THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Augur: A Delay Aware Forwarding Protocol for Delay-Tolerant Networks

Ahmad El Shoghri

BE in Computer and Communication Engineering,
The American University of Beirut, LEBANON

A thesis submitted for the degree of Master of Philosophy at

The University of Queensland in 2016

School of Information Technology and Electrical Engineering

Abstract

Real life applications of Delay-tolerant Networks (DTNs) are evolving rapidly in many medical, environmental and engineering fields. DTN supports networking where the existence of contemporaneous communication links between the nodes is not guaranteed. In fact, DTN nodes are usually deployed in extreme terrestrial or spatial environments that lack continuous network connectivity. The nodes can range from sophisticated devices with abundant resources such as those used in interplanetary networks to small devices that are very limited in resources such as those operating in wireless sensor networks. Moreover, the mobility of the nodes and other disruption factors that may be present in the field cause frequent disconnections and nodes isolations leading to an intermittent connectivity.

An efficient DTN routing protocol should have two main characteristics: high delivery probability and low delivery delay. Many routing algorithms have been proposed in various DTN applications. The majority of these protocols optimize delivery probability whereas very few of these algorithms address low delivery delay. Specifically, no previous work has produced a delay aware protocol to route data in DTN scenarios using historical spatiotemporal data of mobile nodes.

Spatio-temporal information and encounter statistics provide useful measures to understand a node's mobility. The time dependent behaviour of a mobile node and its periodic reappearances at specific locations around similar times for similar durations can predict future presence of that node. This characteristic can assist the performance of a routing protocol by estimating the time, location and duration of possible upcoming transfer opportunities.

This thesis addresses the delay issue in DTN by studying the effect of including the spatial and temporal dimensions of mobility in the decision metric of DTN protocols. "Augur" a new delay aware routing protocol for DTNs is introduced. In particular, Augur is targeted to optimize and minimize message delivery delay based on historical spatiotemporal behaviour of the participating nodes.

Two versions of Augur will be presented in this work namely Augur Temporal and Augur SpatioTemporal. The two versions differ in the amount of historical information which is

used regarding the movement characteristics of the mobile nodes. Augur Temporal uses only the temporal dimension to build its decision metric. This protocol is targeted to nodes having highly repetitive and periodic movements every day. The SpatioTemporal version makes use of both space and time dimensions to suit high and low periodicity scenarios at the expense of more processing and storage.

This work investigates the performance of the proposed Augur algorithms in an application related to mobile ad hoc networks. It compares their performance to the state of the art DTN protocols using the same set of parameters; e.g., number of nodes, load, buffer size and movement model. The implemented scenario considers a Helsinki city map in which bus nodes move along predefined trajectories. Messages are generated by every bus and they need to be routed to static sinks present at some bus stations. The work explored the performance results for different values of input parameters, mainly the message generation rate and node speeds. Two sets of experiments were conducted; in the first the bus nodes travel at constant speed during the simulation to maintain highly repetitive bus movement while in the second set of experiments variable speeds are used. In the constant velocity scenario, the bus follows exactly the same route at exactly the same time each day, and movement is very predictable. In the variable speed scenario, the bus travels the same route, but at a randomly variable speed within a certain range, so the bus position at any given time is less predictable. The scenarios were implemented in ONE simulator which is specifically designed to evaluate DTN routing protocols and assess their performance in various applications.

This thesis presents key findings through comparative evaluations and extensive simulation studies. Specifically, spatiotemporal information of DTN nodes improves the network performance when this information is incorporated in the design of the routing protocol. Augur outperforms the state of the art DTN protocols in terms of delivery probability, overhead ratio and latency in the implemented scenario. It is found that at low traffic rates Augur reduces the overhead ratio by up to 94%, and by up to 88% at high traffic. It is also observed that the improvement in latency was reduced by up to half over the existing protocols in both traffic rates while still improving the delivery probability of messages. Lastly, this thesis found that reducing delivery delays indirectly improves the delivery probability. When the protocol ensures routing the messages in shortest possible time it gives the opportunity to other messages to be delivered before their expiry or loss.

It is expected that spatiotemporal based protocols like Augur excel in cases where the movements of the users show repetitive patterns.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

Conference paper:

A. El Shoghri, B. Kusy, R. Jurdak, N. Bergmann, "Augur: A Delay Aware Forwarding Protocol for Delay-tolerant Networks," *In Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp.460-467, 2015.

Journal paper (Not related to this work):

D. Rozado, **A. El Shoghri,** R. Jurdak**,** "Gaze Dependant Prefetching of Web Content to Increase Speed and Comfort of Web Browsing**,"** *International Journal of Human-Computer Studies**,*** 78:31-42**,** 2015

Publications included in this thesis

No publications included.

Contributions by others to the thesis

No contributions by others.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgements

I would like to thank all the people who provided their encouragement and support during my MPhil study.

In particular, I would like to send immeasurable appreciation and sincere gratitude to my principle supervisor, Prof. Neil Bergmann (UQ), my co-supervisor, Prof. Raja Jurdak (CSIRO) and Dr. Branislav Kusy (CSIRO) for their continuous support, patience, sound advice and immense knowledge. Their guidance helped me during the years of study to achieve all the research milestones and I could not have imagined having better advisors and mentors for my MPhil study.

Last but not the least, I would like to express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

This accomplishment would not have been possible without all of you. Thank you!

Keywords

Mobile ad-hoc network, delay tolerant network, spatiotemporal data, opportunistic network, opportunistic network environment ONE, network simulation

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080502, Mobile Technologies, 50%

ANZSRC code: 080503, Networking and Communications, 50%

Fields of Research (FoR) Classification

FoR code: 0805, Distributed Computing, 100%

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Wireless Sensor Networks

Wireless Sensor Networks (WSN) are a connected network of environmental sensors. Wireless communication technologies such as IEEE 802.11, Bluetooth and other low power radios are key components of mobile sensing and computing. The last decade has witnessed a wide deployment of various sensor networks in different areas of science and industry. In fact, the availability of small, low-cost, reliable sensors, micro-controllers and low cost radios has created the opportunity to equip almost every machine or device with sensing and processing capabilities – the so-called Internet of Things. This has opened up a wide range of exciting applications related to large-scale environment surveillance, animal tracking, public security and several other areas that benefit society.

1.2 Delay Tolerant Networks

1.2.1 Overview

In many cases, wireless sensor networks are not compatible with the current Internet's underlying "always-on" network assumptions. In reality, the deployment of the wireless sensor networks in areas that lack pervasive network infrastructure, including extreme environments like deep space, deep oceans, volcanic regions, battlefields and developing regions creates challenges and limitations for communications. Links can be obstructed repeatedly by intervening moving or static objects in addition to periodic shut downs for energy conservation. Physical obstacles such as buildings, houses, vehicles or environmental obstacles such as water and mountains accompanied with a limited radio coverage from the nodes can significantly affect the performance of a network. Moreover, the high mobility of the wireless sensor nodes that form the network lead to an extremely dynamic mobile network topology in which the contact durations between the nodes are short. This means that a quick forwarding decision has to be made and also prioritized messages have to be considered. All these challenges create frequent disruptions and disconnections among the nodes leading to an intermittently connected network.

Conventional network protocols such as TCP/IP fail where frequent discontinuities exist along the journey of a message or a packet from source to destination. A basic assumption of the traditional networks is to have a fully connected path between the sender and the receiver while the data is sent. If this requirement between the two endpoints is violated, data cannot be delivered.

New protocols are needed to solve the issues of networks with intermittent connectivity or what is called Intermittently Connected Networks (ICNs). ICNs are networks where a path from a source to a destination is only sometimes available. To overcome these problems, Delay Tolerant Networks (DTN) have been proposed where there is no guarantee that an end-to-end link between any two nodes exists. DTN assumes that no end to end path between the nodes is available at the time when the message is created but rather links will appear over time. In this type of network, data is incrementally moved hop by hop every time a link is available until data ultimately reaches its destination. Figure 1 shows a scenario of how an intermittently connected network evolved through time where a direct end to end link between the source and the destination never existed. In this scenario, the source node S benefited from the physical movement of the surrounding nodes to deliver its message to destination node D. DTN solves the technical issues of partitioned networks, lack of continuous connections and limited network resources at the expense of delivery delays and packet loss.

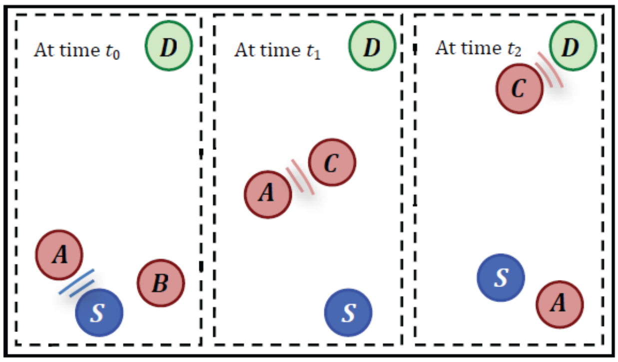


Figure 1. Illustrative example of a time evolving DTN.

In DTN, the nodes present in the network may be randomly scattered in the field and they typically switch between three types of functionalities. A source node initiates data transmission. An intermediate node or relay helps in forwarding the message to other nodes in the network. Often, this node carries the data and waits for potential links to develop. The destination node can be the final target of the data, or it can be an access point for sending the data to the wider Internet. In general, the purpose of these networks is to gather and route data through wireless nodes dispersed in the field and transmit them to specific destinations for future use.

1.2.2 Potential DTN Applications

DTN networks are useful in sensing and routing notifications related to emergencies, weather updates, safety and traffic conditions [1]. In addition, they can serve in routing advertisements and marketing data sent from companies to their end users and vice versa to improve the quality of service through continuous feedback. Furthermore, mobile wireless networks can be deployed to collect data from distributed sensing units (sensors) that monitor environmental conditions such as climate change, air quality, and pollution level [2]. Mobile networks can also be exploited to afford Internet connectivity to rural, undeveloped or uncovered areas and also enable various other non-real time services. Examples of these services could be sending and receiving electronic mail, file sharing and cached web access through vehicle ad hoc networks (such as buses, taxies, trains, trams and ferries). Furthermore, even in the usual presence of good infrastructure covering the region, nodes may face communications difficulties during disasters and emergencies. Natural disasters or military conflicts may render conventional communication in the region impossible [3]. During these disasters the infrastructure providing the link between the devices and the network will be unavailable. DTN with an ad-hoc feature can replace traditional networks to successfully send notifications and forward valuable information to rescue personnel in the area. Another motivating application lies in the area of Smart Cities where a huge number of personal devices, moving vehicles and distributed sensors coexist. Mobile phones and smart watches can act as mobile routers since they possess sufficient battery lifetime, radio range, buffer capacity and processing power. These devices can interact with each other to create an intermittent dynamic network and participate in exchanging data in the network to provide a myriad of services to city dwellers [4].

1.2.3 Benefits of DTN

In many challenging network environments there are no guarantees of contemporaneous connections between the nodes in the network. However, the nodes should interact to support data flows to sources and destinations that may never have a single end-to-end connection. DTN can play an important role in creating a mobile opportunistic network composed of moving sensors on animals, vehicles and pedestrians. In fact, customer feedback, large scale environmental monitoring and many other applications do not require real-time data to be received or nodes to be uniformly distributed in the network. Weather monitoring, studying wildlife and other scientific and statistical analyses are based on sensor data that are collected over a long period of time. These types of applications can tolerate delays and missing data [3].

In other scenarios, social networks continue to expand rapidly to play an increasingly important role in society. Through social networks, people share their personal information, experiences, secrets, updates, memories and private stories with individuals or groups. Yet, the personal privacy of people is threatened since these applications are mostly web based applications serviced through a central site or database where the owner and other legitimate employees have access to all data. DTN networks can play an innovative role in these applications by providing more privacy to the users through the shift to a peer-to-peer infrastructure (enveloped with encryption). Using DTN in social networks will also remove the need for continuous Internet access through 3G or Wi-Fi connection which is expensive for mobile users. For these reasons, there is continued interest in DTN, and this thesis will investigate enhancements to existing DTN protocols.

1.3 Focus of the Thesis

As mentioned above there is motivation to investigate additional networking techniques that work under challenging conditions and enable the communication between source and destination without the support of a fixed network infrastructure. Delays and packets loss in DTN are unavoidable because it is an opportunistic network with no guarantees of timely packet delivery, however very long delays in some time critical scenarios limit the benefits that could be gained from this technology. Currently, there has been limited work on any explicit routing algorithm that specifically deals with minimizing delays in realistic scenarios, and this creates a research gap in this area. **The core of this thesis is to investigate new algorithms to deliver the DTN packets to their corresponding destinations with the minimum delay possible in dynamic topologies.** Studies have shown that many of the participating nodes in the particular DTN scenarios exhibit some periodic patterns in their daily movement. For instance, people commute with high regularity in terms of starting positions, final destinations, starting and arrival times and the routes they follow [5][6][7]. Studies have also shown the existence of periodic mobility patterns for a range of animals and birds for certain activities like daily foraging and yearly migration [2][8]. Another example is the scenario of bus movement in which a bus follows a specific timetable to visit different stops and stations. Sensors and routing devices associated with these actors inevitably demonstrate the same movement regularity. Therefore, if the periodicity characteristic is used to estimate in advance the future presence of nodes at certain locations or estimate the times at which they are reachable, significant improvements can be made to the performance of a DTN. Spatiotemporal information from these mobility patterns should be able to improve routing and forwarding decisions to more successfully deliver messages to their ultimate destinations. Interesting research questions now emerge. If a spatiotemporal method is realized to minimize delay in DTN, how sensitive will it be to different network topologies? What is the impact of the movement periodicity patterns on the performance of this method?

This research investigates how spatiotemporal data can be extracted from DTN nodes. Then, the availability of this information will be exploited to produce a new DTN algorithm, Augur which opportunistically routes messages to their destinations in the lowest possible delay times. In this work, two versions of Augur are developed, which differ in the amount of information used to build their decision metric. The protocols are named Augur Temporal and Augur SpatioTemporal which will be described in details in the following sections. The work will also analyse two aspects of Augur: its individual performance dynamics and how it compares to other protocols. First, the response of the new protocol is explored in relation to different simulation parameters, namely the message creation rate, the message time to live, the node buffer size and the node speed. Second, the performance of Augur is compared to well-known DTN routing protocols in terms of delivery probability, overhead ratio and latency. Two operational scenarios are investigated. In the first scenario, constant speeds for the nodes are considered to maintain highly repetitive movement. In the second, variable speeds are used to tackle a more realistic scenario and weaken the periodicity factor of the moving nodes. The performance evaluation results will show that Augur outperforms the state of the art DTN protocols in all the undertaken scenarios.

The remainder of the document is organized as follows. Section 2 gives background information about the field and presents some related work. In section 3 the research questions and methodology are provided. Section 4 describes the Augur routing protocols proposed in this work including the operation principles and algorithms. Section 5 focuses on the performance evaluation of Augur against other protocols through a set of experiments and also includes the all the results and discussions. Section 6 concludes the study and discusses potential future work.

2 Background

2.1 Overview

Traditional data networks are modelled using connected graphs whereby the existence of at least one end-to-end path between any source-destination pair is always guaranteed [4]. Indeed, a message sent from a mobile device to a specific destination goes through a well-designed network infrastructure consisting of communication towers, bridges and routers. In order to have Internet connectivity a device should be directly connected to at least one element of the infrastructure forming the network. If there is no connectivity when a message needs to be sent, data will typically be discarded or dropped by the node carrying the information.

For example, the widely used TCP protocol is a connection oriented protocol which means that source and destination sides have to establish end-to-end communications before their application processes can send data [4]. At first, the TCP protocol identifies the two end points that are involved using their corresponding IP addresses and port numbers and establishes a connection through a 3 way handshake procedure. In TCP/IP networks, data are routed through static routers and network links that are always available to provide paths to various destinations guaranteeing fast and reliable transmission. Characteristics between nodes like continuous connectivity, very low packet loss, stable general network topology and low propagation delay are implicitly assumed [9]. However, in DTN not all of these assumptions exist at the time when packet delivery is needed and sometimes none of them exist [4][9]. DTN especially addresses significant link delays and intermittently connected links in challenging network environments that cannot be served by conventional TCP/IP protocol models [10][11][12].

Wireless computing devices often use a Mobile ad hoc Network (MANET) which is an infrastructure-less self-configuring network that is effective in various networking application where the infrastructure is absent, unavailable or impractical. MANET technology is used in data monitoring applications because of its ability to facilitate the collection of sensor data originating from distributed sensors in fields such as air pollution analysis, wild life animal tracking and vehicle traffic monitoring [9][2][8]. However, when the sensors are placed on highly mobile devices the continuous change of the network topology is challenging. The objects' motion will cause frequent link changes and disruption in the network. Nodes might be interchanging data when they are within others' range and might be disconnected when no neighbours exist in their radio transmission range [13][14][15]. This type of network is known as an Intermittently Connected Network (ICN) [4][9]. In DTN, two or more nodes exchange messages only in “contact” phase which is the stage when the participating nodes are within the transmission ranges of each other and this phase lasts until this condition is broken [16][17][18].

DTN is becoming an effective approach used to deal with the technical issues found in environments that lack continuous communication links between nodes. DTN is a class of network characterized by its intermittent connectivity [19]. DTNs lack an instantaneous end-to-end path between source and destination resulting in long variable propagation delays. Under conditions of extreme delays, limited bandwidth, node mobility and recurring communication obstructions, nodes should cooperate to guarantee the delivery of their messages and to maximize their lifetime. Thus, DTN protocols should adopt store-carry-forward communication approaches through which data are incrementally moved and stored throughout the network so that data will eventually reach its destination [20][21][10].

A parallel line of research has focused on DTN using personal communication devices. These devices hold valuable mobility information that can be exploited to improve DTN performance. Nearly 3.3 billion people worldwide use cell phones. Mobile phones are integrated with GPS, Wi-Fi, Bluetooth which create many contact opportunities and large storage for message buffering. They are inexpensive and versatile tools [22]. Recent research used devices like laptops and phones to capture essential features of mobility for behaviour modeling purposes. Spatiotemporal mobility statistics provide realistic measures to understand a mobile user’s behavioural preferences and data transfer opportunities relevant to DTNs [23]. The time dependent behaviour of a node and its periodic reappearance around similar times at certain locations for specific durations can estimate future presence of the node and aid future routing decisions [23]. The users holding the devices have a high probability of meeting again at similar times for similar durations. Mobility models can heavily affect the performance of the DTN protocols in terms of delivery delay. Therefore incorporating spatiotemporal characteristics in DTN routing protocols are expected to improve DTN performance [22][23].

Recent studies of the mobility of staff and students on a campus equipped with PDAs or laptops able to be connected to wireless access networks, show that they follow common mobility patterns [7]. They show that significant aspects of the behaviour can be characterized by power-law distributions. Specifically, the session durations and the frequencies of the places visited by users follow power laws. This means that users typically visit a few access points frequently while visiting others rarely, and that users may stay at few locations for long periods while remaining at other locations for very short periods. The study also observed that 50% of users studied spent 62% of their time attached to a single access point and this proportion decreased exponentially. If these wireless access network studies are taken to represent a class of mobile node behaviour, it can be considered that these observations are applicable to at least certain DTN scenarios [7].

Rural and low income residents lag behind in Internet access especially in developing countries where people suffer from the lack of good infrastructure. The attempt to bridge all of these areas through the construction and deployment of new cellular towers and other communication systems fails due the very high cost of such projects [24][25]. Public transportation systems like buses, trains and trams hold sufficient power to support communication access points and they relatively run on scheduled routes. Several studies have used these factors to provide communication mechanisms to rural areas through the existing vehicular bus network [20][10][26]. For example, Postmanet and Maxprop are algorithms that aim to realize DTN routing in urban public transport system through postal systems and vehicle based systems respectively [10][20][26][27][28].

Before moving to the details of routing in delay tolerant networks, some of the DTN properties, network characteristics and protocol assessment metrics are presented.

2.2 Network Characteristics in DTN

2.2.1 Basic DTN Model

In general, a DTN model is composed of nodes having computing, sensing and processing abilities spread in the network. These nodes constitute the media responsible for carrying and transferring the data to specific destinations [11]. The objective of this network is to deliver data or “messages” to their corresponding destinations. A message M can be represented by a tuple (S, D, T, L) where S is the source node that initiated the message, D is the destination node of that message, T its time of creation and L its length or size in bytes. When two nodes are in the transmission radio range of each other a wireless communication link is formed, and this is also known as a “contact”. The availability of such a link between the nodes gives the opportunity to send data from one node to another. Note that more than two nodes can be in the contact range of each other, in this case multiple links are formed and multiple relaying options are available [24]. When a node loses contact with all other nodes, no data transfer is possible anymore and the node is called “isolated”. Links disconnect as the nodes move away over time or can be caused by other interference or disruption events. At each contact opportunity, the node has to make a decision on whether to forward one or more of its message to its neighbour node or to keep the messages in its buffer and wait for future events. Once a message is transferred, the “receiving node”, “relay node” or “intermediate node” buffers the message and waits until a next hop or a contact opportunity is available. Conceptually, the nodes will take decisions that increase the chance of message delivery and minimize the delivery delay as much as possible [29].

2.2.2 Delay Components

In DTN, a message faces several forms of delay during its journey from source to destination. These delays can be summarized by waiting delay, queuing delay, transmission delay and total delay [24]. The waiting delay is the amount of time the message is kept in each node’s buffer until being transferred to the next hop. The queuing time is the amount of time taken till the message reaches its turn to be relayed or delivered, and this happens when a set of messages need to be sent or when other messages are given higher priority. The transmission delay is the time taken for a message to be fully transferred from one node to another, and this delay is determined by the transmission speed and the size of the message being transferred. Finally, the total delay or total latency of a message is the amount of time that a message takes from the time of creation until the time of its delivery at the destination [29].

2.3 Network Challenges in DTN

Several challenges arise in delay tolerant networks. Many of them stem from the frequent disconnections and isolations that traditional networks cannot handle. Other problems are due to the limited network resources available since the network media consists of a wide range of devices like laptops, mobile phones, tablets and moving vehicles.

2.3.1 Buffer Size

In order to overcome the issues of intermittent connectivity, the nodes in DTN must store the messages in their buffer until reaching the destination or finding a better candidate to relay the message to. This waiting delay which is considered the longest among the other delay components may range from minutes to hours or even days. This means that the intermediate nodes require large buffer capacities to be able to handle the set of messages waiting to be relayed or delivered. Some routing strategies consider the available free space in the nodes' buffer upon making a forwarding decision, while other strategies do not include this resource in making their routing decisions [24]. In fact, it is difficult to find a buffer size that perfectly suits all possible applications since this resource is highly affected by the message generation rate. The available options are either a limited buffer that suits the considered scenario or assuming unlimited buffer size for all nodes. Note that an unlimited buffer does not always ensure packet delivery since delivery is also bounded by other factors like channel capacity, transmission speed and the duration of contact with the destination.

2.3.2 Contact Capacity

Another challenge that highly affects the performance of a DTN is the contact capacity which reflects the amount of data that can be exchanged by two nodes during the contact phase [24]. This is dependent on the duration of contact between the two nodes and the link technology being used. The longer the contact lasts the more data can be exchanged. However, in dynamic DTN where the network consists of mobile nodes, this duration is usually short and hence the contact capacity at each encounter will be limited [30]. This factor can dramatically affect the performance of the network especially if the volume of traffic being exchanged is relatively small compared to the capacity of contacts in the network or when the size of messages is large [24][29]. It is noted from the previous work in the field that very few of the DTN routing protocols use this information in building their routing decisions although several researchers have studied the contact duration factor in real world applications. Therefore, including the contact duration of the nodes in the routing strategy as part of the network topology will definitely help in achieving more accurate and efficient relaying decisions.

2.3.3 Mobility

In dynamic topologies, nodes are mobile and can exhibit various mobility patterns. Nodes' mobility is an important factor in DTN, and it is highly dependent on the application under consideration. Participating nodes can range from static elements to moving elements and also from constant speed to variable speeds with different irregularities in their movement. In fact, highly dynamic environments cause frequent disconnections and short transmission times [24][31][32]. Many routing protocols exist nowadays in DTN, and they address the various types of mobility seen in different applications. These mobility traces can be classified in categories based on how predictable they are [24][33]. Figure 2 shows a spectrum of mobility examples moving from very precise schedules to completely random ones.

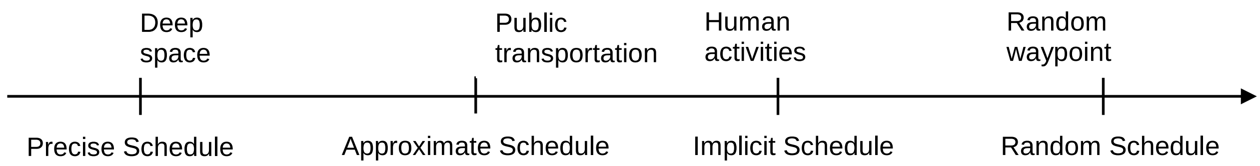


Figure 2. A spectrum of mobility predictability

For instance, interplanetary applications and deep space network disruptions are caused by moving objects but their movement can be precisely calculated. This highly predictable schedule of disruptions as well as connections helps significantly in performing effective routing through precise contact schedules. In other examples such as buses and human movements, overall journeys may be regular but the starting and ending times may vary. Their mobility schedule is not precise due to variable traffic conditions. These activities have implicit schedules such as bus departures and arrivals and human commuting to work or shopping. There is no guarantee when the arrival time of a bus is or when a person arrives at work but their schedule of activities is fairly regular [24][33][31][1]. Therefore, mobility and regular patterns can be exploited to improve DTN routing decisions. Studies have also investigated DTN in scenarios where the mobility is proactive [31][34]. In this type of mobility the participating nodes move in response to communication needs, thus the movements are considered semi predictable. However, this type of controlled mobility is outside the scope of this research.

2.3.4 Processing Power and Energy

Delay tolerant networks make use of various devices attached to people, animals or vehicles to gather data and route it to specific destinations. Normally these devices are small in size and have limited processing power. Consequently, DTN nodes are not able to run complex routing algorithms at each contact especially in the presence of short contact durations [35]. Energy consumption also constitutes a significant challenge in DTNs since processing tasks in addition to data transmission like sending and receiving messages require the nodes to consume more energy [36]. Studies in DTN and wireless sensor networks have extensively studied the issue of low processing power and energy consumption. Other researchers in our group have explicitly investigated DTN forwarding decisions based on available energy [37]. However, the proposed Augur protocol in this thesis is designed for scenarios such as public transport nodes which are not energy limited. Augur is not designed for extremely small devices and hence the processing power and energy consumption aspects of the performance are not assessed in this thesis.

2.4 Performance Metrics

This section presents the common evaluation metrics used to assess a DTN routing protocol and compare its performance to other routing strategies. In all the experiments considered in this work, the DTN routing protocols are analyzed and evaluated using the following quantities:

- Created messages (CM): is the total number of messages created by all the nodes during the experimental period.
- Number of forwards (NF): is the total number of successfully transmitted messages between any two nodes in the network. It is the sum of transmissions done for each created message, and at every message relay the count is incremented by one. For example, if a message goes from node A to B and then B to C this is considered as 2 forwards.
- Delivered messages (DM): is the total number of unique messages successfully delivered to their destination.
- Delivery probability (DP): is the fraction of created messages that has been received correctly at the destinations within the simulation time period. The delivery probability is one of the most important metrics in DTN [19].

$$DP = \frac{DM}{CM}$$

- Overhead ratio (OR): shows the number of transmitted messages which are not the final message to the destination, compared to the number of transmitted messages which are final messages to the destination [19]. For example, if all messages are delivered in two hops then the OR is 100%. This metric estimates the extra transmissions needed by the routing protocol for the actual delivery of the message and it is also considered as an approximation of the required computational resources and energy consumption [19][24].

$$OR = \frac{NF - DM}{DM}$$

- Average latency (AL): gives the average of the times taken by the delivered messages from their creation to their first delivery at the destination [19].

$$AL = \frac{\sum_{n=1}^{DM}(delivery\ time\ of\ Mn - creation\ time\ of\ Mn)}{DM}$$

- Channel Utilization (CU): gives the percentage of time the link between a node and a destination is used for data packet transmission during the total contact time [38]. In the common case where all messages from all nodes are sent to single data sink, such as an Internet access point, then a channel utilization of 100% indicates that the message generation rates has reached or exceeds the capacity of the wireless links to deliver that data.

Performance Objectives

Routing in DTN seeks to maximize the delivery probability of messages and minimize their delivery delays while maintaining an acceptable overhead value [9][24]. Delays in DTN are unavoidable since it is an opportunistic network, however very long delays in some application scenarios limit the benefits that could be gained from this technology. To make this technology more effective requires more research into new routing protocols that actively minimize delays, while maintaining high delivery probability with low overhead ratio.

2.5 Routing in DTN

Routing in DTN is a challenging task due to the lack of a constant network topology over time. Therefore, routing protocols in DTN use store-carry-forward approaches. Mobile nodes have to take a series of independent relaying/forwarding decisions as they move. Some protocols prefer to take simple forwarding decisions such as relaying to every contact within range. Other protocols go for more complex decisions such as ones based on mobility patterns, energy availability, number of copies allowed (single copy/multiple copies) and other temporal and spatial conditions [39]. Routing in DTN has been the major aspect of the protocol that has attracted the interest of the researchers. Studies have presented several approaches to address unstable links, limited resources and other problems faced in intermittently connected networks. In general, the performance of a routing scheme in an opportunistic network may vary according to the network's characteristics including how the mobile nodes move, how dense they are and how far the two end points are apart [12]. Therefore, forwarding protocols in DTN have to be adaptive to the networks' needs and be well designed to cover and suit most possible scenarios. Many routing protocols with different complexity levels have been proposed. These protocols are mainly classified into two main routing categories, namely flooding based routing approaches and estimation based routing approaches. The principle of the flooding based strategy is to replicate and relay the messages to many possible nodes so that the destination node has a higher chance of being one of the receivers. However, replication approaches consume high network resources. On the other hand, the estimation routing based approach makes use of local and global network knowledge to estimate the best candidate link to forward the message to [16]. In the following sections the details of these strategies and some of their corresponding protocols will be presented [16].

2.5.1 Store-Carry-Forward

Store-carry-forward has become a key concept used in DTN technologies [9][10][3]. Store-carry-forward is an asynchronous message passing paradigm that a node follows after receiving a message. The "Store" phase is adding the message to the node's buffer which allows the data to wait for a suitable time or peer to forward the message. "Carry" is the stage that allows the message to propagate to other regions of the network physically through the movement of the node carrying the data instead of relying on its transmission through the limited available network media. Finally, "Forward" is the stage when the node decides to send the message to another node due to the availability of other better candidates or to the message's final destination [10]. Figure 3 clearly illustrates these stages.

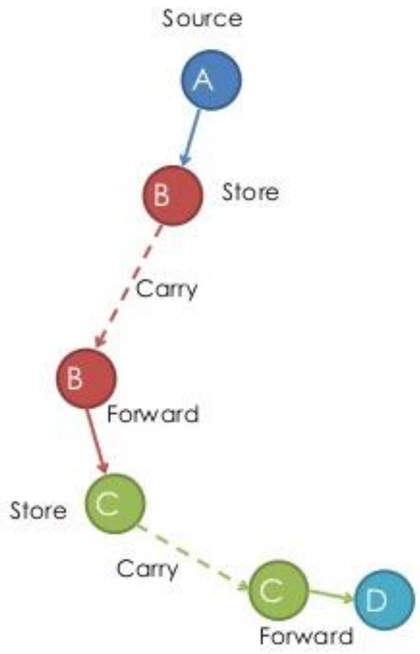


Figure 3. Store-Carry-Forward example

2.5.2 Flooding Based Routing Approaches

In the flooding routing strategy or also called replication routing strategy, multiple copies of the same message may be created and relayed by the source node to a set of nodes in the network. Similarly, the receiving nodes may relay copies of their messages when they are in a "contact" phase, and ultimately the destination node receives a copy of the message sent to it [16][11]. The decision of how many copies to relay and the selection of the relay nodes differ from one protocol to another. Most of the protocols in this category assume unbounded resources in terms of energy, buffer and bandwidth and also assume highly random mobility of the nodes participating in the network. Obviously, these assumptions increase the chance of delivering the messages to their appropriate destination since the nodes are using all possible opportunities. In realistic DTN this is not always the case, and flooding multiple copies of the messages with limited available resources can drastically affect the performance of the network.

2.5.2.1 Direct Delivery or Single Hop Transmission

Direct Delivery is the simplest strategy used to send data to their destination in DTN. It is a degenerate form of flooding in which the message is forwarded to the minimum number of nodes. In this protocol a message will be forwarded only when the source and destination are in direct contact. In other words, a message delivery is successful only if the source node and the destination node are immediate neighbours or one hop away from each other. No relays are made in this strategy since every node has to deliver its messages on its own. The one advantage from this protocol is that minimal network resources are required due to the no relays made saving energy and buffer [24]. However, this protocol has main drawbacks of limiting the opportunities to deliver the messages exposing them to high possibility of loss before their expiry or to very high delivery delays [16][40][41]. Mostly, this protocol is used as a lowest overhead ratio comparison for other more practical protocols.

2.5.2.2 First Contact or Two Hop Relay

In this strategy, the source node relays one copy of the message to the first neighbour node it gets in contact with and then no more copies or relays of the same message are made by any node. After this process, the source node and the relay node switch to a direct delivery mode in which the message will be delivered only if the nodes get into the contact range of the message's destination node [42][43]. In this protocol, the probability of a message delivery is higher than the delivery probability of Direct Delivery due to the higher use of network bandwidth and buffer storage [44]. Although this protocol attempts to increase the delivery probability and reduce delays while consuming minimum resources, it still limits the delivery opportunities and shares the same disadvantages of Direct Delivery.

2.5.2.3 Epidemic

In the Epidemic routing protocol, a node carrying a message forwards a copy of the message to all other nodes encountered on its path. Similarly, the receiving nodes will follow the same behaviour [19][39]. The nodes maintain a summary vector of the messages in their buffers so if the encountered node already has a copy, the message is ignored. In this algorithm, the message will spread through all set of available nodes in the network until hopefully being delivered to the destination by one of them [16][19]. This protocol assumes small size messages and that the nodes in the network have unlimited network resources like energy, buffer capacities and bandwidth where a node is capable of sending all absent messages in the neighbour's summary vector at each contact. Clearly, this protocol maximizes the message delivery probability at the expense of very high consumption of the available node and network resources such as the network bandwidth and the node's memory [19][45]. Also, Epidemic routing should cut the delivery delays to the lowest possible since the message is propagating through all possible paths. Note that in this flooding strategy the messages continue to be replicated and spread over and over among the nodes even after the message has been successfully delivered to its destination [24]. Many studies have addressed minimizing the consumption of resources in Epidemic routing [46]. Although the assumptions of this protocol may never be met, Epidemic remains a routing option that can be used when no better alternatives are available [24][47]. It also provides an upper bound of the delivery probability for comparison with other protocols.

2.5.2.4 Prioritized Epidemic

This algorithm is similar to Epidemic in routing concepts. A node relay its messages to all set of encountered nodes to hopefully reach the final destination. However, this protocol puts some constraints on the network resources and imposes partial ordering of the messages in the nodes' buffers [16]. In this scheme, the buffer capacities and network bandwidth are no longer unlimited causing message lifetime expiries and buffer overflows. In this algorithm, priority of transmissions and deletions are used based on some variable parameters such as the message generation time and the message expiry time [48].

2.5.2.5 SWIM

SWIM is similar to the Epidemic algorithm however the network has fixed sinks which serve as destination nodes. A message is considered as delivered once it reaches any of the available sinks. SWIM has high spreading rate, low delay and high use of network resources [9].

2.5.2.6 Spray and Wait

The Spray and Wait protocol forwards a fixed number, L, of message copies. Spray and Wait exists in two main versions, namely, vanilla and binary, which differ in the way they flood the L copies of the message to the nodes during their "spray phase". Vanilla is the simplest and it transmits the L copies to the first L-1 distinct encountered nodes. A node following the binary version of Spray and Wait starts by transferring half of the L copies to the first node it encounters and the other half to the second encountered node.

The receiving nodes then follow the same behaviour and so on. The second phase is identical for both versions, where after a receiving node is left with one copy of a message it enters the “wait phase”. In this phase, the message can be delivered only by direct delivery when the carrying node meets the destination [49][50]. A node holds the message until it is delivered or the time to live of the packet is violated. This algorithm reduces the use of the network bandwidth through the parameter L that controls the number of copies of the message in the network. By this strategy, the unbounded replication problem of the Epidemic protocol is also solved. However, the disadvantages for this protocol are high delays and system failure if the nodes that received the copy of the message never cross paths with the destination [9][16].

2.5.2.7 Spray and Focus

This protocol is similar to Spray and Wait algorithm, however with a modification to the second phase [51]. After relaying the L copies of the message and entering the second phase of the protocol, a node does not wait until the destination is encountered in order to do an additional relay and deliver the message. In the “focus” phase, the node is allowed to forward a copy of its message to a neighbour node that is potentially a better candidate to deliver the message using a designed utility-based scheme [51]. This protocol is specially designed to suit application where the movement of the nodes can be traced. It is shown that this algorithm works well in some scenarios. However, this protocol requires highly mobile nodes moving in all directions to be present in the field. In many practical scenarios this requirement cannot be met, nodes are more likely to move in limited small areas for the majority of the time. For instance, every student tends to move mostly in the department that he belongs to for long time and probably without visiting other places in the university. In this case, the nodes would have spread all their copies quickly to the immediate neighbours, but then few if any of the nodes carrying a copy might ever contact the destination [49][50][51].

2.5.2.8 MaxProp

MaxProp is a flooding based protocol which has limited storage and bandwidth. The effectiveness of this algorithm lies in the decision of which messages should be transmitted or dropped first. This is done by maintaining an ordered-queue based on the destination of each message and ordered by the estimated likelihood of a potential transitive path to that destination in the future. The protocol prefers to transmit the message with highest hop count and to delete the message having lowest probability to be delivered. Figure 4 shows the design of MaxProp message queue. The disadvantage of this protocol is its high processing cost when it is running in a large scale network [16][19][39].

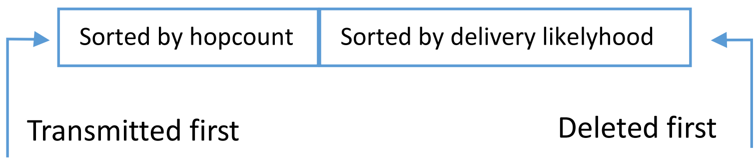


Figure 4. MaxProp Message Queues

2.5.2.9 Opportunistic Routing with Window-aware Adaptive Replication

This protocol is also known as ORWAR [53] which is a resource efficient protocol for routing in DTN. ORWAR aims to decrease the use of network resources and also the number of dropped messages during their transmission through the exploitation of the radio transmission range, speed and direction of movement of the mobile nodes. This protocol considers a scenario where the nodes are highly mobile and move at high speeds. In such scenario, the nodes face frequent connection abortions and reconnections with other nodes, and lots of partially sent messages. The ORWAR routing algorithm estimates the size of the contact window (number of bytes) of each connection to make better forwarding decisions and minimize the possibility of the messages being partially transmitted. This approach optimises the use of bandwidth through choosing the appropriate messages to be sent during contacts. Also, it reduces the energy consumption by cutting the possible energy that might be wasted on partially sent messages. Instead, the protocol will allocate more resources for high utility messages. The experiments done using ORWAR showed lower overhead ratio and higher delivery probability compared to MaxProp and Spray and Wait [53]. This protocol is limited only to nodes which are equipped with GPS, gyroscopes and accelerometers to give them the capability of measuring their speeds and their direction of travel. To route in intermittently connected networks, ORWAR applies a multi-copy routing scheme, using a controlled replication and a fixed number of copies distributed over the network. At every node encounter, half of the messages are forwarded to the neighbour node. However, the enhancements that this protocol performs are done in four directions: 1. Messages having the highest utility per bit ratio are first selected and then forwarded to the neighbour only if the message meets the contact properties without allowing partial transmissions to happen. 2. The replication factor is a function of message utility which increases the delivery probability for bundles with highest utility. 3. Messages having the least utility per bit are deleted first from the node's buffer. 4. Delivered messages are removed from the node's buffer [53].

2.5.2.10 RAPID

This protocol name stands for Resource Allocation Protocol for International DTN routing [19]. RAPID is a utility driven approach that aims to solve the resource allocation problems in the network. Under this protocol a message is routed by being replicated until a copy is delivered to the destination [19]. Specifically, RAPID addresses the problem of which packets to replicate given limited bandwidth in order to optimize a specified routing metric in the network such as minimizing average delay of packets, maximizing packets delivered within a deadline or minimizing maximum delay. At each transfer opportunity between two nodes, metadata will be exchanged and packets destined to each other will be delivered. Then, each node will compute the change in utility that can be gained from each message if it is replicated. The node will only replicate the message with the highest marginal utility among those in its buffer [19].

2.5.3 Estimation Based Routing Approaches

This forwarding approach makes use of local and global network knowledge to assess the available candidate links at each contact. Then, the node will forward the message to the best candidate with the highest likelihood to deliver the message to its destination. The local and global knowledge built by a node varies depending on the complexity of the algorithm, the input data being used and whether the algorithm is centralized or decentralized. In this strategy, most of the protocols do not replicate the messages but instead only a single copy of the message is forwarded until it reaches its destination [16]. Flooding the messages is not allowed, and this approach limits the use of the network resources and relies on making accurate relaying decisions to improve the network performance. The rest of this section presents some of the estimation based strategies used for routing in DTN.

2.5.3.1 Location Based Routing Strategy

This approach is one of the simplest in this routing category, it builds a basic knowledge about the location of each node in the network. The protocol basically uses the distance separating the source and the destination as the cost needed to successfully deliver the message. The protocol assumes that all the nodes are equipped with GPS devices and also have a general knowledge about the position as GPS coordinates of all other nodes. To calculate the cost of the link between two neighbours the source node uses a distance formula [16][54]. Then, the source node computes all the routes leading to the destination node and aggregates all their corresponding link costs. Finally, the decision is made and the message is forwarded through the link having the lowest cost. The main drawback of this protocol is that the shortest link in distance to the destination may have the highest disruptions among other possible links [16][54][55]. This will cause very high delays or the message being dropped. Another drawback is in the case of mobile nodes where the physical coordinates of the source, the destination and the intermediate nodes are constantly changing. Consequently, the costs of the links are not guaranteed to remain constant after the decision is made [16][54][55].

2.5.3.2 ProPHET

Probabilistic History based on Encounters and Transitions ProPHET is a well-known routing protocol in DTN that aims to use the statistics of previous encounters made by a node with other neighbours [19]. These encounters will be used to build and update a probability decision metric that estimates the probability of delivering the message to the destination. The delivery predictability estimate increases at each node encounter and decreases exponentially through time. The ProPHET protocol also takes into account the case where two nodes rarely meet but they frequently encounter a node in common through a transitivity parameter β [19][39][61]. In ProPHET, a node a keeps record of all previously encountered nodes, and for every encounter a probability value $P(a,b) \in [0,1]$ is assigned where b is the encountered node [19]. When two nodes meet, the summary vector holding all the predictability values for every known node is updated and also exchanged between the nodes in contact. A message relay between a source node a and an intermediate node b happens if $P(a,d) < P(b,d)$, where d is the final destination node of the message under consideration [19][39]. The probabilistic metrics between two nodes are updated through an encounter equation, an aging equation or a transitivity equation [39][61].

- Encounter equation:

$$P(a,b) = P(a,b)_{old} + (1 - P(a,b)_{old}) \times P_{encounter}$$

where $P_{encounter}$ is a constant initialized at the start of the scenario.

- Aging equation:

$$P(a,b) = P(a,b)_{old} \times \gamma^k$$

where γ is an aging constant and k is the elapsed time since the last encounter.

- Transitivity equation:

$$P(a,b) = P(a,b)_{old} + (1 - P(a,b))_{old}) \times P(a,c) \times P(c,b) \times \beta$$

where c is a transitive node and β is the transitivity scaling factor.

While ProPHET uses information about the delivery probability, it does not take into account the expected time delay for that delivery.

2.6 Critical Analysis of DTN Routing Protocols

In [19], the authors performed a performance analysis between RAPID, Epidemic and ProPHET routing protocols. The scenario consisted of a 4500x3400 m world in which there are 126 nodes having 5MB buffer size, 2 Mbps transmit speed and 10 m transmit range. The nodes were moving for 12 hours under a shortest path map based movement. The simulation for each protocol was repeated 6 times for different Time-To-Live parameters: 60,120,180,240,300,360. The node speed varied from 0.5 to 1.5 m/s, the message creation rate varied from 25 to 35 sec and the message size between 500KB and 1MB. Under their scenario, the results showed that in terms of delivery probability RAPID performed better than Epidemic and ProPHET which both had similar delivery probabilities. In terms of overhead ratio RAPID also performed better, followed by ProPHET and then Epidemic. And finally, in terms of average latency all three protocols provided very similar values.

In [39], another evaluation of DTN routing protocols was done. The comparison was made between Epidemic, ProPHET and MaxProp. The simulation area was 4500x3400 meters in which there are 50 nodes running under shortest path map based movement for 12 hours ranging in a speed from 0.5 to 1.5 m/s and buffer size 5 to 500 MB. The results showed that MaxProp performed better in terms of delivery rate, overhead and latency.

In [56], the work aimed to minimize the consumption of DTN resources in in terms of energy. The used approach was to calculate the size of the contact window between the nodes and use the available bandwidth efficiently while omitting possible partially sent messages. The forwarding decisions are made based on increasing this utility. Their simulations showed that the proposed protocol provides low overhead ratio and high delivery rate compared to the state of the art protocols. However, this work did not consider the delay factor in DTN.

The authors of [57] presented ALARMS which is a message scheduling approach in DTN. The proposed strategy uses predefined routes and arrival schedules of ferry nodes to exchange messages and successfully route them to their destinations. The protocol calculates the best route for a message and performs routing accordingly through nodes that will arrive earliest to the destination. The simulation results done in this work showed that ALARMS outperformed Epidemic, Spray and Wait and Spray and Focus in terms of all assessment metrics. The authors in this work considered fixed network topology and node movement scheduled ahead of time.

The work in [58] proposed a protocol for a very high number of mobile nodes. Their Firework routing approach starts by carrying the message near to the destination node. Then, the protocol switches to the explosion phase in which the receiving nodes floods the message in the area in an attempt to reach the destination. The results showed that this protocol was superior to Spray and Wait in terms of delay and aggregate throughput in a wide range of parameters. However, this protocol assumes that the nodes have a global knowledge about the location of the other nodes in the network either directly or indirectly. This assumption limits the applications where this strategy can be deployed.

In [32] a geographical routing protocol called AeroRP is presented. This protocol is targeted to suit highly dynamic environments such as aeronautical network architectures where jets travel with very high velocities. The approach uses the velocity of the node and the direction of travel to estimate the time of intercept with other nodes potentially moving towards the destination. In this work the authors used ns-3 simulations in which they showed that their protocol has several advantages over other MANET routing protocols. However, this protocol requires high processing capabilities at each encounter and also assumes that the location of destination node is known to the source node.

2.7 DTN Simulators

Finding a proper environment to assess a DTN routing protocol and prove its efficiency over others is yet another challenging task. Real experiments to capture the behaviour of a routing scheme are hard to realize most of the times and difficult to implement in a physical test-bed [59]. In addition, the cost of realizing such experiments in several environments is high in terms of tools and time since the experimental period may last for months or even years [59]. Therefore, evaluating a DTN routing scheme across various possible scenarios requires a suitable simulation tool that is able to replace real field experiments. The simulator should be able to abstract, simplify and analyse the behaviour of a DTN protocol and estimate its performance using specific DTN measures [59][60][12]. DTN simulators are classified into two approaches, namely time-driven and event-driven.

2.7.1 Event-Driven Approach

In an event-driven approach, the simulator operates as a discrete system performing a sequence of events in time. Every discrete event occurs at a specific instance in time during the simulation and mark changes in the variable states of the system. The main difference from the time-driven approach lies in the assumption that no changes in the system are going to occur between any two consecutive events. Thus, the simulator does not need to continuously track the system dynamics over the time. This approach is also called activity-based simulation and can run faster than a continuous simulation [59]. For example DTNSim2 is an event-driven simulator written in JAVA and used to implement and test various DTN routing protocols such as First Contact. In this simulator, a simulation can be configured through simple scripts that define the contact schedules and the traffic parameters. However, no updates have been published since the development of DTNSim2 in 2006. The implementations of recent DTN routing protocols do not exist in this simulator or they are not available for public [60]. Moreover, the simulator is restricted to one mobility model which limits the ability to implement various scenarios. Similarly, ns-2 and nsdtn-1 were developed with fixed mobility models and they work only for limited scenarios [59].

2.7.2 Time-Driven Approach

The time-driven approach is the leading technique used in the simulations of natural world [59]. In a time-driven approach, the simulator advances its global clock by fixed time steps t for continuous systems. At every interval t, the simulator checks if any of the events are due, performs corresponding actions and updates the state variables [59]. The t interval is carefully chosen by the user when setting up the simulation depending on the needed resolution for the experiment. A lower t interval reflects a higher experimental resolution and vice versa. This approach is sensitive to the value of t and presents a trade-off between precision and efficiency. A t value larger than the minimum time between two consecutive events means that some events are missed or not captured on time. This will decrease the precision of the system and may lead to wrong results. On the other hand, unnecessary processing and high resource usage will be witnessed in the case where the time step t is very small [59].

2.7.2.1 The ONE Simulator

The Opportunistic Network Environment (ONE) is a well-known time-driven DTN simulator written in JAVA. It is a powerful tool especially designed to implement DTN scenarios, evaluate DTN routing protocols in real time and get the results after the completion of the simulation [12][39]. The simulator contains a list of the state of the art protocols in the field such as DirectDelivery, FirstContact, SprayAndWait, ProPHET and Epidemic which can be used in the simulations of new protocols for performance comparison purposes. The simulation of a DTN scenario plays a very important role in assessing the performance of the routing protocol. For this reason, the ONE simulator presents extensible simulation frameworks to support a variety of possible models for node mobility, event generation, message exchange, protocol implementation, data reports and statistical analysis [12].

ONE calls itself a discrete event simulation engine since it generates variety of events such as messages creation events and movement events, however its global clock advances by fixed time steps so it also has time-driven aspects [59]. ONE is able to model the node movements and the inter-node contacts using various interfaces. Furthermore, ONE provides good libraries to use for the purpose of visualization, statistics and report analysis in addition to other post-processing tools [12][16][39].

In ONE, the existence of connectivity between nodes or its absence is based on the nodes' locations, their communication radio range and their bit rate. Nodes can be assigned specific routing functions that decide which message should be forwarded over the existing contacts of the node. In ONE each node can be given different capabilities in order to implement a realistic scenario. These capabilities can be summarized by the radio interface, storage, movement models and energy consumption [12].

The ONE simulator will be used as the basis of the simulation experiments in this research since it fits the needs of our scenario.

3 Research Questions and Methodology

DTN research seeks to solve the intermittent network connectivity issues and improve the routing performance in challenging applications. To date, research has proposed several DTN routing protocols. As presented in the Background, the protocols can be divided into two main categories namely flooding based routing approaches and estimation based routing approaches. In each of these categories the protocols use different input information to make relaying and forwarding decisions. The majority of the routing strategies has focused on maintaining a high delivery probability while reducing the overhead ratio. Although DTN can tolerate for some delays, excessive delays can limit usefulness of this technology in time-critical scenarios.

A key observation of existing DTN forwarding protocols shows the lack of researches that address optimizing delivery delays. In particular, no previous work has studied the incorporation of historical spatial and temporal information of mobile nodes in the DTN forwarding decision. The space and time dimensions provide useful measures to understand the underlying mobility of the nodes in the network and hence improve its routing performance. Another observation, is the nonexistence of a work that specifically benefits from the periodic and semi-periodic movement patterns of the nodes to route in DTN. Regular and repetitive movements are found in various real DTN scenarios, however none of the existing protocols is carefully designed to excel in these types of movements. Lastly, the nonexistence of an explicit DTN routing protocol that targets specifically long delays using historical spatiotemporal data of mobile nodes creates an additional gap in this area.

This research investigates new DTN routing algorithms which seek to reduce delivery delay while maintaining low overhead ratio and high delivery probability. The new algorithm attempts to predict nodes that will most quickly be able to deliver messages, and is called “Augur”, which means [62]:

> *VERB*
>
> *1 (Of an event or circumstance) portend a good or bad outcome:*
>
> *NOUN*
>
> *(In ancient Rome) a religious official who observed natural signs, especially the behaviour of birds, interpreting these as an indication of divine approval or disapproval of a proposed action.*

3.1 Thesis Questions

Other works have recognized that delay in DTN could be unbounded in some scenarios if the protocols don't deal with it carefully, but their solution is not sufficient since it does not consider spatiotemporal historical data. To address this problem, **this thesis is looking to use historical data of spatiotemporal movements to minimize the delay in DTN**. The feasibility of designing a DTN protocol using this information is studied and the dependencies are explored. The specific research questions to be answered in this thesis are:

RQ1: How can existing mobile DTN routing protocols be modified to incorporate the spatiotemporal history of nodes in the new "Augur" algorithm?

RQ2: What performance improvements are gained by the use of "Augur" in terms of average latency, delivery probability and overhead ratio?

RQ3: How does the performance of "Augur" vary with algorithm parameters (such as history length, buffer size), with network topology (static or mobile sinks) and with movement patterns (periodic, semi-periodic)?

3.2 Tools

ONE simulator

In this research, the Opportunistic Network Environment (ONE) simulator will be used to implement the "Augur" protocol, visualize the node movements and analyse the results outcome.

Our selection of ONE simulator which is written in Java (1.6) was based on the following reasons:

- The capability to generate node movement using different movement models (random, random way point, working day, bus, trams...)
- The capability to route message with many already implemented DTN routing protocols (Epidemic, Spray and Wait, ProPHET...)
- The capability to visualize mobility and message relay/delivery in real time through a graphical user interface GUI
- The capability to import real life data traces trough an external movement model.

- The capability of producing various report types after the simulation for future statistical analysis and performance assessment.

Datasets

In this work, two basic movement models will be used in our simulations. These movements exist by default in ONE and can be used to simulate different movement types and models. The primary movement model will be based on a Helsinki city model with its streets, bus routes, bus stops, tram routes and tram stops. In the first case, regular bus movements will be used, and in the second model, the bus speed will be varied to explore less regular behaviour.

3.3 Research Tasks

***Task-RQ1*:**

This task will investigate how spatiotemporal data can be extracted from nodes in DTN networks. Then, the availability of this information in each node will be exploited to opportunistically route message to sink destinations. A decision metric will be added to the DTN routing protocol based on forwarding the data packet to the node that is more likely to deliver the information in lower delay time.

Task-RQ2:

In RQ2, several possible scenarios will be implemented in ONE. First, the “Augur” algorithm performance will be tested with regard to the performance metrics. Second, the work will test the behaviour of “Augur” under the implemented scenarios including the accuracy of the predicted delay. Third, simulations will be done under the same scenarios using other traditional DTN protocols such as Epidemic, Spray and Wait, and ProPHET. Their performance results are compared to those of “Augur” routing protocol. The experiments and comparisons will be repeated again by using realistic mobility traces imported to ONE simulator. The comparison criteria will be based on the packet delivery probability, the overhead ratio and the average message latency.

Task-RQ3:

To answer RQ3, “Augur” performance will be evaluated with changing algorithm parameters, network topology and movement patterns. The scenarios will be simulated with the following variations:

- Varying the node history size
- Varying the node buffer size
- Varying the message time to live (TTL)
- Varying the message creation interval
- Varying the periodicity of the nodes movements
- Varying the number of nodes

4 Augur Routing Protocol

In this section, Augur the delay aware protocol based on spatiotemporal data is presented. First, the way how a node gathers spatiotemporal information to build its routing decision metric is introduced. Second, it is shown how the nodes make use of this data to forward a message greedily to the optimal local candidate having the least expected delay until hopefully reaching its destination.

In DTN, a node is characterized by a specific interface transmission radio range. A connection or a contact is realized when two nodes are within each other's radio transmission range. The time interval in which the two nodes stay in contact is called contact duration. Therefore by combining these two parameters, the time of connections and their contact durations, a node can record all meeting and intermeeting times (time between consecutive meetings) with other nodes during a day. Basically, Augur makes use of the time series of meeting and intermeeting occasions to understand the underlying movement topology, to predict future contacts and to route messages only through nodes having lower expected delay for reaching the message destinations.

4.1 Time Series of Connections and their Durations

A time series is an ordered sequence of observed data on a variable of interest during a specified time interval. Usually, a time series is represented by $Y = \{y_t : t \in T\}$ where T is the index set and ***t*** is an instance of the index set.

In Augur, each node in the network produces two time series of connections and contact durations at the end of each day. Fig. 5 shows a portion of the time series of connections and durations between a specific pair of nodes in terms of time of the day. These time series relate the observations seen during a day with a specific time at which they occurred. Notice the fact that delays only happen after a connection is finished. When the network is connected, the delay time until next connection remains zero. Augur saves up to 30 observation days in its history. Table 1 is another representation that summarizes the observations seen in Fig. 5.

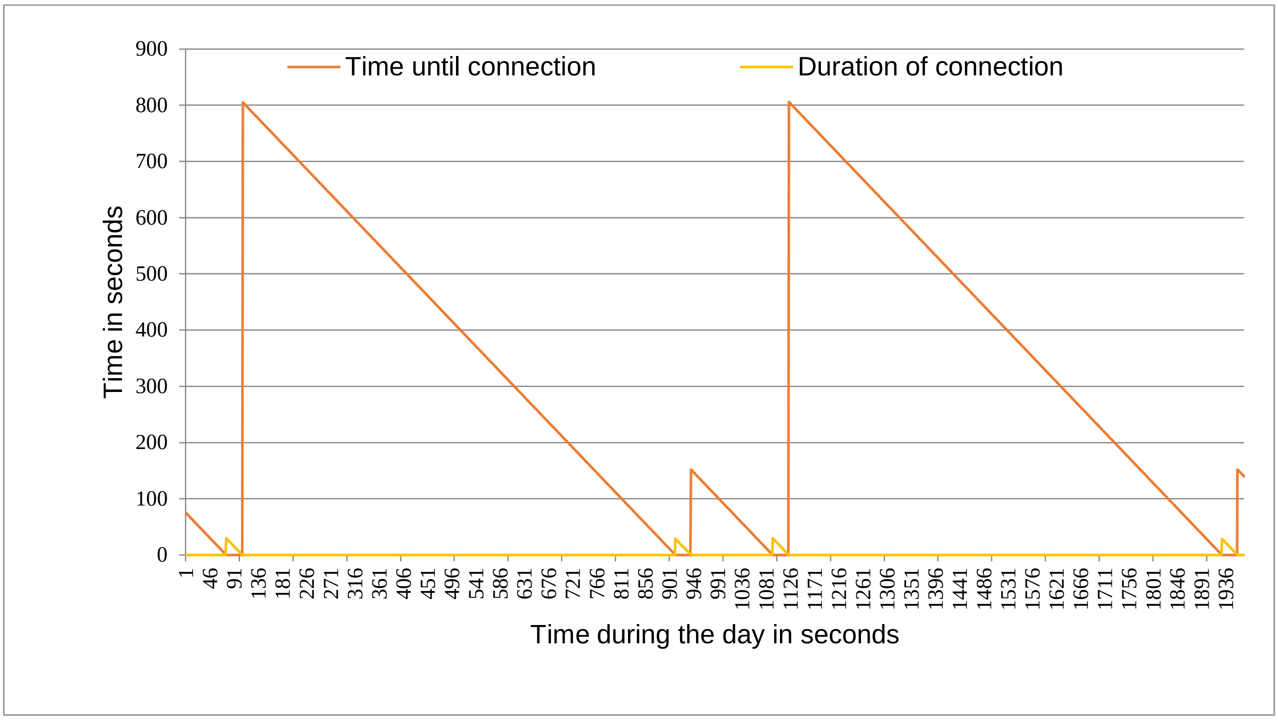


Figure 5. Times series of connection occasions and connection durations between a specific pair of nodes

Observations
Connection at **00:01:15** for **00:00:30**
Connection at **00:15:13** for **00:00:29**
Connection at **00:18:12** for **00:00:30**
Connection at **00:32:08** for **00:00:29**

Table 1. List of observed connections and their corresponding durations

4.2 Augur Temporal

At this stage, a node is able to derive a new time series which summarizes its delays to meet a destination from any point in time within the day. The delay value in y_t , where t is a point in time within the current day, is computed as follows. From the historical data, Augur averages all the delays during the previous days from time t till the earliest reach of the destination. This average is then used as the current day estimate at time t to reach the destination. Similarly, a node will follow the same procedure in creating the time series vector holding the average contact durations. From the node's history, Augur averages the durations of the earliest connections with the destination during the previous days from time ***t*** and considers this value as the duration estimate of the future connection with the destination. Based on this information, from time ***t***, each node can predict its delay until the next opportunity to meet a destination and also its contact duration time with that destination. Consequently, this knowledge will greatly assist nodes in performing effective routing decisions in DTNs as the results will show.

Fig. 6 illustrates a scenario where two nodes A and B meet in two consecutive days but at different times. At the beginning of the scenario and during DAY 1 both nodes have empty routing decision metrics since no historical data is present in their buffers. This means that none of the nodes is able to do any message forwarding unless the node encounters the message final destination. However, during DAY 1, nodes A and B recorded all the encounters that happened with other nodes during their journey. This task is repeatedly performed, each node records information about all its own daily encounters. Particularly, A and B each recorded their meeting on DAY 1 at 12:02:00 PM which lasted for 30 seconds. At the end of DAY 1, node A and B each compute the delays from all points in time to the earliest future contact with the other node. The durations of contact are also computed and stored in the node's buffer. Basically, the duration of contact values of the days with node B will be a count down from the time of contact until disconnection and 0 otherwise. Both nodes will consider the seen delay values as their decision metric for the next day, DAY 2, since no data for other days are present yet. Similarly on DAY 2, nodes A and B each record their meeting at 12:00:00 PM for a duration of 26 seconds. At the end of DAY 2, each node updates its routing decision metric to include the recent observations seen. The delays of DAY 2 are first computed and then the average of the delays and the durations in the history for every point in time is calculated. The averages respectively estimate the time at which node A and B will meet during DAY 3 and the duration of this contact. Finally, the delay value of A at time ***t*** until meeting B in DAY 3 is the delay value at ***t*** present in the node's decision metric. Node B follows similar procedure. In fact, the routing decision metric is an averaged summary of the nodes' encounters and their correspondent durations seen in the previous days.

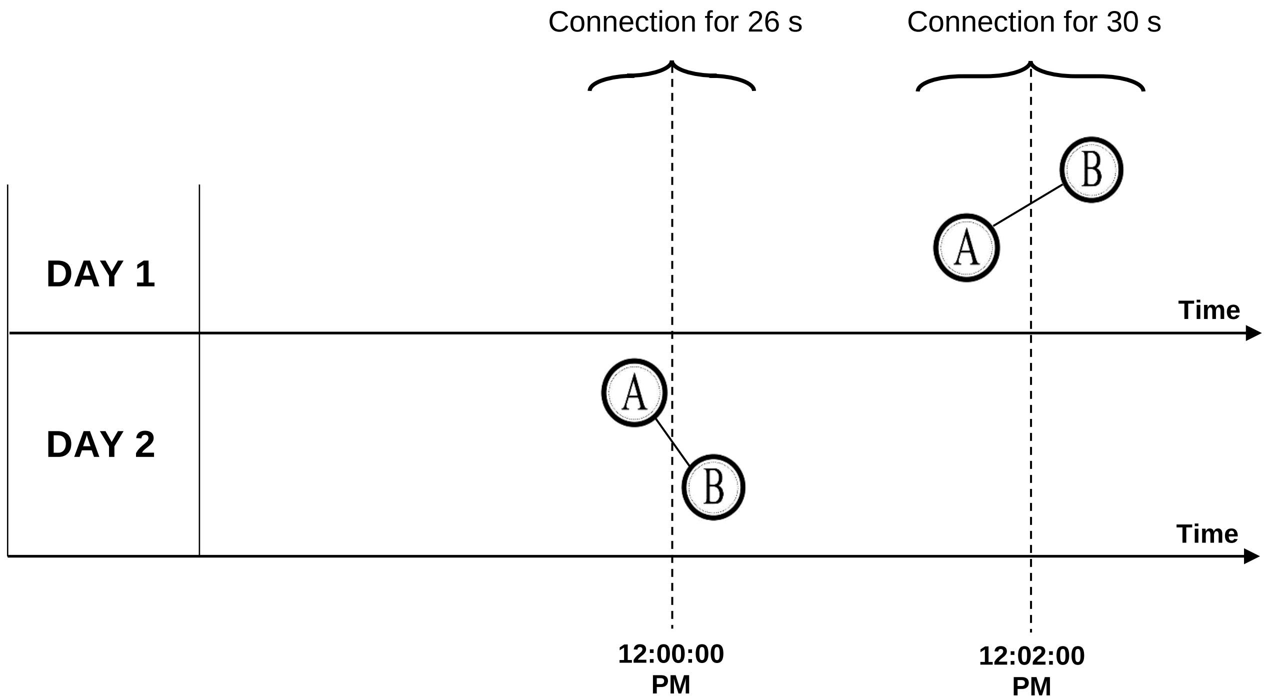


Figure 6. Illustration showing the computation of Augur decision metric (Delay prediction)

	Expected delay until the encounter of node B				
	…	11:57:59 AM	**11:58:00 AM**	11:58:01 AM	…
DAY 1	…	N/A	**N/A**	N/A	…
DAY 2	…	241 seconds	**240 seconds**	239 seconds	…
DAY 3	…	$\frac{241+121}{2} = 181 \text{ seconds}$	$\mathbf{\frac{240+120}{2} = 180 \text{ seconds}}$	$\frac{239+119}{2} = 179 \text{ seconds}$	…

Table 2. Table showing the Augur forwarding decision metric used in each day of the first 3 days considering that a forwarding opportunity happened at time 11:58:00 AM.

Table 2 shows how the computations of the delays in the previous scenario are done. For simplicity, it is considered that a forwarding opportunity happened to be available at 11:58:00 AM on each day. The table shows 3 seconds of the time series present in node A's buffer that summarizes its estimated meeting delays with node B. On DAY 1, no delay values are available. During DAY 1 the nodes' contact was at 12:02:00 PM which means node A is estimated to be 4 minutes or 240 seconds away from encountering node B on DAY 2 at 11:58:00 AM. Similarly, A is 4 minutes and 1 second away from B at 11:57:59 AM which is 241 seconds, and so on. On DAY 2, the nodes' encounter happened at 12:00:00 PM which is 120 seconds ahead of 11:58:00 AM, 121 seconds ahead of 11:57:59 AM and finally 119 seconds ahead of 11:58:01 AM. To update the delay decision metric, Augur averages the delay values seen in all previous historical data at every point in time during a day. The computation of the estimated delays for the following days follow the same procedure by averaging the observed delays at every point in time in the node's history. In the case where the nodes A and B do not meet every day, both nodes will have no contact observation during the non-meeting days. However, Augur averages the delays from the historical days during which at least one meeting observation between the two nodes is seen.

4.3 Augur SpatioTemporal

In the spatiotemporal version of Augur, the protocol will produce two time series similar to the ones in Augur Temporal. In addition, a third time series data structure holding the locations that the node visits on each day is created. In other words, all times and observations during a day are geographically tagged by latitude and longitude coordinates. In addition, the direction of the nodes movement is also recorded with each position information, to assist the node in knowing if it is going toward or away from the destination. The notation of the delay value is now y_l, where l is the location of the node during a specific time of the day. At the end of each day, the locations visited by a node will be matched within a radius of 100 meters to those already saved in its decision list. Newly visited locations will be added and the previously visited places will be updated by taking the average of the delays observed in the last 30 days. This version gives the node the ability to estimate the delay needed till reaching a sink from any location it may visit. Similarly, the estimated contact duration is computed and saved in a corresponding data structure.

In the two versions of Augur the computation of the expected delay is similar, namely using the time until the node reaches a sink; and the time till the message reaches its turn to be delivered depending on its position in the queue. However, notice that the two versions differ by the input given to the protocol to estimate the delay. During the expected delay broadcast phase, Augur Temporal uses the current time of the day to do the estimation whereas Augur Spatiotemporal uses the node's current location at which the broadcast has to be made. In the case where Augur Spatiotemporal couldn't match the current location with any previous location in its history, or in other words the node is visiting a certain place for the first time, the closest location to the current location within a radius of 100 meters is used instead. Otherwise no expected delay is broadcasted.

	Expected delay until the encounter of node B				
	…	Location 1	**Location 2**	Location 3	…
DAY 1	…	N/A	**N/A**	N/A	…
DAY 2	…	200 seconds	**199 seconds**	198 seconds	…
DAY 3	…	205 seconds	**204 seconds**	203 seconds	…

Table 3. Table showing the Augur forwarding decision metric used in each day of the first 3 days considering that a forwarding opportunity happened at Location 2.

Table 3 shows a portion of the delay decision metric which contains the locations visited by the node during the day. In this example, it is considered that a forwarding opportunity happened to be at Location 2 every day. Location 2 can be anywhere. During DAY 1, node A recorded the time taken until meeting node B from Location 2. The delay value for DAY 1 that was needed by A to reach B from Location 2 is now used as the delay estimate for A to reach B from Location 2 during DAY 2. For simplicity, it is considered that the delay happened to be 199 seconds. Also on DAY 2, it is considered that 209 seconds was the delay needed for node A to meet node B from Location 2. In general, at the end of each day, the decision delay metric is updated by taking the average of the delays recorded at each location in the historical data. In this example, the delays of 199 and 209 are averaged and the resulting value 204 seconds is now the estimated delay from Location 2 to reach node B. A similar process is used on each day to update the delay estimates. Note that multiple meeting occasions between the two nodes may happen during a single day.

4.4 Augur Forwarding Algorithm

This section presents the details of the Augur forwarding algorithm whose pseudocode is shown in Fig. 9. It is considered that nodes exist in a network and each holds a number of data messages in its buffer that need to be delivered to specific destination nodes. When two or more nodes are within contact range, each of the nodes broadcasts a predicted delay value for every message reflecting the node's earliest opportunity to deliver the message to its destination starting from the time of broadcast. This broadcasted delay has two components: the time until the node reaches the destination; and the time until the message reaches its turn to be delivered depending on its position in the queue[1]. Figure 7 summarizes the broadcasted delay. By adding these two components the node's expected delay time for message delivery can be calculated. The queue delay is then compared with the duration of contact with the destination to check the availibility of enough time to deliver the message. This check ensures that a node having less delay does not get overloaded with relayed messages that it is unable to deliver. If the queue delay exceeds the contact duration, the node checks the next contact opportunity with the destination and broadcasts the correspondent delay value till delivery. At this stage, a node compares its delay value to the broadcasted delays of other candidates. If its delay is the lowest among all candidates the message is kept in its buffer, otherwise the message is relayed to the node broadcasting the lowest delay value. In a scenario where a node A holds a message for node D and A is in contact with a node B that is currently also in contact with D, B returns an immediate 0 delay value after checking that enough contact time is available for A's message to be delivered. In this case, the value in the delay time series is not checked or used since the return value is more accurate and based on live observation. If the receiving node doesn't have enough space in its buffer, the oldest message in the buffer is checked. If that message is older than the one to be transferred, the oldest message is dropped or otherwise the relay process is declined. This process is repeated until the node provides enough space for the incoming message. Once a message is relayed, the sender deletes it from its buffer. In other words, only ever one copy of the message is kept in the network. In general, the nodes in the

[1] In Augur, messages are prioritized based on their creation time, where newer messages receive higher priority.

network will tend to relay as many of their messages to neighbours boradcasting lowest delay values. Fig. 8 shows the interaction between two nodes in contact.

Expected Delay = Time until the node reaches the sink
+ Time till the message reaches its turn

Figure 7. The components of the expected delay

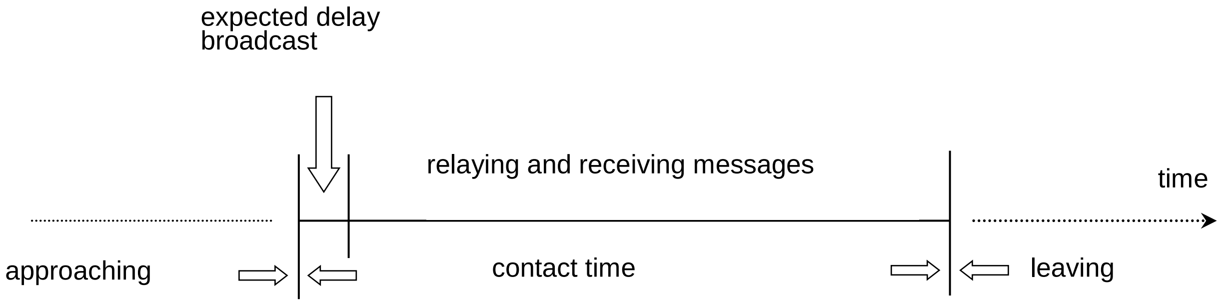


Figure 8. The interaction between 2 nodes in contact

Buffer

The buffer strategy adopted in Augur in the case of full buffer is dropping the oldest message in the node's buffer to guarantee the fastest possible delivery for newer messages. In this way, the message will follow the least encountered delay route at each contact and hence improve the protocol peformance in terms of all metrics, especially the latency.

Scalability

In the experiments, 25 mobile nodes are used in the Helsinki city model however the proposed protocols have no assumptions towards the number of existing nodes in the network or any other factor. All observations are saved in finite time slots during a day to a resolution of seconds. The algorithm is decentralized and each node builds its own decision metric.

Augur algorithm

Augur algorithm

```
IF (thisNode is in contact with neighbourNode)
    Sort thisNode's messages in descending order of their creation time
    Update connection time series
    Update duration time series
    Update delay decision metric
    IF (neighbourNode is a destination of any of thisNode's messages)
        Deliver those messages
    Broadcast delays
    IF (thisNode's delay < neighbourNode's delay)
        IF (message size > thisNode's bufferSize)
            Decline message
        While (thisNode's freeBufferSize < message size)
            IF (oldest message in thisNode's buffer < the creation time of the current message)
                Delete oldest message from thisNode's buffer
        thisNode accepts message from neighbourNode
```

Figure 9. Augur Forwarding Pseudocode

5 Performance Evaluation and Discussion

5.1 Simulation Setup

5.1.1 Simulation Tools

To evaluate Augur the Opportunistic Network Environment (ONE) is used with the Helsinki city model. In order to analyse the benefits and performance of Augur algorithms mentioned in the previous section and compare it with other protocols, ONE 1.4 was used since it includes different routing protocols such as Epidemic, Spray and Wait binary and vanilla versions, ProPHET as described in section 2.5.3.2 and in [61], and Direct Delivery and visualizes the simulations interactively in real-time and provides the various reports after their completion. Fig. 10 shows a screenshot of ONE simulator in action. The lines represent the streets of the city and the green circles represent the transmission radio range of every node. The small squares reflect the messages available at each node. On the sidebar the nodes that are participating in this network are listed.

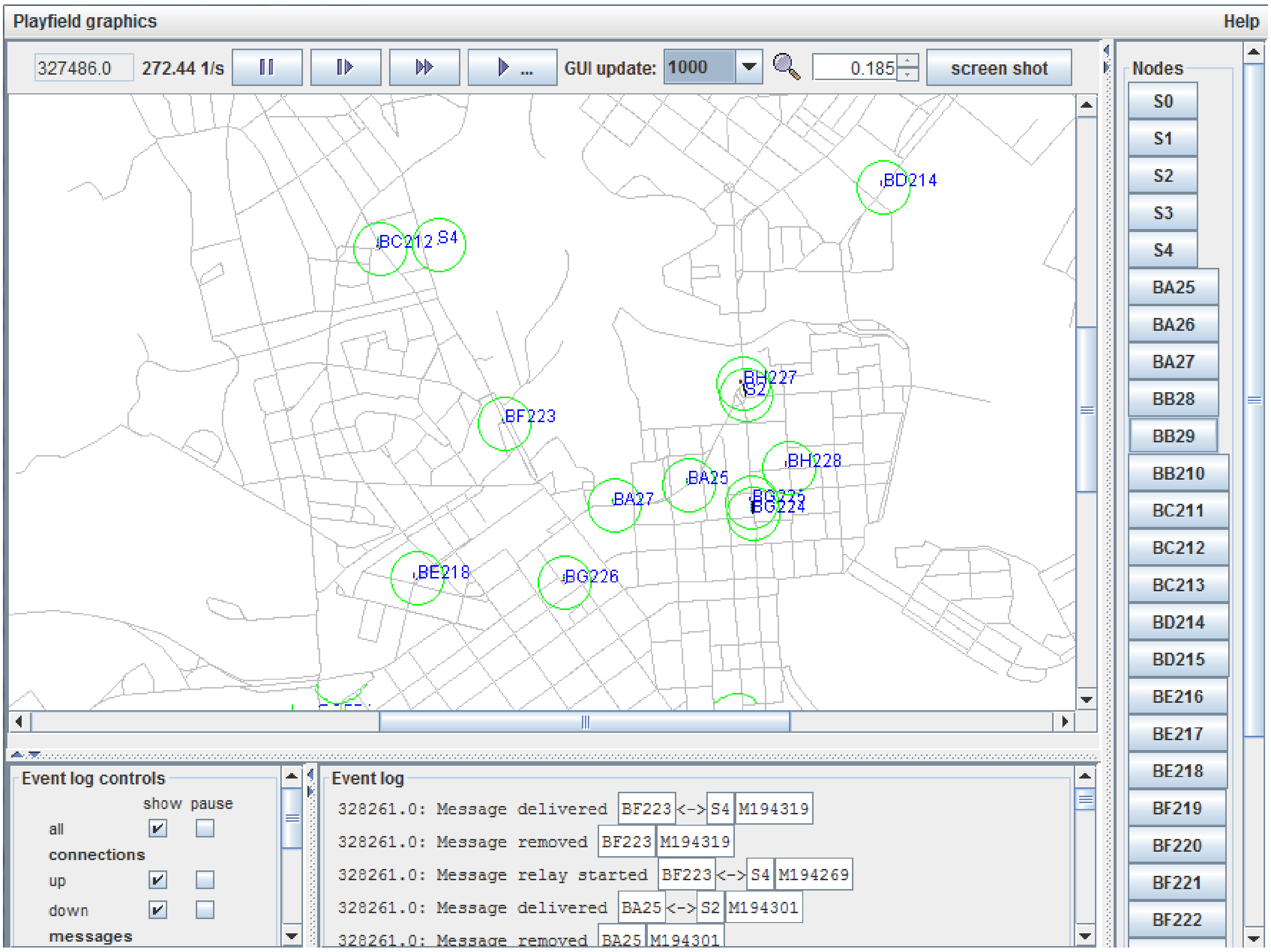


Figure 10. **Snapshot of ONE simulator during operation**

In order to simulate the behaviour of Augur in different scenarios, the ONE simulator is extended and 2 new routing classes AugurRouter.class and RelayedMessagesInfo.class are added. The core of Augur algorithm is implemented in the AugurRouter class which models the gateway of each node and it is responsible for building and updating the node's forwarding decision metric. The original ActiveRouter and MessageRouter classes are also extended to AugurActiveRouter.class and AugurMessageRouter.class to realize the correct behaviour of Augur. Each of the mentioned files exists in two versions belonging to each of the proposed protocols, one for Augur temporal version while the other for Augur spatiotemporal.

5.1.2 Motivating Target Application

It is expected that spatiotemporal based protocols like Augur excel in cases where the nodes show repetitive predictable movement patterns. Predictable time dependent movement will help these protocols to build trusted knowledge that they can depend on when making forwarding decisions. There are vaious scenarios that exhibit regular patterns with episodic connectivity in DTN, and in which the nodes are able to collect information in space and time about their mobility and also are able to keep track of historical contacts with the surrounding nodes. Some examples are people commuting to work daily, stopping by certain points of interest regularly like supermarkets, shopping centres and petrol stations and returning back home at the end of the day. Another example could be formed by animals that regularly visit specific foraging areas and resting sites. In these examples, the protocols are expected to perform best where the nodes' movements are perfectly repetitive and periodic everyday. The performance of the network is expected to degrade as the nodes' mobility deviates towards random movements.

An interesting scenario to examine is in the area of vehicular ad hoc networks using the Helsinki City map that already exists in the libraries of ONE. This application considers a city bus network in which the communication nodes consist of buses moving along predefined bus trajectories in Helsinki city. The busses travel circularly or back and forth on the routes of this map. This route based model consists of predefined locations such as bus stops or bus stations in which each node waits for some time before it continues the journey to the next stop using the shortest path algorithm. In this scenario, all the bus nodes move with a constant speed on the map routes to maintain a repetitive movement every day. Five fixed destination nodes are defined in this scenario and they are considered to have stationary movement located at specific bus stations of Helsinki map.

These nodes are also called sinks, and they are responsible for collecting the data generated by the busses. The generated messages can go to any sink. This scenario will give the ability to test the proposed Augur protocol and the state of the art protocols in perfectly periodic movement as well as semi-periodic movement by altering the bus speeds. Also, the existence of bus movements in ONE made the experimental environment easier and simpler to implement.

For this scenario, Table 4 summarizes the setup and the different simulation configurations used for our experiments:

Environment Parameters	**Value**
Total simulation time	432000 seconds = 5 days
World size	Helsinki City Map
Number of Bus Nodes	25
Number of fixed sinks	5
Bus Movement Model	BusMovement (MapRouteMouvement)
Interface Transmission Speed	250KBps
Interface Transmission Range	100 metres
Node Buffer Size	10MB; Infinite
Node Movement Speed	Constant 7 m/s, Variable speeds
Message Creation Starting Time	172800 seconds = after 2 days
Message Creation Rate	15;30;60;120;240 seconds
Message Time To Live	120 minutes [2]
Message Size	250KB

Table 4. Table of simulation parameters

[2] The TTL value of 120 minutes is large enough for this scenario.

5.2 Simulation Results for Constant Speeds

5.2.1 Exploring the Performance of Augur Temporal:

To capture the performance of Augur under different possible available resources, the study starts by exploring how Augur responds to the variation of the nodes' buffer size. To show this, the message creation rate is fixed to 1 message every 20 seconds at each node and the message time to live TTL value is set to 120 minutes. The experiment is run for 9 different buffer sizes consecutively [4MB; 6MB; 8MB; 10MB; 12MB; 14MB; 16MB; 18MB; infinite] and 4 main metrics are tracked: delivery probability, overhead ratio, average delay and channel utilization.

Fig. 11 shows that increasing the nodes' buffer size results in a higher number of successfully delivered messages since the nodes are carrying higher number of messages that can be delivered during their contact time with the sinks. This can be clearly shown through the increase in the channel utilization metric plot for buffer sizes from 4MB to 14MB. Further increases in the buffer size seem to give negligible improvement in the delivery ratio due to saturation in channel utilization and in opportunities to deliver messages as can be seen in the channel utilization metric that reached around 99% for a buffer size larger than 14MB.

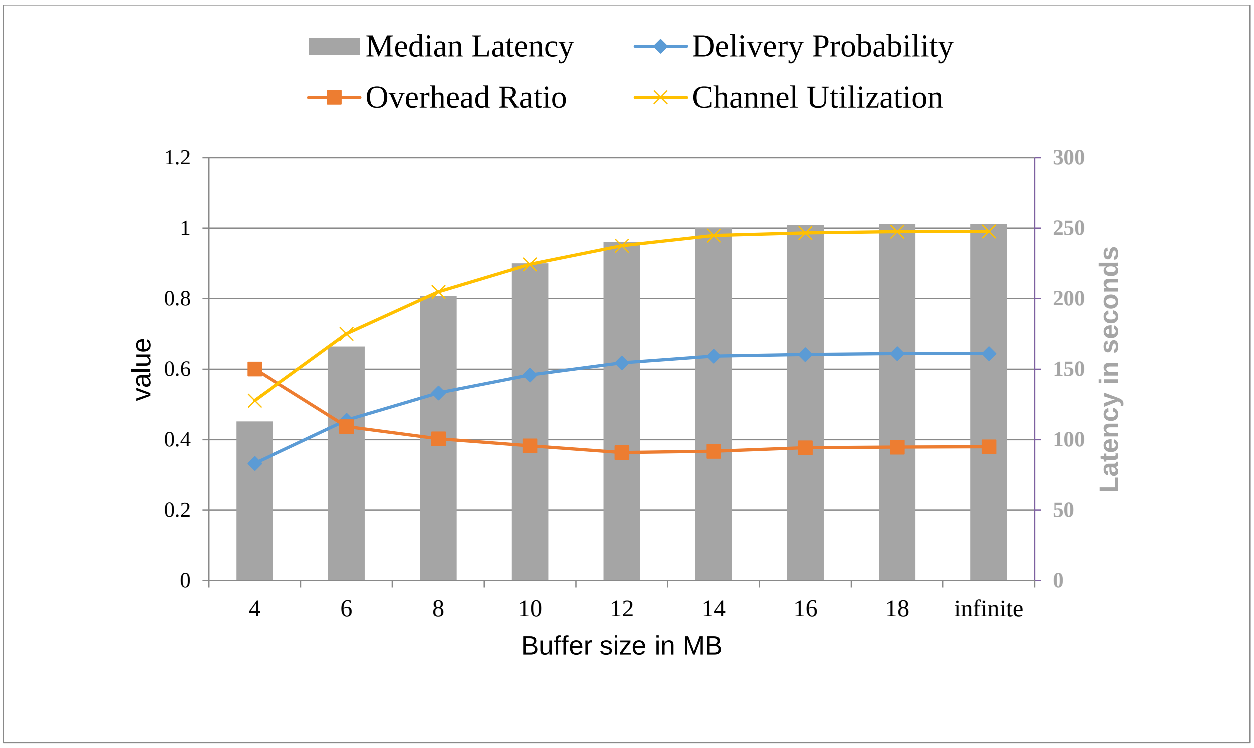


Figure 11. Performance of Augur Temporal for varying buffer size

Also notice that the overhead ratio decreases with the increase of the nodes' buffer size. The same number of messages should have been created by the end of each simulation (324000, same message creation interval). Also, the same contact opportunities and relaying decisions should have happened since the same movement model (constant speeds) is used for these simulations. However, with the increase of the buffer size the nodes are able to keep the messages for longer time and deliver them to the sinks before they are dropped due to a full buffer. This increase in the number of delivered messages along having the same number of forwards explains the drop in the overhead value. For the last metric which is the average latency, Fig. 11 shows that it increases with the increase of the buffer size, however this higher delay is primarily due to more delivered messages. The larger buffer size allowed the delivery of larger number of messages, by that means the messages waited for longer time in the buffer until being delivered and consequently altered the overall average latency. In this example, the message generation rate of 1 message every 20 seconds is higher than what the nodes can deliver at the destinations. It is noticed that an infinite buffer does not impose any useful benefit over buffer sizes ranging from 14 MB to 18 MB. This is due to the saturation in the available channel capacity where no additional forwarding/relaying opportunities could be exploited to improve any of the performance metrics.

5.2.2 Comparing Augur Temporal with other Protocols

In this section, a comparative evaluation among the proposed Augur Temporal algorithm, SnW in its two versions binary and vanilla where L=5 for both, ProPHET, Epidemic, as well as two simple protocols namely Direct Delivery and First Contact is conducted for comparison purposes. In Direct delivery no relays are permitted, whereas in First Contact the node that creates the message relays one copy of it to the first contact it meets and then no further relays of this message are allowed. The metrics of interest are delivery probability, overhead ratio, average delay and channel utilization. The desirable properties of the DTN routing protocol are to maximize delivery probability, minimize average latency and minimize the overhead ratio.

An important metric for the remaining simulations is the buffer size which may have an impact on the performance of the protocols. Fig. 12 illustrates the effect of increasing the buffer size on the different metrics for Augur Temporal and other protocols. The figure shows that the performance of Augur Temporal is relatively insensitive to the increase in buffer size, whereas all other protocols are much more sensitive. When the buffer size is increased the other protocols get much worse latency values.

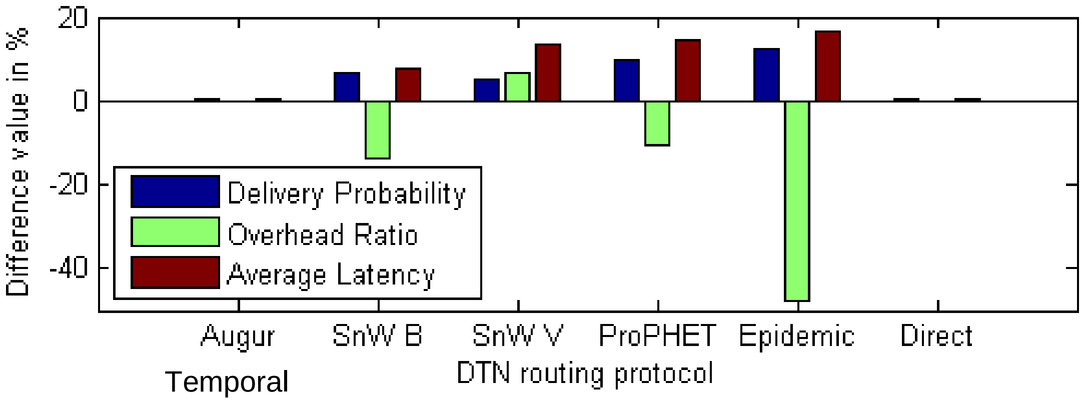


Figure 12. Relative gain/loss between 10MB and infinite buffer.

It should be noticed that our proposed protocol has no assumptions about buffer size or preference towards any other parameter. For this reason, since the proposed protocol is delay aware and this work focuses on minimizing the delay metric in DTN, a buffer size of 10MB is chosen to not disadvantage other protocols.

Delivery probability

Figure 13 shows the delivery probability of the seven different routing schemes under various traffic loads varying from 1 message every 15 seconds to 1 message every 240 seconds. The increase in the message creation interval resulted also in an increase in the delivery ratio of all the considered protocols. However, the delivery probability of Augur Temporal is always higher than the delivery probabilities of the other 6 schemes through all the experiments. The delivery probability of Augur Temporal becomes higher as the message creation rate slows starting with 0.4 at 15 seconds and reaching around 0.99 for a message interval of 120 seconds and above. In the case when the message creation interval was short namely every 15s and 30s, the delivery ratio of Augur Temporal was higher than all the other protocols with a value of 0.45 and 0.76 respectively followed by Direct Delivery with 0.44 and 0.72 respectively. Note that Direct Delivery performed better than the rest of the protocols except Augur Temporal in terms of delivery probability at high traffic loads. This means that when nodes are facing high traffic in the network it is better to stop taking any relaying decision similar to those done by the other protocols. These protocols appear to overload specific nodes with a high number of messages. The nodes are unable to deliver them later on due to the limited TTL, buffer size, bandwidth and contact time with the sinks. This leaves other nodes in the network with a small number of messages to deliver during their contact time with the sinks and the remaining contact time is wasted. This can be clearly seen in Figure 14 by the lower usage of the available channel by SnW, ProPHET, Epidemic and First Contact protocols at 15s and 30s, however on the contrary, Augur Temporal followed smarter strategies of load balancing among the nodes leading to a more efficient use of channel utilization that reached around 94% and 77% at 15 and 30 respectively. It is worth noting that the flooding based routing protocols used almost the full channel capacity at all times but most of the delivered messages were redundant and previously delivered by other nodes. These messages are not counted in this metric because the links were not efficiently used.

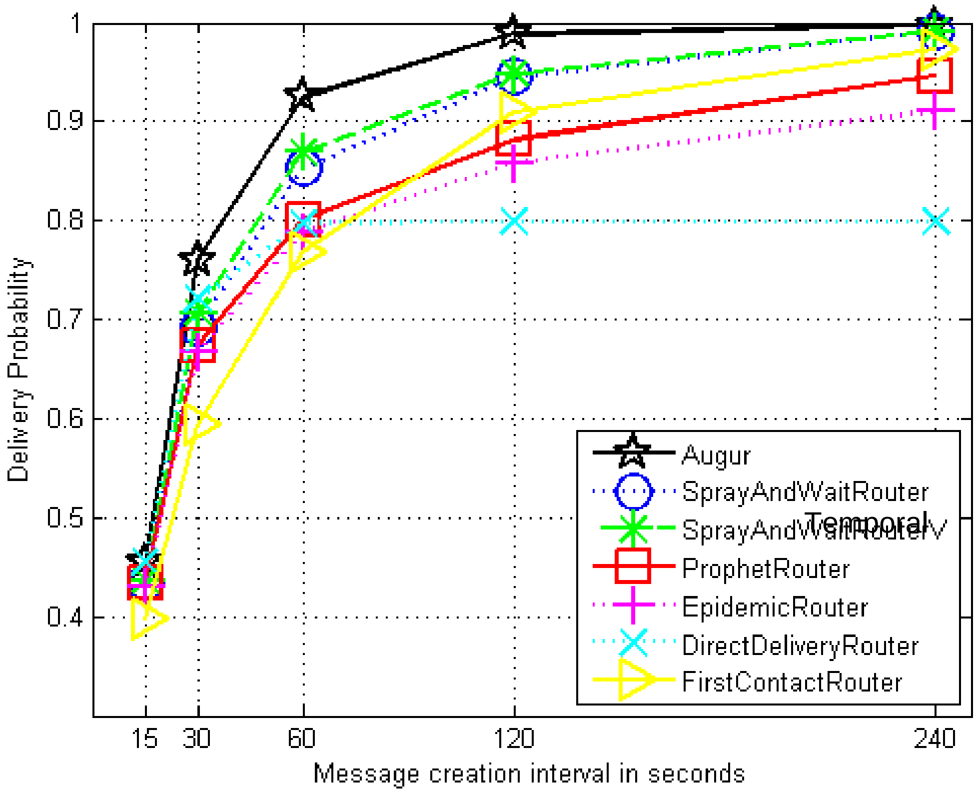


Figure 13. **Delivery probability for considered protocols for varying traffic loads**

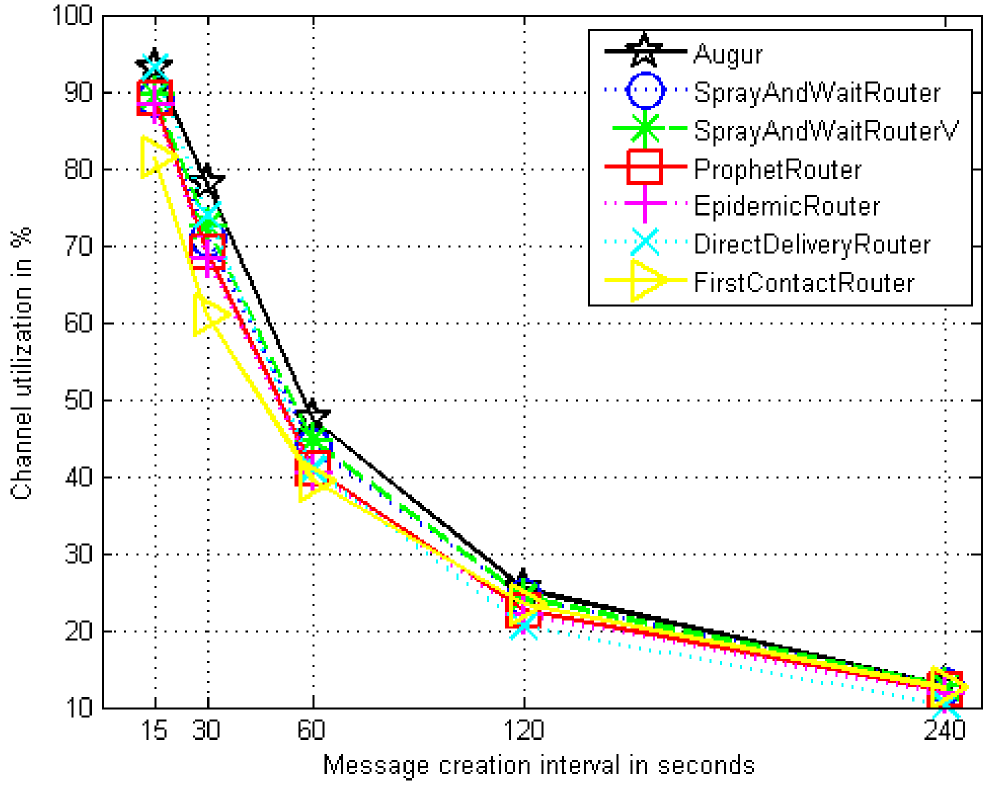


Figure 14. **Channel utilization for considered protocols for varying traffic loads**

Overhead ratio

Another important metric is the overhead ratio which gives an idea about how efficient the protocol is in terms of correct relay decisions and energy consumption.

Figure 15 illustrates the influence of the message interval on the overhead ratio. Obviously, the overhead ratio of Direct Delivery remains 0 since no relays are made by this protocol. Augur Temporal performs significantly better than SnW, ProPHET and Epidemic in terms of overhead ratio with the ability to keep its value less than 0.5 for all simulations. In this scenario, Augur Temporal reduces the overhead ratio by 77% to 94% for low traffic rates and by 73% to 88% for high traffic rates. The low value of overhead ratio of Augur Temporal at high traffic loads means that only accurate relays were made or in other words most of the relayed messages to other nodes were successfully delivered. This can be understood by looking at the low number of relays made by Augur at 15s shown in Fig. 16 compared to those made by the other protocols. This decision of limiting the number of relays at high traffic loads allowed Augur to excel in saving bandwidth and maintain higher delivery accuracy.

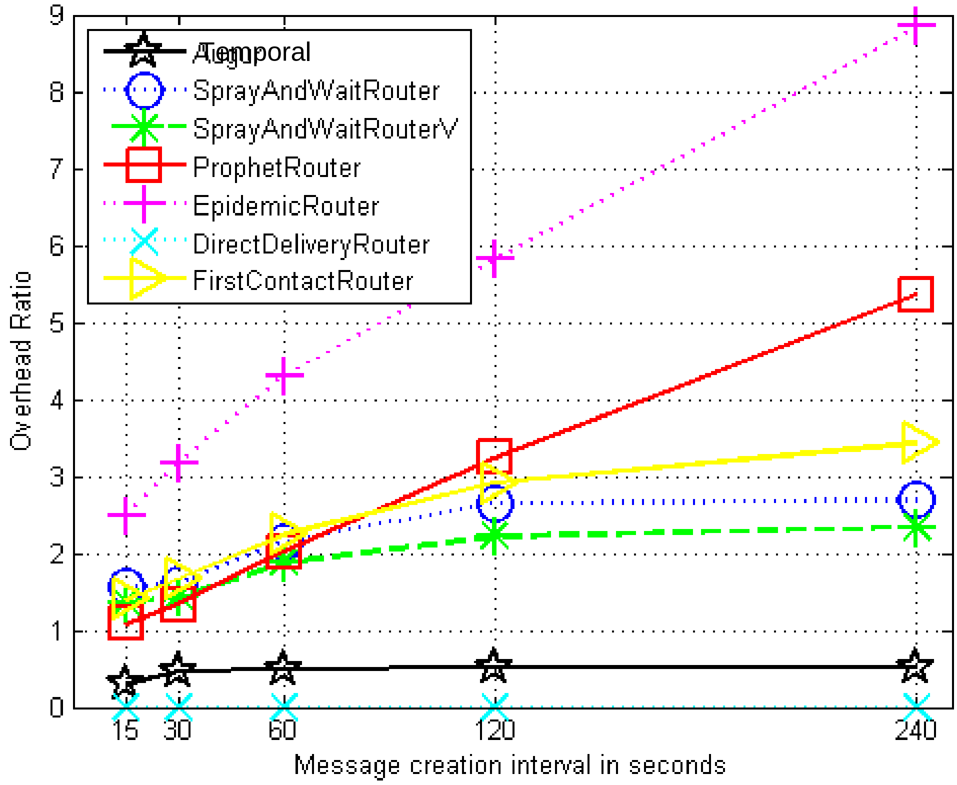


Figure 15. Overhead ratio for considered protocols for varying traffic loads

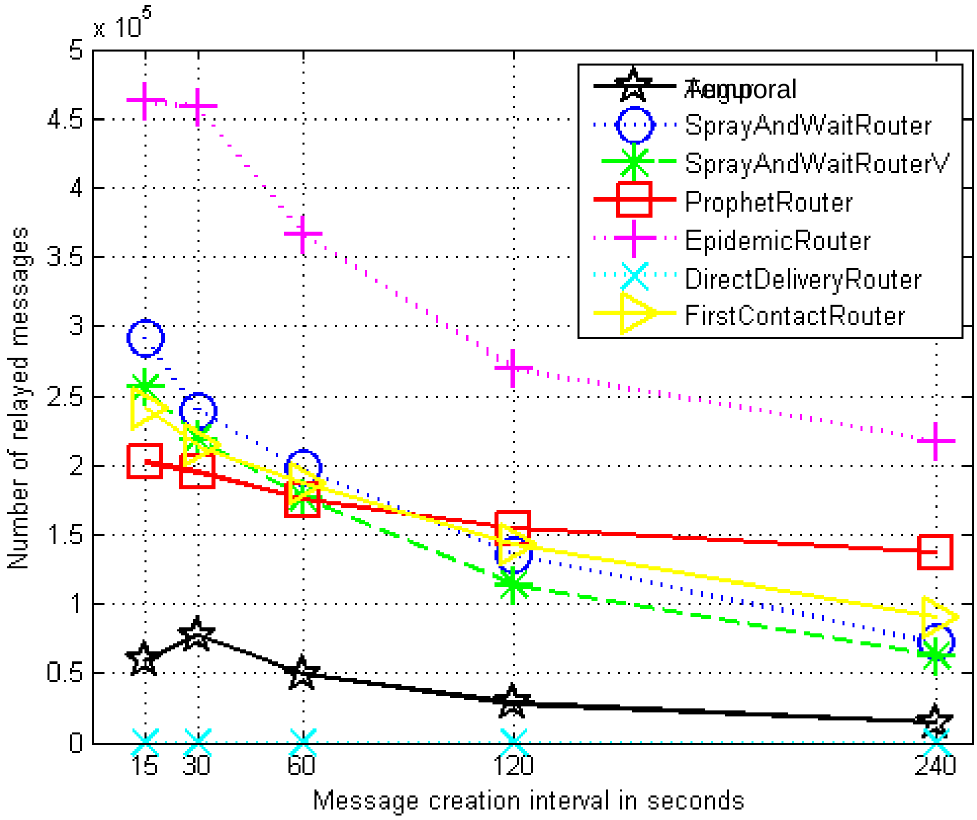


Figure 16. **Relayed messages for considered protocols for varying traffic loads**

Latency

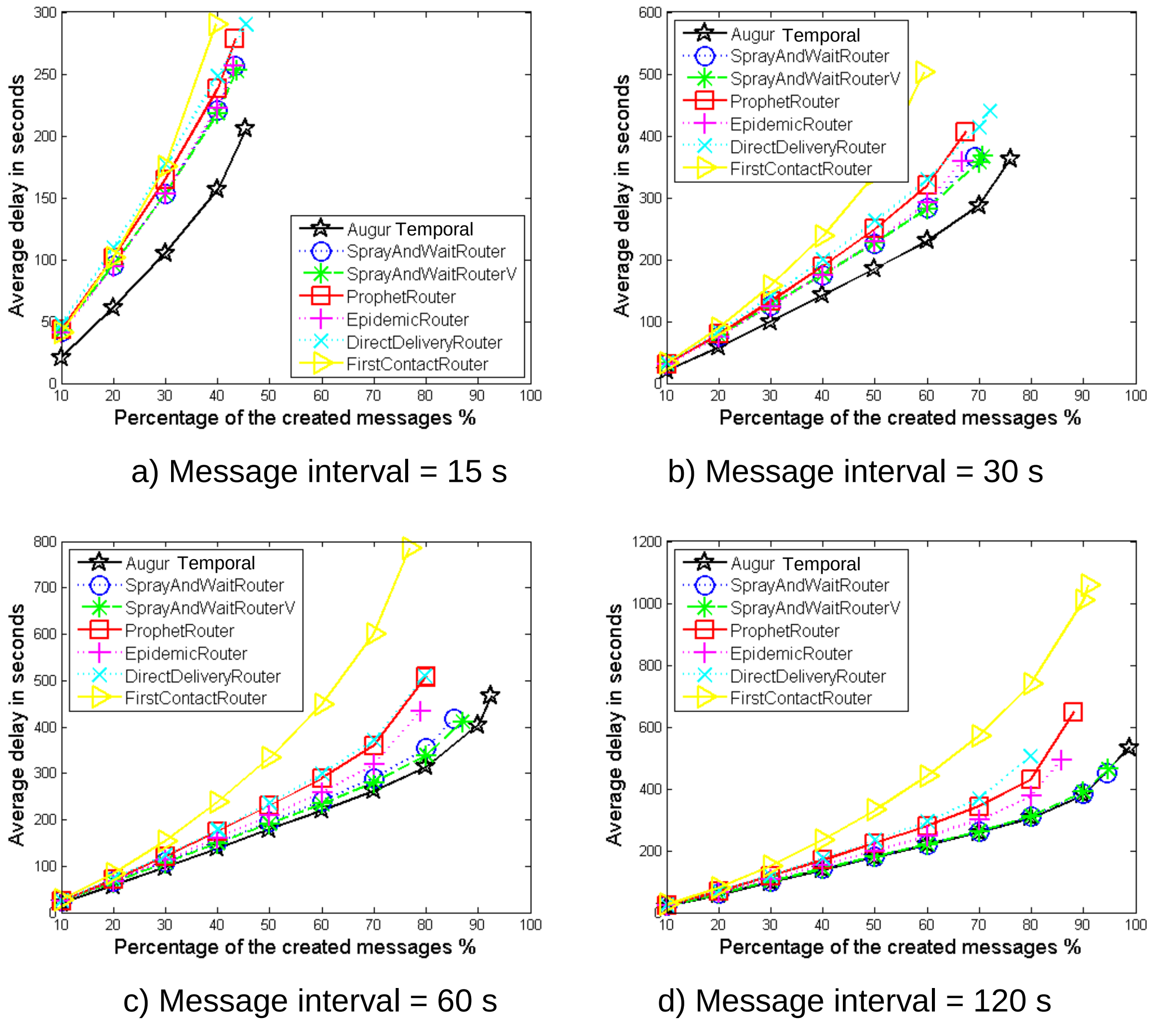


Figure 17. Average latency for considered protocols for varying traffic loads

Next, the performance of the all protocols is analyzed in terms of latency. The delivered messages are sorted in ascending order with respect to their delivery latency, and then the ordered messages are divided into 10 equal size bins and the average latency for these bins is calculated. The first bin includes the fastest 10% of messages; the second one includes the fastest 20% of the messages, and so on. Fig. 17 shows the average latency of the protocols for different traffic loads, 15, 30, 60 and 120 respectively. These plots can illustrate two metrics at a time. The x axis approximates the delivery percentage of each protocol and the y axis gives the average delay for a given percentage of delivered messages. Note that the right most point of each trace indicates the overall delivery rate and the overall average latency of the simulation for the corresponding protocol. For these plots, the "best" protocol is the line closest to the X axis (lowest latency) and which extends furthest towards the right (highest delivery probability). Augur Temporal is best in all cases.

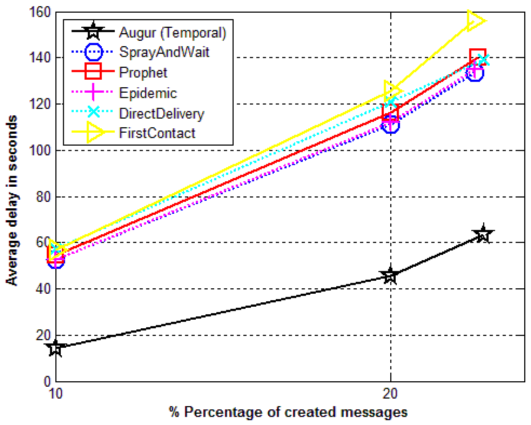
From the figures, it can be stated that at low traffic rates Augur Temporal, binary Spray and Wait and vanilla Spray and Wait perform similarly in terms of average latency with an average difference around 3.5 seconds for Augur across all bins. In other words, Augur Temporal is able to deliver each message 3.5 seconds quicker than the closest other protocol candidate. The time saved is approximated by a reduction of 3% to 54%. As the speed of message creation increases the performance of the other protocols deteriorates and that the delay difference between Augur Temporal and the closest competitor starts to be clearly seen. The average delay improvement of Augur Temporal compared to the closest performing protocol increases to reach 25 and 53 seconds respectively for the rates of 60 and 30 seconds scenarios. At the highest tested traffic rate of 15 seconds, Augur outperforms all other protocols in reducing the delivery delays. Augur Temporal saves around 60 seconds and decreases the average latency by 32% to 46% from binary Spray and Wait which was the best among the rest of the protocols. The simulations showed that Augur Temporal performs best again in the latency metric followed by Spray and Wait in its two versions then ProPHET, Epidemic, Direct Delivery and First Contact which came last. Higher delivery probability was reached by Direct Delivery for fast message creation rates at the expense of delivery latency. This is due to keeping one copy of the message in the network so no redundant deliveries are taking the bandwidth of other new deliveries. Augur follows the same strategy and improves its latency by load balancing among nodes, relaying and delivering newer messages first, dropping the oldest message in case of a full buffer and following lower delay routes for quick deliveries.

The analysis clearly shows that Augur Temporal gives the best results for delivery probability, overhead ratio and latency for all traffic loads. The superiority of Augur Temporal stems mainly from the use of temporal data of meeting and intermeeting occasions with sink nodes. This information gives the nodes the ability to approximate delays until likely future connections with sinks in addition to their durations. In the adopted periodic movement scenario, routing based on this metric showed an exact match between the estimated values and the actual meetings. This led to a lower value of overhead ratio, and since the forwarding algorithm leads the messages to follow the least possible delay routes, it is noticed that the messages are delivered faster, saving more bandwidth and giving the opportunity for other messages to be delivered. Consequently, Augur increases the overall delivery ratio.

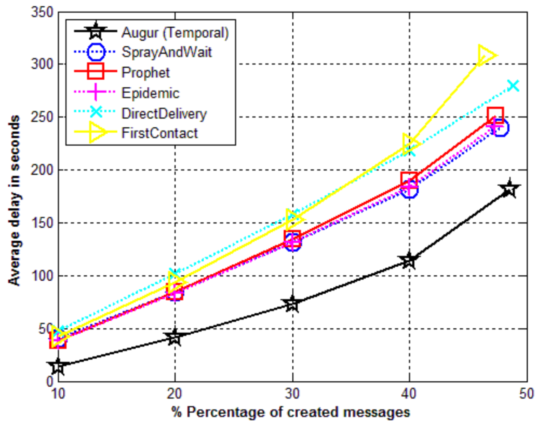
5.3 Simulation Results for Variable Speeds

In the previous experiments, all the nodes were moving periodically with a constant speed on each day. This effectively means that the spatial aspects of the moving nodes have limited effects on the scenario. As a continuation to our previous work, Augur Temporal is tested in a more challenging scenario where the nodes' movements are closer to reality. To realize the above, the periodicity of the movements is altered to semi or aperiodic. The nodes are still buses moving towards bus stops and bus stations however their speeds are varying during their journey. The same Helsinki city map and bus movement type as well as the positions of the sinks and number of nodes and other parameters are used. However, in this experiment, new random node speeds within the specified speed range are generated at every stop that a bus visits, the node will continue with the same speed until reaching the next bus stop at which a new random speed will be generated. Also, a higher message creation rate of 1 message every 7 seconds is added to the simulations to test the protocols in more challenging environment. The scenario's other parameters are the same as previously given in Table 4.

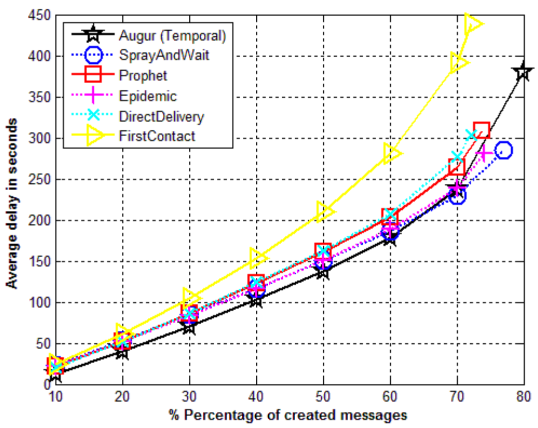
Figure 18 shows the performance of the different protocols in terms of average latency under the updated scenario or more specifically in the case of variable speeds. In this experiment, a speed range between 10 and 15 m/s is chosen for each of the moving nodes. Subfigures from (a) to (f) correspond to message creation rates of 7, 15, 30, 60, 120 and 240 seconds respectively. It is noticed from the figures that Augur Temporal performed better than the rest of the protocols at low message interval 7s and 15s in terms of average latency to reach a difference around 1 minute, however at an interval of 30s and higher its performance deteriorates gradually to perform close to DirectDelivery and prophet at 240 seconds. Figure 19 represents the performance of the protocols in terms of delivery probability. It is noticed that due to the limited available channel capacity all the protocols performed very similar in terms of delivery probability at high message creation rate. However, at a slower rate the delivery probability of Augur exceeds the delivery probability of the other protocols because only one copy of the message is allowed to be in the network giving opportunities to other messages to be kept in the buffers and eventually be delivered. From 7s to 30s, due to the high traffic loads and the limited buffer size the nodes had insufficient buffer capacity to accept all messages relayed from other nodes. In this case very few relays between the nodes were made, this can be easily noticed from Figure 20 which represents the overhead ratio values of the protocols for different traffic loads, with values of 0.2 and 0.4 at 7s and 15s respectively. So there is little effect on the protocol itself of estimating the delay. In Augur, the decision of allowing a message to be relayed or not is based on its time of creation compared to the time of creation of the messages that are already present in the receiver's buffer. If the message to be relayed is newer than the oldest message in the receiver's buffer the relay is accepted and the oldest message is dropped keeping only the messages that can most probably lower the overall average latency. Otherwise, no forwarding actions are taken by any of the two nodes. Compared to the other protocols that use first in first out message scheduling for delivery to the sink, the least latency first strategy that Augur uses proved its effectiveness in this scenario. Since the channel capacity is limited and the delivery probability can't be improved, it is best to use the least latency first to always guarantee the maximum delivery of fast messages within the contact period of the nodes with the sinks.



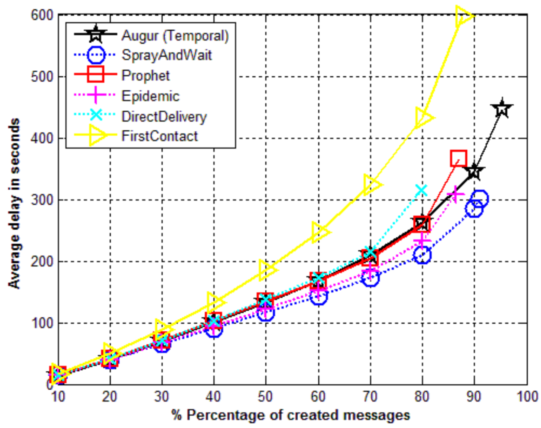
a) Message interval = 7 s



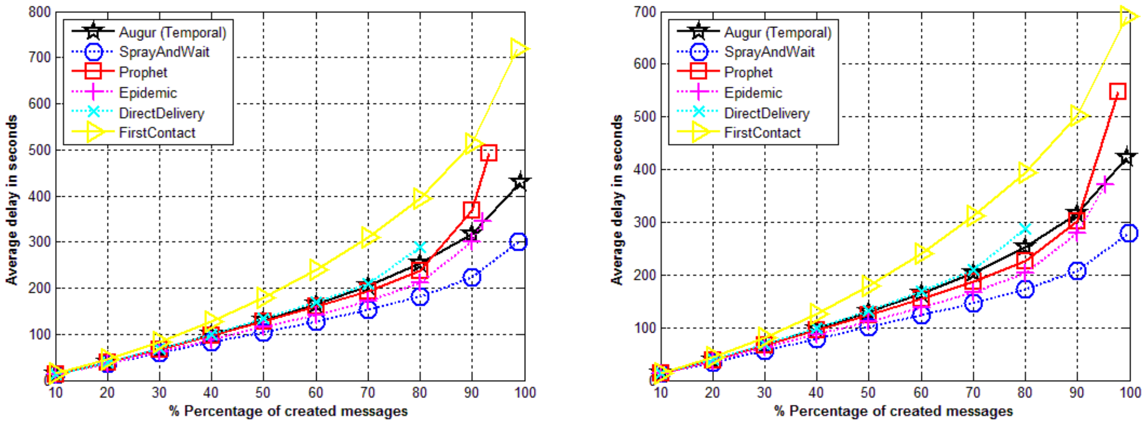
b) Message interval = 15 s



b) Message interval = 30 s



c) Message interval = 60 s



e) Message interval = 120 s   f) Message interval = 240 s

Figure 18. Average latency for considered protocols for varying traffic loads

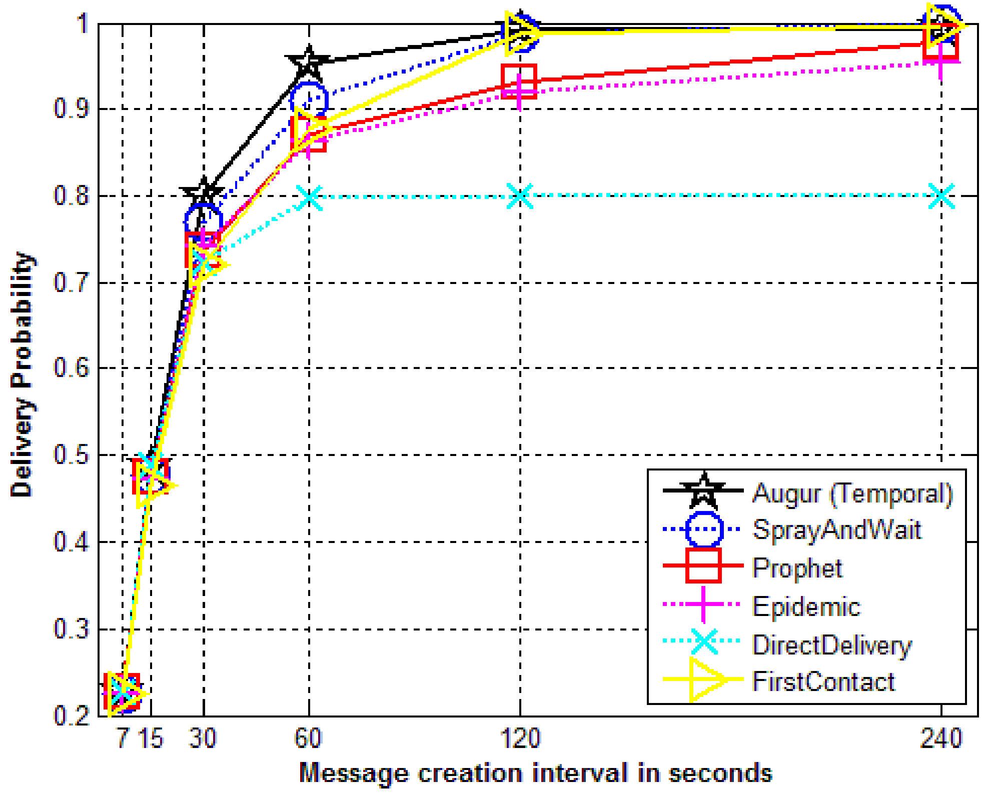


Figure 19. **Delivery probability for considered protocols for varying traffic loads**

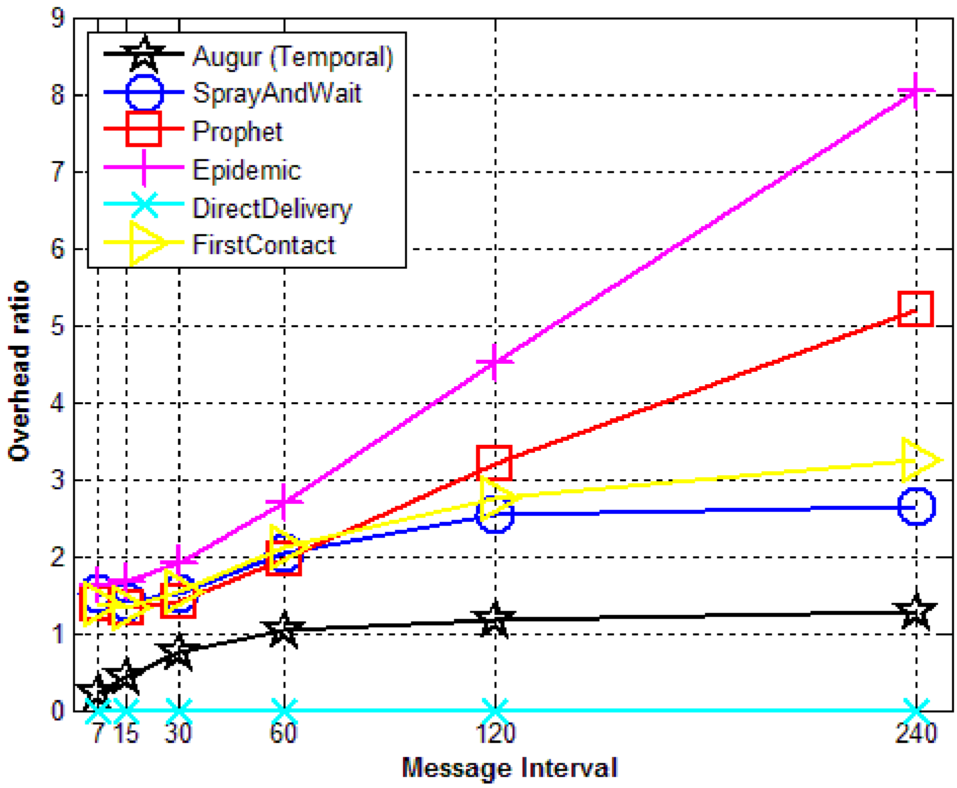


Figure 20. Overhead ratio for considered protocols for varying traffic loads

Problem of Augur Temporal in the scenario of variable speeds

The results at 30s and more represents the point where the performance of Augur Temporal starts to decrease noticeably in terms of average latency. The nodes are starting to have available spaces in their buffers and hence the possibility of accepting more relays from other moving nodes in the network. At this stage, the existence of this possibility means that the protocol has to proceed in performing the rest of its core stages and compute delay estimates to be broadcasted during the delay broadcast phase. However, the results showed that routing through the adopted algorithm is not effective and poor relay decisions were taken due to inaccurate or misestimated delays made by the nodes. In fact, since the movement of the nodes is not periodic or exactly repetitive every day a mismatch existed between the spatial and temporal characteristics of the moving node. In other words, the nodes are not at the same locations at same times every day as in the case of constant speed movement. For instance, in our scenario, a bus may not be at the same location at exactly similar times each day, however the bus might be around the location running late due to traffic or even running ahead of the schedule on its way to the bus station. The Augur Temporal algorithm in this experiment uses only the time during the day to deduce the estimated delay assuming that the location of the node in its history at that time were the same. The introduction of variable speeds broke the periodicity factor which made the scenario more realistic and led to several overestimated or underestimated delays.

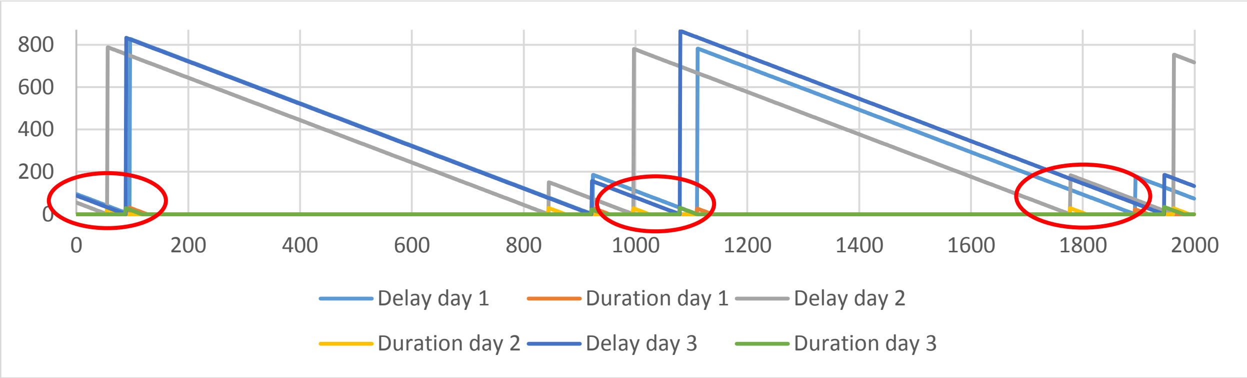


Figure 21. **Time series of the delays of a node for 3 different days in the case of variable speeds**

Figure 21 shows a portion of a node's time series of connections and durations in terms of time of the day in 3 consecutive days. Notice that the relation between the movement and the time is not exact on each day. A lag or lead time difference exist due to the different node speeds. For instance, all the contact observations that happened within the circled areas belong to a connection made by the node with the same sink, however at different times during the days. This means that basing the decision metric of the nodes only on the time during the day will not be effective in non-periodic movements because a node might be in different location at the same time on different days. For this reason, the spatial aspect of the nodes has to be incorporated into the decision metric of the forwarding protocol for more accurate routing decisions. Therefore, the advanced Augur SpatioTemporal version was introduced to exploit the spatial and temporal aspects of the mobility and address minimizing the delivery delays.

Figure 22 illustrates the steps and events that a node goes through at every encounter with a sink or specific destination.

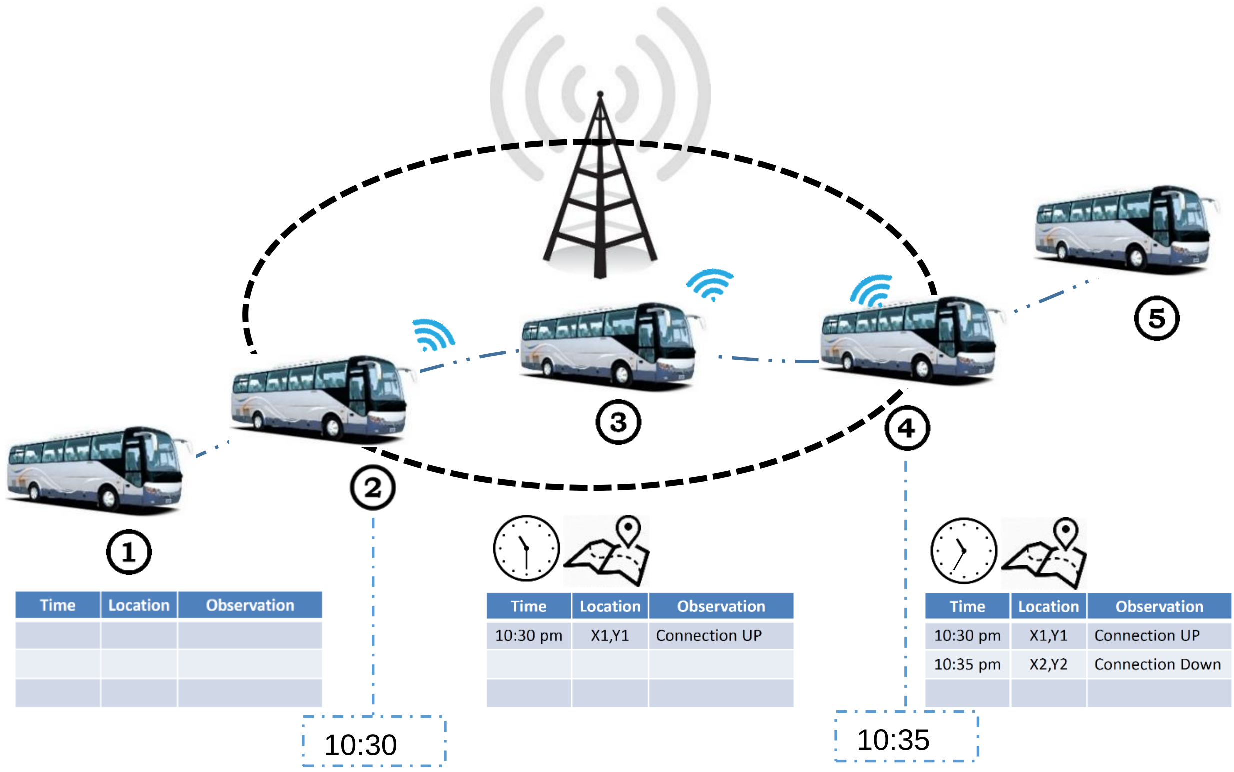
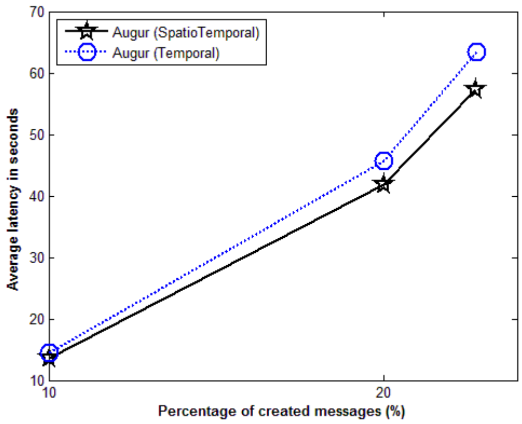


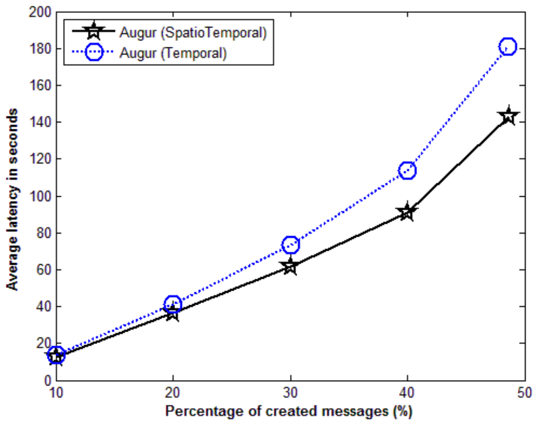
Figure 22. Illustration of the different steps and updates that a bus performs when encountering a sink with Augur SpatioTemporal

Comparison between Augur Temporal and Augur SpatioTemporal in the case of variable speeds

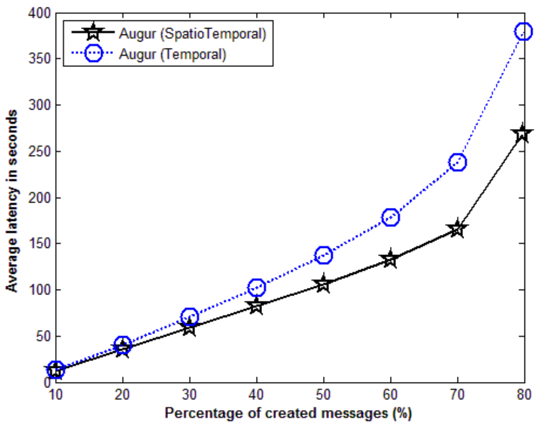
In this section a comparison is made between the two versions of Augur, Temporal and SpatioTemporal, to see the gains and losses that could have been made by Spatiotemporal over Temporal. Figures 23, 24 and 25 show the performance of Augur in terms of average latency, delivery probability and overhead ratio respectively for different message creation intervals ranging from 7 to 240 seconds. It is noticed from the figures that Augur SpatioTemporal was able to record faster message delivery throughout all tested message intervals. Augur SpatioTemporal was faster by about 10 to 20 seconds at high rate and by approximately 65 seconds at low rate of 120 seconds after which the average delay remained constant for higher message creation intervals. In terms of delivery probability both versions were able to deliver approximately the same number of messages during the simulations. In terms of overhead ratio both versions performed similarly with a slightly higher ratio for SpatioTemporal over Temporal. At a creation rate of 7s the overhead values for both protocols were the same. For the rest of the creation rate the average difference between the values of Augur temporal and Augur SpatioTemporal was about 0.1.



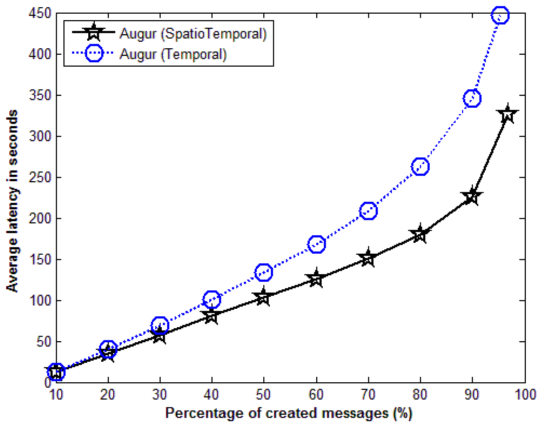
a) Message interval = 7 s



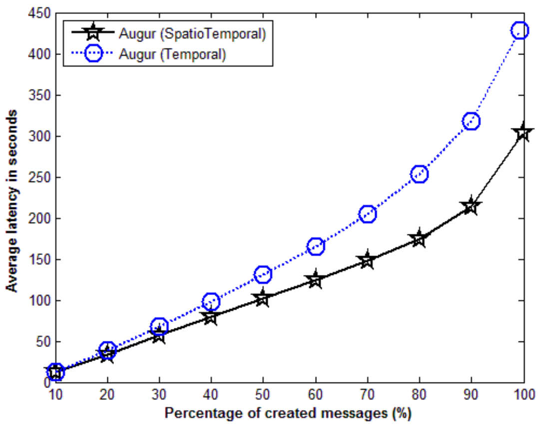
b) Message interval = 15 s



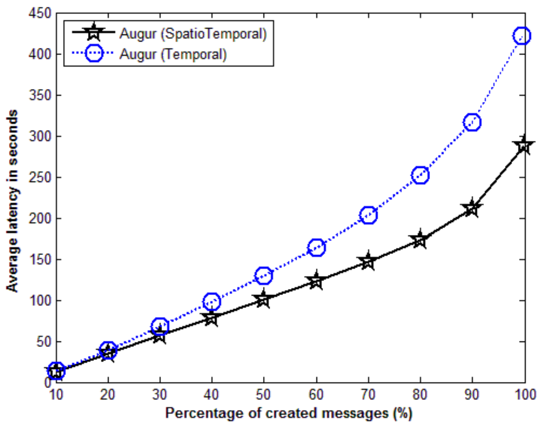
c) Message interval = 30 s



d) Message interval = 60 s



e) Message interval = 120 s



f) Message interval = 240 s

Figure 23. Average latency for considered protocols for varying traffic loads

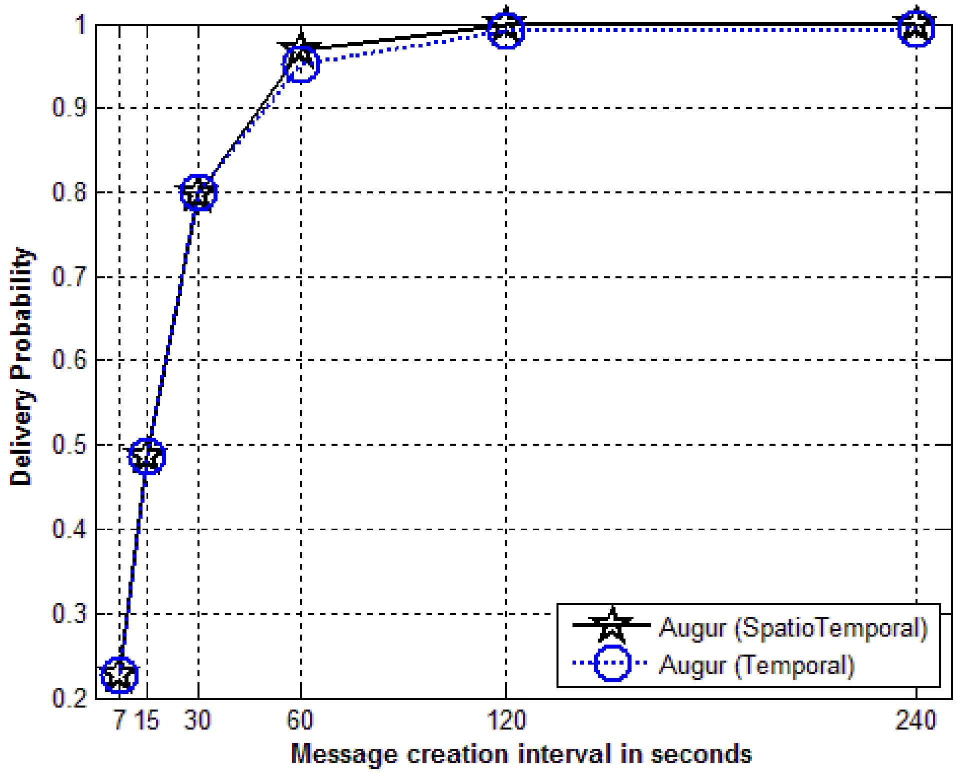


Figure 24. **Delivery probability for considered protocols for varying traffic loads**

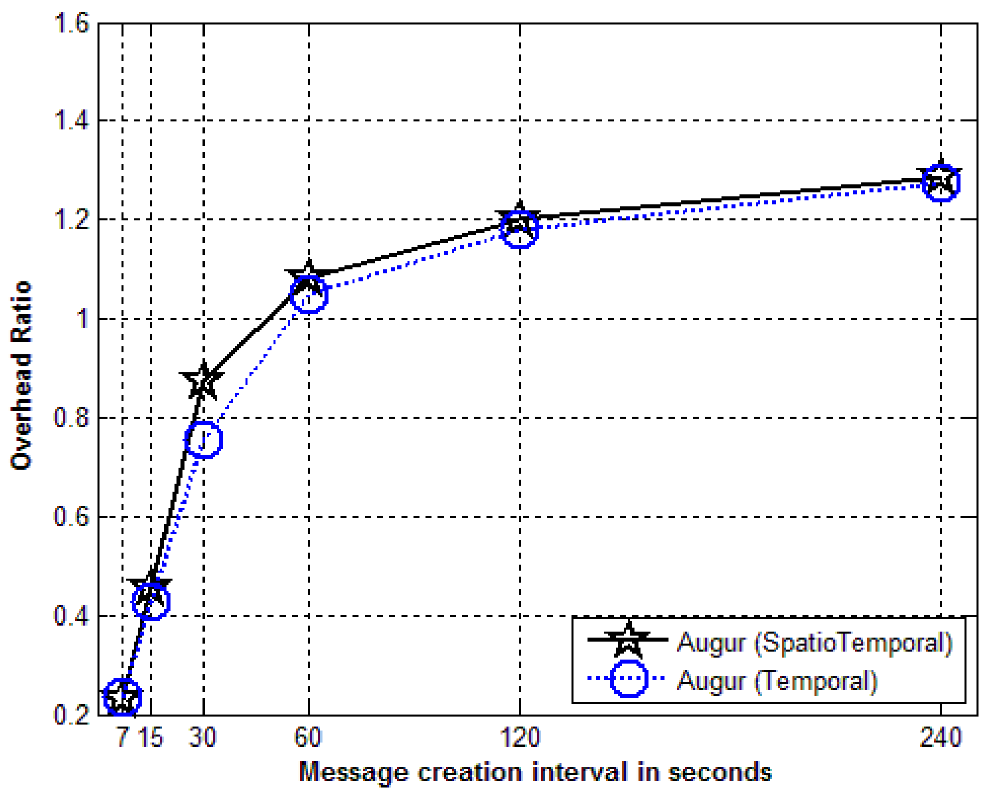


Figure 25. **Overhead ratio for considered protocols for varying traffic loads**

Varying the speed range intervals

This section explores the effect of varying the speed ranges of the nodes on the message average latency with results in Fig. 26 and 27. The delivery ratios can also be estimated from the same figures through the projection of the end point of each plot on the x axis that represents the percentage of delivered messages out of those created. In this experiment, the two Augur protocols are run in 4 different speed ranges 15:15; 12.5:17.5; 10:20; 5:25 m/s and for message creation intervals of 15, 30, 60 and 120 seconds. However, only the extreme cases of message rate 15 and 120 seconds and for speed ranges 15:15 10:20 5:25 are shown for simplicity. For a constant speed of 15 m/s in which the nodes' meeting and intermeeting times are deterministic both versions performed exactly similar. The two protocols were able to deliver their messages with an overall average delay of 116s at 15s and 269s at 120s. As the speed range becomes larger the average delay increases for both protocols however at a faster rate for Augur Temporal. At 15s, Augur SpatioTemporal had an average of 130s and 170s for speed ranges 10:20 and 5:25 respectively, whereas Augur Temporal recorded averages of 160 and 210s. At 120s, the average delay became around 270 seconds for Augur SpatioTemporal for all speed ranges. Whereas, Augur Temporal reached an overall delay of 484 and 385 seconds for speed ranges of 5:25 and 10:20 respectively. It is also noticed that the increase in the speed ranges causes a decrease in the delivery probability for each of the protocols, however Augur SpatioTemporal remained slightly higher than Augur Temporal in all these experiments by about 1.5% to 2%.

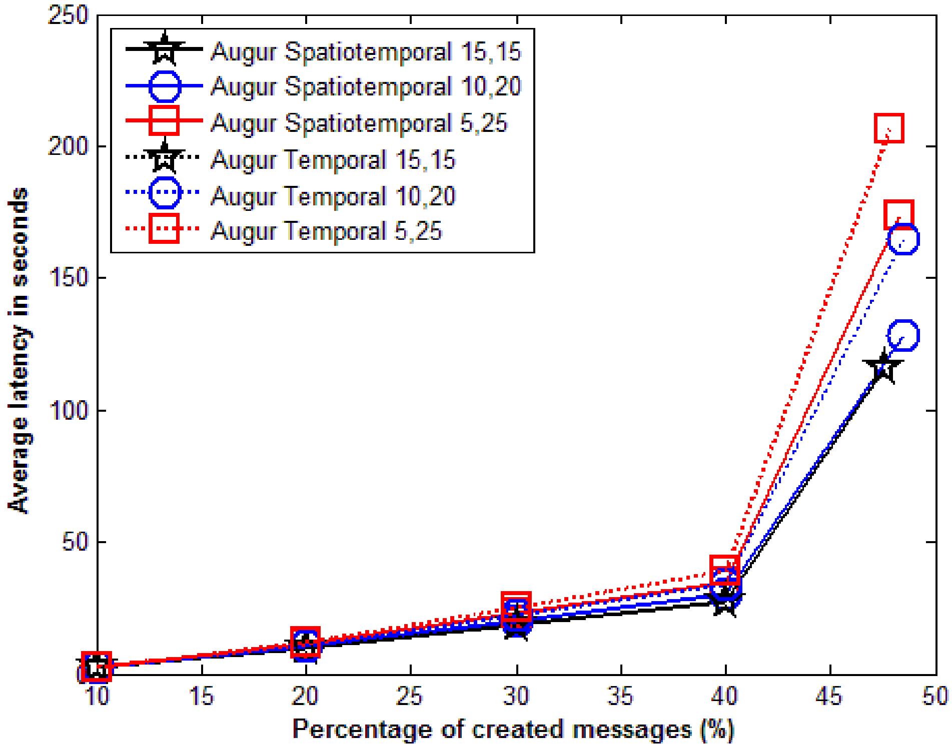


Figure 26. **Average latency for Augur Temporal and Augur Spatiotemporal for different speed intervals and for a message creation interval of 15 seconds.**

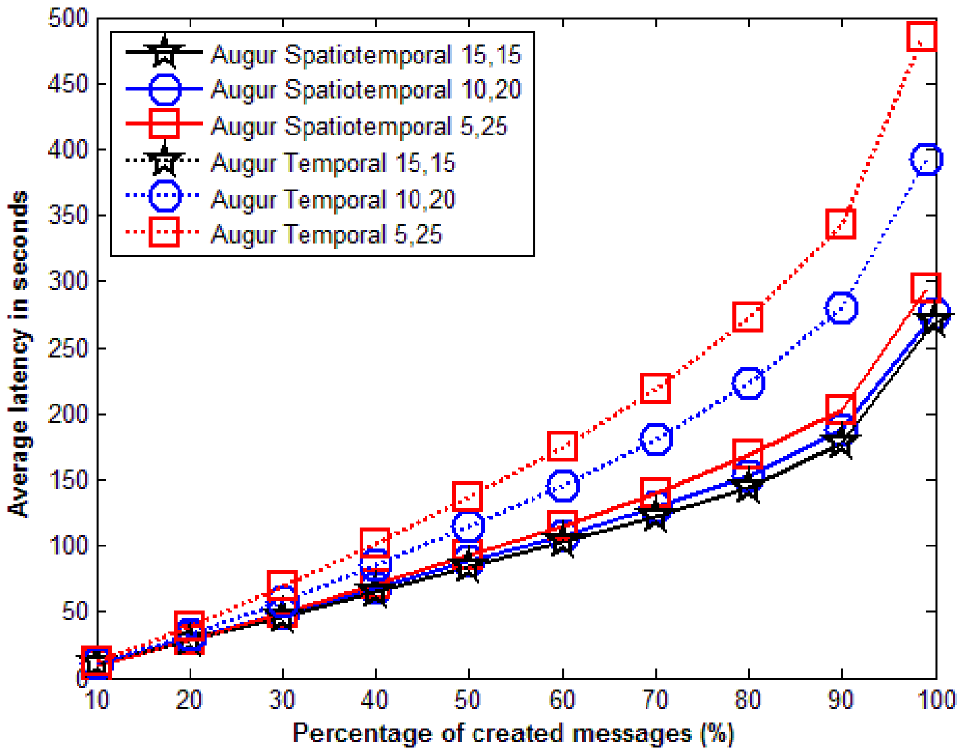


Figure 27. **Average latency for Augur Temporal and Augur Spatiotemporal for different speed intervals and for a message creation interval of 120 seconds.**

Another interesting metric to explore is the guess accuracy metric which presents the accuracy of the delays estimated by the two protocols in the delay broadcast phase. It also presents the efficiency of the relays made by the nodes during the simulation. To show this metric all characteristics of the relayed messages were tracked during the experiments. In particular, the broadcasted delay estimate, the time of broadcast and the time of relay were used to calculate the likely time of delivery of the message. This time is then compared to the actual delivery time of the message to the sink, the delay guess is considered correct if the difference between the two times falls within a [-5 5] seconds interval. Figure 28 shows the difference between the two versions of Augur in terms of the accuracy of the estimated delay. Notice that at constant speed both protocols were able to perfectly calculate the time needed for their messages to be delivered to the sinks and hence in this case the messages were granted fast routes. It is also noticed from the figure that the accuracy of the estimated delays decreases with the increase of the speed ranges for both protocols. The accuracy percentage of Augur SpatioTemporal remained clearly higher throughout the experiments. It is noticed that Augur Temporal suffers a huge drop in accuracy as soon as the movement breaks its perfect periodicity.

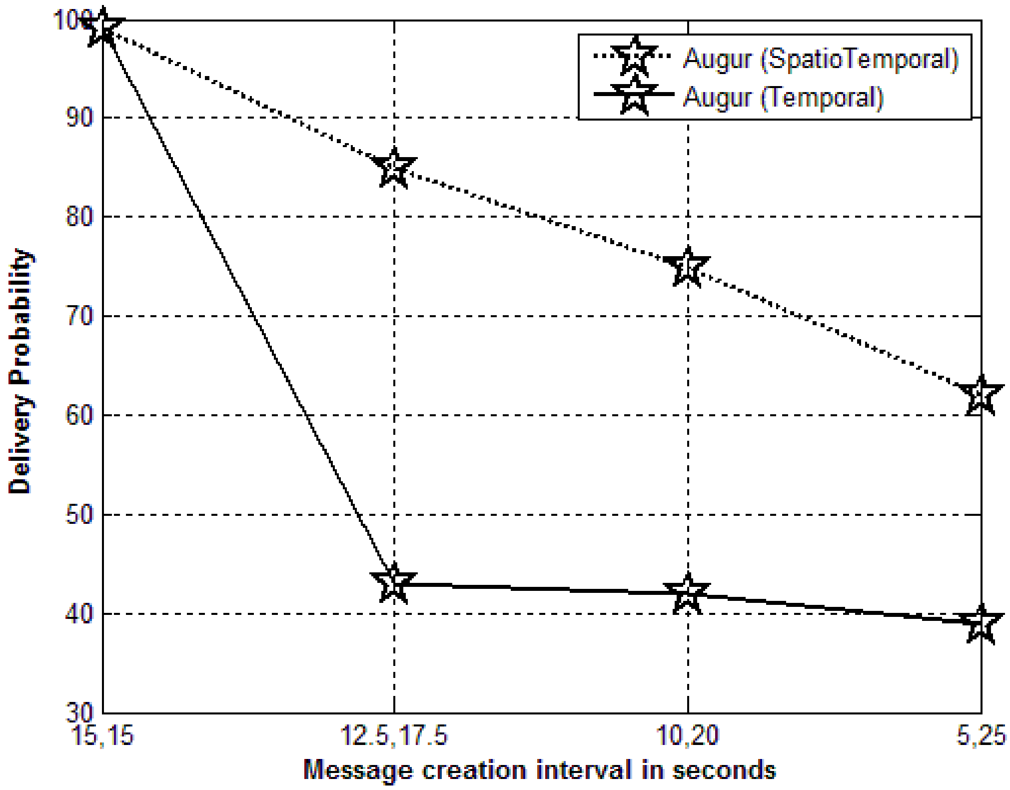


Figure 28. **Accuracy of the estimated delay made by Augur Temporal and SpatioTemporal during the simulations.**

Overall assessment of Augur Temporal and Augur SpatioTemporal with all the other protocols in the case of variable speeds

In this section, another comparative evaluation is conducted among Augur SpatioTemporal, Augur Temporal and the rest of the state of the art protocols. However, this evaluation aims to assess the protocols in the variable speeds scenario. Particularly, at speeds within a range of 10 m/s and 20 m/s. The evaluation metrics of interest are still the delivery probability, the overhead ratio and the average latency of the messages. The simulations are run for 20 days before the actual start of creating messages and generating reports to let ProPHET and Augur build some history information.

In order to choose the appropriate parameters for Spray and Wait for this scenario, a few experiments were done with different number of copies ranging from 1 to 30. Then, the results in terms of average latency and delivery probability are observed. It is obvious that more copies of the message will lead to a higher overhead ratio so the plot is not shown. Figure 29 shows the results obtained. Having only one copy of the message with no relays will be similar to a direct delivery protocol and will have the highest average latency, but this plot is kept as a reference. The case of 2 copies is not tested because it will act like first contact. The figure popping out from the circled zone is a zoom in to the end points of the plots. Notice from the figure that the increase in the number of copies results in an increase in the average delay and a decrease in the delivery probability. For this scenario, the value of 5 copies is chosen for SnW since this value recorded the least average delay with highest delivery probability.

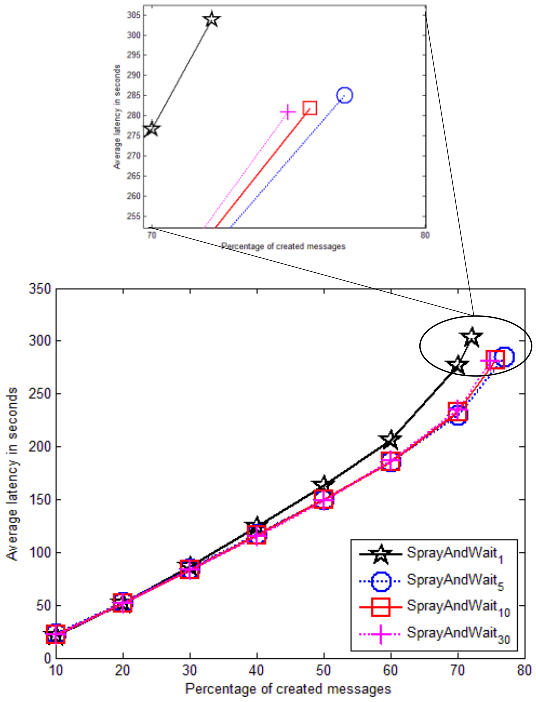


Figure 29. **Average latency for Spray and Wait protocol for varying number of copies 1, 5, 10, 20. The figure at the top is a zoom at the end points of the plots.**

Delivery probability

Figure 30 shows the delivery probability of the different routing schemes namely Augur SpatioTemporal, Augur Temporal, Spray and Wait, ProPHET, Epidemic, Direct Delivery and First Contact. The delivery probability of the protocols are presented for different message creation intervals varying from 7 to 120 seconds (where a smaller interval means higher traffic). The increase in the message creation interval resulted also in an increase in the delivery ratio of all the considered protocols. However, the delivery probability of Augur Spatiotemporal remained higher than the delivery probabilities of the other 6 schemes through all the experiments. The delivery probability of Augur becomes higher as the message creation rate slows starting with 0.22 at 7 seconds until reaching around 0.99 for a message interval of 120 seconds and above. Notice that at a very high message creation rate of 7 seconds all the protocols performed very similarly in terms of delivery probability with a value of 0.22 due to limited available channel capacity. It is also noticed that in the two scenarios of constant speeds and variable speeds DirectDelivery saturates at a delivery probability of 0.8 for a message creation interval 60 seconds and above. This saturation is due to the expiry of the messages' TTL before reaching a sink in addition to the presence of nodes that don't meet any sink through all the simulation time.

Since DirectDelivery doesn't do any relays, the messages of these nodes will be dropped due to TTL expiry or to create space in the buffer for newly created messages.

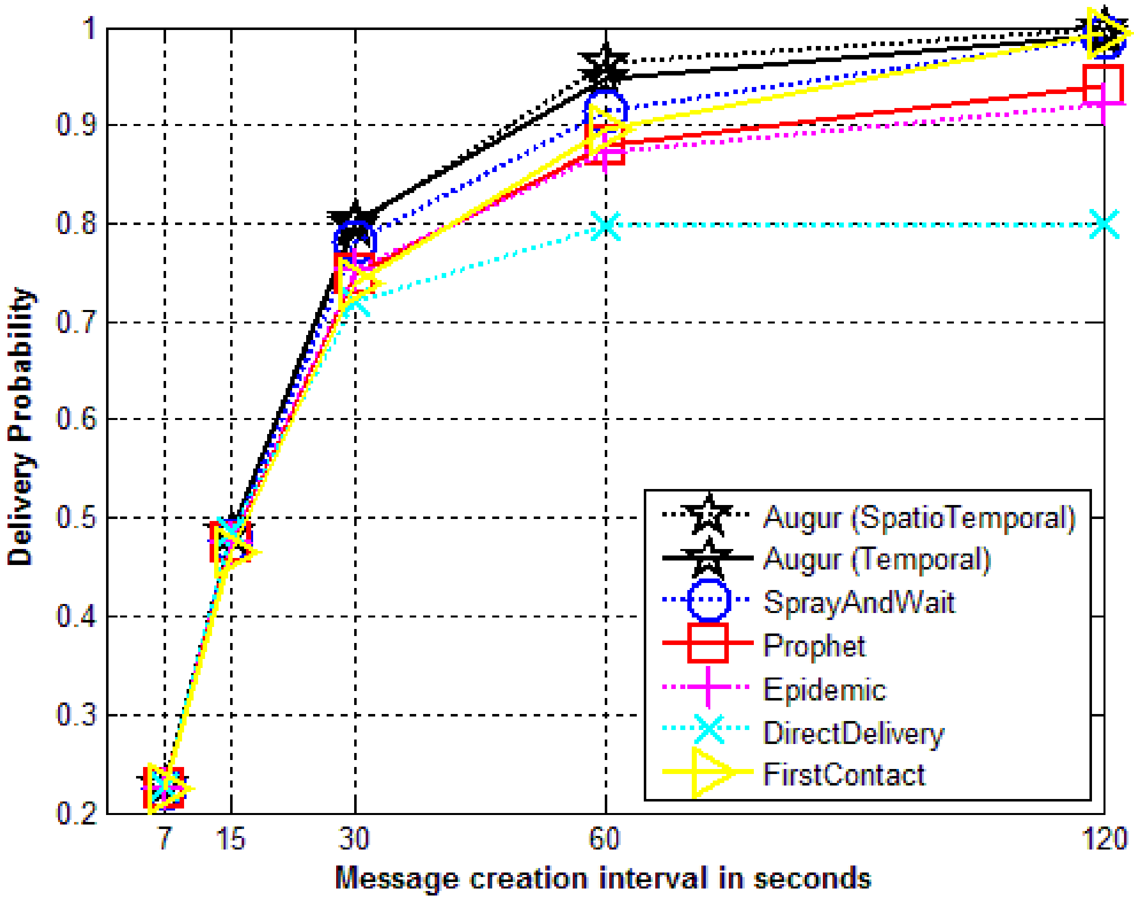


Figure 30. Delivery probability for considered protocols for varying traffic loads

In addition, it is noticed in the figures of the two scenarios that the delivery probability of all the protocols except DirectDelivery tend towards 1 or unity as the message creation interval is increased. In fact, the increase in the message creation interval or in other words the decrease in the message generation rate ensures less traffic in the network and less queued messages in the nodes' buffer. This means that packets are less likely to be unduly delayed from data sinks and consequently have higher chances to be delivered at the destination before their expiry or being dropped due to a full buffer.

Overhead ratio

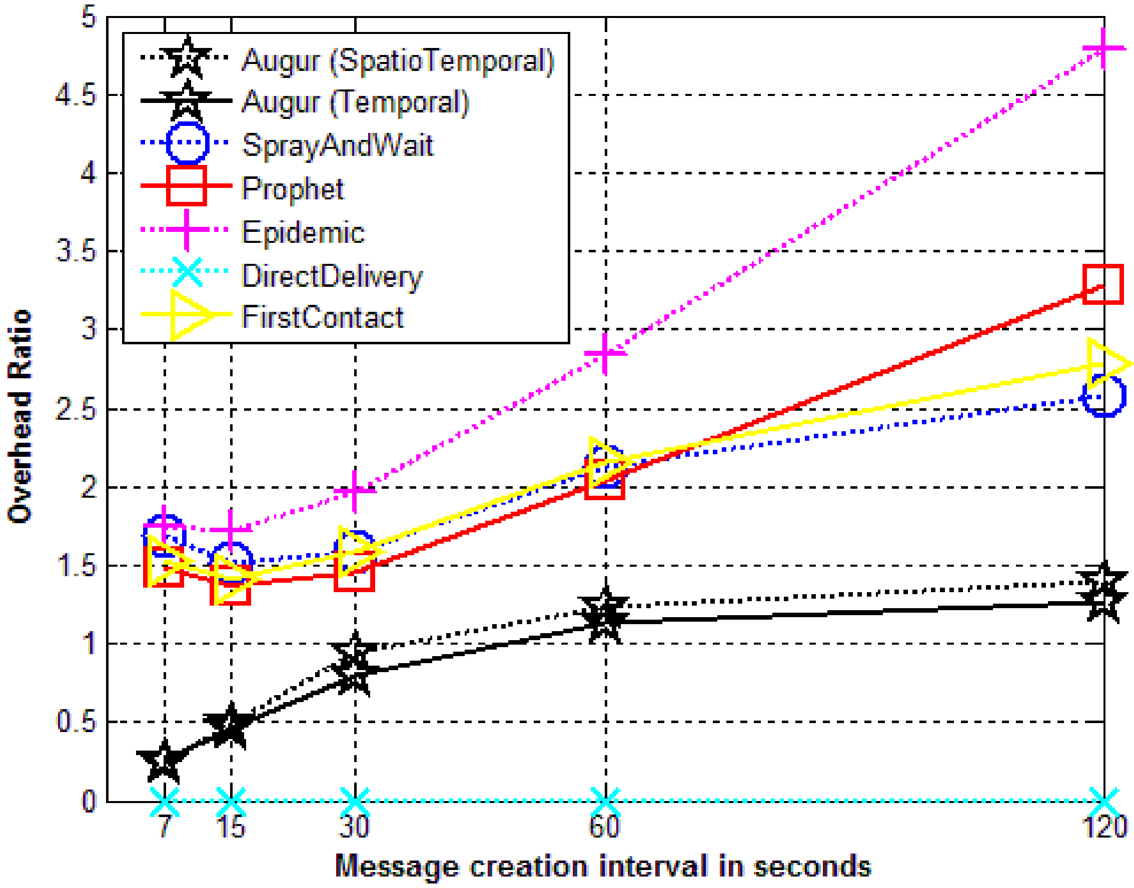
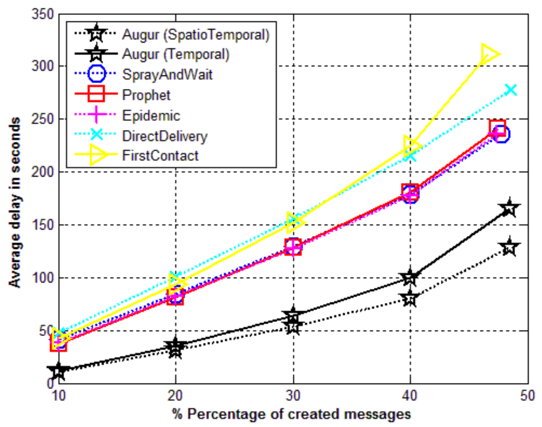


Figure 31. Overhead ratio for considered protocols for varying traffic loads

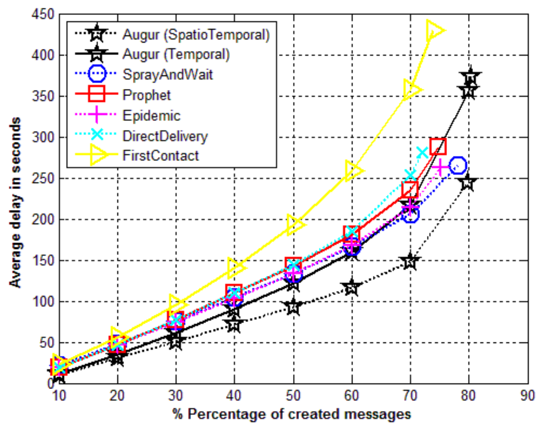
Figure 31 illustrates the influence of the message interval on the overhead ratio. Both Augur protocols performed significantly better than the rest of the protocols in terms of overhead ratio. However, Augur Temporal performed about 0.15 less than the values of Augur SpatioTemporal for a message creation create of 30 seconds and above at the expense of less delivery probability. It is noticed that at a rate of 7 seconds Spray and Wait, ProPHET, Epidemic and FirstContact recorded an overhead ratio between 1.47 and 1.6, then at 15 seconds the values get lower before they increase again at message rate of 30 seconds and higher. At 7 seconds, the protocols relay high number of messages to each other and because of the fast message generation rate the nodes keep dropping messages out from their buffers to deal with new entries. The messages are counted as relayed, energy is consumed, but most of the relayed messages have never been delivered to the sinks. This explains the high overhead ratio. At 15 seconds the messages had more time to stay in the buffer and had the opportunity to be received by the sinks which made the overhead ratio lower. At higher message intervals, more spaces in the nodes' buffer are available which allows more messages to be relayed in the network with the same number of delivered message which explains the increase of overhead ratio. In this scenario, Augur SpatioTemporal reduces the overhead ratio by around 87% for low traffic rates and by 44% to 62% for high traffic rates.

Latency

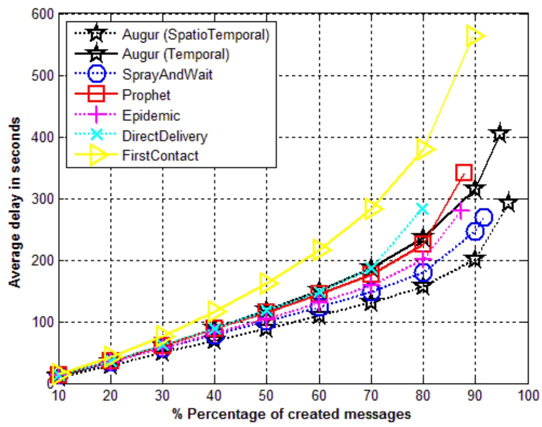
Next, the performance of the all protocols is analyzed in terms of average latency. Figure 32 shows the average latency of the protocols for different traffic loads, 15, 30, 60 and 120s respectively. The same visualization scheme of that in the previous sections is used. The first bin includes the fastest 10% of messages; the second one includes the fastest 20% of the messages, and so on. As a reminder, high traffic rates means low values of message interval and low traffic rate means high message interval. Furthermore, the right most point of each trace indicates the overall delivery probability for the corresponding protocol at every simulation.



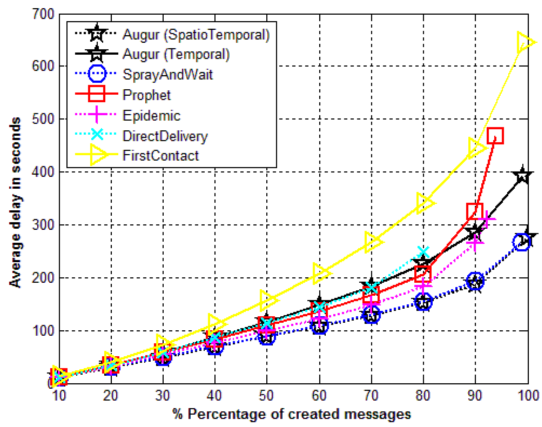
a) Message interval = 15 s



b) Message interval = 30 s



c) Message interval = 60 s



d) Message interval = 120 s

Figure 32. Average latency for considered protocols for varying traffic loads

In terms of latency, at high message rate Augur Temporal and Augur SpatioTemporal outperformed the other protocols. However, due to the spatial effect on the scenario the performance of Augur Temporal starts to decay from a message interval of 30s and more. In summary, Augur SpatioTemporal outperformed all the other considered protocols for all traffic loads by an average of 50 seconds.

From the figures, it can be observed that at low traffic rates Augur SpatioTemporal and binary Spray and Wait perform similarly in terms of average latency with an average difference around 2 seconds for Augur across all bins. In other words, Augur is able to deliver each message 2 seconds quicker than the closest other protocol candidate. As the speed of message creation increases the performance of the other protocols deteriorates and that the delay difference between Augur SpatioTemporal and the closest competitor starts to be clearly seen. It is noticed that Augur Temporal performs better at high traffic rates than at low traffic rates, however this is only due to the buffer being full most of the times during the simulation and hence less relays decisions are done. Consequently, the misestimated delays did not heavily affect the performance.

In order to explore the variability of the delay and better understand its distribution in the previous simulations, the standard deviation of the delay of the proposed Augur algorithms and the state of the art protocols are studied. Figure 33 shows the standard deviation values in seconds of all the considered protocols for message creation intervals ranging from 7 seconds to 120 seconds. Basically, a small standard deviation means that on average the delays during the simulation are close to the mean of all the recorded delays, and a large standard deviation means the delays are on average farther away from the mean. At an interval of 7 seconds, the standard deviations of Augur Temporal and Augur SpatioTemporal were around 25 seconds which is less than the other protocols which had a standard deviation around 60 seconds. The standard deviation increases at higher intervals until reaching saturation for message creation interval of 60 seconds and higher. It is noticed that the standard deviation for all the protocols increases with the increase in the message creation interval. However, the standard deviation value of Augur SpatioTemporal remained lower than the values of all other protocols during the experiments. This means that the delivery delays of Augur are more concentrated around the mean of every simulation. This can also be noticed from Figure 34 which shows another summary on the variability of the delays for the considered protocols. In this representation, boxplots are used to display the distribution of the delays based on minimum, first quartile, median, third quartile, and maximum values. The 4 subfigures a, b, c and d refer to the 4 different message creation intervals 15, 30, 60, 120 respectively. The experiments confirm the behaviour of the delays seen through the standard deviation. Through all the simulations Augur SpatioTemporal appears to have the lowest variability among the protocols. This can be noticed through the Range of the delay (the distance between the smallest value and the largest value) which is clearly the shortest among all the protocols with a value of 390 seconds. In addition, this is seen through the comparatively short IQR (InterQuartile Range- the difference between the 3rd quartile and 1st quartile) in all the scenarios which means that overall delays have high level of agreement with the mean. In particular, at high traffic or at 15 seconds it is noticed that 50% of the delay values were between 30 and 180 seconds. Also, it can be seen that the distribution of the delay values of Augur SpatioTemporal are relatively symmetric, the delays are evenly split at the median. These observations are particularly benefiting from Augur's ability to deliver the packets as fast as possible and also from Augur's buffering and forwarding strategy. Note that Augur forwards newly created messages first and drops oldest messages first when needed which gives priority to packets that ensure less delays over packets that will record large delays when delivered.

It can be stated from the above analysis that the Augur SpatioTemporal algorithm shows less variation and dispersion in terms of delay or latency. This gives more confidence to Augur SpatioTemporal over the other protocols in the studied scenarios.

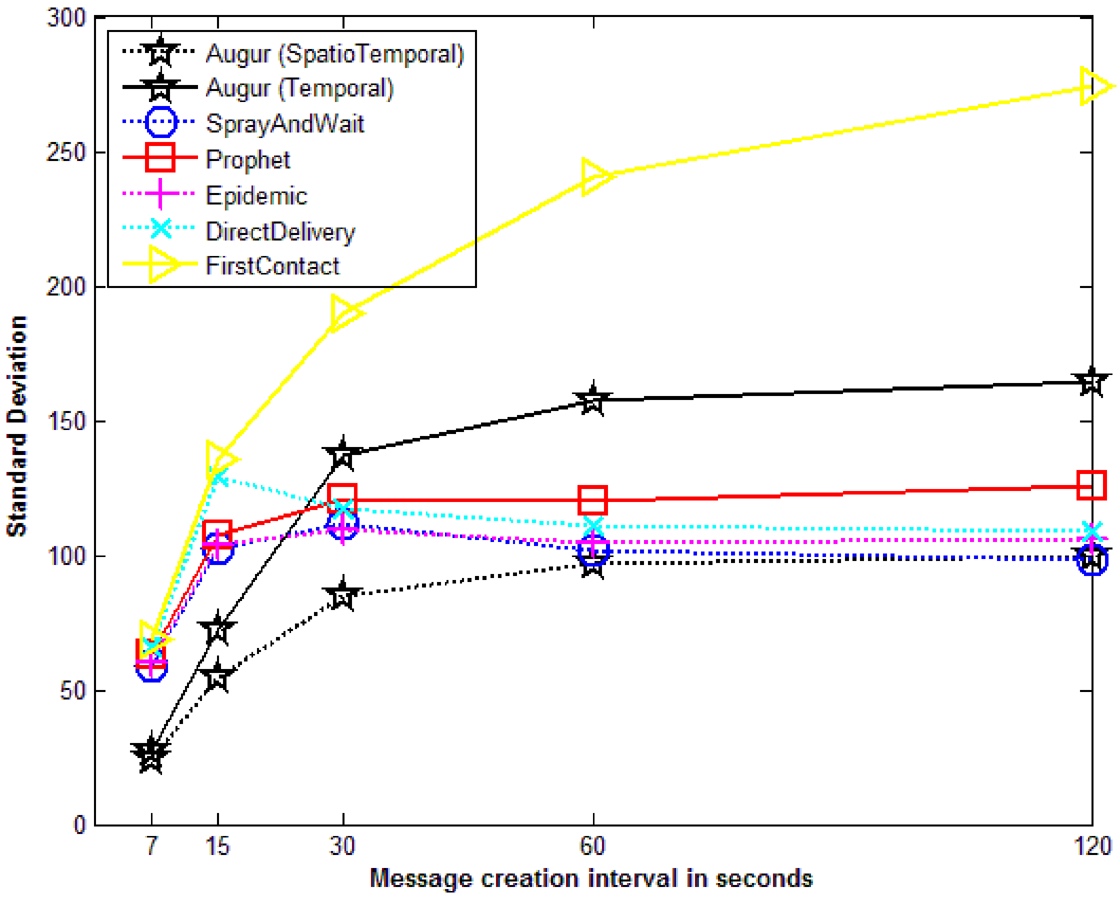
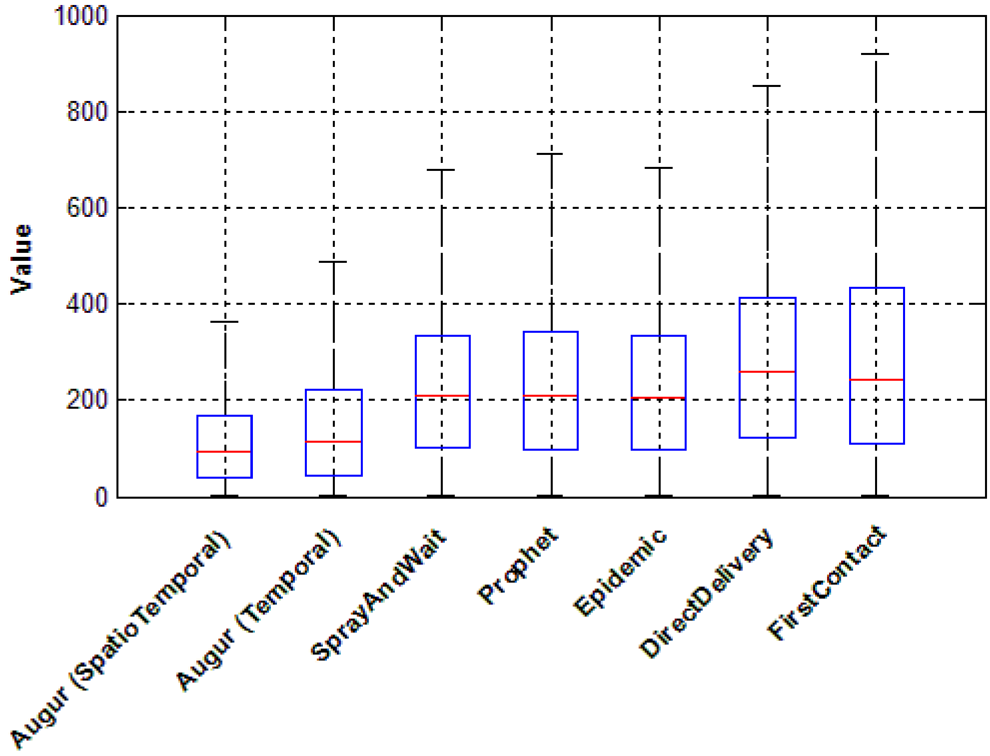
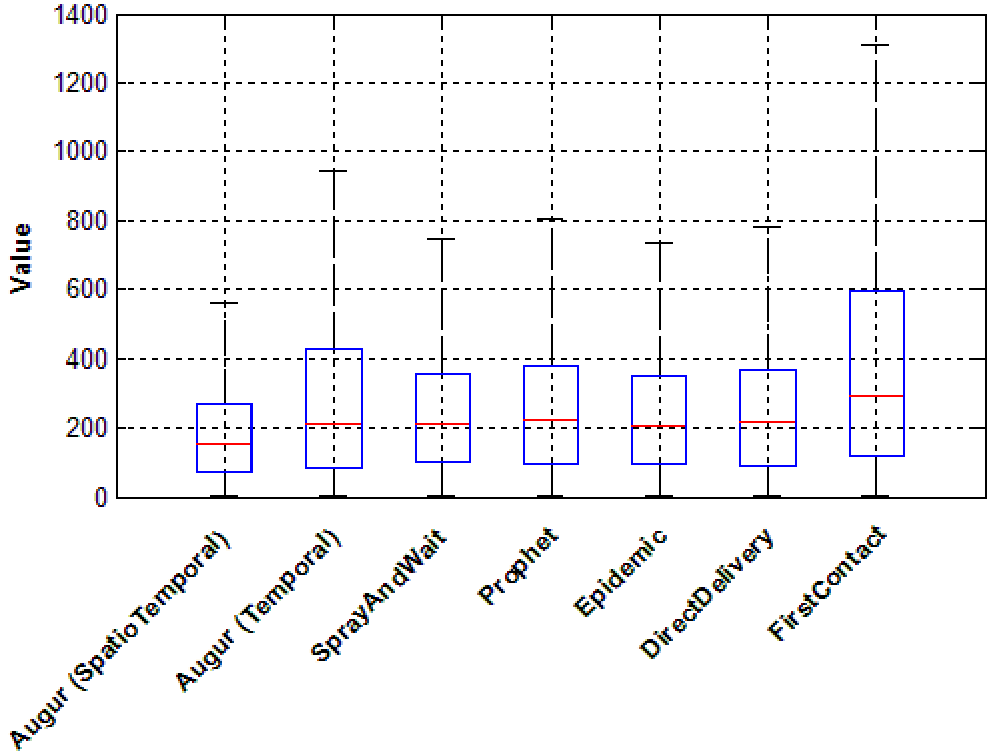


Figure 33. Standard deviation of delays for considered protocols for varying traffic loads

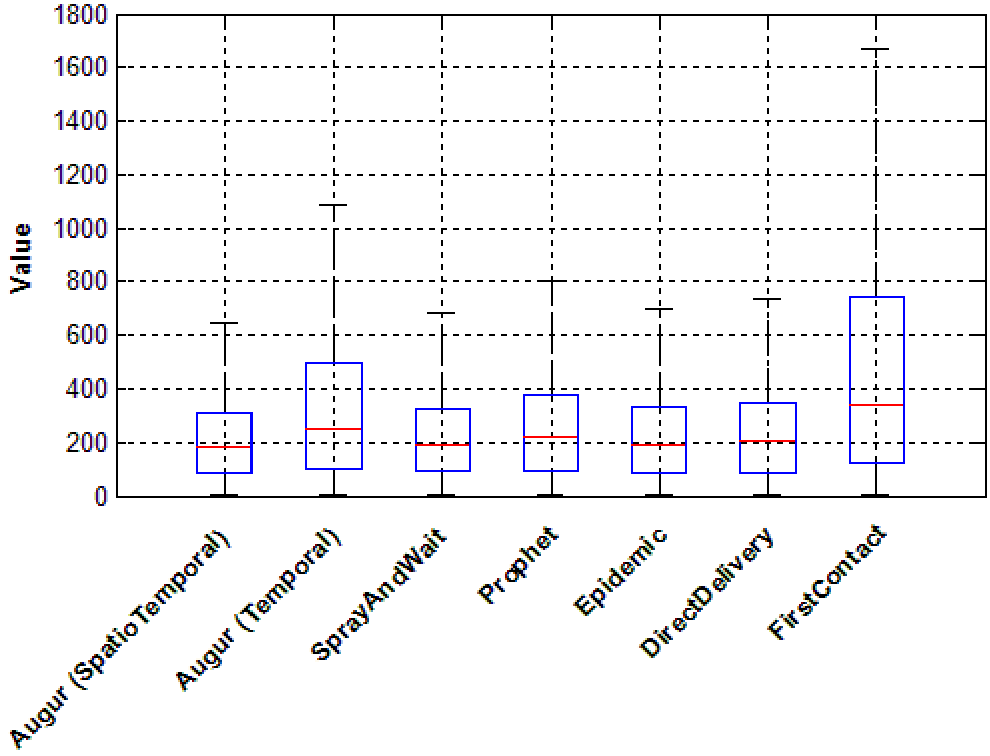
The analysis clearly shows that Augur SpatioTemporal gives the best results for delivery probability, overhead ratio and latency for all traffic loads. The superiority of Augur SpatioTemporal in the variable speeds scenario stems mainly from the use of spatiotemporal data of meeting and intermeeting occasions with sink nodes. The historical spatial and temporal information of the moving nodes gives the nodes the ability to approximate delays until likely future connections with sinks in addition to their durations.



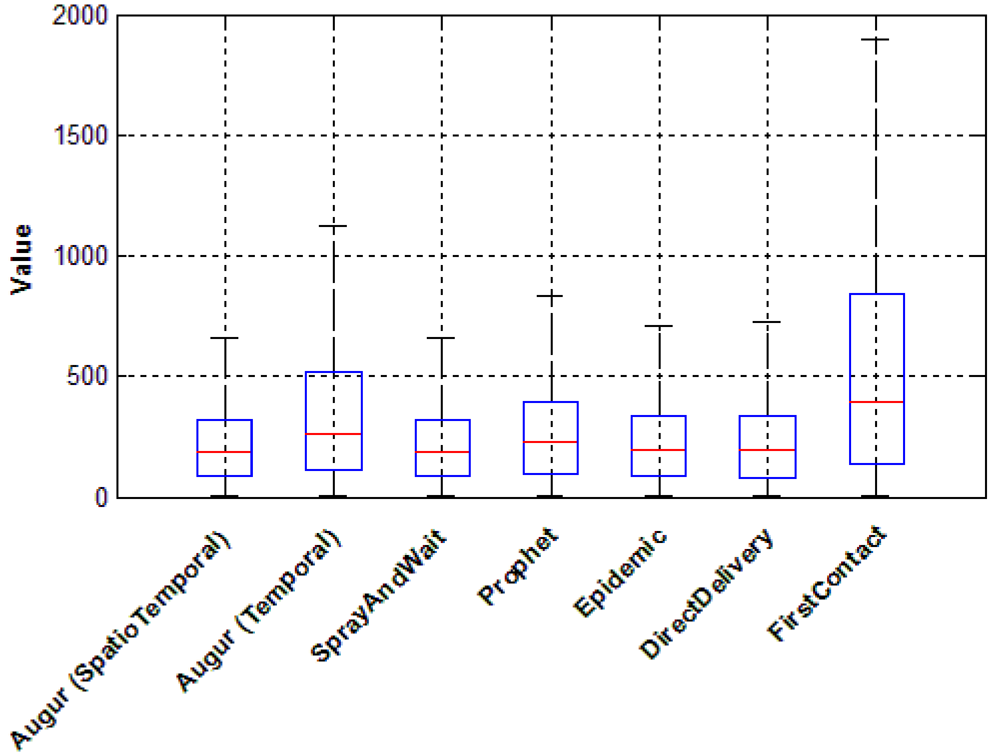
a) Message interval = 15 s



b) Message interval = 30 s



c) Message interval = 60 s



d) Message interval = 120 s

Figure 34. Delay distribution for considered protocols for varying traffic loads

6 Conclusion and Future Work

Delay in DTN is unavoidable because it is an opportunistic network, however very long delays in some time-critical scenarios limit the benefits that could be gained from this technology. Many routing protocols are proposed in various DTN applications. However, very few of these protocols address low delivery delays. In addition, based on our literature, none of the previous work exploited the time dependent mobility of mobile nodes to produce a delay aware routing protocol in DTN.

Since real life movement often exhibits some periodicity factor such as ones present in taxis, buses and human mobility, it has been shown in this work through Augur that with the use of spatial and temporal information of nodes, the required delay until the delivery of a particular message by a given node can be estimated and this improves the quality of service of routing protocols in DTN. In this work, two versions of Augur were introduced: Augur Temporal and Augur SpatioTemporal. The two versions differ in the amount of historical information which is used regarding the movement characteristics of the mobile nodes. Augur Temporal uses only the temporal dimension to build its decision metric, whereas the SpatioTemporal version makes use of both space and time dimensions.

The thesis presented findings through comparative evaluations and extensive simulation studies. Specifically, spatiotemporal information of DTN nodes improves the network performance when this information is incorporated in the design of the routing protocol. Augur algorithms outperforms the state of the art DTN protocols in terms of delivery probability, overhead ratio and latency in the implemented scenario.

Simulation results clearly showed that in constant speeds scenario the Augur Temporal routing protocol outperforms Spray and Wait, ProPHET, Epidemic, Direct Delivery and First Contact routing significantly in terms of the performance metrics especially the overhead ratio and the average latency. The first was reduced by 77% to 94% and by 73% to 88% at low and high traffic rates respectively, while the second was improved by 3% to 54% at low traffic rates and by 32% to 46% at high traffic rates.

In constant speeds scenarios, Augur Temporal and Augur SpatioTemporal perform similarly since the spatial dimension has limited effect on the scenario. On each day, the nodes visited the same location at exactly similar times. The simulations also showed that when the periodicity of the movement is broken or when the nodes move with variable speeds, the performance of Augur Temporal deteriorates gradually with the increase of the traffic load. This is due to the increase of the spatial effect in the scenario which Augur

Temporal does not consider in its routing decision metric. Additionally, the performance of Augur Temporal decreases with the increase of the speed range intervals in terms of delivery probability, overhead ratio and average latency.

In variable speeds scenario, simulation results showed that Augur SpatioTemporal outperforms all the considered protocols in this thesis in terms of all the performance metric at both high and low traffic loads. Specifically, Augur SpatioTemporal was able to reduce the average latency by up to 60%. The superiority of Augur SpatioTemporal in all the scenarios stems mainly from the use of the historical spatial and temporal data of the moving nodes which gives the ability to estimate future connections with sink nodes and consequently estimate the needed delay until message delivery.

Augur SpatioTemporal is a more advanced protocol than Augur Temporal at the expense of more information storage and computations, but gives superior results for semi-periodic movements.

Spatio-temporal algorithms perform better in partly dynamic environments (same route variable timing). In these types of movement, historical space and time data can be exploited to extract significant behavioural preferences to successfully approximate future actions. This is why Augur performs well in the bus movement scenarios. However, highly dynamic systems are unpredictable, and so neither version of Augur would be expected to perform well.

This work investigated Augur in the context of a favourable system scenario with features such as fixed stationary destination nodes and also regular bus trajectories. An interesting direction in this research is to remove these constraints and further explore the performance of Augur in real data traces and more complex movement. This could require adding a probability measure to the delay metric and also a weighted moving average delay where larger credibility weights are given to the more recent events. Also, future work can further study the mobility influence on the performance of Augur and explore other evaluation metrics such as the amount of processing, energy and memory needed by Augur to deal with the saved historical data.

7 References

[1] W. Zhao, M. Ammar, E. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks,” *Proceedings of ACM MobiHoc*, (New York, NY, USA), pp. 187–198, ACM Press, May 2004.

[2] P. Corke, T. Wark, R. Jurdak, Wen Hu, P. Valencia, and D. Moore, "Environmental wireless sensor networks," *Proceedings of the IEEE* , vol.98, no.11, pp.1903-1917, 2010.

[3] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol.44, no.11, pp.134-141, 2006.

[4] M.J. Khabbaz, C.M. Assi, and W.F. Fawaz, "Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol.14, no.2, pp.607-640, 2012.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, “Impact of human mobility on the design of opportunistic forwarding algorithms,” *Proceedings of IEEE INFOCOM*, 2006.

[6] J. Leguay, T. Friedman, V. Conan, “DTN routing in a mobility pattern space,” *Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN)*, 2005.

[7] A. Balasubramanian, B. N. Levine, and A. Venkataramani, “*DTN Routing as a Resource Allocation Problem”* Technical Report 07-37, *UMass Amherst*, 2007.

[8] P. Sommer, B. Kusy, A. McKeown, and R. Jurdak, “The big night out: experiences from tracking flying foxes with delay tolerant wireless networking,” *5th workshop on Real-World Wireless Sensor Networks (RealWSN)*, pp.15-28, 2014.

[9] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol.8, no.1, pp.24-37, 2006.

[10] S. Yamamura, A. Nagata, and M. Tsuru, "Store-carry-forward based networking infrastructure: vision and potential," *2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp.594-599, 2011.

[11] K. Fall, "A delay-tolerant network architecture for challenged internets," *2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp.27-34, 2003.

[12] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," *2nd International Conference on Simulation Tools and Techniques*, 10 pages, 2009.

[13] N.I. Sarkar and W.G. Lol, "A study of MANET routing protocols: joint node density, packet length and mobility," *2010 IEEE Symposium on Computers and Communications (ISCC)*, pp.515-520, 2010.

[14] D.W. Kum, J.S. Park, Y.Z. Cho, and B.Y. Cheon, "Performance evaluation of AODV and DYMO routing protocols in MANET," *2010 7th IEEE Consumer Communications and Networking Conference (CCNC)*, pp.1-2, 2010.

[15] S. Mittal and P. Kaur, "Performance comparison of AODV, DSR and ZRP routing protocols in MANET'S," *International Conference on Advances in Computing, Control, & Telecommunication Technologies*, pp.165-168, 2009.

[16] R.S. Mangrulkar, and M. Atique, "Routing protocol for delay tolerant network: a survey and comparison," *2010 IEEE International Conference on Communication Control and Computing Technologies (ICCCCT)*, pp.210-215, 2010.

[17] R. Jathar, A. Gupta, "Probalistic Routing using Contact Sequencing in Delay Tolerant Networks", in *IEEE Explorer*,2010.

[18] A. Mathurapoj, C. Pornavalai, "Fuzz-Spray: Efficient Routing in Delay Tolerant ad-hoc network based on Fuzzy Mechanism", *Proceeding of FUZZ-IEEE 2009*, pp. 104-109, 2009.

[19] H.S. Bindra and A.L. Sangal, "Performance comparison of RAPID, epidemic and prophet routing protocols for delay tolerant networks," *International Journal of Computer Theory and Engineering*, vol. 4, no. 2, pp.314-317, 2012.

[20] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine, "MaxProp: routing for vehicle-based disruption-tolerant networks," *Proceedings of IEEE INFOCOM*, pp.1-11, 2006.

[21] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing, ,* vol.6, no.6, pp.606-620, 2007.

[22] M. Musolesi and C. Mascolo, "Spatio-temporal communication primitives for delay tolerant systems." *3rd Minema Workshop*, pp.1-5, 2006.

[23] G.S. Thakur, U. Kumar, H. Helmy, W.J. Hsu, "On the efficacy of mobility modeling for DTN evaluation: analysis of encounter statistics and spatio-temporal preferences," *7th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp.510-515, 2011.

[24] E. P. Jones, P. A. Ward, "Routing strategies for delay-tolerant networks," *Submitted to ACM Computer Communication Review*, 2006.

[25] A. Galati, T. Bourchas, S. Siby, S. Frey, M. Olivares, and S. Mangold, "Mobile-Enabled Delay Tolerant Networking in Rural Developing Regions," *in IEEE GHTC*, San Jose, CA, USA, October 2014, pp. 699–705.

[26] M. Demmer, "A Delay Tolerant Networking and System Architecture for Developing Regions," *PhD. Dissertation, University of California,Berkeley*, 2008.

[27] R. Y.Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, "Turning the postal system into a generic digital communication mechanism," *Proceeding of SIGCOMM*, September 2004.

[28] M. Doering et al., "DTN Routing in Urban Public Transport System," *Proceeding of CHANTS'10*, pp.55-62, Sep. 2010.

[29] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41,pp. 128–136, June 2003.

[30] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, "Pocket switched networks and human mobility in conference environments," *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05)*, pp. 244–251, August 2005.

[31] Q. Li, D. Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks," *Proceedings of ACM MobiCom*, pp. 44–55, August 2000.

[32] K. Peters, A. Jabbar, K. Egemen, E. K. Cetinkaya, and J. P. G. Sterbenz, "A geographical routing protocol for highly-dynamic aeronautical networks," *Proceeding of the IEEE Wireless Communication Networks. Conf. (WCNC'11)*, 2011, pp. 492–497.

[33] D. Marasigan, P. Rommel, "MV routing and capacity building in disruption tolerant networks," *Proceedings of IEEE INFOCOM*, vol. 1, pp. 398–408, March 2005.

[34] W. Zhao, M. Ammar, E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," *Proceedings of IEEE INFOCOM*, vol. 2, pp. 1407–1418 vol. 2, April 2005.

[35] J. N. Al-Karaki, A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[36] H. Jun, M. H. Ammar, E. W. Zegura, "Power management in delay tolerant networks: A framework and knowledge-based mechanisms," *Proceedings of IEEE SECON*, September 2005.

[37] Kaviani, M., Kusy, B., Jurdak, R., Bergmann, N., Zhao, K. and Liu, V., "Delay Tolerant Routing Protocols for Energy-Neutral Animal Tracking," *Proceedings of the 3rd International Workshop on Energy Harvesting & Energy Neutral Sensing Systems* (pp. 9-14). ACM.

[38] J. J. P. C. Rodrigues, "*Advances in Delay Tolerant Networks (DTNs), Architecture and Enhanced Performance*", Woodhead Publishing Series in Electronic and Optical Materials: Number 67, Elsevier.

[39] F. Alnajjar, and T. Saadawi, "Performance analysis of routing protocols in delay/disruption tolerant mobile ad hoc networks," *10th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications*, pp.407-417, 2011.

[40] R H. Frenkiel, B.R Badrinath, J. Bores, R D. Yates, "The infestations challenge: balancing cost and ubiquity in delivering wireless data," *Proceeding of IEEE personal Communications*, Vol. 7,no 2, pp. 66-71,2000.

[41] M. Grossglauser, D.N.C Tse, "Mobility increases the capacity of adhoc wireless networks," *Proceeding of IEEE/ACM Transactions on Networking*, voI.10, pp.477-486,2002.

[42] R. C. Shah, S. Roy, S. Jain, W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," *Proceedings of Sensor Network Protocols and Applications*, pp. 30–41, May 2003.

[43] D. Nain, N. Petigara, H. Balakrishnan, "Integrated routing and storage for messaging applications in mobile ad hoc networks," *Mobile Networks and Applications (MONET)*, vol. 9, pp. 595–604, December 2004.

[44] T. Small, Z. J. Hass, "Resource and perfonnance tradeoff's in delay-tolerant wireless networks," *Proceeding of the ACM SIGCOMM workshop on delay-tolerant networking (WDTN'05)*, pp.260-267,Aug 2005.

[45] T. Small, Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN'05)*, pp. 260–267, August 2005.

[46] K. A. Harras, K. C. Almeroth, E. M. Belding-Royer, "Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparse mobile networks," *Proceedings of IFIP-TC6 Networking*, vol. 3462, pp. 1180–1192, May 2005.

[47] J. A. Davis, A. H. Fagg, B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," *Proceedings of Int'l Symp. on Wearable Computers (ISWC'01)*, pp. 141–148, October 2001.

[48] R Ramnathan, R. Hansen, P. Basu, R. R Hain, R Krishnan, "Prioritized Epidemic Routing for Partially Connected Ad Hoc Networks", *Proceeding of ACM MobiOpp'07*, 2007.

[49] T. Spyropoulos, K. Psounis, C. S. Raghvendra, "Spray and Wait Efficient routing in intermittently connected Networks," *Proceeding of Mobile Computer and Communication review*, Vol. 7, no. 3, July 2003.

[50] T. Spyropoulos, K. Psounis, C. S. Raghavendra, "Spray and wait: Efficient routing in intermittently connected mobile networks," *Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN)*, 2005.

[51] T. Spyropoulos, K. Psounis, C.S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," *Proceeding of IEEE Fifth Int'l Conf. Pervasive Computing and Comm. Workshop (PERCOMW)*, pp. 79-85, 2007.

[52] T. Spyropoulos, K. Psounis, C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *Proceeding of Transactions on Networking*, 2007.

[53] D. Anzaldi, "*ORWAR: a delay-tolerant protocol implemented on the Android platform,*" M.S. thesis, *Linköping University*, 2012.

[54] P. Cheng, K. Lee, "GeoDTN+Nav: A Hybrid Geographic and DTN Routing with Navigation Assistance in Urban Vehicular Networks," *Proceeding of ISVCS*, 2008.

[55] F. Khadar, T. Razafindralambo, "Perfonnance Evolution of Gradient Routing Strategy for Wireless Sensor Network," *Proceeding of the IFIP networking*, vol 5550, pp.535-547, May 2007.

[56] G. Sandulescu, S. Nadjm-Tehrani, "Opportunistic DTN Routing with Window-aware Adaptive Replication," *Proceeding of AINTEC' O8*. 2008.

[57] Yi Xian, Chin-Tser Huang, and Jorge Cobb, "Look-Ahead Routing and Message Scheduling in Delay-Tolerant Networks," *Proceeding of The 35th IEEE Conference on Local Computer Networks (LCN 2010)*, pp. 40-43, October 2010.

[58] A. Sidera and S. Toumpis, "DTFR: A geographic routing protocol for wireless delay tolerant networks," *Proceeding of the 10th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop* 2011 (MedHocNet '11), pp. 33–40, June 2011.

[59] Ko, E., Park, H., and Yeom, I. , "A new event-driven network simulator for delay-tolerant networks (DTNs)," *Proceedings of The SIMUTools* (2010).

[60] Mah-Rukh Fida, M. Ali, A. Awais and A.S. Arsalaan, "DTN-RSim: An Event-Based Simulator for DTN," *1St ICCNIT*, July11-13, 2011.

[61] A. A.Lindgren and O.Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol.7 no.3, July 2003.

[62] "augur", Oxford Dictionary Online, Available: http://www.oxforddictionaries.com/definition/english/augur