SAIL: A Scalable Approach for Wide-Area IP Mobility

Zhenkai Zhu
Computer Science Department
Univeristy of California, Los Angeles
zhenkai@cs.ucla.edu

Ryuji Wakikawa
Toyota Infotechnology Center
Mountain View, CA
ryuji@us.toyota-itc.com

Lixia Zhang
Computer Science Department
University of California, Los Angeles
lixia@cs.ucla.edu

***Abstract*—The Internet is becoming increasingly mobile, with not only smartphones outnumbering stationary hosts, but also cars, buses, trains and airplanes all coming online. This makes Internet mobility support more important than ever. However the existing standard mobility support protocols, mainly Mobile IPv6 and Network Mobility (NEMO), suffer from triangle routing problem. In this paper we present a new mobility support protocol called SAIL that provides an effective and efficient solution to the triangle routing problem while being completely compatible with Mobile IP. SAIL is built upon the Global HAHA protocol which uses multiple distributed home agents to minimizes triangle routing, but removes its high overhead by one-hop DHT. We evaluate the SAIL design through extensive simulations and our results show that SAIL can provide superior performance over Mobile IP while keeping the overhead low.**

I. Introduction

This paper concerns effective and efficient mobility support in the global Internet. In the past few years the world has witnessed a rapid growth of mobile computing devices in a global scale. Today there are probably more people who access Internet via mobile devices of one type or another, such as laptops, tablets, and smartphones, than the number of users on stationary hosts with wired connectivity. This trend is likely to accelerate in coming years with wide deployment of various new mobility services such as 4G cellular networks and vehicular networking (Intelligent Transport Systems), making it increasingly important for Internet to provide effective and efficient mobility support.

Towards that goal IETF has standardized Mobile IPv6 [1], NEMO [2] protocols and their extensions as the base systems for mobility services in WiMAX, 3GPP, and 3GPP2. Mobile IPv6 enables a mobile node (mobile in short) to be reached through a stable IP address regardless of where and how it may be connected to Internet, with the help of a *home agent*. Unfortunately, the use of a home agent introduces a well known triangle routing problem. That is, all data packets destined to the mobile are routed towards the home agent first, and then the home agent forwards the packets to the mobile through encapsulation, resulting in a non-optimal data path if the home agent is off the shortest path between the mobile and correspondent nodes. NEMO is a simple extension of Mobile IPv6 to support mobile networks, and thus inherited the same triangle routing problem. Although solutions to this problem have been proposed for both Mobile IP and NEMO [1] [3], the resulting designs are complex with various constraints that can be a hurdle to their deployment.

Wakikawa *et al.* proposed Global HAHA [6] that uses Mobile IPv6 to support Internet-Scale mobility without the triangle routing problem. Generally speaking, the triangle routing problem diminishes if the home agent is near the mobile all the time. Global HAHA approximates this ideal situation by distributing multiple home agents widely, and allows a mobile to use the nearest home agent node as it moves. This approach effectively minimizes triangle routing and is much simpler than the proposed route optimization solution [1]. However Global HAHA has a relatively high overhead cost as it floods all mobile nodes' location change information to all the home agents.

In this paper we present SAIL which revises the Global HAHA design to remove its high overhead. As in Global HAHA, SAIL uses a set of home agents that are distributed over large geographic areas, all of them announcing the same IPv6 prefix (mobile prefix) to form an anycast group. Different from Global HAHA, SAIL eliminates the broadcast overhead in Global HAHA through the use of a simple and robust one-hop DHT. Furthermore, SAIL requires no modification to mobiles or correspondents; the only changes are made on the home agents. Hence SAIL can be deployed incrementally, and the mobile nodes can immediately benefit from the deployment without waiting for all SAIL home agents being installed. We evaluated the design of SAIL through extensive simulations and our results show that in most cases SAIL achieves better or equal end-to-end delay as Global HAHA, but with orders of magnitude smaller control overhead.

Our contributions in this paper can be summarized as follows. First, we present SAIL as an effective and efficient solution to address Mobile IP's triangle routing problem. Second, our simulation evaluation demonstrates SAIL's effectiveness. Finally, our design and evaluation efforts shed new lights on the design principles and tradeoffs in mobility support.

The remainder of this paper is organized as follows. Section II reviews previous works. Section III describes the concept and protocol operations of SAIL. We present experiment results in Section IV, discuss our findings in Section V, and conclude the paper in Section VI.

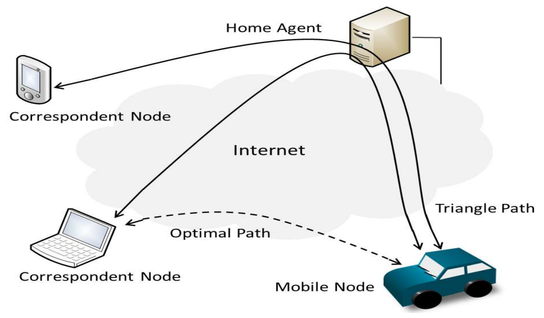


Fig. 1: Mobile IPv6 Overview

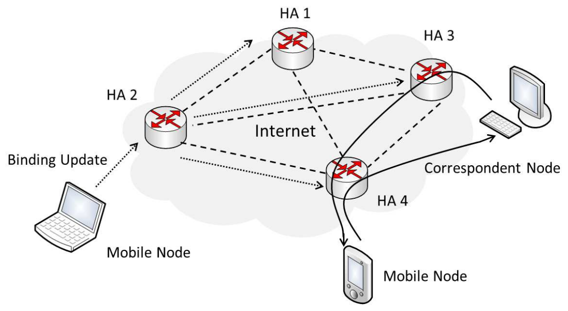


Fig. 2: Global HAHA Protocol

II. Background

In this section we first briefly introduce the Mobile IPv6 protocol and analyze its limitations. We then provide an overview of related works.

A. *Mobile IPv6*

Each mobile node M has a home agent, from which M acquires its home address (HoA), the address used by all correspondent nodes to reach M. The home agent is a special router at M's home network, designated to forward M's traffic to its current location. M also obtains a care-of address (CoA) from its current access router. Whenever the mobile node moves and gets a new CoA, it sends a binding update message to notify the home agent. The home agent acknowledges the binding and sets up an IPv6-in-IPv6 tunnel with the mobile. All the communications between a mobile and correspondent nodes go through this tunnel. Figure 1 shows the overview of Mobile IPv6 protocol.

B. *Route Optimization for Mobile IPv6*

There have been several proposed solutions to address Mobile IP's triangle routing problem.

1) Return Routability Procedure: Mobile IPv6 standard comes with a route optimization scheme called Return Routability Procedure, which allows the mobile to send binding updates to its correspondent nodes, if they also support Mobile IPv6, in addition to its home agent. With this procedure, packets flow directly between the mobile and its corresponding nodes. Unfortunately this Return Routability Procedure also brings its own issues. First, it introduces additional complexity and latency for handoff: the mobile node must exchange four messages to generate a key that will be used to authenticate the binding between HoA and CoA. This binding is sent to every correspondent node every time the mobile moves. Worse yet, if the mobile cannot successfully perform the return routability test (e.g. perhaps due to firewall policy), it will not be able to communicate with the correspondent nodes until the old bindings at the correspondent nodes expire. Furthermore, route optimization also raises privacy concerns, as it reveals mobile's CoA to all correspondent nodes.

2) Global HAHA: Global HAHA [6] aims to minimize the triangle routing problem in Mobile IP and NEMO by distributed home agents. Distributed home agents also remove the concern of a single home link becoming a bottleneck and eliminate the vulnerability due to a single home agent. While the main scenario is a mobile node roaming over large geographical regions, with home agents distributed in large cities to provide optimal data paths, Global HAHA is also beneficial even in small geographical areas, for example the (topologically) closest home agent to a mobile node may be different when the mobile switches between different access technologies, such as WiFi, WiMAX, 3G, etc.

All the home agents in Global HAHA announce the same home prefix from their different locations, forming an anycast group. They also interconnect with each other to form a mesh overlay network. A mobile node M sends a binding request to the anycast address. This request will be received by the home agent H that is closest to M, and H becomes M's primary home agent. H then notifies all other home agents of the binding [M, H], so that the binding information databases for all the mobiles in all the home agents are always synchronized. When a mobile moves, it may switch its primary home agent to another one that becomes closest to the mobile. A correspondent node sends packets to a mobile M's home address. Because of anycast routing, the packets are received by the home agent H_c which is closest to the correspondent. H_c encapsulates the packets to the IP address of M's primary home agent, which will finally deliver the packets to M. In the reverse direction, this approach works exactly the same as Mobile IP. Figure 2 illustrates the Global HAHA protocol. If the home agents are distributed widely, the triangle routing problem is naturally avoided without Route Optimization.

C. *Other Related Works*

There are also research works on dynamic assignment of home agents in order to provide better data paths [7], [8]. In [8], multiple home agents are set up in an Autonomous System (AS), and each home agent is assigned a priority to help the mobile node to associate with a closer home agent. However, in order to set up the priority list it requires that the mobility pattern of the mobile nodes be known beforehand. In comparison, SAIL enables a mobile to automatically select the best home agent through anycast routing.

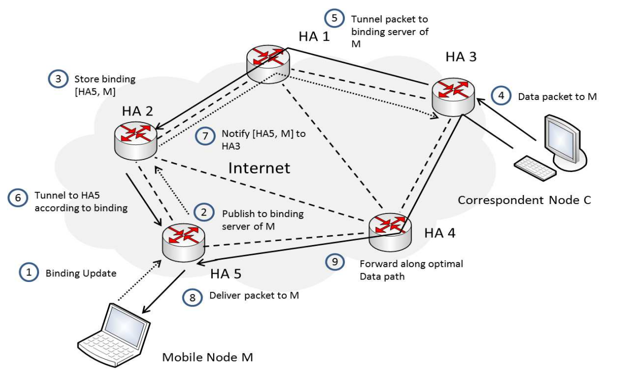


Fig. 3: Deliver Packets in SAIL

There are a few end-to-end mobility solutions such as HIP [4] and BTMM [5]. They put the latest CoA information into DNS, and thus eliminate triangle routing entirely. However this class of solutions requires prompt updates on DNS database, as well as changes to all hosts, both mobiles and correspondents.

There are also a number of works which apply DHT in mobility management [10] [11] etc. However, DHT is used for discovering the closest home agent to the mobile in these works, while in SAIL this is achieved by anycast and the one-hop DHT is used to distribute the location information about the mobile among the home agents. There are also works that utilize DHT to manage mobility in the application layer [12]. Our paper focuses on improving the data path and scalability of IP mobility.

III. SAIL Protocol

Among the proposed solutions of the triangle routing problem, Global HAHA seems most promising as it requires no modifications to end hosts and is incrementally deployable. However its control overhead is rather high compared to Mobile IP: When a mobile's movement leads to the change of its primary home agent, Global HAHA must notify all the home agents of this change, a cost that grows with the number of distributed home agents. This leads us into a dilemma: one wishes to increase the number of home agents to both minimize triangle routing and be able to serve more end-hosts, while at the same time one must restrict the number of home agents to keep the control overhead below acceptable threshold.

SAIL resolves the above dilemma by providing a distributed directory service using one-hop DHT. In Global HAHA, the binding information between a mobile and its primary home agent is stored at every home agent; in SAIL, this binding information is stored at one specific home agent selected by DHT. One-hop DHT is chosen here as it has low lookup delay and "free" load balancing [9]. We describe SAIL's design and operation below.

A. Mapping Bindings to home agents

The binding between a mobile M and its primary home agent H is expressed in a form of $[key = k, value = v]$ pair, where the key is M's home address and v is H's unicast IP address. Each home agent is identified by its unicast IP address, and knows the complete list of all home agents identifiers. H published this binding via directory service using a hash function $\mathcal{F}$ to map k to a home agent with identifier $\mathcal{F}(k) = b_k$. Home agent b_k, also referred to as the *binding server* for k, then stores the binding $[k, v]$. When another home agent H_c receives packets with destination address k, it uses the same hash function to find out which home agent is k's binding server and retrieves the value associated with k. Home agent H_c then sends a lookup request to b_k to retrieve the value v. H_c may also cache the lookup result in case additional packets to k arrive shortly.

When some home agents fail or recover, it leads to changes to the available home agent set, thus a set of keys must be re-hashed to different binding servers. In order to reduce the overhead and disruption, SAIL uses consistent hashing for $\mathcal{F}$. Formally, given a set $S = H_1, H_2, ..., H_n$ of home agents, and a key k,

$$\mathcal{F}(k) = min_{\forall H_i \in S}\{\mathcal{D}(\mathcal{H}(k), \mathcal{H}(H_i))\} \tag{1}$$

where $\mathcal{H}$ is a regular hash function, and $\mathcal{D}(x, y)$ is a metric function to compute the distance from x to y on the circular hash-space of $\mathcal{H}$. That is, $\mathcal{F}$ maps a key to the home agent with the closest hash result not exceeding the result of the key on the hash space of $\mathcal{H}$.

B. Traffic-driven Binding Resolution

When a node M with home address HoA_m sends a binding request and this request is received by home agent H_p, H_p becomes the primary home agent for M. H_p keeps a local binding between M's home address and care-of address (the same as Mobile IPv6 does) and publishes the binding $[HoA_m, H_p]$ to the directory service. H_p first determines the binding server for M by applying hash function $\mathcal{F}(HoA_m) = H_b$, and then sends a publish message instructing H_b to store the binding $[HoA_m, H_p]$. H_b acknowledges the publisher after storing the binding.

When a correspondent node C sends data packets to M, C uses HoA_m as the destination IP address. Now the packet will be routed to a home agent H_c that is closest to C. H_c computes $\mathcal{F}(HoA_m) = H_b$ and tunnels the packet to H_b. Home agent H_b then looks up the primary home agent of M and tunnels the packet to H_p, which finally deliver the packets to M's care-of address. However, this data path, traveling through the binding server, may not necessarily be the optimal path. Hence, the binding server H_b also notifies H_c about M's current location by replying the binding $[HoA_m, H_p]$ to H_c, which locally caches this binding. Thus, while the first few packets traverse a possible longer data path, the following data packets flow through an optimal data path, bypassing the binding server, until the cache expires or M changes its primary home agent. Figure 3 illustrates above steps involved in data packet delivery.

C. Handling host mobility

There are two types of host mobility in SAIL: local mobility where a mobile M's movement leads to a change of its care-of address but not its primary home agent, and global mobility where M's movement leads to a change of its primary home agent. In the first case, only the primary home agent needs to update its local binding between M's home address and its CoA; no other home agents need to be aware of M's location change. In the second case, M's periodic binding request will reach the new closest home agent H_p^{new} first and then arrives at H_p^{old}, which indicates H_p^{old} is no longer the closest home agent to M. H_p^{old} then instructs M to register at H_p^{new} by sending a home agent switch message [13] to M and deletes the local binding after receiving the acknowledgment from M. M then sends another binding request to register at H_p^{new}. After it accepts M's binding request, H_p^{new} publishes the new binding to the binding server as described above.

D. Reactive Cache Update

When a mobile M moves, the cached binding information of M to its primary home agent at other home agents are not updated. Thus it is highly possible that H_p^{old} may receive packets for M after the information at the binding server H_b has been updated. SAIL takes advantage of this fact to update such stale caches. H_p^{old} caches the new binding $[HoA_m, H_p^{new}]$ for a short time period after the mobile node leaves; when it receives packets destined to M from other home agents (due to stale caches), it explicitly notifies them about the new binding for M. To minimize data flow disruption, H_p^{old} also forwards these misrouted packets to H_p^{new}.

IV. Evaluation

We evaluated SAIL's performance by extensive simulations. In this section we first describe the simulation environment, then report SAIL's performance under different settings.

A. Evaluation Methodology

We implemented both Global HAHA and SAIL in Qualnet simulation platform [14] which has built-in Mobile IP module. We simulate a set of 64 home agents evenly distributed over a 1600m x 1600m area with a grid topology. Note that the exact shape of the topology does not affect the comparative results among the above mentioned three protocols.

We use the VanetMobiSim [15] package developed by EURECOM to generate both node movements and random maps. The mobility speed varies from 0 km/hr to 54km/hr, and the mobile node speeds up or slows down smoothly.

Each simulation run executes for 1800 seconds of simulated time and the mobile is communicating with the correspondent during the last 1600 seconds. Multiple runs are conducted for each scenario. When we plot the average we also indicate the confidence intervals.

The metrics used in SAIL evaluation are:

- **Relative Control Overhead:** instead of counting the number of control message, we treat the control overhead of Mobile IPv6 as one and adjust the values of control overhead for Global HAHA and SAIL as compared to Mobile IPv6.
- **Relative End-to-End Delay:** similar to the above metric, we treat the delay of Mobile IPv6 as one and adjust the delays for Global HAHA and SAIL.

We use relative overhead and delay to evaluate the performance of a protocol, because the absolute values of control overhead and delay are closely related to the mobility of the mobile node. When we use different mobility files for the same scenario, we may get results with large differences regardless of which protocol we use. However, for each scenario, the ratio of the control overhead of Global HAHA or SAIL to the control overhead of Mobile IPv6 stays more or less the same, as the mobility of the mobile node affects these three protocols equally. The same argument also works for the delay.

Given a mobility scenario, there are two major factors that affect the comparative results among Mobile IPv6, Global HAHA and SAIL: traffic patterns and home agent locations. SAIL relies on traffic-driven home agent binding, hence is sensitive to traffic patterns; Mobile IPv6 on the other hand is sensitive to home agent locations. We show their impacts below.

B. Experimental Results

1) Impact of Traffic Pattern: To evaluate the performance of SAIL under various traffic patterns, we simulate the communication between a correspondent node and a mobile with different time intervals between data packets. The nearest home agent of the correspondent node is at the bottom right corner of the simulation area and the initial home agent for the mobile node is at the upper left corner. For each mean time interval value, we repeat 30 runs with different mobility files.

Figure 4a shows the relative control overhead of Mobile IPv6, Global HAHA and SAIL with 90% confidence interval. Overall, Mobile IPv6 incurs the least control overhead. Although both using distributed home agents, SAIL's control overhead is more than an oder of magnitude lower than that of Global HAHA, which is not surprising. Because Global HAHA needs to notify all home agents for all mobiles' change of their primary agents, hence if there are N home agents and M mobiles changing primary home agents per second, then the overhead of binding updates in Global HAHA is $2 \times (N-1) \times M$ packets/s; in SAIL it is $2 \times M$ packets/s. The overhead of SAIL is higher when the correspondent node sends packets at a higher rate and becomes lower when the interval between packets becomes larger, because SAIL incurs overhead only when the mobile is communicating, thus the number of resolutions required drops with data rate.

Figure 4b shows the end-to-end packet delivery delay. Both Global HAHA and SAIL achieve much smaller delay compared to Mobile IP. The delay of SAIL goes up as the interval between packets becomes larger, due to the fact that caching is timed out already when next packet comes. When the inter-packet gap is greater than the cache timer, all the packets traverse via the binding server. Nevertheless, cache helps a lot in most cases and on average SAIL performs much better than Mobile IPv6 in terms of end-to-end delay.

2) Home Agent Location: We fix the correspondent node at the bottom right corner and randomly assign an initial home agent for the mobile node to see how does this impact the performance comparisons of the three protocols. We simulated scenarios with both short interval (0.1 second) and long interval (50 second) between data packets. To provide a statistically meaningful result, we repeat 200 runs for each scenario with different mobility files.

Figure 5b shows the cumulative distribution curve for the relative delay. In the short interval scenario SAIL always outperforms Mobile IPv6. In the long interval scenario, the performance of SAIL and Mobile IPv6 is more or less the same, this is because the next data packet comes after the cache timer expires, thus all packets flow through the binding server, which is equivalent to all packets going through home agent in Mobile IPv6. However, real applications rarely generate such traffic pattern (the mobile receives no more than one packet between long distance movements), so we expect SAIL to outperform Mobile IPv6 in most real world scenarios.

Global HAHA demonstrates similar performance as SAIL in the short interval scenario. It maintains the same performance in the long interval scenario by paying the high control overhead indicated in Figure 5a, which is an order of magnitude greater compared to SAIL.

V. Discussions

The DHT-based nature of SAIL requires that all the home agents keep a consistent view of the home agents set for consistent hashing results. However a home agent may fail or get disconnected at any time. Thus SAIL must have effective means to ensure that every home agent sees a consistent list of the active home agents in the face of network dynamics.

A. Detecting Changes of Home Agents Set

If all the home agents are within the same AS, one may detect the failure or recovery of a home agent from the information provided by underlying routing protocols (e.g. OSPF link state advertisement). Otherwise, we propose a solution of periodically distributing a list of reachable home agents (RHAs) to all the home agents. Inspired by the design in [16], we choose a small set of the home agents to be configured as candidate leaders (CLs), each with a unique leader-priority value. This list of CLs should be chosen wisely so that at least one of the candidate leaders is reachable under all conditions with a high probability. We leverage the existing DNS system to store the list of CLs. When a home agent bootstraps, it queries the DNS to obtain the CLs, and periodically sends "Hello" message to leader home agent with the highest priority. The leader home agent replies the "Hello" message with a message indicating the current RHA list.

The leader home agent H_L detects failures of any other home agents if it does not hear any of them for three consecutive Hello intervals. Vice versa, if a home agent does not get replies for three consecutive "Hello" messages, H_L is marked as unreachable. When the leader home agent fails, all other home agents send "Hello" to the CL with second highest priority, which takes on the leader role. When H_L recovers, it sends "Hello" to the current leader, and the latter will add the former to the RHA list and propagate the change to all other home agents. Meanwhile, the current leader also resigns from the leader position. To ensure that this change is propagated to all the home agents, it continues to respond to "Hello" message for three Hello intervals after its resignation. This centralized home agent list distribution can greatly reduce the possibility of inconsistency, while the simple but effective leader election provides robustness in face of network changes and failures.

Although the above procedure incurs certain amount of overhead, it accounts only for a negligible portion of the total overhead in supporting a large number of mobiles (which includes binding resolutions, binding updates, etc.). An analogy would be the overhead of routing protocol in IP networks: the networks support so many communications among numerous end hosts that we usually ignore the routing overhead entirely.

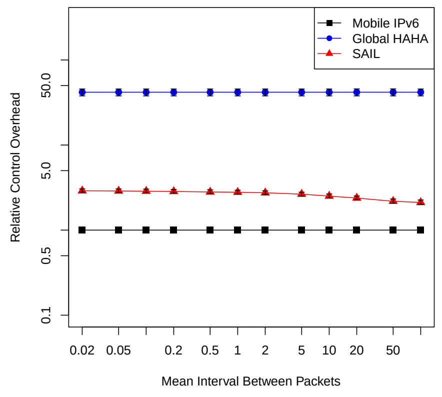
B. Responding to Changes of Home Agents Set

To prepare SAIL for wide deployment we must consider various network failure scenarios. If network failures do not partition the network, the set of home agents does not change, and SAIL is not affected by such topology changes. However, if a home agent fails or recovers, or the network is partitioned, the binding information stored on the failed home agents must be transferred to other agents. This is done by requiring the home agents who published bindings to monitor the status of corresponding binding servers on the current RHA list, and to re-publish bindings to other servers if any failure is detected. For example, when a home agent learns the failure of another home agent H_f from the RHA list, it scans its published bindings to find those that are previously published to H_f, computes the new binding server H_b^{new}, and then publishes them to H_b^{new}. It also scans the caches and removes those whose value field is H_f. Similar actions need to be taken when a previously failed home agent recovers, except that there is no need to scan caches.

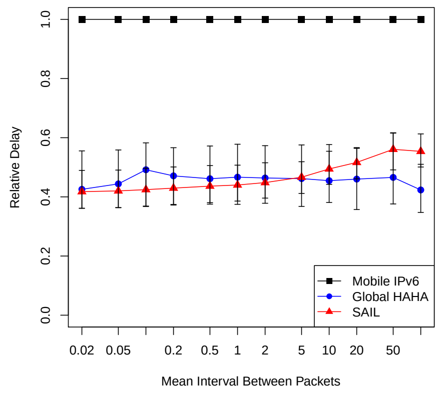
VI. Conclusion

In this paper we presented SAIL for supporting wide-area IP mobility. SAIL builds upon Global HAHA's use of multiple distributed home agents and added to it a distributed binding management with one-hop DHT to be a simple and efficient solution for the triangle routing problem in Mobile IPv6; it also improves home agent scalability and availability in Mobile IPv6. Compared to Global HAHA, SAIL's overhead is orders of magnitude smaller while achieving equal or even better end-to-end delay in most cases. Even under the (unlikely) worst traffic conditions, SAIL's performance is no worse than that of Mobile IPv6. Furthermore, SAIL is incrementally deployable as well as directly applicable to NEMO, a network mobility extension of Mobile IPv6.

In the process of designing SAIL we encountered a number of tradeoff decisions. As the next step we would like to eval-

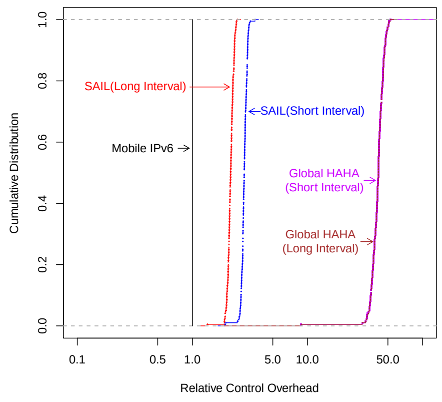


(a) Control Overhead

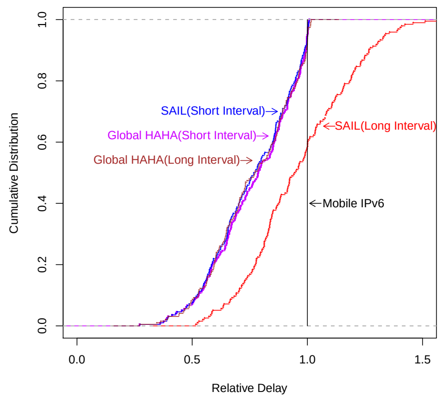


(b) Packet Delivery Delay

Fig. 4: Performance Comparisons Under Different Traffic Patterns



(a) Control Overhead



(b) Packet Delivery Delay

Fig. 5: Cumulative Distributions in Different Scenarios

uate our design in more complex settings to further quantify SAIL's performance and verify its design, and to move SAIL towards adoption by operational networks.

REFERENCES

[1] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", *RFC 3775*, 2004.

[2] V. Dearapalli, R. Wakikawa, A. Peterson, R. Thubert, "Network Mobility (NEMO) Basic Support Protocol", *RFC 3963*, 2005.

[3] S. Ayaz, C. Bauer, F. Arnal, "Minimizing end-to-end delay in Global HAHA networks considering aeronautical scenarios", *7th ACM international symposium on mobility management and wireless access*, 2009.

[4] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, "Host Identity Protocol", *RFC 5201*, 2008.

[5] *www.apple.com/mobileme/features/mac.html*

[6] Ryuji Wakikawa, Guilaume Valadon, Jun Murai, "Migrating home agents towards internet-scale mobility deployments", *ACM CoNEXT*, 2006.

[7] S. Mtika, F. Takawira, "Mobile IPv6 Regional Mobility Management", *4th Int. Symp. on Information and Communication technologies*, 2005.

[8] W. D. Ivancic, D. Stewart, T. L. Bell, P. E. Paulsen, D. Shell, "Use of Mobile-IP Priority Home Agents for Aeronautics Space Operations and Military Applications", *IEEE Aerospace Conference*, 2004.

[9] "Floodless in SEATTLE: A Scalable Ethernet Architecture for Large Enterprise", *SIGCOMM'08*, 2008.

[10] R. Cuevas, A. Cabellos-Aparicio, A. Cuevas, J. Domingo-Pascual, A. Azcorra, "fP2P-HN: A P2P-based route optimization architecture for mobile IP-based community newtorks", *Computer Networks*, vol 53, issue 4, 2009.

[11] R. Cuevas, C. Guerrero, A. Cuevas, M. Calderon, C. J. Bernardos, "P2P based architecture for global home agent dynamic discovery in IP mobility", *65 th IEEE Vehicular Technology Conference*, 2007.

[12] S. Park, K. Park, Y. Choi, "SAMP: scalable application-layer mobility protocol", *IEEE Communications Magazine*, June 2006.

[13] B. Haley, V. Devarapalli, H. Deng, J. Kempf, "Mobility Header Home Agent Switch Message", *RFC 5142*, 2008

[14] *http://www.scalable-networks.com*

[15] J. Harri, M. Fiore, F. Fethi, C. Bonnet, "VanetMobiSim: generating realistic mobility patterns for VANETs", *3rd ACM International Workshop on Vehicular Ad Hoc Networks*, 2006.

[16] Deborah Estrin, Mark Handley, Ahmed Helmy, Polly Huang, David Thaler, "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing", *IEEE INFOCOM '99*, 1999.