Mechanisms to Enhance Versatility, Robustness and Reliability in Delay Tolerant Networks

by

Lee Feng Cheng

A thesis submitted to
School of Computer Engineering
in partial fulfillment of the requirements for the degree of
Master of Engineering

Supervisor: Assoc. Prof. Yeo Chai Kiat

August 2010

ABSTRACT

The Delay Tolerant Network (DTN) is a network architecture which has the capability of overcoming difficulties posed by the disruptive nature of challenged networks. It is suitable for deployments that have infrastructure constraints whereby connectivity among peers is intermittent and interoperability in heterogeneous network access technologies is a requirement. In the existing DTN implementation, there are some issues with the discovery mechanism, convergence layers, security and routing protocols.

This thesis contributes a few key enhancements for DTN. To achieve highly versatile interoperability support in DTN, a new 'plug-n-play' framework is proposed. It supports DTN node cascading and allows parallel network features that will expand DTN peers' access capability. Adding to the framework, a new Ethernet convergence layer using raw socket programming is developed. This is more lightweight than the current TCP convergence layer to support WiFi and other access technology.

In Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) for DTN, the delivery predictability metric involved can be exploited by adversary nodes to improve their flooding attacks. The flooding attacks can be more penetrative as malicious nodes have prior knowledge of the delivery predictability of their victims. To overcome this threat, a queue buffer policy is formulated to utilize the delivery predictability metric in PRoPHET to mitigate the flooding attacks. The proposed policy is shown to be capable of alleviating five different types of flooding attacks on DTN using PRoPHET.

In another enhancement for PRoPHET, a history of messages concept is introduced to provide PRoPHET with more complete knowledge in prioritizing messages. It involves the use of a benefit system to rate the importance of the messages in its forwarding and management of queue buffer. The new enhancement improves the message delivery performance of PRoPHET and is comparable to MaxProp.

ACKNOWLEDGEMENTS

I would like to use this opportunity to thank the following people, DSO and NTU for their help in one or more ways in my fulfillment of my Master of Engineering degree.

- I would like to thank my supervisor, Associate Professor Yeo Chai Kiat, for her guidance, precious advice, encouragement and keeping me on track in achieving a Master's degree.

- I would like to thank DSO for giving me the opportunity to do research on the projects and work on the testbed implementation.

- I would like to thank Dang Duc Nguyen, Xia Yang and Zoebir for their technical expertise and sharing of knowledge.

- I would like to thank the laboratory technicians, Chua Poo Hua and Teo Cheng Kee Cindy, at CeMNet for the administrative work in my projects.

- Last but not least, I would like to thank my family for giving me time to do post graduate study as well as to thank my relatives and friends for their support.

Lee Feng Cheng

Nanyang Technological University

Academic Year 2009-2010

TABLE OF CONTENTS

	Page No.
Abstract	**I**
Acknowledgements	**II**
List of Figures	**VI**
List of Tables	**VIII**
List of Acronyms	**IX**
Chapter 1 - Introduction	**1**
1.1 - Challenged Networks	1
1.1.1 - The Delay Tolerant Network Approach	1
1.1.2 - DTN Applications in Challenged Networks	2
1.2 - DTN Architecture	3
1.2.1 - DTN Application Layer	4
1.2.2 - DTN Routing Layer	4
1.2.3 - DTN Bundle Layer	4
1.2.4 - DTN Convergence Layer	5
1.3 - Motivation of this Thesis	5
1.4 - Contribution of this Thesis	7
1.5 - Organization of this Thesis	8
Chapter 2 - Literature Survey	**9**
2.1 - Enhancing Network Access Heterogeneity	9
2.2 - Routing Protocols in DTN	10
2.3 - Defense against Flooding Attacks	12
Chapter 3 - A 'Plug-and-Play' Framework to Enhance Heterogeneity and Versatility in Delay Tolerant Networks	**14**
3.1 - Introduction	14

3.2 - Architecture 17
3.2.1 - Cascading Concept 17
3.2.2 - Modifications to Existing DTN Discovery Mechanism 19
3.3 - Comparison with Existing DTN Implementation and Related Work 20
3.4 - Heterogeneous DTN Testbed Implementation 22
3.5 - Heterogeneous DTN Testbed Evaluation 24
3.6 - An Alternative to TCP CL – New Ethernet CL 27
3.6.1 - Facilitating Store-and-Forward mechanism 29
3.6.2 - Recovery from duplicate and missing frames 30
3.6.3 - Fragmentation 31
3.7 - Ethernet CL Testbed and Results 32
3.7.1 - File transfer 32
3.7.2 - Video streaming 33
3.8 - Conclusion 34

Chapter 4 - A Queuing Mechanism to Alleviate Flooding Attacks in Probabilistic Delay Tolerant Networks 36
4.1 - Introduction 36
4.2 - Possible Flooding Attacks against PRoPHET Protocol 38
4.2.1 - Random Flooding 38
4.2.2 - Selective Destination Nodes Flooding 38
4.2.3 - Non-Existent Destination Node Flooding 39
4.2.4 - Spoof Flooding 39
4.3 - Factors for Consideration 40
4.3.1 - Message Ferry 40
4.3.2 - Malicious Nodes Increasing Delivery Predictabilities 40
4.3.3 - Messages' Source or Previous Node Factor 40
4.3.4 - Storage Space Issue 42
4.4 - Proposed Solution 42
4.4.1 - Checking Previous Node and Event of Message Ferry 43

4.4.2 - Capitalizing on Delivery Predictability 43
4.4.3 - New Queuing Policy Proposed 43
4.4.4 - Distort Delivery Predictabilities Flooding 46
4.5 - Simulation 46
4.6 - Evaluation of Results 48
4.7 - Conclusion 52

Chapter 5 - Probabilistic Routing based on History of Messages in Delay Tolerant Networks **54**
5.1 - Introduction 54
5.2 - Preliminary Study 56
5.2.1 - Spray-And-Wait Routing 56
5.2.2 - MaxProp Routing 57
5.2.3 - PRoPHET Routing 58
5.2.4 - Analysis on the Routing Protocols 59
5.3 - Proposed Solution 61
5.3.1 - History of Messages Concept 61
5.3.2 - Our Forwarding Strategy Specification 64
5.4 - Simulation 66
5.4.1 - Homogeneous Scenario 67
5.4.2 - Heterogeneous Scenario 67
5.5 - Evaluation of Results 68
5.6 - Conclusion 75

Chapter 6 - Conclusion and Future Works **76**
6.1 - Security Protocol to Counter Masquerade Attack 77
6.2 - Peer Discovery for Multi-hop 78
6.3 - Routing Protocol for Heterogeneous DTN 78

Author's Publications **79**
Bibliography **80**

LIST OF FIGURES

	Page No.
Figure 1.1 - DTN framework	3
Figure 3.1 - Cascaded nodes in operational context	18
Figure 3.2 - DTN framework	19
Figure 3.3 - Different network interfaces multicast and listen to announcement beacons on different subnets	20
Figure 3.4 - Testbed demonstrating cascaded nodes in heterogeneous DTN	23
Figure 3.5 - Scenario 1	25
Figure 3.6 - Scenario 2	26
Figure 3.7 - Scenario 3	26
Figure 3.8 - Map showing the location of the testbed	26
Figure 3.9 - Difference between TCP and Ethernet convergence layers	28
Figure 3.10 - Ethernet Frame Specification (IEEE 802.3)	31
Figure 3.11 - File transfer using DTN and SCP (FTP) in Wireless Adhoc Network	33
Figure 3.12 - Video streaming using DTN and UDP in Wireless Adhoc Network	34
Figure 4.1 - Random Flooding	50
Figure 4.2 - Selective Destination Nodes Flooding	50
Figure 4.3 - Non-Existent Destination Node Flooding	51
Figure 4.4 - Spoof Flooding	51
Figure 4.5 - Distort Delivery Predictabilities Flooding	52
Figure 5.1 - MaxProp routing strategy	57
Figure 5.2 - Preliminary finding	59
Figure 5.3 - Routing scenario	59
Figure 5.4 - Priority for forwarding of messages	65

Figure 5.5 - Priority for dropping of messages	66
Figure 5.6 - Various existing queuing policies and forwarding strategies for PRoPHET	66
Figure 5.7 - Messages delivery in homogeneous scenario	71
Figure 5.8 - Messages delivery in heterogeneous scenario	72
Figure 5.9 - Latency in heterogeneous scenario	73
Figure 5.10 - Overhead in heterogeneous scenario	74

LIST OF TABLES

	Page No.
Table 5.1 - Notations	62
Table 5.2 - A scenario analysis	64

LIST OF ACRONYMS

CL	Convergence Layer
DTLSR	Delay Tolerant Link State Routing
DTN	Delay Tolerant Networks
FTP	File Transfer Protocol
IP	Internet Protocol
LSA	Link State Announcement
MIH	Media Independent Handover
ONE	Opportunistic Network Environment
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
RAPID	Resource Allocation Protocol for Intentional DTN
RTT	Round Trip Time
SCP	Secure Copy Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UHF	Ultra High Frequency
URI	Universal Resource Identifier

Chapter 1.

Introduction

1.1. CHALLENGED NETWORKS

Besides the hugely widespread Internet, some existing specialized networks have been operating and becoming more important in recent years. These specialized networks have their own unique communication requirements to handle various possibilities such as harsh environments, deliberate jamming by malicious nodes or even contending for limited bandwidth among the individual services. Facing the unconventional difficult encounters, these specialized networks are commonly known as "challenged networks". Challenged networks have characteristics that hinder communication such as intermittent connectivity, long and variable propagation delays, asymmetrical data rates and high error rates.

For deployment in challenged environments, current IP based Internet model is not very well-suited to be adopted for this purpose due to various assumptions in its protocols. The TCP/IP model assumes: there is an end-to-end path between the source and destination, greatest round-trip time between nodes is relatively short and packet drop rate is low. However, the regular network partitions and long delay paths in challenged networks would contradict the assumptions in TCP/IP. In fact, they may result in many unanticipated requests for retransmissions, causing the protocols used in TCP/IP to be ineffective in challenged networks.

1.1.1. The Delay Tolerant Network Approach

In [1], Fall puts forth the Delay Tolerant Network (DTN) architecture which can overcome the difficulties posed by challenged networks. In DTN architecture, the

source node does not need to maintain a connected end-to-end path to the destination node to accomplish message delivery. The store-and-forward mechanism is set to overcome the intermittent connectivity in the relaying nodes in the process of delivering message to the destination. Using persistent storage, DTN routers allow messages to be stored for longer period of time before being forwarded to the next hop when connectivity becomes available, thus reducing the need for frequent retransmissions. To overcome long end-to-end path, DTN uses a late binding scheme. In late binding scheme, each node performs the name-to-address mapping only for the next hop. The source node is only concerned about the address of the relaying node at the next hop, and not concerned about the address of the destination node. It is expected that the name-to-address translation of the destination node will be done at the very last hop, where the node relaying to the destination will hold the responsibility of performing the translation. The late binding scheme is suitable for challenged networks, since it would be unrealistic for the source node to find the address of the destination node considering the random disconnections along the path. Hence, DTN fits well to meet the extreme requirements of challenged networks.

1.1.2. DTN Applications in Challenged Networks

Examples of challenged networks that can employ DTN are village area networks [2][3], wildlife [4][5], satellite networks [6][7], military ad-hoc networks [8][9] and wireless sensor networks [10][11]. In village area networks such as KioskNet [12][13][14], to overcome the infrastructure limitations and accessibility constraint, a message ferry vehicle can be deployed to provide accessibility to the village. This network scenario can rely on DTN's store-and-forward service as the vehicle travels from place to place to deliver the messages. In satellite networks, latency is generally high due to data being transmitted over long distances. Given this situation, DTN's persistent storage can provide the buffer to sustain the long propagation delays. In military ad-hoc networks, connectivity is normally intermittent considering the mobility, tough environmental constraints and deliberate jamming by adversary nodes. DTN's store-and-forward capability can overcome the intermittent connectivity among nodes. In wireless sensor networks, nodes commonly have limited power and memory

resources. Data transmission among nodes is usually scheduled for the purpose of conserving power which results in frequent network partitions. DTN is a suitable choice as DTN does not require the source node to establish an end-to-end connection to the destination node to complete a message transfer.

1.2. DTN ARCHITECTURE

The DTN is an overlay architecture operating above the existing protocol stacks of the challenged networks it interconnects. The layers in DTN comprises DTN application layer, DTN routing layer, bundle layer and convergence layer. The DTN application layer is separated from the core of DTN and is left for the community to write their own applications for DTN. The routing layer makes decisions on which DTN nodes to forward the messages to. The bundle layer encapsulates the messages into a recognizable format for DTN and these 'bundled' messages are then known as DTN bundles. The bundle layer and bundles in DTN will be further explained in Section 1.2.3. Lastly, the convergence layer provides an interface between the bundle layer and the respective underlying layers of the networks it interconnects.

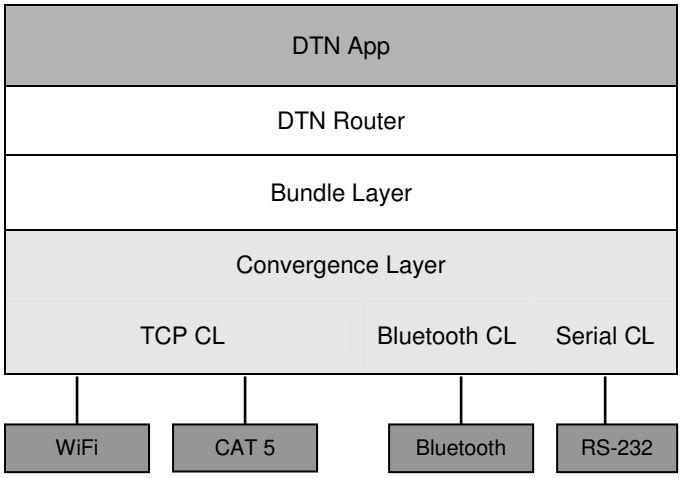


Figure 1.1 DTN framework

1.2.1. DTN Application Layer

At the application layer, DTN currently has basic applications that include DTN ping, file transferring, text messaging and tunneling. Although the number of applications is quite limited, the tunneling application offers DTN the flexibility of allowing current applications using IP to operate over DTN. Hence the existing applications provide DTN with a decent range of possibilities, with expansion capability when more applications are implemented in future.

1.2.2. DTN Routing Layer

In DTN routing layer, it offers choices from its specialized range of routing protocols that are designated to suit the different needs of various challenged networks. For satellite network, routing used in DTN is based on scheduled encounters of the planets and orbiting satellites. For military ad-hoc network, routing is more robust and is opportunistic based because prediction of encounters can only be probabilistic considering the mobility involved and unforeseen situations that can arise. For wireless sensor network, routing in DTN is energy-awareness oriented due to the limited power of the sensors. Alternatively, routing can also be scheduled to conserve power in sensor network scenario. In general, the routing protocols in DTN are unlike traditional Internet routing. In comparison to Internet, DTN cannot assume established end-to-end connectivity, nor low error rates or short round-trip times for shortest routing path decision. For robustness, forwarding decisions made by DTN routing protocols generally factor in the frequent disconnections and possibly longer than expected propagation delays.

1.2.3. DTN Bundle Layer

The bundle layer functions as the core of DTN, providing store-and-forward service that makes DTN more delay and disruption tolerant than IP based Internet model. Data transmission between nodes uses a store-and-forward technique whereby the nodes are expected to use persistent storage to store the message until the entire message has

been fully forwarded. This technique prevents loss of data during disconnection while the entire message is progressively being relayed at the corresponding hops. In DTN, the bundle layer bundles up the entire message as a bundle before transmission. Hence, from the bundle layer's perspective, a "bundle" in DTN actually holds the entire message.

1.2.4. DTN Convergence Layer

The convergence layer provides service to the bundle layer by encapsulating the bundles into a suitable format for transmission at the underlying delivery protocol used by the network access interface. In DTN, the different network access technologies are served by the different types of convergence layers as shown in Fig. 1.1. DTN allows users to choose from its range of convergence layers including TCP/IP, Bluetooth and Serial to interface with the node's physical network access technologies.

1.3. MOTIVATION OF THIS THESIS

The DTN Research Group [15] provides an implementation of DTN, and the existing implementation has a set of convergence layers to support the commonly used network access technologies such as WiFi, Serial RS-232 and Bluetooth. However, the 'Plug-and-Play' support of the network access technologies is not complete as some convergence layers are still lacking in discovery mechanism for opportunistic links. As far as we know in DTN version 2.6.0 [15], the implementation does not have a discovery mechanism for Serial convergence layer. In addition, complication arises with the existence of parallel network links when more than one network access interface use the TCP convergence layer. In such scenarios, the secondary (subsequent) network links are not being discovered. This will be problematic since many existing network access technologies are still TCP/IP based. Hence, it is hard to neglect the existing limitation in the discovery of opportunistic links in DTN.

There are two existing convergence layers in DTN, namely TCP and UDP, which can support current network technologies such as Wi-Fi that uses IP for communication. However, TCP/IP is an end-to-end connection oriented protocol and may not perfectly suit DTN as a convergence layer because DTN does not require end-to-end connectivity for the source and the destination nodes to communicate. For UDP, it can only provide an unreliable service as there is no handshaking mechanism to detect duplicate and missing packets. Hence, using either of these two convergence layers as the underlying convergence layer for Wi-Fi communication in DTN may not be the perfect solution.

Aside from the network access technologies support, there are many open research issues in DTN routing as challenged networks have different requirements that traditional Internet routing cannot satisfy. For the various harsh conditions that different types of challenged networks may pose, DTN requires a considerable set of routing protocols for selection to overcome the specific challenge of the situation that can arise. The unexpected disconnections, possibly long transmission latencies and network heterogeneity that DTN is meant to support make the routing problem in DTN very complicated. There is a group of probabilistic based routing protocols that gauge delivery likelihood of nodes using previous encounter events. The delivery likelihood metric in probabilistic routing protocols provides a potential to be exploited for future enhancements in DTN.

From the adversary perspective, probabilistic routing protocols lead to malicious nodes being able to perform more sophisticated flooding attacks due to the knowledge exchanged when the nodes encounter each other. The adversary, knowing the delivery likelihood of the nodes, can perform spoofing to conceal their attacks as well as selectively flood the highly active nodes. The adversary flooding attacks would add on to the already harsh conditions that the natural surroundings could pose in challenged networks. With the already limited forwarding opportunities in challenged networks, malicious nodes could further deprive the nodes from forwarding useful messages with their flooding attacks. As a consequence of the increase in sophistication of flooding

attacks in DTN, developing defensive mechanisms against various forms will require a different approach.

As such, there are still open research areas in DTN which have yet to be explored thus adding to the motivation of this thesis. As discussed, the protocols in DTN for WiFi communication are imperfect and the discovery mechanism needs more versatility to handle parallel discoveries. In addition, DTN needs a more lightweight delivery protocol than the current TCP/IP convergence layer. At the routing layer of DTN, the delivery likelihood metric of probabilistic routing can be exploited to enhance delivery reliability and also to tackle adversary attacks. Hence, the exploitation of the delivery likelihood metric is another area that this thesis is motivated to explore.

1.4. CONTRIBUTION OF THIS THESIS

In this thesis, some solutions for a few existing issues in DTN, namely, versatility, mitigation against attacks such as flooding, improving delivery ratio of routing protocol, have been proposed. The main contributions of the thesis are as follows:

- A 'plug-and-play' framework with improved discovery mechanism to enhance heterogeneity and versatility in DTN has been proposed and implemented. In addition, a new convergence layer written using raw Ethernet socket programming which is more lightweight than the existing TCP convergence layer for Wi-Fi communication in DTN has been proposed and implemented.

- An analysis of the various types of flooding attacks that malicious nodes can perform on a DTN probabilistic routing protocol, Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) is performed. A new queuing policy has been proposed as a solution for PRoPHET to mitigate the flooding attack.

- A detailed analysis of routing in DTN that reveals the forwarding decisions that could go wrong in DTN's store-and-forward nature is performed. A history of messages concept which PRoPHET can utilize to improve its chances of delivering its messages to their destinations is proposed and evaluated.

1.5. ORGANIZATION OF THIS THESIS

The remaining chapters of the thesis are organized as follows:

- Chapter 2 reviews the related work for the existing issues in DTN.

- Chapter 3 presents the 'plug-and-play' framework and the new Ethernet convergence layer.

- Chapter 4 details a proposed queuing policy for PRoPHET to mitigate the flooding attack.

- Chapter 5 presents the algorithm to use the message hop history in PRoPHET to improve the message delivery ratio.

- Chapter 6 concludes this thesis and provides some recommendations for future work.

Chapter 2.

Literature Survey

2.1. ENHANCING NETWORK ACCESS HETEROGENEITY

Network access heterogeneity supported in traditional IP protocol uses IEEE 802.21 Media Independent Handover (MIH) [16] component to perform handover between network access interfaces. MIH is located in the protocol stack just above the network access technologies at data link layer (layer 2) and below the IP at network layer (layer 3). Its functionality is to determine and initiate handovers of network access technologies in the data link layer. MIH is capable of switching between network access interfaces of the same technology known as horizontal handover, as well as switching between network access interfaces of different technologies known as vertical handover. However, handover using MIH is inflexible as it constrains every node to use a common network host identifier (IP address). Such model will require a converter to resolve its addressing issue, at times requiring additional physical hardware such as TCP/IP to Serial adapter. As a comparison, DTN uses Universal Resource Identifier (URI) naming convention and its underlying convergence layers catered for the different network access technologies without restricting DTN to use a common network host identifier. To enhance DTN's heterogeneity and versatility, it would require a more general technique which does not impose the use of a common network host identifier on the DTN.

In interoperable heterogeneous network research, a related architecture, ParaNets [17], involves providing parallel network links over two end-nodes to address intermittent connectivity and network heterogeneity challenges. DTN is used in parallel with an external cellular network in the evaluation of ParaNets. ParaNets architecture exploits the spare network resources in cellular network, and as a result it

improves the overall performance of DTN. From DTN's perspective, ParaNets is an external solution to enhance its heterogeneity, as ParaNets sits as a layer above DTN to do switching between DTN and another network such as cellular network. However, DTN, as a standalone network, can have its own internal solution to perform the switching between underlying network access technologies with its range of convergence layers providing the transmission service. Within DTN, there are still internal vertical and horizontal handovers between the various convergence layers that can be enhanced.

2.2. ROUTING PROTOCOLS IN DTN

There have been many routing protocols proposed in DTN, each having its own motivation and serves a unique purpose. In [18], Jain et al. formulated the DTN routing problem and introduced a framework to assess the routing protocols. Four knowledge oracles namely, contacts summary, contacts, queuing and traffic demand, were used in their evaluation framework. Contacts summary provides overall summary on the contacts in the network, such as the average waiting time for the appearance of a discovered contact to form a new edge in the network. Contacts knowledge oracle provides information on the contacts between a pair of nodes at any point in time. The contacts summary oracle can be derived from the contacts knowledge oracle. Queuing knowledge oracle provides information about the buffer queue of any node at any time. Traffic demand oracle provides information on the current or future traffic demand. With the queuing and traffic demand oracles, routing of data packets can be better informed to avoid congested areas of the network. Jain et al. further formulated a few routing algorithms. Through simulations, the algorithms are evaluated on their performance with respect to the amount of network knowledge used from the oracles. It was found that the use of network knowledge to solve routing problem has a positive effect on DTN's performance.

Every routing protocol's level of sophistication generally depends on the complexity of the routing problem. In this thesis, we seek ways to enhance the

PRoPHET routing protocol in DTN that is probabilistic based. As such, probabilistic routing protocols are of particular interest in our survey. In addition, we also survey some simplistic and slightly sophisticated routing protocols for the purpose of studying how these non-probabilistic routing protocols benefit DTN. As a result of our survey, the simplistic Spray-And-Wait and the probabilistic MaxProp routing protocols are critical in our discovery of a new effective policy for PRoPHET which will be detailed in Chapter 5.

Some of the routing protocols in DTN are more simplistic and do not involve any knowledge of the network. Yet these simplistic routing protocols can achieve their respective objectives. In [19], Vahdat et al. proposed Epidemic routing protocol that can maximize message delivery in DTN assuming no buffer storage and bandwidth limitation. Epidemic routing achieves maximum message delivery without incurring unnecessary overhead by forwarding a message replica if the node it encounters does not have a replica. In [20], Spyropoulos et al. proposed Spray-And-Wait routing protocol that can achieve low delivery latency without incurring high overhead. Spray-And-Wait routing predefines the number of replicas to be circulated in the network for each message. Hence the messages do not traverse too many hops and the overhead does not exceed a predefined number.

For the slightly more sophisticated routing protocols in DTN, there are Delay Tolerant Link State Routing (DTLSR) [21] and Resource Allocation Protocol for Intentional DTN (RAPID) [22]. Demmer et al. proposed DTLSR which is a modification of the conventional Link State routing. DTLSR sets longer lifetime for its link state announcements (LSAs) and uses a constrained flooding algorithm to distribute its LSAs. The modification suits DTN as challenged networks can have long transmission latency and network partitions. Balasubramanian et al. proposed RAPID which is a routing protocol that can intentionally optimize a particular routing metric such as worst-case delivery delay. Their argument is RAPID does not leave it to chance when it comes to routing in DTN.

There are probabilistic based routing protocols in DTN that measure the delivery likelihood using history of encounters. In [23], Burgess et al. proposed MaxProp routing protocol which factors in delivery likelihood and messages' hop count in its forwarding decision making. In [24], Lindgren et al. proposed PRoPHET routing protocol that computes delivery predictability based on the history of encounters. Both of these routing protocols are suited for network that has a certain mobility pattern. PRoPHET was deployed in Saami nomadic community in the remote areas in Swedish Lapland [25] and MaxProp was deployed in UMassDieselNet [26], both showcasing capabilities in solving real problem for DTN.

2.3. DEFENSE AGAINST FLOODING ATTACKS

In [27], Lindgren et al. discussed some security concerns faced by the PRoPHET [24] routing protocol in DTN. PRoPHET routing uses delivery predictability metric to estimate the delivery likelihood of a node for a particular destination node. Malicious nodes, upon knowing the delivery likelihood of the nodes in the network, can perform more organized flooding attacks on the network. Flooding attacks performed can be less detectable through spoofing of delivery predictabilities and more devastating through selectively targeting high delivery nodes in the network. Adding to the intermittent connectivity in challenged networks, these advanced flooding attacks can further limit the forwarding opportunities in DTN.

In [28], Perlman's network design with byzantine robustness can alleviate the effect of flooding attacks through fair allocation of resources. Perlman's concept suited the traditional routing but may not fit perfectly into the PRoPHET routing protocol in DTN. PRoPHET is probabilistic based and messages tend to converge to the nodes with higher delivery predictabilities as they traverse towards the destination. Therefore, fair allocation of resources may not be appropriate for a normal scenario when there is no malicious flooding attack.

In [29], Lindgren et al. proposed some buffer queue policies for PRoPHET and Epidemic routing protocols. Their evaluation shows that appropriate choice of buffer queuing policy can enhance network performance in probabilistic routing. However, the queuing policies proposed are not meant for alleviating flooding attacks. In [30], Krifa et al. proposed an optimal queuing policy using an estimation of global information. In their evaluation, however, their proposed policy is used with Epidemic routing protocol which does not have probabilistic property. As the flooding attacks against PRoPHET possess the predictive trait, their proposed policy is not suitable to be used with PRoPHET.

Chapter 3.

A 'Plug-and-Play' Framework to Enhance Heterogeneity and Versatility in Delay Tolerant Networks

3.1. INTRODUCTION

Under infrastructure limitations, Delay Tolerant Networks (DTN) can encounter circumstances whereby intermittent connectivity and network heterogeneity [1] are common. In these situations, interface versatility is crucial when maintaining network connectivity among the diverse types of computing devices with assorted network access interfaces. In times of intermittent connectivity, connection can be reestablished through the addition of secondary network access interface. The ability to support 'plug and play' secondary interface will also help to overcome network heterogeneity and allow two or more networks with multiple network access technologies to interoperate with one another.

In several challenged network scenarios, DTN end-nodes are constrained by limited resources [31] and inadequate network access interfaces. These end-nodes depend on a few DTN gateways [32] that have the required network access interfaces to facilitate them in delivering their bundles of data to other networks. A DTN gateway is a DTN node equipped with multiple network access technologies which could be used to bridge two or more different networks that are using different network access technologies. A DTN gateway would be useful in a heterogeneous network setting. However, for the bundles from an end-node to reach a DTN gateway, the delivery path

would be via other end-nodes or direct contact. If direct contact is not probable, the process is likely to take a long time as contacts between intermediate nodes are also not very frequent. This is due to the sparse nature of the network in practical DTN deployments. DTN gateways are also more susceptible to being victims of security threats that may cripple their delivery effectiveness. By targeting a DTN gateway, the security attack would effectively cut off the bridge that connects 2 or more different networks, hence isolating the networks in a heterogeneous network setting. Therefore 'plug-and-play' versatility is needed for the network of nodes to react to these challenges by allowing easy deployment of well-resourced DTN nodes or DTN gateways.

Interface versatility is a worthwhile aspect to be explored in mobile networks architecture research. In interoperable heterogeneous network research, a related architecture, ParaNets [17], involves providing parallel network links over two end-nodes to address intermittent connectivity and network heterogeneity challenges. DTN is used as a representation of a challenged network in the evaluation of ParaNets. The results show that the use of heterogeneous networks in parallel improves the overall performance as ParaNets architecture exploits the spare network resources. For routing protocols in DTN, Spyropoulos et al. in [33], Garetto et al. in [34] and Samuel et al. in [35] proposed routing protocols which improve the utilization of network resources in heterogeneous settings. Hence, it is evident that there is great interest in exploiting the heterogeneity of DTN and thus 'plug-and-play' of multiple network access interfaces could be beneficial in heterogeneous network settings.

With the presence of intermittent connectivity and network heterogeneity issues, it is practical to provide an option for cascading a few end-nodes each equipped with a different network access interface, to form a supernode with multiple network access interfaces. A supernode is a DTN node formed by cascading of two or more DTN nodes with the purpose of allowing them to share their network access technologies to connect to different types of networks. Formation of this supernode therefore also allows it to function similar to a gateway and in effect such expansion can well

facilitate much better utilization of their shared resources. Unrestricted by a fixed solution, the end-nodes and gateways themselves individually can be further cascaded and incorporated with additional secondary network access interface. Thus the overall architecture is versatile enough to provide the best solution to tackle the network access limitations of a single end-node.

For network heterogeneity support, DTN architecture provides a variety of underlying delivery protocols to support a range of network access interfaces. The current DTN architecture can switch among multiple underlying delivery protocols, resembling the IEEE 802.21 [16] architecture with Media Independent Handover (MIH) component. However, the limitation in the existing DTN implementation is observed during horizontal handover among network access interfaces that use TCP delivery protocol. For parallel network links between a pair of standalone nodes, the current DTN implementation [15] can only support multiple interfaces provided they use different delivery protocols. There are complications when multiple network access interfaces all use TCP delivery protocol. The current IP discovery mechanism would fail to discover the secondary network access interface. This will pose significant issue as many current network access technologies are still TCP/IP based. Considering this issue, it is difficult to neglect the existing limitation. Hence, we made modifications to the existing IP discovery mechanism in DTN and this is detailed in Section 3.2.2.

This chapter proposes a versatile framework to expand the network access capability of DTN nodes to allow DTN networks to overcome intermittent connectivity, facilitate interoperability of heterogeneous networks and support parallel networks links. The proposed framework is detailed in Section 3.2 and examples of the application of our framework in the various operation scenarios are described. Our proposed framework exhibits flexibility allowing the cascading of supernodes and basic singular nodes to coexist in a heterogeneous DTN. In Section 3.3, comparison will be made between the existing DTN implementation and our framework in terms of the potential in overcoming the problems in challenged networks. Comparison between

ParaNets and our framework is also detailed. Section 3.4 presents the details of a testbed implementation of the proposed framework with the cascading of heterogeneous DTN nodes. Thereafter, the testbed is evaluated in Section 3.5. Section 3.6 details the implementation of a new Ethernet convergence layer as an alternative to TCP convergence layer to improve the versatility of a DTN node. The Ethernet convergence layer testbed results are presented and discussed in Section 3.7. We conclude the chapter and identify a few recommendations for future works in Section 3.8.

3.2. ARCHITECTURE

We propose a framework that is versatile enough to expand the basic network capability of a DTN node such that a single ill-provisioned DTN node can be transformed into a multi-interface node through plug and play or into a supernode or further cascaded into a gateway. This is accomplished via the concept of cascading multiple nodes as well as supporting the single independent node concept. There exists a subtle difference that differentiates the two concepts. The cascading of multiple nodes concept enhances the capability of small mobile computing devices which usually have constraints such as inadequate network access interfaces and limited memory spaces. By cascading the nodes, the nodes pool their network access interfaces and memory spaces forming a supernode. As a comparison, the single independent node is more suitable for computing devices such as laptops. These computing devices have the capability to operate as standalones since they have multiple network access interfaces and large memory spaces. Nevertheless, these nodes can also be cascaded into a supernode.

3.2.1. Cascading Concept

Relating to operational scenarios, our framework allows a node to cascade itself to another node and use the additional interface provided by the cascaded node. Facing

challenged network conditions, cascading actually empowers small mobile computing devices to overcome connectivity problems through integrating the capabilities of the devices. Illustrating this in Fig. 3.1, the walkie talkie has UHF Radio interface and the video player has WiFi interface, and they are cascaded with a Serial cable. Such a setup allows the walkie talkie to connect via WiFi as well as the video player to share video via UHF Radio and it is useful in the following scenarios.

- When there is network congestion in the UHF Radio network, the walkie talkie can rely on the WiFi interface of the video player to stay connected.

- When the WiFi network is out of range, the video player can continue to share video by tapping on the long range capability of the UHF Radio interface of the walkie talkie.

- When there is interoperability issue with network heterogeneity, the walkie talkie and the video player are cascaded to form a supernode, now acting as a gateway between the WiFi network and the UHF Radio network.

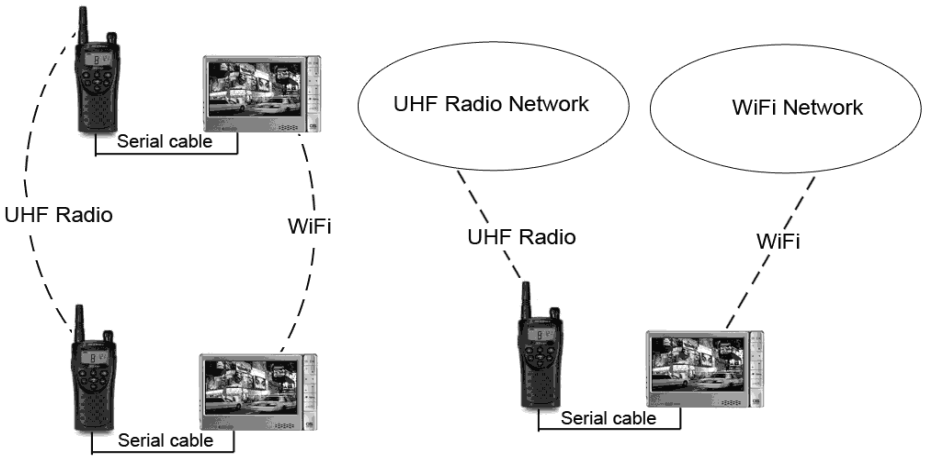


Figure 3.1. Cascaded nodes in operational context

In addition, the system is extensible with more nodes cascaded, even though only two nodes are used as a simple illustration in this case. Besides, the multi interfaces of

a cascaded supernode means parallel networks can be set up in a similar manner to the ParaNets [17] architecture mentioned in Section 3.1.

3.2.2. Modifications to Existing DTN Discovery Mechanism

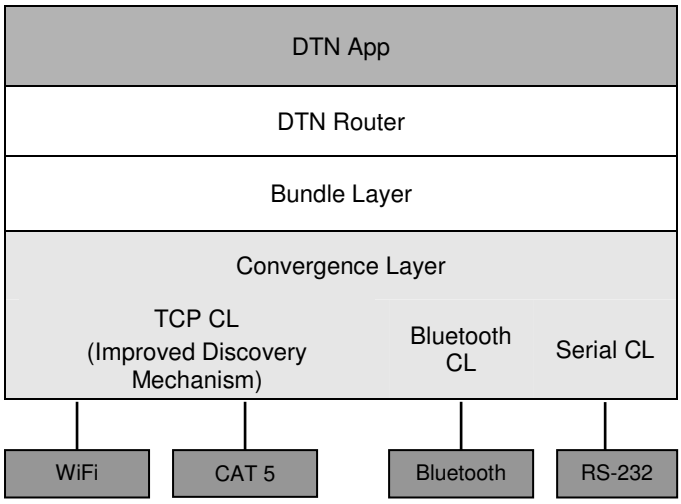


Figure 3.2. DTN framework

In current DTN architecture, convergence layers [36] define the various delivery protocols that the network access interfaces can use. For each convergence layer, discovery mechanism can be provided to discover peers on the network access interfaces that implement that delivery protocol. In the current implementation of the discovery mechanism for interfaces that use TCP/IP, upon discovery of a peer, an opportunistic link for the discovered peer will be created only if the peer has not been discovered before on the TCP convergence layer. Only the first discovery will be stored for each peer. Subsequent discovery of the same peer on a second network access interface that uses the same delivery protocol will be rejected.

In order to set up parallel network links on network access interfaces that use the TCP/IP delivery protocol, slight modification to the discovery mechanism [37] is required. The setup involves allocating separate subnets to different network access interfaces for the discovery mechanism to multicast and to listen to announcement beacons, as shown in Fig. 3.3. By using separate subnets, it clearly differentiates the

various network access interfaces that use the same delivery protocol. The modified discovery mechanism operates such that it will listen to announcement beacons on the individual subnets allocated for the respective network access interfaces. This would allow multiple network access interfaces of the same technology to perform discoveries in their respective allocated subnets. The new discovery mechanism will still create opportunistic links when a peer is being discovered on the other subnets allocated using network access interfaces of the same technology. Hence, the improved discovery mechanism is able to establish parallel opportunistic links to a peer even though the links use the same delivery protocol. Therefore, our proposed framework is able to achieve the concept of "Plug and Play" of a new interface.

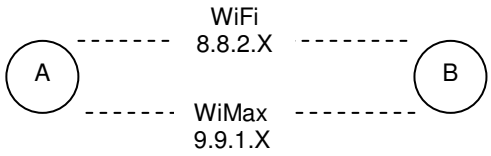


Figure 3.3. Different network interfaces multicast and listen to announcement beacons on different subnets

3.3. COMPARISON WITH EXISTING DTN IMPLEMENTATION AND RELATED WORK

Our proposed framework provides DTN the flexibility of having more options of getting the various networks connected. By cascading of nodes to form a supernode, individual nodes are empowered with the additional network access interfaces and memory spaces available to them, which we believe is a beneficial value-added feature for the existing DTN especially when operating under extreme conditions.

Our framework marks a notable improvement to the IP discovery mechanism of DTN. In the existing DTN implementation, parallel networks links are not fully supported. For instance, the existing discovery mechanism was not able to run two WiFi networks in parallel between two standalone nodes. With new modifications, our

framework fully supports multiple parallel networks links between two standalone DTN nodes, further adding more versatility to the ways nodes are connected in DTN.

With the ability of establishing parallel network links between DTN nodes, we believe DTN when operated as a standalone network can exploit the links to enhance its performance. Lightweight control information can be sent via the low bandwidth network links while actual message data can be sent via the high bandwidth network links, similar to the concept in the ParaNets [17] architecture. The DTN routing layer in our framework can then perform the selection of the appropriate network links to use for the various message types. Hence, our framework similarly gears DTN for more optimized routing as in load balancing among multiple links.

The difference between our framework and ParaNets lies in ParaNets having the advantage of being able to use DTN with external networks. For instance, the DTN network being run in parallel with the Satellite network and the Cellular network. In our framework, when the DTN is operating as a standalone network, parallel networks links are operated within the DTN network itself. To interface the heterogeneous DTN with external non-DTN networks, the use of a proxy service can support our framework in term of scalability. An example of such a proxy architecture is a heterogeneous email system [38] which has successfully integrated DTN and Internet together. The workaround of using a proxy will provide for a scalable network, integrating DTN with external non-DTN networks, similar to ParaNets architecture.

The proposed framework not only allows DTN to interoperate with other networks, it also endows DTN, when used as a standalone network, with more network and other resources as the DTN node has a much wider scope in the utilization of resources. This enhancement to DTN is achieved without the use of external networks, unlike ParaNets. Other than the possibility of exploiting the parallel network links to enhance the performance between two end-nodes, our framework has the potential of a large repertoire of shared memory in a cascaded supernode whereby data can be better distributed and shared. Besides, a cascaded supernode offers an abundance of

alternative network access between two end-nodes, which can be further exploited. The value-add in the cascading of multiple nodes can be very significant especially for small mobile computing devices.

3.4. HETEROGENEOUS DTN TESTBED IMPLEMENTATION

We have implemented a testbed of five laptops forming a heterogeneous network as illustrated in Fig. 3.4. The testbed demonstrated two supernodes that showcase the cascading concept and one normal node. Supernode 1 acts as a relay node between the normal node and Supernode 2. All the three nodes have both WiFi and UHF Radio communication capabilities, forming parallel networks links between the normal node and Supernode 1 and also between the two supernodes.

Both supernodes are formed from the cascading of two laptops with CAT 5 Ethernet cable. The first laptop uses a WiFi connection and the second uses UHF Radio connection. The normal node is a single laptop with both WiFi and UHF Radio connections. The UHF Radio modem used has serial interface and it is first connected to a Serial to TCP/IP converter and then to the laptop via a CAT 5 Ethernet cable.

Ethernet hubs as shown in Fig. 3.4 are used in the supernodes. The reason is the laptop with UHF Radio connection does not have two Ethernet slots required for connecting the UHF Radio and cascading to the other laptop. Under this circumstance, an Ethernet hub is used to connect the two laptops and UHF Radio together. To ensure the laptop with WiFi connection could not communicate via the UHF Radio, no route through the UHF Radio has been set up in the laptop.

All the network access interfaces in the testbed used TCP convergence layer as the delivery protocol for DTN. As there are multiple hops in the way the UHF Radio is connected, the discovery mechanism in DTN could not discover the peers on the UHF Radio network. Hence static links have been configured in DTN only for the UHF

Radio networks. The two WiFi networks in the testbed are in different subnets so that the normal node and Supernode 2 are not directly connected. In addition, no route has been set for the normal node to reach Supernode 2 via UHF Radio and vice versa.

In our testbed, the concept of cascading into a supernode will be put to test. On top of that, the 'plug-and-play' and discovery of multiple network access technologies will also be put to test. For evaluation, the network round trip time (RTT) and the handover time are used as a measure of the testbed performance.

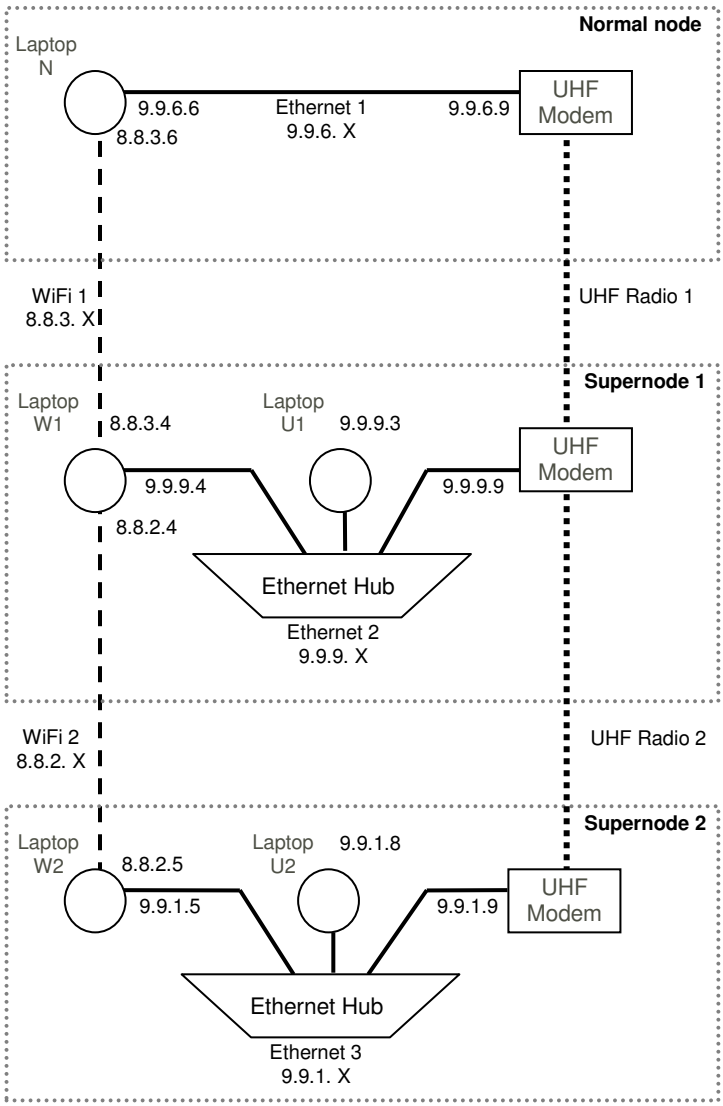


Figure 3.4. Testbed demonstrating cascaded nodes in heterogeneous DTN

3.5. HETEROGENEOUS DTN TESTBED EVALUATION

In our testbed, we successfully demonstrated the cascading concept in heterogeneous network with three scenarios. The scenarios demonstrate two concepts. Firstly, they showcase our supernodes performing the 'plug-and-play' of an additional heterogeneous interface (ie. UHF modem) to an existing laptop with Wifi interfaces. The resulting supernodes are thus enhanced with an additional heterogeneous interface as opposed to the single-WiFi- interface laptop shown in Fig. 3.4. Secondly, the scenarios demonstrate the use of Supernode 1 as a relay node between the normal node and Supernode 2 in a heterogeneous network. This section provides some insight into the performance of our framework. We measure the network round trip time (RTT) under the three different scenarios as well as the handover time during the switching of network interfaces.

The first scenario as shown in Fig. 3.5 involves the normal node being connected to Supernode 1 via UHF Radio and the two supernodes are also connected via UHF Radio. The RTT between the normal node and Supernode 2 was measured to rate the performance of such a relay and it averaged about 1.032 seconds as the data bundles were being relayed via Supernode 1. The long RTT was due to the slightly unstable UHF Radio modem used and the complicated multi-hops setup of the UHF Radio networks involving the Serial to TCP/IP converter and CAT 5 Ethernet cable.

After the first scenario, the second scenario demonstrates the loss of connectivity between the normal node and Supernode 1. This scenario involves the normal node and Supernode 1 changing their connection from UHF Radio to WiFi to get connected. Shown in Fig. 3.6, the second scenario showcases the normal node and Supernode 1 connected via WiFi and the two supernodes connected via UHF Radio. In this setup, Supernode 1 demonstrates the concept of cascading two laptops with different network access interfaces to form a heterogeneous supernode. The RTT between the normal node and Supernode 2 averaged about 602 milliseconds. Again, the slightly unstable UHF Radio modem used accounted for the moderately long RTT. Other than the

instability observed in our UHF Radio modem, the overall testbed was reliable as there was no data loss between the nodes.

Continued from the second scenario, the last scenario demonstrates the loss of connectivity between Supernode 1 and Supernode 2. This scenario involves the two supernodes changing their connections from UHF Radio to WiFi to get connected. Shown in Fig. 3.7, the last scenario showcases the normal node, Supernode 1 and Supernode 2 all being connected via WiFi. The RTT between the normal node and Supernode 2 averaged about 86 milliseconds. The measured RTT is relatively short as the WiFi adapter in our testbed is much more stable than the UHF Radio modem used.

For determination of the handover time, we measured the time taken in switching between WiFi and UHF Radio connections for the normal node and Supernode 1. The handover time from WiFi to UHF Radio averaged about 10 seconds and from UHF Radio to WiFi averaged about 6 seconds. The long handover time takes into account the detection of the loss of network connectivity, breaking contact on the network interface that is disconnected and opening new contact on another network interface. Contributing to the long handover time are the long RTT of about 500 milliseconds incurred by the UHF Radio modem and the IP discovery for WiFi which has been configured to announce beacons at 2- second intervals. The opening contact on UHF Radio took a longer time due to the slower transmission rate and the multi-hops setup of the UHF Radio networks. Hence it explains the difference in the handover times recorded. On top of the measurement taken, there is no data loss during the vertical handover as DTN node possesses data store-and-forward capability.

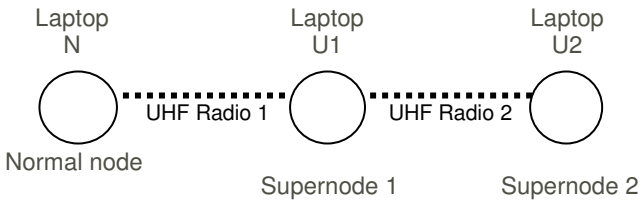


Figure 3.5. Scenario 1

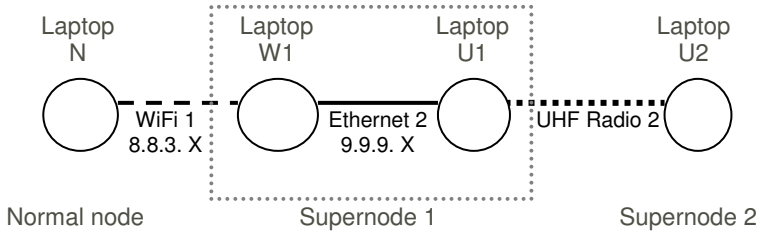


Figure 3.6. Scenario 2

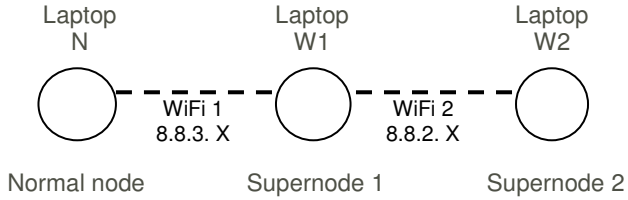


Figure 3.7. Scenario 3

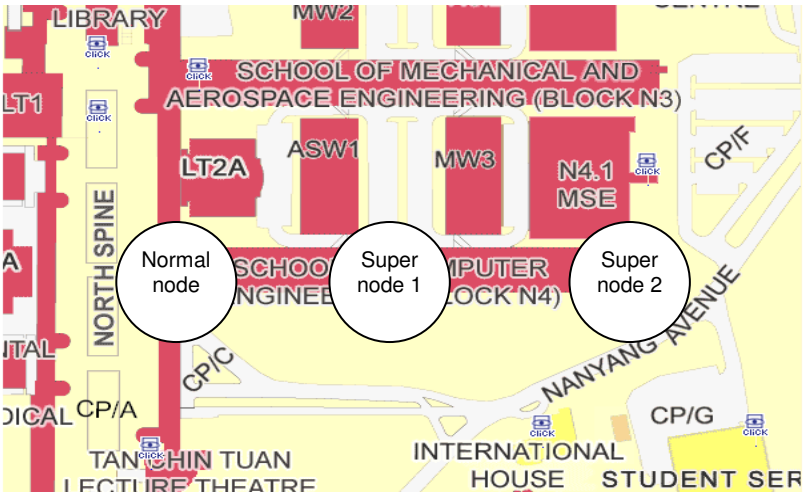


Figure 3.8. Map showing the location of the testbed

3.6. AN ALTERNATIVE TO TCP CL – NEW ETHERNET CL

The current TCP convergence layer implementation in DTN has the following underlying protocols, TCP at the Transport Layer and IP at the Network Layer. However, the traditional TCP/IP in the Internet model is deemed to be unsuitable for challenged networks as nodes can hardly establish end-to-end connections under intermittent connectivity condition. With this in mind, the use of TCP as the underlying convergence layer in DTN is questionable. The protocols at the TCP convergence layer may not be appropriate because DTN does not require its underlying convergence layer to perform end-to-end connectivity for the source and the destination nodes to communicate. As DTN already has some of its own store-and-forward reliability mechanism and its own naming convention, a more light weight convergence layer can be introduced as an alternative to the existing TCP convergence layer in DTN.

TCP/IP protocols can be left out with the implementation of the new Ethernet convergence layer as shown in Fig. 3.9, as end-to-end connectivity is not necessary in DTN. The Ethernet at the Data Link Layer can provide DTN with basic data transfer using MAC addressing. Nevertheless, it is still important to consider whether the reliability mechanisms in TCP such as error detection, flow control and congestion control are really not needed by DTN. Taking this into consideration, we propose a new Ethernet convergence layer which aims to be a basic functional convergence layer for DTN. With the removal of TCP/IP, in its place is a bare minimum reliability mechanism integrated with the Ethernet at the Data Link Layer. This new convergence layer is basic enough to be a preliminary platform for future development of enhancing reliability mechanisms.

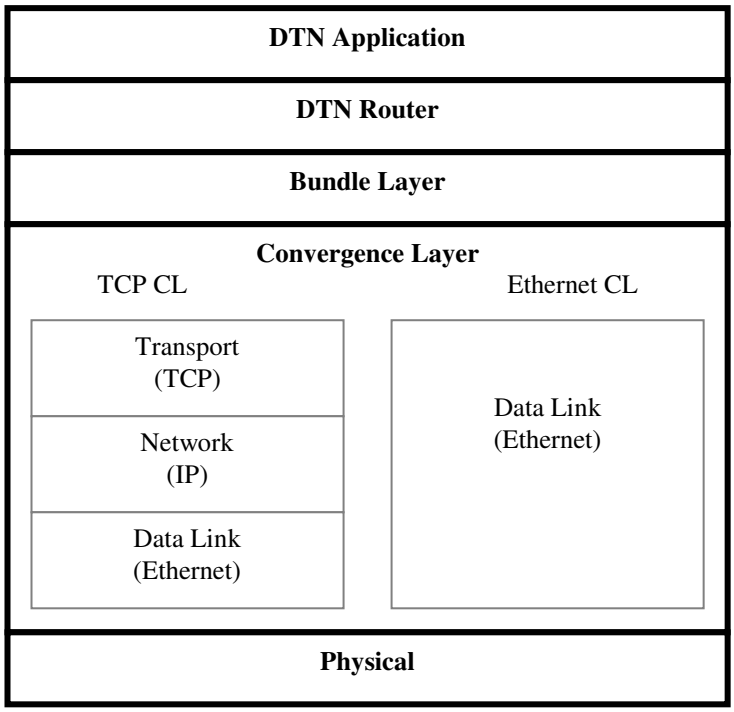


Figure 3.9. Difference between TCP and Ethernet convergence layers

The proposed Ethernet convergence layer is an alternative that can replace TCP convergence layer. Similar to TCP convergence layer, it inherits the Stream convergence layer class in the inheritance hierarchy [39]. The Stream convergence layer is a common class in DTN which standardizes a procedure for data streaming on different types of network access interfaces. Within the Stream convergence layer class, it provides streaming, a lightweight acknowledgment mechanism and segmentation of DTN bundles. Nevertheless, the Stream convergence layer class allows certain flexibility for the inheriting convergence layer class to implement a reliable data transfer at the underlying layer.

The following sections will examine the considerations taken and proposes the design of the new Ethernet convergence layer. The proposed Ethernet convergence layer will support the Stream convergence layer class that it inherits in providing a reliable delivery of the bundles for the bundle layer above. In the process of achieving

reliable data transfer, the Ethernet convergence layer has to meet the requirements of the Ethernet frame specification. Without TCP as the overlaying layer, our Ethernet convergence layer aims to use a more lightweight approach to achieve the same reliability as the TCP convergence layer.

3.6.1. Facilitating Store-and-Forward mechanism

In DTN, the convergence layer provides procedural means to transfer data for the bundle layer above. Here, the convergence layer has to facilitate the bundle layer to achieve the store and forward mechanism in DTN. The bundle layer requires the convergence layer to monitor the network link availability, so it will know when to store the bundles instead of assuming the bundles are successfully sent when the network connectivity is intermittent. However, monitoring of network availability is not provided in the Stream convergence layer class and is left for the inherited class to implement. The Stream convergence layer provides only a lightweight acknowledgment mechanism which operates as follows: the receiver can wait for more frames to arrive before sending a single acknowledgment indicating the total data size of the frames it received. Considering this acknowledgment mechanism for monitoring the network link availability, it will not be suitable as the receiver does not acknowledge every single frame. Without the acknowledgement, there is no means for the timeout mechanism to signal the loss of network connectivity.

As mentioned, supporting store-and-forward mechanism in DTN requires monitoring of network availability in the underlying layer. The monitoring can be achieved in one way by using beacons, checking if the beacons sent out by the neighboring node are periodically received. An alternative way is to use an acknowledgment mechanism, with the sender expecting acknowledgments from the receiver during data transmission and having no acknowledgment received upon timeout will signal that the network is unavailable. To monitor the network link availability, we implemented the former method, using a beacon mechanism inside the Ethernet convergence layer class. As beacons are periodically sent, the network link

can be assumed to be unavailable when there is still no beacon received upon timeout. The benefit of using beacons for the monitoring is the beacon mechanism serves dual purpose, since the discovery mechanism for opportunistic network links would also require its service.

3.6.2. Recovery from duplicate and missing frames

In data transfer for the bundle layer above, the convergence layer is expected to provide its own means of facilitating delivery guarantee. The Stream convergence layer that our Ethernet convergence layer is inheriting keeps track of only the total data size of the frames being received but is not responsible for checking duplicate and missing frames. If a reliable data transfer service is required, the inheriting convergence layer class is supposed to implement its own mechanism to guarantee the reception of every single frame. In the case of TCP convergence layer, TCP provides delivery guarantee by resending until the reception of the packet is acknowledged by the receiver. As TCP is omitted in our Ethernet convergence layer, a similar acknowledgment mechanism or other alternative solution has to be included to ensure delivery guarantee in our implementation.

For the Ethernet convergence layer's quality of service, we propose using sequence identifiers to detect duplicate or missing Ethernet frames. For duplicate frames recovery, the duplicate frames are dropped upon receiving frames of repeated sequence identifiers. For missing frame recovery, it is more complex, albeit it can be done without an additional acknowledgment mechanism at the Ethernet convergence layer. An "ask for resend" mechanism can be used instead when the receiver notices skipped sequence identifiers. This recovery mechanism is more lightweight as it saves the receiver from acknowledging every single frame it received, and is feasible because monitoring of network availability is already in place through the beacon mechanism. During data transmission, if the sender can still receive beacons periodically from the receiver and does not hear any resend request from the receiver, the network link cannot be unavailable and it is quite likely that the receiver never ask for any resend

and has received all the frames successfully. Even in the case when a resend request sent by the receiver is missing, the Ethernet convergence layer will still recover in an alternative way as described in the following: The receiver, who has not received the missing frame, will wait for timeout and break contact. It will then wait for the next reconnection for further data transmission. In this way, the sender does not receive acknowledgment from the receiver and will resend the missing frame in the next reconnection.

3.6.3. Fragmentation

In raw Ethernet socket programming, an Ethernet frame needs to be created manually. Referring to Ethernet frame specification [40] in Fig 3.10, a single Ethernet frame consists of a Datalink header, user data payload and frame check sequence. The Datalink header is mainly reserved by the source and destination nodes' MAC addresses and does not have enough room for the additional header fields that might be required for the Ethernet convergence layer in DTN. Taking this into account, the additional header fields required has to be located at the front of the user data payload. Other than the short Datalink header, it is noted that the allocated 1500 bytes of user data specified may not be enough for transferring a large bundle payload. Therefore, the DTN bundles have to be fragmented into Ethernet frame size before data transfer is performed at the convergence layer.

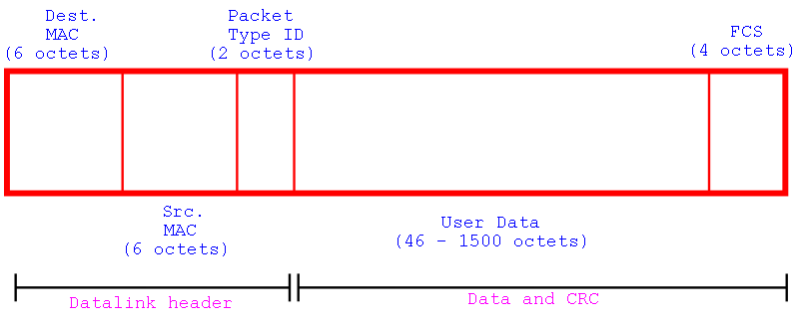


Figure 3.10. Ethernet Frame Specification (IEEE 802.3)

In the Stream convergence layer class, it provides segmentation of the DTN bundles into small segments. It is optional for the inheriting convergence layer class to perform further fragmentation on the segments. Considering the Ethernet frame size constraint, the segment size is configurable to fit the size of an Ethernet frame and hence further fragmentation is needless. Although such configuration is a possible solution, an alternative method is used in the implementation of our Ethernet convergence layer which does not alter the default segment size configuration. Instead, the segments will be further fragmented to fit the Ethernet frame size requirement at our convergence layer class. This approach is more advantageous as it is more flexible without restricting the segment size at the Stream convergence layer class.

3.7. ETHERNET CL TESTBED AND RESULTS

We created a testbed to study how DTN using our Ethernet convergence layer fares against the current IP Wireless Ad-hoc Network and DTN using TCP convergence layer. The focus in this testbed is on data transmission performance under intermittent connectivity condition and across multi-hops. In this testbed, we have 6 laptops using Wi-Fi network access technology. The testbed is configured using static link and data transmission from the source to destination is performed over 5 hops. The first demo showcases file transfer while the second demo showcases video streaming.

3.7.1. File transfer

For the file transfer demo in Fig. 3.11, we transfer a 1.5 MB file from the first node to the last node with all the 4 relay nodes in between running a script that alternatively turns their Wi-Fi interfaces up for 5 seconds and down for 10 seconds over intervals, resulting in intermittent connectivity. We compared the performance of DTN using Ethernet convergence layer with DTN using TCP convergence layer and Wireless Ad-hoc Network using TCP. The time-out values used for both convergence layers are set at 5 seconds. For each case, there are 10 attempts at transferring the file. DTN using

Ethernet convergence layer is able to send the file in an average of 2 minutes and 3 seconds. DTN using TCP convergence layer always had the file stuck at one of the middle relay nodes and the file did not reach the destination after waiting for 10 minutes. Wireless Ad-hoc Network using TCP is incapable of transferring the file as there is no guaranteed end-to-end connectivity between the source and destination.

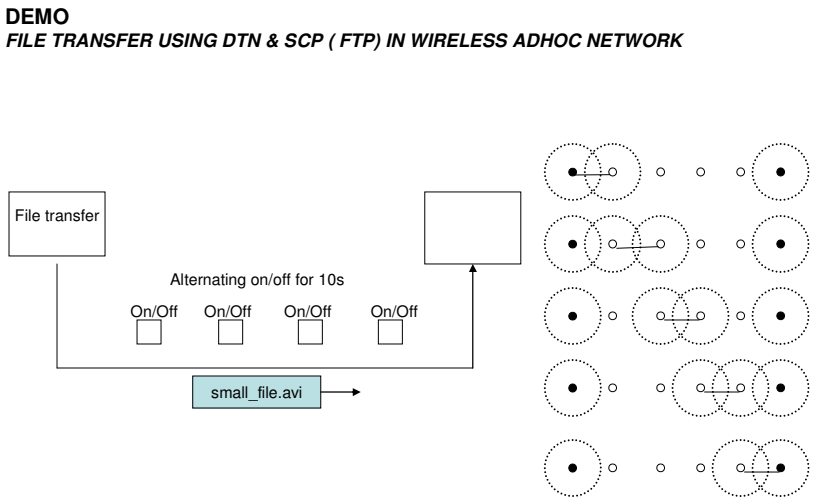


Figure 3.11. File transfer using DTN and SCP (FTP) in Wireless Adhoc Network

3.7.2. Video streaming

For the video streaming demo as shown in Fig. 3.12, we stream a 45 seconds video from the first node to the last node with one of the 4 relay nodes in between running a script that periodically alternates between Wi-Fi interface up for 30 seconds and down for 20 seconds, causing intermittent connectivity. We compare the performance of DTN using Ethernet convergence layer with DTN using UDP convergence layer and Wireless Ad-hoc Network using UDP. The video is streamed using VLC player and we set 30 seconds buffering for the streaming. DTN using Ethernet convergence layer and DTN using UDP convergence layer does not have much visible difference in our evaluation and are able to stream most of the video. Wireless Ad-hoc Network using

UDP is at most only able to stream 30 seconds of the video because connectivity between the source and destination is lost for the next 20 seconds.

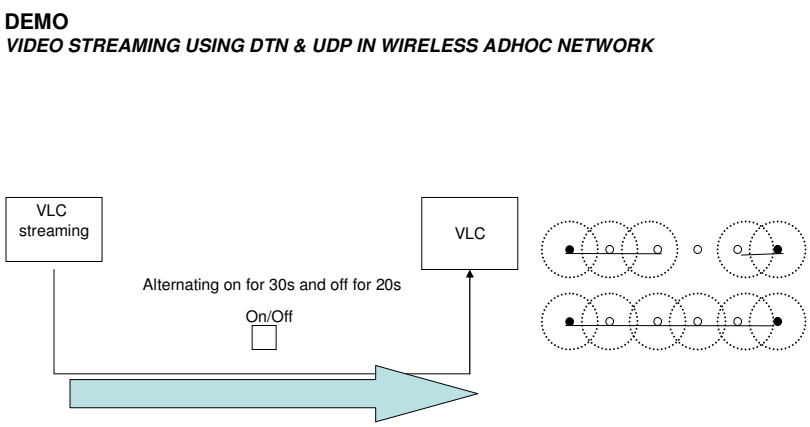


Figure 3.12. Video streaming using DTN and UDP in Wireless Adhoc Network

3.8. CONCLUSION

Our framework enhances DTN to be versatile enough to meet the demands of the myriad scenarios in challenged networks. The concept in our framework is to cater to different combinations of cascaded nodes and single independent nodes in a heterogeneous network environment. For additional versatility, parallel network links between nodes can be used to overcome intermittent connectivity. In our heterogeneous DTN testbed implementation, we have demonstrated our cascading concept in a heterogeneous network. In addition, we have implemented an Ethernet convergence layer for DTN to be used as an alternative to the existing TCP

convergence layer. In our evaluation of DTN using Ethernet convergence layer, it is capable of delivering data under highly intermittent connectivity condition.

We believe our framework has open up another dimension of work in DTN. The potential of the cascading concept with the parallel network links caters for future development of DTN routing protocols to better coordinate the usage of network resources. Another area for future research is the distribution and sharing of data in the memory spaces of cascaded nodes.

Chapter 4.

A Queuing Mechanism to Alleviate Flooding Attacks in Probabilistic Delay Tolerant Networks

4.1. INTRODUCTION

In Delay Tolerant Networks (DTN) [1], many probabilistic routing protocols [23][24][41][42] have been developed and proven to produce high message delivery rate. These probabilistic routing protocols typically keep track of the opportunistic contacts among the nodes in a network and thereafter use the predicted delivery probability for making the forwarding decision. Although the computed delivery likelihood is solely used for the routing purpose, it can be exploited and used for other purposes as well. One area which can capitalize on it is the defense against security threats.

Probabilistic routing protocols can expose DTN to newly enhanced security threats. As these probabilistic routing protocols generally depend on the probabilistic metric of encountering a node, a malicious node can exploit the probabilistic metric to increase its chances of performing a successful attack on the network. One such newly enhanced security threats is flooding attack. Knowing the probabilistic metric of its victims, a malicious node can predict the mobility and plan a variety of more effective flooding attacks on the network. Further, it also allows the malicious node to introduce advanced manipulation techniques to enhance its flooding attack. Hence the threat posed by flooding attack needs to be revisited.

We proposed a queuing mechanism that is based on Radia Perlman's network design with byzantine robustness that can withstand malicious routers as well as alleviate the effect of flooding attacks through fair allocation of resources [28]. Our queuing policy is slightly different from Radia Perlman's design in term of allocation of resources. It involves a drop tail queue management technique using our distinctive formula for the selection of the preferred message to be dropped when the buffer is full.

Quite a few queuing policies have been proposed for DTN. However, the current queuing policies are generally not for the purpose of alleviating flooding attacks. In [29], some queuing policies were evaluated for PRoPHET and Epidemic routing protocols. In their evaluation, probabilistic routing used with appropriate choice of queuing policy can produce good network performance. In [30], an optimal queuing policy that involves an estimation of global information is proposed. They have evaluated their queuing policy with simple queuing policies, drop front, drop last, drop oldest and drop youngest. However, their proposed policy is evaluated with Epidemic routing protocol which is not probabilistic. Without the probabilistic feature, it is difficult to compare our queuing policy with theirs as the flooding attacks evaluated in this chapter possess the predictive trait.

This chapter introduces our new queuing mechanism that can defend against the possible flooding attacks in DTN. Our feature in our proposal is a buffer management technique which has a probabilistic characteristic. In this chapter, we examine the routing protocol, Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [24], of DTN and derive a solution that is capable of countering flooding attacks against the protocol. Possible flooding attack techniques against the PRoPHET protocol will be discussed in Section 4.2. Subsequently, we consider the factors in the design of the proposed mechanism in Section 4.3 before deriving a solution in Section 4.4. Our simulation scenario is described in Section 4.5 and the simulation results are evaluated in Section 4.6. Thereafter, we conclude our chapter in Section 4.7.

4.2. POSSIBLE FLOODING ATTACKS AGAINST PROPHET PROTOCOL

In this section, four types of flooding attacks will be discussed. Random flooding and selective flooding of destination nodes are discussed in Sections 4.2.1 and 4.2.2 respectively while non-existent destination node flooding and spoof flooding are discussed in Sections 4.2.3 and 4.2.4. Selective destination node flooding and non-existent destination node flooding are security threats designed to exploit the loopholes in PRoPHET routing protocol. Advanced defensive mechanism is required to prevent the adversary from exploiting the loopholes in PRoPHET as well as the normal random flooding attack. These security threats are real threats and there is a need to study the approaches used in these flooding attacks. The following sections describe the different techniques employed by the different flooding attacks.

4.2.1. Random Flooding

In this approach, flooding the network is done in a random nature. The malicious node continuously creates fake messages to randomly selected destination nodes. Upon meeting a victim, the malicious node will transfer the fake messages that the victim has a higher predictability of delivering to the destinations. After this, the fake messages will be further relayed through the victim's peers until the messages reach their destinations. The relaying process causes fake messages directed into different paths since they are destined for different nodes.

4.2.2. Selective Destination Nodes Flooding

This attack aims at flooding selective targets. The node that has the highest involvement is a noteworthy node that is highly active in the network and hence is the favored target. For all of PRoPHET's nodes, by summing the delivery predictabilities, the malicious node knows which nodes are actively having frequent meetings with the rest. The node with the highest sum is the most active node among all, and so it will be

the selected victim. Thereafter, the malicious node will create fake messages to flood the victim. To increase the likelihood of the fake messages being forwarded to the victim, the malicious node will set the destinations for the fake messages to be the node which the victim has the highest delivery predictability.

4.2.3. Non-Existent Destination Node Flooding

Flooding attack may be more sophisticated which involves the malicious node setting the destinations for the fake messages to be a non-existent node [27]. Unlike previous attacks, the fake messages will never reach their destinations, and hence it achieves the purpose of allowing the fake messages to remain longer in the network. In order to bluff the victim to accept the fake messages, the malicious node has to increase the victim's delivery predictability for the non-existent destination node. This is achieved by exploiting the transitive property of the PRoPHET protocol. The fake messages will stay in the network till they time out or are discarded by the queuing policy implemented.

4.2.4. Spoof Flooding

A trickier approach to flooding attack is spoofing of identities by a malicious node. By modifying the sources for the fake messages created, the malicious node can bluff the victim to think that the messages are from different sources. In this manner, it would be difficult for the nodes in the network to trace and detect the messages. To further mask the attack, randomly selected destinations are set for the fake messages created. These fake messages will diverge and travel in different paths, and therefore will not be easily detectable.

4.3. FACTORS FOR CONSIDERATION

4.3.1. Message Ferry

Message ferry [43] involves messages collected from an end node to be relayed to another end node. The messages collected are normally in bulk as messages are consolidated before a message ferry is performed. When relaying the messages, it is in a way hard for the receiving node to distinguish it from a flooding attack which also transmits large payload. However, the node that performs a message ferry should not be mistaken as a malicious node that does a flooding attack. Therefore any measure taken for handling flooding attack has to take the case of message ferry into consideration.

4.3.2. Malicious Nodes Increasing Delivery Predictabilities

PRoPHET protocol has a transitive property whereby node A will increase its own delivery predictability for node C when it encounters node B which has encountered node C before. The increment will depend on node B's delivery predictability for node C. As such, a malicious node can increase its victim's delivery predictability for a particular node even for the case where its victim has not met that node before. However, a very high delivery predictability of greater than 1 should not be expected from the malicious node, as the calculated metric of the delivery predictability always stays within the range between 0 and 1. It is required to verify that this condition is met all the time.

$$\text{Delivery predictability: } 0 \leq p \leq 1$$

4.3.3. Messages' Source or Previous Node Factor

The obvious clue of a possible flooding attack is when the messages received are originated mostly from the same source. Two cases are identified as follows:

Case 1 - Single hop: Path {M, V}. For the case of the messages' path in a flooding attack that has only a single hop, malicious node M who is flooding victim V is the messages' source as well as the previous node of victim V, and hence checking the previous node has the same effect as checking the messages' source. If victim V notices it has received too many messages from malicious node M by checking the messages' source, victim V will know there is a possibility of a flooding attack. Similarly if victim V checks the previous node from which the messages come from, victim V will observe many messages from malicious node M as well.

However, malicious node M can spoof many different identities for the messages that are sent to flood a victim. As a result of spoofing, victim V is fooled to think the messages sent by the malicious node M are from different sources. Hence checking the message's source might be as good as not checking. Similarly if victim V checks the previous node where the messages come from, victim V will not notice the messages are from malicious node M.

Case 2 - Multiple hops: Path {M, N1…NK, V}. For the case of the messages' path in a flooding attack involving multiple hops, the immediate node N1 which is the direct neighbor of the malicious node M has the duty of preventing the fake messages sent by malicious node M from reaching victim V. As malicious node M who intends to flood victim V is the messages' source as well as the previous node of node N1, having node N1 checking the previous node has the same effect as checking the messages' source.

Nevertheless, checking the previous node will be less vulnerable to spoofing of identities in flooding attack as compared to checking the messages' source when multiple hops are involved. If all the nodes N1 to NK and victim V check the messages' source, all of them will be fooled to think that the messages sent by the malicious node M are from different sources. However, if all the nodes N1 to NK and victim V check the previous node where the messages come from, only the immediate node N1 is fooled to think that it is receiving the messages from different nodes. As the

messages are forwarded in the hops that follow, the rest of the nodes will observe that the receiving messages are from the same respective previous nodes, and thus assign less priority to the messages.

However, checking on previous node is not a complete solution. The previous node, who forwards many messages, might in actual fact, be performing a message ferry service, and it is hard for the receiving node to distinguish it from a flooding attack. An additional solution has to be included for the message ferry service to be recognized.

4.3.4. Storage Space Issue

In a flooding attack, the fake messages from a malicious node typically fill up the storage. It is undesirable to have an indifferent storage management scheme that allows fake messages to stay in the storage and deprive normal messages of storage space. Hence there is a need for a storage handling procedure to decide which messages to keep and which messages to remove. However, the storage management scheme will have difficulty making a discreet decision as it is hard to identify a malicious node and distinguish the fake messages from normal messages. As such, giving a discreet judgment and totally omitting messages from a suspicious node is too extreme to be a solution to adopt. It is preferred to allocate storage space base on the suspicion level of the nodes; i.e. having suspicious nodes being allocated less storage space rather than no storage space.

4.4. PROPOSED SOLUTION

Our proposed solution incorporates the idea that suspicion level should be raised when there are too many messages from a particular node that is seldom encountered. This idea originates from Radia Perlman's concept of fair allocation of resources [28] that can be used to handle flooding attack. The difference is our solution allocates

storage space based on the number of encounters, with more storage space allocated to nodes that are encountered frequently.

4.4.1. Checking Previous Node and Event of Message Ferry

The allocation of storage space in our proposed solution will limit the total size of the messages from the same previous node but not the messages from the same source. Since every malicious node has to be the previous node of its neighboring nodes, the duty of limiting the effect of a flooding attack is performed by the nodes at the very first hop. For the hops that follow, nodes are freed from the burden of checking the messages' sources. With respect to the issue of differentiating a message ferry service from a flooding attack, we assume an authentication method is provided for ensuring a secure transmission of messages during a message ferry routine.

4.4.2. Capitalizing on Delivery Predictability

The delivery predictability value that is already given in the PRoPHET protocol is being exploited to full advantage and there is no additional information exchange overhead. As the PRoPHET protocol is not based on global knowledge, along with our proposed queuing policy, it is a distributed scheme suitable for DTN. In our proposed solution, the number of encounters is a factor to be considered in the allocation of storage space. To formulate the estimation of the number of encounters, the delivery predictability is used as a factor. Although the problem can be alternatively solved by making do with the actual number of encounters using an incrementing counter, an estimation using delivery predictability is more applicable as it is necessary to take into account the aging property of the PRoPHET protocol.

4.4.3. New Queuing Policy Proposed

To implement this concept, we proposed a new queuing policy PRED. It is used for deciding which message to remove when the storage is full. A calculated metric is used as the determinant of which message to drop in our queuing policy. Equation (4.1)

shows the calculation of this metric (*s*) for a particular node, using delivery predictability (*p*) and total size of the messages (*m*) from that node. Our queuing policy will remove a message that is from the node with the greatest *s* metric. In the event of a tie, a tie breaker selects the older message to be the message for removal.

$$s = -\log_{1-p} m \qquad (4.1)$$

The derivation of equation (4.1) is as follows. Equation (4.2) forms the key concept and is used as the start to derive equation (4.1), where the number of encounters (*n*) will be estimated using the delivery predictability in PRoPHET. The base of the logarithm expression in equation (4.2) *C* is an arbitrary positive constant greater than or equal to 2. The node with the greatest s_c value is one with the largest total size of messages relative to the number of encounters, and hence it has the highest likelihood of being the malicious node.

$$s_c = \frac{\log_C m}{n} \qquad , C \geq 2 \quad (4.2)$$

Next, equation (4.3) shows the first property [24] of the PRoPHET protocol. Here, we introduce node A and node B and $P_j(A,B)$ represents node A's delivery predictability for node B after *j* encounters. To simplify the terms, *p* is used to represent $P_n(A,B)$. In PRoPHET, equation (4.3) is being used to compute and update node A's delivery predictability for node B upon their encounter. In equation (4.4), it assumes node A's delivery predictability for node B is initialized to zero. The number of encounters (*n*) in equation (4.2) is calculated using equation (4.5). Equation (4.5) can be derived from equations (4.3) and (4.4). The calculated value of *n* is only an estimate, noting that the aging property [24] and the transitive property [24] of the PRoPHET protocol also affect the delivery predictability as well. Alternatively, the estimation may use a different derivation that includes the transitive property.

Let $p = P_n(A,B)$

$$P_k(A,B) = P_{k-1}(A,B) + (1 - P_{k-1}(A,B)) \times \lambda \square, 0 \le \lambda \le 1 \quad (4.3)$$

$$P_0(A,B) = 0 \quad (4.4)\square$$

From (4.3) & (4.4): $P_n(A,B) = 1 - (1-\lambda)^n$

$$n = \log_{1-\lambda}(1 - P_n(A,B))$$

$$n = \log_{1-\lambda}(1-p) \quad (4.5)$$

Equations (4.2) and (4.5) are used to derive equation (4.1). Since λ is an initialization constant that is defined to be the same for node A's delivery predictabilities for all nodes, equation (4.6) can be further simplified to equation (4.7) by introducing a positive constant C_λ. As our queuing policy is only interested in finding the node with the greatest s_c value, constant C_λ is redundant when making comparisons among the nodes. Hence equation (4.1) is adequate for our queuing policy to decide which message to remove.

From (4.2) & (4.5): $$s_c = \frac{\log_C m}{\log_{1-\lambda}(1-p)}$$

$$s_c = \frac{\log_{1-\lambda} m}{\log_{1-\lambda} C \times \log_{1-\lambda}(1-p)}$$

$$s_c = \frac{\log_{1-p} m}{\log_{1-\lambda} C} \quad (4.6)$$

$$s_c = -C_\lambda \log_{1-p} m \quad , C_\lambda > 0 \quad (4.7)$$

$$s = -\log_{1-p} m$$

4.4.4. Distort Delivery Predictabilities Flooding

As our queuing policy proposed uses delivery predictability as a factor for selecting the message to remove when the storage is full, a new flooding attack can be developed based on this property. Malicious nodes can perform a collaborated flooding attack whereby they increase the victims' delivery predictabilities for participating malicious nodes using the transitive property of the PRoPHET protocol. With higher delivery predictabilities, our queuing policy will cause the nodes to allocate more storage space for the malicious nodes.

However, the impact of this flooding attack is left to be seen. Although a significant percentage of the storage space will be allocated to the malicious nodes using this attack, the allocated storage space relative to the total size of messages from the malicious nodes is considerably small as the delivery predictabilities cannot be increased to values greater than 1. There will still be a considerable percentage of storage space allocated for the normal nodes under this flooding attack.

4.5. SIMULATION

We ran a simulated network of 40 pedestrians, 20 rovers and 3 trams using Opportunistic Network Environment (ONE) simulator version 1.3.0 [47]. The

pedestrians' mobility speeds were within 1.8 - 5.4 km/h and they were equipped with computing devices that possess 64 MB storage space. The rovers' mobility speeds were within 10 - 50 km/h and they were installed with computing devices with 512 MB storage space. The trams' mobility speeds were within 25 - 36 km/h and they were installed with computing devices that with 1024 MB storage space. In each simulation, 6000 random messages of message size 10 k – 1 MB were created by the nodes within a simulated time of an hour. Five types of flooding attacks, as mentioned in Section 4.2 and Section 4.4.4, are performed. Up to 30 malicious pedestrians were added into the network for the simulations. In the simulations, the first hour is used for initialization of the nodes' delivery predictabilities, the creation of the messages were done in the second hour, and the delivery of the messages were done in the second and third hour.

Our simulations present different queuing policies for the PRoPHET routing protocol. We analyze our simulation results by comparing our queuing policy PRED with the queuing policies that Lindgren et al. evaluated in [29] for the PRoPHET protocol. We further benchmarked PRED against an IDEAL queuing policy which portrays the ideal scenario. Hence it will show the relative performance of PRED against the optimal case. We measured the performance of the queuing policies based on the percentage of messages delivered successfully.

FIFO - FIFO [29] is a queuing policy which removes the oldest message first. The first message that enters the queue is the first message to be removed.

MOFO - MOFO [29] is a queuing policy which removes the most forwarded message first. Each node kept a count for the number of times each individual message has been forwarded. The message that was forwarded the most times is the message to be removed first.

LEPR - LEPR [29] is a queuing policy which removes the message with the lowest delivery predictability. A message that has the lowest delivery predictability has the smallest chance of being delivered to the destination, and hence it is the message

selected to be removed first. In our simulation, we included an additional condition to ensure that the messages being removed must have been forwarded at least once.

FAIR - FAIR is a queuing policy which allocates storage space equally among the nodes. It removes the oldest message that is from the node whose messages occupy the most space in the storage.

IDEAL - IDEAL is a queuing policy which will always choose a message that is from a malicious node if the storage contains at least one such message. For the case whereby none of the messages is from the malicious node, it will remove a message based on MOFO instead of just choosing a random message. This ensures the message to be dropped is a well chosen message at all times.

4.6. EVALUATION OF RESULTS

For a general overview of our simulation results, Figs. 4.1, 4.2, 4.3, 4.4 and 4.5 show our queuing policy (PRED) has the highest percentage of messages delivered and in contrast, FIFO has the lowest. Figs. 4.1 and 4.2 show great similarity in the results, with FAIR being the second best performing queuing policy followed by an observed marginal difference between MOFO and LEPR. In addition, Figs. 4.1, 4.2, 4.3 and 4.5 show PRED performs close to the ideal scenario IDEAL. Only Fig. 4.4 shows a slightly more significant margin between them.

Fig. 4.3 shows in non-existent destination node flooding attack, LEPR has noticeably higher percentage of messages delivered than MOFO and therefore differs from the results produced in other types of flooding attacks. For most of the time, the delivery predictability for the non-existent node is probably very low and consequently LEPR will likely choose to remove the messages from the non-existent node first. Although the attacker has fooled its victims to set a high delivery predictability for the non-existent node, the aging property of the PRoPHET protocol reduces the delivery

predictability over time. The attack may be more effective against LEPR if the malicious nodes collaborate and associate themselves to a common non-existent node, causing more increments to the delivery predictability.

Fig. 4.4 shows that in spoof flooding attack, FAIR has lower percentage of messages delivered than MOFO. The possible explanation for this lies in the malicious node spoofing multiple identities and fooling its victims to think that the messages they received are from different nodes. Although FAIR allocates fair amount of storage space to nodes, the different allocations of storage space could be occupied by messages from the same malicious node. In comparison, MOFO prevents the fake messages from travelling too many hops and spoofing of identities has no additional effect on the queuing policy. Hence MOFO is moderately suitable for spoof flooding attack. Another observation is PRED is far from being ideal as messages delivered is much less than the IDEAL scenario. As PRED is probabilistic based, it does not have the deterministic mechanism to detect the spoof node and hence the spoof flooding attack is effective.

Fig. 4.5 shows in distort delivery predictabilities flooding attack, FAIR has almost the same percentage of messages delivered as PRED. This is because the nodes' delivery predictabilities for the malicious nodes are distorted values. The distorted delivery predictabilities prevent PRED from allocating the rightful amount of storage space to store the messages from the malicious nodes. As a result, the malicious nodes get more than they deserved. However, PRED still performs well against this flooding attack as the distorted delivery predictability is not significantly large with the highest any malicious node can distort being capped at 1.

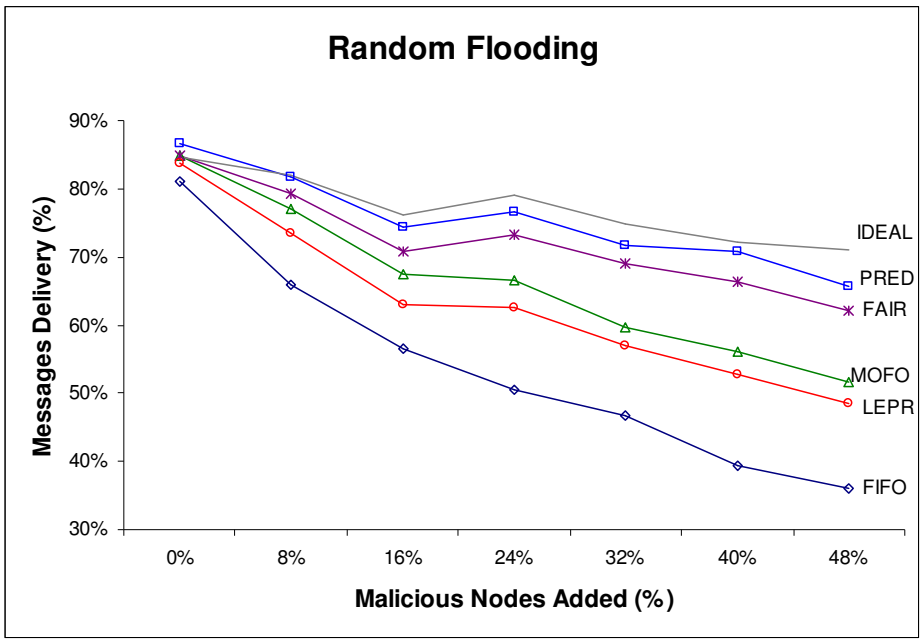


Figure 4.1. Random Flooding

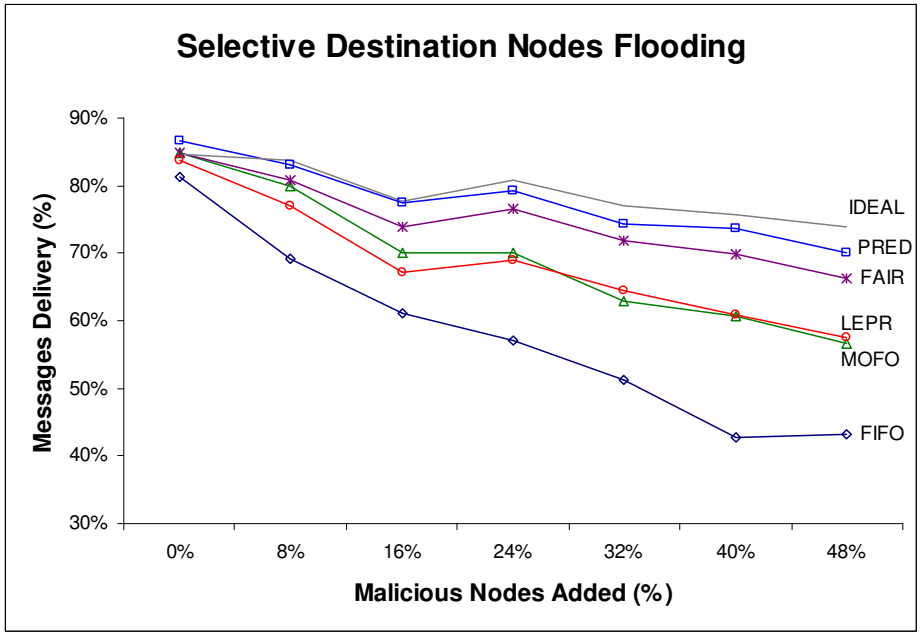


Figure 4.2. Selective Destination Nodes Flooding

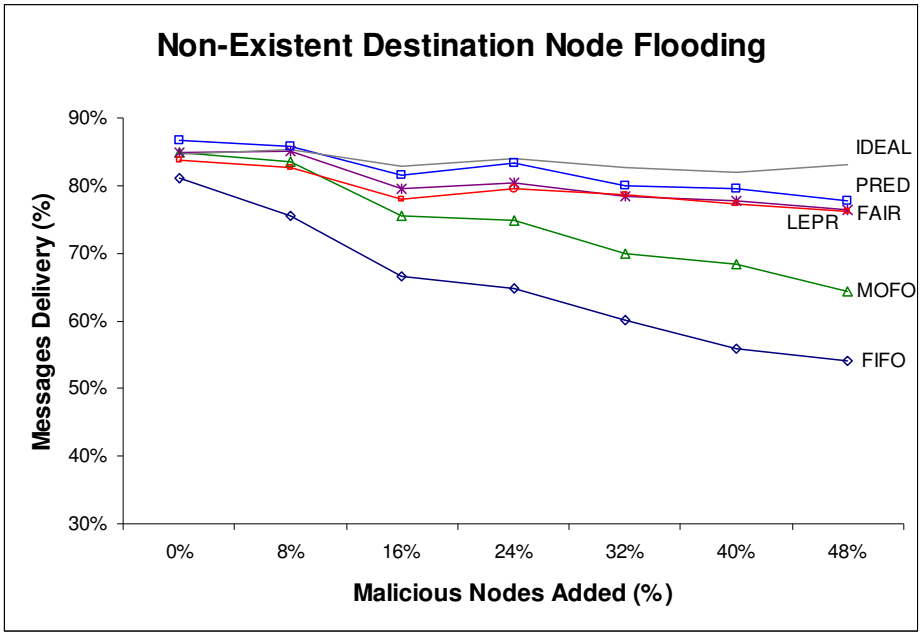


Figure 4.3. Non-Existent Destination Node Flooding

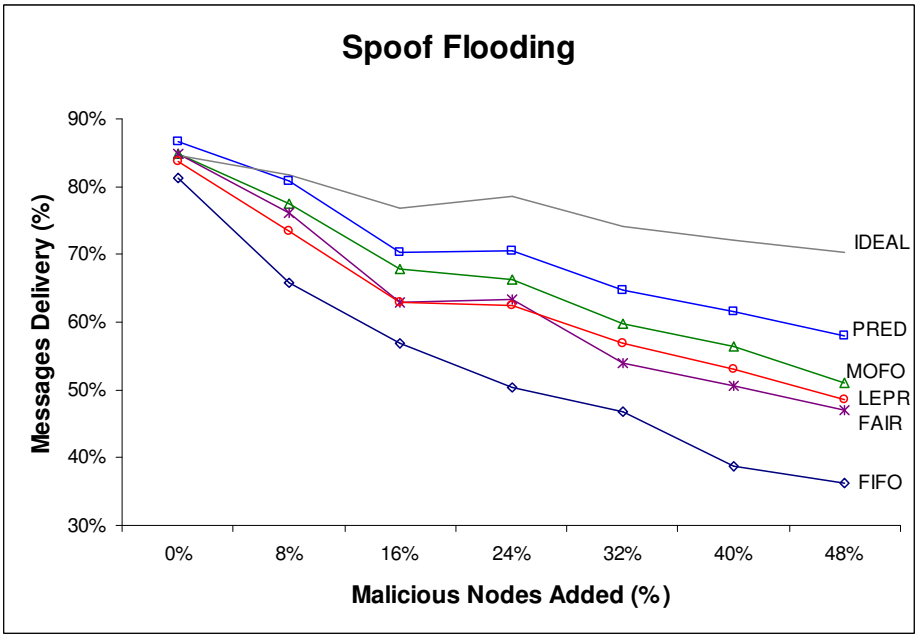


Figure 4.4. Spoof Flooding

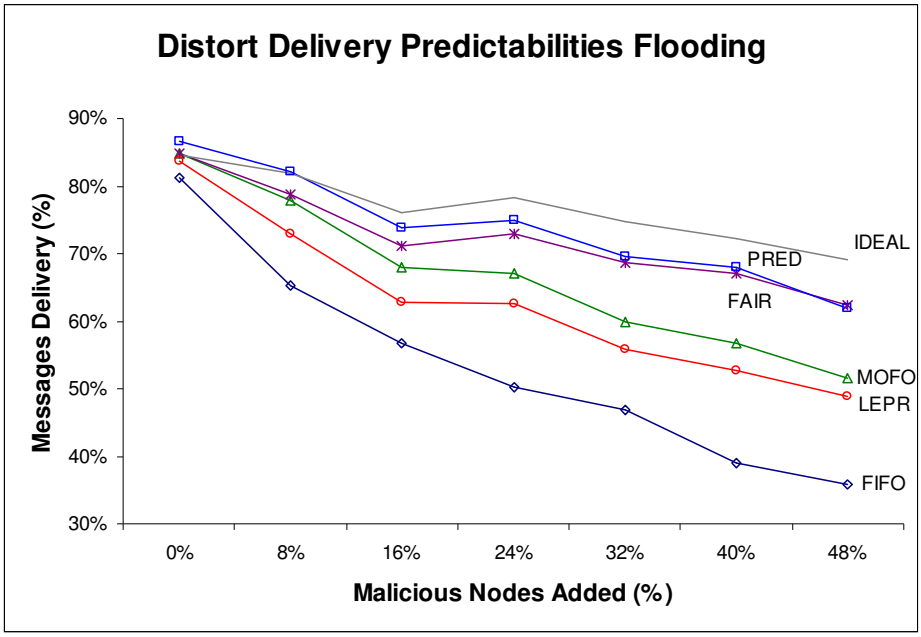


Figure 4.5. Distort Delivery Predictabilities Flooding

4.7. CONCLUSION

For probabilistic routing protocols, malicious node can utilize prior knowledge to perform more effective attacks on the network. Facing more organized security attacks, defensive techniques can follow a similar approach with nodes also utilizing prior knowledge and observed behavior to safeguard themselves against security threats. However, in typical security problems, the defensive party will not have it easy as the offensive party has the advantage of keeping the defensive nodes in suspense. Malicious nodes can distort and confuse the nodes in the network, concealing their intentions in the process. Spoof flooding and distort delivery predictabilities flooding are representatives of the forms of advanced flooding attack that can be anticipated.

From the observation of mobile networks' behavior, we glean an insight that a possible flooding attack is suspected when there are too many messages from a particular node that is seldom encountered. We relate our concept into devising a new queuing policy that capitalizes on the probabilistic feature of the PRoPHET protocol in

DTN to mitigate flooding attacks. In our simulation results, it is observed that our queuing policy has visibly better performance in withstanding the various flooding attacks as compared to the other queuing policies which are mostly used for general routing protocols. Finally, we believe our mechanism can be used to formulate the corresponding defense solutions to mitigate flooding attacks against probabilistic routing protocols other than the PRoPHET protocol.

Chapter 5.

Probabilistic Routing based on History of Messages in Delay Tolerant Networks

5.1. INTRODUCTION

Delay Tolerant Networks (DTN) [1] have been emerging and known to be suitable for deployments in challenged environments whereby connectivity is intermittent. Under harsh conditions, DTN is applicable to withstand disruptive connectivity through the use of persistent storage to buffer messages for long duration. As store-and-forward is a key feature in DTN, the overall delivery performance is very much depending on the choice of messages to keep when the buffer is full and the priority of messages for forwarding.

In DTN, routing protocols are developing with the more sophisticated ones using probabilistic metrics for the forwarding decision. Probabilistic metrics are delivery predictabilities of the node to other nodes, commonly derived from past encounters record. This metric enables a more intelligent forwarding decision to be made. For choosing a message to be dropped when the buffer is full, a queuing policy is normally used for the selection process. The queuing policy can be a sophisticated policy that involves probabilistic metrics comparison [29] or could be a simple traditional policy such as first-in-first-out. More sophisticated forwarding and queuing policies for probabilistic routing generally depend on the history of encounters [24].

For DTN, however, history of messages is as important as history of encounters since the nodes are choosing which messages to keep and which to forward. There are existing routing protocols that give priority to the messages that have lower hop count. Such approach maintains the freshness of the messages, ensuring new messages the opportunity of being propagated. However, using only the hop count metric to represent the history of a message is a little simplistic and there is potential for further refinement. After all, choosing a message to be dropped upon buffer full is about minimizing loss and choosing a message to be forwarded is about maximizing benefit.

There are a few existing routing protocols for DTN. However, with the exception of MaxProp in [23], the current routing protocols focus more on the forwarding strategy but not the queuing policy for buffer storage, which is not sufficient for DTN's store-and-forward purpose. Some traditional policies such as first-in-first-out do not factor in delivery likelihood and may not be the best for probabilistic routing. In [29], some forwarding strategies and buffer queue policies were evaluated for PRoPHET and Epidemic routing protocols. In their evaluation, probabilistic routing used with appropriate combination of forwarding strategy and buffer queue policy can produce good network performance. In [23], MaxProp uses a buffer management policy, prioritizing messages in its buffer storage based on hop count and delivery likelihood. However, a message's history is more complex than a mere measurement of its hop count as employed in some of these related works. As a result of an in-depth study, a more advanced message's history concept to be used for making more intelligent queuing and forwarding decisions is conceived and will be detailed in this chapter.

This chapter examines the routing protocol, Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [24], of DTN and introduces a solution that is suitable for DTN's store-and-forward nature. In this chapter, a new policy is proposed for PRoPHET to manage its forwarding and buffer storage, and the key feature in our proposal is a history of messages concept. We did a

preliminary study and analysis on PRoPHET and a few other routing protocols which will be discussed in Section 5.2. Subsequently, our proposed policy for PRoPHET is detailed in Section 5.3. Our simulation scenarios are described in Section 5.4 and our solution is evaluated in Section 5.5. Section 5.6 concludes this chapter.

5.2. PRELIMINARY STUDY

For preliminary study of routing protocols in DTN, we analyzed and performed simulations using the existing PRoPHET, MaxProp, and Spray-And-Wait [20] routing protocols. PRoPHET has a probabilistic feature using history of encounters and forwarding decision is made by comparing the delivery likelihood of the nodes. Spray-And-Wait routing, though not probabilistic based, is included in our study to find out why it fares well against routing protocols with probabilistic feature. MaxProp, which is probabilistic, is also included in our study given its additional consideration of messages' hop count value which could possibly give it an extra advantage over PRoPHET. Fig. 5.2 shows the preliminary results. The preliminary study done has led us to our finding on what is lacking in PRoPHET that could have made a significant improvement if included.

5.2.1. Spray-And-Wait Routing

Spray-And-Wait [20] routing does not require knowledge of a node's past encounters. It uses a mechanism to restrain the message from traversing further than necessary in the network. Every message is allocated a forwarding allowance which is the number of peers a node can transmit the message to. This forwarding allowance allocated for the message is reduced as the message accumulates more hops, thus preventing the message from being relayed over too many hops. For this reason, Spray-And-Wait routing achieves low delivery latency and an overall low number of message transmissions observed in the network. From our simulation records, the number of messages being delivered successfully in Spray-And-Wait

routing is higher than expected, taking into consideration the overall low number of transmissions observed.

5.2.2. MaxProp Routing

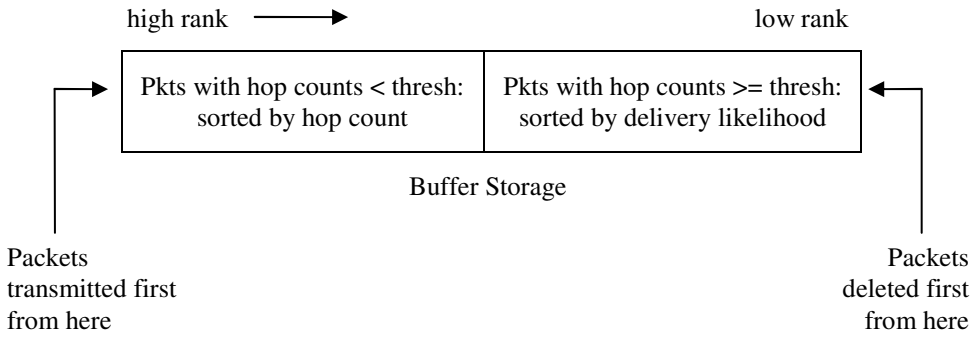


Figure 5.1. MaxProp routing strategy

MaxProp [23] routing factors in the delivery likelihood of the nodes as well as the hop count of the messages. The messages in the buffer are prioritized as shown in Fig. 5.1; messages with hop count below threshold are sorted by hop count giving highest priority to the message with the smallest hop count, and messages with hop count exceeding threshold are sorted by delivery likelihood giving lowest priority to the message that has lowest delivery likelihood. Forwarding decision and choice of message to drop upon buffer full will be based on the importance of the messages according to the sorted priority order in the buffer. The hop count factor in MaxProp ensures that new messages are not deprived of forwarding opportunity. As for the delivery likelihood factor, it ensures MaxProp is more selective in its forwarding of the messages that have already traversed far in the network. In our simulations, MaxProp routing has the highest number of messages being delivered successfully compared to the other two routing protocols discussed.

5.2.3. PRoPHET Routing

Observed in our simulations, PRoPHET [24] is not on par with MaxProp in terms of the number of messages being delivered successfully. Relative to Spray-And-Wait routing, PRoPHET does not show significant higher success rate of messages being delivered, considering that PRoPHET uses more message transmissions. With this observation, we did an analysis on PRoPHET and its various forwarding strategies [29], namely GRTR, GRTRMax, and GRTRSort.

GRTR – GRTR [29] is the most basic forwarding strategy of all; the message will be forwarded if the encountered peer has higher delivery predictability for the destination node. As GRTRMax is an improved version of GRTR, GRTR is omitted from this discussion.

GRTRMax – GRTRMax [29] is similar to GRTR, with the exception that GRTRMax prioritizes the forwarding sequence of the messages while GRTR has no particular priority. Messages are compared and sorted according to the encountered nodes' delivery predictabilities for the respective destinations of the messages, with messages that have higher likelihood of being delivered given higher priority.

GRTRSort – GRTRSort [29] prioritizes the message forwarding according to delivery predictability differences between the node itself and the encountered nodes. This is different from GRTRMax as GRTRMax only takes into consideration the encountered nodes' delivery predictability values for the messages. GRTRSort sorts the messages according to the improvement in delivery likelihood that the encountered node offers.

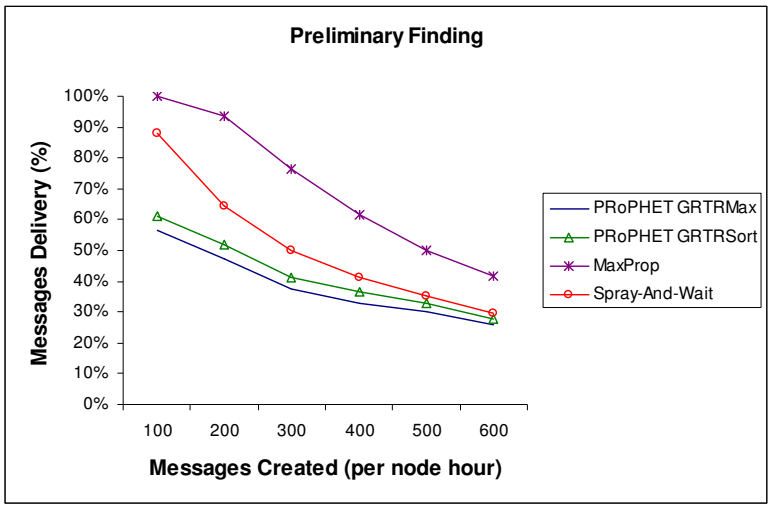


Figure 5.2. Preliminary finding

5.2.4. Analysis on the Routing Protocols

Our analysis is based on the routing scenario shown in Fig. 5.3. Node N_C has messages M_1 and M_2 to be delivered to their respective destination nodes. Node N_C has delivery predictability of 0.6 for message M_1 and 0.8 for message M_2. Node N_E has delivery predictability of 0.8 for message M_1 and 0.9 for message M_2. Node N_C encounters node N_E and has to decide whether to forward message M_1 or M_2 first. Before the encounter, both the messages were received from node N_A.

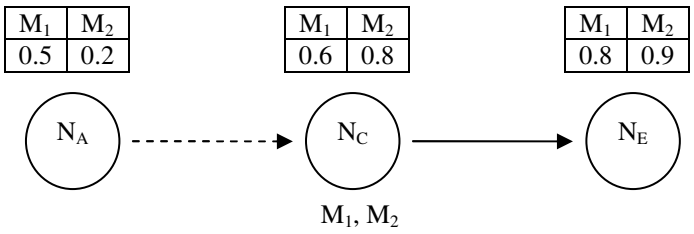


Figure 5.3. Routing scenario

PRoPHET using GRTRMax forwarding strategy will give message M_2 a higher priority than message M_1, as node N_E's probability of delivering message M_2 to its

destination is 0.9, which is higher than the probability of 0.8 for message M_1. However, the delivery predictability difference between node N_C itself and node N_E for the message M_2 is 0.1, which is less significant than the 0.2 difference for message M_1. Consider forwarding message M_2, the bandwidth usage for the little improvement in delivery predictability is actually quite wasteful. Node N_C's delivery predictability of 0.8 for message M_2 shows that the message has a high chance of reaching the destination even without being forwarded to node N_E. The bandwidth would have been better utilized if message M_1 is forwarded instead of message M_2. Hence, GRTRMax's preference for message M_2 over message M_1 in its forwarding priority is questionable.

PRoPHET using GRTRSort forwarding strategy would prefer message M_1 over message M_2 in its forwarding priority as it takes into account both node N_C's and node N_E's delivery predictabilities. Whereas GRTRMax only looks at N_E's delivery predictabilities, GRTRSort is more correct as it is interested in the improvement in delivery predictability that node N_E can offer. However, this forwarding strategy has neglected node N_A's delivery predictabilities for the messages at the previous hop. Looking at both node N_A's and node N_C's delivery predictabilities for the messages, message M_2 might have an overall lower probability of reaching the destination than message M_1. In this case, node N_E offers higher delivery probability improvement for message M_2 than message M_1. Hence, GRTRSort's preference for message M_1 over message M_2 in its forwarding priority is also questionable.

With the analysis of forwarding strategies in PRoPHET in mind, we further examine MaxProp and Spray-And-Wait and discover that the messages' previous hops can be a significant factor. In MaxProp, messages that have not traversed further than the threshold number of hops are given high priority. As for Spray-And-Wait routing, the further the message has traversed, the lower the forwarding allowance allocated for the message. The effective hop count controls in MaxProp and Spray-And-Wait suggest that the benefit of further forwarding a message might diminish at some point as it accumulates more hops. It is important for the

forwarding node to know how far the message has already traversed. However, the hop count value has no indication of the delivery predictabilities of the nodes at previous hops. Referring back to the scenario in Fig. 5.3, node N_A at the previous hop holds delivery predictabilities for messages M_1 and M_2 which can be beneficial to node N_C's forwarding decision. This brought to mind possible ways to model the history of messages.

5.3. PROPOSED SOLUTION

We propose a new policy History of Messages' Movement Events (HOMME) for PRoPHET. Our proposed idea recognizes every message has a history. The messages' history can be utilized to bring out the best in PRoPHET as our solution is well suited for DTN's store-and-forward purpose. The use of history of encounters to determine the delivery predictabilities of the nodes is fundamental and a sound basis for PRoPHET as a probabilistic routing protocol. However, using only the participating nodes' delivery predictabilities without due consideration of the messages' previous hops is inadequate for making a favorable forwarding decision. Notable in the analysis in Section 5.2.4, the knowledge of the messages' hop count being beneficial for MaxProp in its forwarding decision making, is in contrast with the possibly flawed forwarding decision made by the advanced GRTRSort forwarding strategy in PRoPHET as the messages' previous hops is not factored in. The delivery predictability of each node the message has traversed is part of the message's history, and can be exploited when making a probabilistic forwarding decision.

5.3.1. History of Messages Concept

In our history of message concept, every message will carry a probabilistic metric which indicates the likelihood of the message not reaching destination. The metric used is formulated based on the delivery predictabilities of the nodes the message has

traversed. This models the history of messages in a probabilistic way as compared to the hop count used by MaxProp. Forwarding a message that has not traversed far may not be more advantageous than forwarding a message that has not visited any node that has high delivery likelihood for the destination. With the history of the message coupled with the delivery predictabilities of the possible nodes at the next hop, the forwarding node is equipped with a more complete knowledge in its forwarding decision making process. Shown in Table 5.1, are the notations used in our modeling of history of messages.

Variable	**Definition**
$\mathbf{N_i}$	The i^{th} node.
$\mathbf{p_i}$	Delivery predictability of node N_i.
h	Hop count of the message.
$\mathbf{q_i}$	Probability of node N_i and nodes at previous hops are unable to deliver the message to the destination.
$\mathbf{b_i}$	Increase in probability of delivery to destination that node N_i offers.

Table 5.1 Notations

The history of a message's traversed path is defined as a sequence of nodes the message has visited $\{N_0, N_1, N_2, \ldots, N_K\}$. The benefit, b, of forwarding the message from node N_i to node N_{i+1} is equal to the decrease in q, the probability of the message's inability to reach the destination, as shown in equation (5.1). This is different from the GRTRSort forwarding strategy whereby the benefit is equal to the difference in delivery predictabilities, p, of nodes N_i and N_{i+1}. As the message's hop count increases, it accumulates benefits, and hence q decreases which signifies the message has less likelihood of not reaching the destination.

$$b_{i+1} = q_i - q_{i+1} \qquad (5.1)$$

With the messages' q values and the delivery predictabilities, p, of the peers, the node has to weigh the benefits, b, and prioritize the sequence of forwarding the messages to the respective chosen nodes. The computation of the benefit of forwarding is based on dice theory. Think about the dice scenario with the objective is to roll at least one '6' and the player is allowed to buy any number of dice he wants. How much can be benefited from buying an additional die to roll? The q value of a message is represented by the probability of not obtaining a '6' from rolling the number of dices the player considered buying. The delivery predictability, p, is represented by the 1/6 probability of obtaining a '6' if buying an additional die. The benefit of having an additional node to deliver the message is represented by the benefit of buying an additional die to roll, which can be derived from q and p values. With the dice scenario in mind, the benefit of forwarding the message from node N_i to node N_{i+1} is shown in equation (5.2). From equations (5.1) and (5.2), the new q value for the message is shown in equation (5.3).

$$b_{i+1} = p_{i+1} \times q_i \qquad (5.2)$$

$$q_{i+1} = q_i \times (1 - p_{i+1}) \qquad (5.3)$$

With the formulas in equations (5.2) and (5.3), our forwarding strategy can be analyzed based on an example scenario as shown in Table 5.2, whereby the message's traversed path is $\{N_0, N_1, N_2, \ldots, N_8\}$. The respective delivery predictabilities, p, of the visited nodes satisfy the common forwarding rule, $p_i < p_{i+1}$, in PRoPHET. Using our benefit system based on history of the message, the message will benefit from forwarding at first few hops but the benefit starts to diminish after node N_3. At node N_7, the benefit of forwarding the message to node N_8 is almost insignificant, since it is unlikely that the message will not reach the destination, having the probability, q, being just a mere 0.0036.

	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
h	0	1	2	3	4	5	6	7	8
q	0.9	0.72	0.504	0.3024	0.1512	0.0605	0.0181	0.0036	0.00036
1 - q	0.1	0.28	0.496	0.6976	0.8488	0.9395	0.9819	0.9964	0.99964
b	0.1	0.18	0.216	0.2016	0.1512	0.0907	0.0423	0.0145	0.00326

Table 5.2 A scenario analysis

Observing the benefit values shown in Table 5.2, it reveals signs of the features in MaxProp which give it an edge over other routing protocols. Referring back to Fig. 5.1 in Section 5.2.2, sorting in MaxProp gives higher priority to the messages with hop count below a threshold value and lower priority to the messages with hop count values that exceed. In the analysis for our scenario, the benefit of forwarding starts to diminish after the 3rd hop and this explains the threshold value required in MaxProp. The mapping of the benefit values for the respective nodes in Table 5.2 shows why MaxProp prefers to give messages with low hop count a head start before prioritizing using delivery likelihood. For additional note, the benefit of forwarding could start to diminish even before the 3rd hop, if the message first visited node N_8 which has delivery predictability of 0.9 before the 3rd hop.

5.3.2. Our Forwarding Strategy Specification

This section details the specification of our forwarding strategy. Each message will be given a q value and this metric is used instead of hop count. Our forwarding strategy uses a benefit system and comparison of benefits is based on the messages' q values and the peers' delivery predictabilities, p values.

- The sender node N_0 who created the message will initialize the q value of the message to be its delivery predictability p_0.

- When nodes encounter each other, the nodes will update their delivery predictabilities as specified in PRoPHET routing protocol.

- In our forwarding policy, messages will be forwarded sequentially. First, messages will be transferred to the destinations if encountered. Next, the rest of the messages will be transferred according to a priority sequence based on the benefit of forwarding as shown in Fig. 5.4. Each of these messages is assigned to a neighbor peer that has the highest delivery predictability to its destination. The computation for benefit of forwarding a message to the allocated neighbor peer is as follows:

$$b_i = p_i \times q$$

- When a message is being forwarded, the receiving node will update the q value in the message as follows.

$$q_{new} = q_{old} \times (1 - p)$$

- When the buffer of a node is full, our buffer queue policy will drop messages according to a priority sequence based on the messages' q values as shown in Fig. 5.5.

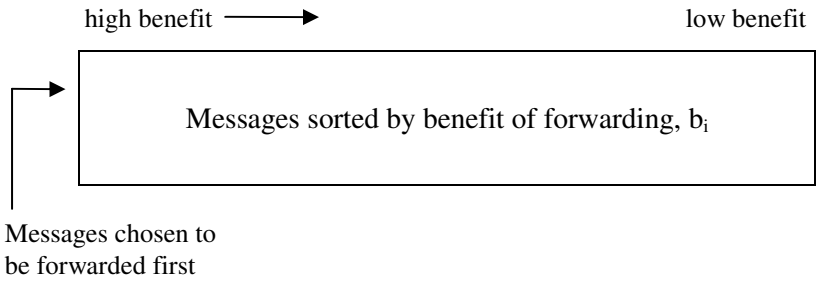


Figure 5.4. Priority for forwarding of messages

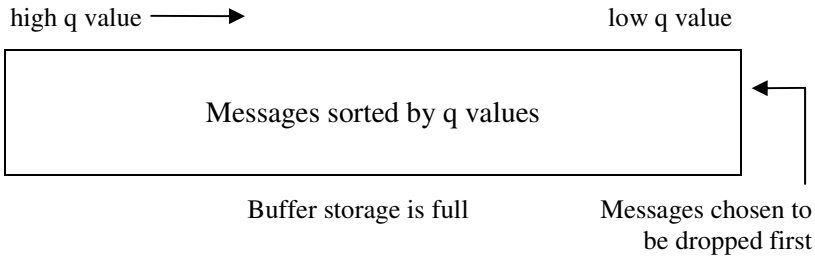
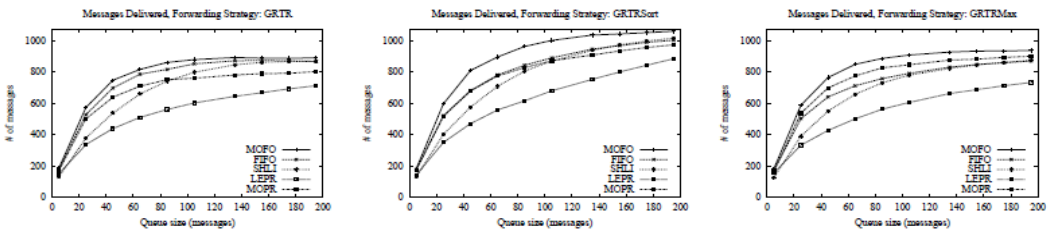


Figure 5.5. Priority for dropping of messages

5.4. SIMULATION

We ran simulations using Opportunistic Network Environment (ONE) simulator version 1.3.0 [47]. In our simulations, we compare our policy HOMME with the forwarding strategies and queuing policies that Lindgren et al. evaluated in [29] for the PRoPHET protocol. We chose the best and worst buffer queue policies in [29] (see Fig. 5.6), namely, MOFO and LEPR respectively and used them in various combinations with the forwarding strategies GRTRMax and GRTRSort as previously discussed in Section 5.2.3. We further benchmarked PRoPHET using HOMME against MaxProp and Spray-And-Wait routing protocols. This will show the relative performance of PRoPHET using HOMME against some advanced routing protocols in DTN. Our simulations consist of two different setups, homogeneous as well as heterogeneous scenarios. We measured the performance of the queuing policies based on the percentage of messages delivered successfully.



(a) GRTR (b) GRTRSort (c) GRTRMax

Figure 5.6. Various existing queuing policies and forwarding strategies for PRoPHET

MOFO - MOFO [29] is a queuing policy which removes the most forwarded message first. Each node kept a count for the number of times each individual message has been forwarded. The message that was forwarded the most times is the message to be removed first.

LEPR - LEPR [29] is a queuing policy which removes the message with the lowest delivery predictability. A message that has the lowest delivery predictability has the smallest chance of being delivered to the destination, and hence it is the message selected to be removed first. In our simulation, we included an additional condition to ensure that the messages being removed must have been forwarded at least once. In this way, newly created messages that have low delivery predictabilities to their respective destinations will have a chance to be forwarded.

5.4.1. Homogeneous Scenario

The homogeneous scenario is a simulated network of 50 rovers. The rovers' mobility speeds were within 20 - 50 km/h, transmission range is 100 m, and they were installed with computing devices with 128 MB storage space. In each simulation, 5000 messages of message size 1MB – 6 MB were created by the nodes within a simulated time of an hour. In the simulations, the first hour is used for initialization of the nodes' delivery predictabilities, the creation of the messages were done in the second hour, and the delivery of the messages were done in the second and third hour.

5.4.2. Heterogeneous Scenario

In the heterogeneous scenario, the number of rovers is reduced to 15 and extra 30 pedestrians and 5 trams are included. The rovers' mobility speeds were within 20 - 50 km/h, transmission range is 120 m, and they were installed with computing devices with 512 MB storage space. The pedestrians' mobility speeds were within 1.8 - 5.4 km/h, transmission range is 100 m, and they were equipped with computing devices that possess 64 MB storage space. The trams' mobility speeds were within

25 - 36 km/h, transmission range is 150 m, and they were installed with computing devices with 1024 MB storage space. The rest of the setup is the same as that in the homogeneous scenario. As this scenario has different mobility speeds and capabilities for different categories of peers, it is closer to real deployments than the homogeneous network scenario.

5.5. EVALUATION OF RESULTS

For our evaluation of DTN's routing protocols, we value message delivery as the most important measurement of the network performance. Overcoming intermittent connectivity that causes difficulty for the messages to reach the destination is our key focus. We value delivery latency as the least important as DTN is meant to be delay tolerant. To compare the performance under the various possible deployments for DTN, we evaluate the network performance in a homogeneous setup and also in a heterogeneous setup.

In homogeneous scenario, the evaluation results in Fig. 5.7 show PRoPHET using our HOMME policy has the highest percentage of messages delivered as compared to PRoPHET using other forwarding strategies and buffer queue policies combinations. Our policy values the limited forwarding opportunities and buffer storage capacity. HOMME understands forwarding decision should not be just based on the two encountering peers, as the messages' history of traversed nodes gives a more complete knowledge for making forwarding decisions that are beneficial. Comparing PRoPHET using HOMME, it has more messages delivered than Spray-And-Wait routing protocol but not as much as MaxProp. This is due to the single type of peers simulated for the homogeneous scenario whereby the same mobility speeds and capabilities among the peers would result in not much difference in the delivery predictability values used in PRoPHET. The results could be different in a more realistic heterogeneous setup whereby peers that move faster will gain more encounters and show higher delivery predictability than slower peers. It is also noted

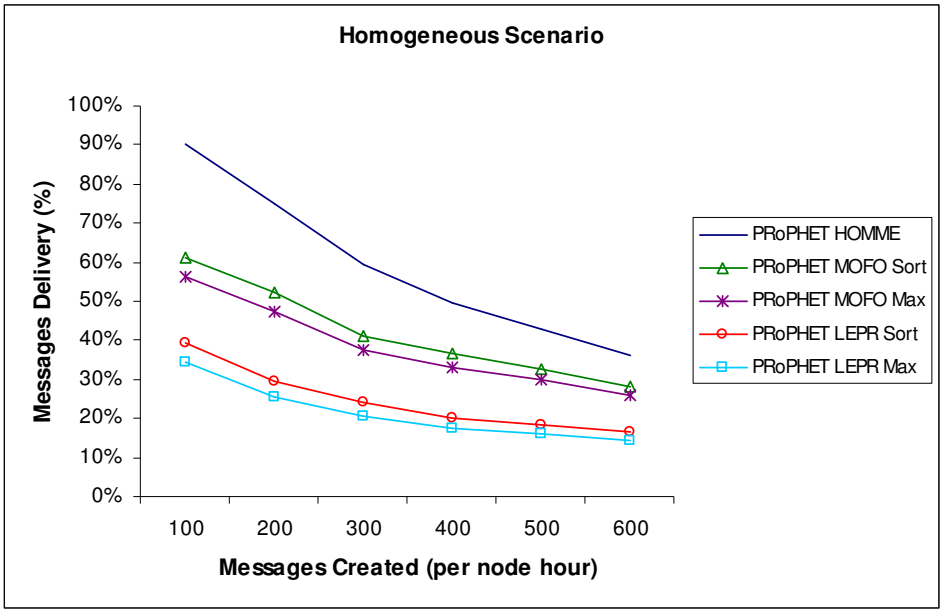
that PRoPHET using MOFO with GRTRSort combination, the second best after HOMME, has lower messages recorded as compared to Spray-And-Wait and MaxProp routing protocols. Our HOMME policy for PRoPHET shows much more improvement over the existing policies used in PRoPHET.

In heterogeneous scenario, the evaluation results in Fig. 5.8 show PRoPHET using HOMME has the highest percentage of messages delivered as compared to PRoPHET using other policies. In contrast to the previous homogeneous setup, the results for heterogeneous setup showed PRoPHET using HOMME also has a slight edge over MaxProp and Spray-And-Wait routing protocols in terms of message delivery. This suggests the delivery predictability used by PRoPHET has a more significant impact for network scenario that has more than one type of peers with the different types having different mobility speeds and capabilities. On a side note, PRoPHET using MOFO and GRTRSort combination has as high percentage of messages delivered as MaxProp and Spray-And-Wait routing protocols.

For the evaluation of delivery latency, Fig. 5.9 shows HOMME has longer latency as compared to the other policies used in PRoPHET. The reason is HOMME has a higher resistance of forwarding the messages after a few hops as the effect of diminishing benefits registers lower benefits at the later hops. As a result, it delays the delivery of messages to their respective destinations. The latency difference between HOMME and other policies is quite significant, and we believe our future work can improve HOMME in this aspect by giving slightly more forwarding allowance at the later hops. It is an issue of a trade-off between extra overhead incurred and delivery latency. Comparing PRoPHET using HOMME with MaxProp routing protocol, the delivery latencies measured are relatively on par. Spray-And-Wait routing has the lowest delivery latency measured, though its forwarding policy also has a high resistance of forwarding the messages after a few hops in a way similar to HOMME. However, Spray-And-Wait restricts the forwarding of a message after the message has used up its allocated number of forwards; nodes are not allowed to forward the message further with the exception of forwarding to the

destination node when encountered. This policy is different from HOMME, as low latency is achieved in Spray-And-Wait by 'predefining' the hop count. In Spray-And-Wait, however, achieving delivery is unlikely if the message does not reach a node that has a high likelihood of encountering the destination node after the allocated number of forwards are all used up. This is the reason Spray-And-Wait records lower message delivery percentage as shown in Fig. 5.7 and Fig. 5.8.

In our evaluation of overhead, we measure the overhead as the extra average number of message relays incurred per message delivery. We first calculate the difference between the total number of messages relayed and the total number of messages delivered, and then divide this difference by the total number of messages delivered. As shown in Fig. 5.10, the overhead incurred in PRoPHET using HOMME is satisfactory as compared to PRoPHET using other policies. Although HOMME incurred slightly higher overhead than MOFO buffer queue policy, this is a trade-off with the higher message delivery performance of HOMME. Comparing with MaxProp routing protocol, PRoPHET using HOMME incurred less overhead. On the other hand, the low overhead incurred in Spray-And-Wait is quite expected, as Spray-And-Wait predefined the number of forwards for each message, and hence the overhead incurred does not exceed a certain number. However, as the number of created messages increases, the overhead incurred in PRoPHET using HOMME converges to a value close to Spray-And-Wait's.



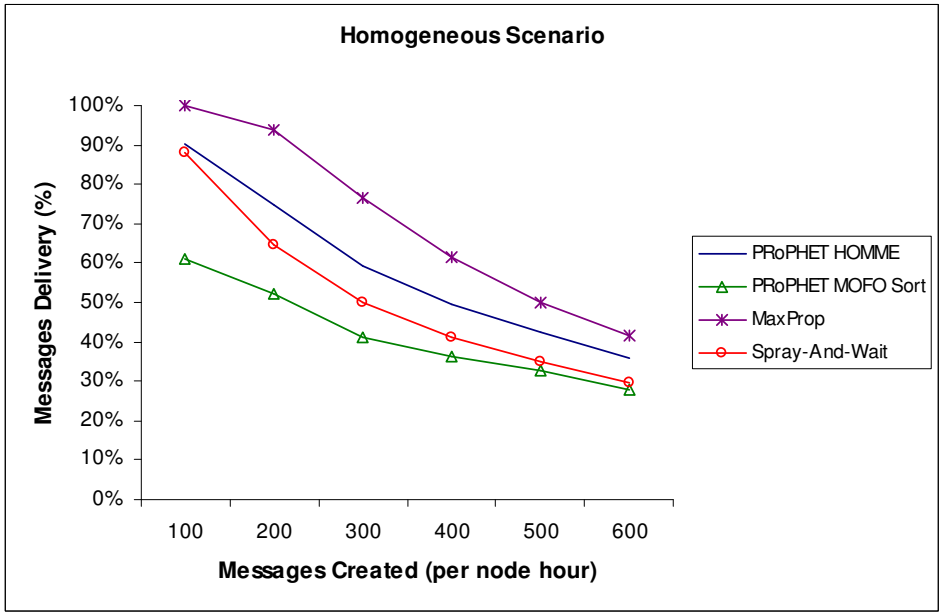
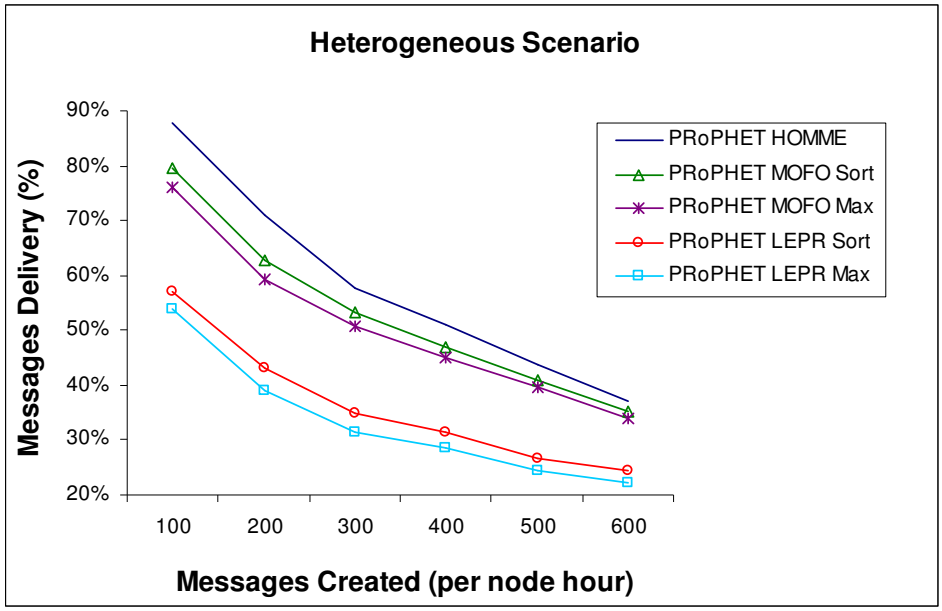


Figure 5.7. Messages delivery in homogeneous scenario



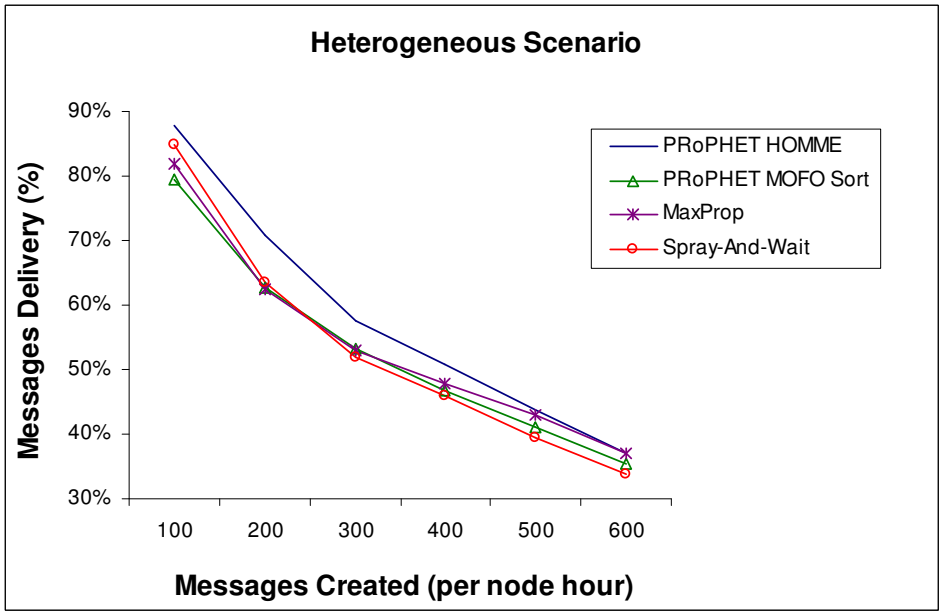
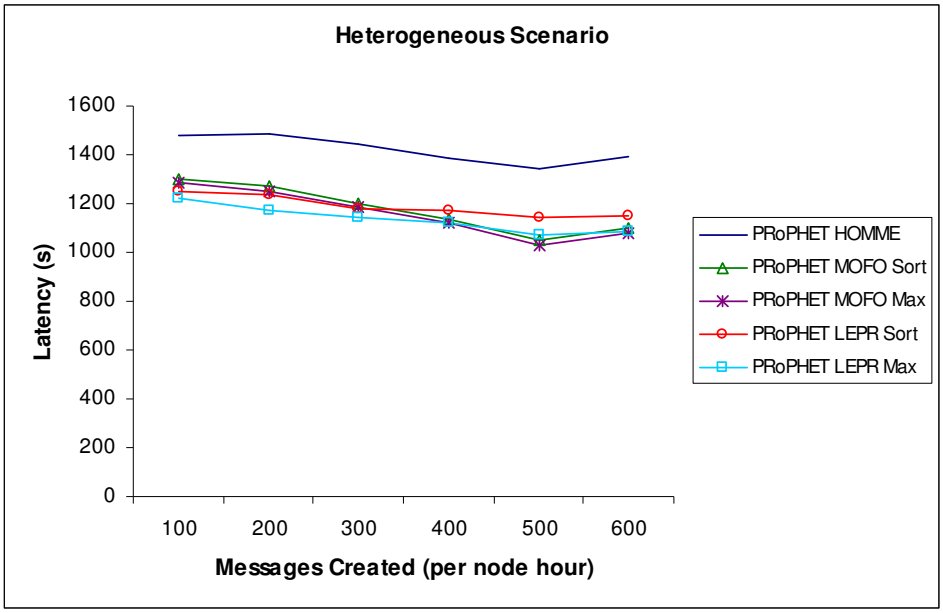


Figure 5.8. Messages delivery in heterogeneous scenario



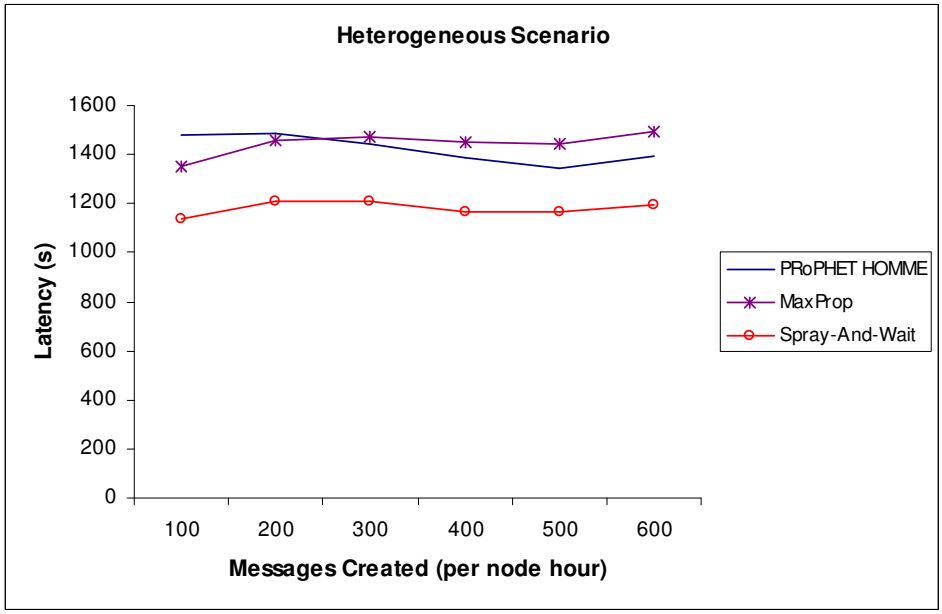
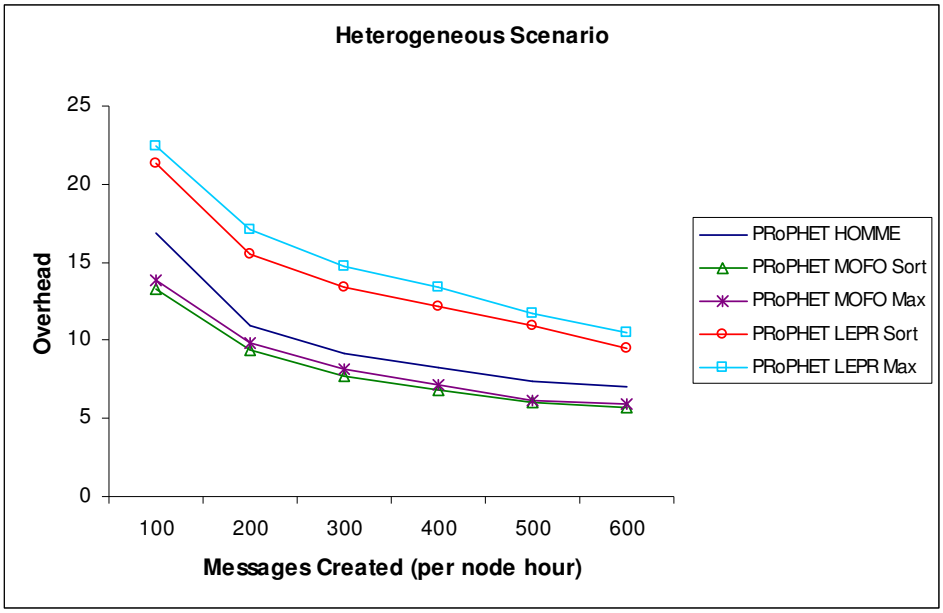


Figure 5.9. Latency in heterogeneous scenario



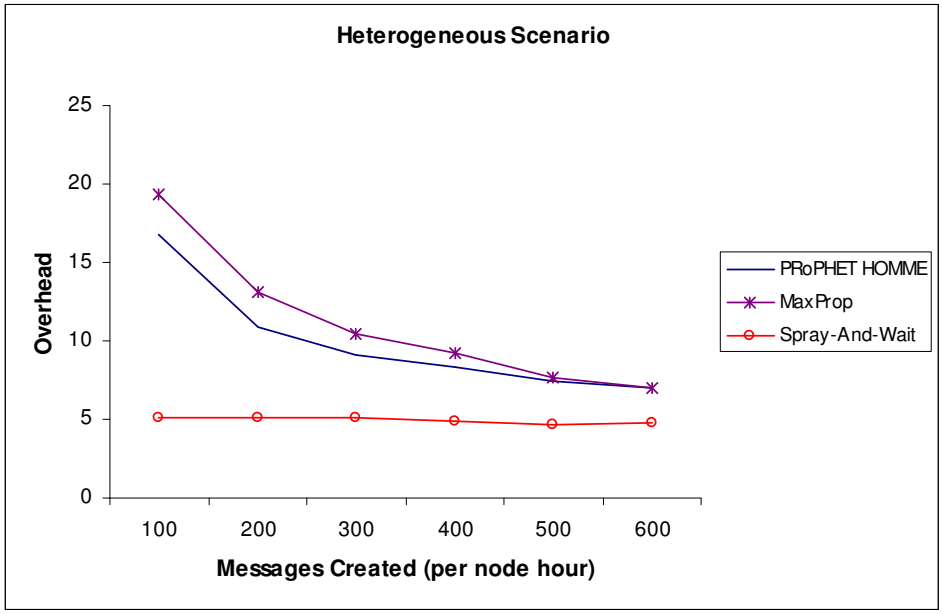


Figure 5.10. Overhead in heterogeneous scenario

5.6. CONCLUSION

This chapter has put forth the concept of history of messages for DTN routing. Having knowledge of messages' history and its use is suited to routing in DTN's store-and-forward nature. Instead of choosing from the nodes purely based on their delivery likelihood, HOMME policy solves routing problem from the messages' history, whether it is beneficial for the messages to be dropped or be forwarded. For probabilistic routing protocols, HOMME is a more sophisticated approach that can replace the hop count factor as it exploits the delivery predictabilities of the nodes to give a more complete account of the history of the messages. Our analysis of the benefit system in HOMME explains the reason behind the effectiveness of MaxProp giving high priority to the messages that has not traversed far in the network. In our evaluation, the results has shown that the sophistication of HOMME churn out a high messages delivery percentage for PRoPHET routing to an extent that it is now comparable to MaxProp's performance. For future research, we believed the delivery predictability metric in probabilistic routing protocols can be further exploited for other purposes.

Chapter 6.

Conclusion and Future Works

With the 'Plug-and-Play' framework mentioned in Chapter 3, operating a heterogeneous DTN has become more versatile. It offers a workaround option in times of intermittent connectivity by using the availability of secondary network access technologies to stay connected. If a node does not support the required network access technology to get connected, the framework can still allow the node to be cascaded to another node that has the network access technology. With our new implementation of the Ethernet convergence layer, WiFi communication in DTN has the option of bypassing the redundant mechanisms of the TCP convergence layer. Our Ethernet convergence layer is more lightweight as it only keeps the basic functions of data transmission between a pair of nodes and omits the unnecessary reliability mechanism for transmission over a connected path of nodes.

Beyond network access technologies support, Chapter 4 highlighted a security concern pertaining to the advanced flooding attacks that can be performed when a probabilistic based routing protocol is used. The additional knowledge of the delivery likelihood of the nodes allows the malicious nodes to better organize their flooding attacks. Spoofing and selectively targeting the highly active nodes are some possible ways the malicious nodes can conduct more effective flooding attacks. PRoPHET is one such routing protocol that is prone to the new flooding attacks as it involves exchanges of delivery likelihood information. However, with our new queuing policy formulated for PRoPHET, the adverse effect of flooding attack can be kept under control. In our evaluation of the new queuing policy, it was proven to be effective in mitigating the more advanced flooding attacks as well as the simplistic random flooding attack.

Further analyzing the PRoPHET, Chapter 5 discussed in details its existing forwarding strategies and buffer queuing policies. In the analysis, it was discovered that the policies neglect the delivery likelihood of the messages' previously traversed nodes. With our proposed history of messages concept, the new forwarding strategy and buffer queuing policy could use a benefit system to prioritize the messages. As a result, PRoPHET could maximize the benefit of forwarding messages when encountering nodes and minimize the loss of dropping messages upon having a full buffer. In our evaluation, PRoPHET using our policy of forwarding messages and managing buffer storage has been proven to be better than the others used by PRoPHET. Comparing with Spray-And-Wait and MaxProp, PRoPHET using the proposed policy performed better in a realistic heterogeneous scenario where the nodes have various mobility speeds, buffer storage space and transmission range.

For future works in DTN, we have identified the following issues to be solved.

6.1. SECURITY PROTOCOL TO COUNTER MASQUERADE ATTACK

Security and authentication issues are never addressed in the current DTN architecture. One security issue that is of concern is masquerade attack. DTN uses URI naming scheme for identification of peers. Every individual link contains the URI of the peer and its logical address. The problem arises when a received announce beacon indicates that a peer has changed its logical address. There are two possibilities to consider. The peer might really have changed its logical address, or the announce beacon might have been sent by an adversary node with the same URI as the peer. In this case, it is difficult to decide whether to reconfigure the link and update the logical address. To handle spoofing of identity security issue, we suggest the implementation of a new security protocol in Delay Tolerant Networks (DTN) under a Heterogeneous Mobile Ad Hoc Network Environment to protect against masquerade attacks.

6.2. PEER DISCOVERY FOR MULTI-HOP

Peers could not be discovered on UHF Radio interface as the modem uses multi-hop communication. The discovery protocol in DTN can only support single hop peer discovery as the broadcast is unable to reach the peers via multi-hop. Hence only static network link is supported for network access technologies that use multi-hop communication. This limitation was encountered in our heterogeneous DTN testbed and we had to set static network link as a workaround for communication using UHF Radio interface. To support dynamic network link for these interfaces, we suggest an extension of the discovery protocol to support multi-hop peer discovery.

6.3. ROUTING PROTOCOL FOR HETEROGENEOUS DTN

Having studied many routing protocols and network heterogeneity in DTN, the various mobility speeds, buffer storage space and transmission range of the nodes makes routing in a heterogeneous DTN a complex problem. The delivery latency is affected by both the transmission speed of the network access technologies and the mobility speeds of the nodes. The routing problem is further complicated by nodes having multiple network access interfaces. For a versatile and scalable heterogeneous DTN, we suggest the development of a new routing protocol that can accommodate heterogeneity well in DTN.

Author's Publications

1. F. C. Lee, Y. Xia, C. K. Yeo, and Y. T. Tan, "A 'plug-and-play' framework to enhance heterogeneity and versatility in Delay Tolerant Networks," The 6th Advanced International Conference on Telecommunications (AICT), Barcelona, Spain, May 2010.

2. F. C. Lee, W. Goh, and C. K. Yeo, "A queuing mechanism to alleviate flooding attacks in probabilistic Delay Tolerant Networks," The 6th Advanced International Conference on Telecommunications (AICT), Barcelona, Spain, May 2010.

3. F. C. Lee and C. K. Yeo, "Ethernet convergence layer for Delay Tolerant Networks," submitted to Journal of Networks and Computer Applications, August 2010.

4. F. C. Lee and C. K. Yeo, "Probabilistic routing based on history of messages in Delay Tolerant Networks," submitted to IEEE Vehicular Technology Conference (VTC) Fall, San Francisco, California, USA, September 2011.

Bibliography

[1] K. Fall, “A Delay-Tolerant Network architecture for challenged Internets,” ACM Special Interest Group on Data Communication Conference (SIGCOMM), Karlsruhe, Germany, August 2003.

[2] Wizzy project, http://www.wizzy.org.za/.

[3] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: Rethinking connectivity in developing nations,” IEEE Computer, vol. 37, no. 1, January 2004, pp. 78-83.

[4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, “Energy-effcient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” The 10th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), San Jose, California, USA, October 2002.

[5] T. Small and Z. Haas, “The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way),” The 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, Maryland, USA, June 2003.

[6] Interplanetary internet special interest group (IPNSIG), http://www.ipnsig.org/.

[7] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, “Delay-tolerant networking: an approach to interplanetary internet,” IEEE Communications Magazine, vol. 41, no. 6, June 2003, pp 128–136.

[8] R. Krishnan, P. Basu, J. M. Mikkelson, C. Small, R. Ramanathan, D. W. Brown, J. R. Burgess, A. L. Caro, M. Condell, N. C. Goffee, R. R. Hain, R. E. Hansen, C. E. Jones, V. Kawadia, D. P. Mankins, B. I. Schwartz, W. T. Strayer, J. W. Ward, D. P. Wiggins, and S. H. Polit, "The SPINDLE

Disruption-Tolerant Networking system," IEEE Military Communications Conference (MILCOM), Orlando, Florida, USA, October 2007.

[9] R. A. Nichols and A. R. Hammons, "Performance of DTN-based free-space optical networks with mobility," IEEE Military Communications Conference (MILCOM), Orlando, Florida, USA, October 2007.

[10] Sensor networking with delay tolerance (sendt), http://down.dsg.cs.tcd.ie/sendt/.

[11] M. Ho and K. Fall, "Poster: Delay Tolerant Networking for Sensor Networks," The 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California, USA, October 2004.

[12] KioskNet, http://blizzard.cs.uwaterloo.ca/tetherless/index.php/KioskNet.

[13] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," The 12th ACM Annual International Conference on Mobile Computing and Networking (MobiCom), Los Angeles, California, USA, September 2006.

[14] S. Guo, M. H. Falaki, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, and S. Keshav, "Very low-internet access using KioskNet," ACM Special Interest Group on Data Communication Conference (SIGCOMM), Computer Communication Review, vol. 37, no. 5, October 2007.

[15] Delay Tolerant Network research group (DTNRG), http://www.dtnrg.org/.

[16] "Draft IEEE standard for Local and Metropolitan Area Networks: Media Independent Handover services (Draft 05)," IEEE 802.21, http://www.ieee802.org/21/.

[17] K. Harras, M. Wittie, K. Almeroth, and E. Belding, "ParaNets: A parallel network architecture for challenged networks," IEEE Workshop on Mobile

Computing Systems and Applications (HotMobile), Tucson, Arizona, USA, February 2007.

[18] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," ACM Special Interest Group on Data Communication Conference (SIGCOMM), Portland, Oregon, USA, 30 August – 3 September 2004.

[19] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, April 2000.

[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," ACM Special Interest Group on Data Communication Workshop on Delay-tolerant networking (SIGCOMM-WDTN), Philadelphia, Pennsylvania, USA, August 2005.

[21] M. Demmer and K. Fall, "DTLSR: Delay tolerant routing for developing regions," ACM Special Interest Group on Data Communication Workshop on Networked Systems in Developing Regions (SIGCOMM-NSDR), Kyoto, Japan, August 2007.

[22] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," ACM Special Interest Group on Data Communication Conference (SIGCOMM), Kyoto, Japan, August 2007.

[23] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based Disruption-Tolerant Networks," The 25th IEEE International Conference on Computer Communications (INFOCOM), Barcelona, Spain, April 2006.

[24] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," The 1st International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR), Fortaleza, Brazil, August 2004.

[25] A. Doria, M. Uden, and D. P. Pandey, “Providing connectivity to the Saami nomadic community,” The 2nd International Conference on Open Collaborative Design for Sustainable Innovation, Bangalore, India, December 2002.

[26] UMass DieselNet, http://prisms.cs.umass.edu/dome/umassdieselnet.

[27] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic routing protocol for intermittently connected networks," draft-irtf-dtnrg-prophet-07, August 2010, http://tools.ietf.org/html/draft-irtf-dtnrg-prophet-07.

[28] R. Perlman, "Routing with byzantine robustness," Technical Report SMLI TR-2005-146, Sun Microsystems, September 2005.

[29] A. Lindgren and K. S. Phanse, “Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks,” The 1st IEEE International Conference on Communication System Software and Middleware (Comsware), New Delhi, India, January 2006.

[30] A. Krifa, C. Barakat, and T. Spyropoulos, “Optimal buffer management policies for Delay Tolerant Networks,” The 5th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Francisco, California, USA, June 2008.

[31] T. Small and Z. Haas, “Resource and performance tradeoffs in Delay-Tolerant wireless networks,” ACM Special Interest Group on Data Communication Workshop on Delay-tolerant networking (SIGCOMM-WDTN), Philadelphia, Pennsylvania, USA, August 2005.

[32] M. Chuah, L. Cheng, and B. Davison, “Enhanced Disruption and Fault Tolerant Network architecture for bundle delivery (EDIFY),” IEEE Global Telecommunications Conference (Globecom), St Louis, Missouri, USA, November 2005.

[33] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in Delay Tolerant Networks comprising heterogeneous node populations," IEEE Transactions on Mobile Computing, vol. 8, no. 8, August 2009, pp. 1132-1147.

[34] M. Garetto, P. Giaccone, and E. Leonardi, "Capacity scaling in Delay Tolerant Networks with heterogeneous mobile nodes," The 8th IEEE International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Montréal, Québec, Canada, September 2007.

[35] H. Samuel, W. Zhuang, and B. Preiss, "Routing over interconnected heterogeneous wireless networks with intermittent connections," IEEE International Conference on Communications (ICC), Beijing, China, May 2008.

[36] K. Fall and S. Farrell, "DTN: An architectural retrospective," IEEE Journal on Selected Areas in Communications, vol.26, no. 5, June 2008, pp. 828-836.

[37] J. Wilson, "Probabilistic routing in Delay Tolerant Networks," Technical Report, Baylor University, 16 November 2007.

[38] T. Hyyrylinen, T. Krkkinen, C. Luo, V. Jaspertas, J. Karvo, and J. Ott, "Opportunistic email distribution and access in challenged heterogeneous environments," ACM Mobile Computing and Networking Workshop on Challenged Networks (MobiCom-CHANTS), Montréal, Québec, Canada, September 2007.

[39] Delay Tolerant Network (DTN) Reference Implementation Documentation, http://www.dtnrg.org/docs/code/DTN2/doc/doxygen/html/index.html.

[40] Raw Ethernet Socket Programming, http://aschauf.landshut.org/fh/linux/udp_vs_raw/ch01s03.html.

[41] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: An efficient routing in Disruption-Tolerant Networks," The 10th ACM International Symposium on

Mobile Ad Hoc Networking and Computing (MobiHoc), New Orleans, Louisiana, USA, May 2009.

[42] I. Cardei, J. Wu, C. Liu, and Q. Yuan, "DTN routing with probabilistic trajectory prediction," The 3rd ACM International Conference on Wireless Algorithms, Systems, and Applications (WASA), Dallas, Texas, USA, October 2008.

[43] M. Chuah and W. Ma, "Integrated buffer and route management in a DTN with message ferry," IEEE Military Communications Conference (MILCOM), Washington, District of Columbia, USA, October 2006.

[44] J. Burgess, G. Bissias, M. D. Corner, and B. N. Levine, "Surviving attacks on Disruption-Tolerant Networks without authentication," The 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Montréal, Québec, Canada, September 2007.

[45] X. Luo, R. K. C. Chang, and E. W. W. Chan, "Performance analysis of TCP/AQM under Denial-of-Service attacks," The 13th IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Georgia, Atlanta, USA, September 2005.

[46] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in Mobile Ad Hoc Networks," Springer - Signals and Communication Technology, Wireless Network Security, 2007, pp. 103-135.

[47] Opportunistic Network Environment (ONE) simulator version 1.3.0, http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[48] F. C. Lee, Y. Xia, C. K. Yeo, and Y. T. Tan, "A 'plug-and-play' framework to enhance heterogeneity and versatility in Delay Tolerant Networks," The 6th Advanced International Conference on Telecommunications (AICT), Barcelona, Spain, May 2010.

[49] F. C. Lee, W. Goh, and C. K. Yeo, "A queuing mechanism to alleviate flooding attacks in probabilistic Delay Tolerant Networks," The 6th Advanced International Conference on Telecommunications (AICT), Barcelona, Spain, May 2010.

[50] F. C. Lee and C. K. Yeo, "Ethernet convergence layer for Delay Tolerant Networks," submitted to Journal of Networks and Computer Applications, August 2010.

[51] F. C. Lee and C. K. Yeo, "Probabilistic routing based on history of messages in Delay Tolerant Networks," submitted to IEEE Vehicular Technology Conference (VTC) Fall, San Francisco, California, USA, September 2011.