Journal of Network Computing and Applications (2016) 1: 1-6
Clausius Scientific Press, Canada

A Clean-Slate ID/Locator Split Architecture for Future Network

Zhu Shaotong[1, a]

[1]Signal and Communication Research institute, Academy of Railway Sciences
[a]zhushaotongscri@sina.com

Keywords: ID/locator split, ID architecture, locator architecture, Switching label, CILSA

Abstract: One of the most important problems of today's Internet is scalable routing. Proposed solutions for this issue are centered on the idea of splitting the network node's identifier (ID) from its topological location. In this paper we present the basic design of a clean-slate ID/locator split architecture (CILSA) for future network. Our design separates ID architecture from locator architecture. A new hierarchical locator based on port is first introduced and the ID/locator splitting happened in network layer, which is distributed realized by the architecture not depend on additional equipment. In order to show benefits of proposed scheme, we analyze solutions in our CILSA under connection setup scenario. Some evaluations are also invested for the proposed CILSA.

1. Introduction

In the conventional Internet architecture, IP addresses are used by the application and transport layers to identify communicating end nodes [1]. This causes problems with mobility and multi-homing management, network renumbering, security and privacy, scalable routing, and traffic engineering. Currently, there are some solutions identified by researchers who have been investigating methods for separating node identifiers from node locators, for example, HIP [2], HRA [3], Shim6 [4], I3 [5], LISP [6], GSE [7], MILSA [8], RANGI[9], etc. But all of them cannot provide a complete solution to the Internet challenges above mentioned.

In this paper we present a clean-slate ID/Locator split architecture (CILSA) for next generation Internet. Our goal is not only to decouple IP address by splitting ID/Locator, but also to design new network architecture in order to support mobility, multi-homing, management, network renumbering, security, privacy, scalable routing and traffic engineering better.

2. ID/Locator Split Architecture

2.1 Hierarchical Identifier Architecture

In our node ID architecture, we design a hierarchical ID to name the nodes in the network. The node's ID consists of two parts: the local name and the hierarchical logical name. The hierarchical logical name defines the organizational affiliation which logically locates the position of the node in the ID architecture. This part is purely about logical affiliation that is different from the DNS name.

In our hierarchical ID architecture, we combine the two parts into one ID. Please refer to the Fig. 1, the full ID of a node Mike logically belonging to peer group(PG) A is ”A. Mike”, in which the first part ”Mike” is the local name in the leaf of the ID architecture while the remaining part reflects its logical position in the ID architecture. If A. Mike wants to communicate with B. Tim, it may use its mask ”A” comparing with its destination’ mask ”B”. If matching is success, which demonstrates they are locate in same PG, otherwise, A. Mike sends this request to its father node ”A”. The ”A” repeats the matching until the matching is successful. This is a pull process. Then the node ”B” pushes this request to its child node ”Tim”.

The construction process of hierarchical ID architecture is: every node assigns its local name, father name and priority. Nodes which possess of same father name belong to same organization, that is to say they are in one PG. Then those nodes select a PG leader (PGL) according to priority, and this is a dynamic process, the black point denotes PGL in Fig. 1. The PGL has the group’s state and aggregate information of the next hierarchy and so on. Finally those nodes are organized into hierarchical tree-like architecture in logic.

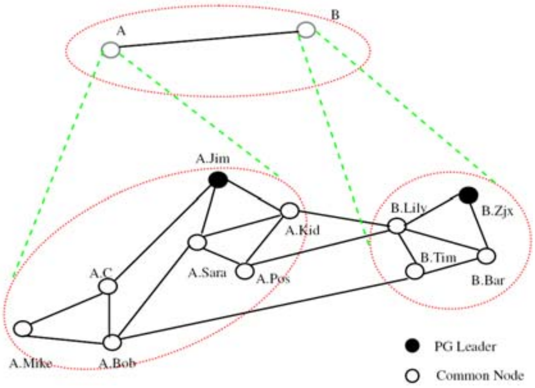

Fig. 1. Hierarchical identifier architecture

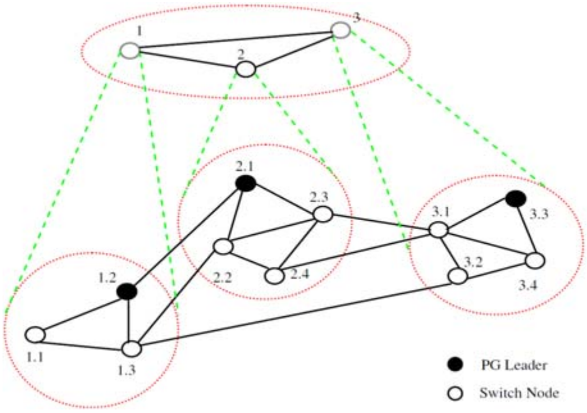

Fig. 2. Hierarchical locator architecture

2.2 Hierarchical locator architecture

The node’s locator consists of two parts: the rough part and the detailed part. The rough part indicates the node’s location and the detailed part points out the location for live ports of node. The difference between them is that the detailed part has smaller granular. This is very helpful to multi-paths routing.

In our hierarchical locator architecture, the rough part locator is produced after topology aggregation. Take Fig. 2 for example, the number of node and link and the connection correlation among them is same as the Fig. 1. The rough locator architecture is formed according to their physical topology. One of likely architecture is shown in Fig. 2 formulating two hierarchies, in which the first hierarchy is composed of three PGs and the second hierarchy is composed of one PG.

The construction process of rough locator architecture is similar to the ID architecture. The detailed locator architecture of PG 1 in Fig. 2 is shown in Fig. 3. The PG ”1.1” has two live ports connecting their neighbors, so it has two children. However, their locators are assigned by their neighbor nodes ”1.2” and ”1.3”. Therefore, the locator of their two children in PG ”1.1” is ”1.2.2” and ”1.3.4” respectively. If you want to locate the node ”1.1”, you can find ”1.1” easily according to their indication of locator ”1.2.2” and ”1.3.4”. It can ensure the two paths are separated radically and don’t need extra multi-paths routing algorithms. This is why we use port-based locator in our scheme.

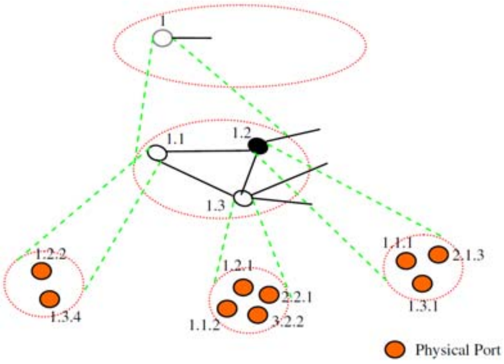

Fig. 3. Detailed locator architecture of PG 1

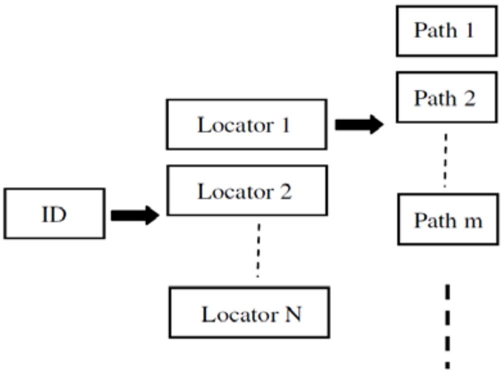

Fig. 4. Mapping relationship between ID and locator

2.3 ID/Locator Split

The mapping relationship between ID and locator is shown in Fig. 5. A node may have several live ports connecting its neighbors and every port has its own locator according to the definition of locator in CILSA, so one ID can corresponding to several locators and several paths can be found for one locator according to routing algorithm.

Different from the current mapping mechanism of DNS or other additional equipment based our scheme employ hierarchical tree-like architecture. We use push and pull hybrid method to traverse the tree. The mapping system is distributed that can work normally if there are some nodes or links failure, so the proposed scheme is robust for failure tolerance.

3. Connection Setup

Suppose node with ID ”A. Mike” wants to communicate with node with ID ”A. Bob”. In order to use the Fig. 1, Fig. 2 and Fig. 3 to explain this process we assume that both communication nodes in the same PG. The procedure of nodes in different PGs can be acquired by analogy in same PG.

The connection setup procedures can be described as follows:

(1) All nodes have their full ID after hierarchical ID architecture built and updated. That is to say the node ”Mike” with full ID ”A. Mike” and node ”Bob” with full ID ”A. Bob”.

(2) ”A. Mike” sends a request to its PGL ”A. Jim” to ask for access; the detailed accessing process is shown in Section II. If the accessing process is successful, ”A. Mike” can perform calling process and a location PW is assigned to it by ”A. Jim”, otherwise, ”A. Mike” is not allowed to access this network.

(3) ”A. Mike” sends a calling packet carried destination ID ”A. Bob” to its PGL ”A. Jim”. Through matching the two prefixes ”A. Jim” knows the two nodes in the same PG ”A”. Then ”A. Jim” forwards this packet to the node ”A. Bob”. If matching is unsuccessful, ”A. Jim” forwards this packet to its father PG ”A” until matching success.

(4) ”A. Bob” sends back its hierarchical locator ”1.2.1”, ”1.1.2”, ”2.2.1” and ”3.2.2” which are shown in Fig. 3 to ”A. Mike”.

(5) ”A. Mike” sends the location PW to its PGL ”1.2” in hierarchical locator architecture which is shown in Fig. 3 to complete admission. If successful, ”A. Mike” can perform routing process, or it is forbidden.

(6) ”A. Mike” sends a routing packet according to the strict indication of locator ”1.2.1”, ”1.1.2”, ”2.2.1” and ”3.2.2” to find destination node ”A. Bob”. The switching labels are collected gradually with the forwarding of routing packet. After the routing packet arrived at ”A. Bob”, ”A. Bob” sends back several switching labels which are collected in routing packet to ”A. Mike”.

It is obvious that the ID is only used in calling process and locator is only in routing process. The ID/locator split is distributed realized by the architecture, not like other schemes which all need additional equipment.

4. Evaluation

In this section, we evaluate the performance of the proposed CILSA. The CILSA is mainly composed of two hierarchical architectures: identifier architecture and locator architecture.

Assume a hierarchical locator architecture has L levels, and the average number of nodes in a PG to be m. Hence, the average number of nodes at level N would be m^N. We also assume the locator from source to destination would traverse d nodes in a PG. Thus, the length of the locator at level N would be d^N. According to [10], the number of edges in a m-node domain can be described approximately by $E = \frac{m}{2(R+1)}[1-(\frac{1}{m^{R+1}})]$. Here, we take R = -0.8, which is the average for the values of R given in [10]. As the number of domains at level N is m^{N-1}, the number of edges at level N can be derived by $E = 2.5 * m^N (1 - m^{-0.2})$.

Fig. 5 shows the relationship among the number of edges, number of levels and number of nodes within a domain. It is seen that when the number of nodes within a domain less than 30, the increasing curve for the number of edges is flat relatively. When the number of nodes becomes more than 30, the number of edges is increased sharply. Therefore, we require the number of node at a domain near 30 in CILSA.

To compare the hierarchical architecture to flat architecture, we assume the same number of nodes for two architectures. Here we consider the flat routing algorithm proposed in [11]. The message complexity is $O(E+2d^L)$, and the calculation time would be $O(d^L)$.

Fig. 6 show the calculating time for 3 levels locator architecture and flat architecture. It is seen that hierarchical architecture consumes more time than flat architecture. As the building process of locator architecture happens before the formal communication, the building time has less impact on the whole network. Once the locator architecture is built, both corresponding sides can find multi-paths easily using this architecture.

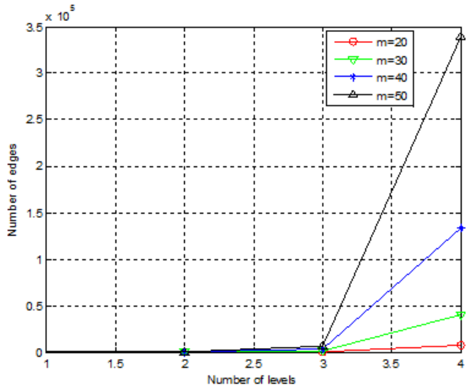

Fig. 5. The relationship among edges, levels and nodes

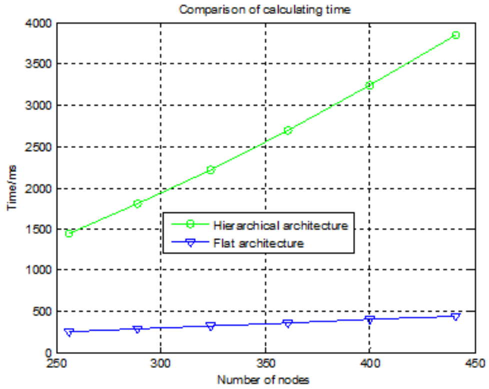

Fig. 6. Comparison of calculating time for hierarchical and flat architecture

5. Summary

In this paper, we propose a clean-slate ID/locator split architecture (CILSA) for future network. A port-based locator is first introduced and ID/locator split happened in network layer, which is distributed realized by ID architecture does not depend on additional equipment. In addition, we employ new switching label-Vector Address, which decreases the cost of building logic connection. We also analyze and discuss the scenario of connection setup in our CILSA. We have evaluated the performance of CILSA through analyzing the calculating time and the computational complexity of the locator architecture building process. For future work, we plan to simulate CILSA on OMNET [12] under the case of mobility and multi-homing, and evaluate its performance.

References

[1] Braden, R., “Requirements for Internet hosts communication layers,” RFC 1122, 1989.

[2] Moskowitz, R., Nikander, P., and Jokela, P., “Host identity protocol (HIP) architecture,” RFC 4423, 2006.

[3] Xu, X. And Guo, D., “Hierarchical routing architecture (HRA),” NGI, 2008.

[4] Nordmark, E. And Bagnulo, M., “Internet draft: Shim6: level 3 multihoming Shim protocol for IPv6,” draft-ietf-shim6-proto-09, 2007.

[5] Stoica, I., Adkins, D., et al, “Internet indirection infrastructure,” ACM SIGCOMM, 2002.

[6] Farinacci, D., Fuller, V., et al, “Internet draft: Locator/ID separation protocol (LISP),” draft-farinacci-LISP-03, 2007.

[7] O’Dell, M., “Internet draft: GSE - an alternate addressing architecture for IPv6,” draft-ietf-ipngwg-gseaddr-00, 1997.

[8] Pan, J., Paul, S., Jain, R., and Bowman, M., “MILSA: a mobility and multihoming supporting identifier locator split architecture for next generation Internet,” IEEE GLOBECOM, 2008.

[9] Xu, X. and Jain, R., “Routing architecture for the next generation Internet (RANGI),” draft-xu-rangi-00.txt, 2009.

[10] Faloutsos, M., Faloutsos, P., and Faloutsos, C., “On power-law relationships of the Internet topology,” ACM SIGCOMM, 1999.

[11] Sarangan, V., Ghosh, D., and Acharya, R., “Distributed QoS routing for multimedia traffic,” IEEE GLOBECOM, 2000.

[12] OMNET++ simulator, available at: http://www.omnetpp.org.