Fast Inter-domain Mobility with In-packet Bloom Filters

Mikko Särelä
Ericsson Research,
NomadicLab, Finland
mikko.sarela@ericsson.com

Jörg Ott
Aalto University, School of
Science and Technology
jo@netlab.tkk.fi

Jukka Ylitalo
Ericsson Research,
NomadicLab, Finland
jukka.ylitalo@ericsson.com

ABSTRACT

We propose a fast inter-domain mobility signaling protocol using in-packet Bloom filters. The intermediate routers collect a bi-directional Bloom filter on the first message, and on subsequent mobility signaling messages. The Bloom filter describes the path from the sender to the receiver and is used to forward the subsequent data packets in the session. In the case of single mobile node, a single message is sufficient to prove authenticity and return routability. For dual mobility scenarios, return routability tests can be delayed until after restarting communications. The protocol also makes bicasting simple, requiring the sender to simply bitwise OR the two Bloom filters describing paths to the old and new locations.

General Terms

Design, Security

Keywords

Mobility, Bicasting, Packet Forwarding, Bloom filter

1. INTRODUCTION

The desire to enable fast handovers for mobile IP nodes within and across link layer technologies and service providers has stirred a lot of research and engineering work over the past 15 years (see e.g., [13] for overview). Increasing number of devices utilize numerous access technologies, such as ethernet, WLAN, and 3G, requiring mobility support across service providers. Ideally, *make-before-break* approaches should not lose or (noticeably) delay a single packet so that the handover appears *seamless* to the transport and application layers—and ultimately to the user.

While this calls for minimizing the number of protocol interactions, security considerations demand authentication of mobility signaling and ascertaining that the mobile node has actually moved to a new location and is indeed reachable at the supplied new address. Without reachability test any attacking node could re-direct any other node's traffic to any unwanted destination.

To protect mobility signaling (and data traffic), many solutions (such as MIPv4, MIPv6, HIP) make use of IPsec tunnels between a mobile node and the peer or an indirection point (e.g., a *home agent*). Return routability checks are used to ensure that the mobile node is reachable at the new location; these checks, however, come at the cost of adding at least 1.5 RTT for the associated mobility signaling. [2]

In this paper, we propose a source-routing based mobility protocol that uses in-packet Bloom filters (iBF) to compactly represent the routes. The iBF is collected both during the initial and during the mobility signaling and is used by the on-path ASes, e.g. the border routers, to forward the payload packets through the interdomain path between source and destination ASes[1]. The collected iBF is bidirectional and secure against iBF guessing by attackers.

We show that binding the location of mobile node to the path the packet takes can secure mobility signaling with a single message. The use of iBFs also makes interdomain bicasting easy, as the sender needs to merely bitwise OR the two (or more for n-casting) iBFs together before sending the packet. The routers will then forward and branch the packet through the multicast tree defined by the iBF.

This paper is organized as follows. In section 2, we introduce key aspects related to node security and efficiency in mobility as well as the background, especially the operation of and existing work on in-packet Bloom filters. In section 3 we present our solution for fast mobility and conclude this paper in section 4 with a brief assessment and a discussion of future work.

2. BACKGROUND AND RELATED WORK

Consider a simple mobility scenario as shown in Figure 1 in which a mobile node *MN* communicates with a correspondent (in this case temporarily fixed) node *CN*.[2] Communication takes place via different autonomous systems *AS1*, *AS2*, and *AS3* that each comprise three (border) routers. MN moves from a point of attachment with router F into the coverage of G (*micro mobility* within the same ISP) and then onwards to Y (*macro mobility* across ISPs). As MN

MobiArch'10, September 24, 2010, Chicago, Illinois, USA.
Copyright 2010 ACM 978-1-4503-0143-5/10/09 ...$10.00.

[1] For this paper, we assume a full deployment: each AS has a single logical iBF router. Partial deployments, as needed for introducing these concepts, are also possible, but their discussion is beyond the scope of this paper.

[2] This could be a home agent or another anchor point in cases where no route optimization takes place and packets from a peer are always forwarded via the indirection point.

obtains new network locators each time when moving (in mobile IP terms), it needs to inform the CN (or its home agent) about the new address.

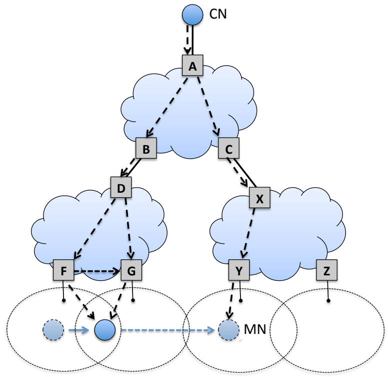

Figure 1: Sample Mobility Scenario

2.1 Mobility Management

Mobility management can be realized with or without support of the routing infrastructure. The latter approach – signaling end-to-end, often assisted by anchor points as intermediary "ends" forming a minimal overlay—dominates mobility support in the Internet. In this case, the packet forwarding paths are associated with interface address pairs of the ends, but not with interfaces of routers along the path. As a result, it is the responsibility of the CN to verify that the packet forwarding path (implied from the new MN locator) is valid after a hand-off, instead of relying on the IP routing fabric to perform this check. This current end-to-end practice results in additional hand-off latency in macro-mobility cases due to extensive reachability test signaling.

The verification triggered by CN is called a reachability test and is needed to overcome potential hi-jacking and flooding attacks. The existing IETF macro-mobility protocols, like MIP [4], MIPv6 [8] and HIP [19], are designed in a way that only the peers and rendezvous nodes participate to the reachability test. This design choice is justified as it is compatible with the current layered Internet-architecture where routing infrastructure provides a common interface for all end-to-end protocols. Therefore, the present mobility solutions can be seen as enhancements to the Internet architecture. They are designed to mitigate the essential re-direction problems at the edges of the Internet topology without adding supportive mechanisms to the routing infrastructure.

This kind of security model is based on an assumption that the Internet routing fabric forwards the reachability messages correctly between peers based on the IP addresses carried in the packets. Another assumption is that CN is able to authenticate MN and send a challenge cookie to it to verify that MN is at the claimed new location after each hand-off. Although, 1.5 RTT reachablity test works fine with macro-mobility protocols, micro-mobility protocols (like HMIP [6], FMIP [18], Cellular IP [5]) suffer from scalability problems related to location update security mechanisms between mobile host and intermediate mobility anchor points. While pure intra-domain micro-mobility optimizations may be beneficial, they are limited in scope and hence cannot solely be relied upon.

To overcome this limitations some overlay approaches, such as i^3 [21], Secure-i^3 [1], and Hi^3 [17], have coupled mobility and packet forwarding plane with each other above IP-layer. Due to the use of Distributed Hash Tables (DHTs), the systems cannot ensure policy compliant paths. However, when the mobility management is realized with support of the overlay routing infrastructure it results in more DoS resistant packet delivery.

2.2 Bicasting

A mobile node may have several (wireless) interfaces or may feature other means (e.g., [7]) to connect to multiple network attachment points (virtually) at the same time. This allows a mobile node to establish a new network connection before discarding the old one (*make-before-break*) or even to continuously maintain connectivity with multiple networks. As long as the dual attachment lasts, the mobile node is reachable via multiple, at least partly, disjoint paths.

In the context of fast mobility, path diversity is employed to minimize the number of lost packets. Since, due to signaling latency, updating and validating the location with the CN may take some time, a mobility protocol may support *bicasting* or *simulcasting* [15, 16, 10]: delivering copies of IP packet to both the old and the new location of the mobile node. Intra-domain or micro-mobility optimizations may use conspiring access routers to replicate and forward packets ($F \rightarrow G$ in figure 1) (e.g., fast handovers[14, 18]) or replicate packets at the last internal router branching router ($D \rightarrow F, G$) (e.g., seamless handovers [10]). For inter-domain or macro-mobility, e.g., when MN moves from G to Y, cross-AS coordination is required and bicasting ideally performed at a suitable router at the last common AS in the path ($A \rightarrow B, C$).

2.3 Source Routing with In-packet Bloom Filters

Bloom filters [3] can be used for efficient multicast [20, 11, 9] forwarding. The idea is to encode the set of links comprising the path or tree into a small Bloom filter, a few hundred bits long, placed in each packet. We call the filter an in-packet Bloom filter (iBF). The iBFs can be delivered to the hosts either in-band or out-band. In this paper, the iBFs are collected with a signaling packet forwarded using IP forwarding.

Each router names its outgoing links with a set of bit positions in the iBF. The link can be added to the iBF by setting those bits using binary OR operation. Similarly, the presence of a link can be tested by checking if the set of bit positions have been set. Assuming m = 128-bit long string, with $k = 5$ bits set to 1, there are $\approx \frac{m!}{(m-k)! \cdot k!} \approx 3 \cdot 10^8$ different link identifiers, making link identifiers statistically unique (assuming the k set bits are randomly distributed).

The set of bit positions, called edge-pair label, can be computed at line speed based on flow ID (i.e. information in the packet header such as source and destination IP address), incoming and outgoing port number and local secret K [9]. The computation can be efficiently accomplished using a fast, spreading hash function (cf. [12, 22]). Computing the edge-pair labels per flow and using cryptographically secure keyed hash makes it difficult for an attacker to guess a valid iBF for a chosen path. Assuming maximum 50% bits set, an attacker has a $2^{-k \cdot l}$ probability of guessing an iBF for length l path with each iBF router setting k bits. This property makes it hard for an attacker to forge iBFs.

Bloom filters have false positives. In iBF forwarding, this results in some additional packets to be delivered in the network, typically over one link. The probability of false positives increases when more links are added to a iBF. For quantitative analysis, we refer to [11]. Shortly, the analysis there revealed that around 35-40 links can be placed into a 256-bit iBF to achieve at least 90% forwarding efficiency (the rate of useful traffic).

3. IN-PACKET BLOOM FILTER BASED MOBILITY

In this section, we present mobility solution based on in-packet Bloom filter (iBF) based forwarding. The basic idea is to run iBF-based forwarding together with IP, so that the first packet between a pair of nodes is routed with IP forwarding and an iBF for the path is collected into the packet. The collected iBF is bi-directional and is used to forward payload packets between MN and CN. As Figure 2 shows, the iBF determines the AS-level path and the forwarding at the edges is based on IP.

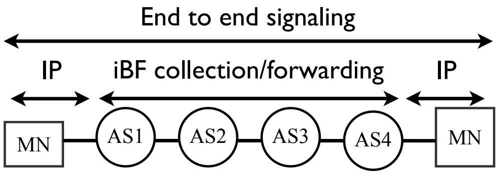

Figure 2: Protocol messages

The protocol is shown in Figure 3. MN sends an I packet to CN. The packet contains a hash anchor $h_{MN}(m_1)$for later mobility event authentication and an iBF collector. As the packet is forwarded through the internetwork, each iBF router adds a local forwarding identifier to the iBF collector in the packet.

The forwarding identifier added to the iBF collector is a local edge-pair label that describes both the next hop and the previous hop links. For scalability[3] each edge-pair label indicates the next-hop AS and previous-hop AS instead of individual routers. The edge ASes may also include an additional link to a Bloom filter capable node close to the MN or CN. 128-bits Bloom filters carried in the packet header should be suitable for the purpose.

Finally, CN receives the packet that now contains the hash anchor and the collected iBF. It replies to MN with so-called init-reply (R) packet, which contains the iBF, and its own hash chain anchor $h_{CN}(c_1)$. The packet is forwarded through the network with the iBF. Once the packet reaches the last iBF-router, it verifies that the destination host is in the part of the network it administers by comparing the IP address to the set of IP prefixes under its domain. Receiving the packet, MN stores the iBF and uses it to send packets to CN.

Later on, after the initial exchange, the CN and the MN add the iBF to each packet sent. The iBF routers determine the next-hop AS by checking which of its neighbors (combined with the incoming AS-number) edge-pair labels match the iBF. This matching is be done by the ingress border router[4].

As the iBF is used to describe the AS-path, the problem of routing the packets to the MN within the destination network still exists. The basic solution for this is to route the packet using the destination IP address. As this leaves some attack possibilities within the network the MN is in (e.g., MN pretending to have an IP address of another node in the same network), the network operator can add an additional link closer to the MN to the iBF.

In the following, we show how to construct edge-pair labels that enable bi-directional in-packet Bloom filters, and how the protocol behaves in various mobility scenarios. Finally, in the end of chapter we discuss how the system could be used for bicasting, as an overlay, and compare it with Mobile IP.

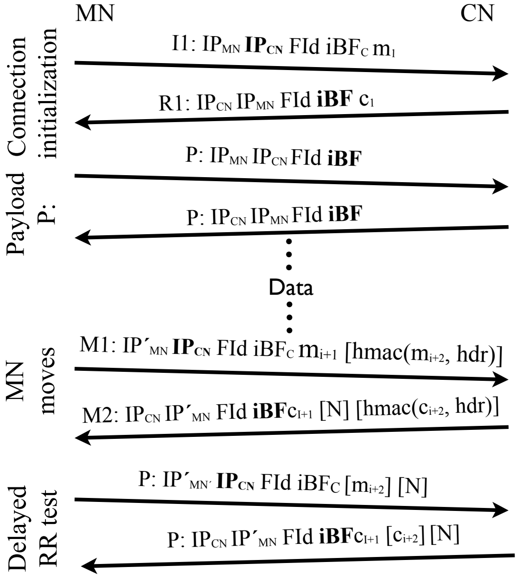
MN
CN
Connection initialization
I1: IP_MN IP_CN FId iBF_C m_1
R1: IP_CN IP_MN FId iBF c_1
Payload P:
P: IP_MN IP_CN FId iBF
P: IP_CN IP_MN FId iBF
Data
MN moves
M1: IP´_MN IP_CN FId iBF_C m_i+1 [hmac(m_i+2, hdr)]
M2: IP_CN IP´_MN FId iBFc_i+1 [N] [hmac(c_i+2, hdr)]
Delayed RR test
P: IP´_MN IP_CN FId iBF_C [m_i+2] [N]
P: IP_CN IP´_MN FId iBFc_i+1 [c_i+2] [N]

Figure 3: Protocol messages

3.1 Collecting and using Bi-directional Edge-pair Labels

As mentioned earlier, each outgoing border router computes an edge-pair label and adds it into the collected iBF. The edge pair-label is computed using the two-part flow-identifier F_1 and F_2 carried in the packet, the next, current, and previous AS numbers and a local secret key as shown in Figure 4 for collecting iBF and in Figure 5 for packet forwarding with iBF.

Each edge-pair label is computed so that the resulting iBF can be used bi-directionally. This is accomplished by utilizing a two part Flow ID: F_1, F_2. They are ordered in the packet header F_1F_2 in one direction and F_2F_1 in the other. Then by comparing the relative size of F_1 and F_2, the iBF

[3] Adding too many edge-pair labels into the iBF would result in either larger iBF or much higher level of each such label added to the collector

[4] The ingress border routers in a single AS need to share the key used to compute edge-pair labels.

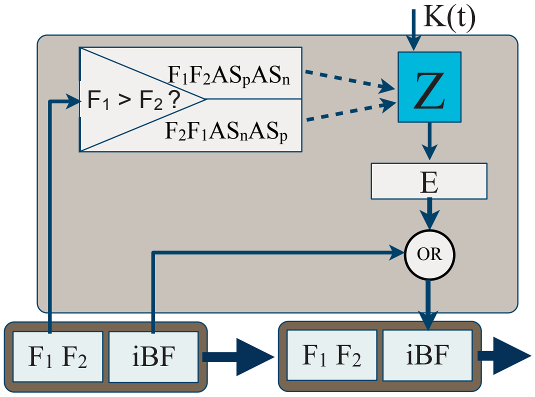

Figure 4: Collecting function in a single router

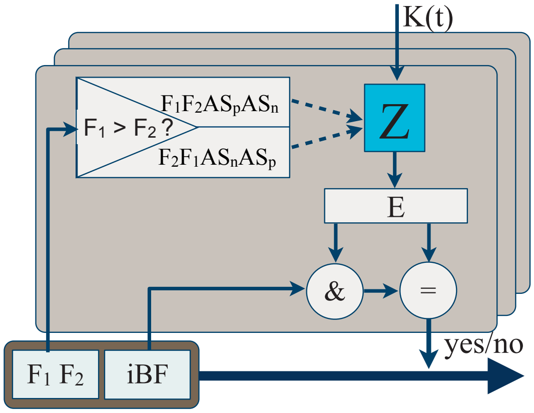

Figure 5: Forwarding function in a single router

routers can determine the relative direction of the packet as shown in the left upper corner of Figures 4 and 5. In other words, if $F_1 > F_2$ then $FlowID = F_1|F_2$, and therefore $E = Z(F_1, F_2, AS_p, AS_n, K(t_i))$; else $FlowID = F_2|F_1$, and therefore $E = Z(F_2, F_1, AS_n, AS_p, K(t_i))$.

The method works, because switching the Flow ID labels F_1 and F_2 enables the routers to distinguish between the two directions. For the rest of the discussion the combination of F_1 and F_2 is just called Flow ID.

Considering packet and payload sizes, IP addresses of MN and CN carried in the packet header can be used as the Flow ID; basically $F1 = IP_{MN}$ and $F_2 = IP_{CN}$. This has the potential benefit of saving space in the packet header, as separate Flow ID is not required. However, to achieve the space saving benefits, either the Flow ID needs to be bound to the current CoAes of the communicating nodes, or the last hop iBF node needs to store connection state translating between the care-of-address and home address.

The forwarding function (Z) is efficient and simple to implement in hardware (e.g. NetFGPA) because of the hash computation is fast and only & (AND) and = (COMP) operations are needed. It is still good to notice that the router needs to call the forwarding function (Z) for all (local policy compliant) neighboring ASes per incoming packet.

3.2 Basic mobility

When MN moves to a new location it sends a location update to CN, as shown in Figure 3. The location update contains a Flow ID, an iBF collector, and the next value in the hash chain. The anchor values of the hash chains are carried in the first packets between the nodes. The intermediate iBF routers add the local edge-pair labels and the packet is delivered to CN. The message is shown in Figure 3.

CN verifies the authenticity of the packet by verifying the revealed hash value and uses the collected iBF to send a reply packet to the MN. The reply packet contains the next value from the h_{CN} and the packet is forwarded using the newly collected iBF. Once MN receives the reply, it stores the new iBF and uses it to send to CN.

Our security solution prioritizes single message easy to compute authentication at the cost of not preventing man-in-the-middle attacks. However, such attacks require that the attacker is on path between MN and CN and can capture the signaling packet. The optional hash chain check in the delayed return routability test (m_{i+2} (shown in Figure 3) prevents this possibility (leaving only a short window of opportunity). The hash chain can be renewed by binding the new hash chain to the current one.

Bicasting can be used in the case of make-before-break. To do so, the mobile node signals the new location with willingness to receive *bicast* for a time. The sender bitwise ORs the two iBFs together and sends the subsequent packet with the resulting iBF. The iBF router at the bicast branching point automatically duplicates the packet to both destinations due to the way the iBF has been constructed[5]. This ensures that the connection can be transferred smoothly from old location to the new one without packet loss, or state requirements in the transit networks.

If the IP addresses of the end points are used as the Flow ID, there are two alternative solutions. Firstly, the destination address in packet header may contain the home-address of the destination node and the last hop iBF router stores the care-of-address, home-address pair. The iBF router then swaps the care-of-address (CoA) to the header before forwarding the packet to the destination. This approach requires especial care in how the iBF routers handle the switch to prevent an attacker from creating false state in the router.

Secondly, it is possible use the CoA as the Flow ID. In this case, during mobility signaling two iBFs are collected. One for the old CoA and another for the new CoA. CN can then bicast data to MN by using the iBF collected using the old CoA. The new CoA has to be added to the packet header so that the final forwarding between iBF router and MN can be done in both destinations.

3.3 Dual mobility

If both nodes are capable of moving, the beginning of the signaling is just as described above. Afterwards, however, the CN performs a delayed return routability test as shown in Figure 3. These protocol messages can be piggybacked in payload packets. The reason for the delayed signalling is that the collected iBF is tied to the *path* between MN and CN current locations, but it is not tied to the MN's *current IP address*. Before CN moves, it needs to verify MN's current IP address in order to send the location update.

[5] For this to work, the Flow ID used to compute the edge-pair labels has to be the same for both paths.

In the case of two mobile nodes, it is assumed that the connection initiation happens to an IP address that is hosted by the MNs rendezvous agent (e.g. home-agent in MIPv6). Assuming two mobile nodes MN1 and MN2 move simultaneously, a rendezvous agent that has a fixed IP address is needed. To prepare for such a case, MN1 sends the mobility update both directly to MN2 and also to MN2's rendezvous agent. The rendezvous agent forwards the packet to the MN2. The update contains the new IP address for the MN1. After receiving the packet, the MN2 sends an iBF collector to MN1 to its new care-of-address. This collector packet acts as a return routability test, to which MN1 responds.

3.4 Evaluation

The iBF mobility is an architectural change intended to make mobility as a first-class citizen in the Internet. The core part of the solution is to bind the communication channel between peers, not only to IP-addresses, but also to the forwarding path between them. All the implications of the change in forwarding fabric cannot be addressed in this paper due to space concerns, but we address here the ones most important for mobility: hand-off security, hand-off latency and DoS vulnerability.

Mobile handoff has two main security requirements. The corresponding node has to know that the handoff message is authentic, i.e. sent by the mobile node (or someone authorized by the mobile node) and that the mobile node is indeed reachable from the address it claims to reside in. Without authentication an attacker can impersonate a mobile node and divert traffic to itself and without reachability test a mobile node can divert the traffic to a location it does not reside in, enabling a DoS attack.

In most cases, when CN receives a location update, the collected iBF and source address suffice and it does not need to make an additional reachability test. This minimizes latency during hand-offs since CN can continue sending packets to MN after receiving a single location update - the M2 message in the location update can be piggybacked in payload traffic. Existing protocols such as MIPv6 and HIP use 1.5 RTT to achieve the same.

The iBF determines the path to the destination AS. Earlier work [9] has shown that creating a valid iBF for a path without access to the secret keys is difficult. The protocol still allows MN to spoof its IP address, but only within the local domain it is located in. This provides incentives for ASes to deploy source address validation in their networks. If the AS level path changes, either node needs to renew the iBF using a location update. A return routability test is needed, if CN itself intends to move, because the security relies on the iBF that describes the path between CN and MN.

With the current mobility architectures, the mobile nodes must establish security association with their peers. For example in MIPv6, MN needs to establish a security association with its home-agent, while in HIP, the peers establish a security association between each other. The iBF-based forwarding only requires a weak security association, based on hash chains, between the peers. The iBF-based forwarding could be coupled with existing IP-based mobility protocols. The mobility protocols can be optimized to utilize the security provided by iBF-based forwarding, or the forwarding fabric can be transparent to these protocols. Further work is needed to better understand the tradeoffs.

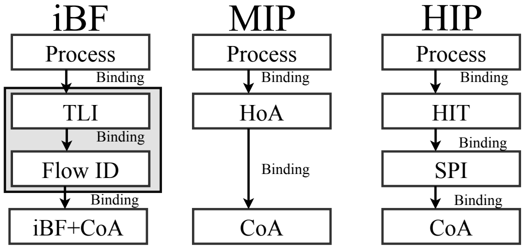

Figure 6: Comparing bindings between iBF, MIP and HIP

Figure 6 compares the *bindings* used in iBF mobility with MIP and HIP. The process is at the top, the forwarding identifiers on the bottom. In the case of MIP and HIP, the forwarding is bound to CoA, which is a single point in the network. Because of this, each solution needs a separate mechanism to verify that the node is actual where it claims to be. In MIP, the verification requires return routability testing both directly and through home agent, since there is no security association between MN and CN. In HIP, the verification is done directly using the security association between MN and CN. In the iBF based mobility, the packet forwarding is bound to the domain path (and CoA). The network does the binding to the domain level path and the weak security association between MN and CN ensures that other nodes cannot spoof the mobility signaling.

This binding of the flow to a network path has also benefits against denial-of-service attacks. As the path binding is done by the network, a faulty implementation in a host (e.g. web server) cannot be used for reflection attacks. It also makes source address validation more effective, as it enough for the AS where the MN is located to validate source addresses to prevent IP address spoofing.

The binding between *Transport Layer Identifier (TLI)* and Flow ID can also enable the mobile node to seamlessly move between heterogenous networks. If the Flow ID is separated from the end point addresses, then the connection can be continued even as the node moves between IPv4, IPv6, and other types of networks.

4. CONCLUSIONS

We have introduced a source-routing-based protocol for fast secure inter-domain handover that requires just a single notification message and inherently supports bicasting for make-before-break mobility. This allows minimizing the handover disruption of a data packet flow. The protocol utilizes the existing IP routing infrastructure and complements mobile IP approaches for the inter-domain case, but it could also be integrated with other mobility solutions such as HIP.

Besides more in-depth quantitative evaluation and an implemenetation, two short-term next steps currently catch our attention: We will investigate how the inter-domain mechanisms could be—possibly recursively— applied to address the intra-domain mobility case for the stub networks to which the mobile node(s) attach. And, while we assumed full deployment in this paper, we believe that the concept can be extended for incomplete (and thus especially incremental) deployment. In the mid-term, it may also be interesting to investigate how to employ multicasting for (group) location updates (e.g., via a rendezvous server) to provide for efficient updates when maintaining connectivity to a group of peer nodes; but, here, especially the secure and privacy considerations require future work.

5. ACKNOWLEDGEMENTS

The authors would like to thank Pekka Nikander, Tuomas Aura, and Jukka Manner for the support and the valuable feedback they have given. Writing of this paper was supported by TEKES (Finnish Funding Agency for Technology and Innovation) as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

6. REFERENCES

[1] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. Towards a More Functional and Secure Network Infrastructure. Technical Report UCB/CSD-03-1242, Univ. California, Berkeley, 2003.

[2] T. Aura, M. Roe, and J. Arkko. Security of internet location management. In *Proc. 18th IEEE Annual Computer Security Applications Conference*, 2002.

[3] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[4] C. Perkins. IP Mobility Support for IPv4. RFC 3344.

[5] A. Campbell, J. Gomez, S. K. A. Valko, C. Wan, and Z. Turanyi. Design, implementation, and evaluation of Cellular IP. *IEEE Personal Commun. Mag.*, 7(4), 2000.

[6] C. Castelluccia. HMIPv6: A Hierarchical Mobile IPv6 Proposal. *ACM Mobile Computing and Communication Review (MC2R)*, 4:48 – 59, 2000.

[7] R. Chandra, P. Bahl, and P. Bahl. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. In *Proceedings of IEEE INFOCOM*, 2004.

[8] D. Johnson and C. Perkins and J. Arkko. Mobility Support in IPv6. RFC 3775.

[9] C. Esteve, P. Jokela, P. Nikander, M. Särelä, and J. Ylitalo. Self-routing Denial-of-Service Resistant Capabilities using In-packet Bloom Filters. *Proceedings of European Conference on Computer Network Defence (EC2ND)*, 2009.

[10] R. Hsieh, Z. G. Zhou, and A. Seneviratne. S-MIP: A Seamless Handoff Architecture for Mobile IP. In *Proceedings of IEEE INFOCOM*, 2003.

[11] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, and P. Nikander. LIPSIN: Line speed publish/subscribe inter-networking. In *SIGCOMM*, 2009.

[12] H. Krawczyk. LFSR-based hashing and authentication. In *Advances in Cryptology CRYPTO'94*, pages 129–139. Springer, 1994.

[13] D. Le, X. Fu, and D. Hogrefe. A review of mobility support paradigms for the internet. *IEEE Communications Surveys & Tutorials*, 8(1):38–51, 2006.

[14] K. E. Malki. Low-Latency Handoffs in Mobile IPv4. RFC 4881 (Experimental).

[15] K. E. Malki and H. Soliman. Simultaneous Bindings for Mobile IPv6 Fast Handoffs. draft-elmalki-mobileip-bicasting-v6-06.txt, Work in progress, July 2005.

[16] H. Matsuoka, T. Yoshimura, and T. Ohya. A Robust Method for Soft IP Handover. *IEEE Internet Computing*, 7(2):18–24, 3 2003.

[17] P. Nikander, J. Arkko, and B. Ohlman. Host Indentity Indirection Infrastructure (Hi3). In *Proc. of SNCNW*, 2004.

[18] R. Koodli. Mobile IPv6 Fast Handovers. RFC 5268.

[19] R. Moskowitz and P. Nikander and P. Jokela and T. Henderson. Host Identity Protocol. RFC 5201.

[20] S. Ratnasamy, A. Ermolinskiy, and S. Shenker. Revisiting IP multicast. *ACM SIGCOMM Computer Communication Review*, 36(4):26, 2006.

[21] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proc. of the ACM SIGCOMM'02*, pages 73–88, Pittsburgh, PA, USA, Aug. 2002.

[22] K. Yuksel, J. Kaps, and B. Sunar. Universal hash functions for emerging ultra-low-power networks. In *Proceedings of CNDS*, 2004.