Distributed Encoding for Multiple-inherited Locators to Accommodate Billions of Objects in the Internet

Xiaozhe Shao[1], Feng Wang[2], Lixin Gao[1], Kenji Fujikawa[3], and Hiroaki Harai[3]

[1]Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003, USA
[2]School of Engineering and Computational Science, Liberty University, Lynchburg, VA 24515, USA
[3]National Institute of Information and Communications Technology, Tokyo 184-8795, Japan

***Abstract*—As the Internet of Things technologies evolve, billions of smart devices will be connected to the Internet. Therefore, the accelerated growth of users, applications and devices pose a great demand on the scalability of the Internet. In this paper, we develop a new Internet architecture – Multiple-inherited Locators (MiL) – to meet the future demand on addressing. MiL is based on Locator/ID addressing and hierarchical address allocation. The benefit of this new addressing scheme is the improved scalability of Internet routing by enhancing the prefix aggregation of locators.**

The number of locators which are needed grows with the Internet scale. In order to represent each locator as a unique binary representation by as few bits as possible, we develop a distributed encoding method to efficiently encode locators. As the Internet topology evolves, the distributed encoding method renumbering locators' codeword for the optimal encoding performance, and meanwhile endeavors to keep their codewords unchanged as much as possible. Because, we want to represent locators by fewer bits as well as keep addressing stable. We adopt an Encoding Table Adjustment algorithm to find a "sweet spot" that balances these two goals. According to our experiment, within a 2% increase on the expected locator length, our algorithm reduces the cost of renumbering locator by 99.9%.

I. Problem Statement

With the growth of the Internet of Things (IoT), both servers and devices increase superlinearly in the current Internet. The rapid growth of the scale leads to the scalability issue of the Internet. Prefix aggregation technology held up the growth of the BGP table. However, with the IPv4 address exhaustion, IP addresses are allocated by even smaller segment which further increases the BGP table size. Moreover, more and more Autonomous Systems (ASes) are multi-homed to multiple providers to avoid a single point failure. When a multi-homed AS advertises its prefixes to its providers, only one provider can aggregate the prefixes, which severely limits the extent of the address aggregation.

In order to enable prefix aggregation with multi-homed AS we first design a new addressing scheme, Multiple-inherited Locators (MiL). In this addressing scheme, each host is assigned a global unique address, which consists of one or several locators and a host identifier. Each locator indicates how the network in which hosts reside, is connected to the Internet. At the meantime, a host identifier identifies the host within the network.

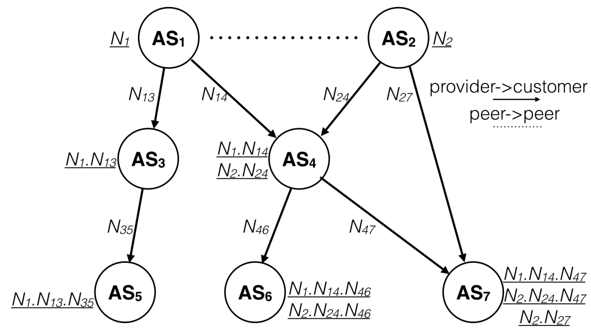

Fig. 1. An example of MiL assignment.

A. Hierarchical Locator Allocation

We first present an overview of hierarchical addressing, which is extended from our previous work [1]–[3]. Locator assignment follows the inter-domain hierarchy where each AS obtains a locator from a provider in the upper layer. Locators are labeled by a prefix-based labeling scheme. Each locator consists of a provider's location label followed by a *customer label*. A provider's location label identifies the provider, and a customer label is to identify a customer. We use a delimiter '.' to join a provider's location label and a customer label. Thus, a locator is a concatenation of a provider's locator and a customer label. In Fig. 1, the label beside the arrow line represents a customer label assigned by a provider to a customer. AS_1 and AS_2 are Tier-1 ASes, and their locators are N_1 and N_2, respectively. AS_1 assigns customer labels N_{13} and N_{14} to AS_3 and AS_4, respectively.

B. Challenges

The main challenge of implementing MiL is the overhead incurred in the packet header. Because we augment the packet header with a list of locators, which potentially make the packet header longer. To tackle the challenge, we leverage a variable-length encoding scheme to assign shorter locators to the ASes with higher possibility of occurring in packets, and longer locators to the ASes with lower possibility. AS a result, we reduce the expected locator length in packet headers.

However, the performance of variable-length encoding relies on the distribution of the locator's frequency. The Internet topology change affects the distribution which in turn leading to renumbering locators for keeping minimized expected length. During the renumbering procedure, we also want as much as locators to keep their previous codewords, since updating locators incurs overhead and stability issue on the Internet. Therefore, we adopt Encoding Table Adjustment for the renumbering procedure to reach the two goals. However, it is difficult to achieve the two goals simultaneously. They are relevant to each other. Minimizing the expected length of customer labels could conflict with the goal of reducing renumbering cost.

II. Our Solution

A. Distributed Locator Encoding

In our hierarchical addressing, a customer label is the proper encoding unit. The reason is listed as follows. A provider's locators are the prefix of its customers' locators, which keeps customers' locators aggregated by provider's locators. If a locator or a set of locators are encoded as a whole, it is hard to guarantee the aggregation property. Therefore, we encode each label independently and concatenate them to build the locator's codeword. As a result, the codeword of a provider's locator is the prefix of the codeword of its customers' locators.

In the proposed hierarchical addressing, every AS can possess its own encoder which performs a specific encoding scheme. In other words, a centralized encoder is not required. We adopt Huffman coding to generates a codeword for each label based on its frequency. Here, we define the frequency of a label as the probability of the label appearing in all packet headers.

B. Encoding Table Adjustment

The goal of ETA is to minimize the updating number within a bound lose of the encoding performance. The basic idea of ETA is to gradually build a final encoding table (a Huffman tree) by adjusting the original encoding table. For each adjustment operation, we greedily reduce the updating number. We propose to adjust the codeword of customer labels with higher frequency. The reason is that it makes larger reduction on the updating number comparing to the adjustment on a customer label with lower frequency.

III. Evaluation

We evaluate the performance of our distributed encoding mechanism and the ETA algorithm. The dataset of these measurement is based on the AS topology in the Internet which comes from the AS relationships dataset of CAIDA [4].

We compare the performance of the fixed-length encoding or the variable-length encoding, namely Huffman coding. As Fig. 2 shows, Huffman coding overwhelms fixed-length coding all the time at the last decade. The expected length of label is reduced by nearly 50%.

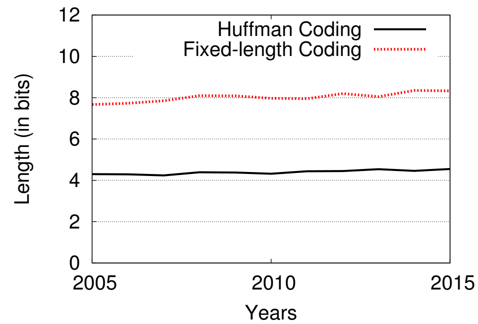

Fig. 2. The expected length of labels with fixed-length coding and Huffman coding.

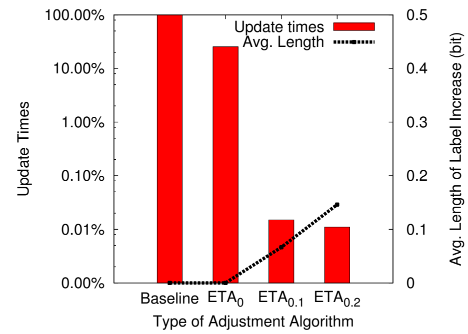

Fig. 3. Comparisons of Huffman Tree Adjustment Algorithms.

With the former Internet topology as the start point, we add or remove an AS relationship following the specific evolution path. During the process, we perform the ETA algorithm at every provider AS to adjust locators for each topology change. Fig. 3 shows the reduction of updating number and the increase of the expected codeword length with different α. "Baseline" refers to no adjustment algorithm, namely we directly adopt Huffman coding to generate the final encoding table without any adjustment. "ETA$_i$" refers to the Huffman Tree Adjustment algorithm with $\alpha = i$. It means that after the adjustment, the expected length of codewords increases at most α. The ETA$_{0.1}$ reduces the address updates by more than 99.9%. However, 0.1 only accounts for about 2% of the expected label length.

Acknowledgment

This work is supported by NSF grant CNS-1402594.

References

[1] K. Fujikawa, A. H. A. Muktadir, Y. Fukushima, H. Harai, X. Shao, F. Wang, and L. Gao, "Design and Implementation of Variable-Length Locator Allocation Protocol for Scalable Internet Addressing," in *APCC*, October 2015.

[2] F. Wang, L. Gao, S. Xiaozhe, H. Harai, and K. Fujikawa, "Compact Location Encodings for Scalable Internet Routing," in *IEEE INFOCOM*, 2015.

[3] F. Kenji, H. Hiroaki, and O. Masataka, "The basic procedures of hierarchical automatic locator number allocation protocol hana," in *Proceedings of the 7th Asian Internet Engineering Conference*, ser. AINTEC '11. New York, NY, USA: ACM, 2011, pp. 124–131. [Online]. Available: http://doi.acm.org/10.1145/2089016.2089037

[4] The caida as relationships dataset, 2000-2015. [Online]. Available: http://www.caida.org/data/as-relationships/