MeDeHa - Efficient Message Delivery in Heterogeneous Networks with Intermittent Connectivity

Rao Naveed Bin Rais, Thierry Turletti, Katia Obraczka

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

MeDeHa - Efficient Message Delivery in Heterogeneous Networks with Intermittent Connectivity

Rao Naveed Bin Rais — Thierry Turletti
— Katia Obraczka

N° 7227

March 2010

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7227--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SOPHIA ANTIPOLIS - MÉDITERRANÉE

MeDeHa - Efficient Message Delivery in Heterogeneous Networks with Intermittent Connectivity

Rao Naveed Bin Rais * , Thierry Turletti †
, Katia Obraczka ‡

Thème : Computer Networks
Équipes-Projets Planète

Rapport de recherche n° 7227 — March 2010 — 28 pages

Abstract: In this report, we present an efficient message delivery mechanism that enables distribution/dissemination of messages in an internet connecting heterogeneous networks and prone to disruptions in connectivity. We call our protocol MeDeHa (pronounced "medea") for Message Delivery in Heterogeneous, Disruption-prone Networks. MeDeHa is complementary to the IRTF's Bundle Architecture: while the Bundle Architecture provides storage above the transport layer in order to enable interoperability among networks that support different types of transport protocols, MeDeHa is able to store data at any layer of the network stack, addressing heterogeneity even at lower layers (e.g., when intermediate nodes do not support higher-layer protocols). MeDeHa also takes advantage of network heterogeneity (e.g., nodes supporting more than one network and nodes having diverse resources) to improve message delivery. For example, in the case of IEEE 802.11 networks, participating nodes may use both infrastructure- and ad hoc modes to deliver data to otherwise unavailable destinations. Another important feature of MeDeHa is that it does not rely on special-purpose nodes such as message ferries, data mules, or throwboxes in order to relay data to intended destinations, and/or to connect to the backbone network wherever infrastructure is available. The network is able to store data destined to temporarily unavailable nodes for some time depending upon current storage availability as well as quality-of-service needs (e.g., delivery delay bounds) imposed by the application. We showcase MeDeHa's ability to operate in environments consisting of a diverse set of interconnected networks and evaluate its performance via extensive simulations using a variety of synthetic– as well as more realistic scenarios. Our results show significant improvement in average delivery ratio and significant decrease in average delivery delay in the face of episodic connectivity. We also demonstrate MeDeHa's support for

* Projet Planète, INRIA, Sophia Antipolis, emailNaveed-Bin.Rais@sophia.inria.fr
† Projet Planète, INRIA, Sophia Antipolis, emailThierry.Turletti@sophia.inria.fr
‡ University of California, Santa Cruz, USA, emailkatia@soe.ucsc.edu

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

different levels of quality-of-service through traffic differentiation and message prioritization.

Key-words: Disruption tolerance, Episodic connectivity, Heterogeneous networks, Node relaying, Store-carry-and-forward

MeDeHa - La distribution efficace des message dans les réseaux hétérogènes avec la connectivité intermittent

Résumé : Dans ce rapport, nous proposons MeDeHa, une architecture de communication pour la distribution de messages dans les réseaux hétérogènes à connectivité intermittente. Le protocole MeDeHa est complémentaire de l'architecture Bundle de l'IETF, dans laquelle le stockage est effectué au niveau de la couche transport, afin de permettre l'interopérabilité entre les réseaux hétérogènes. En revanche dans l'architecture MeDeHa, les nuds peuvent stocker les données au niveau de n'importe quelle couche réseau. L'architecture MeDeHa permet aussi de bénéficier de l'hétérogénéité des nuds. Par exemple, les nuds avec plusieurs interfaces réseau et les nuds avec davantage de ressources permettent d'améliorer les performances du protocole. Nous analysons les performances de MeDeHa dans différents environnements par des simulations avec des modèles de mobilité synthétique et réels.

Mots-clés : La tolerance à perturbation, La connectivité intermittent, Des réseaux hétérogènes, Des réseaux infrastructures

Contents

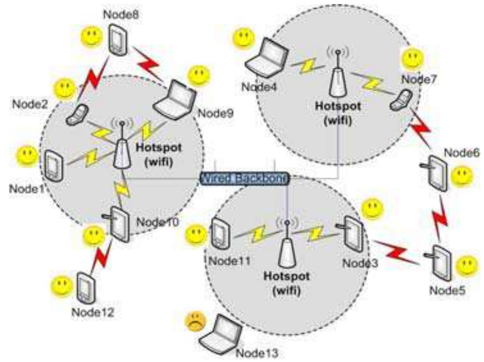

Figure 1: An example of a heterogeneous internetwork with a wired backbone, wireless infrastructure-based, and ad-hoc networks prone to episodic connectivity

1 Introduction

It is envisioned that the Internet of the future will be highly heterogeneous not only due to the wide variety of end devices (in terms of their capabilities, e.g., storage, processing time, battery lifetime, mobility, and traffic characteristics) it interconnects, but also in terms of the underlying networks it comprises. As illustrated in Figure 1, such networks range from wired- and wireless backbones (e.g. community wireless mesh networks) to wireless infrastructure-based and ad-hoc networks (MANETs). Furthermore, current and emerging applications, such as emergency response, environmental monitoring, smart environments (e.g., smart offices, homes, museums, etc.), and vehicular networks, among others imply frequent and arbitrarily long-lived disruptions in connectivity. The resulting disruption- or delay-tolerant networks (DTNs) will likely become an integral component of future internetworks.

Seamless interoperability among heterogeneous networks is a challenging problem as these networks may have very different characteristics. Also, node diversity may make routing difficult, as nodes must also take into account available resources at other nodes along with contact opportunities (given that links are time-varying due the possibility of intermittent connectivity) in order to make correct routing decisions. For instance, in a buffer-constrained network where participating nodes may have different buffering capabilities, it is useless to forward a message to a neighboring node, if the latter is running out of buffer space.

As will become clear in Section 6, which describes related work, to-date, there are no comprehensive solutions targeting message delivery in heterogeneous networked environments prone to connectivity disruptions. Existing proposals either: (1) extend MANETs to handle episodic connectivity [1], [2], [3], [4], (2) augment the coverage of access points in infrastructure-based wireless networks by, for example, making use of multi-channel radios or switching from infrastructure mode in 802.11 [5], [6], [7], [8], (3) provide MANETs with Internet connectivity by using special-purpose gateway nodes and a mechanism to discover them as part of route discovery in on-demand MANET routing [9], or

(4) handle heterogeneity only at higher layers of the protocol stack (e.g., Bundle Architecture [10], [11]).

In this report, we propose MeDeHa (Message Delivery in Heterogeneous, Disruption-prone Networks, pronounced "medea") - a general, yet efficient framework for data delivery in heterogeneous internets prone to disruptions in connectivity. To cope with arbitrarily long-lived connectivity disruptions, we use available storage within the network to save messages for destinations that are currently unreachable; once these destinations re-connect, messages destined to them get delivered. While the Bundle Architecture provides storage above the transport layer (in order to enable interoperability among networks that support different types of transport protocols), in MeDeHa, storage can be provided at any layer of the communication stack (application, network, link etc.). Thus, MeDeHa can be supported by any (intermediate) node including ones that that do not run higher-layer protocols (e.g., access point bridges, relay nodes, etc.). MeDeHa is also able to provide different levels of quality-of-service through traffic differentiation and message prioritization by controlling when messages are forwarded and for how long they are stored.

Besides, unlike existing proposals such as message ferries [12], data mules [13], or throwboxes [14], MeDeHa does not rely on any special-purpose nodes. Note that there is a difference between introducing special-purpose nodes in the network to perform the task of relaying (like message ferries [12], data mules [13], and throwboxes [14]) and making use of existing nodes with special capabilities (e.g., access points, or APs in the case of infrastructure-based wireless networks) that are an integral part of the underlying network. Of course, whenever available, MeDeHa utilizes nodes with more resources and capabilities like APs to perform message delivery more efficiently, but does not count on them. Furthermore, we take advantage of the underlying heterogeneity (e.g., in the context of IEEE 802.11 networks, a nodes' ability to operate in infrastructure or ad-hoc modes) to enable message delivery across different networks.

This report extends our preliminary work presented in [15] where scenarios of limited heterogeneity were addressed and evaluated. Here, we explore significantly higher degrees of network heterogeneity including networks comprising of wired as well as infrastructure- and multi-hop ad hoc wireless networks that are subject to intermittent connectivity. We evaluated MeDeHa through extensive simulations using a variety of synthetic as well as real-world scenarios. Our results show that end-to-end delay can be improved significantly while maintaining high delivery ratio. This is accomplished by selecting appropriate relays when forwarding data, taking advantage of in-network storage as well as node diversity and network heterogeneity (e.g., nodes with more resources, nodes that can switch between infrastructure and ad hoc communication modes).

The remainder of this report is organized as follows: Section 2 provides a overview of MeDeHa's framework while MeDeHa's protocol description is presented in Section 3. The implementation approaches as well as MeDeHa's current implementation is described in Section 4 and Section 5 presents simulation results reporting the performance of MeDeHa using a variety of synthetic– as well as more realistic scenarios. Related work is reviewed in Section 6 and finally, concluding remarks and some future directions are discussed in Section 7.

2 MeDeHa Overview

MeDeHa allows message delivery across heterogeneous networks by accommodating a diverse set of characteristics in terms of mobility, connectivity, and resources. Heterogeneity is supported both at the network– (e.g., allows co-existence of different types of networks like wireless infrastructure-based as well as ad hoc and networks prone to connectivity disruption) and at node level (e.g., nodes with diverse resources like battery power, buffering, mobility characteristics can be part of the network).

MeDeHa embraces node– and network heterogeneity and tries to make use of it whenever possible. For example, MeDeHa tries to take advantage of more resourceful nodes (e.g., APs in IEEE 802.11 infrastructure-based networks) whenever possible and feasible. Additionally, a node that participates in multiple networks will attempt to find a path (or suitable relays) to a destination in all networks of which the node is a member.

MeDeHa's main functional components are:

Message Relaying: Unlike several DTN solutions, which employ specialized nodes to aid with message delivery [12], [13], [14], in MeDeHa any node in the network can relay messages under the store-carry-and-forward paradigm [11]. We thus avoid using any explicit discovery mechanism for finding specialized nodes (e.g., gateway to the backbone). Nodes may also take advantage of network heterogeneity to improve message delivery. For example, 802.11-capable nodes may join different networks by switching between infrastructure– and ad hoc modes. A node can also switch to different frequencies in order to join different networks (this can be done, e.g., using the power save mode of the IEEE 802.11 standard).

Buffering: In an environment with intermittent connectivity, it is necessary to use network nodes to store messages if a route to the intended destination(s) is not available. An important question is where to buffer these messages. In MeDeHa any node can relay messages and therefore store messages whose destination(s) is(are) not available. However, we again try to take advantage of network heterogeneity. For example, Access Points (APs) in infrastructure-based wireless networks or mesh routers in the case of wireless mesh networks, are usually good candidates to serve as temporary storage for undelivered messages as they exhibit higher resource availability [1].

Note that in the Bundle Architecture [10], buffering is performed above the transport layer, which in itself restricts the types of nodes that can perform this functionality. For instance, it rules out APs as buffering nodes, as APs usually run only the lower two or three lower protocol layers. In MeDeHa, buffering can be done at any layer of the communication stack, which enables almost any network-enabled device to relay and buffer messages. This feature makes MeDeHa complementary to the Bundle Architecture, as storage can be provided by MeDeHa-capable nodes at lower layers even if they do not run the higher layers of the communication stack. Moreover, in MeDeHa, quality-of-service is supported by enforcing application-specific requirements at the message for-

[1] It is true that most current off-the-shelf APs do not typically come equipped with mass storage. We argue that adding this capability to next-generation APs is viable and will not considerably impact cost, especially if there is market demand. Furthermore, co-locating a general-purpose computing device with APs is another alternative given current AP technology.

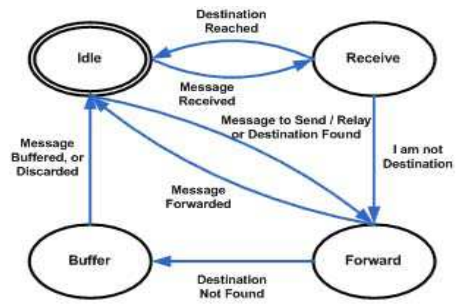

Figure 2: State diagram showing MeDeHa's overall operation. A MeDeHa-capable node can be in one of the four states, *Idle*, *Receive*, *Forward*, and *Buffer*

warding and storage level. For instance, data belonging to real-time flows would be discarded after a pre-defined time interval specified by the application.

Topology and Content Information Exchange: Nodes periodically exchange information that is used in building their routing and contact tables. This information includes a node's knowledge about the topology (e.g., its own neighborhood as well as what it knows about other nodes). Nodes also exchange a summary of their message buffer and their current state in terms of resources (e.g., how much storage left, remaining battery lifetime, etc.). This information is used by relay selection [17], [18], [19] and contributes to the overhead incurred by MeDeHa. Clearly, there is a tradeoff between the overhead incurred by the protocol, how fresh paths are, and how well relay selection performs. Note that if neighborhood information is already made available by the underlying layer-2 protocol (e.g., beaconing, AP association / disassociation in IEEE 802.11 infrastructure mode), MeDeHa simply makes use of it.

Traffic Differentiation: In order to satisfy application-specific needs, MeDeHa uses message tags to carry information such as message priority, time-to-live (or TTL, which is the maximum amount of time the message should remain in the network), scope (e.g., maximum number of hops the message should travel), etc. Besides performing traffic differentiation and supporting quality-of-service, message tags are also used for buffer management purposes. For instance, a message that has been stored pass its TTL would be discarded.

2.1 Overall Operation

Figure 2 illustrates MeDeHa's overall operation.

Idle: By default, a node starts in *idle* state. It switches to *receive* state upon reception of a message, or to *forward* state if it has some message to send. This message can either be generated by this node, or can be the message that the node has stored for some unavailable destination. Thus, in *forward* state, if the destination is not found, the node stores the message and goes back to *idle*. Later if the destination is found, the node goes to *forward* state, delivers the message and changes its state to *idle*.

Forward: When a node has a message to send either as the message originator or relay, it checks if it has a path to the destination, and if so, sends the message along that path and switches to *idle* state. Otherwise, it tries to find

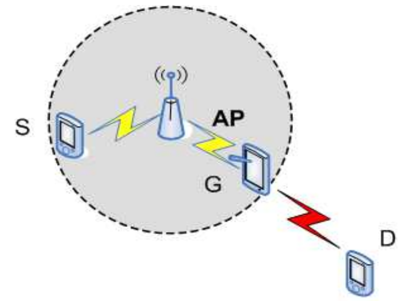

Figure 3: Multi-hop message delivery involving infrastructure-based and "ad hoc" nodes that may be intermittently connected. Source S wants to send a message to destination D. This is made possible with the help of node G that acts as gateway between the two networks. S and D do not need to be connected to more than one network nor be part of the same network in order to send/receive messages.

a "suitable" relay. If it does not succeed, it switches to *buffer* state to store the message locally.

A number of destination-dependent and destination-independent heuristics can be used to select a relay for a *(message, destination)* tuple including: (1) when the node last encountered the destination (or age of last encounter), (2) how frequent the destination was encountered, (3) how mobile a node is, and whether the scope of the mobility is "local" or "global", (4) how "social" a node is, etc. A number of these heuristics/utility functions has been presented in [17]. MeDeHa's framework is flexible enough to employ any kind of utility function for choosing a relay to carry a message to a destination. When selecting relays, MeDeHa can also account for the underlying heterogeneity among participating nodes, e.g., the amount of available resources such as storage, processing, and battery lifetime. For instance, more resourceful entities (like APs) may be preferred when messages need to be stored.

Receive: When a node receives a message and it is not the message's intended destination, it switches to *forward* state and follows the steps described above. Otherwise, the message is passed to the application layer.

Buffer: MeDeHa nodes make use of different buffer management strategies based, for example, on the application QoS requirements such as message priority and time-to-live (TTL).

3 MeDeHa Protocol Description

This section describes in detail the protocol which implements MeDeHa's functional components presented in Section 2 above.

3.1 Notification Protocol

As illustrated in the example of Figure 3, MeDeHa's notification protocol plays a key role in seamless message delivery across multiple heterogeneous interconnected networks. It collects information about a node and its neighborhood and shares that information with other nodes by exchanging *notification messages* (described below). Neighborhood information is then used by MeDeHa-capable nodes to construct their routing/contact tables. In the current MeDeHa implementation, the notification protocol is run at the network layer and is able to work on more than one interface: where each interface may have a different network identifier (e.g., IP address). This and other implementation issues will be discussed in more depth in Section 4.

In the specific example of Figure 3, the access point (AP) gathers two-hop network information from the nodes that are associated to it; it then can use the associated nodes (node G in the example) to forward a message AP carries to a node that is connected through one of the associated nodes (in this case , node D). This particular example shows that MeDeHa extends message delivery beyond the range of access points in infrastructure-based networks to destinations that can only connect (intermittently) on ad-hoc mode.

MeDeHa's notification protocol has itself 2 main components, naming *neighborhood sensing* and *neighborhood information exchange*, which are described in detail below.

3.1.1 Neighbor Sensing

If neighbor detection is provided by the underlying network (e.g., beaconing and management messages in IEEE 802.11 infrastructure-based networks), MeDeHa takes advantage of that information. For instance, in the case of IEEE 802.11 infrastructure mode, a node senses the presence of an nearby AP when it is *associated* with the AP at the link layer. This information is forwarded to the network layer as soon as the presence of a node (a station or an AP) is detected. On the other hand, a link disconnection is detected when a node is *disassociated* with an AP. Thus, in infrastructure-based network, neighbor sensing is performed implicitly with the help of underlying link-layer protocol.

In MeDeHa-capable ad hoc networks, neighbor (or link) sensing is done using *HELLO* notification message exchange. Nodes periodically broadcast *HELLO* notifications in order to inform other nodes in the neighborhood (if any) about each other's presence. In MeDeHa's current implementation, the *HELLO* notification interval is empirically set to 2 seconds, by default. In an effort to minimize the overhead incurred by the protocol, information in *HELLO* messages is kept to a minimum and may include:

- **Node identifier(s) (e.g., IP address):** Nodes may announce multiple identifiers if they have more than one.
- **Infrastructure affiliation indicator:** A flag indicating whether transmitting node is currently affiliated (associated) with an infrastructure based network.
- **Identifier of Infrastructure based node:** In case of affiliation with an infrastructure based network, identifier of the associated AP.
- **Memory status:** Available memory in number of bytes.
- **Energy level:** An indication about the status of the node's current power capacity (e.g., remaining battery life).

- **Social affiliation:** An indication of the social affiliation of the transmitting node (association with a particular community). This can also be used to indicate the mobility characteristics of the node (bus, pedestrian, car etc.).

3.1.2 Neighborhood Information Exchange

A *HELLO* notification only contains information about the *HELLO*-originating node, and not about its neighborhood. As previously mentioned, this is done in order to limit protocol overheard; this is especially beneficial in the case of highly partitioned networks. Having received a *HELLO* notification, a node responds with a *NEIGHBOR_INFO* unicast notification. This completes the handshake between two neighboring nodes and also eliminates uni-directional wireless links implicitly. A *NEIGHBOR_INFO* notification message may contain any combination of the following:

- ***CURRENT_NEIGHBORS*:** List of one-hop neighbor identifiers minus the identifier(s) of the node to which the message is being sent.
- ***RECENT_NEIGHBORS*:** List of node identifiers who have been encountered within a pre-defined period of time. It may also include additional information related to encountered nodes (e.g., number of encounters, encounter time, social affiliation of node, speed of nodes etc.) which are used in computing the *utility function* employed in relay selection (see details in Section 3.3.
- ***MSG_VECTOR*:** List of of application-level message identifiers (sequence numbers, source-destination identifiers and ports, etc.). This notification may be sent in order to avoid forwarding a message to a node (relay) that already has a copy of it. This can be used when a multi-copy replication scheme is used to reduce unnecessary message duplication.

Table 1 summarizes the different notification messages exchanged in MeDeHa-capable ad hoc networks. Note that neighborhood information exchange in ad hoc mode allows each node to keep two-hop neighborhood information.

In the case of infrastructure-based networks, neighborhood information is exchanged between a node and its *associated* AP and among APs that are connected (either wired or wireless). The notification messages between APs are triggered on the reception of an association or a disassociation event (e.g., *NODE_PRESENT*, *NODE_LEAVE* etc.). The notification messages between a node and its associated AP can result from a link layer association of the node (e.g., *ASSOC*), or based on sensing a neighboring node in ad hoc mode (e.g., *NEIGHBOR_PRESENT*). Nodes that pass their one-hop neighborhood information to their associated APs act as gateways to connect infrastructure-based networks with nodes in ad hoc mode. Notification messages that are exchanged in an infrastructure-based network are presented in Table 2. Note that the notification protocol messages exchanged amongst APs are broadcast and confined to APs within an Extended Service Set (ESS).

Table 1: Notification Information Exchanged for Ad hoc Networks

Notification Name	**Includes**	**Contents**	**Description**
HELLO		Node IDs flagAssociated Affiliated AP's ID Buffer level Energy level	Broadcasted by each node periodically to inform neighboring nodes about its IDs
NEIGHBOR_INFO	CURRENT_NEIGHBORS	IDs of neighbors	Sent in response to *HELLO* in order to inform receiving node about neighboring nodes
	RECENT_NEIGHBORS	IDs of encountered nodes Encounter time Number of encounters *Any other heuristic*	Sent in response to *HELLO* to inform receiving node about the nodes recently seen by the transmitting node
	MSG_VECTOR	Sequence no. of messages Source of messages Destination of messages	Sent in response to *HELLO*, and contains sequence numbers of messages stored at transmitting node

3.2 Routing and Contact Table Management

In MeDeHa, each node maintains routing and contact tables which are built using information from neighbor sensing and neighborhood information exchange. MeDeHa routing tables contain forwarding information for nodes that are currently accessible. Using information from *HELLO* and *CURRENT_NEIGHBORS* messages allows nodes to maintain 2-hop routing information for other currently connected nodes. Routing information is updated after each *HELLO* notification exchange. If a node does not hear an update from a node (for which it has a routing entry) for as long as two times the period of *HELLO* exchange, it removes the routing entry from its routing table and stops propagating the node's availability in subsequent *CURRENT_NEIGHBORS* notifications. All entries in the routing table for which the unavailable node is used as a gateway are also removed at this point.

A node's contact table comprises of information about other nodes that are encountered by this node over a pre-defined period of time. The contact table information is then propagated via *RECENT_NEIGHBORS* notifications. The information about a "contact" is entered into the contact table of a node when the node received a *HELLO* notification from a newly connected neighbor. This information comprises of the time at which the contact occurred as well as an encounter counter. This counter is only incremented once during a contact duration (even if nodes exchange more than one *HELLO* notification), and is an indicator of the number of contact opportunities the two nodes have had with each other. Contact table entries of a node are removed when they

Table 2: Infrastructure-based Notification Protocol Messages

Notification Name	Originator	Destination	Description
ASSOC	Node	AP	Notification message sent to nettwork layer as soon as a node is associated with an AP.
NODE_PRESENT	AP	AP	Upon arrival of *ASSOC*, this notification message is sent to all other APs to inform about association.
NODE_LEAVE	AP	AP	This notification message may be sent when a disassociation process is completed (implicit or explicit).
FETCH_FRAMES	AP	AP	On the arrival of a *ASSOC*, an AP may send this notification message to other APs asking about any stored messages.
NEIGHBOR_PRESENT	Node	AP	This notification message is sent from an node to its associated AP, and contains information about immediate neighbors of the transmitting station.
INDIRECT_ASSOC	AP	AP	This notification message is sent on the reception of *NEIGHBOR_PRESENT* to inform other APs about an indirect association.
NEIGHBOR_LEAVE	Node	AP	As soon as departure of a neighboring node is detected, this notification message is sent from an associated node to its AP.

time out. This timeout period is configurable, and depends on how long an information remains useful about a “contact” in a specific environment. A node stops propagating a contact information after this timeout.

3.3 Relay Node Selection and Forwarding

In MeDeHa, selection of a relay node depends upon information advertised by candidate relays and propagated as part of neighborhood information exchange. This information is used to compute the *utility* of the node as a relay. When multi-copy replication is supported, it is mandatory that two nodes exchange *MSG_VECTOR* information as part of *NEIGHBOR_INFO* notifications in order to avoid forwarding an application-level message to a node that already has a copy of the message. The choice of utility metrics for relay selection also depends upon the network environment, node heterogeneity, as well as application-specific requirements.

For instance, in an environment where infrastructure-based networks are available and all APs are connected to each other, providing an “almost connected” network, APs may have high utility as relays when compared to other nodes (see Section 5.2). This is because in such environments handing over a copy of a message to an AP means that the network now contains the number of copies of that application-level message equal to the total number of APs. This increases the probability of message delivery to a destination. Now consider an example where connectivity between different villages is only provided using buses that move between them. In this case, buses would be given priority as relays to carry inter-village traffic (see Section **??**).

Another important parameter in choosing a "suitable" relay node is the buffer capacity (e.g. in bytes) announced by a candidate relay node. If a node has more messages to send than the messages that can be accommodated by a candidate relay node, it could only forward a subset of stored message to the latter node and look for some other relay node to carry the other remaining messages. Similarly, a node's energy level is another parameter to be considered when choosing relay nodes as it may be useless to forward messages to a node who is going to die soon.

Before forwarding an application-level message (or a set of messages) to a relay node, the corresponding route for the destination is entered in the routing table of the node that is forwarding the message with next hop set as the chosen relay. This route remains in the node's routing table until it times out.

To perform data forwarding, MeDeHa employs the hop-by-hop reliability mechanism as specified by the reference DTN architecture [11] which works as follows. When a source (or a relay) encounters a destination (or another relay) for which it carries a few messages, it forwards the messages and considers that a message handover is successful when it receives an acknowledgment (using TCP ack or explicit ack on top of UDP). This makes sure that the message is transfered reliably and that the number of messages transfered are proportional to the contact duration, thus avoiding any unnecessary message loss. Along with providing reliability, this mechanism also serves the purpose of controlling the number of duplicate messages flowing in the network.

3.4 Message Delivery in MeDeHa: An Overall Picture

A source, when having a message to send, consults its routing table to find next-hop information for the message destination. If the information is found, it forwards the message through the specified interface. The message is stored locally on the node, if no information about the destination is found in the routing table. Nodes participating in an infrastructure-based network may use as default route the corresponding AP. Therefore, if a node is currently associated to an AP and has a message to send (forward), but no information about the destination is found in the node's routing table, the node forwards the message to its AP. The AP then consults its routing table for the destination's information and if no information is available, the message is stored locally until information about the destination is received: this information can either be that the destination is connected to the AP (directly or via an associated node), or that a connected node seems to be a better relay to carry a message to the destination. In an network where all APs are connected to each other, and there is only one copy per message, it may be better to keep the message stored at an AP and not forwarding the message from an AP to a relay, as keeping a message stored at an AP increases the chances of message delivery; the message is delivered as soon as the destination's information is found at any AP.

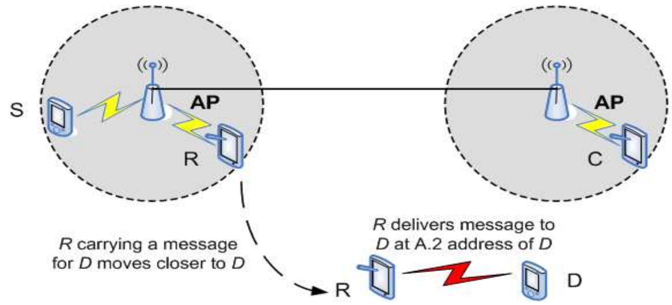

Figure 4: An example of message delivery in heterogeneous networks

When a node (or a relay), carrying a message for a destination, encounters a more "suitable" relay (with the help of *RECENT_NEIGHBORS* exchange), it adds an entry in its routing table for the destination, declaring the relay as its next hop, and forwards messages for that destination to the relay. The routing table entries are refreshed periodically with the help of *CURRENT_NEIGHBORS* and *RECENT_NEIGHBORS* notifications, and all the entries for which there is no update, are removed from the routing table after a timeout. Each node maintains two types of tables, routing and contact table. Forwarding a message to available nodes is performed by looking up the routing table entries. Contact tables are used to maintain utility function metrics for each encountered node within a specific time window. As soon as a node detects that a neighboring node has left its surrounding (if it does not hear from the latter for a period of two hello intervals), it removes the node's entry from its routing table, and updates its contact table entries for the departing station.

Advertising the addresses of all interfaces of a station in *HELLO* notification allows message delivery to any of the available interfaces of a destination. Consider the scenario shown in Figure 4. A source S with two interfaces, I.1 for infrastructure mode and A.1 for ad hoc mode, and a destination 'D' has two interface identifiers I.2 and A.2 for infrastructure and ad hoc mode respectively. S is *associated* to AP *BS1* and has a message to be sent to I.2 address of D, but D is not currently *associated* to any of the APs in the network. A relay R meets D in ad hoc mode, and is able to deliver message to D, because in its hello advertisement, D announces the possession of both I.2 and A.2. Thus, in ad hoc mode, the message from S would be sent to A.2 address of D via R.

4 MeDeHa's Implementation

Our current implementation of MeDeHa performs message delivery in an internet comprised of infrastructure-based and ad hoc wireless networks where mobile nodes roam freely and may become temporarily disconnected. Infrastructure-based nodes (e.g., APs) that are connected to each other form an Extended Service Set (ESS), and thus share network information via notification protocol. While moving, nodes encounter each other in ad hoc mode and exchange control information and data. The messages could be stored at APs as well as at relay nodes. Moreover, when a source moves and finds itself in a region of no connectivity, it starts caching its messages for the destination. In this way, the source stores messages at its end, and as soon as it finds either a destination, or a relay for the destination or an AP, it may start forwarding the messages.

The IEEE 802.11 standard does not define when and how a disassociation process should be initiated, except for the case when an authentification fails. In an ESS where AP regions do not overlap, a station would be disconnected from one AP before associating with another AP in the same ESS. Since in MeDeHa, APs need to know when a station leaves its connectivity region, we extended the IEEE 802.11 implementation to support explicit disassociations. Thus, a node keeps on checking the received power levels of beacons from its associated AP and triggers an explicit disassociation if the power level of the received beacon is less than a certain threshold. This threshold is currently set to 90% of the received power threshold of a station. In order to allow roaming, all APs that are part of the same ESS use the same channel and SSID.

Disassociation messages may get lost (e.g., because the station is already out of the AP's communication range when sending this message, because the corresponding frame collided with another frame, etc.). In this situation, the AP would still think that the station is associated, though the station has already left. This could cause data packet loss and could be minimized using an additional implicit disassociation mechanism at APs. Using implicit disassociation an AP keeps a timer running for associated stations, and in the case there is no data received from an associated station for a specific period of time, it sends a disassociation frame to the station and removed its entry from list of associated stations. It is possible that the station is still there but simply had no data to send data during that period of time. In this case, the station associates itself again with the AP.

4.1 Possible Implementation Approaches

MeDeHa can be implemented as a layer 2 solution. The advantage of this approach is that MeDeHa's protocol could be implemented on nodes that only have two layers (e.g., AP bridges). Also, in an internet involving infrastructure-based networks, it is easier to collect and use association/disassociation based information which is exchanged between APs. This implementation approach was chosen in our prior work [15]. The disadvantage is that message routing is more challenging.

An alternate approach is to implement MeDeHa at layer-3, which is what we have done currently. This facilitates the development of the routing function. On the other hand, in infrastructure-based networks, association and disassociation information is passed to layer 3 from layer 2. But, this solution requires that all nodes in the network including infrastructure-based nodes (e.g., APs) must run layer 3. An application-layer solution is also possible where application level routing could be performed between MeDeHa nodes; in infrastructure-based networks, the association (and disassociation) information could be passed from layer-2 to the application layer.

4.2 Implementation with NS-3 Simulator

To date, network heterogeneity is not supported in most open-source network simulators. We use the Network Simulator 3 (NS3) [16], which provides only basic network heterogeneity support required for our framework. We had then to extend NS-3's heterogeneity support. As previously described, we developed explicit– and implicit disassociation mechanisms in the simulator. In explicit disassociation, a station, before disconnecting from an AP, sends a disassociation frame to the AP, and then starts scanning all channels. This is done by comparing the received power with a threshold that is just above the minimum received power. Whereas, in case of implicit disassociation, the AP keeps a timer for nodes associations and removes stations from its association list by sending them a disassociation frame when the timer expires. These functionalities are done at the simulator's layer 2.

We have also used a cross-layer information exchange in order to pass association/disassociation information from layer 2 to layer 3.

Buffer management policies have also been implemented to provide per-flow and per-destination priority mechanisms. For instance, when a node's buffer is full, the oldest message with lower priority is dropped. Or, if a lower priority message arrives and the node's buffer is full with higher priority messages, the incoming message is discarded (dropped).

5 Performance Evaluation

We showcase MeDeHa's functionality and evaluate its performance through extensive simulations using a wide range of scenarios including traffic of different priorities. We used both synthetic, but realistic mobility patterns as well as real mobility traces.

5.1 Performance Metrics

To measure MeDeHa's effect on message delivery in heterogeneous internets subject to connectivity disruptions, we measure packet delivery ratio (PDR). Average delivery delay (AD) is also used as performance metric to show the benefits of embracing network heterogeneity. To this end, we compare scenarios where more than one network is supported against an infrastructure-only network [15].

It is important to note that for a message delivery protocol like MeDaHa that involves wireless communication, performance of the protocol in terms of message delivery depends upon how quickly neighborhood changes are detected. *HELLO* messages are used for this purpose in ad hoc networks, while beacons are utilized in infrastructure-based networks. Message delivery can be improved by sending neighborhood detection messages such as *HELLO* more frequently, but on the other hand, it increases protocol overhead. So, this tradeoff needs to be considered when setting the protocol's parameters.

5.2 Case 1: Convention Center Type Scenario

We consider a convention center type environment with different rooms and seminar halls where connectivity is provided by APs, but connectivity is not guaranteed everywhere (e.g., outside rooms or in hallways) in the convention center. Visitors carrying portable devices may move from one room to another and roam around across multiple AP coverage areas. These APs are connected to each other via Ethernet or point-to-point links. Without MeDeHa, visitors (nodes) get disconnected temporarily while moving from one room to another and hence may loose some messages (transfer of files and/or chat messages) destined to them. With MeDeHa, the network stores messages temporarily. When no destination information is available, APs store messages temporarily. When using more than one network, a message can either be delivered to a destination in infrastructure mode, in ad hoc mode, or the message can be handed over to a relay, which may carry the message to its destination.

This case is similar to the one we used in [15] in which we employed Random Waypoint (RWP) mobility with attraction points [20], [21]. Attraction points correspond to rooms and nodes move only in between these attraction points. Each attraction point is defined with a specific standard deviation along with an intensity to select the attraction point by the RWP mobility model. The standard deviation is of Gaussian distribution with zero mean and is used to specify the distances of nodes to the attraction point [22]. In other words, the standard deviation acts as a radius for the region of influence for an attraction point. Nodes are made to move in between these attraction point regions at a speed that is uniformly distributed between 1 and 2.5 m/s. Also, while within the coverage area of an attraction point, a node stays there for a time that is uniformly distributed between 0 and 60 seconds. A network of 9 APs is used spanning a 1000m x 1000m area; there are 16 attraction points, each having an effective radius of 20 meters, indicating its region of influence. There are 50 nodes in the network and we have run the simulations for a duration of 30 minutes. In order to perform simulations, we create some mobility traces using random waypoint mobility model with attraction points using the BonnMotion Mobility Model tool [22].

5.2.1 Uniform and Non-uniform AP Distribution

In the first set of experiments, 20 mobile stations exchange data, forming source-destination pairs. In other words, there are 20 sources and 20 destinations. Constant bit rate (CBR) traffic is generated using messages of 1 KBytes and different average data rates (in messages/minute). There is no buffer limit at APs as the goal is to study the impact of data rates and the AP distribution.

First, we place the APs uniformly across the entire network. This means that the distance between all the APs is constant. This is done so as to have low and uniform disconnection times when nodes move, representing an almost-connected network, comprised of connectivity "black holes". The deployment of APs and that of attraction points is shown in Figure 5.

Here, we compare two cases of MeDeHa: one where all stations support infrastructure based networks only (IS only), and the second case is where stations are able to connect to infrastructure based network as well as with other nodes in ad hoc mode (IS+Adhoc). Our goal is to evaluate the impact on delivery ratio (PDR) and delivery delay (AD). In ad hoc mode, we use number of encounters with a station as relay selection strategy, and set its value to 2. In other words, a message is forwarded to a relay, if it has seen the destination at least twice. Delivery ratio is shown in Figure 6, while the average delivery delay is presented in Figure 7.

All stations exhibit more than 90% delivery ratio irrespective of whether they are member of one or two networks for the case of both 8 messages/minute and

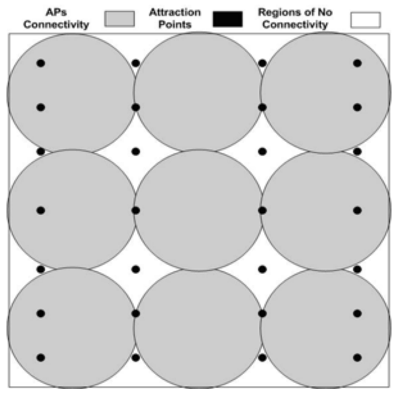

Figure 5: Uniform deployment of APs and attraction points in order to have equivalent areas of no connectivity with respect to APs

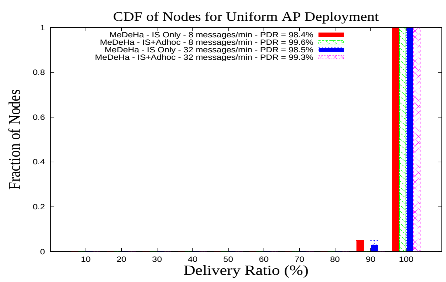

Figure 6: Fraction of Nodes vs. Delivery Ratio for uniform deployment of APs

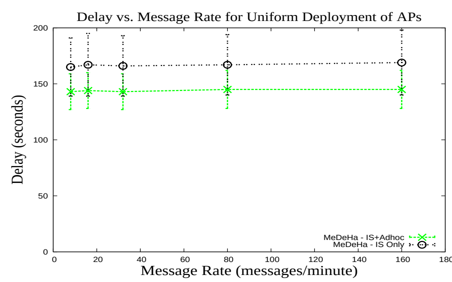

Figure 7: Delay vs. message rates for uniform deployment of APs

32 messages/minute [2]. While delivery ratio is not significantly affected, taking advantage of multiple networks decreases average delivery delay significantly irrespective of the data rate.

Next, we consider the case when the APs are distributed in the network in such a way that the distance between APs is non-uniform. The idea is to simulate an environment where the average disconnection time for stations is higher. Figure 8 shows the non-uniform deployment of APs for our simulations. All other simulation parameters are the same as the previous case. Delivery ratio and delay for the non-uniform AP deployment is shown in Figure 9 and Figure 10, respectively.

Here, we can see that 80% of stations have more than 90% delivery ratio in case of stations using only Infrastructure-based network (IS Only), as compared to more than 90% of stations having more than 90% delivery ratio when stations support both IS and ad hoc networks. Again, we can see that the average delay is higher as compared to the uniform AP deployment scenario, but we still observe an improvment in average delivery delay by using more than one network. The average delay is higher because the overall disconnection time is higher due to

[2] We used other values for average data rate and observed similar performance trend

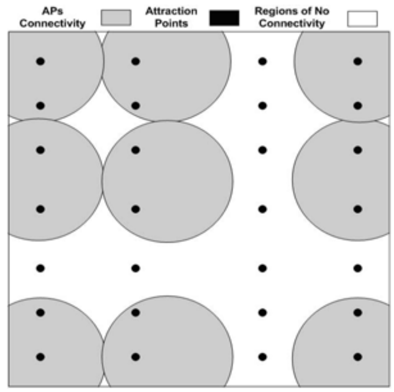

Figure 8: Non-Uniform deployment of APs and attraction points in order to have non-uniform disconnection times and areas of no connectivity with respect to APs

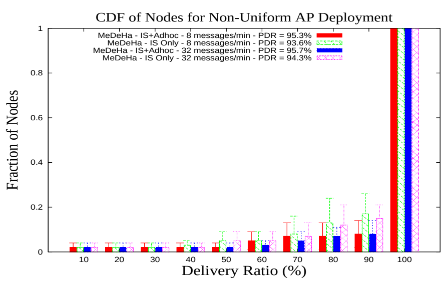

Figure 9: Fraction of Nodes vs. Delivery Ratio for non-uniform deployment of APs

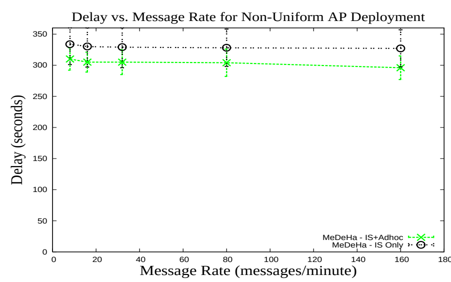

Figure 10: Delay vs. message rates for non-uniform deployment of APs

non-uniform AP positions. The same is the reason for slightly lower PDR as compared to uniform deployment case.

5.2.2 Buffer Sizes

The goal of these experiments is to evaluate MeDeHa's performance when buffer capacity at nodes is limited. Further, we inject traffic of different priorities. We use the uniform AP deployment leaving all other parameters the same. The results are given for 160 messages/minute and for stations supporting both infrastructure and ad hoc networks. Delivery ratio for different buffer sizes and 2 traffic priorities (high and low) is shown in Figure 11.

Our results confirm that MeDeHa gives preference to high priority messages, i.e., they achieve higher delivery ratio as compared to lower priority messages; this is especially true for the cases where buffer capacity is more limited.

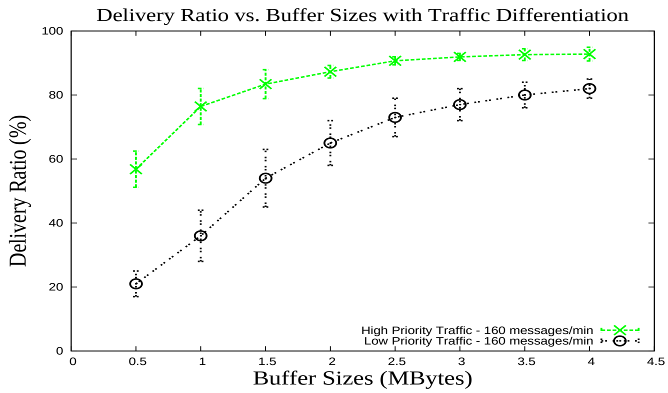

Figure 11: Impact of varying buffer sizes on Delivery Ratio for high and low priority messages (message rate: 160 messages/min)

5.3 Case 2: Communication between Clusters of Nodes

In this scenario, we simulate three clusters each which composed of a set of buildings; each cluster is equipped with 3 APs connected to each other providing connectivity to the whole cluster, though, there are possibly some regions of no connectivity within each cluster. The clusters are placed well apart and thus disconnected from each other (e.g., they can be seen as different villages in remote, disconnected areas). Each cluster is configured with 20 users carrying mobile devices: 14 of which only move within the boundary of their complex at pedestrian speed, while 6 visit other clusters with 0.2 probability. These nodes are potential relays to carry and forward inter-cluster traffic and are assumed to take buses when moving between clusters at speeds uniformly distributed between 30 and 50 km/h. Pedestrians that move inside a cluster do so at speeds uniformly distributed between 3 and 6 km/h. Total simulation area is 1000m x 1000m, and total simulation time is 60 minutes. The performance metrics used are percentage of nodes that receive a certain amount of delivery ratio, average packet delivery ratio (PDR), and average delivery delay (AD). Figure 12 shows the map of the scenario and the corresponding AP locations.

5.3.1 Forwarding versus Replication

For this scenario, we have chosen "community affiliation" as the relay selection strategy, where a community corresponds to a cluster. In other words, a source (or a relay) forwards a message to another node, if the latter belongs to the same community as that of the destination. Here, we compare the behavior of forwarding (where there is only one copy of a message) with 2-copy replication.

Additionally, traffic is divided into two parts: intra-cluster and inter-cluster traffic. Intra-cluster traffic is such that both the source and destination belong to the same cluster and thus does not leave the cluster for the duration of simulation. For each cluster, 10 source-destination pairs are chosen to generate intra-cluster traffic. Inter-cluster traffic is traffic exchanged by nodes belonging to different clusters. For this traffic, 10 source-destination pairs are selected from all 3 clusters. The message rate is 3 messages/minute. Figure 13 shows

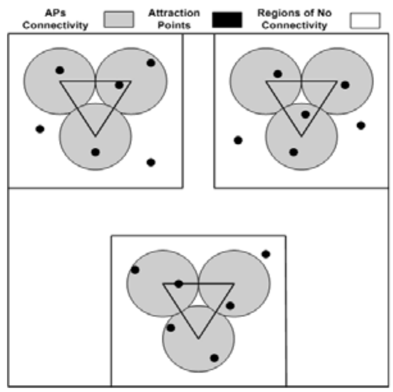

Figure 12: Deployment of APs and attraction points in a scenario with 3 disconnected clusters.

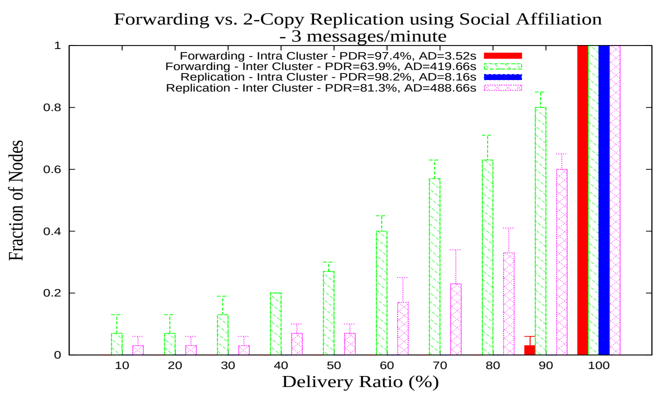

Figure 13: CDF of fraction of nodes vs. delivery ratio showing the comparison between forwarding and 2-copy replication for inter-cluster and intra-cluster traffic. Message rate is set to 3 messages/min

the CDF of the fraction of nodes as a function of delivery ratio using forwarding and replication for both kinds of traffic.

By comparing the results of forwarding and replication, we can see that in the case of forwarding, 63% of the nodes have less than 80% delivery ratio, whereas using 2-copy replication, only 33% of nodes have less than 80% delivery ratio, which is a significant improvement. The overall average delivery ratio (PDR) of all the nodes is also greatly improved using replication as compared to forwarding in the case of inter-cluster traffic (from 64% to 82%), and a slight improvement is observed in the PDR in the case of intra-cluster traffic. This improvement is because the traffic is local and any local node can become a relay node for a message, so the probability of message delivery is high. The minor increase in average delivery delay (AD) is due to the increase in PDR from 97.4% to 98.2%. For inter-cluster traffic, average PDR is greatly improved by using 2-copy *replication*, but increases the average delay as well (from 419.66 seconds to 488.66 seconds). This increase in average delay (AD) is due to the significant improvement in PDR, as the messages that get delivered very late contribute towards increase in AD. These messages donot contribute in *forwarding* case as they are never delivered.

The results discussed above are for a duration of 60 minutes. We observed that increasing the simulation time to 90 minutes resulted in increase in average delivery ratio (PDR) of nodes for inter-cluster traffic from 81.3% to 98.4% in case of replication, but also increased averge delay (AD) from 488.66 seconds to 712.48 seconds. This is because more messages are delivered to the destinations by increasing the simulation time to 90 minutes; these messages were undelivered but stored at nodes for the 60-minute case. The increase in PDR also caused delay (AD) to increase. On the other hand, for forwarding, increasing simulation time resulted improvement in PDR from 63.9% to 90.3%, as well as an increase in AD from 419.66 seconds to 822.37 seconds.

5.3.2 Relay Selection Strategy

Selecting a "suitable" relay to carry messages is an important component of MeDeHa and can have considerable effect on the performance of the protocol. One can employ different relay selection strategies depending upon a number of factors including network–, node–, and application characteristics as described in Section 2. In this subsection, we focus our attention on evaluating different criteria as the *utility* of a node to become a relay. First, we show the impact of using "number of past encounters" , which we name as Encounter-Based Replication (ER). Following ER, a source (or a relay) hands over a message to a node only if the latter has already encountered a destination at least twice. The idea behind this utility metric is that if a node has already seen a destination at least twice, there is a strong probability that it will again encounter the destination in the future. Note that, depending on the mobility pattern of nodes, this utility function may not be a good indication of the likelihood of future encounters.

Next, we choose "community affiliation" as the utility function for relay selection. In this way, a relay is chosen only if it belongs to the community of a destination. This utility function is meaningful here since in order to send traffic between different clusters, we have to rely on nodes that visit different clusters. 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 cluster. We call this relay selection strategy as Social Affiliation-based Replication (SAR) scheme. In Figure 14, a comparison is shown between ER and SAR selection strategies for 2-copy replication.

From the figure, it is clear that encounter-based replication performs well in terms of delivery ratio (PDR) while community affiliation-based replication provides lower average delay. The reason for this is that encounter-based replication only hands over a message to a relay if the relay has seen the destination at least twice. This adds buffering delay for waiting for a suitable relay, thereby increasing overall average delay, but may increase chances of message delivery as if a node has already encountered a destination twice, it is more probable that it is going to encounter the destination again in the future. The results obtained also favor this principle as is clear from increase in average PDR (from 81.3% to 85.2%). On the other hand, the decrease in average PDR in case of SAR is due

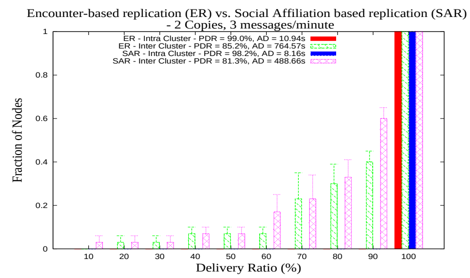

Figure 14: CDF of nodes vs. Delivery Ratio for 2-copy Encounter Replication (ER) and Social Affiliation Replication (SAR) - (3 msgs/min)

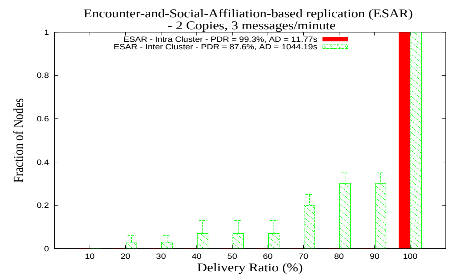

Figure 15: CDF of nodes vs. Delivery Ratio using 2-copy Community-and-Encounter Replication (ESAR) - (3 msgs/min)

to the fact that "community-based affiliation" metric chooses a relay node only based upon a node's affiliation with a particular community (cluster/complex here). It is not certain that every relay node will encounter a destination when it goes back to its parent cluster.

The delay for intra-cluster traffic is very low as compared to the delay for inter-cluster traffic. This is because intra-cluster traffic does not involve nodes belonging to different clusters, and therefore, a destination is found quickly within the cluster. Moreover, for inter-cluster traffic, 40% of nodes have less than 90% delivery ratio in case of ER, whereas 60% of nodes have less than 90% of delivery ratio in case of SAR.

Now, we combine the above two utility functions into the Encounter and Social Affiliation-based Replication (ESAR). So, a relay is chosen to carry a message to a destination only if it belongs to the same community as that of the destination as well as if it has encountered the destination at least twice. The result obtained for 2-copy replication is shown in Figure 15.

The choice of this hybrid utility function improves the average PDR for both types of traffic. The average delay (AD) for intra-cluster traffic is increased by using the hybrid function. This is because of the strict condition to choose a relay where a node has to keep on waiting for a suitable relay, and keeps a message stored until it encounters a node that follows the hybrid utility function, thereby adding an additional delay. On the other hand, the advantage of doing that is the improvement in average PDR. Thus, there is a tradeoff between increasing average PDR and decreasing average AD. In terms of fraction of nodes attaining a particular level of delivery ratio, using hybrid utility metric (ESAR), only 30% of nodes attain less than 90% delivery ratio (PDR) as compared to 40% of nodes using ESAR and 60% of nodes using ER (Figure 14).

5.4 Case 3: KAIST Campus Traces

We used a subset of real traces for the KAIST Campus available from CRAWDAD [23]. These traces record mobility of 50 students via their locations during a day. We took a 2-hour window over the trace from 10 AM to 12 PM. We superimpose this mobility pattern on top of an area of 1.4 km x 2.4 km with 9 APs. All APs are connected to each other as in Case 1. Students visit different places of campus during the time and their speed change (students take shut-

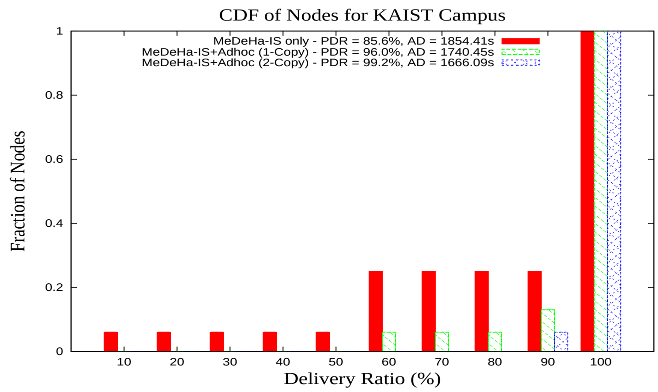

Figure 16: CDF of nodes vs. Delivery Ratio for KAIST Campus Traces for two hours using IS only and IS+Adhoc modes (message rate: 3 messages/min)

tles while moving from one place to another, and move at pedestrian speed or not at all). Again, we evaluated this scenario for 20 source-destination pairs of students with CDF of nodes (students) attaining a particular delivery ratio (PDR) for the case of having only infrastructure-based network (IS only), and having both infrastructure-based and ad hoc network (IS+Adhoc) for forwarding and replication. We also observed average packet delivery ratio and average delay (AD). The result is shown in Figure 16. Here, we used encounter-based replication (ER) for relay selection, and set its value to 2.

From the figure, it is clear that using network heterogeneity (IS+Adhoc) improves the performance both in terms of delivery ratio (PDR) and delivery delay (AD). IS+Adhoc *replication* attains the best average PDR and AD. In terms of fraction of nodes, we can see that only 6% of nodes have less than 90% delivery ratio for 2-copy heterogeneous network (IS+Adhoc) as compared to 25% of nodes having less than 90% of delivery ratio when using only infrastructure-based network (IS only).

6 Related Work

Most efforts that target heterogeneity in 802.11 networks aim towards extending network coverage and thus increasing network capacity. To extend network connectivity beyond regions covered by APs, these proposals employ different mechanisms such as: the use of different frequencies in Flex-Wifi [7], and a new layer between IP and link layer in MultiNet [8]. AODV+ [9] proposes a scheme to connect the Internet backbone to MANETs by introducing a gateway discovering mechanism. The common problem in all these schemes is the failure to deliver data in the presence of frequent network partitioning.

The seminal work of the IRTF's Delay-Tolerant Networking Research Group (DTNRG) pioneered research on DTNs with their delay-tolerant network architecture [11] a.k.a. Bundle Architecture. Their proposal is based on bundle switching with the ability to store bundles in transit for arbitrarily long periods of time. This is referred to as store-carry-and-forward. Storage is performed above the transport layer to provide interoperability among networks that support different types of transport layers. Our mechanism is orthogonal to the Bundle architecture that can be used with MeDeHa to support networks with different transport layers. In such cases, it is useless to store data at lower layers of nodes that act as DTN routers or gateways. But the need to store messages at lower layers in other nodes of network would still be the same, and MeDeHa would be useful especially when intermediate nodes don't support higher layers, and where the Bundle layer mechanism cannot be incorporated.

Propositions exist to integrate DTNs with MANETs. Ott et al. [2] introduce specialized DTN capable end point nodes to bridge islands of networks, but this solution doesn't provide backbone connectivity. Natasa et al. [1] use the mobility patterns of the nodes over time to make nodes communicate in between different islands, but again, with the help of nodes that move in between these islands. Besides, some studies use the concept of node relaying in order to bridge otherwise partitioned networks. These propositions include message ferries [12], throwboxes [14], and use of data mules [13]. They suggest the use of specialized nodes, fixed or mobile that are used as data carriers, and/or forwarders. Specialized nodes are resourceful entities (storage space, battery power etc). The concept is very fruitful in increasing the delivery ratio, and in some cases, reducing the overall delay, but the problem of number of these special-purpose nodes in the network, and their routes is not trivial.

Some initiatives target relay node selection in a disruption tolerant environment. One notable example is [17], which presents different utility functions to be utilized for intermittent connected networks with different characteristics. Exponential Age Search (EASE) algorithm is presented in [18], where a destination location is estimated by using the encounter database maintained locally by each node for every other node. A similar approach is presented in [19] where future rate of node encounters is predicted using number of past encounters with nodes. For this purpose, an encounter metric is computed locally by each node, and is used as utility metric when choosing a relay for a message.

7 Conclusion

This paper introduced MeDeHa, a robust and flexible message delivery framework targeting heterogeneous internets subject to intermittent connectivity. We believe that this work is an important building block to enable current and upcoming applications since; (1) future internets will likely become increasingly more heterogeneous and (2) in many scenarios/applications, late delivery is preferred over loss of data.

MeDeHa's contributions are two fold. First, we address the problem of frequent and/or long-lived connectivity disruptions in heterogeneous networks. Heterogeneity is handled at the network– as well as at the node level. Second, with our scheme, there is no need to introduce special-purpose nodes in order to support heterogeneity. This makes MeDeHa quite general and flexible. We also showcase MeDeHa's performance in simulations with some synthetic but realistic scenarios as well as with real traces and show that it helps in improving message delivery ratio, while reducing delivery delay.

We are currently extending MeDeHa with MANET routing support. In the longer run, we will address group communication and QoS-based data delivery.

References

[1] N. Sarafijanovic-Djukic, M. Piorkowski, and M. Grossglauser, *Island Hopping: Efficient Mobility-Assisted Forwarding in Partitioned Networks*, In Proceedings of IEEE SECON'06, pp. 226-235, 2006.

[2] Jorg Ott, Dirk Kutscher, and Christoph Dwertmann, *Integrating DTN and MANET Routing*, In Proceedings of ACM SIGCOMM workshop on Challenged Networks (CHANTS), pp. 221-228, Pisa, Italy, 2006.

[3] A. Vahdat and D. Becker, *Epidemic Routing for Partially Connected Ad Hoc Networks*, Technical Report CS-200006, Duke University, 2000.

[4] T. Spyropoulos, K. Psounis, and C.S. Ragha Vendra, *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*, ACM SIGCOMM Workshops WDTN, pp. 252-259, Philadelphia PA, 2005.

[5] J.-C. Chen, S. Li, S.-H. Chan, and J.-Y. He, *WIANI: Wireless Infrastructure and Ad-Hoc Network Integration*, In Proceedings of IEEE International Conference on Communications (ICC), pp. 3623-3627, Seoul, Korea, 2005.

[6] J He J. Chen, S.-H. G. Chan, and S.-C. Liew, *Mixed-mode Wlan: The Integration of Ad Hoc Mode with Wireless LAN Infrastructure*, In Proceedings of IEEE Globecom, pp. 231-235, 2003.

[7] Carlo Parata, Gabriella Convertino, Vincenzo 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), pp. 1-6, Finland, 2007.

[8] R. Chandra, P. Bahl, and P. Bahl, *MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a SingleWireless Card*, In Proceedings of IEEE Infocom, pp. 882-893, Hong Kong, 2004.

[9] 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, pp. 189-201, Springer Berlin, 2005.

[10] K. Scott and S. Burleigh, *RFC 5050, Bundle Protocol Specifications*, IRTF DTN Research Group, November 2007.

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

[12] W. Zhao, M. Ammar, and E. Zegura, *A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks*, In Proceedings of ACM/IEEE MOBIHOC, pp. 187-198, Japan, 2004.

[13] R. Shah, S. Roy, S. Jain, and W. Brunette, *Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks*, In Proceedings of IEEE Workshop on Sensor Network Protocols and Applications (SNPA), pp. 30-41, Seattle, WA, 2003.

[14] Wenrui Zhao, Yang Chen, Mostafa Ammar, Mark Corner, B.N. Levine, and Ellen Zegura, *Capacity Enhancement using Throwboxes in DTNs*, In Proceedings of IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), pp. 31-40, Canada, 2006.

[15] Rao Naveed Bin Rais, Thierry Turletti, and Katia Obraczka, *Coping with Episodic Connectivity in Heterogeneous Networks*, In Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pp. 211-219, Canada, 2008.

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

[17] T. Spyropoulos, T. Turletti, and K. Obraczka, *Routing in Delay Tolerant Networks Comprising Heterogeneous Node Populations*, IEEE Transactions on Mobile Computing (TMC) Vol. 8, No. 8, pp. 1132-1147, August 2009.

[18] Matthias Grossglauser and Martin Vetterli, *Locating Mobile Nodes with EASE: Learning Efficient Routes from Encounter Histories Alone*, IEEE/ACM Transactions on Networking, Vol. 14, No. 3, pp. 457-469, June 2006.

[19] Samuel Nelson, Mehedi Bakht, and Robin Kravets, *Encounter-Based Routing in DTNs*, In Proceedings of IEEE Infocom'09, Brazil, 2009.

[20] Christian Bettstetter and Christian Wagner, *The Spatial Node Distribution of the Random Waypoint Mobility Model*, In Proceedings of the First German Workshop on Mobile Ad-Hoc Networks (WMAN), GI Lecture Notes in Informatics, Vol. 11, pp. 41-58, 2002.

[21] 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, Springer Berlin, 2004.

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

[23] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seongjoon Kim and Song Chong, *CRAWDAD data set ncsu/mobilitymodels (v. 2009-07-23)*, Downloaded from http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels, July 2009.

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399