

Towards Truly Heterogeneous Internets: Bridging Infrastructure-based and Infrastructure-less Networks

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

Marc Mendonca
University of California
Santa Cruz, CA, USA
Email: msm@soe.ucsc.edu

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—There is no doubt that networks are becoming increasingly heterogeneous and future internetworks will likely interconnect different types of networks including wired, infrastructure-based wireless as well as infrastructure-less wireless networks, a.k.a., multihop mobile ad-hoc networks (or MANETs). Integrating MANETs to infrastructure-based networks (wired or wireless) allows network coverage to be extended to regions where infrastructure deployment is sparse or nonexistent as well as a way to cope with intermittent connectivity. However, to date there are no comprehensive solutions that integrate MANETs to infrastructure-based networks. In this paper, we introduce a message delivery framework, MeDeHa++ that is able to bridge together infrastructure-based and infrastructure-less networks. Through extensive simulations, we demonstrate the benefits of MeDeHa++, especially in terms of the extended coverage it provides as well as its ability to cope with arbitrarily long-lived connectivity disruptions. Another important contribution of this work is to deploy and evaluate our message delivery framework on a real network testbed as well as conduct experiments in “hybrid” scenarios running partly on simulation and partly on real nodes.

I. INTRODUCTION

The vision of a world where users can be connected “anytime, anywhere” which seemed quite futuristic a decade or so ago, is becoming more and more a reality. One of the critical enabling technologies for this “universal connectivity” is the emergence of an internet that interconnects different types of networks, ranging from wired, infrastructure-based wireless (e.g., cellular-based networks, wireless mesh networks) to infrastructure-less wireless networks (e.g., mobile ad hoc networks, or MANETs, vehicular networks, or VANETs). Interconnecting such different networks poses several challenges including seamless message delivery. Additionally, a number of emerging applications such as environmental monitoring, emergency response, vehicular communications, to name a few, require that future internetworks be tolerant to frequent, long-lived connectivity disruptions.

As it will become clear from the description of related work presented in Section VII, current solutions only provide partial solutions to the heterogeneity problem faced by future internets. As an attempt to providing heterogeneity support, we developed MeDeHa (Message Delivery in Heterogeneous,

Disruption-prone Networks [13]) which proposes a framework to allow message delivery across an internet consisting of different types of networks. However, the design presented in [13] is very preliminary and only provides direct delivery in ad-hoc disruption tolerant networks (DTNs). Also, to date, MeDeHa does not offer a solution for integrating infrastructure-less multi-hop mobile wireless networks (MANETs). This is precisely the focus of this paper, i.e., provide ad-hoc network support and a flexible mechanism to bridge together infrastructure-based and infrastructure-less networks even under intermittent connectivity, and a solution to fill in connectivity gaps left by infrastructure-based networks using MANETs.

In this paper, we present MeDeHa++, a comprehensive framework to provide message delivery across heterogeneous disruption-prone networks. MeDeHa++ involves a complete 2-hop ad-hoc DTN routing protocol, and integrates infrastructure-based networks with infrastructure-less networks, which was not provided by the original MeDeHa framework. Additionally, MeDeHa++ is able to provide message delivery to non-MeDeHa MANET nodes without proposing any modification to existing MANET protocols. Moreover, we show that we can take advantage of the multi-hop connectivity information of MANET networks to create transit networks that connect otherwise disconnected infrastructure-based networks. Unlike previous proposals (e.g., [4], [5], [6]), MeDeHa++ does not require any modification to existing MANET routing protocols.

We demonstrate MeDeHa++ operation and performance benefits through extensive simulations using a range of scenarios, including scenarios represented by real mobility traces. Additionally, we implement MeDeHa++ as a user space daemon in Linux and (1) run “live” experiments on a real testbed, as well as (2) conduct “hybrid” experiments running partly on simulation and partly on real nodes.

The rest of the paper is organized as follows. Section II describes the MeDeHa++ framework to bridge infrastructure-less and infrastructure-based networks for message delivery. Major functional components of MeDeHa++ are presented in Section III followed by Section IV which presents how the

framework is used to fill in connectivity gaps and is able to deliver messages to MANET nodes. An implementation of MeDeHa++ using Linux 2.6 is described in Section V while Section VI presents a thorough simulation-based evaluation of MeDeHa++, as well as experiments with real machines and hybrid scenarios. Related work is reviewed in Section VII followed by Section VIII which concludes the paper highlighting some directions for future work.

II. MEDEHA++ FRAMEWORK

The MeDeHa++ framework achieves the following goals:

- Seamless message delivery between two nodes¹ irrespective of network type.
- Partition mending through multihop ad-hoc (MANET) “transit networks”.
- MANET routing protocol independence. This allows MANET nodes to communicate with MeDeHa++ nodes without running MeDeHa++.

We base our design on the principle that in order to join two networks, there must be a gateway that is able to understand traf?c for both networks. This node can either have two interfaces (e.g., a cellular phone with a 3G and a Wifi interface), or it can use the same interface card to join more than one network by using different frequency bands to communicate [7]. In our framework, we define gateway nodes (GW) to be MeDeHa++ nodes (MDH) with interfaces to multiple networks. Thus, a GW node is able to receive and pass traffic from one network to another.

For instance, when involving MANETs, the GW is a node that runs MeDeHa++ software and is configured with a MANET routing protocol. Thus, when this GW node hears a “hello” message from a MANET node, it learns about the presence of the MANET and passes this information to other connected networks (ad-hoc or infrastructure-based). In this way, nodes in the other networks are able to forward messages to the MANET nodes via the GW node.

In infrastructure-based networks, the message delivery is achieved by keeping track of nodes connections and disconnections. It also involves communication between connected infrastructure-based nodes, such as access points (AP) in an Extended Service Set (ESS), to share nodes connectivity information. These infrastructure-based nodes also store messages for unavailable nodes for a pre-defined amount of time. Nodes running MeDeHa++ in ad-hoc modes detect other nodes by broadcasting “hello” messages and provide message delivery while coping with disconnections (storing messages for unavailable nodes). Nodes are also able to keep a 2-hop network view as a result of their neighborhood exchange mechanism. A node that is connected to an infrastructure-based network and also supports ad-hoc mode acts as a GW node to interconnect these networks.

MDH nodes, whether they are connected in infrastructure or ad-hoc modes, are able to gather MANET nodes information

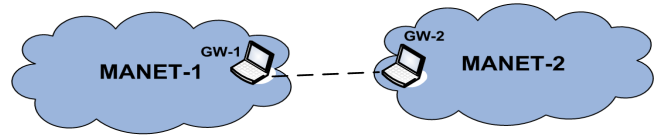


Fig. 1. GW nodes connecting two different MANETs

through a GW connected to a MANET.² In this way, the MDH nodes are able to see all MANET nodes as 2-hops away with the GW node as the next hop. Moreover, GWs in two different MANETs can join the MANET networks, when the GWs encounter each other, as shown in Fig. 1. GW nodes can also learn about the presence of other GW nodes in a MANET, and can exchange information about connected networks. This mechanism allows MANETs to act as “transit networks” to bridge disjoint networks (Fig. 2).

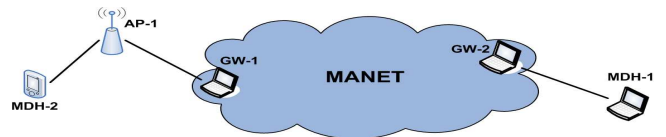


Fig. 2. MDH-2 is able to communicate with MDH-1 by traversing through MANET using GW-1 and GW-2

As mobile nodes may join and leave a MANET at any time, passing information from GW to other networks is event-based, and whenever a change in the MANET network is detected, a message is sent from the GW node to nodes in other networks. Following heuristic-based routing principle of disruption tolerant networks, when a GW node leaves a MANET network, it maintains information about nodes in the MANET, and passes this information as “recently visited nodes” to other nodes it encounters. An example of multihop message delivery from a source S to a destination D in a MANET with intermittent connectivity is presented in Fig. 3.

III. MEDEHA++ FUNCTIONAL COMPONENTS

The MeDeHa++ framework is comprised of the following main components:

A. MeDeHa++ Notification Protocol

The aim of MeDeHa++ notification protocol is to learn and collect network information from all the networks a node is connected to, and to use this information to construct the routing/contact tables. It also passes this information from one network to another in order to provide interoperability. MeDeHa++ notification protocol has two main components, neighbor sensing and neighborhood information exchange.

¹Multi-point delivery is one of our future work directions.

²For the rest of the paper, we generally use GW to term a MeDeHa++ capable MANET node.

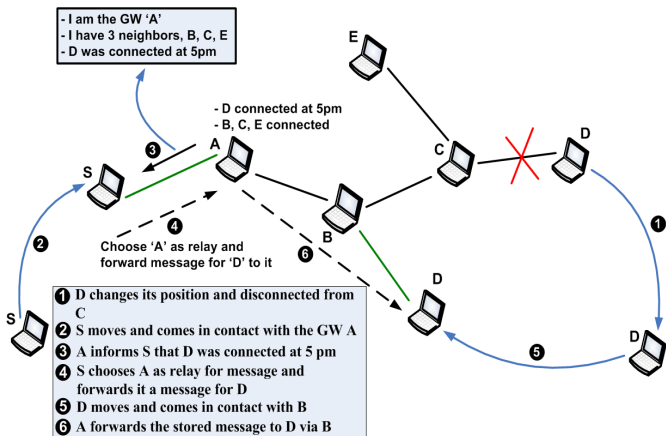


Fig. 3. A typical example of message delivery in MANET while coping with intermittent connectivity, where the source **S** is a MDH node and the destination **D** is a regular MANET node. **S** sends a message to **D** via the GW **A** that it encounters

1) *Neighbor Sensing*: The neighbor sensing mechanism is used to detect the presence of immediate neighbors. If the functionality is provided by underlying network (e.g., association/disassociation management frames in IEEE 802.11), MeDeHa++ nodes use this information to learn about the neighbors. In networks where neighbor sensing is not already available (e.g., in ad-hoc networks), it is achieved by periodically broadcasting *HELLO* messages. Nodes broadcast their IDs (e.g., IP address) and associated networks (e.g., availability of an Infrastructure-based network or a MANET network) in their *HELLO* messages. Nodes may also send their current status in terms of battery life, memory level, or any other tags such as mobility pattern (“bus”, “pedestrian” etc.).

2) *Neighborhood Information Exchange*: Neighborhood information exchange is performed using the information collected via neighbor sensing. For infrastructure-based networks, MeDeHa++ uses the notification messages that are proposed in [13], plus some new protocol messages to exchange MANET nodes information such as *NEIGHBOR_PRESENT* and *MANET_PRESENT*, sent from a GW to its associated infrastructure-based node to inform about a node presence in ad-hoc mode and in a MANET respectively; *LEAVE_MANET* sent from a GW to an infrastructure-based node to inform that it is no longer part of the MANET network. In ad-hoc networks, nodes respond to *HELLO* messages with *NEIGHBOR_INFO* messages, which contains information on all the neighbors of the transmitting node. This completes the *Hello Handshake* between two nodes. The *NEIGHBOR_INFO* may contain several notifications including *CURRENT_NEIGHBORS*, *RECENT_NEIGHBORS*, *MSG_VECTOR*, and *MANET_NEIGHBORS*. These notifications are summarized in Table I.

B. Routing and Contact Table Management

Based on the information learned using the notification protocol, MeDeHa++ Nodes (MDH) build their routing and contact tables. The Routing tables contain the information

TABLE I
NOTIFICATIONS IN *NEIGHBOR_INFO* MESSAGE SENT IN RESPONSE TO *HELLO* BROADCASTS FOR AD HOC NETWORKS

Notification Name	Description
CURRENT_NEIGHBORS	Sent to inform receiving node about neighboring nodes
RECENT_NEIGHBORS	Sent to inform receiving node about the nodes recently seen by the transmitting node
MSG_VECTOR	Contains sequence numbers of messages stored at transmitting node
MANET_NEIGHBORS	Sent by a MeDeHa-capable MANET node to inform about the connected MANET nodes

about the immediate neighbors that are connected. For MDH nodes, the routing tables are comprised of 2-hop network information learnt via *CURRENT_NEIGHBORS* notification. If the MDH node is connected to a MANET, the routing table maintains multihop connectivity information learned from the MANET, while the contact table contains information for all nodes that have been encountered within a pre-defined time period. This information is propagated to other nodes using *RECENT_NEIGHBORS* notifications.

C. Relay Selection and Forwarding

Selection of a relay is based on the contact tables of MDH nodes, and the information gathered from *RECENT_NEIGHBORS* notification. The message carriers compute the utility for a rely based on this information, and choose a relay that has a higher utility. MDH nodes also share their stored messages information when they encounter by using *MSG_VECTOR* notification, and this information is used to avoid replicating a message to a relay that already has a copy of that message.

D. Interaction with MANETs

MeDeHa++ allows integration of infrastructure-less networks including MANET routing protocols and does not require any change to existing routing protocols to work with them. This allows GW nodes to get a multi-hop connectivity information about MANET nodes when they are connected to a MANET network. The GW nodes are also capable of using the multi-hop node information to discover other GW nodes in the MANET and to use the underlying MANET network as a bridge to connect networks that are otherwise disconnected.

IV. MEDEHA++ WITH MULTIHOP AD-HOC NETWORKS

A. MANET Information Exchange

The presence of a MANET at a GW node is detected by neighbor sensing procedures of MANET routing protocols (e.g., receiving a “hello” broadcast), and is notified to the MeDeHa++ routing component, which starts looking up the MANET routing table to get the information about the available MANET neighbors, and forwards any stored messages to the MANET nodes. Also, each time that the MANET routing table is changed at the GW node, a notification is sent to the node’s MeDeHa++ routing component. Thus, the GW node

consults the MANET routing table to keep information about all available MANET nodes, and treats them as immediate neighbors. Note that nodes form a MANET whenever two or more MANET-capable nodes approach each other.

The GW sends *MANET_NEIGHBORS* notifications to other encountered MDH nodes that are not participating in the MANET. In this way, MeDeHa++'s 2-hop ad-hoc protocol is utilized, and MDH nodes assume that all MANET nodes announced by the GW are 2-hop away. Thus, they are able to forward any stored messages for MANET nodes via the GW (e.g., MDH-1 in Fig. 4 considers MANET-3 as 2-hop away via GW-1).

The GW node also keeps track of history of past encounters for MANET nodes over a period of time and passes this information to other MDH nodes when it meets them using *RECENT_NEIGHBORS* notification. This helps MDH nodes to choose the announcing GW as a relay for stored messages, and forward the message to the GW if the latter fulfills a particular utility function being used as relay selection strategy (e.g., if the GW node has seen a MANET node a specific number of times).

As soon as a GW node is associated to an infrastructure-based node (e.g., an AP), it passes information about all MANET nodes to the AP using *MANET_PRESENT* notification. As a result, the AP forwards stored messages to the MANET nodes via the GW, and also sends an *INDIRECT_ASSOC* notification to all connected APs within the ESS. Moreover, a GW also sends a *LEAVE_MANET* notification to the AP, when it leaves a MANET network, so that the AP removes route information of the MANET nodes. When a GW node leaves, the AP will remove routes for all nodes that were accessible only through the departing GW.

B. Gateway Discovery in MANETs

GW nodes in a MANET use the MANET nodes connectivity information to discover other GW nodes, and exchange data and control information about other networks. This helps in treating MANETs as “transit networks” to transfer MeDeHa++ protocol information across different networks. The discovery is performed by sending MeDeHa++ *HELLO* messages periodically to the MANET nodes to inquire if any node supports MeDeHa++³, and is done on the top of the MANET protocol, so the routing protocol does not require to be modified. Once a GW node discovers another GW, two GW nodes can talk to each other to exchange other nodes information (e.g., current and past neighbors, messages stored) over multihop as if they are direct neighbors, using regular MeDeHa++ protocol. Exchange of data messages between two GW nodes that are multihop away in a MANET cloud is performed using IP encapsulation.

C. Proactive vs. Reactive MANET Routing

A MANET routing protocol does not require any modification while working with MeDeHa++, though the performance

³In MANET routing protocols where a mechanism to discover a gateway joining more than one network is already present (e.g., HNA control messages in OLSR), GW discovery overhead can be reduced by contacting only the gateway nodes to check whether they support MeDeHa++.

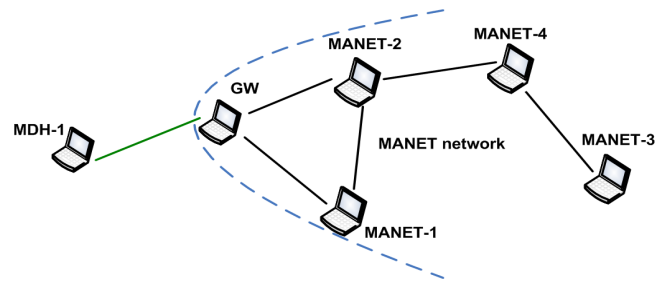


Fig. 4. GW node acts as a bridge to provide communication between MANET nodes and MDH nodes

may vary with the choice of a particular MANET routing protocol. A GW node running a reactive routing protocol such as AODV, may not have complete information about all MANET nodes, at the time when it encounters a MDH node. It only has information about the nodes for which a route request has recently been sent, or about the nodes for which the GW node is a source. Whereas, a proactive protocol does a better job with MeDeHa++, because of the availability of the complete route information at the time the two nodes meet. Therefore, we selected the Optimized Link State Routing (OLSR) protocol to incorporate MeDeHa++. In this way, when a GW node joins a MANET network, it passes the route information to the MeDeHa++ routing component as soon as it learns about the MANET nodes. Also, when this GW node encounters a MDH node, it immediately forwards the MANET route information to the latter using the *MANET_NEIGHBORS* notification. The OLSR protocol also helps in finding GW nodes in MANETs using Host and Network Association (HNA) messages, which is used to announce non-OLSR interfaces of each node.

D. Message Delivery to MANETs

As mentioned earlier, MeDeHa++ is able to deliver messages to regular MANET nodes via GW nodes. Fig. 4 shows how a GW node is used to bridge MDH nodes to MANET nodes. The GW node also passes utility function metrics (e.g., encounter history with MANET nodes) to other MDH nodes that it meets using *RECENT_NEIGHBORS* notification. So, if a source (or a relay) carrying a message for a MANET node encounters a GW node, it may forward stored (or generated) messages to the destination via the GW node if the latter has the destination node in its MANET routing table. GW nodes may also hand over a stored message to a MDH node, if the MDH node is selected as a relay for the message. An infrastructure-based node such as AP will forward messages to the MANET via an associated GW. Messages that are stored for a long time at a source (or a relay) are eventually expired.

E. Message Delivery across MANETs

Multihop communication between two GW nodes is possible by using a MANET routing protocol. In this way, a GW node treats the other GW node as if they are direct neighbors and both GW nodes exchange information about other

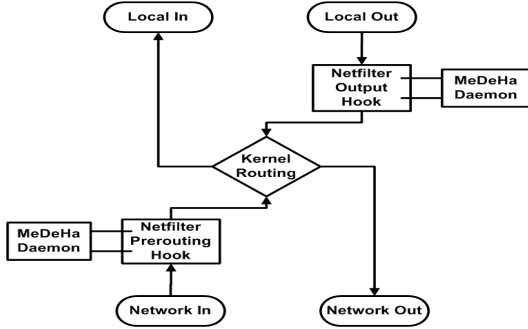


Fig. 5. MeDeHa++ real implementation. Both Incoming and Outgoing packets are intercepted for processing before being passed to Linux kernel

networks. This exchange of information is performed using MeDeHa++ notification protocol control messages. These GW nodes can then advertise the availability of other networks (MDH nodes) to the infrastructure-based network to which they are connected or to other MDH nodes they encounter (Fig. 2). Besides exchanging the network control information, the nodes can forward/receive data packets using IP encapsulation. This enables us to provide message delivery in between networks that do not have any connectivity except that they may be joined by MANETs.

When using OLSR, nodes that belong to different networks via multiple interfaces are detected by OLSR HNA announcements. Once, a GW node receives a HNA announcement, it tries to contact the node that transmits this HNA by sending a MeDeHa++ *HELLO* message to this node. If the other node is also a GW node, the two nodes exchange their neighborhood information via MeDeHa++ neighborhood exchange.

V. IMPLEMENTATION

We implement the MeDeHa++ framework on both real machines as well as the ns-3 network simulator. This approach along with the emulation and real-time scheduling available in ns-3 allows us to create hybrid scenarios that contain both real and simulated nodes simultaneously.

A. Testbed Implementation

Fig. 5 shows the development approach that we take to implement MeDeHa++ for the physical testbed. To achieve high portability and compatibility with the existing infrastructure, the notification protocol is implemented at the network layer as a Linux user-space daemon. All required MeDeHa++ information is included as part of the IP header (as IP option) and no transport or application data is modified (Fig. 6). This allows MDH nodes to function over existing networks with existing protocols.

The Linux implementation uses Netfilter [16] to hook into the Linux protocol stack with a kernel module and pass packets to the user-space daemon for further processing. As shown in Fig. 5, all incoming and outgoing packets are intercepted before passing through the kernel routing algorithm. The daemon determines whether a packet should be buffered or forwarded based on whether a connected next hop destination

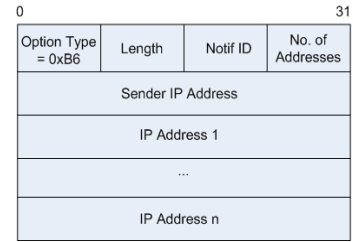


Fig. 6. MeDeHa++ notification header implemented as IP option header

exists. Connectivity information must also be used to manage the kernel routing table and continue to accept packets from user applications even if it appears that connections are disrupted. Neighborhood information in infrastructure networks is determined through a combination of MeDeHa++ control messages and 802.11 management frames.

For MANET networks, we integrate MeDeHa++ with the olsrd 0.6.0 implementation of the OLSR protocol. The MeDeHa++ daemon listens for changes made to the olsrd routing table to determine which nodes are currently accessible via the MANET network. It then exchanges notification messages with other MDH nodes participating in the MANET and shares this information with networks (such as an infrastructure-based network) on other interfaces.

B. Simulator Integration

Our goal of using hybrid networks is to allow more interesting scenarios as well as validate our simulation results. Creating a hybrid network composed of simulated and real nodes has several benefits over simulated or testbed-only networks. Testbed scenarios can be limited by many factors, including size, cost, and limited mobility. While simulated scenarios do not have these constraints, it is not guaranteed that the results are a representation of what would have happened on real hardware. By combining the two approaches, we are able to demonstrate the functionality and scalability of the MeDeHa++ framework on real networks.

We integrate the ns-3 MeDeHa++ implementation with the testbed through the ns-3 emulation and real-time scheduling capabilities. Specifically, we use ns-3 TAP [18] to bridge part of the simulated network to the testbed network. This works by creating a “ghost” node on the ns-3 network that passes all Ethernet frames between a Linux TAP device on the real machine and the simulated links to which the node is connected. Packets can then be routed between the simulated network and the networks to which the real machine is connected. To our knowledge, there are very few studies that attempt and perform this kind of hybrid experiments.⁴

C. Experimental Setup

The testbed consists of laptops and mobile briefcase devices equipped with 802.11g wireless cards, Linux 2.6, and the

⁴We will provide the source code of our implementation for the camera ready version.

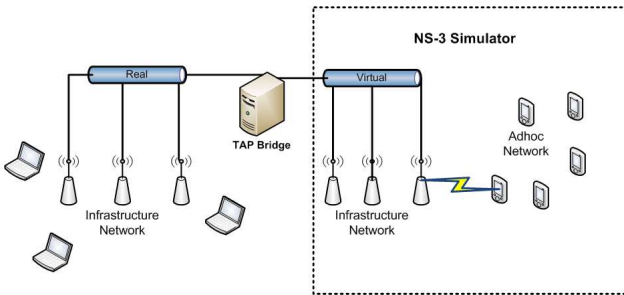


Fig. 7. Hybrid experimentation setup involving real machines acting as APs and stations, and virtual machines running in the NS-3 simulator

MeDeHa++ framework. Depending on the scenario, a number of laptops are configured as access points connected via Ethernet while the remainder are set up as wireless infrastructure stations. In addition, some of the laptops are equipped with an additional wireless interface that can be used to connect to a MANET or ad-hoc network. The mobile briefcase devices are configured in ad-hoc mode to connect only to a MANET. We use hostapd [17] to implement the wireless AP functionality and olsrd 0.6.0 to provide MANET routing.

Finally, a simulated heterogeneous network (involving infrastructure-based and ad-hoc networks) is connected to the testbed with the ns-3 TAP bridge. As shown in Fig. 7, this creates a larger hybrid network that allows more interesting scenarios beyond the limitations imposed by a physical testbed.⁵ The simulator machine, which is identical to the laptops of the testbed, is configured with an Intel 2.4 Ghz Dual-Core processor and 4 GB of RAM.

VI. PERFORMANCE EVALUATION

We define Packet Delivery Ratio (PDR) as the total number of messages received by all destinations divided by total number of messages sent by all sources, and Average Delay (AD) as the average of deliver delay of all messages that are received by destinations. We take these two parameters as performance metrics to evaluate MeDeHa++. We are also interested in estimating how many nodes are able to attain a certain amount of percentage of message delivery.

A. Scenario 1: Convention Center

We consider a convention center type environment with different rooms and seminar halls spanned over a region of 1000mx1000m, and where connectivity is provided by a network of 9 APs that are connected to each other via Ethernet. Each AP has its specific region of connectivity, and the regions of connectivity of different APs may overlap. Almost 60% of the network is under AP connectivity. The APs are not positioned uniformly, which means that at some places, mobile nodes will have longer periods of disconnection than at some other places. Visitors carrying portable devices may move from one room to another and roam around across multiple

⁵Though the amount of simulated traffic for a hybrid network is more limited than a pure simulation network due to real-time scheduling requirements, we still find them to be a useful supplement to a physical testbed.

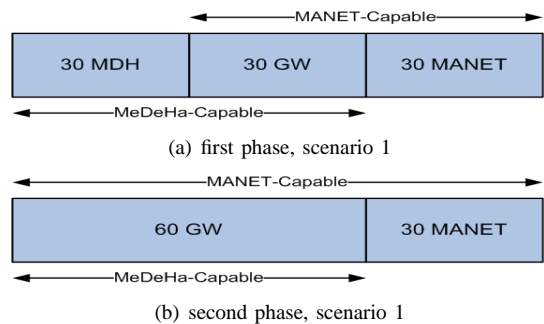


Fig. 8. Types and distribution of nodes used in Scenario 1

AP coverage areas. Also, visitors while moving may make MANETs, and can use MANET connectivity to exchange messages where APs do not provide connectivity.

There is a total of 90 visitors in the convention center, moving at a speed that is uniformly distributed between 1 and 2.5m/sec. While moving, visitors stay at different places for a duration that is uniformly distributed between 0 and 60 seconds. We use the BonnMotion mobility model for nodes mobility [19]. Attraction points [20] are considered as rooms or seminar halls, and nodes visit these attraction points. Each attraction point has its own region of influence that is defined by a standard deviation with zero mean, and corresponds to the maximum distance of visitors to an attraction point. For this experiment, 20 MDH sources are chosen in the network, which send messages at an average rate of 6 messages/minute to 20 non-MDH MANET destinations, and the duration of simulation is 1 hour. The results shown here are obtained by running the experiments 6 times. Among the 90 visitors, 30 visitors are MDH, 30 run the regular OLSR protocol, and the remaining 30 are GW (i.e., they are MDH visitors which also run OLSR protocol), in the first phase of this experiment, as shown in Fig. 8(a).

1) *Forwarding versus Replication*: First, we want to observe the performance of the protocol by comparing forwarding with replication. Forwarding means that only one copy per message exists in the network and a node hands over the ownership of a message, when it delivers the message to a relay, while replication means that there are more than one copy of a message in the network at a given time. For this experiment, we use 2 copies per message. Also, we define Encounter-based Replication (ER) as the “utility function” used by nodes to select a relay, in which a node is selected as relay for a given destination if it has encountered the destination at least twice and it has seen the destination more recently than the node that currently carries the message. The purpose is that depending upon the number of past encounters, a node has a strong probability of encountering a destination in the future. But this may not be fruitful if nodes do not encounter each other too often because of their mobility patterns. Fig. 9 plots the percentage of nodes against delivery ratio comparing forwarding and 2-copy replication.

Forwarding vs. 2-Copy Replication using Encounter-based Replication (ER)
- 6 messages/minute

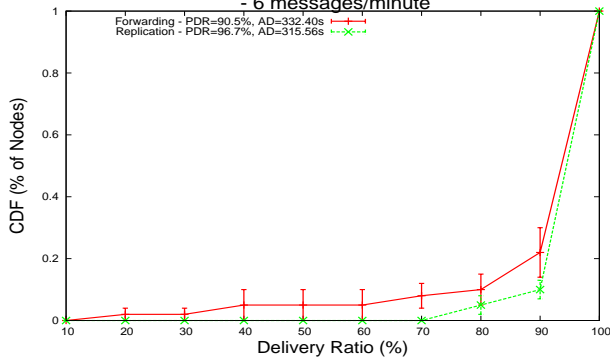


Fig. 9. Forwarding vs. 2-copy Replication using ER scheme for 1st phase of scenario 1 (30 MDH, 30 GW, 30 OLSR visitors)

Encounter-based vs. Social Affiliation-based Replication
- 6 messages/minute

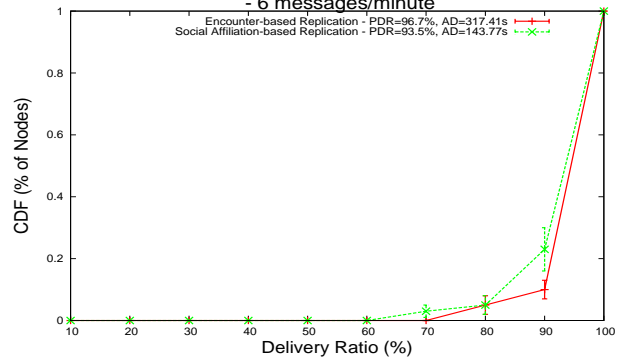


Fig. 10. Comparison between ER and SAR schemes using 2-copy replication for 1st phase of scenario 1 (30 MDH, 30 GW, 30 OLSR visitors)

We see that with forwarding scheme, about 25% of nodes have less than 90% of delivery ratio, as compared to the replication scheme where only 12% of nodes have less than 90% of delivery ratio. While looking into the overall PDR of all 20 nodes, we observe that replication increases delivery chances (from 90% to 97%), while minimizing end-to-end AD. This is because using one more copy of a message would increase the likelihood that a source (or a relay) encounters another relay (or a destination). This is done at the cost of increasing packet overhead, thus requires more resources at nodes. Note that the AD shown is only taken for the messages that are received both in forwarding and replication experiments.

2) *Relay Selection Strategy*: A “suitable” selection is very important as a good choice of a relay can help in both improving message delivery ratio and reducing average end-to-end delay. In this section, we show a comparison of different relay selection schemes with respect to average delivery ratio and average delivery delay. We divide 60 MANET-capable visitors in 3 groups (20 visitors each) by labeling them to different MANET identifiers. Here, we define another relay selection strategy which we name as Social Affiliation-based Replication (SAR), in which we choose “group affiliation” of nodes as utility function for selecting relays. So, a node chooses a relay only if the relay is a member of the same group to which the destination belongs. This utility function is meaningful here since in order to pass the traffic to MANET nodes that are otherwise inaccessible, we have to rely on nodes that belong to these MANET networks, and thus visit them off and on. Thus, it is useful to forward a message to a visiting node for a destination if both destination and visiting node belong to the same group (i.e. MANET, in our case). A comparison between ER and SAR relay selection approaches using 2-copy replication is shown in Fig. 10.

We observe some interesting behavior here. First, using ER, only 10% of nodes have less than 90% of delivery ratio, whereas about 25% of nodes have less than 90% of delivery ratio in case of using SAR. Second, in terms of average PDR, ER performs slightly better than SAR (an increase from 93.5%

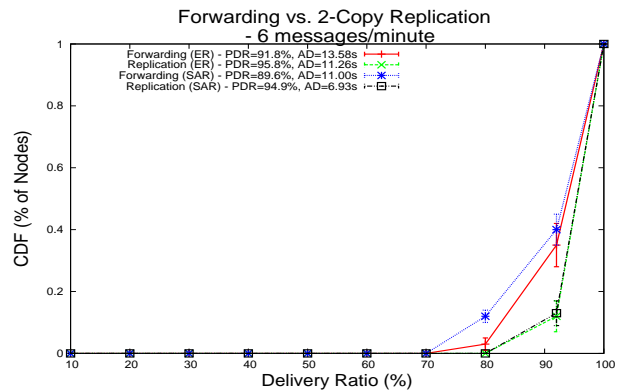


Fig. 11. Forwarding vs. 2-copy Replication using ER and SAR schemes for 2nd phase of scenario 1 (60 GW, 30 OLSR visitors)

to 96.7%). On the other hand, SAR outperforms ER in terms of AD, reducing delay to more than half. Again, note that the AD shown is only taken for the messages that are received using both ER and SAR.

The reason for increase in AD in case of ER over SAR is because of the strict relay selection metric employed in ER, where a relay is chosen only if it has encountered a destination at least twice in the past. This implies an increase in delay but also an increase in average PDR. But on the other hand, there is very little initial delay in forwarding a message to a relay in case of SAR, the message can be forwarded to any node that belongs to the destination’s group.

Next, we slightly change this scenario and make all 90 visitors MANET-capable of which 60 nodes are GW, as shown in Fig. 8(b). The visitors follow the same mobility pattern as before. The result obtained for a comparison between forwarding and 2-copy replication is shown in Fig. 11.

Here, we used both ER and SAR to show a comparison between forwarding and replication. The result is consistent with what we obtained in Fig. 9. The only interesting point here is the drastic decrease in AD. This is due to the increase of MANET participating nodes, which make MANET networks more often than what we have in the previous case. A comparison between ER and SAR is also shown in Fig. 12.

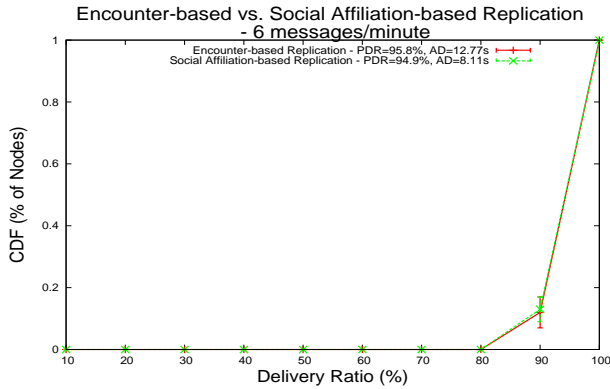


Fig. 12. Comparison between ER and SAR schemes using 2-copy replication for 2nd phase of scenario 1 (60 GW, 30 OLSR visitors)

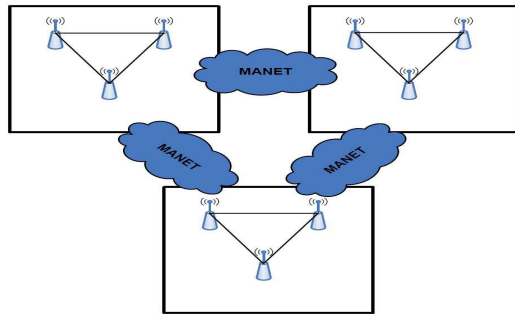


Fig. 13. Scenario 2: Three communities with GW nodes are joined by three "transit MANETs".

Again, the behavior is consistent with what we obtained in Fig. 10, i.e., increase in average PDR and increase in delay while using encounter based replication (ER), as compared to SAR. The only difference is the drop in AD due to the reason mentioned above.

B. Scenario 2: Community Intercommunication with MANETs

In this scenario, we consider that there are 3 different communities; each community is comprised of 600m x 600m area, and has 20 GW mobile nodes and 3 APs. The APs which are in the same community are connected to each other, and thus run MeDeHa++ notification protocol to exchange connectivity information about nodes. The APs do not provide connectivity everywhere in a community. The GW nodes do not move out of their respected communities, and move according to the mobility model mentioned earlier. These communities are not connected to each other except via three "transit MANETs", as shown in Fig. 13. This implies that if a source in one community wants to send a message to a destination in another community, it has to rely on the "transit MANET" that joins the two communities. Each "transit MANET" is comprised of 10 nodes, 8 of which are non-MDH mobile nodes and 2 others are GW that are static.

We carry out a comparison between forwarding and replication in this environment, and the result obtained for fraction of nodes attaining a specific amount of delivery ratio is

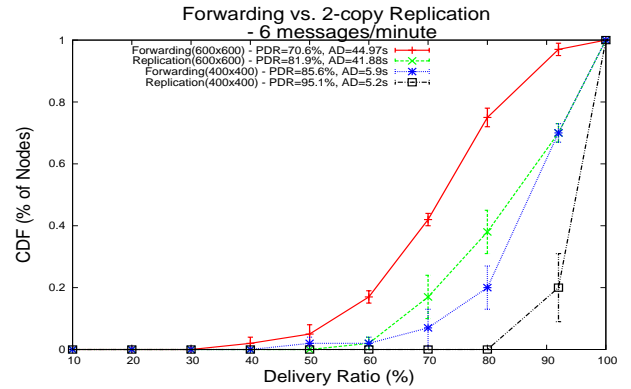


Fig. 14. Forwarding vs. 2-copy Replication using ER scheme for scenario 2

shown in Fig. 14. There are 20 sources chosen from all three communities, which send messages to destinations that do not belong to their own communities. It is obvious that regular MeDeHa framework would yield 0% PDR in this case as the source-destination pairs are only connected through MANETs. The simulation time is 1 hour, and the average message rate is 6 messages/minute. The result is obtained by running the experiment 6 times.

We observe that with forwarding, more than 75% of nodes have less than 80% delivery ratio, as compared to replication which yields that only 40% of nodes have less than 80% of delivery ratio. The average PDR is also improved significantly using replication (82%) over forwarding (71%). Also, AD improves by almost 3 seconds. We are not close to 100% of PDR in this scenario as the only connection between source-destination pairs is MANETs, and depending upon the mobility of nodes, they may never encounter MANET GWs to pass the traffic across MANETs which affects the PDR. We verify this by reducing the community areas to 400mx400m, and notice that average PDR is more than 95% for replication and 86% for forwarding. The AD is also reduced quite significantly (Fig. 14).

We proceed to play with ER scheme to see the impact of changing the encounter threshold, and used number of encounters as 2 and 4 for both forwarding and 2-copy replication. A comparison of forwarding and 2-copy replication is shown in Fig. 15.

We see that average PDR slightly improves for both forwarding and replication while using encounter parameter as 4, but on the other hand, it slightly increases AD. This is because when choosing encounter parameter as 4, nodes have to wait slightly more to find a suitable relay, which increases AD but improves average PDR, as relay selection is more accurate. On the other hand, choosing a high value of encounter parameter also decreases number of message forwarding.

We also evaluated the impact of number of copies per message on PDR for this scenario but did not observe any significant improvement in average PDR with the increase in number of copies, though the decrease in AD was notable.

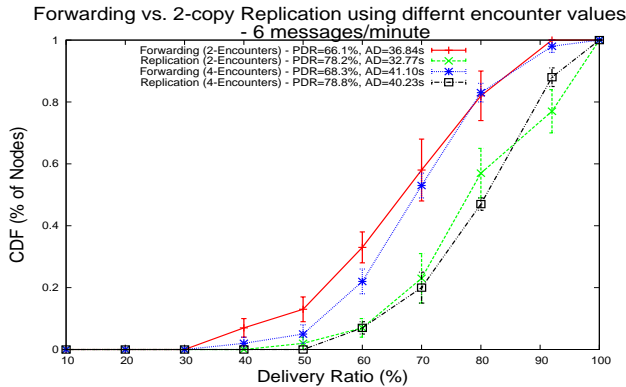


Fig. 15. Impact of different encounter parameters on fraction of nodes while comparing forwarding and replication for scenario 2

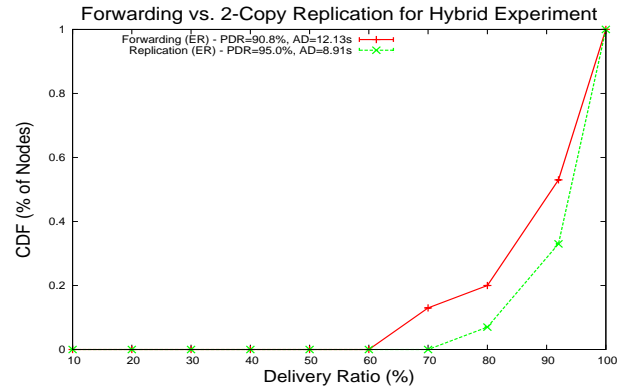


Fig. 17. Forwarding vs. 2-copy Replication comparison resulting from a hybrid scenario involving real and simulation machines

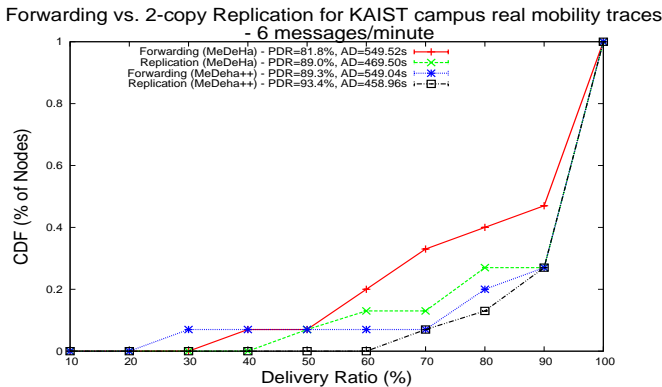


Fig. 16. Forwarding vs. 2-copy Replication showing a comparison between MeDeHa and MeDeHa++ using KAIST mobility traces for 40 nodes

C. Scenario 3: KAIST Real Mobility Traces

We evaluate MeDeHa++’s performance with real traces from KAIST campus available from CRAWDAD [21]. Here, we took a subset of student mobility traces across the campus. It includes 2 hours of mobility from 10 A.M. to 12 P.M. of 40 students for an area of 1.2 km x 1.5 km. We placed 9 APs in the area by looking at department positions at KAIST, with all APs connected to each other. Students either take campus shuttles to move from one area to other, move at pedestrian speed, or do not move at all. We choose 15 students sending data at an average rate of 6 messages/minute 15 other students across the campus⁶, and provide a comparison between the results obtained using MeDeHa++, and using regular MeDeHa protocol. Using OLSR, students that approach each other form small MANETs when moving across the campus and thus able to exchange data and control messages over multiple hops. The comparison between forwarding and 2-copy replication using MeDeHa with 2-hop ad hoc, and MeDeHa++ is shown in Fig. 16.

The behavior is consistent with what we obtained for other scenarios, i.e., there is a marked improvement in PDR and a

⁶we have also run the experiment for file transfer between students and observed similar results.

decrease in AD for replication over forwarding. Moreover, 2-copy replication using MeDeHa++ yields the best result, where PDR is improved to a great extent, while AD is decreased. This is because students form small MANETs while moving, thereby have a larger view of the network most of the times, which allows them to exchange messages faster and efficiently.

D. Scenario 4: Hybrid Experiment Results

Our testbed consists of 7 laptops and 2 mobile briefcases equipped with 802.11g wireless cards: 4 of the laptops are configured as wireless stations and the other 3 laptops are set up as AP routers connected over a wired network, while 2 briefcases and one of the 3 wireless stations (GW station) run the OLSR protocol. During the experiment, wireless stations move and change connectivity with different APs; OLSR briefcase also move and make OLSR network, and are accessed via the GW station. While moving, stations also remain disconnected for some period of time when they are in a region of no connectivity. All 3 APs are connected to simulated APs via a machine that runs ns-3 and acts as a Tap bridge to the ns-3 nodes. In the simulator, we use 30 stations along with 6 APs. Stations in the simulator use the same mobility pattern as described in Section VI-A.

In the experiment, there are a total of 15 source-destination pairs sending data at an average rate of 6 messages/minute, out of which 10 pairs are present inside the simulator, 2 simulator nodes sending data to 2 wireless stations (laptops), and 1 simulator node is sending data to an OLSR briefcase. The two remaining sources are wireless stations that send data to 2 simulator nodes. We compare 1-copy forwarding against 2-copy encounter-based replication and run this experiment for a period of 30 minutes. The results are shown in Fig. 17. We have also conducted other experiments, and some results are presented in [14].

As observed from earlier simulation results, we see that 2-copy replication performs better than 1-copy forwarding both in terms of PDR and AD. Also, while looking at individual delivery ratios of nodes, only 6% nodes have less than 80% delivery ratio with 2-copy replication, as compared to 20% nodes having less than 80% delivery ratio. While comparing

the results obtained using this “hybrid” experiment, we see that the behavior of MeDeHa++ is similar to what we got with pure simulation results in previous sections, which validates our simulation results.

VII. RELATED WORK

In the past, several studies have been proposed to make MANETs impermeable to connectivity disruptions, which either propose a completely new protocol [1], [2], [3], or patch existing MANET protocols [4], [5], [6]. While these solutions offer disruption tolerance to MANETs, they do not deal with network heterogeneity, nor they provide backbone connectivity. On the other hand, MANET solutions that handle network heterogeneity to some extent (by providing backbone or Internet connectivity through gateways [8], [9]) fail in the presence of frequent or long-lived connectivity disruptions.

A number of solutions support partial heterogeneity either by augmenting the coverage area of infrastructure-based wireless networks [10], by providing MANETs with Internet access [8], or by handling heterogeneity at a higher layer (e.g., Bundle Architecture [15]). There are a few architectures that address message delivery in heterogeneous networks. Notable examples include EDIFY [11] and CCN [12]. While EDIFY mainly targets the identification problem in a disruption tolerant environment, and CCN deals with naming the content rather than nodes, both lack true heterogeneous support (treating some specific networks in specific environments). Episodic connectivity in EDIFY is assumed to be provided by mobile message ferries that carry traffic for other nodes, whereas the performance of CCN 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. As previously pointed out, we developed MeDeHa [13] as an attempt to provide message delivery in heterogeneous networks, but to date, MeDeHa only provides message delivery for internets including infrastructure-based wireless and 1-hop ad-hoc DTN forwarding.

VIII. CONCLUSION

Providing seamless message delivery in heterogeneous internets comprising infrastructure-based and ad-hoc networks is becoming a critical enabling technology for future internets. The contribution of this paper is three-fold: (1) we introduced a flexible mechanism to bridge together infrastructure-based and infrastructure-less networks while supporting episodic connectivity; (2) validated the proposed mechanisms using extensive simulations using a variety of scenarios, including real mobility traces; and (3) developed an implementation on Linux, and conducted “live-” and “hybrid” experiments, which ran partly on simulator and partly on real nodes.

Directions for future work include a naming scheme for MeDeHa++ as well as multipoint message delivery. Currently, we assume that each node has an IP address per interface and a source uses any of the IP addresses of a destination to communicate, and the destination is able to receive messages

on any of its interfaces. We believe that an identification scheme is required for better performance of MeDeHa++, where destinations are named by persistent identifiers.⁷

REFERENCES

- [1] Y. Zhu and X. Wu, *Mobility Assisted Routing Strategy (MARS) for Hybrid Ad Hoc Networks*, in Proceedings of 4th International Wireless Communications and Mobile Computing Conference (IWCMC), 2008.
- [2] L. Ding, B. Gu, and X. Hong, *Routing Across Colonies in Delay Tolerant Networks*, Technical Report No. TR-2009-02, Computer Science, University of Alabama, 2009.
- [3] N. Sarafijanovic-Djukic, M. Piorowski, and M. Grossglauser, *Island Hopping: Efficient Mobility-Assisted Forwarding in Partitioned Networks*, in Proceedings of IEEE SECON’06, 2006.
- [4] C. Liu and J. Wu, *Efficient Adaptive Routing in Delay Tolerant Networks*, in Proceedings of IEEE International Conference of Communications (ICC), 2009.
- [5] C. Kretschmer, S. Rhrup, and C. Schindelhauer, *DT-DYMO: Delay-tolerant Dynamic MANET On-demand Routing*, in Proceedings of 29th IEEE International Conference on Distributed Computing Systems Workshops, 2009.
- [6] J. Ott, D. Kutscher, and C. Dwertmann, *Integrating DTN and MANET Routing*, in Proceedings of ACM SIGCOMM workshop on Challenged Networks (CHANTS), 2006.
- [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] A. Hamidian, U. Korner, and A. Nilsson, *Performance of Internet Access Solutions in Mobile Ad Hoc Networks*, Wireless Systems and Mobility in Next Generation Internet/Lecture Notes in Computer Science, First International Workshop of the EURO-NGI Network of Excellence, 2004.
- [9] K. R. Khan, A. V. Reddy, and R. Zaman, *An Efficient Integrated Routing Protocol for Interconnecting Mobile Ad Hoc Networks and the Internet*, in Proceedings of ACM International Conference on Advances in Computing, Communication and Control (ICAC309), 2009.
- [10] C. Parata, G. Convertino, and V. Scarpa, *Flex-WiFi: A Mixed Infrastructure and Ad-Hoc IEEE 802.11 Network for Data Traffic in a Home Environment*, In Proceedings of the First IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC), 2007.
- [11] M. Chuah, L. Cheng, and B. Davison, *Enhanced Disruption and Fault Tolerant Network Architecture for Bundle Delivery*, in Proceedings of IEEE Globecom, 2005.
- [12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. L. Braynard, *Networking Named Content*, in Proceedings of ACM CoNext, 2009.
- [13] R.N.B. Rais, T. Turletti, and K. Obraczka, *Coping with Episodic Connectivity in Heterogeneous Networks*, in Proceedings of 11th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2008.
- [14] M. Mendonca, R.N.B. Rais, T. Turletti, and K. Obraczka, *Message Delivery in Heterogeneous Disruption-prone Networks*, in ACM Mobicom 2010.
- [15] K. Scott and S. Burleigh, *RFC 5050, Bundle Protocol Specifications*, IRTF DTN Research Group, 2007.
- [16] *Netfilter*, <http://www.netfilter.org/>.
- [17] *Hostapd*, [hostapdhttp://hostap.epitest.fi/hostapd/](http://hostap.epitest.fi/hostapd/).
- [18] *NS-3 Tap Bridge*, http://www.nsnam.org/doxygen-release/group__tap_bridge_model.html.
- [19] BonnMotion, University of Bonn, *A Mobility Scenario Generation and Analysis Tool*, <http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion>.
- [20] M. Feeley, N. Hutchinson, and S. Ray, *Realistic Mobility for Mobile Ad Hoc Network Simulation*, Ad-Hoc, Mobile, and Wireless Networks, Lecture Notes in Computer Science Vol. 3158, 2004.
- [21] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim and S. Chong, *CRAW-DAD data set ncsu/mobilitymodels (v. 2009-07-23)*, Downloaded from <http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels>, 2009.

⁷This work is currently in progress.