Computing and Informatics, Vol. 32, 2013, 1229–1255

HMS: A HIERARCHICAL MAPPING SYSTEM FOR THE LOCATOR/ID SEPARATION NETWORK

Xiaoqian Li

National Computer Network Emergency Response Technical Team
Coordination Center of China
e-mail: `xiaoqianli@bjtu.edu.cn`

Huachun Zhou, Hongbin Luo, Hongke Zhang

Beijing Jiaotong University

Feng Qiu

Institute of Information Engineering, CAS

Ilsun You

Korean Bible University
Seoul, South Korea

Abstract. The current Internet is facing serious scalability problems and the overloading of Internet Protocol (IP) addresses is regarded as an important reason. The Locator/ID Separation Protocol (LISP) is proposed as a network-based solution that separates IP addresses into Routing Locators (RLOCs) and Endpoint Identifiers (EIDs) to address the routing scalability problems. It is a critical challenge for LISP to design a scalable and efficient mapping system. In this paper, we propose a hierarchical mapping system (HMS). HMS consists of two levels with the bottom level maintaining the EID-to-RLOC mappings in an Autonomous System (AS) and the upper level storing the mappings between EID-prefixes and ASs in the global network. We adopt one-hop Distributed Hash Table (DHT) to organize

EID-to-RLOC mappings in the bottom level and use a protocol like Border Gateway Protocol (BGP) to propagate EID-prefix-to-AS mappings in the upper level. HMS aggregates the prefixes in an AS and decreases the global mapping entries in the upper level. The evaluation results show that the number of mapping entries in HMS grows slower than the routing table size, which makes HMS scalable. In addition, the mobility in HMS does not cause mapping changes in the upper level. It makes HMS efficient in supporting host mobility. We estimate the map-requests sent to the mapping system, which show the load on HMS is small. Last, we compare HMS with LISP-TREE and LISP + ALT by quantitative analysis, in terms of resolution cost, and qualitative analysis. The results show that HMS has a good performance.

Keywords: Locator/ID separation, mapping system, hierarchical, one-hop DHT, resolution

1 INTRODUCTION

It is commonly recognized that today's Internet routing and addressing system is facing serious scaling problems. The ever-increasing user population, multi-homing, traffic engineering, and policy routing have been driving the growth of the Default Free Zone (DFZ) routing table size at an increasing and potentially alarming rate. The overloading of Internet Protocol (IP) addresses with the semantics of both "who" (endpoint identifiers) and "where" (locators for routing system) is considered to have deep implications for the routing scalability [1]. Separating the address space into identifiers and locators has been proposed to address this problem, such as the Identifier-Locator Network Protocol (ILNP) [2, 3, 4], Global locator, Local locator, and Identifier Split (GLI-Split) [5], the Locator/ID Separation Protocol (LISP) [6]. In this paper, we build the mapping system in LISP, but it is suitable for other locator/identifier separation networks with minor modification. LISP splits the IP addresses into Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). EIDs are used to identify end hosts in edge networks while RLOCs are used to forward packets in transit networks. Packets with EIDs as the source and destination addresses need to be encapsulated with RLOCs by Ingress Tunnel Routers (ITRs), which are located at the borders of edge networks, before they are sent to transit networks. Egress Tunnel Routers (ETRs) take charge of stripping the LISP-header when packets arrive at the borders of the destination edge networks.

An ITR can not encapsulate and forward packets if it does not have the EID-to-RLOC mapping for the destination EID. The mapping system is built to supply EID-to-RLOC mappings for ITRs and it is a very important component of LISP. There have been several proposals to address this issue [7, 8, 9, 10]. These proposals can be classified as hierarchical and Distributed Hash Table (DHT) architectures. LISP + ALT [7] and LISP-TREE [9] are hierarchical architectures. They have good scalability and low resolution delay as [9] shows, but they do not mention the support to mobility. LISP-DHT [8] and DHT-MAP [10] are based on DHT. LISP-DHT is a distributed mapping system which stores the EID-to-RLOC mappings on a Chord-like overlay and it has a large lookup latency [9]. DHT-MAP [10] is a mapping system that supports flat EIDs based on Content-Addressable Network (CAN) [14] while using such flat EIDs is unlikely to scale [9].

In this paper, we propose a hierarchical mapping system (HMS). It uses a hierarchical structure as well as using one-hop DHT. HMS comprises two levels. The bottom level stores the EID-to-RLOC mappings in an Autonomous System (AS) and the upper level stores the global mappings between EID-prefixes and ASs. Each AS can organize its own mapping system in the bottom level and we suggest the use of one-hop DHT to implement this. It can guarantee one hop lookup in an AS while the hierarchical architecture or other DHTs can not. We treat each EID as an individual, flat identifier in the bottom level. In the upper level, HMS propagates the EID-prefix-to-AS mapping information using a protocol like Border Gateway Protocol (BGP). By aggregating the prefixes in an AS, the number of mapping entries in the upper level can be reduced to 60 % of the current routing table size (see Section 4.1). The number of EID-prefix-to-AS mappings does not increase as fast as that of the routing table, which makes HMS scalable. We also evaluate the number of prefixes announced by an AS and the results show that the number of mapping entries in the bottom level is very small. Due to the mobility management, the mobility in HMS does not cause mapping updates in the upper level, which makes HMS support host mobility efficiently. Furthermore, we estimate the percentage of map-requests that need to be resolved by HMS, which shows that the load on HMS is very small. We compare HMS with LISP-TREE and LISP + ALT by quantitative analysis, in terms of resolution cost, and qualitative analysis. The results show that HMS has the lowest resolution cost and several benefits including scalability, fast lookup, autonomous and good mobility support.

The rest of this paper is structured as follows. In Section 2, we give an overview of LISP, one-hop DHT and present the related work. Section 3 describes HMS in detail. In Section 4, we present the evaluation results of HMS including mapping entries, mapping updates and map-requests. Section 5 compares HMS with LISP-TREE and LISP + ALT by quantitative and qualitative analysis. Finally, we conclude this paper in Section 6.

2 BACKGROUND

In this section, we present overviews of LISP and one-hop DHT followed by the related work.

2.1 The Locator/ID Separation Protocol

The locator/identifier overload of the IP address semantics is one of the causes of the routing scalability problem [1]. LISP [6] is a network-based solution that isolates

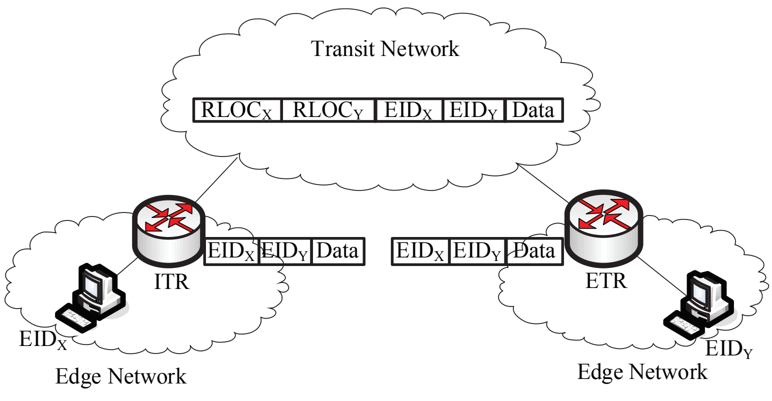


Figure 1. The network topology of LISP

transit networks from edge networks. It separates current IP addresses into EIDs and RLOCs to alleviate scaling issues caused by the use of a single numbering space. Building the solution into the network, LISP requires no change to the end-systems and minimizes the required changes to the Internet infrastructure.

In LISP, an EID is allocated to a host from an EID-prefix block associated with the edge network where the host locates. RLOCs are numbered from topologically-aggregated blocks where the topology is defined by the connectivity of transit networks. ITRs/ETRs are tunnel routers to encapsulate or decapsulate packets. Figure 1 shows the network topology of LISP. When an end host EID_X needs to contact a remote end host EID_Y, it sends a normal IP packet with EID_X and EID_Y as the source and destination addresses. When the packet arrives at the ITR, it encapsulates the packet with RLOCs and forwards the LISP-packet to the transit network. The transit network routes the packet to the ETR based on RLOCs. The ETR strips the LISP header and forwards the packet to the destination host EID_Y.

The mapping system is a major component of LISP. It is built to provide mapping information on EID-to-RLOC for ITRs when they encapsulate IP packets. If an ITR does not have the EID-to-RLOC mapping for the destination host EID_Y, it sends a map-request to the mapping system. The mapping system finds the EID-to-RLOC mapping and sends a map-reply to the ITR. The ITR stores the mapping in its cache, so subsequent packets can be directly tunneled. How to supply EID-to-RLOC mappings efficiently and correctly is a critical challenge for the mapping system.

2.2 One-Hop DHT

DHT provides scalable and practical solutions to store, locate, and retrieve information dispersed in distributed environments. One-hop DHT is a distributed hash table maintaining full routing table to achieve one hop lookup. It is reasonable when network churn is not very high. There are several proposals to perform one-hop DHT, such as passing tokens [15], D1HT [16], one-hop lookups [17].

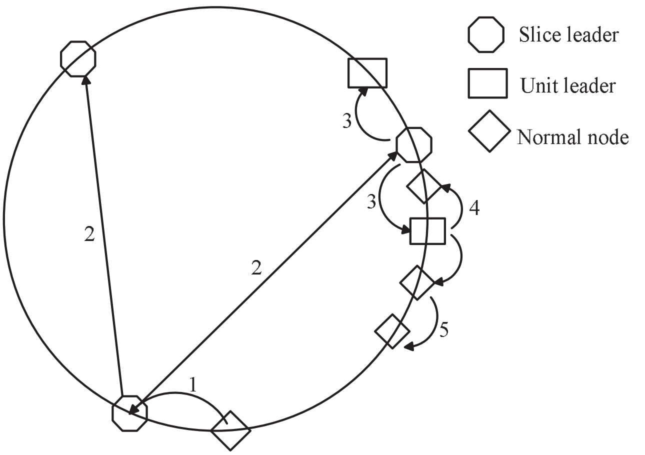


Figure 2. Notifications flow in one-hop lookups [17]

To perform one-hop DHT actually means to keep a full accurate routing table in the address space, therefore the membership changes need to be disseminated all over the network. We take the method in [17] as an example. One-hop lookups in [17] superimpose a well-defined hierarchy on the system. It divides the identifier space into equal contiguous intervals called slices which are further divided into equal-sized intervals named units. Each slice or unit has a leader. Figure 2 shows the flow of event notifications in one-hop lookups. When a normal node detects a membership change, it sends a notification message to its slice leader. The slice leader aggregates messages and dispatches to other slice leaders. The slice leaders send the aggregate messages they receive to all unit leaders in their respective slices. Afterwards, a unit leader sends the message to its successor and predecessor. The normal nodes propagate the message in one direction, and then the event notifications can be spread around the system. If a slice or unit leader fails, its successor detects the failure and becomes the new leader after a while. Doing this, the nodes can always maintain a full routing table.

2.3 Related Work

The Internet Architecture Board (IAB) Workshop on Routing and Addressing points out that the routing and addressing system is facing scalability problems and the overload of IP addresses is one of the reasons [1]. There are several proposals to split IP addresses into locators and identifiers. LISP [6] is proposed by D. Farinacci et al. It separates IP addresses into EIDs and RLOCs. The packets with EIDs as the source and destination addresses should be encapsulated with RLOCs to be routed in the transit network. ILNPv6 [2, 3, 4] replaces the IP address with the "Locator" and the "Identifier". The high-order 64-bits of the IPv6 address become the Locator and the low-order 64-bits of the IPv6 address become the Identifier. ILNPv6 Locators use the same number space as IPv6 routing prefixes, which ensures that no changes are needed to deployed IPv6 routers when deploying ILNPv6. In ILNPv4 [4], the IP Address in the IPv4 header becomes the Locator and the Identifiers are either as an IPv4 Option or as an IPv6-style Extension Header placed after the IPv4 header and before the upper-layer protocol. GLI-split [5] splits the current IPv6 address into a global locator, a local locator and an identifier, and encodes them in IPv6 addresses. The identifier of a GLI-address is fixed while the locator can be replaced by hosts or gateways on the path to the destination.

A mapping system is to provide identifier-to-locator mappings. There are several proposals to build the mapping system for LISP. LISP + ALT [7] stores the EID-to-RLOC mappings in a distributed manner. It is an overlay network to advertise EID-prefix reachability information using BGP [11] along with Generic Routing Encapsulation (GRE) [12]. LISP-TREE [9] is based on Domain Name System (DNS) and EID blocks are assigned to the levels of the hierarchy by following the current IP address allocation policies. It compares the performance of LISP + ALT, LISP-TREE and LISP-DHT with the simulator CoreSim [26]. Both LISP + ALT and LISP-TREE are hierarchical architectures. LISP-DHT [8] is a distributed mapping system which stores the EID-to-RLOC mappings on a Chord-like overlay. It uses the highest EID in an EID block as the key. DHT-MAP [10] proposes a mapping system that supports flat EIDs based on CAN [14]. A Future InteRnet Mapping System (FIRMS) [21] is a mapping system not only for LISP but also for other routing approaches based on the locator/identifier split. It introduces Map-Base (MB), Map-Base Pointer (MBP), Map-base Pointer exchange node (MBPX) and Map-Resolver (MR) to perform the mapping system. It also considers the resilience and security features.

3 A HIERARCHICAL MAPPING SYSTEM

In this section, we design a hierarchical mapping system which has two levels. We design different mechanisms for the two levels, which uses a hierarchical architecture as well as one-hop DHT. We describe the registration and resolution process in detail. We also present the support to mobility and multihoming in HMS.

3.1 The Architecture of HMS

In the Locator/ID separation network, the mapping system is built to organize the mapping information between EIDs and RLOCs. Figure 3 shows the network topology of HMS. HMS is an overlay system built to store the mapping information for EIDs. HMS consists of two levels: the upper level stores the global mappings between EID-prefixes and ASs; the bottom level maintains the mappings of EID-to-RLOC in an AS. To perform the hierarchical mapping system, we introduce three

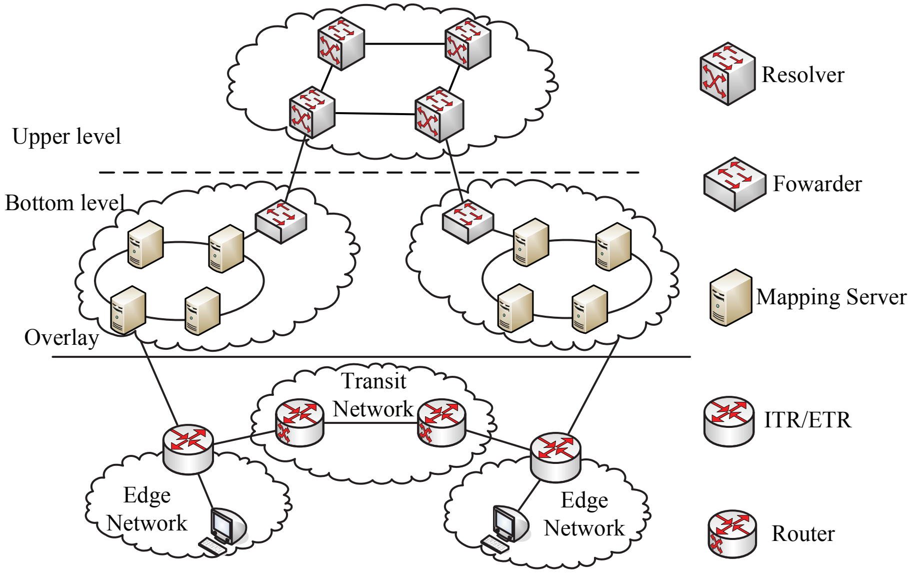


Figure 3. The network topology of HMS

types of network components: Mapping Server, Forwarder and Resolver. Mapping Servers store the local mapping information of EID-to-RLOC in an AS in a distributed manner. A Forwarder is an agent in an AS to aggregate the EIDs, to issue mappings between EID-prefixes and Forwarders, denoted by EID-prefix-to-Forwarder, and to report the mappings to the Resolvers. An AS can have one or more Forwarders. If the AS has only one Forwarder, EID-prefix-to-Forwarder is actually the mapping of EID-prefix-to-AS. Resolvers are used to store the global mapping information of EID-prefix-to-Forwarder. The Resolvers compose an inter-AS mapping system to exchange EID-prefix-to-Forwarder mappings. The three types of components are located at transit networks, and each one has at least one routing locator. They can be new entities added to the network or virtual functions running at the routers.

In the bottom level, we treat each EID as a flat identifier in an AS and suggest using one-hop DHT to organize Mapping Servers in the bottom level for several reasons. The most important one is its fast lookup. It can make sure that a large fraction (e.g. 99 %) of lookups will succeed at the first attempt [17]. Another reason is that the number of Mapping Servers in an AS is not very large so the routing table in each node can be kept small, making the lookup efficient. In addition, Mapping Servers in the mapping system are rather stable, so the network churn rate is not very high which is suitable for one-hop DHT. The edge network designer can change the organization of Mapping Servers to other architectures as long as they can exchange the EID-prefix-to-Forwarder mapping information with the upper level. In the upper level, Resolvers propagate the EID-prefix-to-Forwarder mappings running a protocol like BGP, so that each resolver stores the global mapping information. Doing this, the lookup in the upper level is very fast. When there is one Forwarder in an AS, the EID-prefixes can be highly aggregated without considering the address space deaggregation in an AS. Such a hierarchical architecture can also stop the mapping changes within an AS from impacting the upper level, so the mappings in the upper level are rather stable.

3.2 The Registration of HMS

HMS has two basic operations: registration and resolution. When an end host attaches to an ITR, the ITR assigns a RLOC to it, and registers the mapping to HMS. When an ITR receives packets from an end host, it resolves the mapping in HMS if it can not find the required EID-to-RLOC mapping in its local cache.

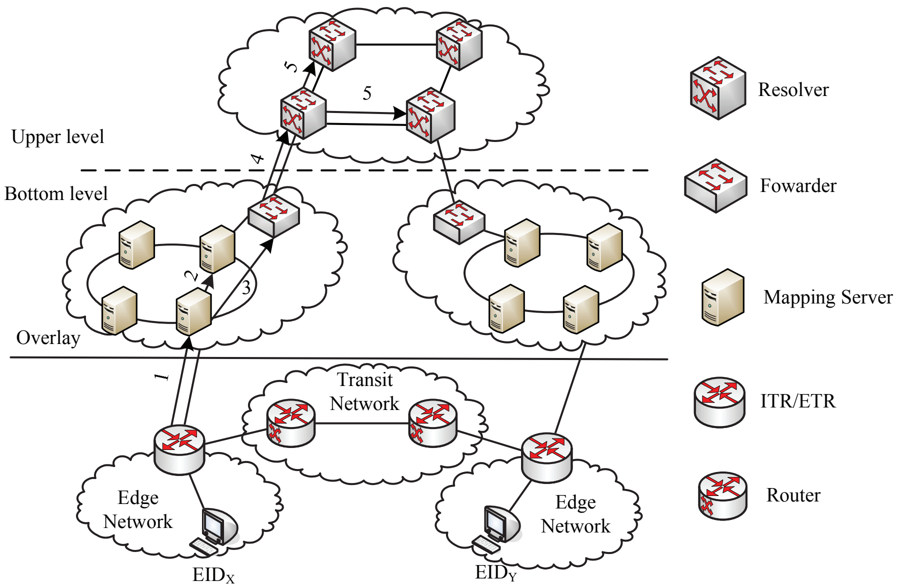


Figure 4. The registration process of HMS

After the end host EID_X attaches to an ITR, the ITR delegates a RLOC to it and registers the EID-to-RLOC mapping to the mapping system. Figure 4 depicts the registration process of HMS.

Step 1: After an ITR assigns a RLOC to EID_X, it reports the EID-to-RLOC mapping to the Mapping Server it connects to.

Step 2: Beside storing the EID-to-RLOC mapping, the Mapping Server hashes the EID_X and forwards the mapping to its successor which we call destination Mapping Server.

Step 3: The Mapping Server aggregates the EIDs in its local database and reports the EID-prefixes to the Forwarder in the same AS. We define the area that a Mapping Server covers as a Mapping Domain. It may involve several ITRs. The Mapping Server receives EID-to-RLOC mappings from all the ITRs in its Mapping Domain.

Step 4: The Forwarder receives messages from all the Mapping Servers in an AS. It aggregates the EIDs, issues the EID-prefix-to-Forwarder mapping for each EID-prefix and reports the mapping to the Resolver it connects to.

Step 5: After the Resolver receives a new mapping, it propagates the information via a protocol like BGP to other Resolvers.

Consequently, all the Resolvers maintain a full-scale EID-prefix-to-Forwarder mapping table. Notice that the mappings in Mapping Servers and Forwarders are local information within an AS, while Resolvers store the global information of the whole network.

3.3 The Resolution of HMS

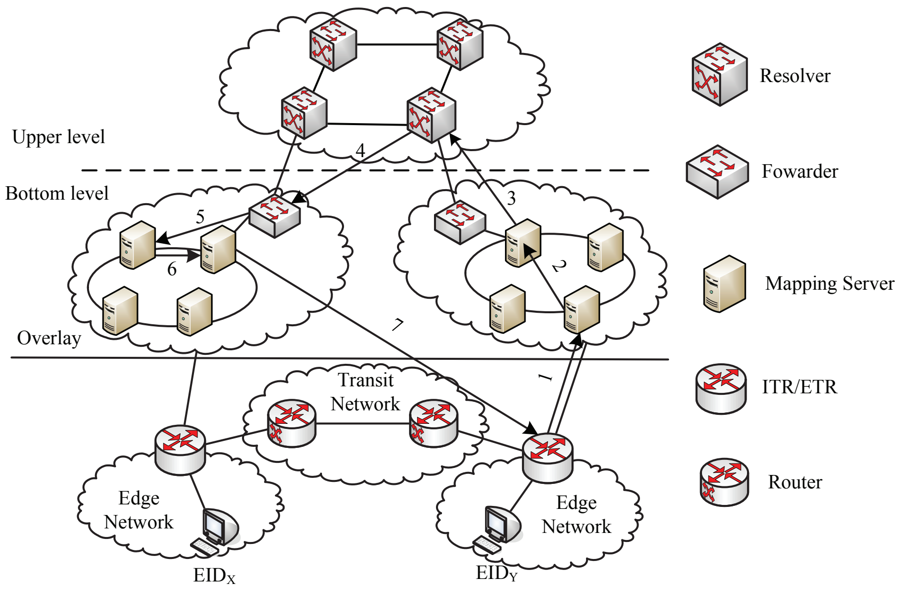


Figure 5. The resolution process of HMS

When EID_Y wants to communicate with EID_X, it sends the packets to the ITR it accesses to. The ITR resolves the mapping of EID_X in HMS if it does not find the EID-to-RLOC mapping in its cache. The resolution process is shown in Figure 5.

Step 1: The ITR first sends a map-request message to its default Mapping Sever including EID_X and the RLOC of the ITR.

Step 2: When a Mapping Server receives a map-request, it finds in its local mapping database. If EID_X and EID_Y are in the same Mapping Domain, the Mapping Server can supply a RLOC for EID_X and send a map-reply to the ITR. Otherwise, the Mapping Server hashes EID_X and forwards the map-request to the successor, i.e., the destination Mapping Server in the source AS.

Step 3: The destination Mapping Server finds the RLOC for EID_X in its mapping table. If EID_X and EID_Y are in the same AS, the destination Mapping Server can supply the EID-to-RLOC mapping for EID_X. If not, the destination Mapping Server forwards the map-request to the Resolver it is configured with.

Step 4: The Resolver finds the EID-prefix-to-Forwarder mapping for EID_X using the longest prefix matching and forwards the map-request to the Forwarder in the mapping.

Step 5: The Forwarder chooses a Mapping Server in its AS randomly and forwards the map-request to it.

Step 6: The Mapping Server hashes EID_X and forwards the map-request to its successor, i.e., the destination Mapping Server in the destination AS.

Step 7: The destination Mapping Server sends a map-reply including the EID-to-RLOC mapping for EID_X to the ITR.

After the ITR receives the map-reply, it stores the EID-to-RLOC mapping in its cache for a certain time, so the subsequent packets can be encapsulated directly without resolving in HMS.

3.4 Mobility and Multihoming

Given that the majority of communications devices are mobile terminals, efficient mobility support should be a key feature in the future Internet [27], which has been a hot topic over the last years [28, 29]. In this paper, we consider the mobility management in the mapping system.

In HMS, if a Mobile Node (MN) moves in an AS, it just updates the EID-to-RLOC mapping in the Mapping Severs and has no effect on the upper level. When the MN moves across different ASs, the Forwarder in the current AS sends an update message to the Forwarder in the MN's home AS. There is no need to update the Resolvers. When the map-request arrives at the home Forwarder, it forwards the request to the current Forwarder, and then the mapping is resolved in the current AS. The mobility management makes that the mobility in HMS does not cause mapping changes in the upper level. When an edge network changes its provider, there is no need to update the mappings in the upper level if it does not change the locator of the Forwarder. Doing this, the mappings in the upper level are rather stable, which keeps the EID-prefix-to-Forwarder mappings synchronous. The mobility management makes HMS scalable and support mobility efficiently.

If an edge network connects to multiple ITRs, i.e., the edge network is multihomed, it only needs one entry for each EID-prefix in the upper level, thus reducing the load of Resolvers. Edge networks can perform traffic engineering in the mapping system by setting different preferences and weights to the mappings.

4 EVALUATION RESULTS

In this section, we evaluate the number of mapping entries needing to be stored in HMS, mapping updates caused by mobility and map-requests sent to HMS.

4.1 Mapping Entries

LISP separates the network into transit networks and edge networks. Transit networks comprise transit routers and border routers identified by RLOCs, also some entities to perform network management. Edge networks consist of individual hosts identified by EIDs. The mapping system needs to store mapping items for EIDs so that ITRs can find the mappings for EIDs. Therefore, HMS needs to organize the mappings for edge networks. The network can be classified into transit ASs and stub ASs [18, 19]. There are some prefixes in the transit AS not for transit service. Only the IP boxes associated with transit services constitute transit networks. To evaluate the worst case, we consider all the ASs in the network including not only the stub ASs. To analyze the number of prefixes, we use the BGP data from the RouteViews Oregon Collector [20].

In the bottom level, the mapping system stores the EID-to-RLOC mappings for EIDs in an AS. The load of Mapping Servers depends on the number of EIDs in an AS. We take a sample of the RIB data [20] on 1st October, 2010 to analyze the number of prefixes that each AS announces. In Figure 6 we plot the Cumulative Distribution Function (CDF) of the number of prefixes announced by an AS with the x value on the logarithmic x-axis. The average number of hosts per EID-prefix is about 1 000 [21]. The figure shows that 42.52 % ASs announce only one prefix, that is, the Mapping Servers in 42.52% ASs need only to manage 1 000 EID-to-RLOC mappings. More than 99 % ASs announced fewer prefixes than 134, i.e., 134 000 EIDs. To gain better utilization, the network managers can combine several ASs together to organize the Mapping Servers. The largest number of prefixes announced by one AS is 4 481, i.e., 4 481 000 EIDs, which is distributed stored in Mapping Servers.

There are a large number of address space fragmentations in the routing system due to multihoming and traffic engineering. In HMS, the Forwarders aggregate the EID-prefixes in an AS and report them to the upper level. When there is only

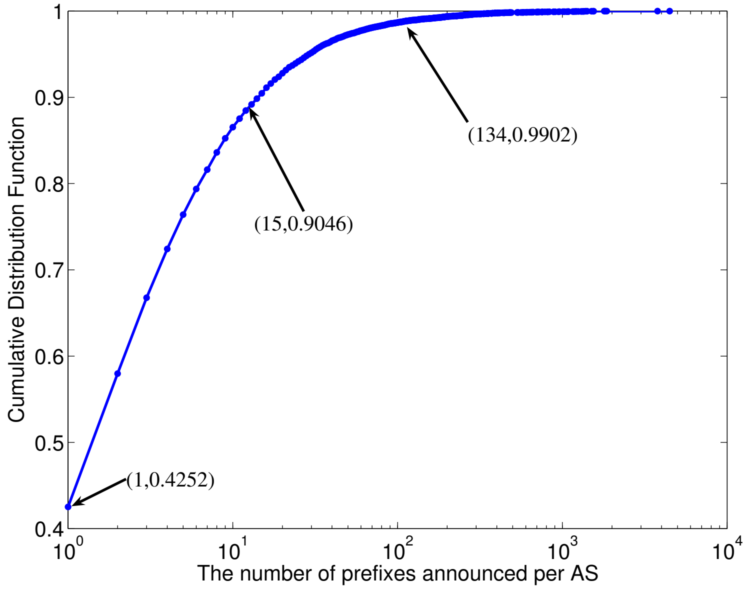


Figure 6. CDF of the number of prefixes announced per AS

one Forwarder in an AS, the Resolvers actually store the mappings of EID-prefix-to-AS. Therefore, the prefixes can be Classless Inter-domain Routing (CIDR) [13] aggregated in an AS. We program in C language under Linux operating system to perform the aggregating function. We take a sample of the first day for each month from October 2006 to October 2010. We run the aggregating function till there is no prefix to be aggregated. In the following, we count the multihoming prefixes which can further decrease the number of mapping entries in the Resolvers. Figure 7 shows the number of prefixes in the global routing table, being aggregated and after getting rid of the multihoming prefixes. From this figure we can see that, after being aggregated, the number of mappings reduces to about 60 % of the routing table size. There are about 2 000 multihoming prefixes which can further minimize the mapping table, and the curve is very close to the aggregated one in the figure. What's more, the curves also reveal that the mapping table in the upper level grows slower than the global routing table, which makes HMS scalable.

4.2 Mapping Updates

When an end host moves or an edge network changes its provider, it is supposed to update the mapping information in HMS. If the edge network changes its provider without changing the Forwarder it connects to, there is no mapping update in the

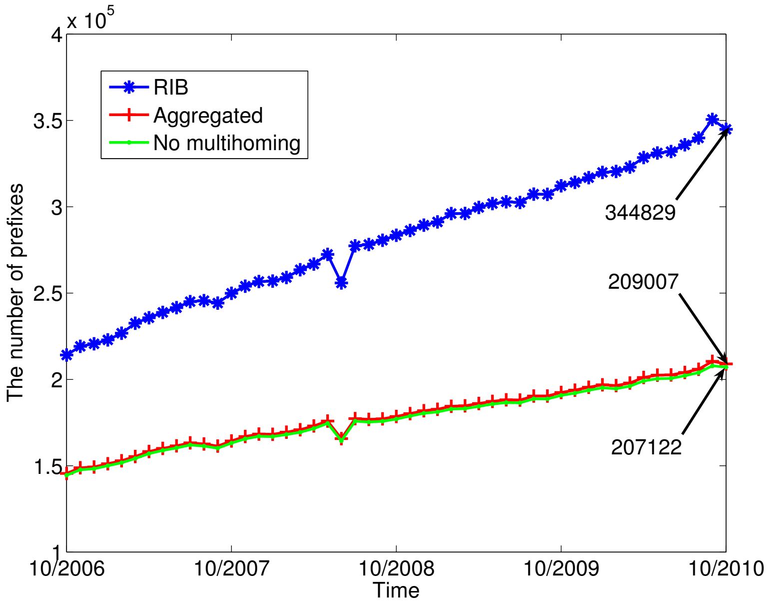


Figure 7. The number of prefixes

upper level. When an MN moves in an AS, HMS just updates the bottom level. When an MN moves across multiple ASs, it needs to send update messages to the home Forwarder without updating the upper level. Thus we estimate the mapping updates in the bottom level and sent to the Forwarder caused by mobility.

We evaluate the MN's movement behavior to estimate the mapping updates. We assume that there are n ITRs in an AS, and the area an ITR covers is s. When MNs are moving at an average speed of v, we derive from [22] that the rate r for an MN moves across an ITR is

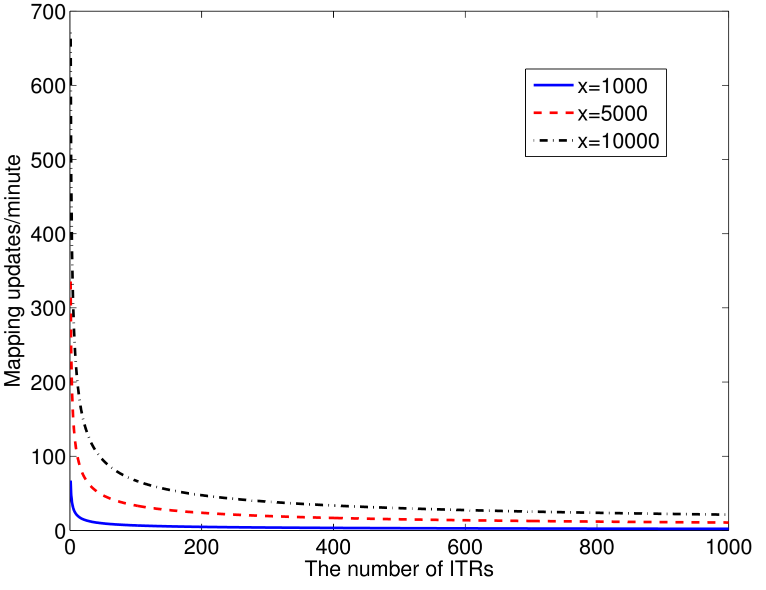
$$r = \frac{4v}{\pi\sqrt{s}} \tag{1}$$

while the MN's movement direction is uniformly distributed over $[0, 2\pi]$. The border crossing rate λ for an MN out of an AS is

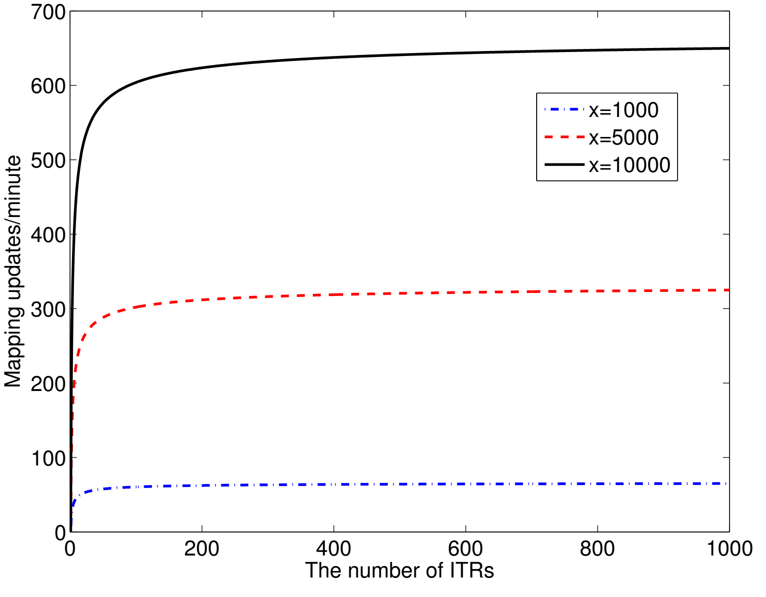
$$\lambda = \frac{4v}{\pi\sqrt{ns}}. \tag{2}$$

A MN crossing an AS moves across several ITRs, so the rate μ for a MN stays in the same AS is

$$\begin{aligned} \mu &= r - \lambda \\ &= \frac{4v}{\pi\sqrt{s}} \cdot (1 - \frac{1}{\sqrt{n}}). \end{aligned} \tag{3}$$



a)



b)

Figure 8. Mapping updates per minute; a) updates to the forwarder, b) updates in the bottom level

We assume that there are x MNs in an AS, so the number of micro mobility m causing mapping updates in the bottom level is

$$\begin{aligned} m &= x \cdot \mu \\ &= \frac{4xv}{\pi\sqrt{s}} \cdot \left(1 - \frac{1}{\sqrt{n}}\right). \end{aligned} \tag{4}$$

The number of macro mobility M which needs to send update messages to Forwarders is

$$M = x \cdot \lambda = \frac{4xv}{\pi\sqrt{ns}}. \tag{5}$$

Figure 8 shows the number of updates per minute when n varies from 1 to 1 000. We can see from Section 4.1 that more than 90 % ASs announce fewer prefixes than 15 and the average number of hosts per prefix is about 1 000 [21]. Therefore, $x = 10\,000$ and $n = 1\,000$ is enough to evaluate the performance of HMS. We set the parameters to typical values found in [22], that is, s is 10 square kilometers (km^2) and v is 10 kilometers per hour (km/h). When there are 1 000 ITRs and 10 000 MNs in an AS, there are about 650 mapping updates in the bottom level per minute and 21 updates per minute sent to the Forwarder. The more ITRs in an AS, the larger area an AS covers. Thus, the less possible an MN moves across ASs, the fewer mapping updates are sent to the Forwarder. The mobility management keeps mappings in the upper level stable and guarantees the resolution in HMS is accurate, which makes HMS support mobility efficiently.

4.3 Map-Requests

To evaluate the number of map-requests sent to the mapping system, we collected traces of the traffic from and to our campus network. Our campus network connects to the ChinaNet and CERNET through a border router which has two Gigabit links. We rely on the Netflow measurement facility supported by our border router to collect the traffic. The records provided by Netflow contain the information of the timestamp, the source and destination IP addresses and the size of packets. We use the traffic collected from 0:00 A.M. to 23:59 P.M. on October 12, 2008 to do the evaluations.

To analyze the traffic, we regard the border router as an ITR in LISP and the IP addresses as EIDs. The ITR caches the EID-to-RLOC mappings not exceeding the cache timeout. We add an EID-to-RLOC mapping to the ITR's cache when the EID is first communicated with, and remove it when it exceeds the cache timeout. We find the mapping of the destination EID in the ITR's cache. When there is none, the ITR should send a map-request to HMS.

We let ρ denote the percentage of packets that need to send map-requests to HMS, that is,

$$\rho = \frac{\textit{packets to be resolved/minute}}{\textit{packets received by ITR/minute}}. \tag{6}$$

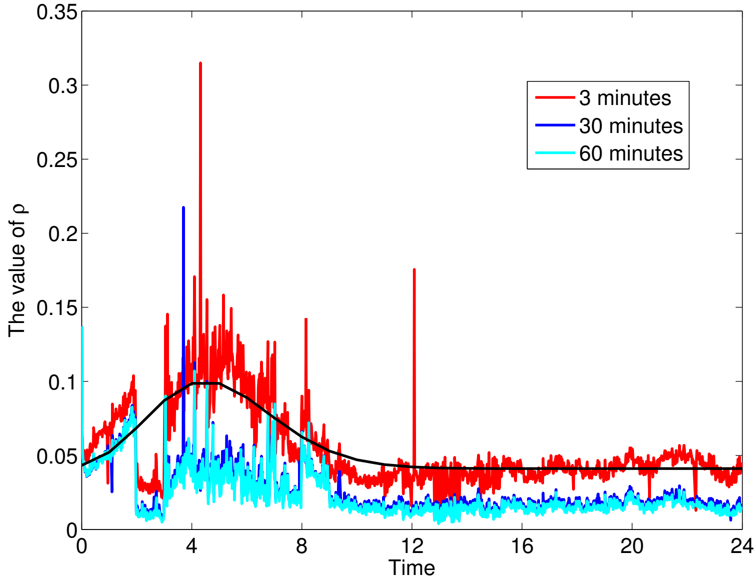


Figure 9. The percentage of packets to be resolved

Apparently, ρ is related to the cache timeout of the ITR. The ITR can store more EID-to-RLOC mappings if the cache timeout is larger, resulting in fewer map-requests. Figure 9 shows the evaluation results of ρ when the ITR's cache timeout is 3, 30 and 60 minutes. One can clearly see from the picture that even when the cache timeout is 3 minutes, ρ is less than 5 percent during the daytime and less than 15 percent during the night. In LISP, the mapping system seems to complicate the network, but only 5 percent packets issuing a map-request while 100 percent packets need to be routed by the overload routers. In this sense, it is worthy adding the mapping system to decrease the burden of the routing system. What's more, the values of ρ are almost equal when the cache timeout is 30 and 60 minutes. In this regard, we can set the ITR's cache timeout to 30 minutes to reduce the mapping entries in the mapping cache. Furthermore, the small values of ρ demonstrate that the load on HMS is very small.

We notice that ρ is related to the time during one day and it is largest in the early morning, since the active users increase and numerous items expire the cache timeout. We do the parameter estimation using the fminsearch function in Matlab [24] under an unconstrained nonlinear optimization [25]. We find a good fit between the hour t and ρ when t is integer,

$$\rho = 0.3279 \cdot \left(\frac{5^t \cdot e^{-5}}{t!} \right) + 0.0411. \tag{7}$$

The black curve in Figure 9 shows the fitting function when cache timeout is 3 minutes.

5 COMPARISONS

In this section, we make comparisons among HMS, LISP-TREE and LISP + ALT by quantitative analysis, in terms of resolution cost, and qualitative analysis.

5.1 Resolution Cost

In the following discussion, we compare the resolution cost of HMS with LISP-TREE (the iterative mode which has a better performance) [9] and LISP + ALT [7]. We do not consider LISP-DHT [8] since [9] has pointed that LISP-DHT has a large resolution delay. We evaluate a complete resolving process although there are some map-requests which can be resolved by the mapping cache.

To calculate the resolution cost, we consider the transmission cost and processing cost. We define the following parameters for HMS as Table 1 shows according to the resolution process described in Section 3.3. Figure 10 describes the resolving processes of LISP-TREE and LISP+ALT according to [9]. According to the message flows in Figure 10, we define several parameters for LISP-TREE and LISP + ALT in Table 2.

Notations	Descriptions	Values
C_{ims}	The transmission cost between the ITR and Mapping Server	$\delta \times 1$
C_{mm}	The transmission cost between Mapping Servers in an AS	$\delta \times 5$
C_{mr}	The transmission cost between the Mapping Server and Resolver	$\delta \times 5$
C_{rf}	The transmission cost between the Resolver and Forwarder	$\delta \times 10$
C_{fm}	The transmission cost between the Forwarder and Mapping Server	$\delta \times 5$
α_{ms}	The processing cost at the Mapping Server	1+log(N)
α_{dms}	The processing cost at the destination Mapping Server	1
α_{rs}	The processing cost at the Resolver	log 200 000
α_{fw}	The processing cost at the Forwarder	0

Table 1. The parameters defined for HMS

The transmission cost is related to the distance l between the entities. We assume that if the entities are directly connected, the distance is 1; if the entities are in the same AS, the distance is 5; if the entities are in the different ASs, the distance is 10. We define δ as the proportionality constant between the distance and the transmission cost, i.e., the transmission cost $C_{tran} = l \times \delta$ [23].

The processing cost is related to the number of entries that a lookup needs to search and the data structure the resource being stored. In HMS, the EID-to-RLOC

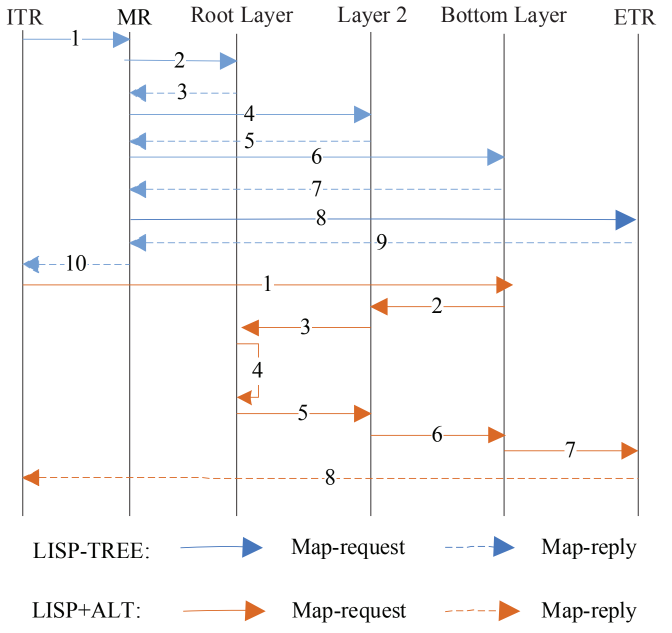


Figure 10. The resolving process of LISP-TREE and LISP + ALT (see [9] for the details)

mappings at the Mapping Server use the hash table and the complexity is $O(1)$. We assume the processing cost to search the mapping table at the Mapping Server is 1. The processing cost to search the finger table is proportional to the logarithm of the length of the finger table. Since we adopt one-hop DHT, the length of the finger table is the number of Mapping Servers in an AS. In a Mapping Server, it first searches the mapping table to find if there is the required mapping, and then searches the finger table to forward a map-request to the successor. Therefore, the processing cost at the Mapping Server is $\alpha_{ms} = 1 + \log(N)$, where N is the number of Mapping Servers in an AS. While in the destination Mapping Server, it just searches the mapping table, so the processing cost is $\alpha_{dms} = 1$. In addition, we assume that the lookup in LISP-TREE and LISP + ALT is based on the longest prefix matching and most implementations use the traditional Patricia trie, so the complexity of the lookup is proportional to the logarithm of the length of table in each layer. According to [9], we obtain the length of table in each layer. In LISP-TREE, the root LISP-TREE Server (LTS) just stores the information of layer 2, so the length of table at the root layer is 256. However, the root layer in LISP + ALT also stores the information about other root layers, so the length of table is 256+8. The number of children of layer 2 is about $1\,000 \sim 6\,072$ [9], and we take 6000 as the length of table at layer 2. Since the length of table in the upper two layer is fixed, the resolution cost is related to that in the bottom layer. The length of table in the bottom layer, denoted by

Notations	Descriptions	Values
C_{imr}	The transmission cost between the ITR and MR in LISP-TREE	$\delta \times 1$
C_{mr1}	The transmission cost between the MR and the root layer in LISP-TREE	$\delta \times 10$
C_{mr2}	The transmission cost between the MR and layer 2 in LISP-TREE	$\delta \times 10$
C_{mr3}	The transmission cost between the MR and the bottom layer in LISP-TREE	$\delta \times 10$
C_{ib}	The transmission cost between the ITR and the bottom router in LISP + ALT	$\delta \times 5$
C_{l12}	The transmission cost between the root layer and layer 2 in LISP + ALT	$\delta \times 10$
C_{l23}	The transmission cost between layer 2 and the bottom layer in LISP + ALT	$\delta \times 10$
C_{me}	The transmission cost between the MR and the ETR in LISP-TREE	$\delta \times 10$
C_{be}	The transmission cost between the bottom router and the ETR in LISP + ALT	$\delta \times 5$
C_{rr}	The transmission cost between root routers in LISP + ALT	$\delta \times 10$
C_{ie}	The transmission cost between the ITR and ETR in LISP + ALT	$\delta \times 10$
α_{al1}	The processing cost at the root layer in LISP + ALT	$\log(256 + 8)$
α_{tl1}	The processing cost at the root layer in LISP-TREE	$\log 256$
α_{l2}	The processing cost at layer 2 in LISP-TREE and LISP + ALT	$\log 6\,000$
α_{l3}	The processing cost at the bottom layer in LISP-TREE and LISP + ALT	$\log(k)$
α_{etr}	The processing cost at the ETR in LISP-TREE and LISP + ALT	$\log 100$
α_{mr}	The processing cost at the MR in LISP-TREE	0

Table 2. The parameters defined for LISP-TREE and LISP + ALT

k is equal to the number of prefixes that reports to it. When there are N bottom routers or Mapping Servers in an AS and we take 150 as the number of prefixes in an AS according to Section 4.1, we can obtain

$$N = \frac{150}{k}. \tag{8}$$

Furthermore, to calculate the resolution cost, we set the number of prefixes connected to an ETR as 100. We also obtain the number of entries at the Resolver of about 200 000 from Section 4.1, so the processing cost at a Resolver is $\log 200\,000$. What's more, we assume that the processing cost is 0, if the entities just forward a packet without lookup, such as a MR forwards a map-request in LISP-TREE.

Based on the above analysis, we obtain the values of parameters stated in the third column of Tables 1 and 2. The resolution cost C consists of the transmission cost C_{tran} and the processing cost C_{proc},

$$C = C_{tran} + C_{proc}. \tag{9}$$

According to the resolution process in Section 3.3 and Figure 10, the resolution cost of HMS C_{hms}, LISP-TREE C_{tree} and LISP + ALT C_{alt} can be calculated as follows:

$$\begin{cases} C_{hms} &= C_{ims} + C_{mm} + C_{mr} + C_{rf} + C_{fm} + C_{mm} \\ & \quad + 2\alpha_{ms} + 2\alpha_{dms} + \alpha_{rs} + \alpha_{fw} \\ C_{tree} &= 2C_{imr} + 2C_{mr1} + 2C_{mr2} + 2C_{mr3} + 2C_{me} \\ & \quad + 5\alpha_{mr} + \alpha_{tl1} + \alpha_{l2} + \alpha_{l3} + \alpha_{etr} \\ C_{alt} &= C_{ib} + 2C_{l23} + 2C_{l12} + C_{rr} + C_{be} + 2\alpha_{al1} \\ & \quad + 2\alpha_{l2} + 2\alpha_{l3} + \alpha_{etr}. \end{cases} \tag{10}$$

With the values of parameters shown in Tables 1 and 2, we can obtain

$$\begin{cases} C_{hms} &= \delta \cdot (1 + 5 + 5 + 10 + 5 + 5) \\ & \quad + \beta \cdot (2 \cdot (1 + \log(N)) + \log 200\,000 + 2 \cdot 1 + 0) \\ C_{tree} &= \delta \cdot (2 \cdot 1 + 2 \cdot 10 + 2 \cdot 10 + 2 \cdot 10 + 2 \cdot 10) \\ & \quad + \beta \cdot (\log 256 + \log 6\,000 + \log(k) + \log 100) \\ C_{alt} &= \delta \cdot (5 + 2 \cdot 10 + 2 \cdot 10 + 10 + 5) \\ & \quad + \beta \cdot (2 \cdot \log(256 + 8) + 2 \cdot \log 6\,000 + 2 \cdot \log(k) + \log 100) \end{cases} \tag{11}$$

where β is the weighting factor of the lookups, and the classic value is 0.7 [23]. We take the classic value of δ from [23], which is 0.2. We show the resolution cost versus the values of k in Figure 11. The results show that HMS has the lowest resolution cost. Since the size of the finger table at the Mapping Server decreases with the increase of k, the resolution cost of HMS decreases while that of LISP-TREE and LISP + ALT increases.

In addition, we evaluate the change of the resolution cost per unit time c during a day. Based on Equation (7), we obtain

$$\begin{aligned} c &= \rho \cdot \lambda_\alpha \cdot C \\ &= \left[0.3279 \cdot \left(\frac{5^t \cdot e^{-5}}{t!}\right) + 0.0411\right] \cdot \lambda_\alpha \cdot C \end{aligned} \tag{12}$$

where λ_α is the average packet arrival rate, and we take 4 as the value of λ_α [23]. We obtain the values when t is integer and plot the trends when k=100 in Figure 12. The resolution cost per unit time is largest in early morning, since there are more map-requests sent to the mapping system. We can also see that HMS has the lowest cost and the cost of LISP-TREE and LISP + ALT is very close due to the similar hierarchical architecture. Notice that we consider LISP-TREE and LISP + ALT

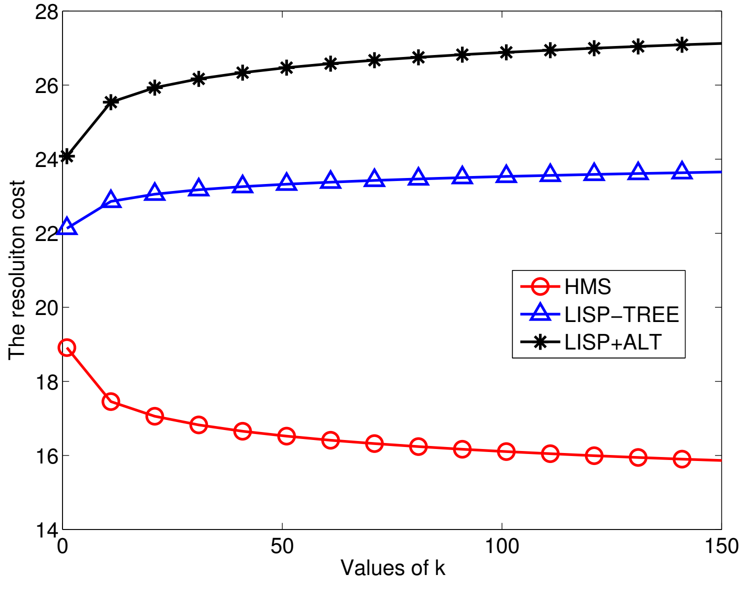


Figure 11. The resolution cost versus the values of k

with three layers while there may be more layers in the network, which causes larger resolution cost.

5.2 Qualitative Analysis

HMS has several benefits, and we compare it with LISP-TREE and LISP + ALT. The results are shown in Table 3.

	HMS	LISP-TREE	LISP + ALT
Scalability	yes	yes	yes
Fast lookup	yes	no	no
Mobility	yes	no	no
Autonomous	yes	no	no
Resolution Cost	low	high	highest

Table 3. Comparisons of HMS, LISP-TREE and LISP + ALT

1) Scalability

Scalability is one of the most important objectives of the mapping system.

HMS can highly aggregate the mapping entries, so it can save the storage space and reduce the resolution cost. The mapping entries in HMS increase slower than

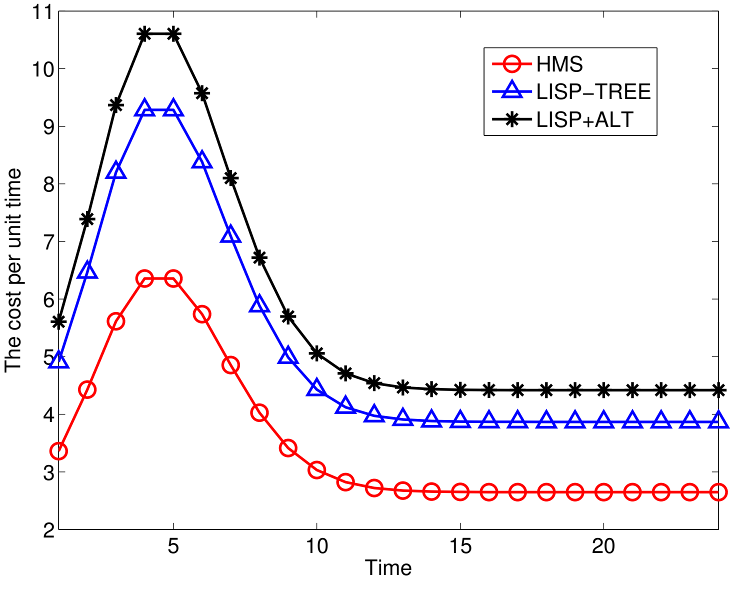


Figure 12. The resolution cost per unit time

the size of routing table, which makes HMS scalable. In addition, the hierarchical architecture prevents the mapping changes within an AS from impacting the upper level. It makes the mappings in the upper level rather stable, reducing the error caused by the mapping information asynchronization. The less dynamics of HMS reduces the processing cost and makes a less possibility to have scalability problems.

LISP-TREE is a DNS-based mapping system. It employs DNS due to its scalability, so LISP-TREE is scalable. The LISP + ALT network is built in a roughly hierarchical network, partial mesh which is intended to allow aggregation where clearly-defined hierarchical boundaries exist. Building such a structure should minimize the number of EID-prefixes carried by LISP + ALT nodes near the top of the hierarchy, to make the mapping system scalable.

2) Fast lookup

Fast lookup is another important objective for a mapping system. We suggest the use of one-hop DHT in HMS, which can achieve one hop lookup in an AS. It reduces the lookup latency considerably. In the upper level, each Resolver stores the global EID-prefix-to-Forwarder mappings, so the lookup is very fast. However, in LISP-TREE and LISP + ALT, a map-request has to be forwarded step by step to the root layer and causes large lookup latency. [9] considers a three-layer mapping system, while there may be more layers, which increases the lookup latency further.

3) Mobility

The architecture and the mobility management of HMS make it support mobility efficiently. When an end host moves in the same mapping domain, it just updates the local Mapping Server and has no effect on others. Even when an end host moves in an AS, it does not cause new EID-prefix-to-Forwarder mapping in the upper level. If an end host moves across several ASs, we introduce the mapping scheme to stop the mapping changes from impacting the upper level. The mobility management keeps the upper level stable and makes HMS have a good support to host mobility. By contraries, LISP-TREE and LISP + ALT do not consider the support to mobility.

4) Autonomy

In HMS, we separate the mapping system into two levels. In the bottom level, the network manager can organize its own mapping system as long as it can exchange mapping information with the upper level. Doing this, each AS can choose the most appropriate architecture to gain a better performance, which has no effect on the upper level and other ASs. LISP-TREE and LISP + ALT are hierarchical structures, and an AS can not design its own mapping system.

5) Resolution cost

The calculation in Section 5.1 demonstrates that HMS has the lowest resolution cost while LISP + ALT has the highest.

6 CONCLUSION

The mapping system is a very important component built to provide EID-to-RLOC mappings for ITRs in LISP. In this paper, we propose a hierarchical mapping system which has two levels. In the bottom level, we suggest the use of one-hop DHT to store the EID-to-RLOC mappings within an AS. It can achieve one hop lookup in an AS. In the upper level, HMS propagates the global mapping information between EID-prefixes and ASs using a protocol like BGP. HMS can aggregate the fragmentized prefixes in an AS, thus decreasing the size of mapping table. The number of mapping entries in HMS grows slower than the routing table size, which makes HMS highly scalable. In addition, the mobility management in HMS keeps the mapping table in the upper level rather stable, making HMS support host mobility efficiently. We also evaluate the number of map-requests sent to HMS, which shows that the load on HMS is small. Last, we compare HMS with LISP-TREE and LISP + ALT by quantitative and qualitative analysis. The results show that HMS has the lowest cost and several benefits including scalability, fast lookup, autonomous and good support to mobility.

Acknowledgements

This work is supported in part by 863 Project No. 2011AA010601, in part by National Natural Science Foundation of China (NSFC) under Grant No. 60870015 and No. 60833002, and in part by the project under the Grant No. 2011ZX03002-005-03.

REFERENCES

[1] Meyer, D.—Zhang, L.—Fall, K.: Report from the IAB Workshop on Routing and Addressing. IETF RFC 4984, 2007.

[2] Atkinson, R.—Bhatti, S.—Hailes, S.: Evolving the Internet Architecture Through Naming. IEEE Journal on Selected Areas in Communications, Vol. 29, 2010, No. 8, pp. 1219–1325.

[3] Atkinson, R.—Bhatti, S.—Hailes, S.: ILNP: Mobility, Multi-Homing, Localised Addressing and Security Through Naming. Telecommunication Systems, Vol. 42, 2009, No. 3-4, pp. 273–291.

[4] Atkinson, R.— Bhatti, S.: Identifier-Locator Network Protocol (ILNP) Architectural Description. IRTF RFC 6740, 2011.

[5] Menth, M.—Hartmann, M.—Klein, D.: global locator, local locator, and Identifier Split (GLI-Split). University of Würzburg Institute of Computer Science Research Report Series, 2010.

[6] Farinacci, D.—Fuller, V.—Meyer, D.—Lewis, D.: Locator/ID Separation Protocol (LISP). IETF RFC 6830, 2013.

[7] Fuller, V.—Farinacci, D.—Meyer, D.—Lewis, D.: LISP Alternative Topology (LISP-ALT). IETF RFC 6836, 2013.

[8] Mathy, L.—Iannone, L.: LISP-DHT: Towards a DHT to Map Identifiers Onto Locators. In Proceedings of Re-Architecting the Internet (ReArch 2008), Madrid, Spain, 2008.

[9] Jakab, L.—Cabellos-Aparicio, A.—Coras, F.—Saucez, D.—Bonaventure, O.: LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System. IEEE Journal on Selected Areas in Communications, Vol. 28, 2010, No. 8, pp. 1332–1343.

[10] Luo, H.—Qin, Y.—Zhang, H.: A DHT-Based Identifier-to-Locator Mapping Approach for a Scalable Internet. IEEE Transaction on Parallel and Distributed Systems, Vol. 20, 2009, No. 12, pp. 1790–1802.

[11] Rekhter, Y.—Li, T.—Hares, S.: A Border Gateway Protocol 4 (BGP-4). RFC 4271, 2006.

[12] Farinacci, D.—Li, T.—Hanks, S.—Meyer, D.—Traina, P.: Generic Routing Encapsulation (GRE). RFC 2784, 2000.

[13] Fuller, V.—Li, T.: Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. BCP 122, RFC 4632, 2006.

[14] Ratnasamy, S.—Francis, P.—Handley, M.—Karp, R.—Shenker, S.: A Scalable Content-Addressable Network. In ACM Conference of the Special Interest Group on Data Communication (SIGCOMM'2001), California, USA, 2001.

[15] Leong, B.—Li, J.: Achieving one-Hop DHT Lookup and Strong Stabilization by Passing Tokens. In Proceedings of the 12th International Conference on Networks (ICON 2004), Singapore, 2004.

[16] Monnerat, L. R.—Amorim, C. L.: D1HT: A Distibuted one hop Hash Table. In Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006), Rhodes Island, Greece, 2006.

[17] Gupta, A.—Liskov, B.—Rodrigues, R.: Efficient Routing for Peer-to-Peer Overlays. In 1st Symposium on Networked Systems Design and Implementation (NSDI 2004), California, USA, 2004.

[18] Dhamdhere, A.—Dovrolis, C.: Ten Years in the Evolution of the Internet Ecosystem. In Proceedings of Internet Measurement Conference (IMC 2008), Vouliagmeni, Greece, 2008.

[19] Oliveira, R.—Zhang, B.—Zhang, L.: Observing the Evolution of Internet AS Topology. In ACM Conference of the Special Interest Group on Data Communication (SIGCOMM 2007), Kyoto, Japan, 2007.

[20] The RouteViews project, [Online]. Availaible on: `http://www.routeviews.org`.

[21] Menth, M.—Hartmann, M.—Hofling, M.: FIRMS: A Future InteRnet Mapping System. IEEE Journal on Selected Areas in Communications, Vol. 28, 2010, No. 8, pp. 1326–1331.

[22] Kong, K. S.—Song, M.—Park, K.—Hwang, C. S.: A Comparative Analytical Study on the Performance of IP Mobility Protocols: Movile IPv6 and Hierarchical Mobile IPv6. The 2nd International Conference on Advances in Mobile Multimedia (MoMM'2004), Bali, Indonesia, 2004, pp. 437–446.

[23] Xie, J.—Akyildiz, R.: An Optimal Location Management Scheme for Minimizing Signaling Cost in Mobile IP. The 2002 IEEE International Conference on Communications (ICC 2002) New York, USA, 2003, pp. 3313–3317.

[24] Fminsearch (Matalb Funciton Reference). Availaible on: `http://www.mathworks.com/access/helpdesk/help/techdoc/ref/fminsearch.html`, 2001.

[25] Lagarias, J. C.—Reeds, J. A.—Wright, M. H.—Wright, P. E.: Convergence Properties of the Nelder-Mead Simplex Algorith in low Dimensions. SIAM Journal on Optimization, Vol. 9, 1998, No. 1, pp. 112–147.

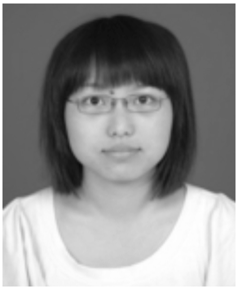
[26] Coras, F.—Jakab, L.—Cabellos-Aparicio, A.—Domingo-Pascual, J.—Dobrota, V.: Coresim: A Simulator for Evaluationg Locator/Id Separation Protocol Mapping Systems. In Trilogy Future Internet Summer School poster session, Belguim, 2009.

[27] Kafle, V. P.—Inoue, M.: Locator ID Separation for Mobility Management in the New Generation Network. Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications, Vol. 1, 2010, No. 2/3, pp. 3–15.

[28] Bernardos, C. J.—Gramaglia, M.—Contreras, L. M.—Calderon, M.—Soto, I.: Network-Based Localized IP mobility Management: Proxy Mobile IPv6

and Current Trends in Standardization. Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications, Vol. 1, 2010, No. 2/3, pp. 16–35.

[29] Yan, Z.—Zhou, H.—You, I.: N-NEMO: A Comprehensive Network Mobility Solution in Proxy Mobile IPv6 Network. Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications, Vol. 1, 2010, No. 2/3, pp. 52–70.



Xiaoqian Li received her B. Sc. degree in June 2008 and her Ph. D. degree in June 2013 from Beijing Jiaotong University, Beijing, China. Her research intersets are in the areas of the network architecture and the future Internet. She now works in National Computer Network Emergency Response Technical Team Coordination Center of China.

Huachun Zhou received his B. Sc. Degree from People's Police Officer University of China in 1986. He received his MSE and Ph. D. Degrees from Northern Jiaotong University and Beijing Jiaotong University, in 1989 and 2009, respectively. His research interests include computer network and network management technologies.

Feng Qiu received his B. Sc. degree in June 2007 and his Ph. D. degree in June 2012 from Beijing Jiaotong University, Beijing, China. His research intersets include mobility management in wireless Internet, mobility support for the next generation internet. He now works in Institute of Information Engineering, CAS.

Hongbin Luo received the B. Sc. degree from Beijing University of Aeronautics and Astronautics in July 1999, the M. Sc. degree and the Ph. D. degree in communications and information science from University of Electronic Science and Technology of China in June 2004 and March 2007, respectively. From July 2002 to March 2007, he was a research associate at the Key Lab of Broadband Optical Fiber Transmission and Communication Networks, UESTC. In June 2007, he joined Beijing Jiaotong University, where he is an Associate Professor. His research interests are in the areas of communication networks including optical networks, wireless networks, and Internet technologies.

Hongke Zhang received his M. Sc. and Ph. D. degrees in electrical and communication systems from the University of Electronic Science and Technology of China (formerly known as Chengdu Institute of Radio Engineering) in 1988 and 1992, respectively. From September 1992 to June 1994, he was a post-doc research associate at Beijing Jiaotong University. In July 1994, he joined 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 author of eight books written in Chinese and the holder of more than thirty patents. Dr. Zhang received the Zhan Tianyou Science and Technology improvement award in 2001, the Mao Yisheng Science and Technology improvement award in 2003, the first class Science and Technology improvement Award of the Beijing government in 2005, and various other awards. He is now the Chief Scientist of the National Basic Research Program ("973" program).

Ilsun You received his M. Sc. and Ph. D. degrees in computer science from Dankook University, Seoul, Korea in 1997 and 2002, respectively. Since March 2005, he has been an Assistant Professor in the School of Information Science at the Korean Bible University, South Korea. He served or is currently serving on the organizing or program committees of international conferences and workshops including CISIS '10–11, IMIS '07–11, MIST '09–11, MobiWorld '08–11, BWCCA '10–11 and so forth. Also, he has served as a guest editor for more than 10 international journals. He is on the editorial boards of International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Computing and Informatics (CAI), and Journal of Korean Society for Internet Information (KSII). His main research interests include mobile Internet security and formal security verification.