Identifier and Locator Mapping Service Research

Feng Liu\textsuperscript{1,a}, Cheng Gong\textsuperscript{2,b}, Rui Tu\textsuperscript{1}, Huatao Tang\textsuperscript{1,c}

\textsuperscript{1}School of Electronics Engineering, Naval University of Engineering, Wuhan, Hubei, China
\textsuperscript{2}No 91001 Armed Forces, Beijing, China
\textsuperscript{a}68398519@qq.com, \textsuperscript{b}8161467@qq.com, \textsuperscript{c}trbomb@163.com

Abstract. The original TCP/IP architecture has faced some critical challenges from scalability, mobility, security and so on. It also hampered the development of some new network technologies such as multi-homing, traffic engineering. One of the most important reasons is IP address semantic overloading. Academy commonly considered that we should give a clean-slate design for the naming and addressing architecture of the future Internet. “Locator/ID Split” is one of the most important research areas. The scalable flat-labels based mapping service is the kernel of “Locator/ID Split”. HLIM uses hash-based routing to provide a deterministic mapping resolution for the edge network through distributed mapping servers. HLIM satisfies the scalability of the flat-labels mapping service, and can adapt mapping nodes’ dynamic join and exit.

1. Introduction

In today’s TCP/IP protocols, the IP address had dual semantics, which implicated both the network node’s topological location and identifier (IP address semantic overloading). The IP Overload problem hampered the application of some new network technologies such as multi-homing, traffic engineering and caused the critical challenges to network scalability, mobility and security.

The IP address mixed up the function border of locator and identifier. The locator is a PA (Provider Assigned) address which is assigned by network topological structure to be global routing and has the address aggregation feature. The Identifier is a PI (Provider Independent) address which is hard to be aggregated due to the lack of network topological structure and can’t support global routing. So it is hard to use one address to apply above two functions until there is a technical breakthrough in the flat identifier routing [1]. The “Locator/ID Split” [2] is an idea to use two address space to distinguish the network node’s identifier and locator. With the “Locator/ID Split”, the organizations and users can use the PI address to implement multi-homing and traffic engineering avoid the costs of address renumbering caused by PA address. For example, many IP address-based access control strategy and configuration need to be changed. Moreover, the network security level can be enhanced by integrating the “Locator/ID Split” based routing and addressing with the source address verification.

In order to implement the “Locator/ID Split”, two address space (locator and ID) will be imported with the conversion methods. The ID address can’t support global routing, so the ID must be mapped to the locator firstly to send the packet to the remote domain. As a result, the scalable mapping service is the kernel of the “Locator/ID Split” based routing and addressing architecture.

The scalable mapping service met some challenges such as the how to control the delay of query/response, how to avoid the packet lost and how to control the size of mapping table. There are three factors to influence the scalability of mapping service: the update frequency of mapping data, the...
state size of mapping service and the delay of mapping query. If the “Locator/ID Split” is implemented without reasonable structure in the whole Internet, the size of mapping data states will be O (10¹⁰) [3]. Moreover, the update frequency of mapping data should be restricted to reduce the traffic of update message and accelerate the convergence of mapping states. The update frequency of mapping data is determined by the reachability of node and the variety of location. The deploy location, the get method and the security of the mapping service should also be considered, e.g. DOS attack.

2. Related works
It is reasonable to use DNS to implement the mapping system of ID and locator by adding host id-IP record. Although the DNS’s query is very fast, the query frequency and update speed of DNS is low which can’t satisfy the requirement of node mobility.

The GSE [4] implemented the mapping service by DNS. A new resource record format was imported to the DNS server, which assigned the locator part of address. The GSE made fixed identifier by DNS name and mapped the DNS identifier to IPv6 address by normal DNS query. When the host application wants to send packet, the DNS query will return the destination’s IPv6 AAAA record. The AAAA record will include the destination’s routing stuff (locator) and ESD (ID).

SHIM6 [5] created a mapping between the ULID (ID) and the locator to find the locator before sending packet. SHIM6 mapped the ULID into the locator through DNS reverse query. SHIM6 made the AAAA record in the reverse tree, and set the SRV record to define the static priority and weight to support traffic engineering.

Six/One [6] provided many address bunches, and all the address in the address bunch are only different in the routing prefix. The host can switch smoothly in the different address of the same address bunch without breaking off the upper layer session. Moreover, the host can modify the address bunch and publish the changes through DNS.

HIP [7] imported the rendezvous server method besides the DNS mapping service. The HIP rendezvous server mapped the host identifier into a set of IP address and forwarded the first several HIP message for the both sides of the communication. The remainder of HIP message transmitted directly between the sender and receiver. If a host’s IP address has changed, the host would inform the rendezvous server. Although the rendezvous server method accelerated the update rate and mapping information transmission of the host identifier, the rendezvous server needs to maintain a complete mapping database. The host identifier name space is tremendous, so it is hard to realize the rendezvous serve.

Some methods (e.g. TIDR [8], LISP-ALT [9]) tried to build an overlay-based mapping service system to provide mapping service for the edge network. These methods modified routing protocols (e.g. BGP) to carry mapping information. The routing protocol runs on the overlay, exchanging and updating mapping information among each mapping service nodes. Although the routing protocol-based mapping service has well extensibility, the update and convergence speed of mapping information is slow.

TIDR build a tunnel on the edge network to forward subnet traffic, which need to inform the provider AS the tunnel traffic’s client network prefix that could be accept. So TIDR imported a new transitive BGP attribute: LOCATOR which included the tunnel destination’s IP address and the tunnel encapsulation type of the given prefix, e.g. GRE [10] (Generic Routing Encapsulation). The prefix that LOCATOR attribute declared would be the pure end node identifier. The inter-domain routing doesn’t need these prefixes which will be stored in the TIB.

LISP-ALT used optional logic topology (overlay) to build mapping service. The overlay use BGP and GRE to run on the Internet. ALT is connected with GRE, uses BGP to carry the reachable information of EID prefix, announces and converges the EID prefix.

DHT is extensible and robust with self-configuration and self-maintenance characters, which may be suitable for the “Locator/ID Split” mapping service. LISP-DHT [11] is a DHT based distributed mapping system with the mapping information stored in the DHT nodes. LISP-DHT used Chord
algorithm to provide mapping service mad each node in the Chord ring use ID to be Chord-ID with mandatory. All the Chord routing information related the Chord node’s mapping item (ID, Locator) with the Chord-ID (Key) to build key-value pairs.

2.1. HLIM mapping service
HLIM (Hash-based Locator/ID Mapping) used network equipment (edge router) to perform the mapping between the identifier space and locator space by distributed mapping service system. HLIM ensured the “Locator/ID Split” transparency to the hosts. The host’s protocol stack and upper application can use the old address format which is a pure identifier without global routing capability, avoiding massive hosts update, which enhances the address space’s utilization.

2.2. Framework
HLIM imported two address space: Identifier and Locator. In order to support host mobility, the Identifier address prefix became a flat label space without aggregation feature. Every Identifier is a fixed PI address. The Locator address space is a new 64-bits PA address space and each address includes topology structure with well aggregation feature to support global routing.

HLIM adopted “mapping plus encapsulation” method. After receiving the packet from host, the edge router added a new header contained Locator on the packet, according to the mapping information gathered. In the new packet, the inner header’s source address and destination address are Identifier (IP address) and the outer header’s source address and destination address are Locator. When the encapsulated packet arrived at the destination edge router, the router decapsulated the packet and forwarded it to the destination host according to the Identifier.

2.3. HLIM basic mapping
The size of the mapping information records is tremendous, so the centralized mapping service can’t satisfy the requirement of extensibility. To solve the problem, HLIM adopted distributed mapping service which deployed several mapping servers in the network to form a logical mapping database, as showed in figure 1. HLIM ensured the extensibility of the flat label mapping service with the support of mapping service node’s dynamic join and leave.

The Identifier is a flat label space can not be aggregated, which greatly influences the extensibility of the mapping service. To solve the problem, HLIM used hash function to divide and classify the flat label space. Moreover, HLIM used the distributed hash mapping to provide the mapping resolution for the edge network, which borrowed the idea of CARP (Cache Array Routing Protocol) [12] in the web agent service.

HLIM used hash mapping to divide Identifier space into several parts. Each part had a mapping server and all the servers formed a logical whole to maintain a complete mapping service database. Each Identifier mapping data has a definite store location, that is one destination Identifier is mapped to a unique mapping server.

HLIM is a single hop resolution request, the algorithm is as followed:
1) Set \( h(\cdot) \) as the hash function which maps the Identifier to a hash space \( H \). All the edge router has the same \( h(\cdot) \) function.

2) Assume that the mapping service system is composed of \( N \) mapping server, which divides the hash space into \( N \) parts and each mapping server maintain a part of hash space \( H \)’s mapping information of the Locator/Identifier.

3) When the edge router sends packet, if there is no mapping Locator information of the destination Identifier, the router will run the \( h(i) \) (\( i \) is the destination Identifier). The function result is the hash part’s index.

4) If the \( h(i) \) result is the \( n \) part, the edge router will send the resolution request to the \( n \) mapping server. If the mapping server has the mapping record, it will send the mapping response packet to the edge router with the mapping record data. Besides launching the mapping resolution, the edge router is the data source of the mapping service system.

Each edge router has the mapping record of the local host’s Locator/Identifier and uploads the mapping record with the same hash resolution algorithm to the corresponding edge router. In case the edge router’s local mapping record is changed (e.g. the host’s join and leave), it will update the record in the corresponding mapping server.

2.4. HLIM maximum selecting mapping

The HLIM basic mapping is only fit for static node structure. In case of the mapping server’s dynamic join and leave, each mapping server’s mapping records information will be greatly moved in the whole mapping servers, which will greatly reduce the success rate of the mapping query. This situation is very similar to the CARP that for each one array node increase or reduce, one half of the records’ store location will be changed.

Assume a hash space \( H \) is \([0, 1]\) and the mapping service system has \( N \) servers, then \( H \) is divided into \( N \) parts:

\[
[0, \frac{1}{N}], [\frac{1}{N}, \frac{2}{N}], [\frac{2}{N}, \frac{3}{N}], \ldots, [\frac{N-1}{N}, 1].
\]

When a mapping server is added, the number of nodes is \( N+1 \) and \( H \) is divided into \( N+1 \) parts:

\[
[0, \frac{1}{N+1}], [\frac{1}{N+1}, \frac{2}{N+1}], [\frac{2}{N+1}, \frac{3}{N+1}], \ldots, [\frac{N-1}{N+1}, \frac{N}{N+1}], [\frac{N}{N+1}, 1].
\]

From (1) and (2), the moved mapping record block are as followed:

\[
[\frac{1}{N+1}, \frac{2}{N+1}], [\frac{2}{N+1}, \frac{3}{N+1}], \ldots, [\frac{N}{N+1}, 1].
\]

The accumulation of moved record block is

\[
\sum_{i=1}^{N} \frac{1}{N(N+1)} = \frac{1}{2}
\]

From (4), if a mapping service system adds or deletes \( M \) servers, about \( \frac{2^{M-1}}{2^M} \) mapping records will be moved. In the HLIM basic mapping, the variety of the mapping nodes number will cause great data migration.

In the HLIM basic mapping, there are two types of data migration: the one is the migration in the local node, the other one is the mapping record migration to the new nodes. If the number of mapping service system nodes is added from \( N \) to \( N+1 \), then \( 1/N+1 \) mapping records will be moved to new node. The second type of data migration is unavoidable, the first one can be avoided by accurate store algorithm with the optimum migration rate of \( 1/N+1 \).

The algorithm of the HLIM is as followed:

1) Each edge router has two hash function: \( h1(i) \) (\( i \) is the destination Identifier) and \( h2(I) \) (I is the Locator of the mapping server).

2) The edge router uses \( h1(i) \) to compute a hash value for a destination Identifier, uses \( h2(I) \) to compute a hash value for all the servers in the mapping server list, e.g. \( h2(I1), h2(I2), \ldots h2(IN) \).
3) Plus \( h_1(i) \) and \( h_2(l) \) to get a set of value: \( h_1(i)+h_2(l_1), h_1(i)+h_2(l_2), \ldots, h_1(i)+h_2(l_N) \). Choose the mapping server has the maximum hash value(\( \text{Max}(h_1(i)+h_2(l)) \)) to be the store location of the mapping record.

4) The edge router sends the mapping resolution request to the mapping server with the maximum hash value.

Assume a hash space \( H \) and the mapping service system has \( N \) servers, \( H \) is divided into \( N \) parts. With arbitrary \( i \), compute the value of \( \text{Max}_i(h_1(i)+h_2(l)) \) and store the mapping record in the \( N \)th mapping server. If the number of the servers is added to \( N+1 \), for the same \( i \):

\[
\text{Max}_{\text{new}}(h_1(i)+h_2(l)) = \text{Max}_{\text{old}}(h_1(i)+h_2(l)) \quad \text{or} \quad h_1(i)+h_2(l_{N+1}) \quad (5)
\]

If \( \text{Max}_{\text{new}}(h_1(i)+h_2(l)) = \text{Max}_{\text{old}}(h_1(i)+h_2(l)) \), the record’s store location will not be changed and will be stored in the \( N \)th mapping server. If \( \text{Max}_{\text{new}}(h_1(i)+h_2(l)) = h_1(i)+h_2(l_{N+1}) \), the record will be moved to the \( (N+1) \) mapping server. Considering the last case, the \( (N+1) \) mapping server has \( H/N+1 \) parts, so \( 1/N+1 \) old records have been moved. If the number of the mapping servers is added \( M \), the migration rate will be

\[
1 - \prod_{i=1}^{2^M-1} \left( 1 - \frac{1}{N+i} \right) = \frac{2^M-1}{2^M} \quad (6)
\]

which is much less than the HLIM basic mapping’s migration rate: \( \frac{1}{2^M} \). If the number of the mapping servers is reduced, the old records will not be moved.

From (6), it is evident that the migration rate is related with the number of the mapping service system’s original nodes in the HLIM maximum selecting mapping. The larger the number \( N \) is, the fewer records will be migrated. As a result, the nodes size should be large enough when building a mapping service system.

2.5. HLIM multilevel mapping

There are some scalability problems with the above two algorithms. Firstly, the potential size of the Identifier space is very large, each mapping server’s mapping records table is much larger than the routing table of the core router. Secondly, each edge router must maintain a Locator reachable list of all the mapping servers. When launch a query, every item in the list must be computed. If the number of the mapping servers is very large, the computation cost is large. As a result, the HLIM multilevel mapping is proposed.

The HLIM multilevel mapping system is composed of several level mapping subsystems. Considering a two-level mapping system as figure 2. The Tier0 is the root system which directly responds the mapping resolution request from the edge router. Each node in the Tier0 divides its hash space into a new tier (Tier1). Assume a hash space \( H \), the Tier0 has \( N \) nodes, so each node represents \( H/N \) hash space. The Tier1 has \( M \) nodes and each node (include root node n) represents \( H/MN \) hash space.

![Figure 2: HLIM multilevel mapping service](image)

The algorithm of the HLIM multilevel mapping is as followed:

1) According to the destination Identifier and the Tier0 mapping server list, the edge router uses the hash function to compute the store location (e.g. node n) in the Tier0 and sends the mapping resolution request to the node.
When the node n receives the resolution request, it will send the response message to the edge router if there has local mapping record. Otherwise, the node n uses its hash function to compute the destination Identifier and the Tier1’s mapping servers list to get the Tier1 node n’ which has the mapping record and forwards the edge router’s resolution request to the node n’.

3) When the node n’ receives the resolution request, it will send the response message to the edge router according to local mapping record.

The HLIM multilevel mapping will greatly reduce the mapping records size that each mapping server maintained. The edge router only needs to maintain a smaller Tier0 mapping servers list, which reduces the edge router’s pressure. But the HLIM multilevel mapping is no longer a single hop resolution, which increases the forwarding path of the resolution request and adds the mapping query delay of the edge router.

3. Performance analysis

The HLIM basic mapping and maximum selecting mapping are single hop resolution, their resolution time complexity is $O(1)$. The LISP-DHT is a mapping service based on Chord algorithm and its average resolution time complexity is $O(\log n)$ ($n$ is the number of the nodes in the Chord ring). The HLIM basic mapping and maximum selecting mapping’s resolution is more efficient than the LISP-DHT.

The edge router’s query time ($T_{query}$) is composed of hash computation time ($Process_{hash}$), transmission delay ($delay_{RS}$) and server record query time ($Process_{lookup}$):

$$T_{query} = Process_{hash} + delay_{RS} + Process_{lookup}$$ (7)

The cost of the hash computation and server process is a fixed value, so we concentrated on the transmission delay’s influence on query time. Assume the HLIM multilevel mapping has N levels and each level’s mapping server number is $M_0$, $M_1$, … $M_{N-1}$. If the nth level’s query probability is $P_n$, then

$$P_0 = \frac{1}{M_0 M_1}, \quad P_1 = \frac{1}{M_0 M_1 M_2}, \quad \ldots \quad P_{N-2} = \frac{1}{M_0 M_1 M_2 \ldots M_{N-2}}, \quad P_{N-1} = \frac{1}{M_0 M_1 M_2 \ldots M_{N-2}} \left(1 - \frac{1}{M_{N-1}}\right).$$

The transmission delay of the HLIM multilevel mapping is

$$delay_{multi-tier-RS} = P_0 \cdot delay_0 + (1 - P_0)P_1 \cdot delay_1 + \ldots + (1 - P_0)(1 - P_1)\ldots (1 - P_{N-2})P_{N-1} \cdot delay_{N-1}. \quad (8)$$

Assume each level’s delay is equal to $delay_{const}$, from (8) we can get

$$delay_{multi-tier-RS} = delay_{const} \cdot \left(\sum_{i=0}^{N-1} \left((\prod_{j=0}^{i-1} (1 - P_j)) \cdot P_i\right)\right). \quad (9)$$

Import the $P_n$ to (8):

$$delay_{multi-tier-RS} = delay_{const} \cdot \left(\sum_{i=0}^{N-2} \left((\prod_{j=0}^{i-1} (1 - \frac{1}{\prod_{k=0}^{j+1} M_k})) \cdot \frac{1}{\prod_{k=0}^{i+1} M_k}\right) + (\prod_{i=0}^{N-2} (1 - \frac{1}{\prod_{k=0}^{i+1} M_k})) \cdot \frac{1}{\prod_{k=0}^{i+1} M_k}\right). \quad (10)$$

Considering to the HLIM single level mapping, assume the transmission delay is $delay_{const}$, we can get

$$\frac{delay_{multi-tier-RS}}{delay_{single-tier-RS}} = \left(\sum_{i=0}^{N-2} \left((\prod_{j=0}^{i-1} (1 - \frac{1}{\prod_{k=0}^{j+1} M_k})) \cdot \frac{1}{\prod_{k=0}^{i+1} M_k}\right) + (\prod_{i=0}^{N-2} (1 - \frac{1}{\prod_{k=0}^{i+1} M_k})) \cdot \frac{1}{\prod_{k=0}^{i+1} M_k}\right) \cdot \frac{M_{N-1} - 1}{M_{N-1} - 1 - \frac{1}{\prod_{k=0}^{M_{N-1} - 1} M_k}}. \quad (11)$$

4. Conclusion

The core of the “Locator/ID Split” is the scalable mapping service based on flat label. The HLIM adopts a single hop resolution based on hash and provides a mapping resolution mechanism for the edge network by a distributed mapping service system. The HLIM has the scalability of the flat label
mapping service and partly resolves the mapping data migration problem caused by mapping service node’s dynamic join and leave. We also analyzed the multilevel mapping’s influence on query delay and drew the relation between the query delay and the multilevel mapping structure. At present, the HLIM can only support IPv4 address. A mapping query mechanism to support IPv4/IPv6 will be presented in the future.

References
[1] Matthew Caesar, Tyson Condie, Ion Stoica, Scott Shenker ROFL: Routing on Flat Labels. ACM SIGCOMM'06, 2006, pp.363-373.
[2] D.Meyer, L.Z., K.Fall. Report from the IAB Workshop on Routing and Addressing. Internet Draft, 2006.
[3] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, z. The Locator Identity Separation Protocol (LISP). The Internet RFC ,2006, pp.6830.
[4] Matt.Crawford, A.M., Thomas.Narten, John W.Stewart, Lixia Zhang. Separating Identifiers and Locators in Addresses: An Analysis of the GSE Proposal for IPv6. Internet Draft, 1999.
[5] E.Nordmark, M.B., Shim6: Level 3 Multihoming Shim Protocol for IPv6. Internet RFC, 2009, pp.5533.
[6] Vogt, C. SiX/One:A Solution for Routing and Addressing in IPv6. Internet Draft, 2007.
[7] J. Laganier, L. Eggert. Host Identity Protocol (HIP) Rendezvous Extension. Internet RFC, 2016, pp.8004.
[8] Adan, J.J. Tunneled Inter-domain Routing (TIDR). Internet Draft, 2006.
[9] V. Fuller, D. Farinacci, D. Meyer, D. Lewis. LISP Alternative Topology (LISP-ALT). Internet Draft, 2010.
[10] D. Farinacci, T.L., S. Hanks, D. Mayer, P. Traina, 2000. Generic Routing Encapsulation (GRE). RFC 2784.
[11] L. Mathy, L.U., L. Iannone, O. Bonaventure. LISP-DHT: Towards a DHT to map identifiers onto locators. Internet Draft, 2008.
[12] Vasilis Sourias, Ioannis Psara. Efficient Hash-routing and Domain Cluster Techniques for Information-Centric Networks. Computer Networks, vol.103, no.1, pp.67-83, 2016.