J. Ecophysiol. Occup. Hlth. 16(1&2), 2016, 59–65

DOI : 10.15512/joeoh/2016/v16i1&2/15638

Design and Implementation of a Hierarchical ID/Locator Mapping System

Byeong-Ok Kwak[1], Sang-Ho Lee[2]* and Tae-SooChung[1]

[1]Trustworthy Networking Research Section, ETRI, 34129, Korea; kwakbok@etri.re.kr, tsjeong@etri.re.kr
[2]Department of Software, Chungbuk National University, 28644, Korea

Abstract: Background/Objectives: *The Identifier (ID) and Locator(Loc) split is a key concept in the current Internet environment and the future Internet environment. The binding of ID to Loc is necessary to route/forward packets, which is called as ID/Loc mapping system.* **Methods/Statistical analysis:** *We propose a novel ID/Locator Mapping System (ILMS) mechanism using the bloom filters (BFs) to store information about the identifier. The proposed bloom filters stores the information about the identifiers belonging to the domain managed by the ILMS.* **Findings:** *The critical challenge in building the mapping system is scalability simply because the vast number of ID exists in the network and there is no mechanism that can be aggregated as an IP address, since IDs are assumed to be flat. The proposed mapping system is a network of ILMS server in a forest by multiple trees with peering relationship. The main idea is to construct the hierarchical ILMS to manage only a limited number of IDs and so can extract a bit that can be tested for given ID in each ILMS.* **Improvements/Applications:** *Scalability issue was addressed by compressing the BFs to indicate the IDs of the child ILMS server.*

Keywords: Bloom Filter, Future Internet, Locator/Identifier Separation, Mapping System

Introduction

In all ID/Loc separation architectures, location independent identifier(ID) is assigned to the communication entity itself unlike IP address assigned to the network interface, where the communication entity includes host, content, service, etc.[1,2,3]. The critical challenge in building the mapping system is scalability simply because the vast number of IDs exists in the network and there is no mechanism that can be aggregated as an IP address, since IDs are assumed to be flat.

In order to address the mobility issue and other current Internet issues such as multi-homing, content networking, trustworthy, diversity of the internet, etc., several future internet architecture projects[4,5]are exploring an ID-based networking architecture, where ID denotes the communication entity itself. ID can be hierarchically structured or flat. However, location-independent ID is assumed in ID-based networking, which means that ID does not contain anything about the node's topological or geographical information. Thus, the binding of ID to Locator (Loc) is necessary to route/forward packets in ID-based networking, which is called as ID to Loc mapping system.

The mapping system is basically storing and managing the mapping data between identifiers and locators of communication entities in many ways such as developing distributed mapping database or centralized mapping database, building a forwarding network, etc. The mapping system must be scalable to the number of the explosive increasing communication entities, i.e. IDs. Thus, an important challenges in the design of ID-based networking[6]is how to design a mapping system.

The important issue in the design of the future internet is mobility support simply because of the future internet will be developed into mobile-centric environment. The current internet was

***Email: shlee@cbnu.ac.kr**

designed based on the fixed network rather than the mobile network. And many challenging researches are in progress with the ID-based Networking. With these observations, we design a novel ID/Loc Mapping System (ILMS) using bloom filters in ID-based Networking. ILMS is assumed to have a hierarchical structure to achieve scalability. This paper provides the overall structure and procedures of ILMS.

A Hierarchical ID/Loc Mapping System

One of the key challenges in design of the ILMS is scalable to ensure the increasing number of IDs. In order to resolve the scalability issue, IDs are needed to distributed rather than centralized. Bloom Filter (BF)[7] are well known as having a space efficient probabilistic data structure and it is used to check whether a specific member is a elements of a set. BF also has several disadvantages such as false positive error and no member deletion. However, we can minimize the false positive error up to the tolerant point although there is no way to get rid of it. We can also use alternative ways to address the no membership deletion such as reconstructing the BF periodically or using a counting BF.

Structure

ILMS stores and maintains ID to locator mapping information, where it takes an ID as its input and produces locator that the ID is currently associated with.

We design a set of mapping servers (MSs) with an extended hierarchical tree structure, where the network of MSs includes a peering relationship, as well as to define a parent-child relationship. Figure 1 shows an example of the hierarchical ID/Locator mapping system structure. At the top of the hierarchy, the MSs are fully peered. It means that each server in the peer relationships shares its information of all IDs. In other words, MSs at the top level can be seen to have a global knowledge of all IDs in the network. At the bottom, a lowest MS

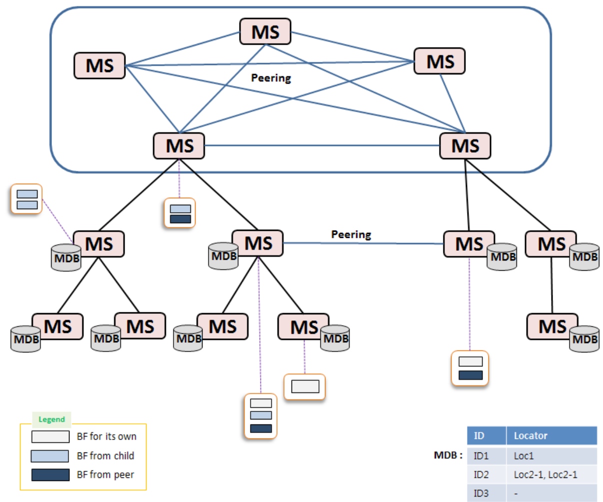


Fig. 1 Hierarchical mapping system structure.

knows the information about all the ID/Loc pair that it manages. At the intermediate level, the MS scan knows the information about the ID/Loc pair of all the identifier directly registered to them but only poses information about the IDs that managing their child MSs and peering MSs.

Component

ILMS consists of MSs (Mapping servers) and MDBs (Mapping Data Bases), which are described in the following subsections. Fig. 1 shows one of examples of HRS structure. We note that MS and MDB are conceptually separated servers but they can be implemented together into as one server.

Mapping Server

An MS consists of multiple bloom filters and an ID lookup table. Although probability, bloom filter has constant search time and provides a space-efficient data structure to represent the membership information of vast amount. Hence bloom filter is ideal filter in order to implement the MS.

ID to Locator Mapping Database

The ID to Locator Mapping Database (MDB) manages and maintains the mapping table which is managed all the ID/Locator mappings information for all IDs that are directly registered to the MS. Fig. 2 is one of examples of mapping table. The mapping table consists of an ID and associated Locator(s). It is possible that an ID has more than one Locator. In this case, MS responds with a set of locators. It is also possible that an ID is registered to the MS but has not been presented at the network. In this case, the ID is inserted in the mapping table but Loc will be inserted when it is actually presented in the network.

Procedure

When a specific entity wants to access the network, that entity must register itself to an MS. The communication entity has to perform four steps in order to access the MS. First, the new communication entity must be able to contact with the MS. Hence, communication entity with some bootstrap method brings some information about contacting MS. Until the new communication entity becomes a member of the MS, the communication entity is used as an entry point to the MS.

Then, the new communication entity needs to be registered a mapping server in the hierarchical ID/Loc mapping system. Depending on the implementation of the MS, a communication entity may also select arbitrary or specific mapping servers on its own or it chooses on the basis of the current state of the system. Third, to reflect the presence of the new communication entity, it needs to set appropriate bits in bloom filters through all parents. Fourth, the new communication entity notifies of its presence.

ID	Locator
ID1	Loc1
ID2	Loc2-1, Loc2-2
ID3	Loc3-1, Loc3-2, Loc3-3
ID4	-

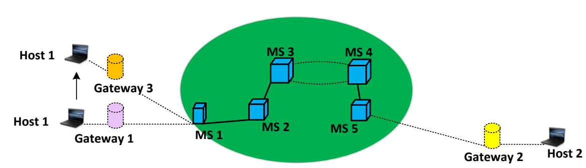
Fig. 2 Mapping table on MDB.

Registration, Presence and Location Update Procedure

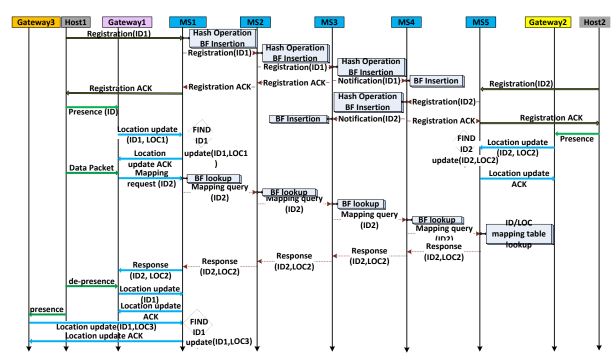
Fig. 3 shows registration, presence, and location update procedures. ILMS in Fig. 3 is composed of five MSs.MS1 (MS5) is a leaf MS storing the ID1/Loc (ID2/Loc) mapping. MS3 is fully peered with MS4.

In Registration Procedure

1. Since the registration request message finds its path depending on the parent-child relationship and peering relationship, the host1 sends the registration request message to the MS1. MS1, MS2, and MS3 receive the registration request message, successively. The output of the hash function for the ID1 is inserted in the bloom filters managed by MS1, MS2, and MS3.
2. XS3 notifies the registration of host1 to the XS4. MS4 inserts the output of the hash function for the ID1 in bloom filter. MS3 (MS2) sends the registration acknowledgement (ACK) message to MS2 (MS1).



(a) Scenario



(b) Procedure

Fig. 3 Registration, presence and location update.

In the Presence and Location Update Procedure

1. The host1 sends the presence message to the gateway1.
2. The gateway1 delivers location update message to the MS1.
3. MS1checks which server stores the ID1/LOC mapping. MS1 adds ID1/LOC mapping in ID/LOC mapping table. MS1 sends the ACK of location update message to gateway1. If MS1 is not allowed to store ID1/LOC mapping, MS1 sends ID1/LOC mapping to permitted MS. Permitted MS adds ID1/LOC mapping information in ID/LOC mapping table and sends the ACK of location update message to gateway1.

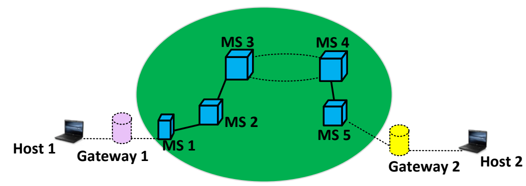
When the Host1 Operates the Location Update Procedure

1. When the host1 moves new domain managing gateway 3, the host1 sends the presence message to gateway3 and de-presence message to gateway 1. The gateway3 sends the location update message including ID1/LOC3 and the gateway 1 sends the location update message including ID1 to MS1.
2. MS1 sends the location update acknowledgement message to gateway 1 and gateway 3.

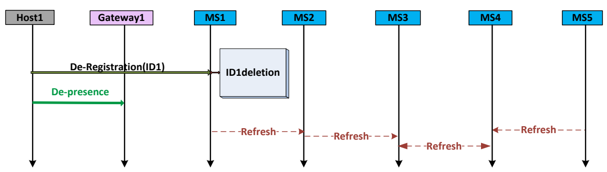
De-Registration and Refresh

When a communicating entity wants to leave the network, that entity must de-register itself to a MS. When the MS receives the de-registration request from the communicating entity, the MS deletes the ID/Loc mapping information associated with the entity. De-registration message is propagated through the parent-child relationship and peering relationship.

Fig. 4 shows the example of de-registration and refresh procedure.



(a) Scenario



(b) Procedure

Fig. 4 De-registration and refresh.

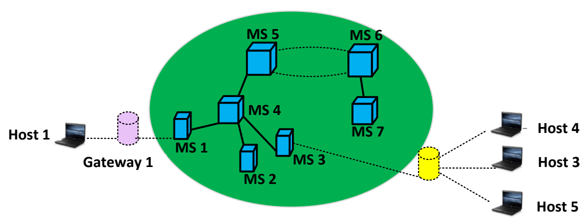
1. When the host1 registered to network leaves the network, the host1 sends the de-registration message to ILMS1 and de-presence message to gateway1. MS1 deletes the ID1/LOC mapping.
2. MS1 refresh the bloom filter at the refresh time.

ID/Loc Mapping Request and Response Procedure

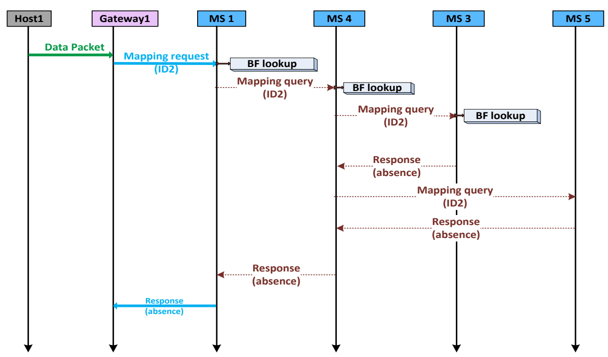
The mapping request and response operation is a process to find the location information for a given identifier. One example of mapping request and response procedures is illustrated in Fig. 5.

When a host2 is registered and the host1 sends data packet to host2,

1. The host1 sends data packet to gateway1, the gateway1 sends ID2/LOC mapping request to MS1.
2. MS1 checks the ID2 in ID/LOC mapping table or the bloom filter, and sends ID2/LOC mapping request to MS2. Since MS5 has the ID2/LOC mapping, the bloom filters of MS4 and MS5 includes the ID2. Thus, the ID2/LOC mapping request is delivered to MS3, MS4, and MS5, successively.
3. The ID2/LOC mapping response is delivered to MS5, MS4, MS3, MS2, and MS1, successively.



(a) Scenario



(b) Procedure

Fig. 5 Absence and false positive example.

False Positive and Absence Procedure

The bloom filter has a weakness of false positive occurrence, and multiple bloom filters may give positive responses. When the bit sequence corresponding to the nonexistent identifier are set to 1s, in relation to the specific identifier is not present in the network, a false positive occurs. Anyway, this eventually triggers a search in parallel in order to search the location information associated with the identifier.

Fig. 5 shows one example of false positive scenario.

When the host2 is not registered to MS and the host1 tries to send packet to host2,

1. The host1 sends the data packet to gateway1 and gateway 1 sends ID2/LOC mapping request to MS1. MS1 checks the ID/LOC mapping table or its bloom filter.
2. Since the host2 is not registered to MS and MS3 manages the bloom filter including the false positive bit sequences for host2, the ID2/LOC mapping request is delivered to MS4, MS3.
3. ILMS3 checks the nonexistence of ID2/LOC in the ID/LOC mapping table.
4. And ILMS3 sends the response of absence to MS4 and 1. MS4 sends ID2/LOC mapping request to MS5.
5. MS5 checks the nonexistence of ID2/LOC in the network and sends the response of absence to MS4 and 1.

Fig. 6 summarizes the operation flows with a simply structured ILMS which has only 2 binary trees of height 1. The scenario is that two hosts are registered in two different ILMS servers. Thus, each ILMS server does the bloom filter (BF) update of their parents and then the two ILMS servers at the top update each other. When a host presents in the network, its locator is stored in the lookup table where its ID is stored. Fig. 6 also depicts a simple lookup operation as well as the operations of de-presence and deregistration.

Implementation Results

We have implemented a prototype for our ILMS: ILMS server, parent server, and child server. All ILMS servers perform the same functions, but we implement by separating the top server from another servers for convenience of implementation. We used the parallel processing of a Graphics Processing Unit (GPU) to accelerate the performance of

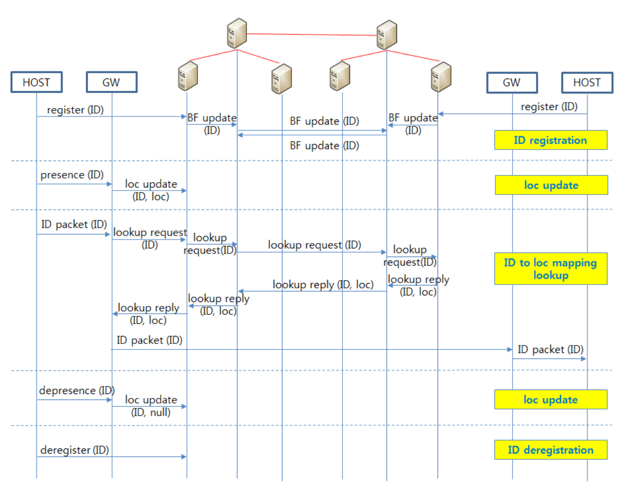


Fig. 6 Operation flows.

the bloom filter check as a result of low latency at each ILMS server.

Fig. 7 shows the overview of the BF processing method utilizing the GPU. The main idea is to allow to extract the bits sequence corresponding to the given ID from all BFs in the GPU memory of each server and examine the extracted bits in parallel to find the given ID if any chunk provides 1 by bit wise AND operation.

In this implementation, we use11 hash functions in order to ensure the false positive probability less than or equal to $4.586*10^{-4}$on the assumption that each of the BF can have a maximum information of 10^6IDs, and16Mb Bloom Filter size. We have used the static tree structure such that each ILMS was so managed by configuration files. We have also implemented the ILMS without GPU usage to compare the effect of the GPU usage. Figure 8 shows the execution time of 10,000 lookup queries, where the x-axis is the number of bloom filters in each server and for GPU, NVIDIA TITAN X 6GB is used. Fig. 8 shows that for the smaller number of bloom filter; CPU version has the lower execution time. But once the number of bloom filters is greater than about 2800, GPU version keeps the lower value than CPU one.

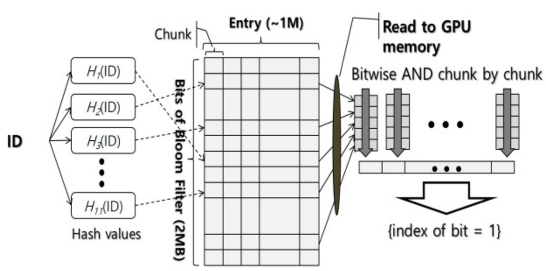


Fig. 7 BF processing method utilizing the GPU.

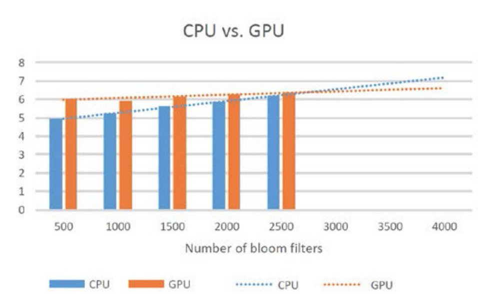


Fig. 8 Execution time (s).

Conclusion

In this paper, we proposed a hierarchical ID/Locator mapping system in ID-based networking. The proposed mapping system is a set of MS server in a forest by multiple trees with peering relationship. Scalability issue was addressed by compressing the BFs to indicate the IDs of the child MS server. The peering relationship was used to reduce the traffic load to the MS servers at the higher level of the proposed system. We implemented our system for proof of concept and presented some experimental results

In the future research work, we will continue to improve the functional features of a hierarchical ID/Loc mapping system and try to find solutions we had the unsolved issues.

Acknowledgment

This work was supported by IITP grant funded by the Korea government (MSIP) [B0717-16-0040, Development of self-certifying ID based trustworthy networking technology]

References

Jacobson, et al. (2007) Networking named content V. Proceedings of the ACM CoNEXT conference, New York, NY, USA. *Association for Computing Machinery (ACM).,* 1–12.

Anand, A., et al. (2011) XIA: Architecture for an evolvable and trustworthy internet. In the Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X), Cambridge, MA. *Association for Computing Machinery (ACM).,* Nov 14–15, 2011.

Vu, T., et al. (2012) D Map: a shared hosting scheme for dynamic identifier to locator mappings in the global internet. *Institute of Electrical and Electronics Engineers (IEEE) International Conference on Distributed Computing Systems (ICDCS).*, Jun, 698–707.

Seskar, I., Nagaraja, K., Nelson S., and Raychaudhuri D. (2011) Mobility first future internet architecture project. *In the Proceedings of the 7th Asian Internet Engineering Conference (AINTEC), Bangkok, Thailan*., Nov. http://mobilityfirst.winlab.rutgers.edu/

Kwak, B., Lee, T., and Chun, W. (2012) ID based communication in domain-insulated autonomous network architecture (DIANA). *ICTC*., Oct.

Hong, J., Taewan Y., and Heeyoung J. (2015) Qualitative analysis of mapping systems in the Internet architectures based on identifier/locator separation. *International Journal of Parallel Emergent and Distributed Systems*., Aug, 31(1), 1–15.

Bloom BH., (1970) Space/time trade-offs in hash coding with allowable errors. *Association for Computing Machinery (ACM)*., 13(7), 422–6. Article title. http://www.sail-project.eu/,

Scalable and Adaptive Internet Solutions (SAIL).

Vu, T., Baid, A., Zhang, Y., Nguyeny, T., Fukuyamaz, J., Martin, R., and Raychaudhuri, D. (2012)

DMap: a shared hosting scheme for dynamic identifier to locator mappings in the global internet.

In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Distributed Computing Systems., 698–707.

Ambrosio, MD., Dannewitz, C., Karl, H., and Vercellone, V. (2011) MDHT: a hierarchical name resolution service for information-centric networks. *ICN, Toronto, Ontario, Canada*., August 19, 7–12.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M., Dabek, F., and Balakarishnan, H. (2001) Chord: a scalable peer-to-peer lookup protocol for internet applications. *In the Proceedings of the Association for Computing Machinery (ACM) SIGCOMM'01, San Diego, CA*., Aug, 149–60.

Sevedio, VDP., and Caldarelli, G. (2004) Vertex intrinsic fitness: how to produce arbitrary scale-free networks. *Physics Review.*, E70(5), 056–126.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001) A scalable content - addressable network. *In the Proceedings of the Association for Computing Machinery (ACM) SIGCOMM, San Diego, CA*., Aug, 31(4), 161–72.

Rowstron, A., and Druschel, P. (2001) Pastry: scalable, distributed object location and routing for large - scale peer-to-peer systems. *18th IFIP/ Association for Computing Machinery (ACM) International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany.*, Nov, 1–22.

Faujdar, N., and Ghrera, SP. (2016) Performance evaluation of parallel count sort using GPU computing with CUDA. *Indian Journal of Science and Technology.,* Apr, 9(15), 1–12.

Castro, M., Druschel, P., Hu, YC., and Rowstron, A. (2002) Topology aware routing in structured peer-to-peer overlay networks. *Technical Report MSR-TR-2002-82, Microsoft Research*., 1–19.

Jung, H., Lim, W., Hong, J., Hur, C., Lee, J., You, T., Eun, J., Kwak, B., Kim, J., Jeon, H., Kim, T., and Chun, W. (2015) IDNet: beyond all-IP network. *ETRI Journal*., Oct, 37(5), 833–44.

Jayavel, K., and Nagarajan, V. (2016) Survey of migration, integration and interconnection techniques of data centric networks to internet- towards Internet of Things (IoT). *Indian Journal of Science and Technology.*, Feb, 9(11), 1–7.