Naming for Heterogeneous Networks Prone to Episodic Connectivity

Rao Naveed Bin Rais
University of Nice – INRIA
Sophia Antipolis, France
Email: nbrais@sophia.inria.fr

Mariem Abdelmoula
INRIA
Sophia Antipolis, France
Email: marabdel@sophia.inria.fr

Thierry Turletti
INRIA
Sophia Antipolis, France
Email: turletti@sophia.inria.fr

Katia Obraczka
University of California
Santa Cruz, CA, USA
Email: katia@soe.ucsc.edu

***Abstract*—In this paper, we present a naming scheme for heterogeneous networks composed of infrastructure-based and infrastructure-less networks where nodes may be subject to intermittent connectivity. The proposed scheme aims at decoupling object identification from location and is designed to operate with status-quo Internet routing. We showcase the proposed naming scheme implemented on the ns-3 network simulator and demonstrate that nodes are able to receive messages in both infrastructure-based and infrastructure-less networks despite frequent disconnections and changing location identifiers (i.e., IP address), while visiting different networks.**

I. Introduction

In most traditional communication models, the identification of an object (object id) is tightly coupled with the object's location. For example, in the current Internet, the object's location, i.e., the IP address of the machine it resides, is exposed to both the application– and transport layer protocols. Whereas, as pointed out in [6], application– and transport layer protocols should not need to know IP addresses in order to be able to access data. What is more, in heterogeneous networked environments, where devices may be multihomed since they possess multiple interfaces (e.g., PDAs and smart phones may use Wifi and 3G for connectivity), and thus multiple IP addresses, knowing a priori which address to use to communicate with these devices may not be possible. This is especially true in opportunistic forwarding in environments subject to connectivity disruptions.

Problems also arise when a node moves and changes its IP address. While mechanisms like Dynamic Host Configuration Protocol (DHCP) simplify the administration of private IP address spaces, they make IP addresses even less stable. For example, hosts may change their IP addresses because of being turned off or temporarily disconnected even if they have not physically moved.

MobileIP [1], [2] targets "last hop" mobility by allocating a globally routeable address to each mobile node (MN), which may not be feasible in many cases (e.g., allocating a routable IPv4 address to each MN). On the other hand, Shim6 [19] provides mobility solution for multihomed devices by differentiating upper layer identifiers (ULID) from locators, but requires pre-configuration of all interface addresses of the devices. Moreover, both MobileIP and Shim6 suffer from the very basic problem where endpoints are named using topological identifiers (i.e., IP address), so applications have to rely on IP address to communicate with peers.

Proposals like [6] and DONA [7] advocate decoupling identification from location so that, instead of an IP address, applications bind to a location-transparent identifier and the network uses this identifier to find the object, e.g., irrespective of the current network interface being used by the node hosting the object at the time the request for the object was issued. As described in more detail in Section V, some of the proposed approaches that try to separate object identification from location employ a "clean-slate" design philosophy ([3], [4], [5]), whereas others propose patches to current Internet routing ([8], [6], [7], [10]). In this paper, we adopt the latter approach; and our aim is to propose a naming solution that accommodates intermittent connectivity. To our knowledge, this is the first proposal that tries to operate with status-quo Internet routing and still accommodates intermittent connectivity.

We propose a naming mechanism, HeNNA (Heterogenous Networks Naming Architecture), which allows message delivery to nodes independent of their locations while coping with disruptions in connectivity. HeNNA decouples object identification from their location, enabling applications to use "universal object identifiers" independent to where the object may be located. HeNNA is designed to be used with the current Internet routing, while accommodating node mobility, address changes, as well as temporary or long-lived disconnections. We implement HeNNA with our framework MeDeHa (Message Delivery in Heterogeneous, Disruption-prone Networks [12], [13]) , which allows message delivery across an internet consisting of different networks and involving diverse node capabilities. We show that HeNNA augments MeDeHa to use location-transparent naming and thus makes MeDeHa better equipped to support network and node heterogeneity.

HeNNA is able to handle network heterogeneity in a broader perspective. In MeDeHa, nodes use IP address to communicate, which becomes unfeasible when devices are multihomed and are capable to connect to multiple networks. HeNNA targets this problem of node identification and internetwork communication in MeDeHa while managing the change of IP addresses of nodes. We implemented HeNNA on the ns-3 network simulator and showcase its operation with existing Internet routing protocols. We demonstrate that nodes are able to receive messages in both infrastructure-based and infrastructure-less networks despite frequent, arbitrarily-long

disconnections and changes in their point of attachment while visiting different networks.

The rest of the paper is organized as follows. HeNNA and details on its operation are presented in Section II. Section III presents the current implementation of HeNNA and its interoperability with MeDeHa. A simulation-based evaluation of HeNNA is presented in Section IV followed by a review of the related work in Section V. Finally, Section VI concludes the paper with some directions for future work.

II. The HeNNA Naming Mechanism

HeNNA decouples identification with location and allows message delivery across heterogeneous networks, including infrastructure-based (IS-based) and ad-hoc networks, while coping with nodes intermittent connectivity. A source does not have to care about the current location (IP address) of a destination node, and the destination may be connected to any network using any interface at the time of message arrival. For this purpose, applications bind to nodes identifier instead of IP addresses to communicate and each node's location information is maintained by an always reacheable node which we call as the Location and Management Server (LMS). The LMS is a node that has a globally reachable address and maintains location information about the registered nodes. It is also responsible for storing messages on behalf of the nodes when they are unavailable. Details on the functionality of the LMS are presented in Section II-B. The idea is that nodes contact the LMS of other nodes to locate them. Nodes in ad-hoc network can also be reached with a gateway that is connected to an IS-based network, which extends message delivery beyond IS-based networks.

In HeNNA, each node has a globally unique identifier (GUID), and we assume that there is a global DNS-like service with which nodes register their GUIDs against their hostnames. This DNS-like service can either have the normal DNS functionality or a Dynamic DNS service [18], except that nodes are registered with their GUIDs instead of their IP address. How a source resolves a destination's hostname to its GUID is out of scope of this work. GUIDs are persistent identifiers, though a node may change its GUID by registering a new GUID against its hostname in the global DNS-like service. Moreover, applications use GUID of nodes for data communication instead of their IP address. The GUID of a node contains a routeable address of the node's LMS along with the node's identifier which is unique within the context of the LMS. A GUID can also be used to identify a content instead of a node without requiring any major change in the architecture (see Section II-E).

We now present the design details of HeNNA along with description on its major components.

A. Control Messages

HeNNA defines a number of control messages that are used between nodes and the LMS. They are:

***LOC_UPDATE*:** A node sends the *LOC_UPDATE* to its LMS to inform about its current location. This message is sent each time a node changes its location or its IP address is changed, and is directly or indirectly connected to a IS-based network. A node is indirectly connected, when it is in ad-hoc mode and is connected an IS-based network via an associated node. The LMS updates the location information only for the node that is registered[1] with it. This message comprises of the GUID of a node and its current IP address.

***LOC_REQ*:** A node sends this message to the LMS of a destination, inquiring about the destination's location, when it has a message to send, and is connected to a backbone network. This message contains the GUID of the destination.

***LOC_RESP*:** The LMS responds the *LOC_REQ* with the *LOC_RESP* either by sending the inquired node's (destination) current IP address, or its own IP address (if the destination's location is unavailable). The latter case implies that the LMS will store messages for the destination. This message comprises of the destination's GUID and its IP address.

B. Location and Management Server (LMS)

The LMS is responsible for keeping track of a node's current location (i.e., a globally routeable address). It is a node that must be connected to the Internet and has a persistent routeable address. The LMS may maintain the location information for one or more nodes, and can either be maintained by an Internet Service Provider (ISP), or by a company on behalf of its employees, or by an individual to maintain personal location updates. It is also responsible for storing messages on behalf of a node, if the node's information is unavailable.

The LMS keeps a list of the registered nodes, and maintains a mapping between the nodes' GUID and their latest routeable address. The mappings are valid for a specific amount of time, and are expired if the LMS does not get a *LOC_UPDATE* from nodes for a long time. As a node moves to a new location (or changes its IP address), it informs its corresponding LMS by sending a *LOC_UPDATE*, only if it is directly or indirectly connected to an IS-based network. As a result, the LMS adds a new entry for the node's GUID or updates node's GUID mapping to point to the new IP address, and in response, sends all messages that it has stored for the node, during the time when the node was unreachable.

Each node locally maintains a cache that comprises of the GUID to IP mappings for the recently inquired nodes. Thus, a source or a message carrier ***S***, when has a message for a destination ***D*** with identifier *GUID(D)*, consults its local cache to check if it has a corresponding entry of IP address against *GUID(D)*. If the node does not have an entry, it contacts the LMS of ***D*** to acquire ***D***'s current routeable address by sending a *LOC_REQ*. As a result, the LMS sends back the current routeable IP address of ***D*** via the *LOC_RESP*, if it has information about it. ***S*** then uses the received routeable address to route the message towards ***D***. If a LMS does not have information about ***D***, it sends back its own IP address, which implies that it is going to store messages for ***D***. An

[1]The registration process can be made secure so as to prevent unauthorized/malicious nodes from providing wrong location information about the nodes to the LMS. However, we do not consider this case in this paper.

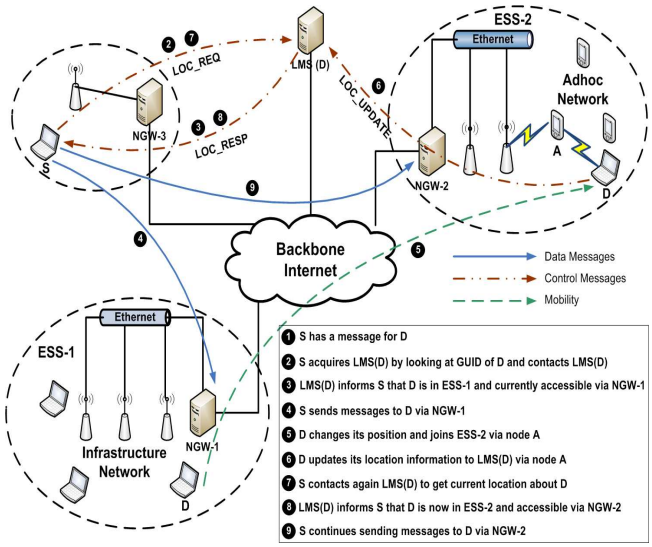

Fig. 1. An example of message delivery using HeNNA. **S** having **GUID(D)** sends a message to **D** by first contacting **LMS(D)**.

exemplary scenario is shown in Fig. 1, in which ***D*** moves from ESS-1 to ESS-2, and is connected to ESS-2 via ad-hoc interface when the *LOC_REQ* was sent to its LMS by ***S***. There is a time to live (TTL) associated with each stored message, and messages passed their TTL are expired at the LMS.

The functionality of the LMS can be compared to that of the home agent (HA) in MobileIP, with the following differences. The HA implicitly intercepts the messages sent to a MN which means that both HA address and MN home address must belong to the same subnet. In HeNNA, a request is explicitly sent to the LMS to locate a node before any communication takes place. Also, in HeNNA, the LMS is also responsible for storing data for nodes when they are unavailable whereas the HA is expected to have location information about a node all the time which may not always be true. Note that if MobileIP infrastructure is already available, the functionality of the HA could be modified to make it as the LMS.

A comparison can also be made between the functionality of the LMS and that of the rendezvous server (RVS) in HIP [16]. Like LMS, a RVS also maintains location information about registered nodes, but unlike LMS, a RVS does not store any messages on behalf of unavailable nodes. Moreover, nodes use the RVS only to exchange HIP base with the mobile nodes, but the data is never routed via the RVS. Implicitly, it requires that both initiator and responder is available for the data exchange to take place. There is no such constraint in HeNNA, as a source can send data even if a destination's is unavailable.

C. Local Network Operation

When a nodes are behind a Network Address Translation (NAT) server, a DHCP server may be assigning addresses to the participating nodes (local nodes) from a private address space. In this case, only the local gateway (e.g., NAT Server) has a globally routeable address. In the context of HeNNA, we call this gateway as the Network Gateway (NGW).

***Network Gateway (NGW)*:** The NGW comes into operation when a DHCP server is assigning IP addresses to the local nodes, or nodes are using private static addresses in an ad-hoc network and are connected to the backbone via a gateway. Besides the regular NAT server operation, the NGW is responsible to keep a mapping between the local nodes GUID and their IP addresses. To perform this task, the NGW intercepts location updates (*LOC_UPDATE*) from the local nodes, replaces the local IP address with its own IP before forwarding the update to the LMS. The process is transparent to nodes. This also implies that in this case, the *LOC_UPDATE* does not need to be sent to the LMS for each newly acquired IP address, as long as the node is in the same local network. This concept is similar in approach to the Hierarchical MobileIP (HMIP) [11], where local movement is not propagated to the HA. Note that as GUID to IP address mappings at the LMS may often expire, the *LOC_UPDATE* messages are forwarded to the LMS, before an entry expires at the LMS, even if the node's NGW does not change.

The NGW keeps a mapping of a local node's GUID and IP address of the interface with which it has sent the *LOC_UPDATE*. In case of an indirect connection to a IS network for a node with ad-hoc interface, this can be its ad-hoc IP address. If a node is simultaneously using its ad-hoc and IS interface, its IS-based IP address is kept in the mapping. Besides, If a source sends a *LOC_REQ* to the LMS, the NGW may intercept the request to respond on behalf of the LMS, if it already knows the destination (e.g., if destination is available locally, the NGW responds the *LOC_REQ* with the local IP address of the destination by looking into the local mapping).

D. Ad-hoc Network Operation

Communication operation in an ad-hoc network is simple and is performed without involving the LMS or the NGW, as long as the communicating nodes are in the same network. In this way, nodes exchange their GUIDs as part of their neighbor sensing procedure (e.g., using "hello" messages). As a result, this GUID information is propagated to other neighbors, just the same way as the neighbors IP address information is passed in the regular ad-hoc routing protocols. In a network where routing is performed using IP address, nodes also exchange their IP address along with GUID and all nodes keep local mappings between GUID and IP address of other nodes. Entries in this local mapping are either expired, if a node does not receive an update from another node for a specific period of time, or refreshed if the node changes its IP address. Consequently, this mapping is passed to the corresponding LMS of nodes, as soon as one of the participating nodes holding the mapping, joins an IS-based network.

E. GUID as Content Identifiers

Till now, we assume that GUIDs represent endpoint nodes, and nodes use GUIDs to communicate. Instead of a node identifier, the GUID can also be treated as a content identifier without requiring major changes to HeNNA. Thus, applications use the GUID as the content identifier, and users searching for a specific content use the GUID to contact the

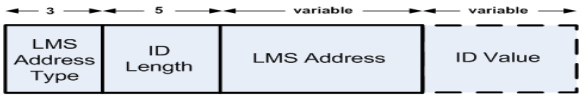

Fig. 2. Composition of a GUID.

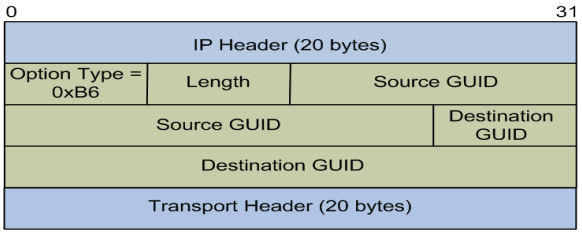

Fig. 3. GUID header in the protocol stack.

LMS of the content in order to locate it. The LMS, in return, passes the current routeable address of a node carrying the content. In case where more than one node carry the same content, a mechanism is required at the LMS to maintain one-to-many mappings between GUID and IP address of the nodes holding the content. We do not currently deal with one-to-marry mappings at the LMS, but we believe that it is not very hard to maintain.

F. GUID format

As shown in Fig. 2, a GUID is composed of:

***LMS Address Type*:** Indicated by 3-bits – 1 for IPv4, 2 for IPv6, 3 for DTN EIDs, other types are unused.

***ID Length*:** 5-bits indicating in how many bytes the ID of a node is represented. A zero value means that the ID value is absent (a personal LMS).

***LMS Address*:** Address of the LMS of a node. The length of this field is variable and depends upon the type of address being used (e.g., 4 bytes for IPv4 address).

***ID Value (Optional)*:** Node identifier within the context of the LMS. Length is variable (maximum: 32 bytes).

A GUID header is placed between the IP and the transport headers of a message, as in [7], which allows intermediate nodes to get information about a destination's GUID, in case a path is disconnected, and a message needs to be stored. To allow messages to traverse nodes that run regular TCP/IP stack, we place the header as an IP option. Position of the GUID header is shown in Fig. 3, with 5 bytes representing GUIDs (1 byte control, 4 bytes IPv4 address). Note that there is an overhead associated while adding GUID headers to each message. For the case of Fig. 3, this overhead is 12 bytes/message. Also, there is an overhead due to the exchange of control messages between nodes and the LMS, and the amount of this overhead depends how frequently the LMS is contacted by the nodes.

III. HeNNA Implementation

We implement HeNNA in NS-3 [14] and use it with an extended version of our framework MeDeHa [13] that allows message delivery across an internet consisting of different networks and involving nodes with diverse capabilities. In MeDeHa framework, the nodes use IP address for identification, which implies that a node should a priori know an IP address of a peer node. This also means that the communication is vulnerable to changing IP addresses which is not uncommon. HeNNA overcomes this shortcoming of MeDeHa by using persistent identifiers for the nodes.

When operating with HeNNA, MeDeHa (MDH) nodes use GUID for communication. A MDH node sends the *LOC_UPDATE* to its LMS, when it is associated to an IS-based node (e.g., an AP or base station), or when it is indirectly connected via a neighboring node associated to an IS network. Besides, the MeDeHa notification protocol [12] has been extended so that APs exchange GUIDs of the associated MDH nodes instead of IP addresses in IS network, and in ad-hoc mode, MDH nodes exchange both their GUID and IP address in neighbor sensing handshake (comprising of *HELLO* and *NEIGHBOR_INFO*). Besides, nodes also exchange GUID of the nodes that they encountered in the past. This information is used in the relay selection process.

A MeDeHa node ***S*** when wants to send a message first checks for the destination ***D*** information locally, as IS-based nodes in MeDeHa maintains local connectivity information within an Extended Service Set (ESS). If the information is not found, the LMS of ***D*** is consulted to get the location. Messages are forwarded based on MDH nodes' GUID (rather than their IP addresses in the original MeDeHa framework), which enables MDH nodes to get their messages even if their IP addresses are changed due to temporary disconnection or joining a new network. APs may store messages for temporary unavailable destinations within an ESS, but if a destination is not connected to the ESS for a long time, APs transfer the stored messages to destinations' corresponding LMS.

IV. Results

We show how HeNNA helps in message delivery to nodes irrespective of their point of attachment to the network and IP address. We consider that 40 students move within and between 3 campuses of a university. These campuses do not belong to the same subnet, and are not directly connected, as shown in Figure 4. Students carry portable devices that run MeDeHa framework and HeNNA. The students move between 3 campuses and while traveling between campuses, they remain disconnected for a long period of time. Using their devices, the students are also able to connect both in IS and ad-hoc modes. At a campus, the students use the local ESS for connectivity, are behind a NAT, and a DHCP server is assigning IP addresses dynamically from a private address space. Nodes change their IP address due to disconnection or a change of association to APs, even when present in the same ESS. Moreover, connectivity is not guaranteed everywhere within a campus. Two of the campuses are comprised of 6 APs while the third has 3 APs. Each campus has a NGW that has a globally routeable IP address. We assume that there are two LMS (LMS-1 and LMS-2), each responsible for location

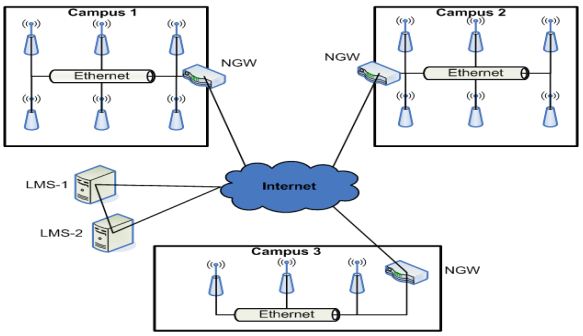

Fig. 4. Three campuses are connected to the Internet via NGWs.

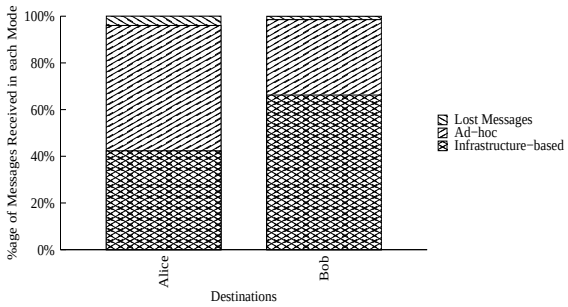

Fig. 6. Percentage of messages received in both IS and ad-hoc networks.

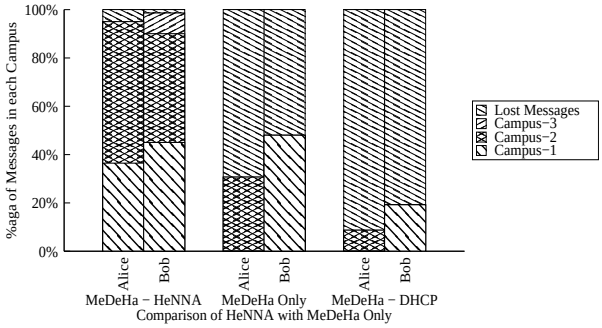

Fig. 5. Percentage of messages received in each campus.

information of 20 students. We assume that two students, Bob and Alice are downloading a file from a server in the Internet, and want to continue downloading it while moving, while file contents are sent at an average rate of 5 messages/s (5KB/s). The mobility traces are obtained using BonnMotion Mobility Model [15] and the students move at a speed that is uniformly distributed between 1 and 3 m/s, and stay at some places for a time that is distributed between 0 and 300 seconds, and total simulation time is 2 hours. Campus 1 and 2 has an area of 600m x 600m, while Campus 3 spans over an area of 600m x 300m, and the total simulation area is 3km x 1.5km.

For opportunistic ad-hoc forwarding in MeDeHa, we use Encounter-based Replication mechanism (ER) as described in [12], where a source or a relay forwards a message to another relay, if the latter has encountered the destination at least twice and more often than the former. As both Bob and Alice change their IP address with the change in the network attachment point, it is interesting what percentage of the file they receive in each network that they visit. Moreover, measuring the overall delivery delay gives us an estimate about how long they remain disconnected. We compare the performance of HeNNA with 2 cases where HeNNA is not used. Fig. 5 provides the distribution of the percentage of messages received and lost in all 3 campuses.

With HeNNA (MeDeHa-HeNNA), Bob received data in all 3 campuses, and got 98.5% of the file (45% each in Campus 1 and 2, and 8.5% in Campus 3), while Alice received data in Campus 1 and 2 only and got 95% of the file (36% in Campus 1 and 58.5% in Campus 2).[2] Some messages are expired (expiry time is 40 minutes) while being stored at the LMS. This loss of data can be coped with by adding application level reliability. The average delivery delay for Bob and Alice is 242.3 and 233.6 seconds respectively. When using regular MeDeHa (MeDeHa only) in which nodes IP addresses are static (which is neither practical nor scalable), the delivery ratio is 48% for Bob and 30.7% for Alice. This is because connectivity information is not passed beyond the ESS in MeDeHa. Bob was initially in Campus 1, so he could receive data either in Campus 1 or via relays. APs in Campus 1 keep the messages stored for a long time when Bob is unavailable; hence, a lot of messages are expired. Similarly, Alice was initially in Campus 2, and received all messages in Campus 2. The delivery delay for Bob is 628.4s and for Alice is 25.9s). We also used dynamic addressing mechanism with MeDeHa (MeDeHa-DHCP), in which students change their IP address when moving/reconnecting. This has a drastic effect on MeDeHa's performace (delivery ratio reduces to 19.1% for Bob and 8.67% for Alice). The delivery delay in this case is very low (0.97s and 0.62s respectively) as both students only received messages in the beginning before their IP addresses are changed. The message size is 1 KByte, and HeNNA control messages and the GUID header included in each message caused an overhead of 1.61%. For this experiment, Bob and Alice used 12 and 6 IP addresses respectively for IS interface, while their ad-hoc interface IP addresses are static.

During the mobility, Alice and Bob communicate with other nodes they encounter within or outside campuses in ad-hoc mode, and receive data destined to them either via relays that carry data for them, or when they are indirectly connected to an IS-based node. Hence, it is interesting to see what percentage of data both Alice and Bob has received during each mode (IS and ad-hoc) in all three campuses, and even while moving between campuses. Figure 6 shows the distribution of file received in both IS and ad-hoc modes.

We notice that out of total of 98.5% of the file contents,

[2]Note that Bob and Alice receive a few messages off-campus in ad-hoc mode when encountering relays but we consider these messages as being received in the recently visited campus, as they are not too many.

Bob received 66.2% in infrastructure mode (while connected directly to APs), and 32.3% in ad-hoc mode (via relays or by indirectly connecting to an IS-based network). On the other hand, Alice received more data in ad-hoc mode (53.7%) than while connnected to the IS-based network (42.3%). We also have conducted simulations for communication between students, where both source and destination are mobile and change their point of attachment to the network. However, we do not include these results here due to space limitations.

V. Related Work

CCN [5] is a recent architecture to decouple identification with location, but its performance may suffer in an environment with high mobility, as in CCN, data messages are not routed (only *interests* are routed). So, data content may not reach, if the route to the *interested* peer changes; hence the *interest* has to be resent. EDIFY [4] presents a region based naming architecture where nodes identifiers are comprised of a region ID and a node ID. This makes mobility very difficult to handle and dependent on regions. Also, too many types of identifiers makes the scheme complex and impractical.

LISP [8] separates identification with location, but does not provide a specific mapping system between Endpoint Identifiers (EID) and Routing Locators (RLOC). Balakrishnan et al. [6] present a naming architecture to patch Internet routing by proposing multiple levels of name resolution, i.e., Name to Session IDs (SID), SID to EID, and EID to IP, but their proposal is not designed for high mobility scenarios, and also assumes that a source has access to all resolvers, which may not always be possible.

HIP [10] and DONA [7] use flat, self-certifying names for identification. HIP requires that two communicating nodes negotiate HIP base exchange before data communication takes place, which is not feasible, especially in the absence of a contemporaneous path. On the other hand, DONA [7] suffers from scalability problem as each resolution handler maintains a forwarding table for each content in the network. Moreover, Delay Tolerant Network (DTN) Bundle Architecture [9] defines Endpoint Identifiers (EID) for nodes, but does not agree upon a naming mechanism, and solutions like Intentional naming [3] are based on routing predicates and are not workable with Internet routing.

A Solution like MobileIP [1], [2] solves the mobility problem by assigning persistent home address to nodes, but requires that each node has a globally routeable IP address. HeNNA differs from MobileIP in this respect, i.e., no permanent routeable address is required for nodes; rather a GUID is owned by each node and a routeable address of a node is acquired by a source on-the-fly.

Dynamic DNS (DynDNS) [18] allows hosts to cope with the problem of changing their IP addresses by dynamically updating hostname to IP address mapping with the service provider, but the update mechanism for DynDNS is not very efficient and an IP change update may take a few minutes (and sometimes a few hours), as the update needs to be propagated across all DNS servers. Thus, it is also not very efficient in case where IP address of hosts keeps on changing frequently. DynDNS is not very effective when hosts are behind a firewall.

Node Identity Internetworking Architecture [17] provides an IS-based solution to separate identification with location by defining locator domains (LD), but does not explain the operation in ad-hoc networks and networks with disruptions. It is based on routing hints that are resolved at LDs and serve as source routing. It means that the source is responsible for adding the routing hints when sending a message and if the destination moves and changes its LD, the messages are lost.

VI. Conclusion

In this paper, we have proposed a naming mechanism HeNNA that decouples nodes identification with location. HeNNA is simple and is designed to operate with status-quo Internet routing while coping with nodes temporary disconnections and change of IP address during communication sessions. We have evaluated HeNNA with our framework MeDeHa via simulations, and observed that it is able to deliver messages to nodes even with high mobility. A thorough evaluation of HeNNA's performance and its implementation on a real tested is part of our ongoing and future work. Using HeNNA to integrate the Internet and the DTN Bundle Architecture [9] is another future direction.

References

[1] C. Perkins, *IP Mobility Support for IPv4*, RFC 3344, 2002.

[2] D. Johnson, C. Perkins, and J. Arkko, *Mobility Support in IPv6*, IETF 3775, 2004.

[3] P. Basu, D. Brown, S. Polit, and R. Krishnan, *Intentional Naming in DTN*, DTNRG Internet draft, 2009.

[4] M. Chuah, L. Cheng, and B. Davison, *Enhanced Disruption and Fault Tolerant Network Architecture for Bundle Delivery*, Proc. of IEEE Globecom, 2005.

[5] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, *Networking Named Content*, Proc. of ACM CoNext 2009.

[6] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, *A Layered Naming Architecture for the Internet*, Proc. of ACM SIGCOMM, 2004.

[7] T. Koponen, M. Chawla, B-G Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, *A Data-Oriented (and Beyond) Network Architecture*, Proc. of ACM SIGCOMM, 2007.

[8] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, *Locator/ID Separation Protocol*, Internet Draft, Network Working Group, 2010.

[9] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, *Delay-Tolerant Networking Architecture*, RFC 4838, 2007.

[10] R. Moskowitz and P. Nikander, *Host Identity Protocol Architecture*, RFC 4423, 2006.

[11] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, *Hierarchical Mobile IPv6 Mobility Management*, RFC 5380, 2008.

[12] R.N.B. Rais, T. Turletti, and K. Obraczka, *MeDeHa - Efficient Message Delivery in Heterogeneous Networks with Intermittent Connectivity*, INRIA Research Report, inria-00464085, 2010.

[13] R.N.B. Rais, T. Turletti, and K. Obraczka, *Coping with Episodic Connectivity in Heterogeneous Networks*, Proc. of ACM MSWiM, 2008.

[14] NSNAM Project, *Network Simulator 3 (NS-3)*, http://www.nsnam.org.

[15] BonnMotion, *A Mobility Scenario Generation and Analysis Tool*, http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion.

[16] J. Laganier and L. Eggert, *Host Identitiy Protocol (HIP) Rendezvous Extension*, RFC 5204, 2008.

[17] S. Schtzb, H. Abrahamssona, B. Ahlgrena, and M. Brunnerb, *Design and Implementation of the Node Identity Internetworking Architecture*, Computer Networks, 54(7), pages 1142-1154, 2010.

[18] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, *Dynamic Updates in the Domain Name System (DNS UPDATE)*, RFC 2136, 1997.

[19] E. Nordmark and M. Bagnulo, *Shim6: Level 3 Multihoming Shim Protocol for IPv6*, RFC 5533, 2009.