MN-GW}} + 2T_{\text{GW-RVS}}$ and $P_{\text{RVS}},$ where $T_{\text{MN-GW}} = \kappa H_{\text{MN-AR}} + \tau H_{\text{AR-GW}}$, $T_{\text{GW-RVS}} = \tau H_{\text{GW-RVS}}$, and $P_{\text{RVS}} = \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}})$.

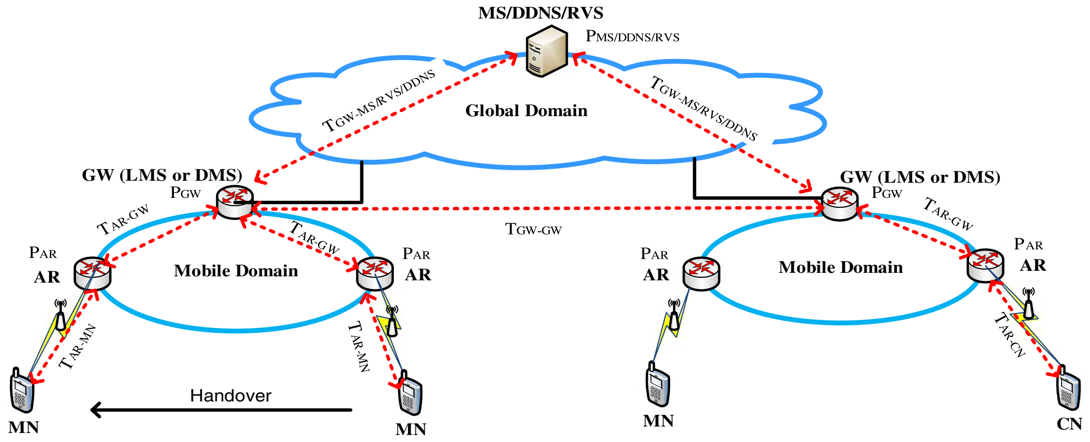

Figure 11. Network model for numerical analysis.

DOI: 10.1002/dac

Table VI. Parameters used for cost analysis.

Parameter	Description
T_{a-b}	Transmission cost of a packet between nodes a and b
P_c	Processing cost of node c for binding update or lookup
$N_{\text{Host/AR}}$	Number of active hosts per AR
$N_{\text{AR/GW}}$	Number of AR in the mobile domain per GW
N_{GW}	Number of GW in the global domain
H_{a-b}	Hop count between nodes a and $b$in the network
S_{Control}	Size of a control packet (in byte)
S_{Data}	Size of a data packet(in byte)
α	Unit cost of binding update with GW and MS/RVS/DDNS
β	Unit cost of lookup for MN at GW or AR and MS/RVS/DDNS
τ	Unit transmission cost of a packet per a wired link (hop)
κ	Unit transmission cost of a packet per a wireless link (hop)
T_{AC}	Address configuration delay
v	Velocity of MN (m/s)
A	Subnet area (m^2)

It is assumed that the processing cost for binding update with RVS (P_{RVS}) is proportional to the total number of active hosts in the domain ($N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}}$) by using a tree-based data structure to implement the database. Accordingly, the BUC with RVS can be represented as follows:

$$\begin{aligned}\boldsymbol{BUC}_{\text{RVS}} &= T_{\text{AC}} + S_{\text{Control}} \times (2T_{\text{MN-GW}} + 2T_{\text{GW-RVS}}) + P_{\text{RVS}} \\ &= T_{\text{AC}} + S_{\text{Control}} \times 2(\kappa H_{\text{MN-AR}} + \tau H_{\text{AR-GW}} + \tau H_{\text{GW-RVS}}) \\ &\quad + \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}})\end{aligned} \tag{3}$$

After handover to another AR, the binding update operation with RVS is performed, as carried out in the initial binding update operation. In addition, the binding update operation is performed between MN and CN to update its caches after the LOC of MN is changed. Therefore, the BUC between MN and CN can be represented as follows:

$$\begin{aligned}\boldsymbol{BUC}_{\text{CN}} &= S_{\text{Control}} \times (2T_{\text{MN-GW}} + 2T_{\text{GW-GW}} + 2T_{\text{CN-GW}}) \\ &= S_{\text{Control}} \times 2(\kappa H_{\text{MN-AR}} + 2\tau H_{\text{AR-GW}} + \tau H_{\text{GW-GW}} + \kappa H_{\text{CN-AR}}).\end{aligned} \tag{4}$$

The DDC of HIP is divided into two parts: the binding query cost (BQC) and the packet delivery cost (PDC). That is, the DDC of HIP can be represented as follows:

$$\boldsymbol{DDC}_{\text{HIP}} = BQC_{\text{HIP}} + PDC_{\text{HIP}}. \tag{5}$$

The BQC_{HIP} can be calculated as follows. First, CN sends $I1$ message to RVS to find the LOC of MN. Then, RVS will look for the LOC of MN in its database, which takes $P_{\text{RVS}} = \beta(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{GW})$. After the lookup, RVS will forward the $I1$ message to MN. After that MN will respond directly to CN with $R1$ message. After receiving the $R1$ message, CN will send $I2$ message to MN, then MN will respond with $R2$ message to CN. This operation takes $4T_{\text{CN-AR}} + 8T_{\text{AR-GW}} + 2T_{\text{GW-RVS}} + 3T_{\text{GW-GW}} + 4T_{\text{MN-AR}}$. Thus, the BQC_{HIP} can be represented as follows:

$$\begin{aligned}\boldsymbol{BQC}_{\text{HIP}} &= S_{\text{Control}} \times (4T_{\text{CN-AR}} + 8T_{\text{AR-GW}} + 2T_{\text{GW-RVS}} + 3T_{\text{GW-GW}} + 4T_{\text{MN-AR}}) + P_{\text{RVS}} \\ &= S_{\text{Control}} \times (4\kappa H_{\text{CN-AR}} + 8\tau H_{\text{AR-GW}} + 2\tau H_{\text{GW-RVS}} + 3\tau H_{\text{GW-GW}} + 4\kappa H_{\text{MN-AR}}) \\ &\quad + \beta(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}}).\end{aligned} \tag{6}$$

After binding query operation the packet delivery operation is performed, then PDC_{HIP} can be represented as follows:

$$\begin{aligned}\boldsymbol{PDC}_{\mathrm{HIP}} &= S_{\mathrm{Data}} \times (T_{CN-AR} + 2T_{\mathrm{AR\text{-}GW}} + T_{\mathrm{GW\text{-}GW}} + T_{\mathrm{MN\text{-}AR}}) \\ &= S_{\mathrm{Data}} \times (\kappa H_{\mathrm{CN\text{-}AR}} + 2\tau H_{\mathrm{AR\text{-}GW}} + \tau H_{\mathrm{GW\text{-}GW}} + \kappa H_{\mathrm{MN\text{-}AR}}).\end{aligned} \tag{7}$$

Therefore, we obtain the total cost of HIP as $TC_{\mathrm{HIP}} = BUC_{\mathrm{HIP}} + DDC_{\mathrm{HIP}}$.

4.2.2. Identifier-locator network protocol. The binding update cost of ILNP is divided into two parts: the initial binding update with DDNS and the binding update cost by handover to another AR. Then, binding update cost (BUC) of ILNP can be expressed as follows:

$$\boldsymbol{BUC}_{\mathrm{ILNP}} = BUC_{\mathrm{DDNS}} + \boldsymbol{\lambda}_{\mathrm{s}} \times (BUC_{\mathrm{DDNS}} + BUC_{\mathrm{CN}}). \tag{8}$$

In ILNP, the initial binding update operation with DDNS is performed, when MN enters a mobile domain and it configures its LOC, which takes T_{AC}. After that, MN performs the LOC update operation with DDNS by exchanging the *LOC Binding Update* and *LOC Binding Update ACK* messages, and DDNS will update its database. This operation takes $2T_{\mathrm{MN\text{-}GW}} + 2T_{\mathrm{GW\text{-}DDNS}}$ and $P_{\mathrm{DDNS},}$ where $T_{\mathrm{MN\text{-}GW}} = \kappa H_{\mathrm{MN\text{-}AR}} + \tau H_{\mathrm{AR\text{-}GW}}$, $T_{\mathrm{GW\text{-}DDNS}} = \tau H_{\mathrm{GW\text{-}RVS}}$, and $P_{\mathrm{DDNS}} = \alpha(N_{\mathrm{AR/GW}} \times N_{\mathrm{Host/AR}} \times N_{\mathrm{GW}})$. It is assumed that the processing cost for binding update with DDNS (P_{DDNS}) is proportional to the total number of active hosts in the domain $(N_{\mathrm{AR/GW}} \times N_{\mathrm{Host/AR}} \times N_{\mathrm{GW}})$ by using a tree-based data structure. Accordingly, the binding update cost with DDNS can be represented as follows:

$$\begin{aligned}\boldsymbol{BUC}_{\mathrm{DDNS}} &= T_{\mathrm{AC}} + S_{\mathrm{Control}} \times (2T_{\mathrm{MN\text{-}GW}} + 2T_{\mathrm{GW\text{-}DDNS}}) + P_{\mathrm{DDNS}} \\ &= T_{\mathrm{AC}} + S_{\mathrm{Control}} \times 2(\kappa H_{\mathrm{MN\text{-}AR}} + \tau H_{\mathrm{AR\text{-}GW}} + \tau H_{\mathrm{GW\text{-}DDNS}}) \\ &\quad + \alpha(N_{\mathrm{AR/GW}} \times N_{\mathrm{Host/AR}} \times N_{\mathrm{GW}}).\end{aligned} \tag{9}$$

After handover to another AR, the binding update operation with DDNS is performed, as carried out in the initial binding update operation. In addition, the binding update operation is performed between MN and CN to update its caches. Therefore, the BUC_{CN} between MN and CN can be represented as follows:

$$\begin{aligned}\boldsymbol{BUC}_{\mathrm{CN}} &= S_{\mathrm{Control}} \times (2T_{\mathrm{MN\text{-}GW}} + 2T_{\mathrm{GW\text{-}GW}} + 2T_{\mathrm{CN\text{-}GW}}) \\ &= S_{\mathrm{Control}} \times 2(\kappa H_{\mathrm{MN\text{-}AR}} + 2\tau H_{\mathrm{AR\text{-}GW}} + \tau H_{\mathrm{GW\text{-}GW}} + \kappa H_{\mathrm{CN\text{-}AR}}).\end{aligned} \tag{10}$$

The DDC of ILNP is divided into two parts: the BQC and the PDC, as follows:

$$\boldsymbol{DDC}_{\mathrm{ILNP}} = BQC_{\mathrm{ILNP}} + PDC_{\mathrm{ILNP}}. \tag{11}$$

In ILNP, the binding query cost can be calculated as follows. First, CN sends a *LOC Query Request* to DDNS to find the LOC of MN. Then, DDNS will look for the LOC of MN in its database, which takes $P_{\mathrm{DDNS}} = \beta(N_{\mathrm{AR/GW}} \times N_{\mathrm{Host/AR}} \times N_{\mathrm{GW}})$. After that, the DDNS will respond with a *LOC Query Response* message to CN. The cost of control message transmission is equal to $2T_{\mathrm{CN\text{-}AR}} + 2T_{\mathrm{AR\text{-}GW}} + 2T_{\mathrm{GW\text{-}DDNS}}$. Thus, the BQC_{ILNP} can be represented as follows:

$$\begin{aligned}\boldsymbol{BQC}_{\mathrm{ILNP}} &= S_{\mathrm{Control}} \times (2T_{\mathrm{CN\text{-}AR}} + 2T_{\mathrm{AR\text{-}GW}} + 2T_{\mathrm{GW\text{-}DDNS}}) + P_{\mathrm{DDNS}} \\ &= S_{\mathrm{Control}} \times 2(\kappa H_{\mathrm{CN\text{-}AR}} + \tau H_{\mathrm{AR\text{-}GW}} + \tau H_{\mathrm{GW\text{-}DDNS}}) + \beta(N_{\mathrm{AR/GW}} \times N_{\mathrm{Host/AR}} \times N_{\mathrm{GW}}).\end{aligned} \tag{12}$$

After the binding query operation the packet delivery operation is performed, then the packet delivery cost can be represented as follows:

$$\begin{aligned}\boldsymbol{PDC}_{\mathrm{ILNP}} &= S_{\mathrm{Data}} \times (T_{\mathrm{CN\text{-}AR}} + 2T_{\mathrm{AR\text{-}GW}} + T_{\mathrm{GW\text{-}GW}} + T_{\mathrm{MN\text{-}AR}}) \\ &= S_{\mathrm{Data}} \times (\kappa H_{\mathrm{CN\text{-}AR}} + 2\tau H_{\mathrm{AR\text{-}GW}} + \tau H_{\mathrm{GW\text{-}GW}} + \kappa H_{\mathrm{MN\text{-}AR}}).\end{aligned} \tag{13}$$

Therefore, we obtain the total cost of ILNP as $TC_{\mathrm{ILNP}} = BUC_{\mathrm{ILNP}} + DDC_{\mathrm{ILNP}}$.

4.2.3. LISP-MN-GLAB. The binding update cost of LISP-MN-GLAB is divided into two parts: the initial binding update with LMS and MS, and the binding update cost by handover to another AR. Then, the BUC of LISP-MN-GLAB can be expressed as follows:

$$\boldsymbol{BUC}_{\text{LISP-MN-GLAB}} = BUC_{\text{LMS}} + BUC_{\text{MS}} + \boldsymbol{\lambda}_{\text{s}} \times (BUC_{\text{LMS}}). \tag{14}$$

In LISP-MN-GLAB, the initial binding update operation with LMS and MS takes T_{AC}. After that, MN will perform the map register operation with GW by exchanging *Map Register* and *Map Register ACK* messages, and GW will update the database. This operation takes $2T_{\text{MN-GW}}$ and P_{GW}, where $T_{\text{MN-GW}} = \kappa H_{\text{MN-AR}} + \tau H_{\text{AR-GW}}$ and $P_{\text{GW}} = \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}})$. After that, GW will perform the map register operation with MS by exchanging *Map Register* and *Map Register ACK* messages, and MS will update the database. This operation takes $2T_{\text{GW-MS}}$ and P_{MS}, where $T_{\text{GW-MS}} = \tau H_{\text{GW-MS}}$ and $P_{\text{MS}} = \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}})$. Accordingly, the binding update cost with LMS and MS can be represented as follows:

$$\begin{aligned}\boldsymbol{BUC}_{\text{LMS}} &= T_{\text{AC}} + S_{\text{Control}} \times (2T_{\text{MN-GW}}) + P_{\text{GW}} \\ &= T_{\text{AC}} + S_{\text{Control}} \times 2(\kappa H_{\text{MN-AR}} + \tau H_{\text{AR-GW}}) + \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}})\end{aligned} \tag{15}$$

$$\begin{aligned}\boldsymbol{BUC}_{\text{MS}} &= S_{\text{Control}} \times (2T_{\text{GW-MS}}) + P_{\text{MS}} \\ &= S_{\text{Control}} \times 2\tau H_{\text{GW-MS}} + \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}})\end{aligned} \tag{16}$$

The DDC of LISP-MN-GLAB is divided into two parts: the BQC and the PDC. Therefore, the DDC of LISP-MN-GLAB can be represented as follows:

$$\boldsymbol{DDC}_{\text{LISP-MN-GLAB}} = BQC_{\text{LISP-MN-GLAB}} + PDC_{\text{LISP-MN-GLAB}}. \tag{17}$$

In LISP-MN-GLAB, the binding query cost can be calculated as follows. First, CN sends a *Map Request* message to GW to find the LLOC of MN. GW looks for the LLOC of MN in its database, which takes $P_{\text{GW}} = \beta(N_{\text{AR/GW}} \times N_{\text{Host/AR}})$. GW will then send a *Map Request* message to MS. The MS will forward *Map Request* message to GW of MN. Then, GW of MN will responds with the *Map Reply* message to CN via GW and AR. Therefore, the associated cost is equal to $2T_{\text{CN-AR}} + 2T_{\text{AR-GW}} + 2T_{\text{GW-MS}} + T_{\text{GW-GW}}$. Thus, the binding query cost can be represented as follows:

$$\begin{aligned}\boldsymbol{BQC}_{\text{LISP-MN-GLAB}} &= S_{\text{Control}} \times (2T_{\text{CN-AR}} + 2T_{\text{AR-GW}} + 2T_{\text{GW-MS}} + T_{\text{GW-GW}}) + P_{\text{GW}} + P_{\text{MS}} \\ &= S_{\text{Control}} \times (2\kappa H_{\text{CN-AR}} + 2\tau H_{\text{AR-GW}} + 2\tau H_{\text{GW-MS}} + \tau H_{\text{GW-GW}}) \\ &\quad + \beta(N_{\text{AR/GW}} \times N_{\text{Host/AR}}) + \beta(N_{\text{AR/GW}} \times N_{\text{Host/AR}} \times N_{\text{GW}})\end{aligned} \tag{18}$$

After binding query operation the packet delivery operation is performed, then the packet delivery cost can be represented as follows:

$$\begin{aligned}\boldsymbol{PDC}_{\text{LISP-MN-GLAB}} &= S_{\text{Data}} \times (T_{\text{CN-AR}} + 2T_{\text{AR-GW}} + T_{\text{GW-GW}} + T_{\text{MN-AR}}) \\ &= S_{\text{Data}} \times (\kappa H_{\text{CN-AR}} + 2\tau H_{\text{AR-GW}} + \tau H_{\text{GW-GW}} + \kappa H_{\text{MN-AR}}).\end{aligned} \tag{19}$$

Therefore, we obtain the total cost as $TC_{\text{LISP-MN-GLAB}} = BUC_{\text{LISP-MN-GLAB}} + DDC_{\text{LISP-MN-GLAB}}$.

4.2.4. Proposed distributed map server. The binding update cost of the proposed DMS scheme is divided into the two parts: the initial binding update with V-DMS and H-DMS and the binding update by handover to another AR, as follows:

$$\boldsymbol{BUC}_{\text{DMS}} = BUC_{\text{V-DMS}} + BUC_{\text{H-DMS}} + \boldsymbol{\lambda}_{\text{s}} \times (BUC_{\text{V-DMS}}). \tag{20}$$

In the proposed DMS scheme, the initial binding update operation is performed with V-DMS and H-DMS. When a host is attached to the network, it will establish a network connection with the concerned AR. With the network attachment of host, the HID binding operation will perform by exchanging *HID Binding Request* and *HID Binding ACK*, this operation takes $2T_{\text{MN-AR}} + P_{\text{AR}}$, where $T_{\text{MN-AR}} = \kappa H_{\text{MN-AR}}$ and $P_{\text{AR}} = \alpha(N_{\text{Host/AR}})$. After that AR will also exchange the *HID Binding Request* and *HID Binding ACK* messages with the visited DMS for HID-LLOC binding. This operation takes $2T_{\text{AR-GW}} + P_{\text{GW}}$, where $T_{\text{AR-GW}} = \tau H_{\text{AR-GW}}$ and $P_{\text{GW}} = \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}})$. The visited DMS will also exchange the *HID Binding Request* and *HID Binding ACK* messages with home DMS for HID-GLOC binding, which takes $2T_{\text{GW-GW}} + P_{\text{GW}}$, where $T_{\text{GW-GW}} = \tau H_{\text{GW-GW}}$ and $P_{\text{GW}} = \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}})$. Accordingly, the binding update cost with V-DMS and H-DMS can be represented as follows:

$$\begin{aligned} \boldsymbol{BUC}_{\text{V-DMS}} &= S_{\text{Control}} \times (2T_{\text{MN-AR}} + 2T_{\text{AR-GW}}) + P_{\text{AR}} + P_{\text{GW}} \\ &= S_{\text{Control}} \times 2\kappa H_{\text{MN-AR}} + 2\tau H_{\text{AR-GW}} + \alpha(N_{\text{host/AR}}) + \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}}) \end{aligned} \tag{21}$$

$$\begin{aligned} \boldsymbol{BUC}_{\text{H-DMS}} &= S_{\text{Control}} \times (2T_{\text{GW-GW}}) + P_{\text{GW}} \\ &= S_{\text{Control}} \times 2\tau H_{\text{GW-GW}} + \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}}) \end{aligned} \tag{22}$$

The data delivery cost of the proposed scheme is divided into the two parts: the BQC and the PDC, as follows:

$$DDC_{\text{DMS}} = BQC_{\text{DMS}} + PDC_{\text{DMS}}. \tag{23}$$

In the proposed DMS scheme, the binding query cost can be calculated as follows. First, a data packet of CN is delivered to AR of CN. The AR checks the information of HID of MN. Then, AR of CN sends a *LOC Query Request* to DMS, which is equal to $T_{\text{AR-GW}}$. After that, GW of CN will forward the *LOC Query Request* to home DMS of MN, which is equal to $T_{\text{GW-GW}}$. Then home DMS of MN sends *LOC Query ACK* message to the visited DMS of MN, which corresponds to $T_{\text{GW-GW}}$. After lookup of its VHR database, the visited DMS updates its RLC and responds to AR of CN via DMS with a *LOC Query ACK* message, after receiving the *LOC Query ACK*, the DMS of CN also updates its RLC and forwards the *LOC Query ACK* message to AR. AR will also update its RLC. This is equal to $4T_{\text{GW-GW}} + 2T_{\text{AR-GW}} + 2P_{\text{GW}} + 2P_{\text{GW}} + P_{\text{AR}}$. Accordingly, the binding query cost of the proposed DMS scheme can be represented as follows:

$$\begin{aligned} \boldsymbol{BQC}_{\text{DMS}} &= S_{\text{Control}} \times (2T_{\text{AR-GW}} + 4T_{\text{GW-GW}}) + 2P_{\text{GW}} + P_{\text{AR}} + 2P_{\text{GW}} \\ &= S_{\text{Control}} \times (2\tau H_{\text{AR-GW}} + 4\tau H_{\text{GW-GW}}) + 2 \times \beta(N_{\text{AR/GW}} \times N_{\text{Host/AR}}) + \alpha(N_{\text{Host/AR}}) \\ &\quad + 2 \times \alpha(N_{\text{AR/GW}} \times N_{\text{Host/AR}}) \end{aligned} \tag{24}$$

After the binding query operation the packet delivery operation is performed, then the packet delivery cost can be represented as follows.

$$\begin{aligned} \boldsymbol{PDC}_{\text{DMS}} &= S_{\text{Data}} \times (T_{\text{CN-AR}} + 2T_{\text{AR-GW}} + T_{\text{GW-GW}} + T_{\text{MN-AR}}) \\ &= S_{\text{Data}} \times (\kappa H_{\text{CN-AR}} + 2\tau H_{\text{AR-GW}} + \tau H_{\text{GW-GW}} + \kappa H_{\text{MN-AR}}). \end{aligned} \tag{25}$$

Therefore, we obtain the total cost of the proposed scheme as $TC_{\text{DMS}} = BUC_{\text{DMS}} + DDC_{\text{DMS}}$.

4.3. Numerical results

On the basis of the cost analysis given in the previous section, we now compare the numerical results. In the analysis, we assume both CN and MN across different mobile domains to simplify the analysis. For numerical analysis, we set the parameter values, as shown in Table VII, which are configured partly based on the results given in [26].

Table VII. Parameter values used for cost analysis.

Parameter	Default	Minimum	Maximum
κ	2	1	10
α	0.1	0.1	1
β	0.2	0.1	1
$N_{\text{Host/AR}}$	100	100	1,000
$N_{\text{AR/GW}}$	10	10	100
N_{GW}	10	10	100
$H_{\text{AR-GW}}$	2	1	50
$H_{\text{GW-MS}}$, $H_{\text{GW-GW}}$	6	1	50
T_{AC}	150	10	400
v	40	20	200
$H_{\text{MN-AR}}$, $H_{\text{CN-AR}}$	1		
τ	1		
S_{Control}	50 (bytes)		
S_{Dat}	1024 (bytes)		
A	800		

Figure 12 compares the total costs of the candidate schemes for different transmission cost over wireless link (κ). It is shown in the figure that the total cost linearly increases for all the schemes, as κ gets larger. However, LISP-MN-GLAB gives better performance than all the existing schemes, because LISP-MN-GLAB uses a local mapping system. For all of the candidate schemes, the proposed DMS scheme gives the best performance. This is because the proposed scheme uses a distributed map server to reduce the traffics required for binding update and data delivery.

Figures 13 and 14 show the impact of unit binding update cost (α) in Figure 13, and the impact of unit lookup cost (β) at MS/LMS/DMS in Figure 14, respectively. From the figures, we can see that the total costs linearly increase, as α and β get larger, for all the candidate schemes. It is shown that the proposed DMS scheme gives better performance than the existing schemes. It is noted that LISP-MN-GLAB gives better performance than ILNP, because LISP-MN-GLAB uses both local MS and global MS, and thus the binding update and query operations will be performed with local MS and global MS.

We show the impact of the number of hosts per AR ($N_{\text{Host/AR}}$) in Figure 15 and the impact of the number of AR ($N_{\text{AR/GW}}$) in the domain in Figure 16. In the figures, we can see that the total cost linearly increases for all the candidate schemes, but the proposed DMS scheme gives better performance from all of the existing schemes. The two existing schemes, HIP and ILNP, provide almost

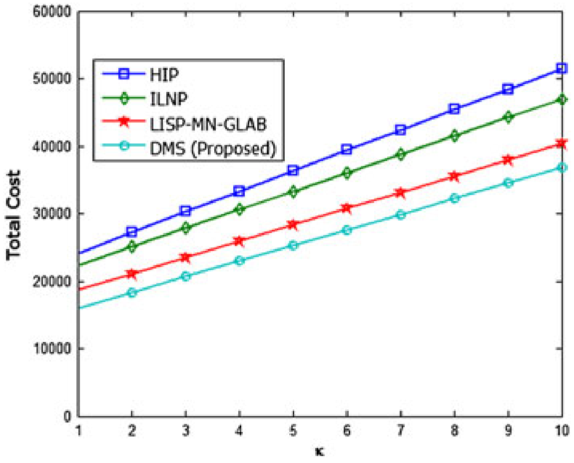

Figure 12. Impact of unit transmission cost over wireless link on total cost.

DOI: 10.1002/dac

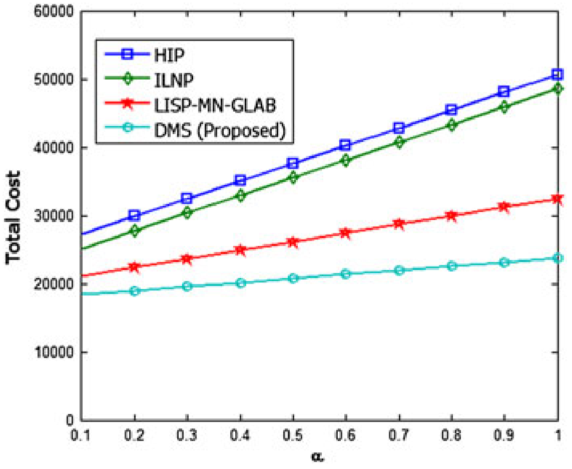

Figure 13. Impact of the unit binding update cost on total cost.

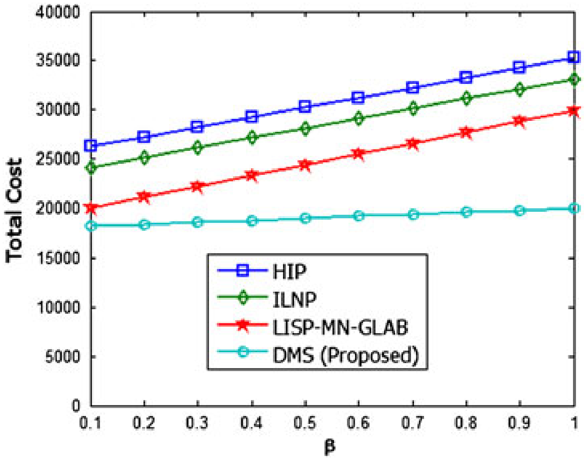

Figure 14. Impact of unit lookup cost on total cost.

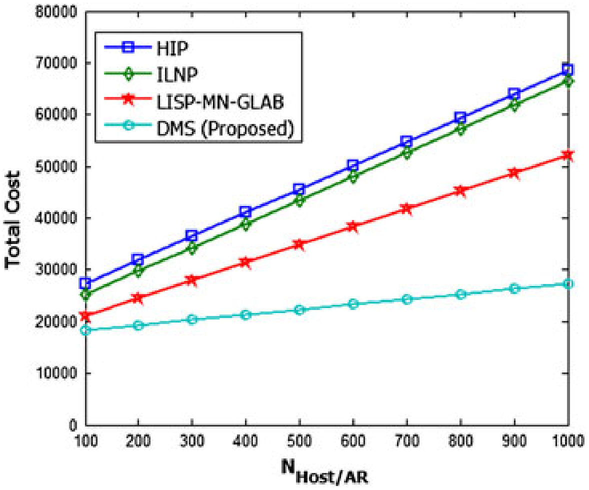

Figure 15. Impact of the number of hosts per AR on total cost.

DOI: 10.1002/dac

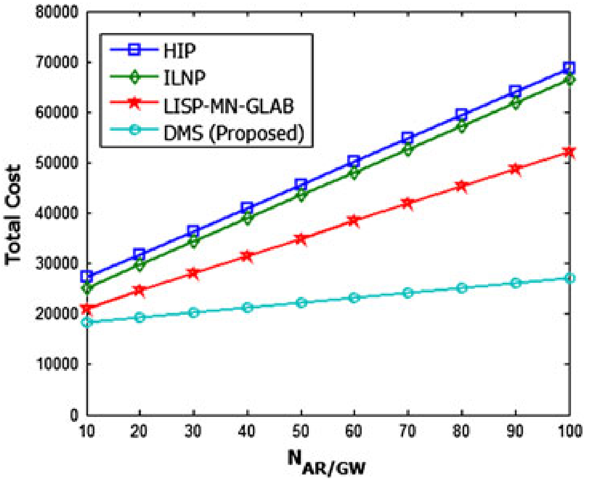

Figure 16. Impact of the number of ARs in the domain on total cost.

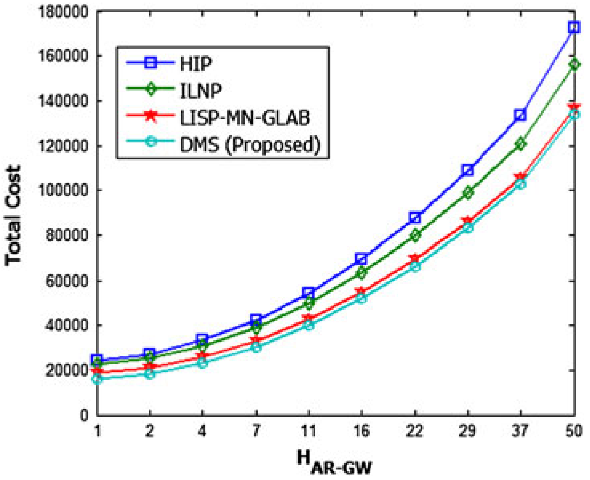

Figure 17. Impact of hop counts between AR and GW on total cost.

the same performance, while LISP-MN-GLAB gives better performance than the HIP and ILNP because LISP-MN-GLAB uses the local and global mapping systems. When MN moves from one AR to another in the same domain, MN will perform the binding operation with only the local mapping system. The proposed DMS scheme gives better performance from all of the existing schemes. This is because the proposed scheme is based on the DMS, and the traffic required for the binding update and query operations will be distributed into the locally distributed DMS servers. In the meantime, the existing schemes are all dependent on the centralized server, and thus the total costs are severely impacted by the number of hosts and ARs in the domain.

Figure 17 shows the impact of the hop counts between AR and GW (H_{AR-GW}). In the figure, we can see that the total cost linearly increases for all the candidate schemes, and that the proposed DMS scheme gives the best performance among the candidate schemes.

Figure 18 compares the candidate schemes in terms of the address configuration delay (T_{AC}). In the figure, we can see that T_{AC} gives a significant impact on the total cost for all of the existing schemes, because all of them are based on a host-based LOC and thus MN should configure its LOC in the network. On the other hand, the proposed DMS scheme is not affected by T_{AC}, because it uses a network-based LOC and thus MN does not need any LOC configuration.

Figure 19 compares the total costs of the candidate schemes for different hop counts between GW and MS ($H_{\text{GW-MS}}$). In the figure, we can see that $\mathrm{H}_{\text{GW-MS}}$ gives significant impacts on total cost

DOI: 10.1002/dac

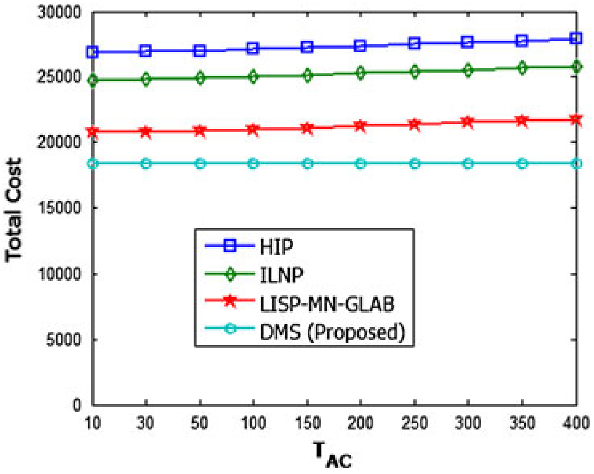

Figure 18. Impact of address configuration delay on total cost.

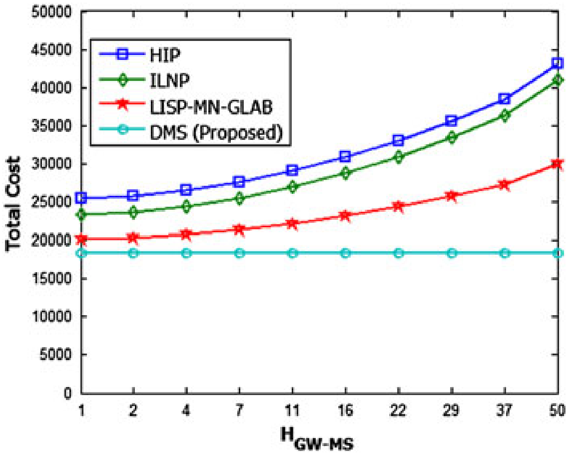

Figure 19. Impact of hop counts between GW and MS on total cost.

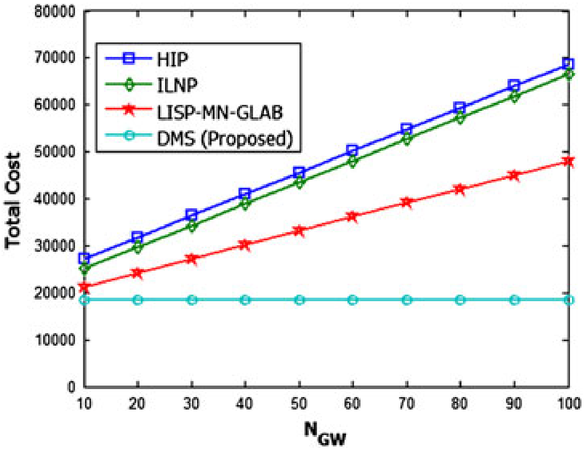

Figure 20. Impact of the number of GWs on total cost.

DOI: 10.1002/dac

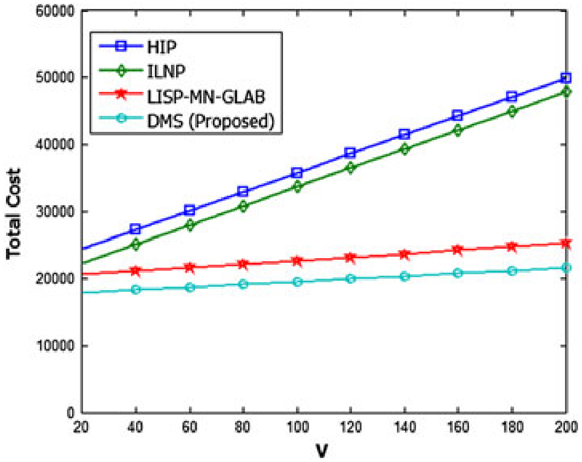

Figure 21. Impact of the velocity of MN on total cost.

for all of the existing schemes. This is because all of the existing schemes depend on a centralized MS in the signaling operations, and the control messages required for signaling should be delivered between GW and the central MS. It is noted that the proposed DMS scheme gives the best performance among all the candidate schemes.

Figure 20 shows the impact of the total number of GWs in the network (N_{GW}). From the figure, we can see that the total cost of all the existing schemes linearly increases, as N_{GW} gets larger. This is because all the existing schemes are dependent on the central MS in the binding update and query operations. On the other hand, we note that the proposed DMS scheme is not affected by N_{GW}, because it uses a distributed MS.

Figure 21 shows the impact of the velocity of MN. In the figure, we can see that the total cost linearly increases for all the candidate schemes, but the proposed DMS scheme gives the best performance among the candidate schemes.

5. CONCLUSION

'Mobile-oriented' is the most envisioned feature of future Internet, and thus it should be properly considered in the design of the ID-LOC separation scheme for future Internet. The recently proposed schemes for ID-LOC separation, such as HIP, ILNP, and LISP, did not consider the mobile environments in the initial design stage. To support mobile environment, those schemes have been extended, but still have some limitations. In particular, those schemes are based on the centralized management of ID-LOC mapping information, and thus tend to incur the large operational costs.

In this paper, we proposed a distributed mapping management scheme of IDs and LOCs in mobile network environments. The proposed scheme is featured by the distributed map server that maintains VHR and HHR for mobility support. By numerical analysis, the proposed scheme is compared with the existing schemes. From numerical results, it is shown that the proposed distributed scheme can give better performance than the existing centralized schemes in terms of the binding update and data delivery costs.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program of NRF (2011-0026529), ITRC program of NIPA (NIPA-2011-C1090-1121-0002), and IT R&D support program of KCA (KCA-2011-10913-05004).

REFERENCES

1. Ho AH *et al.* Handling high mobility in next-generation wireless ad hoc networks. *International Journal of Communication Systems* 2010; **23**(9-10):1078–1092.
2. Koo DJ *et al.* Authenticated route optimization scheme for network mobility (NEMO) support in heterogeneous networks. *International Journal of Communication Systems* 2010; **23**(9-10):1252–1267.

DOI: 10.1002/dac

3. Kim H *et al.* Improving TCP performance for vertical handover in heterogeneous wireless networks. *International Journal of Communication Systems* 2009; **22**(8):1001–1021.
4. Chiang WK *et al.* Mobile-initiated network-executed SIP-based handover in IMS over heterogeneous accesses. *International Journal of Communication Systems* 2010; **23**(9-10):1268–1288.
5. Gözüpek D *et al.* Genetic algorithm-based scheduling in cognitive radio networks under interference temperature constraints. *International Journal of Communication Systems* 2011; **24**(2):239–257.
6. You I *et al.* caTBUA: Context-aware ticket-based binding update authentication protocol for trust-enabled mobile networks. *International Journal of Communication Systems* 2010; **23**(11):1382–1404.
7. Chang C-C *et al.* An efficient e-mail protocol providing perfect forward secrecy for mobile devices. *International Journal of Communication Systems* 2010; **23**(12):1463–1473.
8. Halunga SV *et al.* Performance evaluation for conventional and MMSE multiuser detection algorithms in imperfect reception conditions. *Digital Signal Processing* 2010; **20**(1):166–178.
9. Halunga SV *et al.* Imperfect cross-correlation and amplitude balance effects on conventional multiuser decoder with turbo encoding. *Digital Signal Processing* 2010; **20**(1):191–200.
10. Halunga SV *et al.* Orthogonality, Amplitude and Number of users effects on conventional multiuser detection using turbo decoding. *Proceeding of EUROCON 2009*, Saint-Petersburg, Russia, May 2009; 203–2009.
11. Meyer D *et al.* Report from the IAB workshop on routing and addressing. *IETF RFC 4984*, 2007.
12. Stanley M. Report, Internet trends, 2010. http://www.morganstanley.com/.
13. Li T *et al.* Recommendation for a routing architecture. *IETF RFC 6115*, 2011.
14. Moskowitz R *et al.* Host Identity Protocol. *IETF RFC 5201*, 2008.
15. Laganier J *et al.* Host Identity Protocol (HIP) rendezvous extension. *IETF RFC 5204*, 2008.
16. Atkinson R. ILNP concept of operations, 2011. IETF Internet Draft, draft-rja-ilnp-intro-11.txt.
17. Atkinson R *et al.* ILNP: mobility, multi-Homing, localised addressing and security through naming. *Telecommunication Systems* 2009; **42**(3-4):273–291.
18. Farinacci D *et al.* Locator/ID Separation Protocol (LISP), 2011. IETF Internet Draft, draft-ietf-lisp-15.
19. Farinacci D *et al.* LISP Mobile Node, 2011. IETF Internet Draft, draft-meyer-lisp-mn-05.txt.
20. Menth M *et al.* Improvements to LISP Mobile Node. *Conference of International Teletraffic Congress (ITC)*, Amsterdam, Netherlands, September 2010; 1–8.
21. Chan H *et al.* Problem statement for distributed and dynamic mobility management, 2011. IETF Internet Draft, draft-chan-distributed-mobility-ps-05.
22. Fuller V *et al.* LISP Map Server, 2011. IETF Internet Draft, draft-ietf-lisp-ms-08.txt.
23. Farinacci D *et al.* LISP alternative topology (LISP-ALT), 2011. IETF Internet Draft, draft-fullerlisp-alt-06.txt.
24. Homepage of Mobile Oriented Future Internet (MOFI). http://www.mofi.re.kr.
25. Lee JH *et al.* A Comparative performance analysis on Hierarchical Mobile IPv6 and Proxy Mobile IPv6. *Telecommunication System* 2009; **41**(4):279–292.
26. Jung H *et al.* Distributed mobility control in Proxy Mobile IPv6 networks. *IEICE Transactions on Communications* 2011; **E94-B**(8):2216–2224.

AUTHORS' BIOGRAPHIES

Moneeb Gohar received his B.S. degree in Computer Science from the University of Peshawar, Pakistan, and his M.S. degree in Technology Management from the Institute of Management Sciences, Pakistan, in 2006 and 2009, respectively. He is now a Ph.D. student in the School of Computer Science and Engineering in the Kyungpook National University, Korea. His current research interests include network layer protocols, wireless communication, mobile multicasting, and internet mobility. E-mail: moneebgohar@gmail.com.

Heeyoung Jung joined Electronics and Telecommunication Research Institutes (ETRI) in 1991 after receiving his bachelor's degree from Pusan National University (PNU), and he is currently a principal research member. He received his Ph.D. degree in Information and Communications Engineering from the Chungnam National University (CNU) in 2004. His major research areas include Internet and mobile network technologies and those that are closely related to standardization activities in ITU-T, IETF, and so on. His current research topic is future Internet architecture. E-mail: hyjung@etri.re.kr.

DOI: 10.1002/dac

Seok-Joo Koh received his B.S. and M.S. degrees in Management Science from KAIST in 1992 and 1994, respectively. He also received his Ph.D. degree in Industrial Engineering from KAIST in 1998. From August 1998 to February 2004, he worked for Protocol Engineering Center in ETRI. He has been a professor in the School of Computer Science and Engineering at the Kyungpook National University since March 2004. His current research interests include mobility management in the future Internet, IP mobility, multicasting, and SCTP. He has so far participated in the international standardization as an editor in ITU-T SG13 and ISO/IEC JTC1/SC6. E-mail: sjkoh@knu.ac.kr.

DOI: 10.1002/dac