EVALUATION OF FERRY BASED OPPORTUNISTIC MESSAGE DISSEMINATION WITH DETERMINISTIC INTER CONTACT TIMES

by

SAMREEN TAHIR

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2010

ACKNOWLEDGEMENTS

I want to start off by thanking from the core of my heart, the Almighty Allah, who was the biggest source of strength and hope and to whom I am grateful for the innumerable blessings He bestowed and the difficulties He removed during the course of my studies.

I would also like to take this opportunity to thank my supervising professor and my mentor, Dr. Mohan Kumar, without continuous support and patience of whom, this Master's program would not have resulted in completion. There can be only a few students who can claim to have received the kind of sustained grooming that I did for the duration of my research under Dr. Kumar. I wish to express my gratitude to my committee members, Dr. Yonghe Liu and Dr. Matthew Wright, for going out of their way in accommodating me and pointing me in the right direction of approach towards tackling conceptual problems. Their lucid explanations and counter examples helped me figure out the grey areas and inspired me to strive for the better.

I am indebted to Dr. Bahram Khalili, CSE graduate advisor, whose encouragement, guidance and support from the initial to the final level enabled me to keep on the path of progress and whose readily available advice was imperative for continuance of my studies.

My friends, whom I cannot thank enough, were always there to cheer me on and take out the time to do the little things that matter the most. Specifically, I would like to acknowledge the helpful insights and lengthy discussions provided by Umair Sadiq that helped clear the picture and put things in a perspective. I owe many thanks to Aisha Tahsin for being a supportive and a caring friend.

Lastly, I want to thank my parents and my sisters, who went above and beyond what was due and for whom mere words cannot convey my gratitude for the love and friendship that has always been my privilege.

ABSTRACT

EVALUATION OF FERRY BASED OPPORTUNISTIC MESSAGE DISSEMINATION WITH DETERMINISTIC INTER CONTACT TIMES

Samreen Tahir, M.S.

The University of Texas at Arlington, 2010

Supervising Professor: Mohan Kumar

Opportunistic Networks are characterized by intermittent connectivity and volatile topology. In such a network scenario, traditional Mobile Ad hoc Network (MANET) routing protocols cannot work efficiently. Hence, a novel way of message forwarding is required that operates even when an end-to-end path between a source and destination pair may never exist. Delay Tolerant Networks (DTNs) are an approach to alleviate network partitioning by working on the ‘store-carry-forward’ paradigm [4]. Opportunistic Networks encompass DTNs and typically involve mobile devices that exchange data by taking advantage of their proximity to other nodes. Nodes in opportunistic networks may be heterogeneous with respect to their movement capabilities characterized by versatility of visits to different locations as well as speed of mobility. Special nodes called data mules have been used to ferry the data from restricted or less capable nodes to the message destination. In this thesis, we explore the implications of ferry based message dissemination in terms of *average latency* and *delivery ratio* performance metrics. We analyze two important metrics of measuring network performance, namely, *inter contact times* and *contact frequency* and their impact on performance metrics. We derive an expression for *contact frequency* per hour and show that a deterministic cyclic ferry network has a constant per hour capacity. Through simulations we show how an overlay network of agile ferries can be used to improve network performance and discuss the impact of such movement and routing heterogeneity on the network. Using the analogy of a deterministic cyclic city bus network, we show empirically how increase in number of *travelers* or regular nodes effects the network performance under routing based on deterministic ferry schedules.

TABLE OF CONTENTS

APPENDIX

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ALGORITHMS

Algorithm Page

CHAPTER 1

INTRODUCTION

1.1 Opportunistic Networks

Mobile Ad hoc Networks (MANETs) have gained so much popularity in the last one and half decades that concepts such as ubiquitous computing, pervasive computing and mobile computing have spawned separate research of their own. The world, however, is changing in the way it uses computers to solve various problems and accomplish different tasks. The focus has shifted from the mechanics of computational technology involved to the accomplishment of the task itself. Examples of computer users include health care professionals, workers at a construction site, military personnel in the battlefield, scientists monitoring a volcano, and astronauts aboard a space shuttle communicating with base stations on earth. All of these users have one thing in common; they want to be able to communicate despite occasional or even perpetual network disruption. Opportunistic Networks work on the 'store-carry-forward' paradigm [4]. In this paradigm, a network node that senses insufficient bandwidth, stores the message in a buffer until such time as it senses more favorable channel capacity.

1.1.1 Opportunistic Networks versus MANETs

MANETs are characterized by volatile link capacity and network topology. One of the most popular evolutions of MANETs [4], Opportunistic Networks (ONs) use opportunities of proximity based contact amongst its nodes to forward data from a source to a destination. MANETs partitioned into disconnected clusters of nodes, standalone nodes, sparsely populated wireless network, extremely high mobility network scenarios, black hole network scenarios, are all examples of ONs. All of these networks have one characteristic in common: an end to end path between the source and destination may never exist at any particular instance of time. A major challenge then is: how to effectively route data packets to their destination without intolerable latency? ONs or Delay Tolerant Networks as they are sometimes called [4], pose a significant design issue hitherto not known. It is to be able to route data in the absence of continuous network connectivity. Important issues that must be considered to retain network connectivity are: what is the least amount of bandwidth required to transfer certain amount of data, which network scenarios cater to which type of applications, how can delay be reduced or be made unnoticeable in these applications, where can opportunistic communication be envisaged, and what communication strategies can be used in order to cope with the lack of permanent links. User mobility can be leveraged to enable direct communication with other hosts to take advantage of this brief unplanned encounter. This can be understood in light of the fact that when users move around a university campus, for example, they do so in communities forming clusters or groups. Being part of a community certainly means being collocated with other members of similar interest. Thus there is a high probability that hosts interested in the same content may meet each other frequently and can therefore carry the data on others' behalf till such time as an encounter between the carrier of message and subscriber of the content takes place.

1.2 Delay Tolerant Networks

A Delay Tolerant or Disruption Tolerant Network (DTN) [26] is an architecture wherein communication protocols route messages through heterogeneous networks and are seamed at their boundaries by gateways. Each DTN node runs on top of its local protocol stack, the DTN layer that combines local addresses of the network environment in the node resides with a cross boundary DTN level address. A fitting example of this would be the Inter Planetary Internet Protocol Suite with a bundle overlay.

1.2.1 Interplanetary Internet

A space station set up on the moon [28] could access email accounts through the satellite network with the Internet on planet Earth. In the interplanetary DTN deployment, scientists used their shuttle's wireless interface to connect to a communication satellite. The satellite, as a gateway, marshals the data packets en route to a receiver on Earth into *bundles*. The *bundle layer* is an overlay atop the various networks through which the scientists' email request will pass. The DTN nodes forming the overlay act as gateways and use the network address and packet format of the environment of which they are a part. Thus, local addressing is used whilst the message is being routed within the same network and bundle layer addressing when being routed across two different networks. The result is that where an enormously expensive infrastructure would have been needed to enable this interplanetary communication, existing network capabilities were leveraged so communication could be made possible albeit with a higher delay.

1.3 Examples of Deployed Opportunistic Networks

1.3.1 Shared Wireless Infostation Model

The Shared Wireless Infostation Model or SWIM [29] shows how information about whales swimming near the surface of water can transmit data about their heart rate, pulse, and body temperature to fixed or mobile information stations floating on water. These stations gather information from the radio tags delivered with a crossbow to the whale and store them in the form of tagged packets. Thereafter, with the help of either an overhead satellite or a MANET formed with other SWIM infostations, the information packets can be delivered to terrestrial networks.

1.3.2 ZebraNet

In the vast savannah of central Kenya, zebras are made to wear special sensors embedded in collars that are put around their neck [23]. These sensors gather important data such as body temperature, heart rate, proximity from other zebras, and number of encounters. These data enable the scientists to study the behavior of zebras towards each other and the environment in which they live. It can help in determining important factors that influence the natural ecology that houses the zebras and their long and short affect on these creatures. Data transmission from zebra collar takes place, when another zebra is within sufficient range. Many techniques such as ‘flooding’ of data or ‘history based’ routing can used in order to efficiently route data towards a base station. Occasionally, scientists in a vehicle carrying a mobile base station go around acting as a ‘ferry’ to which all zebra collars upload the data. This is a form of ‘light networking’ and is needed to monitor the animals in as unobtrusive way as possible in order not to disturb the delicate balance of nature.

1.3.3 DakNet

Dak in Hindi stands for ‘post’ and hence the name DakNet. The paradigm of data dissemination in DakNet [10] comprises of using city vehicles to connect village kiosks to Internet gateways in the city. These kiosks upload data to a city bus equipped with a wireless network interface. Each bus also has memory capacity enclosed in a rugged box to protect against mal function due to wear. When a bus approaches a kiosk, the kiosk uploads data to the bus and when the bus approaches the city office housing data request is catered to by connecting to the internet gateway servers. In this way, connectivity for asynchronous services can be provided to the masses without high deployment costs by using existing transportation infrastructure. Contrary to the conception that rural areas do not need computers, FirstMileSolutions [9] has proved that villagers do need computers and would gladly spend the Rs. 50 fee to purchase Rs. 10 service time. This is indeed so because access to the only communication network, that of the telephone, is hard to get by: an average villager must travel up to a day’s journey and spend Rs. 50 or more just for a single time conversation. Hence, potential for a distributed opportunistic network that works on the idea of a delay tolerant architecture is high and once deployed can cater to myriad online connectivity needs of a community with virtually no prior experience or infrastructure for a wireless communication network.

1.4 Mobility Model Traces

An important component of opportunistic networks is the issue of how the nodes are moving in the network. Specifically, the mobility model of the nodes determines how often the nodes encounter each other. Such encounters can help enable data transfers. It is important therefore, that in order to evaluate protocols in such networks, realistic assumptions about mobility are made. Therefore, researchers want to study real mobility traces and this gave rise to the Haggle Project.

1.4.1 The Haggle Project

The Haggle Project [1] funded by the European Union introduces a novel network architecture that departs significantly from the five-layer TCP/IP protocol suite [2] given in Figure 2. The former completely eliminates the layers above data link layer to delegate forwarding and routing to the higher layer at the application level [1]. Researchers in the Haggle Project are interested in realistic mobility traces such as those gathered from UCSD and Dartmouth to characterize two important parameters, namely, *inter contact time* and *contact duration*. Both parameters were found to exhibit heavy tailed distributions leading to the conclusion that in sufficiently heavy tailed distributions, expected latency can be infinite [31]. However, recently, [3] has shown that mobility models exhibit power law only up to half a day and therefore feature exponential decay which obliterates the former pessimistic view of infinite latency.

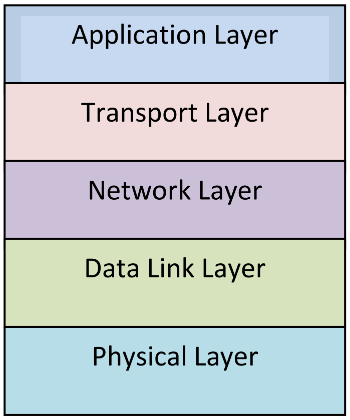


Figure 1.1 TCP/IP Protocol Suite

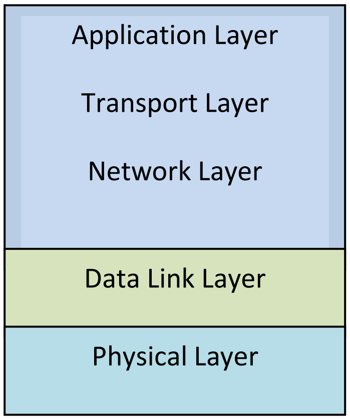


Figure 1.2 Haggle Protocol Architecture

1.5 Ferry Based Routing

In an opportunistic network environment, nodes wanting to exchange data with other nodes can do so in a number of ways. One of these ways is to make use of their mobility. An illustration of this concept is as follows. A source node, say S, wants to communicate with another destination node, say D. Node S is, however, is too far to be connected with D.

Therefore, S may wait to forward message to D till such time as S and D are co located. When this co location occurs, S and D can communicate with each other.

The important thing to note in this example of *proximity based* communication is that scenarios where nodes cannot effectively communicate with each other when they are stationary or have limited mobility, use of *message ferries* or *data ferries can* enable communication. Routing using ferries maybe desirable for an obvious reason: they are special nodes that move around a network wherein other nodes have restricted movement capability.

1.5.1 DieselNet

DieselNet [27] was developed by University of Massachusetts at Amherst and has 40 WiFi enabled city buses. It was deployed in order to understand how well message ferrying approaches would perform under real world network constraints. One of the most important findings of the DieselNet is that as the buses move away from areas of the network areas with route high intersection density, the connectivity decreases sharply. To mitigate this, *throwboxes* [30] were suggested that are essentially solar powered DTN routers and were shown to significantly improve network throughput by acquiring a DHCP lease and connecting to the bus as it passes by. Connectivity and mobility traces have been collected to study up close which would aid in understanding how DTN-like applications can be built.

1.6 Deterministic Cyclic Ferry based Opportunistic Network

Ferries following a deterministic periodic trajectory encounter one another on a regular basis to transfer data at inter contact times that remain unchanged. This may lead to an important consequence in the way the capacity of an opportunistic network is utilized to send data to the destination with bounded delay. Delay Tolerant Networking applications may find this peculiarity useful as guarantees on the quality of service can increase the appeal for such deployments.

Inter contact distribution and contact frequency are two mobility metrics that lend insight into how change in network parameters would affect the capacity and hence the *expected latency* and *delivery ratio* [31]. Based on these observations, it would be straightforward to predict the performance of the network as time tends to infinity.

However, an important issue that arises in the study of such networks is the deployment of test beds that enable the researchers to study real world constraints. Results from simulation based studies are generally inferior to real world deployments in their accuracy in predicting the network performance. Therefore, the objective is to use a mobility model that captures the inherent characteristics of inter contact distribution and contact frequency of popular mobility models such as those evaluated in [3] and [31].

We look into inter contact distribution and contact frequencies plots obtained from simulation runs using map based movement models using the city route map of Helsinki, Finland and show that the plots exhibit the same characteristics of mobility as found in mobility traces of InfoComm ’05, Cambridge [31], and UMass DeiselNet [5]. We predict and empirically prove that a cyclic deterministic network has a constant per hour capacity. Further, we show empirically how introducing heterogeneity in the ferry movement and message forwarding can lead to gain in network performance.

We discuss the background and related work in Chapter 2, a deterministic cyclic ferry based network in Chapter 3, system design and implementation in Chapter 4, simulation results in Chapter 5, and, finally, conclusion along with future work in Chapter 6.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Introduction

Message Ferries (MFs) find applications in various networking scenarios, such as, geographically constrained, disaster struck, or that are cost driven [6]. However, agile nodes that are more mobile can alleviate network partitioning by traversing trajectories encompassing the location of the challenged nodes. While within the transmission range, the agile ferries can transfer message picked up from the source to the destination node. Research work done involving message ferries can broadly be classified into the following categories:

1. Route Design
2. Hierarchical Ferry Routing
3. Buffer Management
4. Ferry Transmission Scheduling
5. Cluster Based Ferry Routing
6. Hybrid Ferry Routing in Partitioned MANETs
7. Optimal Packet Scheduling to minimize power Consumption
8. Differentiated Services

Table 2.1 on Page 14 categorizes some of the research works into applicable classes. We provide a brief overview of these works and discuss the more relevant hierarchical forwarding, route design, differentiated services, transmission scheduling, and cluster based forwarding in the following sections.

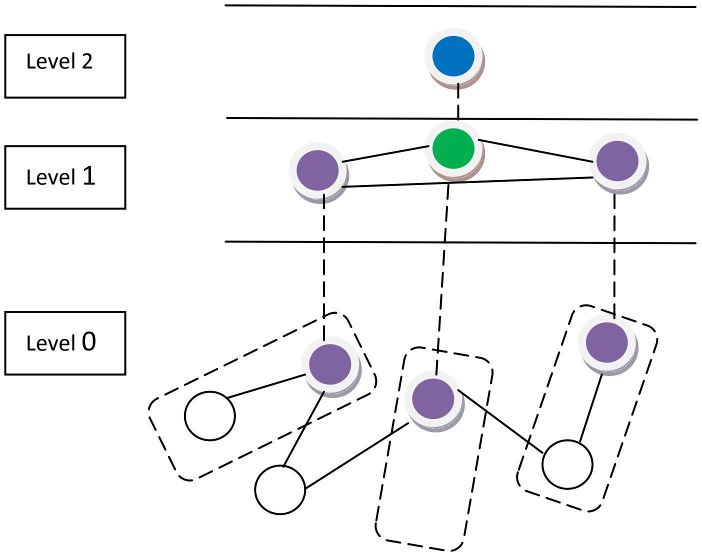


Figure 2.1: Hierarchical Cluster Organization with 3 levels, colored nodes are cluster heads

2.2 Classification on the basis of Ferry Network Topology

Message Ferries (MFs) can disseminate message working under several different types of network configurations [18]. These network characteristics are generally of the following types in which the MFs:

1. Cater to either fixed or mobile regular nodes
2. Make intelligent route design decisions or follow predesigned routes
3. Collaborate to serve network better or work independently
4. Are deployed solely for purposes of message dissemination or installed for accomplishing other tasks but piggyback messages while traversing their trajectories.

We list the network scenarios under which various MF schemes operate in Table 2.2 on Page 15 and compare them with the overlay-based routing to be described in Chapter 3.

2.3 Multilevel Hierarchical Organization

Liu and Wu in [13] present a cyclic deterministic network that supports routing by organizing space time information of future contacts into a hierarchical structure, see Figure 2.1.

Since contact information is organized into multiple levels, a node need only be aware of the next node in the level above to route data to the destination. This leads to a form of compression that has little affect on the routing performance. Specifically, instead of storing future contact information of *N* nodes that are in the system, the nodes instead store *O(log N)* amount of information where *log N* corresponds to the level of hierarchical clustering used. In order to route, each node stores information about its adjacent clusters, i.e., those with which the node is connected and the clusters in the hierarchical level above it.

Link delays maybe invariant and be the same for every cycle or variant in which case they are different for each ferry cycle. When delay is invariant, each link that would take place at some point in time is abstracted as the expected delay weighted static link. In other words, a link between two nodes A and B, if it takes place after 5 seconds from the start of each cycle, then the delay for link between A and B is stored as 5. A routing table of delays is created for all the links a node has at that level.

On the other hand, if the network is described by static time invariant links, the link information can be stored in routing tables and Dijkstra's Shortest Path algorithm can be used on this information to calculate the shortest path between two nodes. However, for time variant delays, this is insufficient. Statistical expected delay value is then used for link delay that would be incurred between two nodes within the LCM of the cycles of the delays for the set of shortest paths between them.

2.4 Hierarchical Routing

Routing using a hierarchical organization of ferries based on similarity of contacts or expected link delay has been evaluated in [24] and [13] respectively. The later technique works by dividing ferries intro clusters and these clusters into a multilevel hierarchical structure discussed in Section 2.8. In this section, therefore, we focus on the technique employed by Liu and Wu in [13]. When a node wishes to send message to a destination node, assume that it must know either the hierarchical cluster address of the node or use location service to calculate the address given the destination ID. It must also have information of the all the clusters to which it belongs at all levels. Routing is then done by identifying the highest cluster for which the source and destination are in disjoint clusters, recursively indentifying and forwarding message to nodes at the lower levels that contain delay information about the cluster heads to which the concerned nodes belong.

2.5 Route Design

Chua and Yang in [19] present an Elliptical Zone Forwarding in which a ferry provides differentiated service to urgent and non urgent messages besides employing a route design mechanism. More specifically, a ferry calculates the delay requirement R_i of a node Y_i and uses R_i to construct an elliptical route with foci at its own location *X* and that of the node to pick the message from location at distance *Y*. All nodes at distance *Z* from *X* falling within the ellipse are served if $XZ + YZ < R_i$. This leads to a scheme that reduces overall delay but has the inherent weakness that it gives more priority to the message that have already been picked as compared to the ones that are waiting to be picked up. This compromises with the urgent messages that are not already present in the ferry's buffer but are none the less just as important. To mitigate this, Chuah and Yang in [15] propose three different route design schemes namely Dynamic *K*-lookahead Scheme (DLAS), Fixed K-Lookahead Scheme (FKLAS), Minimum Weighted Sum First Scheme (MWSF). DLAS has a list of nodes that it would visit in its tour. Those nodes that have a message deadline less than a threshold are visited first. FKLAS maintains an ordered list of *K*-urgent messages along the information of whether they have been picked up or not, In MWSF, three weighted metrics of buffer overflow time for source node, expiry time of urgent message for destination node, are used to decide which node to visit first. Mukarram and Ammar in [22] investigate how regular nodes can be assigned to ferries and how ferries must be allocated to areas so that work load is optimally distributed amongst the ferries. In [18], Zhao and Ammar show that when the regular nodes are stationary, designing a an optimal ferry route using single or multiple ferries in terms of throughput and latency is NP Hard and provide heuristics to design a route.

2.6 Differentiated Services

Chua and Yang in [19] and Viswanathan and Chuah in [20] present schemes of ferry route design wherein the ferry travels first to the nodes that have urgent messages and schedules the messages as having higher priority. These urgent messages are then delivered earlier as the ferry travels to the concerned destination nodes before it travels to other nodes waiting to forward message to the ferry.

2.7 Transmission Scheduling

In [21], Kavitha and Altman propose a method of optimally scheduling transmission between a ferry following a cyclic deterministic route and nodes within the region of its transmission range. The ferry has a finite number of stops it serves and for the duration of its pause, it schedules the order in which it would engage in a communication with nodes waiting to transfer messages to the ferry.

2.8 Cluster Based Forwarding

Ahmed and Kanhere in [24] present a dynamic programming approach to create clusters amongst ferry nodes such as public buses that traverse overlapping and intersecting routes periodic routes. Ferries meeting a set of nodes and not others are clustered into one single set. Recursive hierarchical clustering is done in such a way that a ferry meeting nodes belonging to two different sets appears in both clusters. In this way, a multilevel ferry hierarchy emerges using data from connections adjacency matrix.

Table 2.1 Message Ferry Research Work Classification

Ferry Based Scheme	Route Design	Hierarchical Organization	Cluster based Forwarding	Ferry Scheduling Policy	Buffer Management	Hybrid Routing	Differentiated Services
[6]	x						
[24]		x	x				
[22]							
[25]							
[15]							
[14]				x			
[20]	x			x			x
[16]				x		x	
[13]	x	x	x				
[11]	x						
[7]	x				x		x
[19]	x						x
[21]	x			x			

Table 2.2 Classification of Message Ferry Schemes on the basis of Network Topology

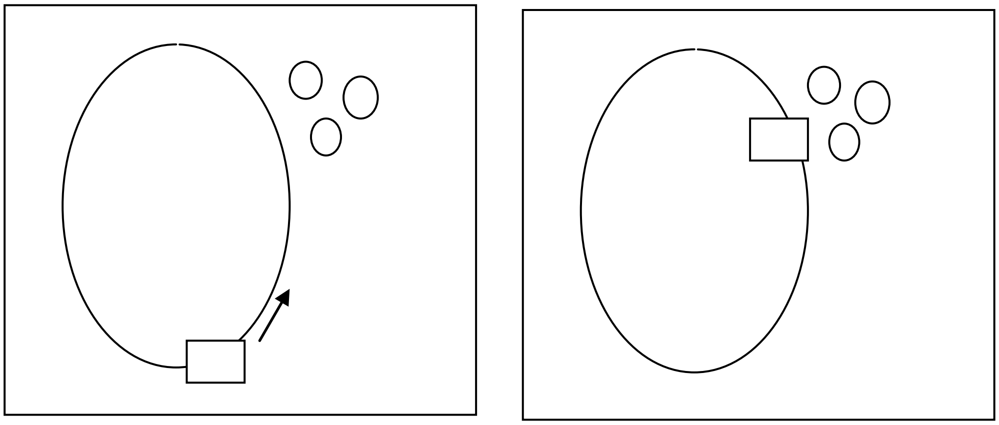
Ferry Scheme	Route determined by Message Forwarding concerns	Route determined by non Message Forwarding concerns	Regular Nodes Stationary	Regular Nodes Mobile	Level of Ferry Coordination: Independent or Coordinated	Level of Regular Node Coordination	Regular node Ferry Designation: yes or no	Number of Ferries	Proactive Ferrying
[6]	Yes	No	-	-	Independent	None	-	Single	Yes
[24]	Yes	Yes	No	Yes	Independent	-	-	Multiple	No
[21]	Yes	No	Yes	No	Independent	None	-	Multiple	Yes
[22]	Yes	No	No	Yes	-	-	No	Single	Yes
[20]	Yes	No	Yes	No	-		No	Single	-
[17]	Yes	No	Yes	No	Independent	None	No	Multiple	Yes
[19]	Yes	No	No	Yes	Independent	None	No		Yes
[7]	Yes	Yes	Yes	Yes	Independent		No	-	Yes
overlay ferry	Yes	Yes	No	Yes	Independent	None	Yes	Multiple	No

CHAPTER 3

OPPORTUNISTIC FERRY NETWORK WITH DETERMINISTIC CYCLIC MOBILITY

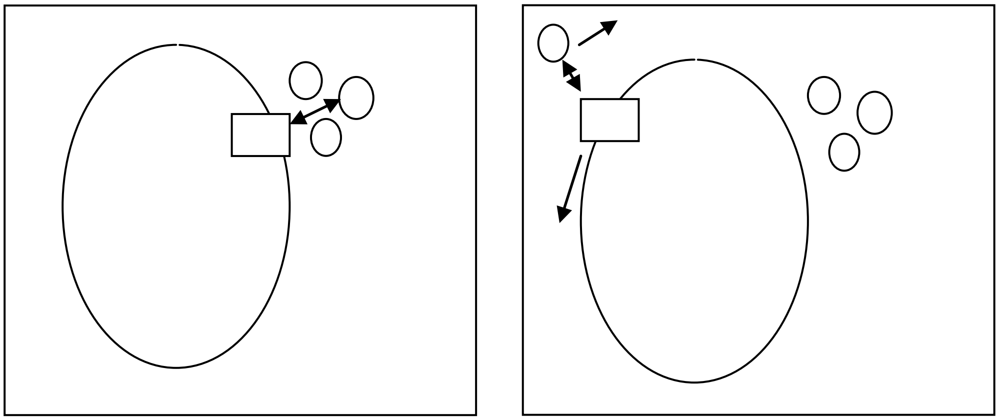
3.1 Deterministic Cyclic Message Ferry Routes and a Constant Network Capacity Per Hour

In Chapter 2, we discussed existing research work on message ferries some of which employ ferries having a fixed cyclic trajectory and contacts can take place either during a pause duration as shown in Figure 3.1a, b and c or they can take place while a regular node passes by the ferry node, see Figure 3.1d. Most of the works are concerned with an efficient route design algorithm, transmission scheduling to reduce power consumption, and evaluating performance under various message forwarding schemes. However, it would be interesting to explore if a deterministic MF scheme can be used to provide guarantees, if any, on the network capacity available. It must be remembered that Delay Tolerant Networking applications such as email, file transfer, personal messages inherently have a larger delay range that would not necessarily disrupt their operation. Nonetheless, if a guarantee as to the network capacity can be provided to applications, novel message scheduling and forwarding algorithms can then be designed that take advantage of this feature of deterministic MF networks. For a ferry node we employ Epidemic Routing [25] to gauge the network performance. In this work, please note we have used the terms bus and ferry interchangeably.



a. Ferry Approaching Regular Nodes on a Cyclic Route

b. Ferry Pausing at a Pre designated stop in its route



c. Message Transfer Opportunity between Regular

d. Message Transfer Opportunity between a Regular Node and Ferry while both are moving and pass by each other

Figure 3.1 Contact between Message Ferry and Regular Node

3.2 Overlay Network of Agile Ferries

Ferries meeting each other after fixed time intervals moving at constant speeds would give rise to contacts that change very little with increase in number of nodes. Although, the latency and delivery ratio are expected to improve with increase in contact opportunities as more nodes are added to the network, this improvement should not be significant. The reason is that if additional ferries follow the same fixed routes, the inter contact duration that arises as a result of difference in geography of ferry movement, would remain preserved with additional ferries having a constant speed. However, existent ferry networks can be deployed for reasons other than message forwarding. For instance, a public transportation network such as trams and buses, that have an almost constant speed, traverse fixed cyclic routes, can be used to piggyback messages between the regular nodes. For this reason, we consider ferries belonging to this category and show that under deterministic network conditions, ferries with higher speeds can bring about significant performance improvement in terms of a slow rising message latency.

3.2.1 Direct Delivery and Epidemic Routing: Two extremities in Opportunistic Message Dissemination

Epidemic Routing [25] operates by flooding messages to all neighbors and not surprisingly incurs the least amount of delay and is the most robust of all forwarding algorithms. Therefore, flooding is used in scenarios where network is highly partitioned and sparse. However, it leads to lots of unwanted copies of messages that results in the increase of the over head ratio. On the other hand, Direct Delivery (DD) incurs the longest latency but the least over head ratio. It operates by storing the message till it meets the message destination and messages are not duplicated en route to the destination node.

3.2.2 Heterogeneous Mix of Direct Delivery and Epidemic for Performance Improvement

If agile ferries use Direct Delivery to route messages between the ferries, significant performance improvement can be achieved in terms of message latency. Use of DD to lower latency at first may not sound convincing, however, it is to be pointed out that high ferry speeds help in offsetting the latency incurred using DD. The concept of overlay ferries is illustrated in Figures 3.2 and 3.3 respectively.

3.3 Ferry Schedule Based Routing

When occurrence of future contact instances between ferries with cyclic deterministic routes and regular non-ferry nodes are known in advance, messages can be routed relayed through ferries to the appropriate destination. In other words, if a source node S wants to send message to a destination node D, it transfers message to ferry using D’s schedule of ferry contact times. Each ferry and non ferry node in the network has schedules having entries listing arrival times at different stops. Regular nodes whose trajectory comprises of co location with ferries when it pauses at one of its stops, can have messages delivered to them if their schedule overlaps with that of a ferry being used to relay the message. We use the terms *bus* for a ferry and *traveler* for a non ferry nodes interchangeably for schedule based routing as throughout this thesis. Message forwarding mechanism for bus and travelers is different and we discuss it in Section 3.3.1 and 3.3.2 respectively.

We would like to measure how *average latency*, *overhead ratio* and *delivery ratio* varies with increasing number of non ferry nodes or traveler nodes. We present plots of affect of scalability in terms of traveler nodes on routing performance in Chapter 5.

1. Bus Router
2. Traveler Router

3.3.1 Bus Router

3.3.1.1 Bus -Traveler Interaction

To reiterate, bus denotes a ferry while traveler connotes a regular node wishing to exchange data. When a ferry either passes by a traveler node or comes to halt and the traveler node happens to be within its transmission range, messages are exchanged between the two nodes. Bus delivers message for the traveler it has encountered, if any, stored in its buffer and in turn receives new messages to be routed to the destination from the user. Upon receiving the message, it checks if the message destination would be encountered while following its own schedule. If not, the bus instead stores message in the outgoing buffer to be relayed to a bus that does have an intersection with the destination node schedule.

3.3.1.2 Bus-Bus Interaction

When a bus A receives message from a bus B, it stores the message in outgoing buffer. However, when a upon a bus encounter, A sends all those messages that are lying in its outgoing buffer but that are not a part of its schedule.

3.3.2 Traveler Router

3.3.2.1 Traveler-Bus Interaction

When a traveler comes within transmission range of a bus node, it tries to send all messages in its buffer until it is no more within communication range or all messages have been transferred.

3.3.2.2 Traveler-Traveler Interaction

If traveler A comes within communication range of traveler B, all deliverable messages are exchanged between the two.

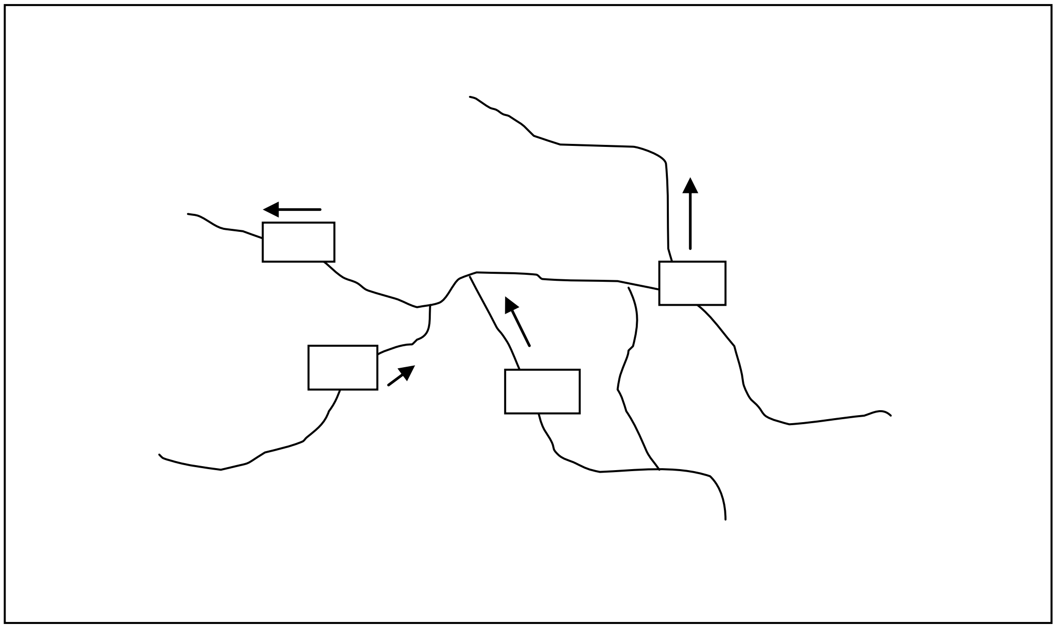


Figure 3.2 Regular Ferries

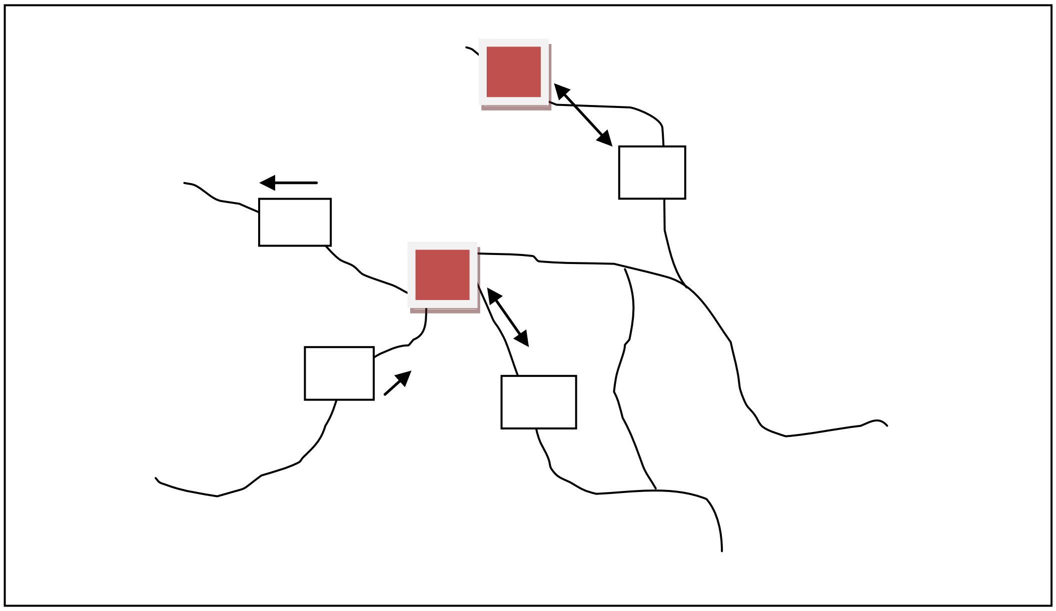


Figure 3.3 Overlay Ferries exchanging messages with regular ferries

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Introduction

Implementation was done using Java based simulator for Opportunistic Network Environments (ONE) [8]. In the sections that follow we will briefly describe the modules designed and implemented in order to facilitate understanding of our forwarding mechanisms.

4.2 Bus or Ferry Movement Model

The ONE has been modeled around the concept of Delay Tolerant Networking discussed in Chapter 1. Each network node besides the capabilities of movement, radio interface, energy consumption, also has the ability for perpetual storage. The main modules include the movement models, routing models, result reporting, connection layer, network layer and most importantly visualization and analysis. Event generators can be external or internal depending on the implementation and can be specified through internal generator class or external event traces. ONE can generate mobility based on existing movement models or external movement traces. It is able to integrate seamlessly with DTNSim2 [12] allowing developers to code in DTNSim2 and import their work to be visualized in ONE [8]. Below we present some important components of the simulator that have been used to analyze deterministic cyclic ferry network, implement overlay ferries, and scheduled routing.

4.2.1 Movement Models

We used the following movement models to evaluate cyclic ferry network and implement Destination Chooser Movement Model to be described in Section ONE provides the following movement models that can used to generate mobility models for opportunistic network nodes.

1. Bus Movement
2. Bus Traveler Movement
3. Shortest Path Map Based Movement

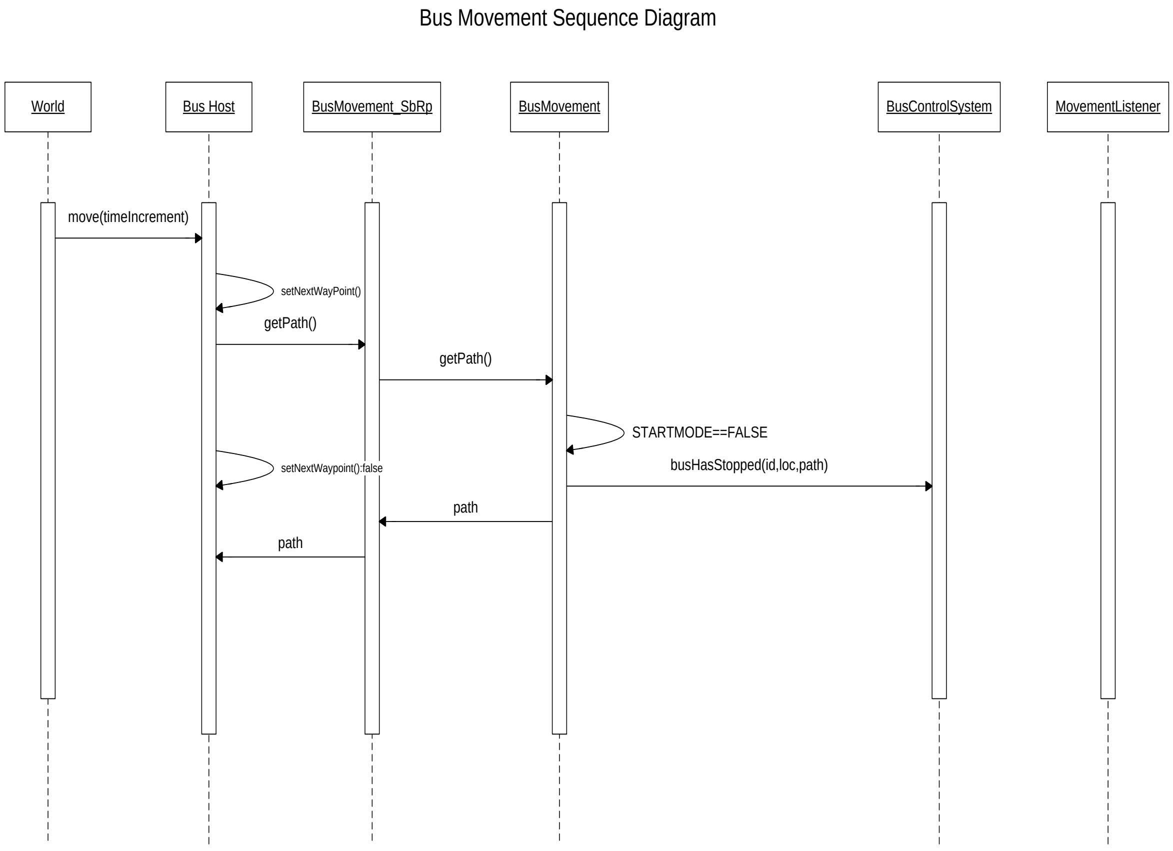


Figure 4.1: Sequence Diagram of Bus Movement Schedule Based Routing interacting with Bus Movement Class

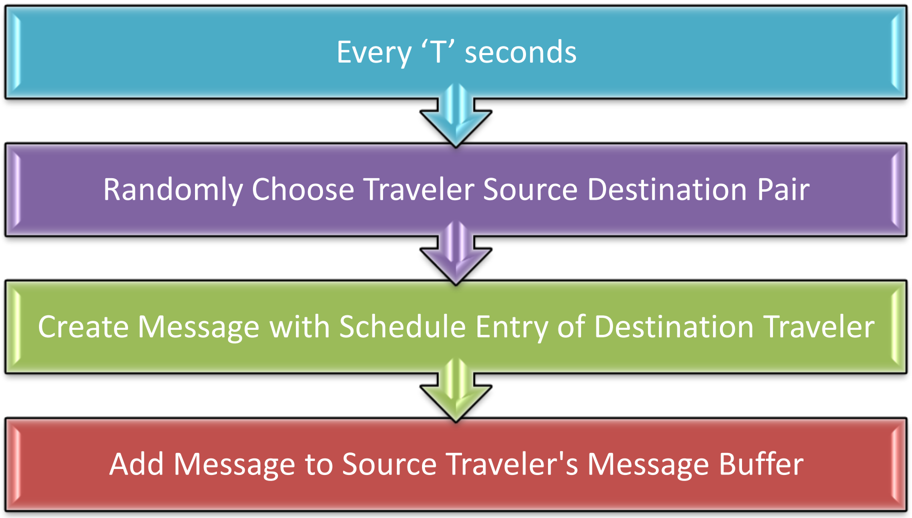


Figure 4.2 Scheduled Routing Message Create Event

4.3 Ferry Node Architecture

Regular Ferry nodes use Epidemic Routing as the Routing module and Bus Movement Model. The architecture is given in Figure 4.3 and the forwarding mechanism in Algorithm 4.1. When a node comes within transmission range, a connection with it is created and added to the list of active connections. Every 1 second, all connections in the `ConnectionList` are updated in which connections that have gone down are removed while new ones that have been formed are added. Further, messages are forwarded using the appropriate routing mechanism. We show here only the routing mechanism using connections in the `ConnectionList` and not their maintenance.

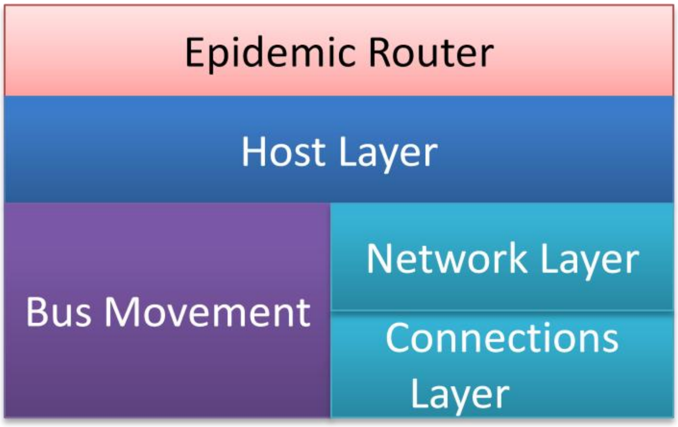


Figure 4.3 Architecture of Regular Node

```
Regular Ferry Node  UpdateConnections()
begin:
for con: ConnectionList
    forwardAllMessages(con.otherHost.id)
end for
end
```

Algorithm 4.1: Epidemic Routing Mechanism used in Regular Ferry Nodes

Overlay Ferry nodes use Direct Delivery Router and the Bus Movement Model shown in Figure 4.4. The forwarding algorithm using Direct Delivery Routing is shown in Algorithm 4.2.

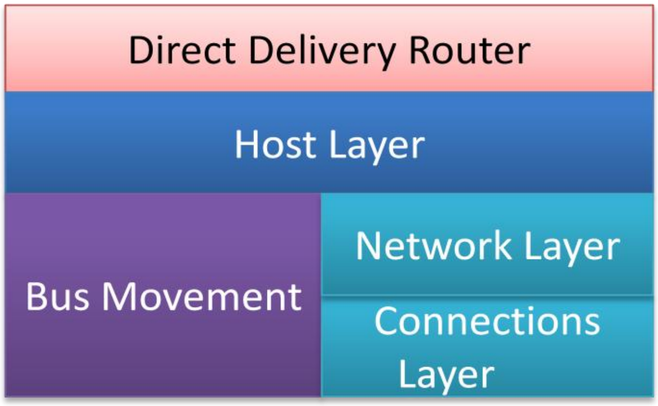


Figure 4.4 Architecture of Overlay Ferry Node

```
Overlay Node UpdateConnections()

begin:

for con: ConnectionList

    if for message: MessageList

            message.id == con.otherHost.id

            forward(message, con.otherHost.id)

    end if

end for

end
```

Algorithm 4.2: Overlay Ferry Forwarding Algorithm

4.4 Schedule Based Deterministic Routing

The design of Scheduled Based Routing consists of three main components:

1. Movement Models for Buses and Travelers
2. Routing Mechanisms for Buses and Travelers
3. Randomly Choosing a Source-Destination Pair

4.4.1 Movement Model for Buses

4.4.1.1 `BusMovement_SbRp`: Ferry or Bus Movement for Scheduled Routing Protocol

Bus Movement for Scheduled based Routing Protocol (`BusMovement_SbRp`) extends Bus Movement. The call to `getPath()` method and additionally comprises of a bus schedule manager (SM). The SM works by executing the following steps:

1. Log stop arrival time for each Stop in an array
2. At the end of first route cycle, use stop arrival log in Step 1 to calculate future stop arrival times for the entire simulation duration.
3. Create List of Schedule Entries comprising of a stop coordinate, stop arrival time and Bus ID, see Figure 4.5.

4.4.1.2 Bus Schedules

A bus schedule consists of the following components:

1. Bus ID: An unique integer ID of the bus that stops at the bus stop specified in the entry
2. Stop: A coordinate of the form (x,y) which is a point belonging to the simulation map and is one of the stops of the Route assigned to the bus control system.
3. Time: A double value at which the said bus arrives at the location ‘Stop’.

BUS ID	STOP COORDINATE	TIME

Figure 4.5: Schedule Entry Format

4.4.2 Movement Model for Travelers

4.4.2.1 `DestinationChooser`

`DestinationChooser` Movement Model (DCM) extends Switchable Movement class and switches between the Bus Traveler Movement and Shortest Path Map Based Movement classes. For details of on Bus Traveler Movement and Shortest Path Map Based Movement please see Appendix A. This switching takes place as follows:

1. Start off in `ShortestPathMapBasedMovement`
2. Every update, check if is time to proceed towards a bus stop.
3. If yes, issue an ALERT and switch to `BusTravelerMovement_SbRp`
4. Else, do nothing.

The switching state diagram can be seen in Figure 4.6.

Bus Traveler Schedule Based Routing Protocol (`BusTravelerMovement_SbRp`) class extends ONE API Bus Traveler Movement class. `BusTravelerMovement_SbRp` class has state transitions for the traveler node, modified from the state transitions in Bus Traveler Movement class, shown in Figure 4.7.

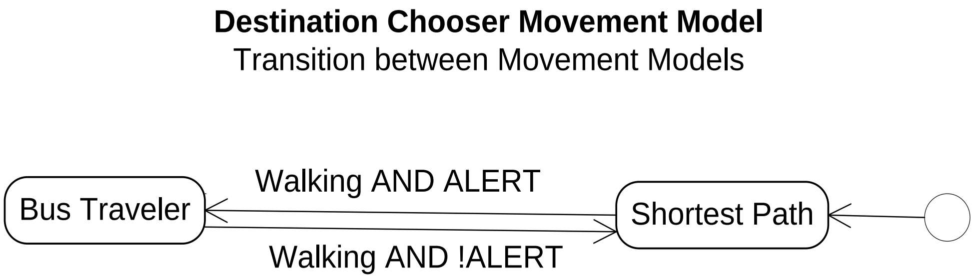


Figure 4.6: State Transition Diagram of `DestinationChooser` Movement Model

The state transition can be illustrated with the following diagram, where explanation of the following terms used in the diagram is in order:

1. EBS = End Bus Stop
2. SBS = Start Bus Stop
3. Loc = current location of traveler
4. Decided = STATE_DECIDED_TO_ENTER_A_BUS
5. Walking = STATE_WALKING_ELSEWHERE
6. Waiting = STATE_WAITING_FOR_A_BUS
7. Traveling = STATE_TRAVELING_ON_BUS

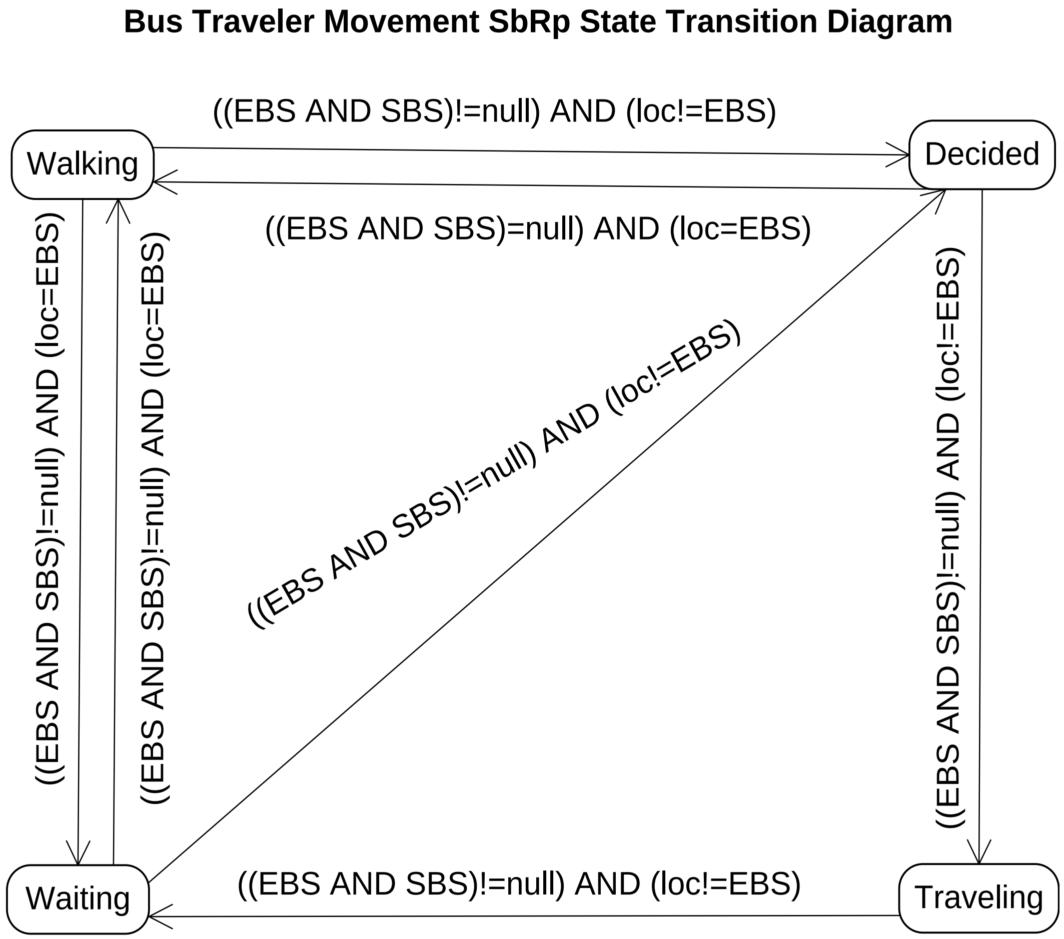


Figure 4.7 Transition Diagram of Traveler States

4.4.3 Bus Router

The routing mechanism of bus nodes discussed in Chapter 3 is given in Algorithm 4.3. When a node comes within transmission range, a connection with it is created and added to the list of active connections. Every 1 second, all connections in the `ConnectionList` are updated in which connections that have gone down are removed while new ones that have been formed are added. Further, messages are forwarded using the appropriate routing mechanism. We show here only the routing mechanism using connections in the `ConnectionList` and not their maintenance.

```
Bus UpdateConnections()

begin:

for con: ConnectionList

    for message: MessageList

        if  con.otherHost instanceof traveler

            if message.id == con.otherHost.id

               forward(message, con.otherHost.id)

            end if

            else if con.otherHost.ScheduleList ∩

                    message.destinationHost.ScheduleList ≠ ∅

                    forward(message, con.otherHost.id)

             end if

end for

   .
```

Algorithm 4.3: Bus Routing Mechanism

4.4.4 Traveler Router

The routing mechanism of traveler nodes discussed in Chapter 4 is given as in Algorithm 4.4. When a node comes within transmission range, a connection with it is created and added to the list of active connections. Every 1 second, all connections in the `ConnectionList` are updated in which connections that have gone down are removed while new ones that have been formed are added. Further, messages are forwarded using the appropriate routing mechanism. We show here only the routing mechanism using connections in the `ConnectionList` and not their maintenance.

```
Traveler UpdateConnections()
begin:
for con: ConnectionList
      for message: MessageList
          if con.otherHost instanceof traveler
             if message.id == con.otherHost.id
                forward(message, con.otherHost.id)
             end if
          else forward(message, con.otherHost.id)
          end if
end for
end
```

Algorithm 4.4: Traveler Routing Mechanism

4.4.5 Message Create Event for Scheduled Routing

In ONE, events can be generated from external trace files that specify the type of event, and time at which the event is to take place. Events, however, can also be generated during the simulation with the help of the Message Event Generator class. Message Create Event for Scheduled Routing extends Message Create Event class and creates a message with an additional schedule entry, the format of which is given in Figure 4.5 on Page 28.

4.4.5.1 Choosing Random Traveler Source-Destination Pairs

The Message Event generator class chooses random source destination pair by generating random addresses using the `Math.Random` class in Java as shown in Figure 4.2 on Page 24.

4.5 Modeling Contact Frequency

4.5.1 Contact Frequency per Route Cycle

Number of contacts taking place per route cycle per ferry can be modeled as follows. Contacts with a ferry can take place while it is paused at a stop or while it is moving towards its destination traversing a path. A node that is within contact range of the ferry would be able to establish a connection with it. N_{stops} is the number of stops in the ferry route, $N_{ferries}$ is the total number of ferry nodes and C is the number of contacts.

1. Contacts at the stops

$$C = (N_{ferries} - 1) * \frac{1}{N_{stops}}$$

2. Contacts in between stops
    - Probability P of a contact between a node pair while traveling on path between two stops depends on:

        a. Ratio of speeds of nodes $\frac{v1}{v2}$

        b. Difference in their directions, θ_1 and θ_2 at contact instance i: $e^{|\theta_1 - \theta_2|}$

    - Number of Contacts between two stops S_a and S_b for a given path is derived as:

a. Probability that a contact will take place between two given stops in one second between ferry node N_1 and N_2 with speed v_1 and v_2 and direction of displacement θ_1 and $\theta_2$respectively, is:

$$p_{12} = e^{-|\theta_1 - \theta_2|} * \frac{v1}{v2}$$

b. Number of contacts taking place between a given pair of stops in t seconds, where t_{path} is the time taken to traverse the path between the two stops S_a and S_b is:

$$p_{12*} t_{path-ab}$$

c. Number of contacts a given ferry node n_1 can have between two stops S_a and S_b for a given path is then:

$$C_{stops-ab} = t_{path-ab} * \sum_{j=1}^{N_{ferries}-1} p_{1j}$$

- Number of paths in a route = N_{stops}
- Therefore, total number of contacts that a ferry can have while traversing a route are:

$$C_{route} = \sum^{Npath} C_{stops-paths}$$

4.5.2 Contact Frequency per Hour for a given Ferry

- If time taken to complete a route cycle is t_r seconds, then number of cycles in one hour would be:

$$N_{cycles} = 3600/t_r$$

- Since, t_r is a constant, N_{cycles} is also a constant. Therefore, we can say that the total number of contacts that a ferry node has within an hour is given by,

$$C_{hour} = N_{cycles} * C_{route} \quad \cdots A$$

- Since, N_{cycles} and C_{route} are constants for constant speeds and routes traversed, therefore, C_{hour} is a constant for a ferry.
- Total Number of Contacts per hour per ferry= C + C_{hour} = CONSTANT. In the same way, Equation A would hold true for all the ferries in the system.

We can see from Figure 5.7 on Page 51, that this is indeed true. In this case, number of contacts taking place per hour in the entire system is constant and equal to 100 contacts per hour.

4.6 System Scenario

The system is defined by a set of network parameters that influence the way dissemination takes place, how much data is transferred, how many messages reach the destination and what is the latency incurred in doing so.

4.6.1 Network Parameters

The parameters along with their units are tabulated in the table that follows:

Table 4.1 Network Parameter Units

Parameter	**Unit**
Rate of Message Generation	seconds
Message Size	Kilo Bytes (KB)
Buffer size of buses	Mega Bytes (MB)
Message TTL	minutes
Speed of Travelers	m/s
Speed of Buses	m/s

A note on each of the terms listed above is as below:

1. Message TTL stands for Message Time To Live measured in minutes
2. Traveler Group Size: Number of travelers belonging to a particular group
3. Bus Group Size: Number of buses belonging to a particular group
4. Message generation rate: Number of messages generated per second in the entire system

4.6.2 Performance Metrics

4.6.2.1 Average Latency

As each message gets delivered to the destination the `MessageStatsReport` logs the amount of time it took between message creation and message delivery. Average Latency is calculated at the end of the simulation and corresponds to the average of latency for the total number of messages delivered in the system.

4.6.2.2 Overhead Ratio

Overhead ratio refers to the ratio of useless message generated to the delivered messages. According this definition, the range of values that the overhead ratio can assume belongs to the interval $[0 , n]$, where $n \in R$, the set of all real numbers.

4.6.2.3 Delivery Ratio

The delivery ratio is the ratio of all messages created in the system to the delivered messages. Note that it does not include the message replicas generated.

4.7 Assumptions

In order to avoid an overtly complex design model of our system, we have made some assumptions about the design issues that we feel are either justified by our observations in the real world or involve an issue which is out of scope of this work. These assumptions are listed below:

1. Simultaneous Connections

Nodes are capable of connecting to multiple nodes at the same time. The function of the network layer in the simulator is to detect whether another node is within its transmission range.

If there is such a node, then the network layers of the concerned nodes form a connection. If any of the nodes have messages to exchange, this is the time when it is done.

2. Message Abortion

If any one of two nodes, that are within transmission range of each other and a message is en route from one node to another, moves away such that the connection between them is severed, the message gets aborted.

3. TTL Check

The TTL is not checked at each update interval, but rather every time that a predefined interval for this check has elapsed.

4.8 GUI Snapshot

A snapshot has been presented to illustrate visually the simulation scenario. Visual components and the capability they represent are given as below:

1. Transmit range: green circles. Smaller transmission range is that of traveler nodes.
2. Light gray line segments: city streets and main roads

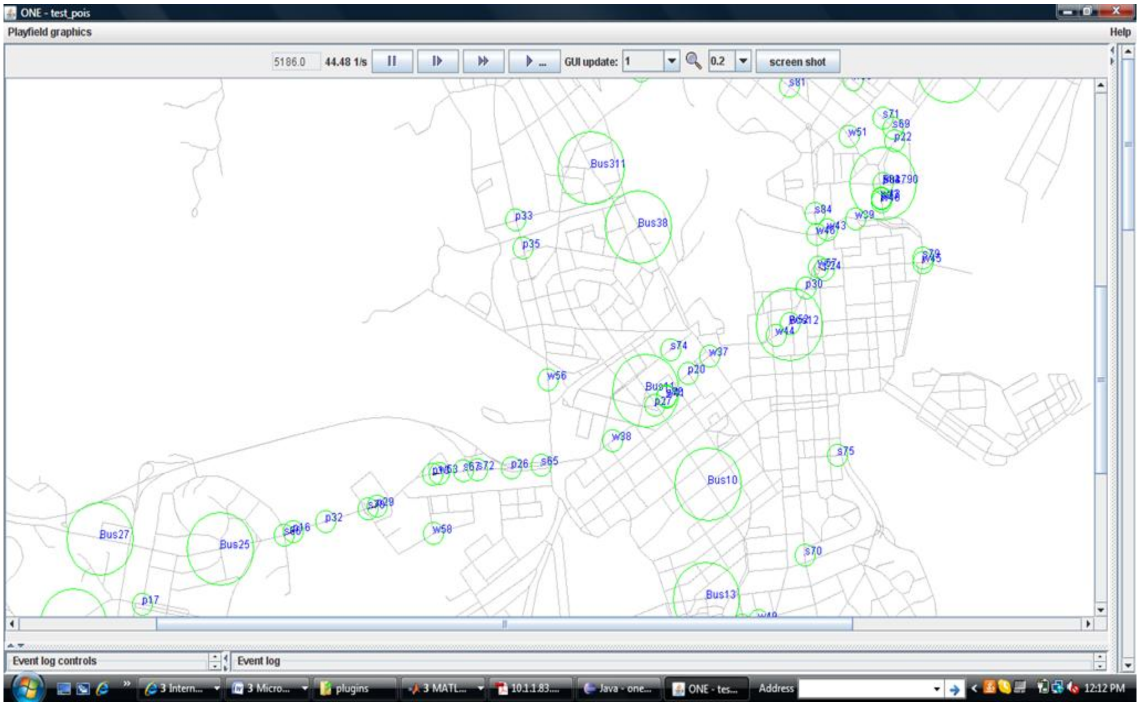


Figure 4.8: Bird’s Eye View of Traveler and Bus Nodes

CHAPTER 5

RESULTS

In order to understand how deterministic mobility affects the performance of the ferry based network, mobility metrics of inter contact duration and contact frequency have been empirically analyzed. Specifically, the impact of mobility model speed and number of ferry nodes on inter contact duration and prediction of network performance via this impact has been presented. Further, we have plotted and confirmed our intuition of a constant per hour capacity of the network and shown that latency growth per message can be impeded with use of agile ferries to form an overlay over the existing ferry network. Finally, plots of histograms of delivery ratio, average latency and over head ratio are presented to demonstrate performance of scheduled routing between ferries and traveler nodes as the number of traveler nodes increases.

5.1 Definition of Performance Metrics

The following metrics will be used to evaluate the performance of the city bus network so that conclusions can be drawn as to which parameters affect efficient message dissemination and to what extent.

1. Delivery Ratio $= \frac{Nr\ of\ Messages\ created - Nr\ of\ messages\ dropped\ or\ removed}{Nr\ of\ Messages\ created}$

2. Overhead Ratio $= \frac{Nr\ of\ useless\ Messages}{Nr\ of\ Delivered\ Messages}$

3. Average Latency$= \frac{sum\ of\ individual\ Message\ Latencies}{Nr\ of\ Messages\ Delivered}$

5.2 Mobility Model Metrics

One of the most important parameters that would affect the way messages are disseminated in a network is the network's mobility model. In our city bus network, there are two types of nodes and hence two different types of mobility models. We present below the three significant parameters that characterize any mobility model.

1. Inter contact Time : Duration of time elapsed between two contact instances measured in seconds (s).
2. Contact duration : Duration of a contact between two nodes measured in seconds (s).
3. Contact Frequency : Contact frequency is the number of contacts that take place system-wide per hour.

5.3 Mobility Model Plots

The mobility model or in other terms the above parameters are already fixed if the bus nodes are traveling on predefined roads. Before we start our analysis with variations of different parameters, we first present the mobility model of only the buses running their routes. It is important to do this so we can understand the basic characteristics of our mobility model and a base line to compare further plots with. Towards this end, we take 8 buses belonging to 8 different regions of the city. The plots of Inter Contact Time Distribution (ICTD) and Contact Duration Distribution (CDD) follow in the next sub section.

5.3.1 Plots of CCDF ICTD curve and the CDD curve

In Figure 5.2, we can see that by plotting the inter contact time log-log CCDF curve, inter contact times have exponential decay.

Further, we would also like to see the frequency distribution of contacts that were of certain duration. From Figure 5.1, we can observe that the mode is at 30s. In other words, 20 bus-bus contacts last for 30s duration, 11 contacts of duration slightly less than 30s, that is, around 29 s which can be utilized to send longer messages without messages getting dropped. We also observe that as we increase the contact duration our curve increases with a low variance, decreases and increases again to yield the second mode at around 130s.

5.3.2 Contact Time Distribution approximately power law

In Figure 5.1, the stem plot of contact time distribution reveals an approximate power law distribution characterized by the early part of the curve till contact duration of approximately 30s.

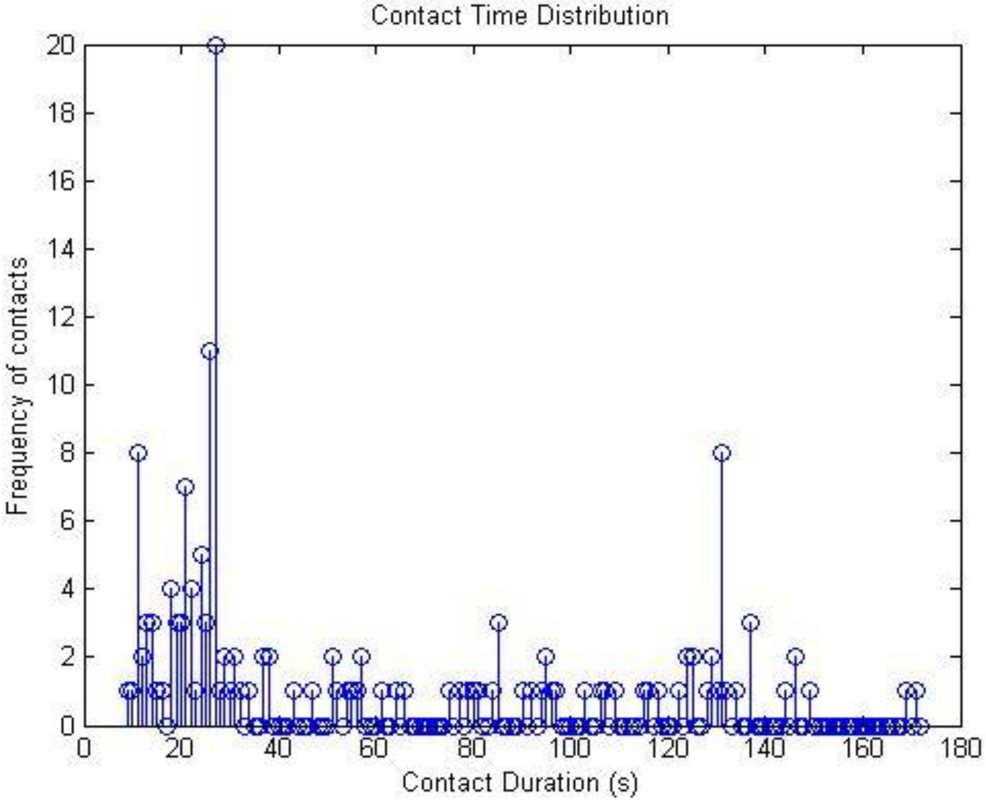


Figure 5.1 Contact Duration Distribution for Regular Ferry Network

5.4 Expected Latency with approximate Power Law Hypothesis

Chaintreau et al in [31] claim that if the inter contact time distribution has the exponent, alpha <1, then the expected latency for any type of forwarding algorithm will be infinite. This can be understand easily in the light of the fact that a small value of exponent means slow decay of inter contact time. This means that delay experienced by messages will decay slowly or rise faster as time tends to infinity. Hence, a faster decreasing alpha would be in our favor. We verify this claim by plotting message latency for each message delivered using epidemic routing as flooding is known to incur the least delay of all the forwarding mechanisms. This can be verified as follows from Figure 5.3 using the parameters given in Table 5.1, which reveals that as the simulation time is increased latency grows at an increasingly faster rate:

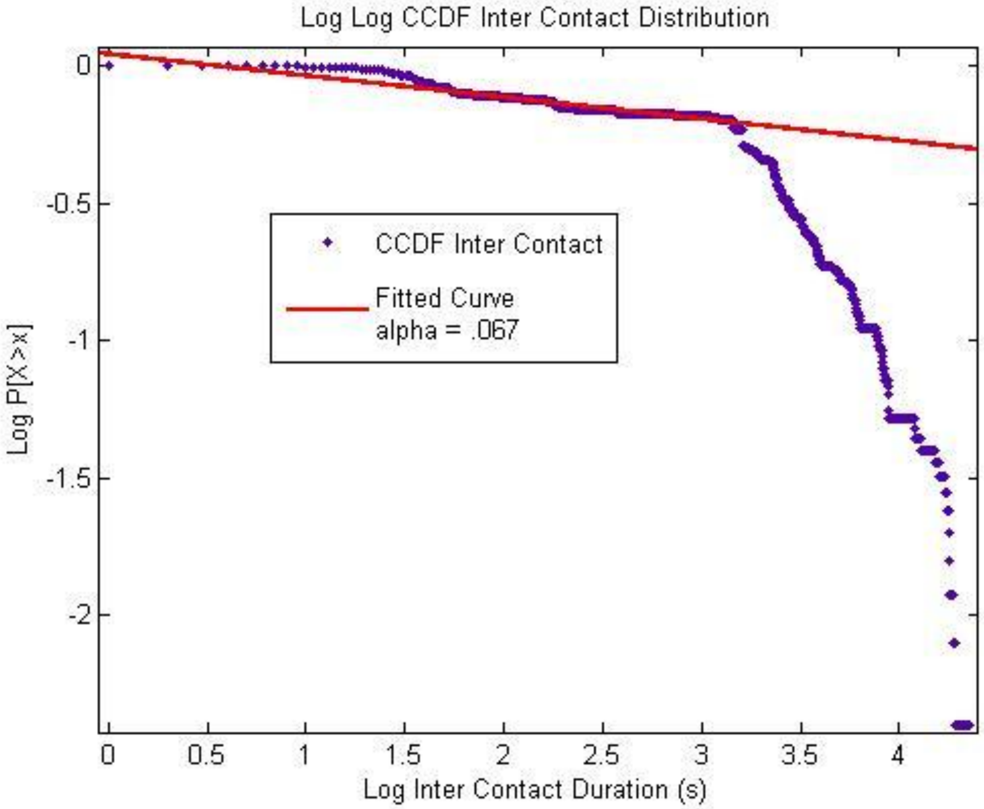


Figure 5.2 Log-Log CCDF of Inter Contact Distribution of Regular Ferry Network

5.4.1 Implications of Fixed Trajectory Ferries for Network Performance

In periodic cyclic deterministic mobility, if the power exponent is less than 1, then no matter how many additional ferry nodes are utilized the expected latency as time grows to infinity would still remain infinite. This is because the routes, path lengths, node speeds and pause durations remain constant over time and give rise to the same inter contact duration (ICTD) distribution. Intuitively, increasing number of nodes with the same mobility has yields a power law decay that does not change with increase in number of ferries following the same mobility pattern.

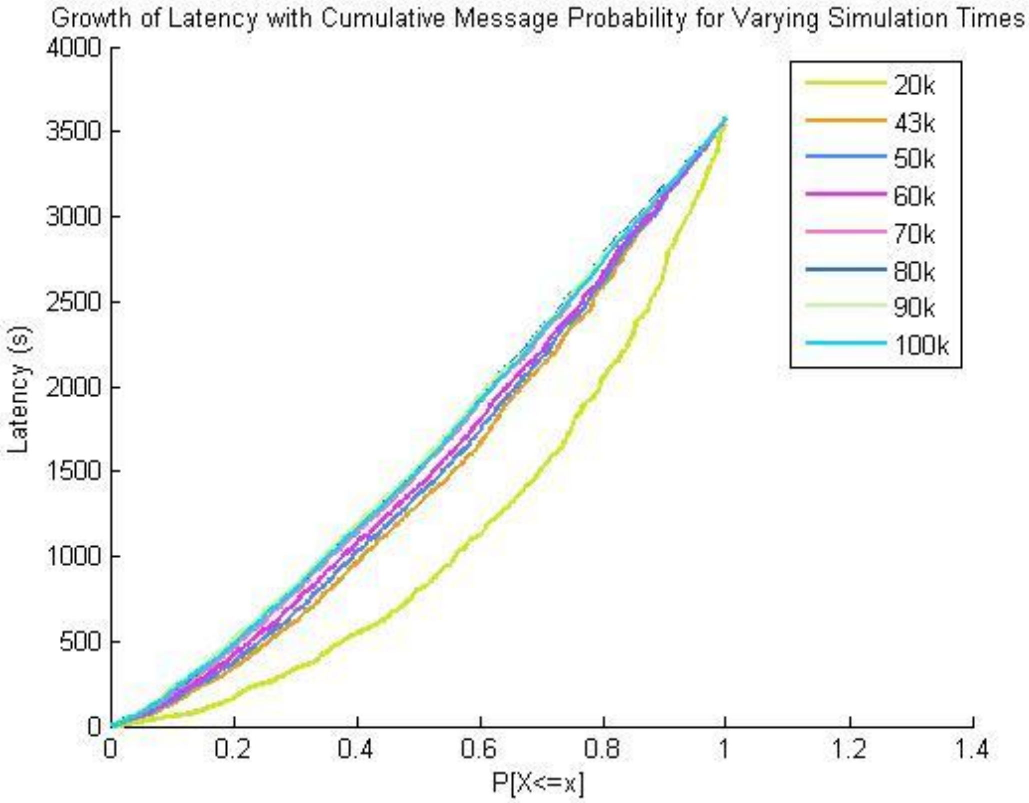


Figure 5.3 Growth of latency with Varying Simulation Duration

Table 5.1 Network Parameters for Regular Ferry Network

Serial No.	**Parameter**	**Value**
1	Message TTL	60 min
2	Message Size	2KB
3	Message generation interval per user	1s
4	Transmit Speed	200KBps
5	Bus Transmit range	60m
6	Traveler Transmit range	60m
7	Buffer size	1000MB
8	Bus Speed	7 m/s
9	Number of buses	8
10	Bus Pause Duration	120s
11	Simulation Area	8x10 sq-km

5.5 Ferry ICTD displays Power Law and Exponential Dichotomy found in Most Mobility Models

Karriangias et al in [3] and Chaintreau et al in [31] present analysis of popular mobility models such as InfoComm ’05, InfoComm ’06, Cambridge, UCSD, Dartmouth and vehicular traces that exhibit a power law decay up to certain values of inter contact duration. However, after these values, the power law behavior starts approximating towards an exponential decay. This dichotomy can also be observed in the ferry movement model that we have used. It is seen in Figure 5.2 that the inter contact distribution has power law decay up to 1000 seconds after which the ICTD decays exponentially.

5.5.1 Explanation of Power Law and Exponential Dichotomy in Ferry Network

This can be understood by considering the fact that ferries are traversing routes that are overlapping at various points in the simulation map. The points at which these routes intersect are not distributed uniformly over the simulation space. Hence, areas of dense route network would give rise to a high frequency of contacts that would have shorter duration between them. This explains the power law decay observed during the early part of the ICTD curve. On the other hand, where there are a small number of routes intersecting at a only a few points, contacts between the buses catering to these routes are rare and occur after large time intervals. The later fact gives rise to the exponential decay observed.

5.6 Heterogeneous Ferry Mobility

In the deterministic scenario considered above, ferries move on pre determined routes with fixed speeds or speeds that vary only by 1 m/s. This was done in order to ensure that the scenario resembles one where ferry mobility is restricted and cannot vary as we assume that the ferries are installed there for another purpose such as public transportation. However, it would be interesting to see if there would be any change in the network performance if ferry nodes that are more agile and have higher speeds can help improve the latency.

It is to be noted that the plots were obtained for the scenario in which we evaluate the delay expected by messages as they are routed amongst the ferries. We plotted the results after varying parameters as given in the Table 5.2 below:

Table 5.2 Network Parameters for Heterogeneous Ferry Network

Serial No.	Parameter	Value
1	Message TTL	60 min
2	Message Size	2KB
3	Message generation interval per user	200s
4	Transmit Speed	200KBps
5	Bus Transmit range	60m
6	Buffer size	1000MB
7	Bus Speed	7 m/s
8	Bus Pause Duration	120s

5.6.1 Introduction of Overlay Ferries to increase heterogeneity in ferry mobility

Ferries can be differentiated mainly on the basis of either the mobility model or the routing mechanism used. We observe that nodes that are more agile and have a higher speed should be able to cover a larger area in lesser amount of time as compared to slower nodes. Choosing faster nodes to disseminate data amongst ferries separated by a large inter contact duration can reduce the latency that would other be incurred. Further, using Direct Delivery Routing for overlay nodes would incur less overhead as compared to what would be incurred if only a ferry network with epidemic routing was used. Using the parameter configuration Table 5.2, we plot the latency incurred vs. messages generated for 10 and 12 nodes having 2 and 4 overlay ferries respectively. The plots are compared with those scenarios using a ferry only network of 10 and 12 nodes.

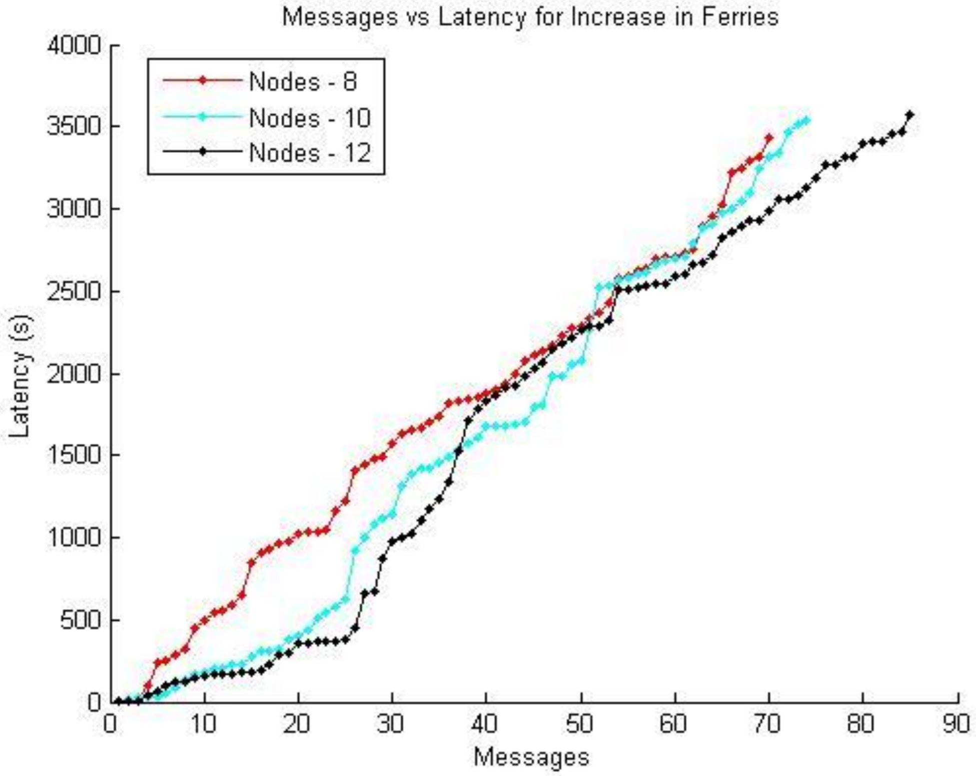


Figure 5.4 Growth in Latency for Varying Number of Regular Ferry nodes

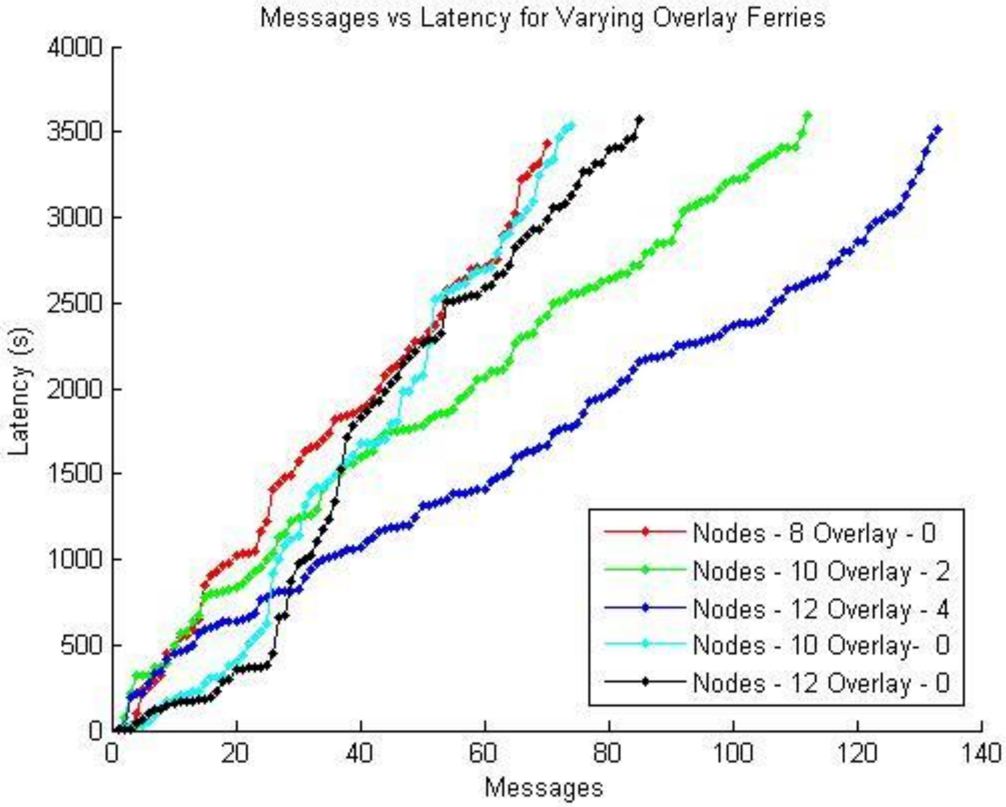


Figure 5.5 Latency Increases Slowly with Overlay Nodes Compared to Equal Number of Regular Ferry Nodes

5.7 Migration from Power Law To Exponential Curve

5.7.1 Heterogeneity of nodes causes Disappearance of Power Law

If nodes' speed in the system is uniformly spread across a sufficiently wide range of values, the duration between two contacts will then tend to be more spread out. In other words, when a small number of nodes take on value of speed from *[min max],* such that |*min-max*|<< *min*, then since the range of speeds available to the node is small, the time elapsed between two contacts repeatedly takes on a value in a narrower range that gives rise to power law distribution of inter contacts.

On the other hand, if nodes have a wider choice of speeds to uptake from, the probability a node will meet another node in a given time slot is more spread to a larger range. Hence, we observe the disappearance of power law and instead appearance of exponential, see Figure 5.6.

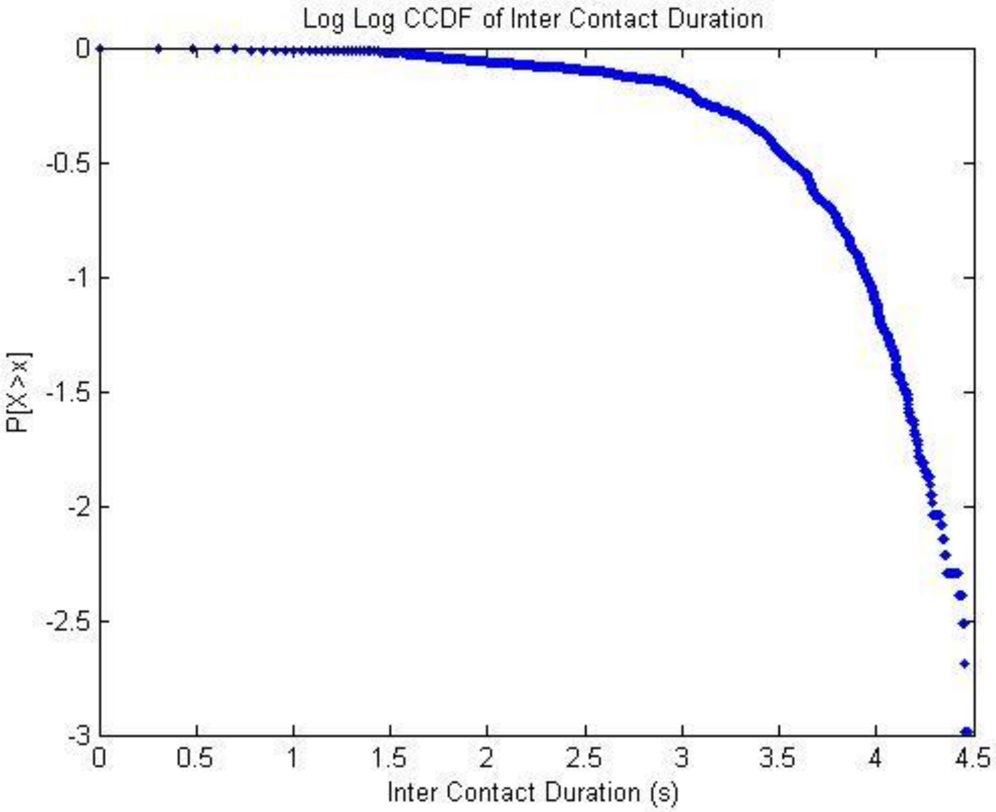


Figure 5.6 Log-Log CCDF Inter Contact Distribution of Regular Ferries and Overlay Ferries

5.8 Network Capacity

Finally we would like to determine the capacity of deterministic ferry mobility network. Network Capacity is simply the product of number of contacts, contact duration and bandwidth. We derived an expression for contact frequency earlier and showed that since the expected contact duration per hour is a constant, hence, the network capacity per hour is also a constant. This can be confirmed in Figure 5.7.

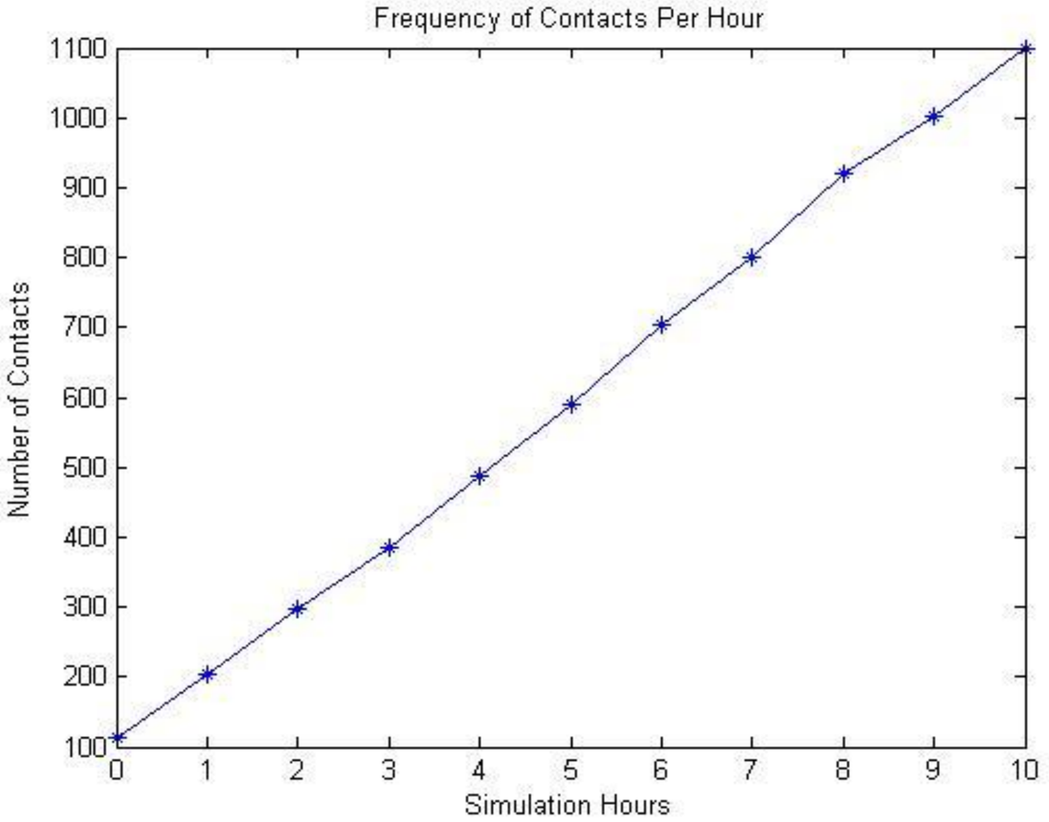


Figure 5.7 Constant per Hour Contact Frequency of 100 contacts per hour for a cyclic Ferry Network

5.9 Scheduled Based Routing

It would be interesting to see how well the ferry based network can serve communication needs of nodes randomly located across the network, incapable of relaying data on other's behalf possibly for want of privacy or insufficient battery power. We plot histograms of the delivery ratio, average latency and over head ratio at varying traveler group sizes to see how well the network scales.

5.9.1 Variation in Traveler Group Size

We plotted the results for variation in traveler group size versus the performance metrics. In the scenario characterized by network parameters, in Table 5.3, the system contains a single group of travelers. Therefore, size of the group is the size of traveler population in the system. Given below is tabulation of simulation parameters:

Table 5.3 Network Parameters for Scheduled Routing Protocol

Serial No.	**Parameter**	**Value**
1	Message TTL	2500 min
2	Message Size	100KB
3	Message generation interval per user	250s
4	Transmit Speed	200KBps
5	Bus Transmit range	200m
6	Traveler Transmit range	60m
7	Buffer size	100MB
8	Nr of traveler bus schedules	4
9	Bus Speed	15 m/s
10	Number of bus groups	3
11	Size of a bus group	2
12	Length of simulation	43000s
13	Bus Pause Duration	120s

5.9.1.1 Average Latency with Variation in Traveler Group Size

We can see from the histogram plot of Figure 5.8, that by varying the number of users, with each user generating a new message for dissemination every 250s, the average latency also varies. As the number of travelers increase in the system, so do the number of messages in the system. This means that buses, that are the only relay nodes, need to store more messages in the buffer. Limited buffer space causes some of the messages to be dropped. Further, since the contact duration between the bus and the user is small, transmitting increasingly more messages in the same amount of time results in abortion of messages. In addition to these two sources of increased overhead ratio, when a message is being transferred between two nodes, another node wanting to deliver or transfer message to one of these two nodes cannot do so for the duration of message transfer.

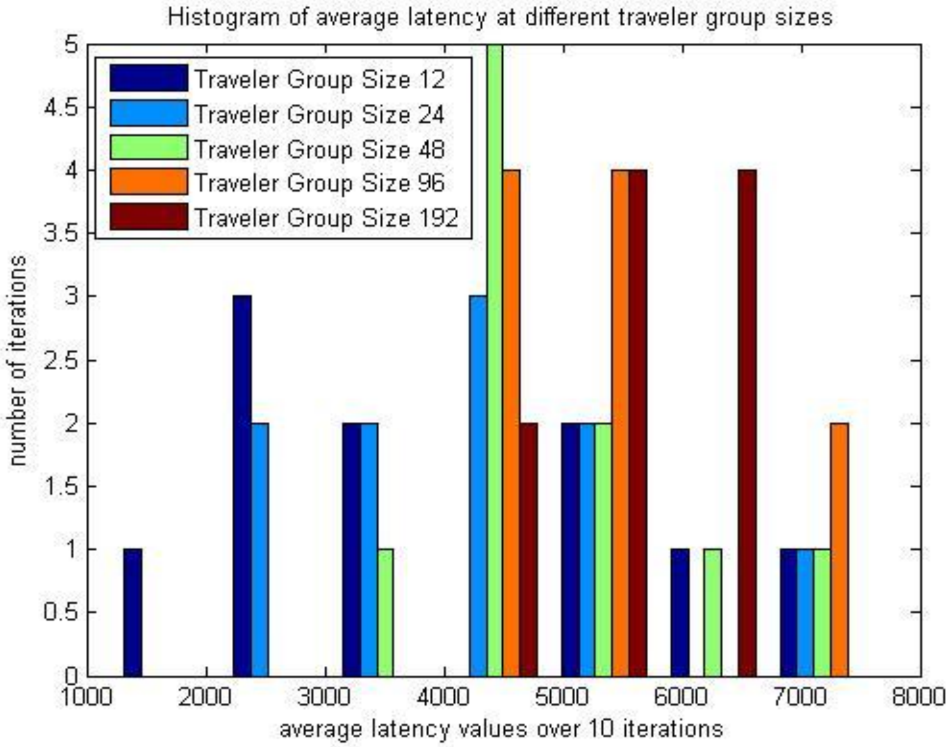


Figure 5.8 Average Latency for Different Number of Travelers in the System

5.9.1.2 Overhead Ratio with Variation in Traveler Group Size

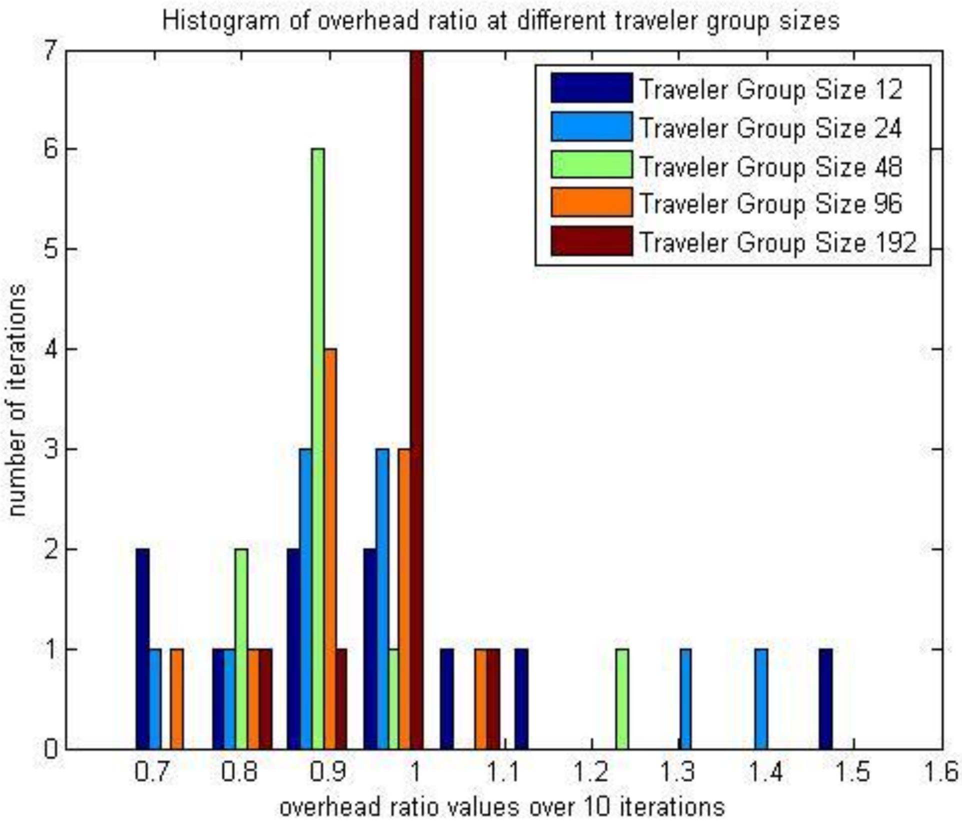


Figure 5.9 Overhead Ratio for Different Number of Travelers in the System

Increasing number of travelers gives rise to overhead ratio plot of Figure 5.9. We can see that as the number of traveler groups, and hence the number of travelers, increases variance of overhead ratio shrinks and the mean overhead ratio increases steadily. An increasing mean can be understood intuitively by the fact that as number of non ferry nodes increase, the number of bus-traveler interactions also rise. This rise in bus-traveler contacts increases the probability that a bus is already engaged in a transfer with some other traveler. Hence, the number of traveler-bus transfers that do not result in a delivery of the message increases the over head ratio. Further, when number of traveler nodes is small, the overhead ratio varies between a wide range of values. This range decreases with rising number of travelers because randomness in the message paths taken by a message to be routed to the destination decreases. In other words, more the probability that a message would be transferred via a popular ferry increases and hence the variance decreases.

Further, when number of traveler nodes is small, the overhead ratio varies between a wide range of values. This range decreases with rising number of travelers because randomness in the message paths taken by a message to be routed to the destination decreases. In other words, more the probability that a message would be transferred via a popular ferry increases and hence the variance decreases.

5.9.1.3 Delivery Ratio with Variation in Traveler Group Size

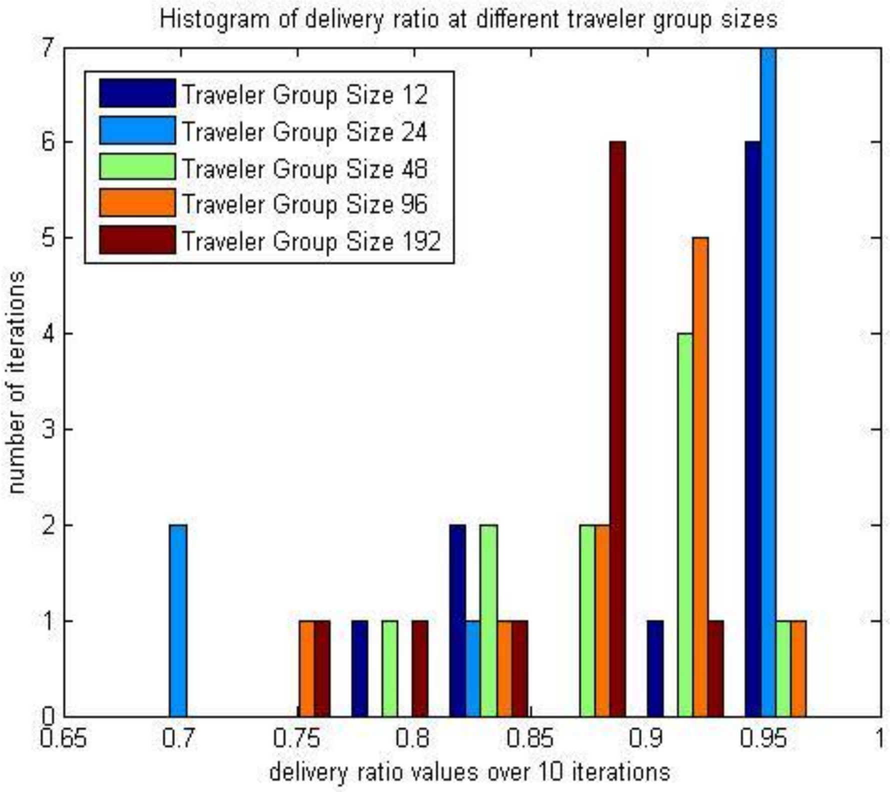


Figure 5.10 Delivery Ratio for Different Number of Travelers in the System

With rise in number of travelers and hence the number of messages created, the mean delivery ratio decreases as can be seen in Figure 5.10. This can be explained in light of the fact that as the number of messages created per traveler increases, the overall system message generation rate also rises. This leads to a higher probability of either a ferry node or a traveler node engaged in a transfer and hence unavailable for message delivery.

CHAPTER 6

CONCLUSION AND FUTURE WORK

We analyzed ferry based opportunistic message dissemination networks using the mobility model traces of inter contact duration and contact frequency. Inter contact duration distribution lent an insight into how the network performance would be affected if number of ferries were increased. Specifically, we modeled network contact frequency and empirically showed that network capacity is constant per hour. Further, we showed that introducing mobility and routing heterogeneity in the form of overlay ferries yields lower latency than the same number of regular ferry nodes. We confirmed the presence of power law exponential dichotomy and showed that increase in heterogeneity on top of the existing ferry movement causes gradual disappearance of power law decay.

Possible extension to our work would be the use of real bus mobility traces [5] to see how well traces used in our scenario having real world characteristics were able to predict network performance of a test bed. Further, a distributed profile based proactive routing in which regular ferry nodes adapt themselves to become overlay nodes based on their state such as speed of movement and versatility of areas visited.

APPENDIX A

OPPORTUNISTIC NETWORK ENVIRONMENT API CLASSES USED FOR SIMULATION

A.1 Network Layer

Each host in the system contains a network layer that has a unique network address. This layer is responsible for:

1. Checking whether the other host is within this host’s transmit range
2. Setting up a connection between two proximate hosts
3. Destroying connections
4. Taking connections down
5. Removing inactive connections

A.2 Bus Control System

A bus control system (BCS) is responsible for keeping track of which buses and travelers belong to which control group. In other words, each bus and each traveler belongs to a bus control system specified by a unique ID. A BCS is assigned a route by the map route movement model class that has a number of bus stops at which all the buses would stop at different points in time. The purpose of a BCS is to inform its own travelers if a bus belonging to this system reaches a bus stop. Only the travelers present at this particular stop at which a bus has stopped get informed of this event. A traveler can then decide whether to board the bus or not.

A.3 Bus Movement Model

All buses implement this movement model and hold a unique ID by which the control system indentifies them. The `DTNHost` class queries Bus Movement instance periodically to get the next path the bus must travel by invoking Bus Movement’s `getPath()` method. Each bus instance upon being created in the system must register with the control system. As soon as a bus reaches one of the stops belonging to the control system, it calls the `busHasStopped(id,path)` method of the BCS, supplying the next path it would travel and the Bus ID. Buses travel only the routes assigned to the bus control system of which they are a part. In this particular scenario, a route consists of a set of city main roads

Each bus implements the '`enterBus(Path p)`' method in which it assigned the path that the calling bus will travel after its waiting time at the current bus stop gets over.

A.4 Routing modules

A separate package comprising of the routing models is part of the ONE. All routing classes extend Active Router class. We use Epidemic Router for regular ferry nodes to evaluate cyclic ferry network and Direct Delivery for overlay ferries. We use Active Router class for scheduled routing discussed in Secion 4.3 to implement our Bus Scheduled Router (or Bus Router) and Traveler Scheduled Router classes.

A.5 ActiveRouter

We extend the capabilities of the `ActiveRouter` class, and describe here briefly the '`update()`' command that is overridden in our routing classes in order to elucidate on how our routing protocols operate. Further, the working of protocol will shed light on how and why the performance metrics change the way they do.

A.5.1 ActiveRouter's `update()` function

The `update()` function is called once every update interval by the `World` class. It monitors the list of sending connections from the particular node in which the routing layer sits. Every time a node goes out of range of the receiver node with which it was connected, the connection is aborted and if there was a message en route, that too is aborted. This affects the overhead and the delivery ratio adversely. On the other hand, those connections that successfully transferred a message are removed from the list of active connections. Those messages TTL of which has become zero or less are dropped. This again lowers the delivery probability and may increase the overhead ratio.

A.6 Message Create Event

All the events in ONE whether internal or external are processed via event queues. When a message is created it is put in the message buffer of a randomly chosen source traveler node. When the message due to be processed via the `processEvent(ExternalEvent ee)` method, schedule entry list of the destination node is added to the message as an additional property. Schedule enables the buses to make routing decisions based on this deterministic knowledge of an event that will take place in the future.

REFERENCES

1. http://www.haggleproject.org/index.php/Main_Page
2. Forouzan, Behrouz A. (2003). *TCP/IP Protocol Suite* (2nd ed.). McGraw-Hill.
3. "Power Law and Exponential Decay of Intercontact Times between Mobile Devices", Thomas Karagiannis, Jean-Yves Le Boudec, Milan Vojnovi´c, MobiCom'07, pp. 183-195.
4. "Opportunistic Networking: Data Forwarding in Disconnected Mobility Ad hoc Networks" L. Pelusi, A. Passarella, M. Conti. IEEE Communications Magazine, Vol. 44, No. 11. (November 2006), pp. 134-141.
5. http://traces.cs.umass.edu/
6. "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks", Wenrui Zhao, Mostafa Ammar, Ellen Zegura, MobiHoc '04.
7. "Integrated Buffer and Route Management in a DTN with Message Ferry", Mooi Choo Chuah, Wen-Bin Ma, Journal of Information Science and Engineering, 2007.
8. "ONE Simulator for DTN Protocol Evaluation", Ari Keränen, Jörg Ott, Teemu Kärkkäinen, SIMUTools '09.
9. http://www.firstmilesolutions.com/
10. A. Pentland, R. Fletcher, A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations", IEEE Computer Society, 2004.

11. “Performance Evaluation of Node Density-Based Adaptive Routing Scheme for Disruption Tolerant Networks”, Mooi-Choo Chuah, Peng Yang, Int. J. of Ad Hoc and Ubiquitous Computing, 2006.

12. http://watwire.uwaterloo.ca/DTN/sim/

13. “Scalable Routing in Cyclic Mobile Networks”, Cong Liu, Jie Wu, IEEE Transactions on Parallel and Distributed Systems, Volume 20 , Issue 9, September 2009, pp. 1325-1338.

14. “Opportunistic Scheduling of a Message Ferry in Sensor Networks”, Veeraruna Kavitha, Eitan Altman, MobiOpp ’10.

15. “Performance Evaluations of Various Message Ferry Scheduling Schemes with Two Traffic Classes”, Mooi Choo Chuah, Peng Yang, CCNC ’07.

16. “Hybrid Routing in Clustered DTNs with Message Ferrying”, Yang Chen, Wenrui Zhao, Mostafa Ammar, Ellen Zegura, MobiOpp’07.

17. “Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network”, Wenrui Zhao, Mostafa Ammar, Ellen Zegura, IEEE InfoComm ’05.

18. “Message Ferrying: Proactive Routing in Highly-partitioned Wireless Ad HocNetworks”, Wenrui Zhao, Mostafa H. Ammar, IEEE Workshop on Future Trends of Distributed Computing Systems ’03.

19. “A Message Ferrrying Scheme for Differentiated Services”, Mooi Choo Chua, and Peng Yang, IEEE MilCom ’05.

20. “Message Ferrying for Constrained Scenarios”, R. Viswanathan, J. Li, M. Chuah, WoWMoM, ’05.

21. “Queueing in Space: Design of Message Ferry Routes in Sensor Networks”, Veeraruna Kavitha, Eitan Altman.

22. “Message Ferry Route Design for Sparse Ad hoc Networks with Mobile Nodes”, Muhammad Mukarram, Mostafa Ammar, Ellen Zegura, MobiHoc ’06.

23. “Energy Efficient Computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet”, Proceedings of International conference on architectural support for programming languages and operating systems (ASPLOS) ’02.

24. “Cluster-based Forwarding in Delay Tolerant Public Transport Networks”, Shabbir Ahmed, Salil S. Kanhere, 32nd IEEE Conference on Local Computer Networks 2007.

25. “Epidemic Routing for Partially Connected Ad hoc Networks”, A. Vahdat and D. Becker, 2000.

26. A Delay-Tolerant Network Architecture for Challenged Internets, K. Fall, SIGCOMM, August 2003.

27. IETF-65 DTNRG group meeting proceedings, Dallas, TX, March 19-24, 2006. https://www.ietf.org/proceedings/06mar/DTNRG.html.

28. I.F. Akyildiz et al, “Computer Networks”, 43 (2003) pp. 75–112.

29. T. Small and Z. J. Haas, “The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way*)”,Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, Annapolis, MD, USA, June 1-3, 2003.

30. W. Zhao, Y. Chen, M. H. Ammar, M. Corner, B. N. Levine, and E. Zegura, “Capacity Enhancement using Throwboxes in DTNs”, In Proc. IEEE Intl Conf on Mobile Ad hoc and Sensor Systems (MASS), Oct 2006.

31. “Real World Mobility Traces and its consequences for opportunistic forwarding”, Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, James Scott, Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory, February 2005.

BIOGRAPHICAL INFORMATION

Samreen Tahir was born March 1982 in India. She received her Bachelor of Technology in Computer Science and Engineering from College of Engineering Roorkee, India in June 2006. Prior to pursuing her Master's she was a Guest Faculty at Women's Polytechnic, Aligarh Muslim University, India for a period of 8 months. In Fall of 2007, she started her graduate studies in computer science. She received her Master's in Computer Science from the University of Texas at Arlington, in May 2010. Her research interests include computer networks and opportunistic and delay tolerant networks.