Investigation of Delay Tolerant Network Routing Protocols with Energy Consumption Analysis

Farhat M. H. Alusta

Submitted to the
Institute of Graduate Studies and Research
in Partial Fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2018
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Ali Hakan Ulusoy
Co-Supervisor

Assoc. Prof. Dr. Gürcü Öz
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Gürcü Öz
2. Assoc. Prof. Dr. Muhammed Salamah
3. Assoc. Prof. Dr. Ali Hakan Ulusoy
4. Asst. Prof. Dr. Yıltan Bitirim
5. Asst. Prof. Dr. Mehtap Köse Ulukök

ABSTRACT

Delay Tolerant Networks (DTNs) are the results of the evolutions in mobile networks in which an end-to-end path may not exist. The main principle of DTN to route messages is store, carry and forward technique, where intermediate hosts store data to be transmitted until it finds an appropriate relay host to forward the message in the route towards its target. DTNs have numerous applications in ad-hoc networking such as life monitoring and crisis management. Several routing and forwarding protocols have been proposed among the past few years. Majority of them uses asynchronous message passing scheme. The primary difference between various DTN routing protocols is the amount of knowledge that they have available to route the message. Flooding protocols such as Epidemic and Spray and Wait (SaW) routing protocols do not use any information. Predictive protocols such as PRoPHET and MaxProp uses past encounters of hosts to expect their future suitability to transmit messages to its destination. Store, carry and forward technique of DTN routing protocols causes a lot of copies of a message in the networks which consuming hosts' resources like energy and buffer. The main challenge in DTN routing is how to increase delivery ratio of messages and consume less resources.

This thesis focuses on the routing issue in DTNs using limited resources and investigate the performance of four well-known DTN protocols which is Epidemic, PRoPHET, MaxProp and SaW with the metrics node's average remaining energy, number of dead nodes, delivery ratio, average latency and overhead ratio using the Opportunistic Network Environment (ONE) simulator. It has been observed that the performance of routing protocols has been affected by the changing of message generation interval, number of nodes, node's speed, buffer size, time to live and the message size. The simulation investigation results that the SaW protocol outperforms other protocols in terms of energy consumption whereas MaxProp protocol has the highest delivery ratio. In contrast, Epidemic results the worst performance.

Keywords: Routing Protocols, Delay Tolerant Networks, Opportunistic Network Environment, Performance Evaluation, Energy Consumption Analysis.

ÖZ

Gecikme Toleranslı Ağlar (DTN'ler), uçtan uca bir yolun mevcut olamayacağı mobil ağlardaki gelişmelerin bir sonucudur. İletileri yönlendirmek için DTN'nin esas ilkesi, depolanan taşıma ve iletme tekniğidir. Burada ara bilgisayarlar, iletiyi hedefe doğru bir rota içinde iletmek için uygun bir geçiş bilgisayarı bulana kadar iletilecek verileri depolar. DTN, yaşamı izleme ve kriz yönetimi gibi özel ağlarda çok sayıda uygulamaya sahiptir. Geçtiğimiz birkaç yıl içerisinde çeşitli yönlendirme ve iletme protokolleri önerildi. Çoğunluğu asenkron mesaj geçme şemasını kullanıyor. DTN yönlendirme protokolleri arasındaki temel farklılık, iletiyi yönlendirmek için kullanabilecekleri bilgi miktarıdır. Epidemic ve Sprey and Wait (SaW) yönlendirme protokolleri gibi taşan protokoller herhangi bir bilgi kullanmaz. PRoPHET ve MaxProp gibi öngörülen protokoller, gelecekteki uygun varış noktalarına ileti göndermek için bilgisayarların geçmiş karşılaşmalarını kullanır. DTN yönlendirme protokollerinin depolama, taşıma ve iletme tekniği, makinelerin enerji ve bellek gibi kaynaklarını tüketen mesajın birden fazla kopyasının üretilmesine neden olur. DTN yönlendirmesindeki ana zorluk, iletilerin dağıtım oranını nasıl artıracağı ve daha az kaynak tüketeceğidir. Bu tez, sınırlı kaynakları kullanan DTN'deki yönlendirme sorununa odaklanmakta ve düğümün ortalama kalan enerjisi, ölü düğüm sayısı, teslimat oranı, ortalama gecikme ve tepegöz oranı gibi ölçü birimleri, Opportunistic Network Environment (ONE) simülatörü kullanarak göstermektedir. Sonuçlardan yönlendirme protokollerinin performansının, ileti oluşturma aralığı, düğüm sayısı, düğümün hızı, arabellek boyutu, yaşama süresi ve ileti boyutunun değiştirilmesinden etkilendiği gözlemlendi.

Benzetim çalışmaları, SaW protokolünün diğer protokollerden enerji tüketimi açısından daha iyi performans gösterdiğini, buna karşılık MaxProp protokolünün en yüksek teslim oranına sahip olduğunu gösterdi. Bunun yanında Epidemic protokolü en kötü performansı sergiledi.

Anahtar Kelimeler: Yönlendirme Protokolleri, Gecikme Toleranslı Ağlar, Opportunistic Network Environment, Performans Değerlendirmesi, Enerji Tüketimi Analizi.

DEDICATION

This thesis work is dedicated to the sake of Allah, my creator and my master, and to my great teacher the prophet Mohammed (May Allah bless and grant him), who taught us the purpose of life.

To my parents, who have been a constant source of support and encouragement during the challenges of graduate school and life. I am truly thankful for having you in my life, who have always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve. I am grateful.

To my brothers, sisters and wife, to the long nights you spent helping me getting through this thesis, the long hours you spent encouraging me that it will all be over, to the times when it was impossible to continue, you were all standing by my side with your encouraging words and love, this work would not have been complete without all of you. I am lucky to have you all in my life. Thank you.

Not least of all, I owe so much to my friends for their undying support, their unwavering belief that I can achieve so much. Unfortunately, I cannot thank everyone by name because it would take a lifetime but, I just want you all to know that you count so much. It had not been for all your prayers and benedictions, it were not for your sincere love and help, and I would never have completed this thesis. So thank you all.

ACKNOWLEDGMENT

In the name of Allah, the most merciful, the most compassionate all praise be to Allah, the lord of the worlds; and prayers and peace be upon Mohamed his servant and prophet. First and foremost, I must acknowledge my limitless thanks to Allah, the ever-magnificent; the ever-thankful, for his help and bless. I am totally sure that this work would have never become truth, without his guidance.

I owe a deep debt of gratitude to our university for giving us an opportunity to complete this work. I am grateful to some people, who worked hard with me from the beginning till the completion of the present research particularly my supervisor and co-supervisor Assoc. Prof. Dr. Gürcü Öz and Assoc. Prof. Dr. Ali Hakan Ulusoy, who has been always generous during all phases of the research.

I would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the way of doing my thesis.

I also would like to express my wholehearted thanks to my family for their generous support they provided me throughout my entire life and particularly through the process of pursuing the master degree. Because of their unconditional love and prayers, I have the chance to complete this thesis.

I owe profound gratitude to my wife, whose constant encouragement, limitless giving and great sacrifice, helped me accomplish my degree. Without their support, this study would not have been possible. Last but not least, deepest thanks go to all people who took part in making this thesis real.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AODV	Ad-hoc On-Demand Distance Vector
DSR	Dynamic Source Routing
DTN	Delay Tolerant Network
EID	Endpoint Identifier
GUI	Graphical User Interface
ICMANET	Intermittently Connected Mobile Ad-hoc Network
MANET	Mobile Ad-hoc Network
MRG	Minimum Reception Group
ONE	Opportunistic Network Environment
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
SaW	Spray and Wait
SCF	Store, Carry and Forward
SV	Summary Vector
TTL	Time to Live
URI	Uniform Resource Identifier
VAN	Village Area Networking
VANET	Vehicular Ad-Hoc Network
WBAN	Wireless Body Area Network

Chapter 1

INTRODUCTION

1.1 Introduction

Mobile Ad-hoc Networks (MANETs) are a gathering of independent portable hosts which constitute a networking framework independent of any infrastructure and rely on remote wireless connectivity as a medium. All the nodes are autonomous and independent in the network and have the capacity of switching to different nodes and other devices within the ad-hoc network radius at any given time. Each node or device in the network acts as a router for the system and so information flows through the system with the assistance of each and every node to achieve its goals and objectives. Hence, every node can and may be used as a hop in the transfer of data packets and information within the system. Making sure that every node or host in the MANET system will constantly be able to keep up with the data transfer and request is one of the essential issues in the setting up and smooth flow of MANET systems. With a specific end goal to rectify and maintain such issues, some MANETs are confined to neighborhood remote hosts and work without anyone else’s input while others might be associated with larger networks [1].

MANETs can have networks that are detached and they are known as Intermittently Connected Mobile Ad-hoc Network (ICMANET). They are also referred to as Delay Tolerant Network (DTN) and this is because of the constraint of the conveyance zone as well as versatility. The DTN are described by their irregular network, variable or long postponement, high blunder rates and unbalanced information rate [2]. In these testing situations there are customary specially appointed directional conventions which act as routing protocols. For example, Ad-hoc On-Demand Distance Vector (AODV) [3] or Dynamic Source Routing (DSR) [4] don’t work serenely in DTN in light of completely associated way amongst source and goal is required for correspondence to be conceivable. To defeat this test, DTN conventions apply “Store, Carry and Forward (SCF)” systems for steering messages which collect and store data in the center interface bouncing cradle. These are supposed to keep these messages alive until they have achieved their goal [5]. However, because of hosts’ portability, recurrence of experiences and message transmission, the vast majority of host’s vitality in this sort of system is constantly drained.

The main procedure of all DTN routing protocols is to forward a copy of a message to a host and/or node that is directly connected to it. The node that receives the copy of the message will forward the message again. This procedure will be repeated until the message achieves its goal or the life time of the message expires. However, the SCF nature of DTN routing protocols increases the delivery ratio of the message to destination hosts. Many copies of messages are stored in numerous hosts which results in the consuming of the hosts’ energy [6]. The movement of nodes and forwarding unlimited copies of messages are the main reasons for the consumption of energy in DTNs. Networking requires energy for sending, receiving and storing messages, which leads to consume the energy and decrease hosts’ lifetime.

1.2 Summary of Contributions and Expected Outcome

The main aims and objectives of this thesis are summarized as follows:

- Give a brief survey of some DTN protocols.

- Understand the well-known routing protocols of DTN.
- Understand how to use the ONE simulator.
- Investigation of DTN routing protocols in terms of energy consumption.
- Investigate the performance of DTN routing protocols using the most important performance metrics.
- Compare the protocols using the metrics node's average remaining energy, number of dead nodes, delivery ratio, average latency and overhead ratio.
- Display the outcomes of various simulation runs of the destined network types in form of diagrams and tables.

1.3 Thesis Outline

The organization of this thesis is as follow: The reviewing of the architecture of DTN and its routing strategies are presented in Chapter 2. Then the well-known routing protocols of DTN are analyzed in Chapter 3. Chapter 4 presents a simulation setup and implementation results of node's average remaining energy, number of dead nodes, delivery ratio, average latency and overhead ratio. Additionally some results and explanations of energy consumption is explored. Investigations in relation to message size, number of nodes, speed of node, message generation interval, buffer size and Time to Live (TTL) with number of dead nodes and average remaining energy of node is also conducted. Finally, Chapter 5 provides conclusions as well as suggestions and recommendations for concepts and ideas regarding future investigations.

Chapter 2

DELAY TOLERANT NETWORK

2.1 Background

MANETs are wireless networks that are formed by a network of hosts. The main assumption of MANET is that the connection of end-to-end for all nodes exists. Although in reality this connection is not always available due to the fact that nodes are constantly moving. Another problem occurs in large areas where the hosts' density is not sufficient to maintain connectivity. To overcome this intermittent connectivity problem, a DTN is used [7]. A host in DTN essentially stores a packet and forwards a duplicate of it to another host when they are in contact. This process is repeated until the goal of the message is achieved or the TTL of the message expires. A traditional routing algorithm for searching a path from a source to a destination cannot be used in DTNs. The reason for this is because such paths are not constantly available due to discontinuous connectivity caused by moving nodes. However, by using the SCF approach, DTNs can tolerate the longer delays and prevent the loss of data [8].

2.1.1 Fundamental issues in DTN

The DTN fundamental properties have a low data transfer rate and high latency period due to the fact that hosts may have a low possibility of meeting each other within a long time interval. This causes a decrease in data delivery rate and increase in latency in data delivery. Long queuing delay, disconnection, limited longevity and resource constraints are other features of DTN. The primary difficulties of discontinuous connectivity include dynamically changing network paths and availability of low-quality network transient connections [9].

DTN have some properties that are not presented in traditional networks. Listed below are some its properties as reviewed by Khabbaz, et al. [10]:

• **Energy**

The movement of hosts and lack of a main power source connection is the reason for limited energy in DTNs. Smooth networking requires energy in terms of sending, receiving and storing information. Such activity results in the increased consumption of a node's battery life or power source.

• **Transmission reliability**

The successful confirmation and data delivery stability of routing protocols should be returning an acknowledgement from the destination to the source after receiving the message to be used later.

• **Buffer space**

In DTN, the buffer may store messages for a long period of time because of discontinuous connectivity until the next chance of exchange. In some cases, most information that do not reach their destination are dropped to avoid buffer overload.

• **Routing objective**

Maximizing the probability of message delivery and minimizing resource consumption are important routing objectives.

2.2 DTN Architecture

The architecture of DTN networks and other MANETs are different as shown in Figure 2.1 since DTN has one more layer known as the "bundle layer" which is an end-to-end message oriented overlay. It is implemented under the application layer and runs above the transport layer. Its protocol is a single unit of composite information data that is forwarded and exchanged in the DTN [10, 11]. This component provides a uniform view of the network, however, different protocols underneath may exist.

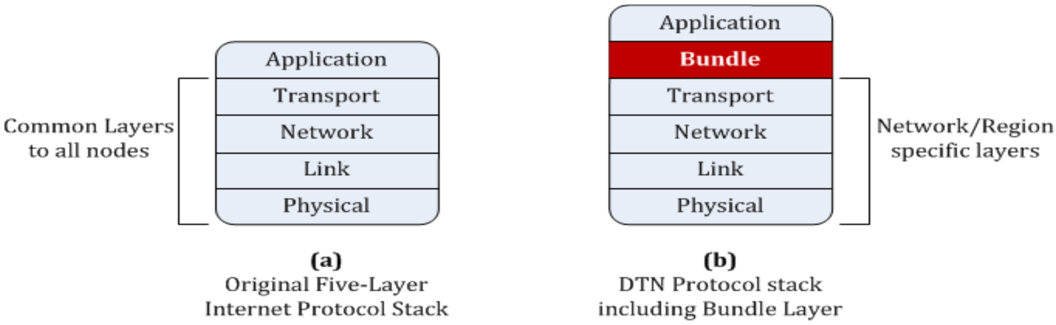

Figure 2.1: MANETs and DTN networks [10].

As mentioned above the intermittent connection of nodes is the main issue of DTN. So for that the DTN can present an optional mechanism called "custody transfer": DTN messages, called "bundles", are stored at intermediate DTN hosts in local databases until the next hop is reached, after which they are delivered whenever connectivity is available. Bundles may be maintained in databases until receiver's acknowledgment [12].

2.2.1 Binding, Naming and Addressing

In DTN, the existence of complete information in terms of names, routers and addresses are not possible all the time, since the services and hosts can appear, disappear and move dynamically. In many situations, the destination host after creation of the bundle may have changed which infers with a specific goal to find hosts in such an alterable area. Accordingly, use of the area, sensed values and parts as name quantities of hosts is critical and in the canonical DTNs Endpoint Identifiers (EIDs).

The EID notifies hiring the linguistic texture of Uniform Resource Identifier (URI), which concedes the DTNs endpoint. Using an EID, a host is expected to choose the Minimum Reception Group (MRG) of the DTNs endpoint which get a unique name by EID. To uniquely identify each host at least one EID is needed. The EID of a bundle processing structure is indicated by canonic EID, it can send bundles oriented to that EID from another host. The aim of naming technique is to connecting name attributes to the canonical EID [12].

Reassembly and Fragmentation

The reassembly and fragmentation of DTN are prepared to progress the effectiveness of bundle transfers by fully utilized contact volumes as well as by avoiding partially-forwarded bundles to be retransmitted.

Fall, et al. [11] stated that there are two types of fragmentation/reassembly of DTN:

a. Proactive Fragmentation

The DTN host may separate a chunk of application data into multiple smaller chunks and transmit each chunk like autonomous bundles. For this case, the destination are responsible to extract the smaller incoming bundle chunks and collect them again to the original larger bundle. Finally, this approach is basically used when connectivity volumes are predicted in advance.

b. Reactive Fragmentation

In the DTN graph the hosts are sharing an edge, and so when a bundle is only partially transferred, it may fragment a bundle cooperatively. For this case, the incoming bundle modified by the receiving bundle layer is able to identify it as a fragment, and thus forwards it naturally. The prior hop sender might learn that only a fraction of the bundle was received to the following hop, and send the remaining portions when next connectivity is obtainable. This approach is basically used when the fragmentation operation occurs after attempts of transmission has been executed.

2.3 DTN Applications

DTN can be utilized in cases whereby delay of data sensitivity does not exist and the primary aim is to receive as much of the created data as possible. Applications differ between the scientific and environmental as well as commercial and non-commercial applications. So therefore, some of the possible applications and performance projects of DTN are set forth.

2.3.1 Providing Residential Internet Access

In the case of a suburban or semi-countryside which required to link people to the Internet with insensitive applications of delay such as emailing, the building of a complete wireless Internet infrastructure or Internet cables extensions will be highly expensive to implement. The use of DTN overcomes this problem by collecting the data from this place to one or many places on the routes coming out of that place so that vehicles can transmit the data to the closest Internet gateway that might be in a neighbor town. This same process can be harnessed for incoming data. Access points can be installed on vehicles to collect data wirelessly, or it can be captured on any digital media such as CDs and then transported using vehicles. Pentland, et al. [13] has commercialized this idea with a system called DakNet.

2.3.2 Sensor Networks and Scientific Applications

There are many non-commercial DTN applications. Most of them are applied in ecological and environmental situations such as monitoring and tracking whales in oceans, wildlife animals, noise pollution regulation and lake water quality monitoring. The collection of data from a wide network of distributed sensors is one of the mutual applications of DTN.

Body Area Networks

Quwaider and Biswas [15] proposed Wireless Body Area Networks (WBANs) which operate as a “Store-and-Forward” protocol. WBANs take advantage of the mobility of human beings as it uses wearable nodes and devices which act as bridges and routers to facilitate hops. Sensor hosts used in WBAN which depend on low-power RF transceivers [16, 17] due to the clothing and postural body movements have good enough effects on the transmission of signals and data. The main idea of this technique is, decreasing the delay of end-to-end in DTNs and guarantee minimum storing delay through transmitting a message from the source host to the destination throughout different route.

DTN can be utilized in a set of other fields such as healthcare, education and economic efficiency. Moreover, the application of DTN was first implemented to facilitate communication and data transfer in outer-space networking. Hence, its advancement will also facilitate interplanetary activity with the uses of WBANs integrated into astronaut suits and gear.

2.3.3 Vehicular Access Networking

Vehicular networking is a fast developing field in the uses and application of DTNs. One of them is the virtual warning signs that alert the vehicle driver to caution him to take necessary precaution in order to avoid accidents or injury. Another concept of vehicular access networking is to supply Internet connection to other vehicles using roadside wireless stations.

2.3.4 Cellphones or Smartphones Implementations

Cell phones and other individual versatile devices can be engaged with DTN scenarios, particularly the ones that depend on social media. Moreover, they can provide numerous communication gateways through Bluetooth, WiFi, 3G, and USB cell networks. Typically these gadgets have specific as well as exclusive working frameworks which normal DTN applications cannot implement independently. Thus, to equip DTN properties for them an advancement effort is needed. Ntareme, et al. [18] has proposed a DTN bundle protocol called Bytewalla written in Java programming language for use on Android devices. The primary scenario concept for Bytewalla is: People conveying an Android cell phone traveling between African villages and acting as “data mules”. Such apps will greatly facilitate the data transfer and flow of information in rural areas, considering the fact that a large number of people use smartphones in those areas.

2.4 Routing in Delay Tolerant Networks

The primary issue with the smooth flow and use of DTN is the problem of unavailability of end-to-end connectivity. Using classic routing protocols will not give good performance, since the acknowledgement mechanisms of the TCP/IP protocol and its timer will fail. The movement of DTN hosts further aggravates the problem, and is especially difficult when such movement is irregular and is unpredictable. Hence, in such scenarios there is bound to be issues regarding lack of connectivity and uncertainty as to when such connectivity will be available [17].

In any network the most important factor is reliability. Various approaches of routing have been provided to work effectively within DTN environments. Many factors should be taken into account, for instance, increasing the delivery ratio, decreasing energy consumption and reducing the average delay of data transfer etc. Hence, every approach has its advantages and disadvantages and is usually suitable in particular cases as dictated by the situation at hand. There are two routing strategies presented in DTN as replication and knowledge.

2.3.1 DTN Strategies

Replication points to how many messages will be utilized in any given process and how to choose the strategy from these copies in addition to how to utilize them to submit the original message to the destination. DTN is characterized commonly when the connection between its hosts may not exist due to the uncertain or unforeseeable conditions. To overcome this, numerous strategies have a tendency to transmit various duplicates of each packet to rise its accuracy therefore a copy at least will be submitted, or to reduce delivery average latency so this is an explicit barter between performance and cost. Accordingly, the concept is that creating numerous duplicates increases the probability of message delivery, because one of these duplicates is sure to reach the final recipient. Although, this will give rise to the total overhead ratio thereby resulting in an upsurge of energy consumption and other resources. One of the cheapest techniques is, creating one duplicate of the packet but a fail to do that may result in the message being lost. It is for this reason that sending replicas or copies of the message to each node in the network is the most reliable mechanism. This method ensures that the message will not be lost in the case whereby only one host carrying the message fails to submit it.

Knowledge refers to how much information will be utilized to send the messages between the hops until it reaches the destination. Routing strategies are diverse in regards to the amount of information which DTN hosts require to send messages around the network. Another trade-off can be observed here regarding how storage of information inside the network leads to increased consumption of energy by the nodes that are storing them. A DTN host in this strategy uses knowledge to make decisions through the process of previous static rules which are configured when the strategy is prepared. Furthermore, all hosts follow those principles and so this will lead to basic executions which request few designs and control messages, because every rule is arranged early. The weak point to this strategy is that it cannot adjust to different networks or cases. Also a host may need to know every future mechanism of all nodes in the framework. Given that will offer exact data. This allows to get really productive using of framework resources through sending a packet through the hosts via the most suitable route. In the middle of these two extremes there is an area of qualities. For example, for several strategies the previous information is not needed, nevertheless, they will automatically learn it or the schedules information of future hosts may partially exist [19].

2.3.2 Carry, Store, and Forward Approach

The incoming messages in classic routing are kept in the present host buffer till the messages are sent to the following hop together with taking in account the decision of routing. In the case of the following hop connection below, the messages may be dropped. Moreover, the buffer capacity is not large enough, as messages may not stay in the buffer for long period of time. Whereas, the DTN nodes use the SCF technique.

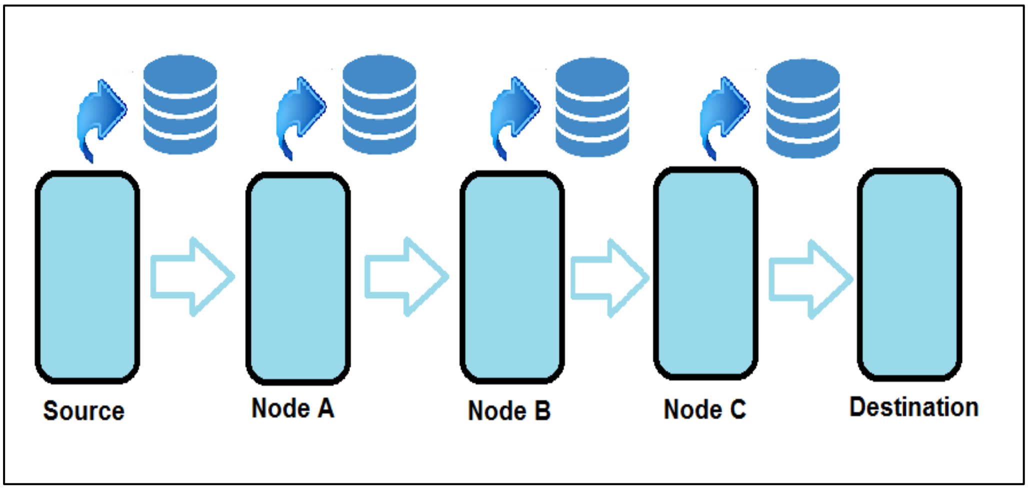

Figure 2.2: The DTN technique SCF

The concept of this SCF technique entails a DTN host carrying a bundle until it is able to connect to another DTN host and then transfers its messages which in turn carry it until it contacts another host. This process continues until the message reaches its final recipient. The chance of end to end connectivity with another DTN host in this process is very slim. In numerous cases like a window of chance may be short so it should be aforesaid. For instance, in Vehicular Ad-Hoc Networks (VANETs) if a cellular equipment in a vehicular comes next to another equipment and a transfer of some data bundles occurs. But before all the bundles can be transferred completely the cellular device moves away and this results in a loss of connectivity. The data transfer may require a given period of time to completely upload to the other node. The issue of this persistent storage and intermittent connectivity in intermediate DTN hosts [20] is clarified in Figure 2.2.

The DTN routing protocols as indicated by the routing strategy properties are classified into two primary classifications as forwarding protocols and flooding protocols.

The strategy of forwarding protocols entails sending one copy of the message from the source to the final receiver through intermediate hosts. In forwarding strategy, it is not required to replicate the data because each host trying to route a message throughout the network should know the network histogram at that given time so as to find the best route in order to reach the destination with lowest possible cost [21].

The flooding protocols strategy involves creating numerous duplication of a message and spreading the copies among the network to other hosts so that the message will reach its goal based on some attributes. As its name implies, the network is flooded with the copies of the message so as to increase the possibility of successful delivery to its final recipient. Thus, the protocol creates numerous copies of the message and route them to alternation hosts that carry and store it in the buffers till reaching its final recipient [21].

Figure 2.3 shows the DTN routing protocols classification. In addition each strategy comprises sub protocols that use various approaches which will be explained in detail in next chapter.

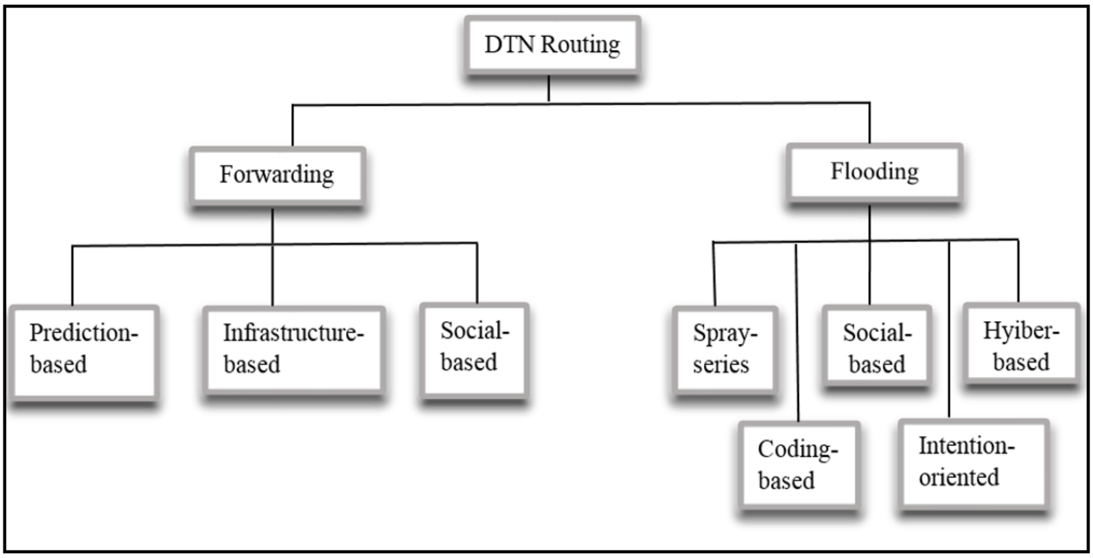

Figure 2.3: The DTN routing protocol classification

In Chapter 3 we will present in details four routing protocols which are PRoPHET and MaxProp that using forwarding strategy, and Epidemic and SaW that using flooding strategy.

Chapter 3

DTN ROUTING PROTOCOLS

3.1 Epidemic Routing Protocol

The Epidemic routing protocol supports the conceivable conveyance of messages to arbitrary goals with minimal suppositions concerning the basic topology as well as essential network connectivity. A discontinuous connectivity is actually needed to guarantee a conceivable message conveyance network. For the ad-hoc networks the protocol relies on the transitive transport of messages with achieving their destination. In addition, a buffer in every host keeps up messages which has emerged and moreover messages which is buffered for the benefit of all the rest of the hosts in the network [22].

Vahdat et al. [22] stated that the main objectives of Epidemic routing are:

1. Distribute messages efficiently into ad-hoc networks which are partially connected in a probabilistic manner.
2. Maximize the delivery ratio.

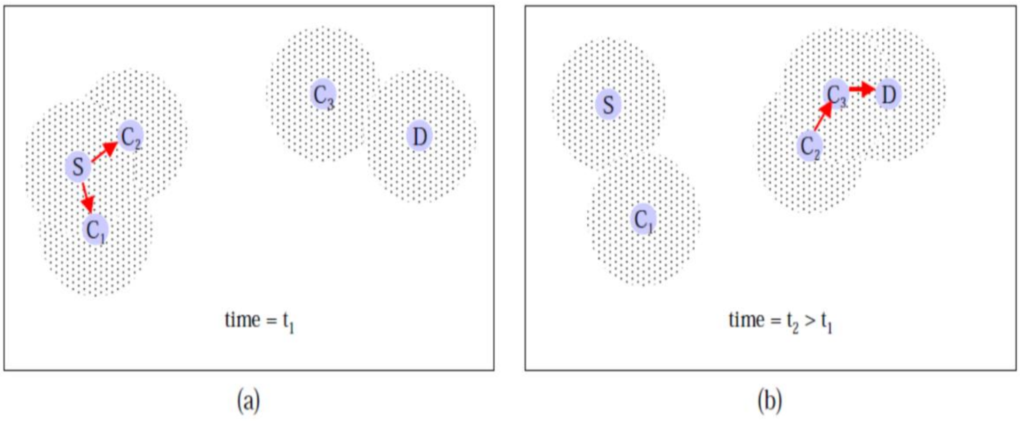

Figure 3.1: Epidemic strategy in high level [22].

Figure 3.1 describes Epidemic routing at a high level, with moveable hosts shown as dark circles identified by letters and their communication range appeared as a dotted circle reaching out from the source. In Figure 3.1(a), a source S, willing to transmit a message to a goal, D, yet no associated way is accessible from S to D. S sends its messages to the closest two hosts, C_1 and C_2, inside the direct correspondence range. After a given period of time, as presented in Figure 3.1(b), C_2 comes into coordinate correspondence run with another host C_3, and sends the message to it. C_3 is in a coordinate scope as D finally transmits the message to its goal. For adequacy, a hash table indexes a message's list, recognized by an unrivaled identifier associated to each message. A bit vector stores in every node known as summary vector which indicates which ingress in their local hash tables has adjusted. While not investigated here [23, 24], a "Bloom filter" would significantly diminish the space overhead connected with the Summary Vector (SV).

At the point when two hosts come into correspondence range of each other, the host with a smaller identifier starts an anti-entropy session with the host which has a greater identifier. To stay away from repetitive associations, each host keeps up a store of hosts that it has recently communicated with. Anti-entropy is not re-started with remote hosts that have been connected within a time duration.

During the process of anti-entropy, the two hosts exchange their SV to decide which messages stocked remotely have not been visible by the regional host. Consequently, every host at that time requires copies of messages which it has not observed yet. The extraditing node keep up total autonomy in determining whether it will accept a message. For example, it may discover that, it is unwilling to send messages larger than a permits bulk or bound for certain hosts [25].

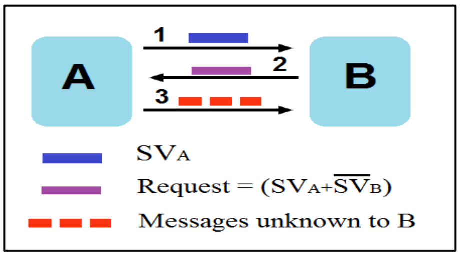

Figure 3.2: Two nodes, A and B, come into contact [22].

The Epidemic protocol SV exchange is shown in Figure 3.2. Node A contacts node B and launches an anti-entropy session. Firstly, A sends its summary vector, SV_A to B. SV_A is a representation of whole messages buffered in A. Following this, the summary vector SV_B of node B is compared with SV_A and node B transmits a request to host A that includes which points to the messages that host B wants in SV_A. That is, B decides the set difference between the packets buffered in A and the packets locally buffered in B. Then B sends a vector requesting these packets from A. Finally, A sends the requested packets to B. This procedure will be repeated when B contacts any new host. The algorithm of Epidemic protocol is presented in Appendix A.

3.2 PRoPHET Routing Protocol

In spite of the fact that the arbitrary way-point portability model is prevalent to use in assessments of mobile ad hoc protocols, real hosts are not likely to move around arbitrarily, but instead move in an anticipated manner based on repetitive patterns of behavior. For example, if a node has gone by an area a few times previously, most probably it will visit that area again. According to this idea Anders Lindgren et al. [26] has proposed PRoPHET routing protocol. PRoPHET is a forwarding probabilistic-based protocol utilizing the historical backdrop of associate experiences and transitivity to improve the probability of conveyance packet. To accomplish this, PRoPHET depends on a conveyance foreseeability, $P(a, b) \in [0, 1]$ as a metric of probability. This alludes to the probable likelihood that this host (a) will have the capacity to pass on a message to its goal (b). The attitude of PRoPHET and Epidemic protocols are the same at the point when two nodes comes in contact, where the SV are traded, including the conveyance predictability acknowledgement which keeps the nodes to update on the interior conveyance predictability vector in order to determine which packets are required from the other host.

There are three steps to calculate the delivery predictabilities [26]:

Updating the delivery predictabilities

The predictability metric of the nodes that come into contact will updated in each time using this equation where the initialization constant is $P_{init} \in [0, 1]$.

$$P(a, b) = P(a, b)_{old} + (1 - P(a, b)_{old}) \times P_{init} \tag{3.1}$$

Aging

At the point when two hosts do not comes in contact for a long time, this will reduce their opportunity to be likely transfer of packets between each other, thus, the protocol decreases the conveyance predictability values by aging them using this equation.

$$P(a, b) = P(a, b)_{old} \times \gamma^{k} \tag{3.2}$$

where γ is the aging constant $\in [0, 1)$, and k is the amount of time units that passed since the previous contact. The time unit must be set based on the predictable tardiness at the destination network.

Updating transitivity

If host b contacts host a frequently, and host a contacts host c considerably, then host c is a good relay host to transfer packets oriented for host b as well. The delivery predictability will be affected by this transitive property and so the protocol uses the following equation to update its transitivity.

$$P(a, c) = P(a, c)_{old} + (1 - P(a, c)_{old}) \times P(a, b) \times P(b, a) \times \beta \tag{3.3}$$

where $\beta \in [0, 1)$ is a scaling constant which determines the extent of the effect of the transitivity on the delivery predictability.

The algorithm of PRoPHET routing protocol is presented in Appendix A.

3.3 MaxProp Routing Protocol

DTNs can be set up on moving hosts like pedestrians or vehicles. Vehicles can stock considerable transport massive equipment and electrical supplies that may not be available to non-mechanized hosts. One of the disadvantage of vehicle based systems is, the hosts move more rapidly, diminishing the measure of time which may be required to establish connectivity and data transfer between nodes. Therefore, one constrained asset in a vehicle-based DTN is the length of time that hosts can exchange information between each other as they move. Capacity can be a limited resource too. It was as a result of this challenge that John Burgess et al. [27] proposed MaxProp routing protocol to address such situations. By using hop counts in packets as a mensuration of network resources and utilizing data which are distributed among the framework, MaxProp keeps a roster of prior alternation hosts to limit information from distributing two times to the same host.

MaxProp uses the likelihood of paths to hosts based on pervious information, acknowledgments, arrangements of earlier transfer hosts and a head begin for new packets. Figure 3.3 shows those techniques that are utilized to construct the stream of packets sent to different hosts and packets' stream to be dropped, where the priority of forwarding is to the packet that has less hops, and the priority of dropping is to the packet that has the most number of hops. MaxProp depends on organizing both these streams to convey the packets with lower transmission time and lower use of resource's capacity.

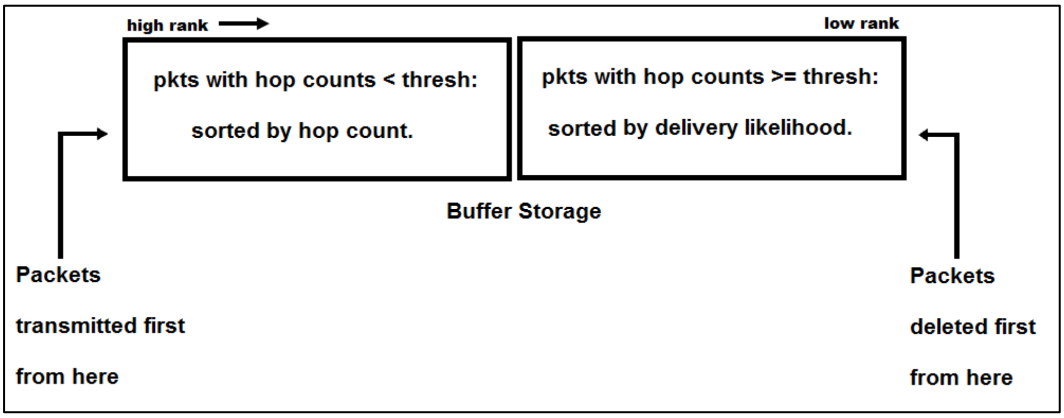

Figure 3.3: MaxProp protocol strategy [27].

In view of a cost predicted to each goal, the protocol arrange the list of the hosts' stored messages. The cost is an estimation of conveyance likelihood. Moreover, when the message is conveyed the data are utilized to inform all hosts. The new messages in MaxProp have a higher priority than the older messages. In addition to this, it likewise tries to obstruct accepting two duplicates of the same message.

Estimating Delivery Probability

Weights assigns to the routes that link hosts by MaxProp as:

Every host has a place within the network and also has likelihood of meeting or alternate hosts *Pab*. At first, this likelihood equal to 1 separated to the quantity of the remainder of hosts. Suppose that five hosts are in a region, the likelihood for every host to contact another host is *Pab*= 0.25. So this likelihood will increase by 1 every time that host *a* and host *b* are contacted, afterward this same technique employed to stabilize all probabilities. A present host gauges the costs to the rest of the hosts which know its probabilities, the cost is computed for each prospect route to the target *t* using the equation $x(i, i+1 \, . \, . \, . \, . \, . \, t)$, up to all hops in between.

The prospect route cost is determined by subtracting an amount from the likelihood that each contacting has happened as [27]:

$$x(i, i + 1, \ldots\ldots, t) = \sum_{c=i}^{d-1} [1 - (Pcc + 1)] \quad (3.1)$$

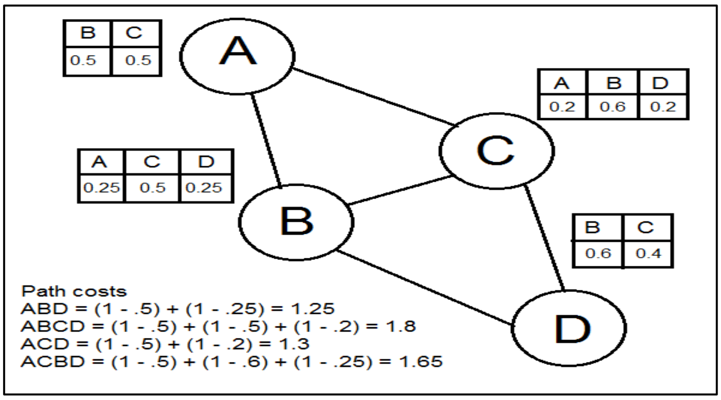

Figure 3.4: The path cost calculation of MaxProp protocol [27].

The target's cost is determined as the route that has the cheapest cost of every single available route. Figure 3.4 indicates that the route which is 1.25 is the suitable route from A to D. The algorithm of MaxProp is given in Appendix A.

3.4 Spray and Wait Routing Protocol

Spyropoulos [28] proposed the Spray and Wait (SaW) routing protocol which uncouples the amount of copies created per message. As a result of this technique the amount of transmissions performed will diminish from the network size. By spreading few duplicates each to a different alternation. This mechanism contains two phases:

Spray stage

Every host attempt to transmit a packet which will be spread in duplicates of packet around the network in the hope that some of those duplicates will reach other hosts which will act as routers and re-transmit them again as a relays until it reaches its destination.

Wait stage

The destination in DTN is not always handy, so if it is not every host which has a duplicate of packet as "Direct Transmission" (it tries to transmit the packet just to its destination).

SaW protocol combines the velocity of Epidemic protocol with provides immediate transmission. At the beginning, both SaW and Epidemic protocols spread duplicates of each message using similar procedure. But to guarantee that one of copies at least will be delivered to the final recipient quickly SaW spreads sufficient duplicates, after that it stops and permits other hosts which have duplicate to carry out direct transmission.

The author proposed another model of SaW in [29] that differs in terms of number of packet's copies that will spread in the network called binary SaW. It has the same process of the previous one but it is different at the point in which each host is permitted to use half of duplicates permitted for the message, and the rest is left when another host comes into contact, this process will be repeated until the host have only one copy which will keep it for the destination host. However SaW uncouples the number of transmission messages and needs large buffer capacity in each host.

Figure 3.5 illustrates the binary mode technique when the source host S initiates L packet duplicates and how it spreads the duplicates to other hosts, after which each relay host transmits half of its duplicates. At first contact, the host sends L/2 of duplicates. Secondly, it sends L/4. It continues this process until it has only one copy which will keep it for final destination. SaW algorithm is presented in Appendix A.

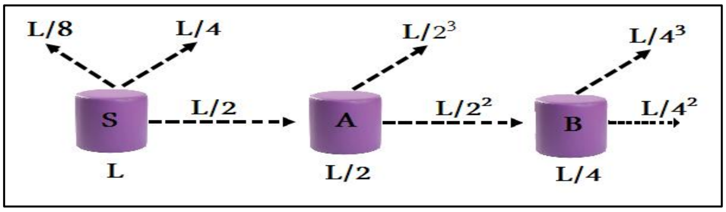

Figure 3.5: The binary mode of SaW protocol.

3.5 Related Work

Numerous research studies have been conducted in the field of routing protocols of DTN. Various simulation environment were utilized to simulate the behavior of different protocols. This research investigates some recent researches about valuation the performance of DTN protocols and its energy consumption. Consuming energy is an important element in the deployment and execution of modernistic communication and computing platforms. Recently mobile phones are rapidly becoming the major communication as well as computing platforms. Since they have the capacity for communication such as Wi-Fi and Bluetooth they are able to convey packets especially in DTN [30]. DTN hardware resources are probably highly restricted and it is substantial to consider the remaining energy of a host when deciding whether data transfer between two hosts come into contact with each other.

From the results presented in [31] it is obviously shown that, the remaining energy of node upsurges when message generation interval increases and message size decreases, speed of nodes and the number of nodes increases. Furthermore, SaW protocol clearly outperforms other protocols with high performance. In [32], they have used different mobility models to investigate the behavior of Epidemic, PRoPHET and SaW protocols. The authors concluded that SaW has best results in terms of average remaining energy and delivery ratio with all mobility models except random walk model where the PROPHET protocol outperforms others just in terms of delivery ratio.

The authors in [33] have evaluated the four well-known protocols and the Bubble Rap protocol with six metrics such as overhead ratio, delivery ratio, hop counts, average latency, average energy consumption and average residual energy. Their results show that, although, Bubble Rap performs better than other protocols in terms of delivery ratio, it performs worse than the others regarding delivery delay and the energy consumption. Max Prop performs obviously worse than the others, it does not consume energy as the rest of the other protocols.

The outcomes given in [34] dissect Epidemic, PRoPHET, MaxProp and SaW. They depict PRoPHET and SaW to be more effective in delivery cost, while MaxProp outperforms all of them in terms of average delay and delivery ratio. Whereas in [35] the outcomes indicate that the Epidemic protocol has best results in terms of average latency and delivery ratio.

In [36], the author concluded that according to the scenario that used in experiments the SaW protocol presents the best performance for overhead ratio and delivery ratio. Another research [37] expressed that PRoPHET and Epidemic routing protocols perform better in delivery ratio, however, their overhead ratios are very high. Whereas, SaW and MaxProp have less delivery ratio, they perform better in overhead ratio.

We use the Opportunistic Network Environment [38] simulator 1.5.1 to simulate the four well-known DTN routing protocols which are Epidemic, PRoPHET, MaxProp, and SaW with five performance metrics that are number of dead nodes, average remaining energy of node, delivery ratio, network overhead ratio and average latency. The protocols that demonstrate best results in the network delivery ratio and energy consumption should be the suitable one in the network. The performance evaluation of different DTN routing protocols will be discussed in the next chapter.

Chapter 4

SIMULATION ENVIRONMENT AND RESULTS

4.1 Performance metrics

In this section, we provide performance metrics to evaluate the performance of DTN protocols as presented in [32, 39]. We focus on just five of these metrics which are nodes average remaining energy, number of dead nodes, delivery ratio, average latency and overhead ratio. Simulation plays significant role in analyzing the behavior of routing protocols of DTN. The majority of researchers use simulators which allow easily for a large number of reproducible environmental conditions. One of these simulators is Opportunistic Network Environment (ONE) simulator which has been used in our implementation. ONE simulator functionality and the PRoPHET, Epidemic, SaW and MaxProp routing protocols are obtainable in “java.docs” format in [39].

These are some performance metrics that used to evaluate the routing protocols:

Node’s Average Remaining Energy

The average energy of nodes that are left at the end of the simulation.

Number of Dead Nodes

The number of dead nodes after its energy reaches almost zero (we start calculate the nodes that cannot execute scanning or transmitting messages process in the network due to its low energy).

Delivery Ratio

The ratio of the total delivered messages (packets) and the total messages sent by the sender [35].

$$Delivery\ Ratio = \frac{\sum_i Packets\ Delivered}{\sum_i Packets\ Sent} \tag{4.1}$$

Average Latency

The average time that all messages require to reach the destination.

$$Average\ Latency = \frac{\sum_{n=1}^{N} Tinit_n - Tdel_n}{N} \tag{4.2}$$

where *Tinit* is the creation time of message *n*, *Tdel* is time taken by node *n* to deliver its destination and where *N* is total number of delivered messages.

Overhead Ratio

The ratio of the messages relayed and the messages delivered to the destination. Thus it is defined by the ONE simulator as:

$$Overhead\ Ratio = (P_r\ (t) - P_d\ (t))\ /\ P_d\ (t) \tag{4.3}$$

where P_r is the total messages relayed by time t and P_d is the total messages delivered by time t.

Total Dropped Packets

The summation of dropped packets for each created packet.

$$Total\ Dropped\ Packets = \sum_{n=1}^{N} Dr_n \tag{4.4}$$

where *Dr* is dropped messages for each created message and *N* is total number of created messages.

Average Hops Count

The proportion of the total hops of each message copies to the total amount of created messages.

$$\textit{Average Hops Count} = \frac{\sum_{n=1}^{N} Ph_n}{N} \qquad (4.5)$$

where *Ph* is the number of hops count for every delivered message and *N* is the total number of created messages.

4.2 Simulator Setup and Settings

4.2.1 ONE simulator

ONE simulator is an agent-based discrete event simulator which was proposed at the Helsinki University of Technology [38]. ONE is a graphical network simulator specially designed for simulating DTNs. It comes with standard routing algorithms including PRoPHET, Epidemic, MaxProp and SaW. The simulator based on JAVA software that provides DTN routing protocols simulation capabilities in a single workshop. Figure.4.1 presents the interaction of the simulator and its elements.

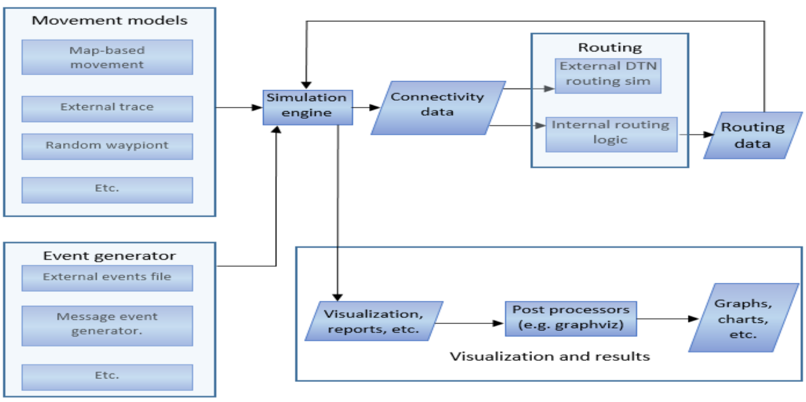

Figure 4.1: The structure of ONE simulator [37].

4.2.2 Simulator Setup

Since we have mentioned above that the ONE simulator is a JAVA based software it therefore requires a JAVA software such as Eclipse to run properly. The simulator files are available in [41]. The ONE is a simulation environment that is capable of generating node movement using different movement models and routing messages between nodes with various DTN routing protocols. We can run it in two different modes, Batch and GUI. The GUI is preferable for exhibition, investigating and testing purposes since we cannot run it with several sets of settings and Batch can be used to run a big number of simulations using several settings. Both of them able to include several forms of reports that provide create simulation statistics. And these statistics are summarized and analyzed as plots and charts.

4.2.3 Simulation Scenarios and Settings

As any simulator there are many settings which affect the routing performance. Some of these settings are fixed like simulation time, world size, radio range and transmission speed. In contrast, there are some varying settings such as number of nodes, message size and message generation interval. We use three groups of nodes as Vehicles (V), Cyclists (C) and Pedestrians (P) with different speed ranges which are 0.5-1.5 m/s for pedestrians, 1.5-2.5 m/s for cyclists and 2.5-12.5 m/s for vehicles. Meanwhile all the groups choose the shortest path map based movement model that defined the forms of simulator maps. We use the map of Helsinki city which is presented in the simulator maps. Figure 4.2 shows the city map while distributing 80 hosts. Finally most of the settings are clarified in Table 4.2. Also there are some specific energy settings which are presented in Table 4.3.

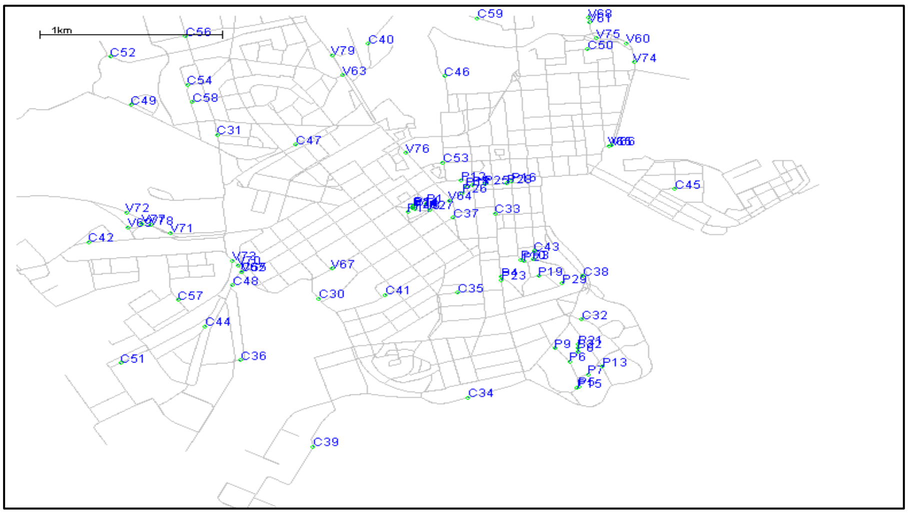

Figure 4.2: Helsinki city map in the simulator with 80 hosts

Table 4.1: Simulation settings

Parameter	Values		
World size (m × m)	4500 × 3400		
Traffic type	Data		
Interface	Bluetooth		
No of nodes	40, 80, 120, 160, 200		
Node type	P	C	V
Node movement speed (m/s)	0.5-1.5	1.5-2.5	2.5-12.5
Buffer size (MB)	3, 5, 7, 9, 11		
Radio range (m)	10		
TTL (h)	1, 3, 5, 7, 9		
Transmission speed (MB/s)	2		
Node movement model	Shortest Path Map-Based Movement (SPMBM)		
	Pedestrian Path	Pedestrian Path	Main Roads
Message size (KB)	100-500 , 500-1024 , 1024-1500 , 1500-2048 , 2048-2500		
Simulation time (h)	12		
Message interval (s)	0-10, 10-20, 20-30, 30-40, 40-50, 100-150, 150-250, 250-350, 550-650		

Table 4.2: Energy hosts settings

Parameter	Values (Units)
Base Energy	0.01
Scan Response Energy	0.1
Scan Energy	0.1
Transmit Energy	0.2
Initial Energy	5000

Base Energy is the consumed energy when a node is idle. Scan Response Energy is the consumed energy while scanning response, Scan Energy is the consumed energy during scanning, Transmit Energy is the consumed energy while transmitting and Initial Energy is the energy assigned to the hosts at the beginning.

4.3 Simulation Results

As we mentioned above we focus on five metrics by varying some settings which include buffer size, nodes speed, message generation interval, message size, TTL and number of nodes to investigate the impact of these settings on the performance.

4.3.1 Impact of Number of Nodes

We applied a set of number of node values as 40, 80, 120, 160 and 200 to evaluate the impact of varying number of nodes on performance of energy consumption, delivery ratio, average latency and overhead ratio whereas other parameters are fixed as TTL is 5 hours (h), message size is 0.5 - 1 MB, message generation interval is 100 - 150 seconds (s), buffer size with two values as 3 MB and 9 MB and node’s speed range as 0.5 – 1.5 m/s for P, 1.5 – 2.5 m/s for C and 2.5 – 12.5 m/s for V.

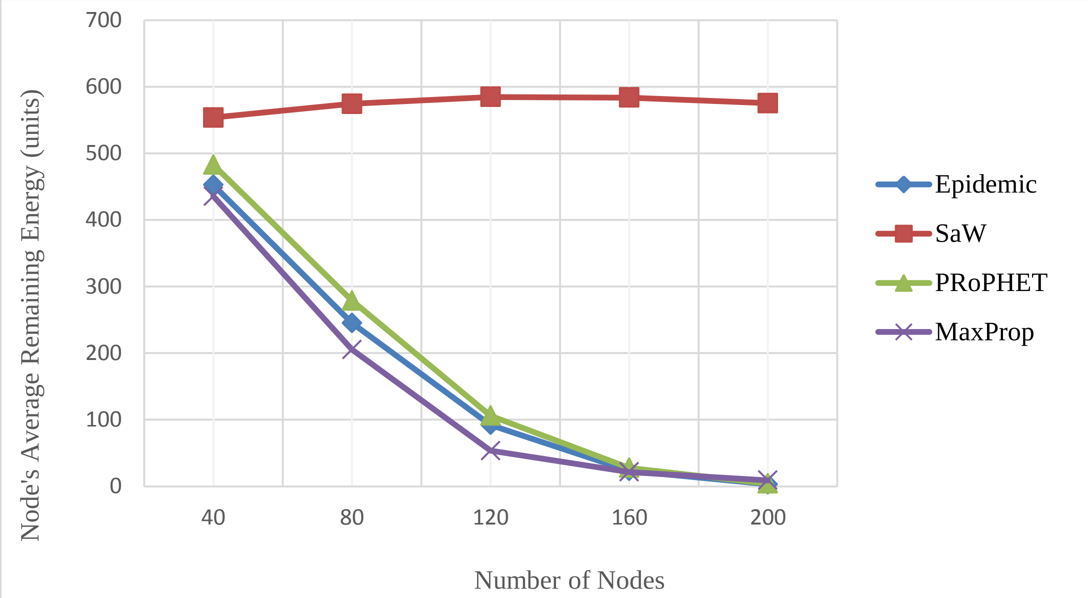

Figure 4.3: Impact of number of nodes on node’s average remaining energy using buffer size as 3 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

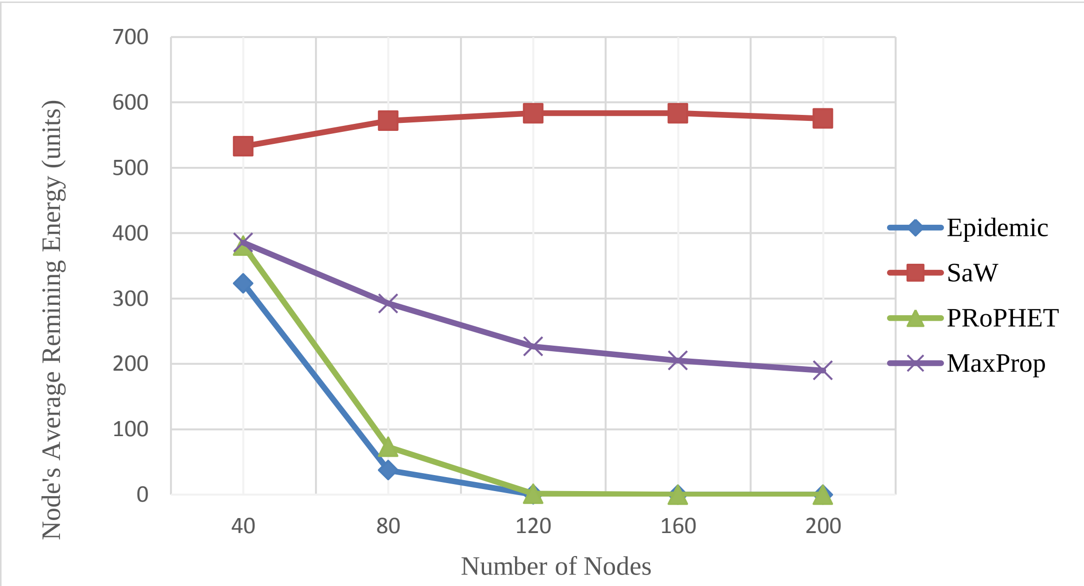

Figure 4.4: Impact of number of nodes on node’s average remaining energy using buffer size as 9 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

It is shown in Figures 4.3 and 4.4 that while increasing the number of hosts, the hosts’ average remaining energy decreases. By rising the number of hosts, the number of packets transmitted increases that cause more transmissions and scans of nodes which consume more energy. It can be observed from the results that SaW outperforms other protocols. This is because of that, in SaW most of the hosts will wait and transmit the packet when the source host did not find the final recipient as addressed in [28]. So this causes less scans and transmissions between nodes which results in lower energy consumption.

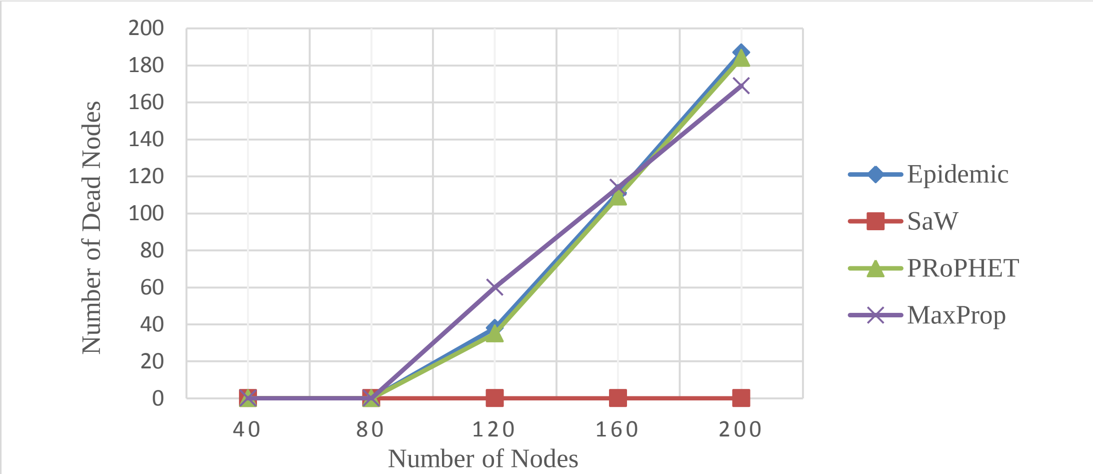

Figure 4.5: Impact of number of nodes on number of dead nodes using buffer size as 3 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

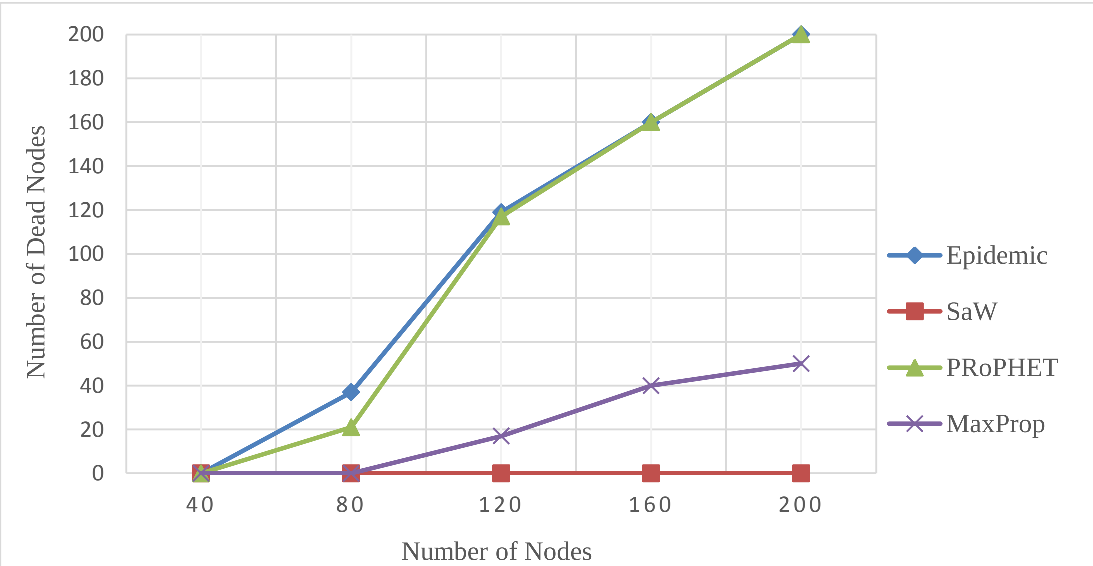

Figure 4.6: Impact of number of nodes on number of dead nodes using buffer size as 9 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

Also as indicated in Figures 4.5 and 4.6, the number of dead hosts goes up along with increasing in number of nodes except SaW with zero dead nodes. Furthermore, in Epidemic and PRoPHET, the node's average remaining energy reduces while using 9 MB of buffer size and the number of dead nodes increases (almost all the nodes are died starting from 120 to 200). In contrast, the energy consumption of MaxProp slightly increases as well as its dead nodes. However, it performs better than while using 3 MB of buffer size because of the protocol behavior which we will explain it when varying buffer size later. SaW, on the other hand, outperforms all protocols in both cases.

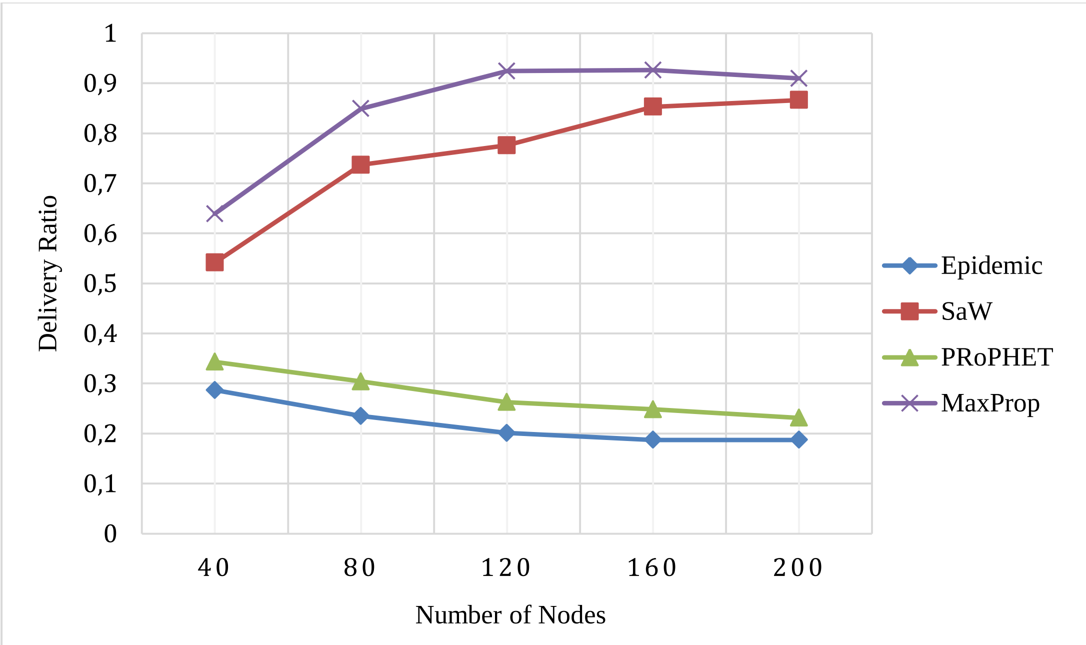

Figure 4.7: Impact of number of nodes on delivery ratio using buffer size as 3 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

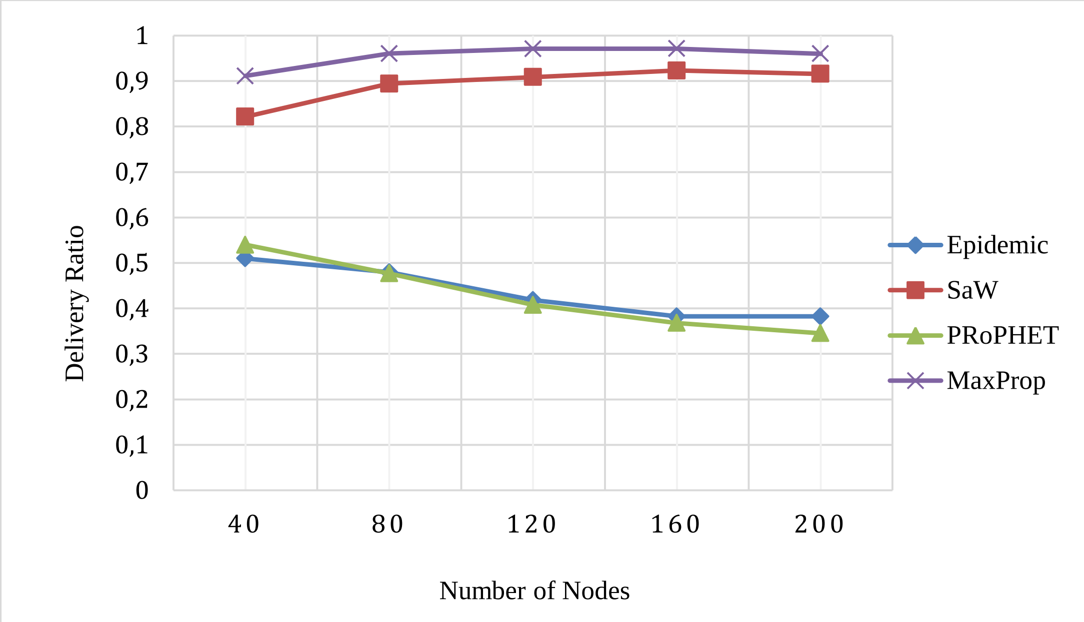

Figure 4.8: Impact of number of nodes on delivery ratio using buffer size as 9 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

In Figures 4.7 and 4.8, there are two different results since MaxProp and SaW go up by increasing the number of nodes due to their behavior as both protocols distribute limited messages. And so an increase in the number of nodes creates shorter paths for the messages to reach their destination. While PRoPHET and Epidemic performance decrease due to their flooding strategy, since they distribute unrestrained number of messages which increase the dropped messages. For the same reason the traffic load go up which causes overhead ratio as shown in Figure 4.9.

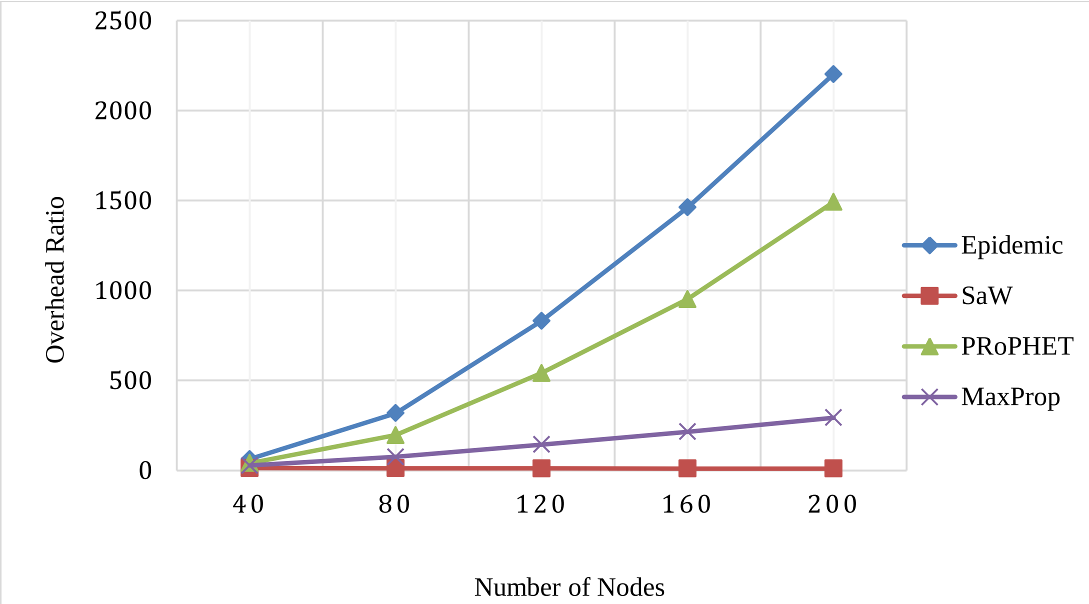

Figure 4.9: Impact of number of nodes on overhead ratio using buffer size as 3 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

There is no effect of number of nodes on average latency for SaW and MaxProp, only a very slight effect is observed. This is due to the same reason, the average latency of all protocols reduces unless SaW results somewhat increased because of the waiting mechanism as depicted in Figure 4.11.

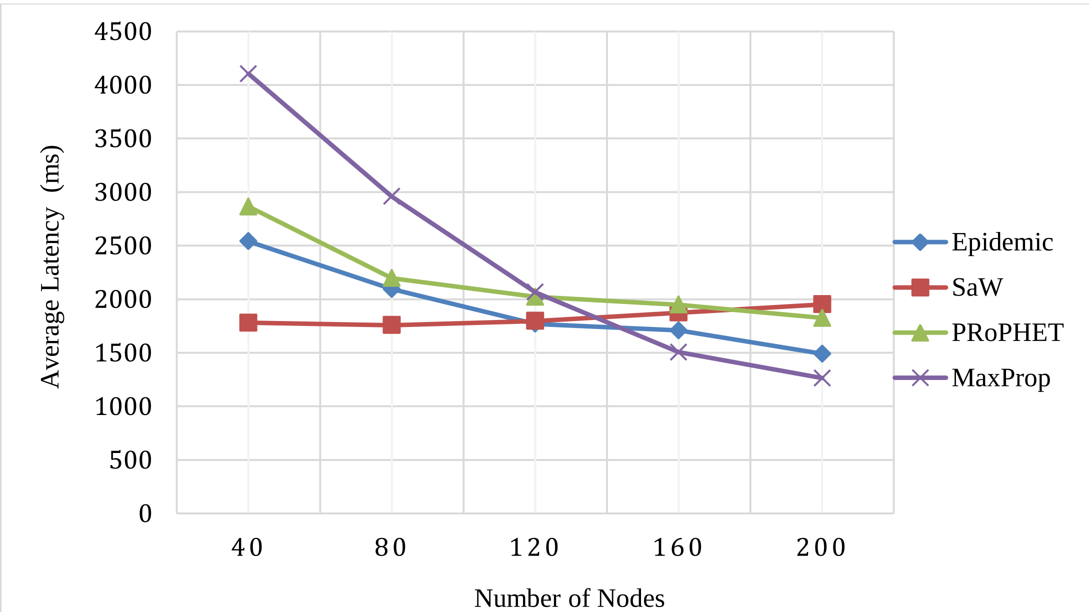

Figure 4.10: Impact of number of nodes on average latency using buffer size as 3 MB (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

Table 4.3 shows overall results of node's average remaining energy and number of dead nodes with varying number of nodes.

Table 4.3: Summary of varying number of nodes

Protocols	**Node's Average Remaining Energy**		**Number of Dead Nodes**	
	Number of Nodes			
	40	**200**	**40**	**200**
Epidemic	Very high	Very low	Very low	Very high
SaW	Very high	Very high	Very low	Very low
PRoPHET	Very high	Very low	Very low	Very high
MaxProp	Very high	Very low	Very low	Very high

4.3.2 Impact of Message Size

We use two sets of message size ranges as 0.05 – 0.25 MB, 0.25 – 0.5 MB, 0.5 – 0.75 MB, 0.75 – 1 MB and 1 – 1.5 MB with a buffer size of 3 MB and 0.1 – 0.5 MB, 0.5 – 1 MB, 1 – 1.5 MB, 1.5 – 2 MB and 2 – 2.5 MB with a buffer size of 9 MB to evaluate the impact of varying message size on performance of energy consumption, delivery ratio, average latency and overhead ratio whereas other parameters are fixed as TTL is 5 h, number of nodes is 80, message generation interval is 100-150 seconds and node's speed range as 0.5 – 1.5 m/s for P, 1.5 – 2.5 m/s for C and 2.5 – 12.5 m/s for V. We have used two values of buffer size since we faced a problem with MaxProp protocol while using 3 MB buffer because when message size is larger than 1.5 MB there will be only one message in the buffer so the comparison process is not possible. This is done so that the protocol has to make some comparisons and the decision must be made as to which messages to keep and which ones to drop.

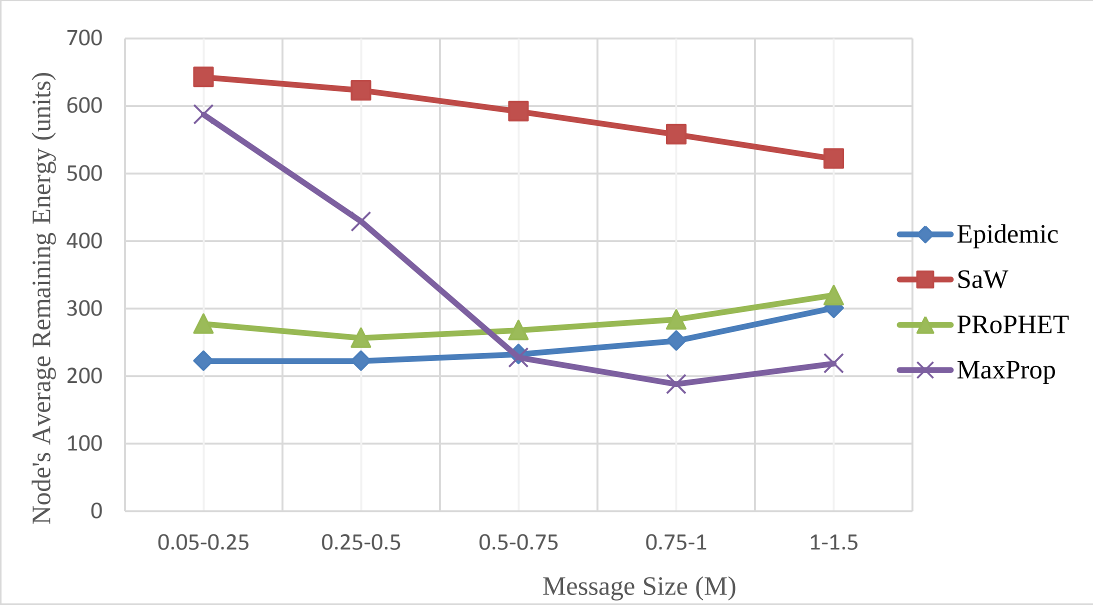

Figure 4.11: Impact of message size in bytes on node's average remaining energy using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

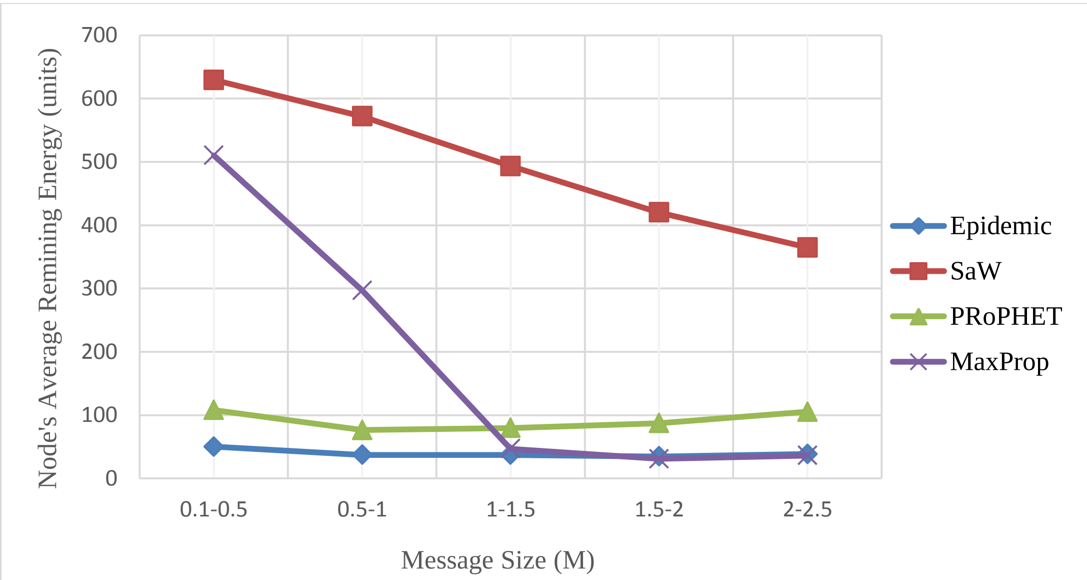

Figure 4.12: Impact of message size in bytes on node's average remaining energy using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

The impact of varying message size on energy consumption divides the protocols into two groups. The node's average remaining energy of the first group (Epidemic and PRoPHET) surges up by increasing message size as this leads to a decrease in the number of messages in the buffer which in turn decreases the number of messages in the network. Vice versa, node's average remaining energy of the other group (MaxProp and SaW) decreases since they send a restricted number of message duplicates, and so the messages require a longer duration of time to reach its destination and that cause more energy consumption through the scanning as shown in Figures 4.11 and 4.12. With all this, there are no dead nodes with all protocols. Although, all protocols show the same behavior while increasing buffer size to 9 MB, a decrease in average remaining energy is observed and this results in some dead nodes for all protocols except SaW which has no dead nodes as presented in Figure 4.13.

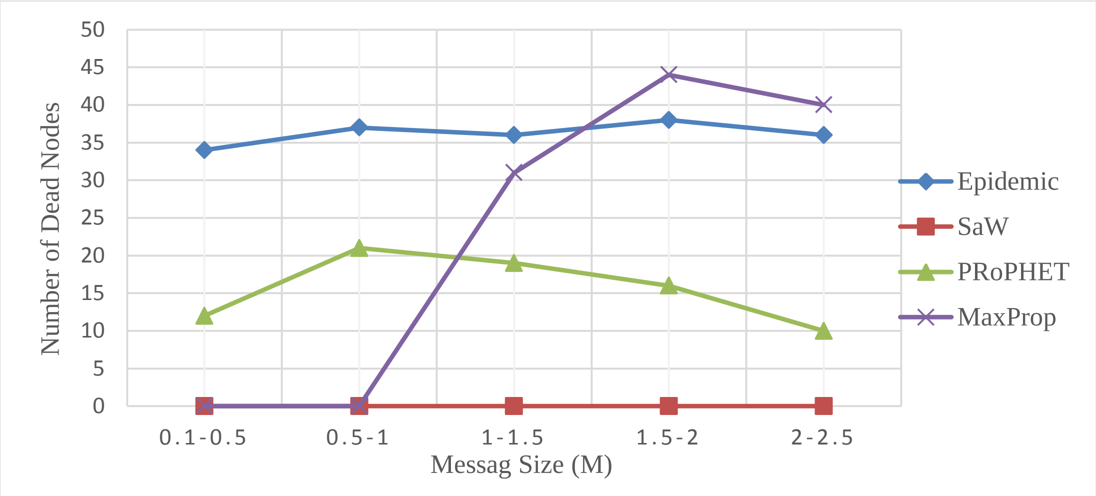

Figure 4.13: Impact of message size in bytes on number of dead nodes using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

It is for the same reason that the delivery ratio while using both 3 MB and 9 MB buffer size decreases with all protocols. Furthermore, the average latency surging up as shown in Figures 4.14, 4.15 and 4.16 respectively. On the other hand, as we see in Figure 4.17 overhead ratio reduces to the minimum values with the first group whereas there is no effect with the other which applies the same behavior.

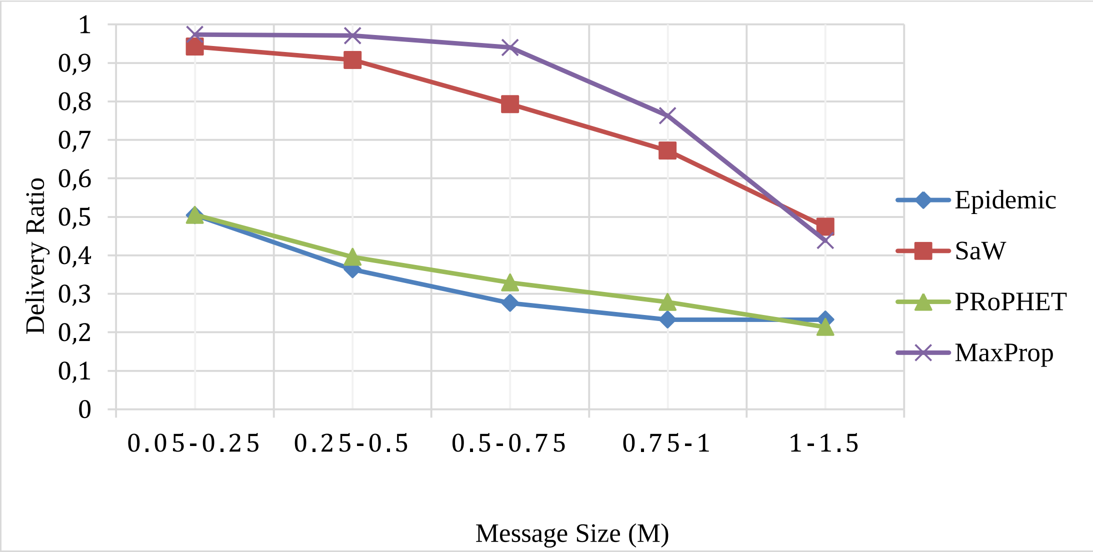

Figure 4.14: Impact of message size in bytes on delivery ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

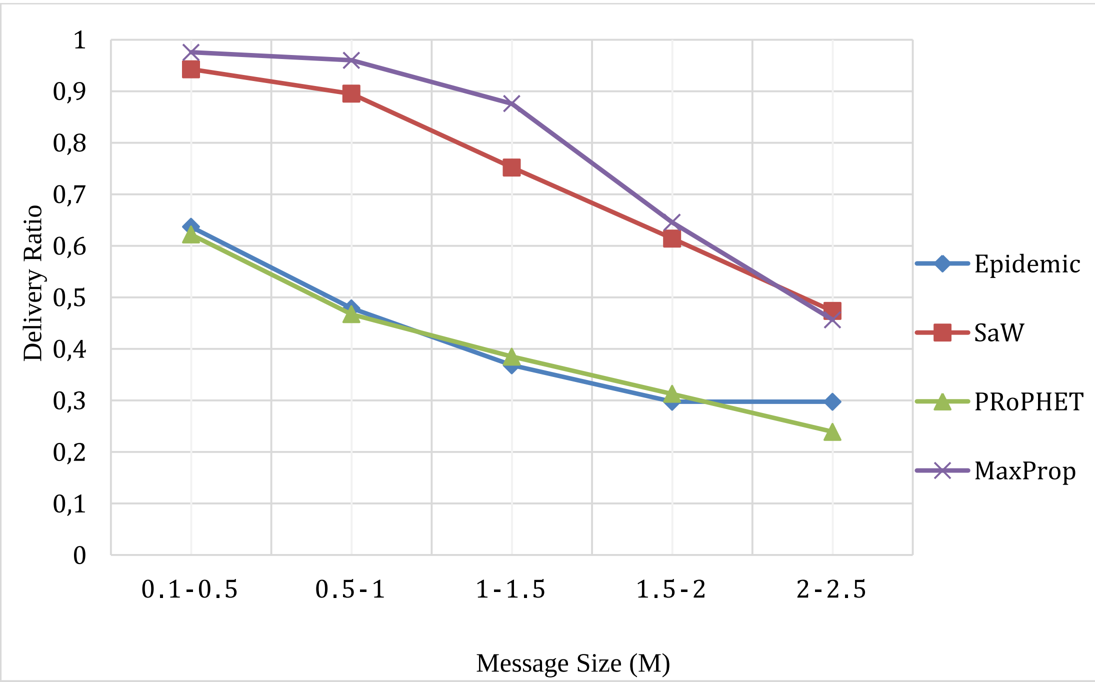

Figure 4.15: Impact of message size in bytes on delivery ratio using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

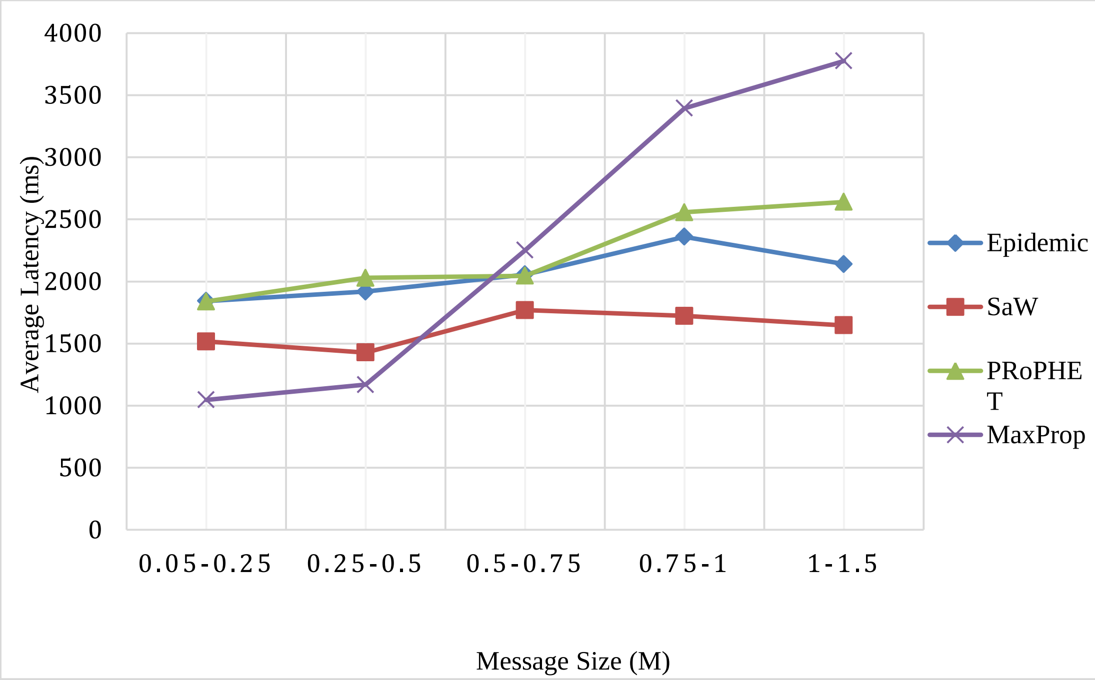

Figure 4.16: Impact of message size in bytes on average latency using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

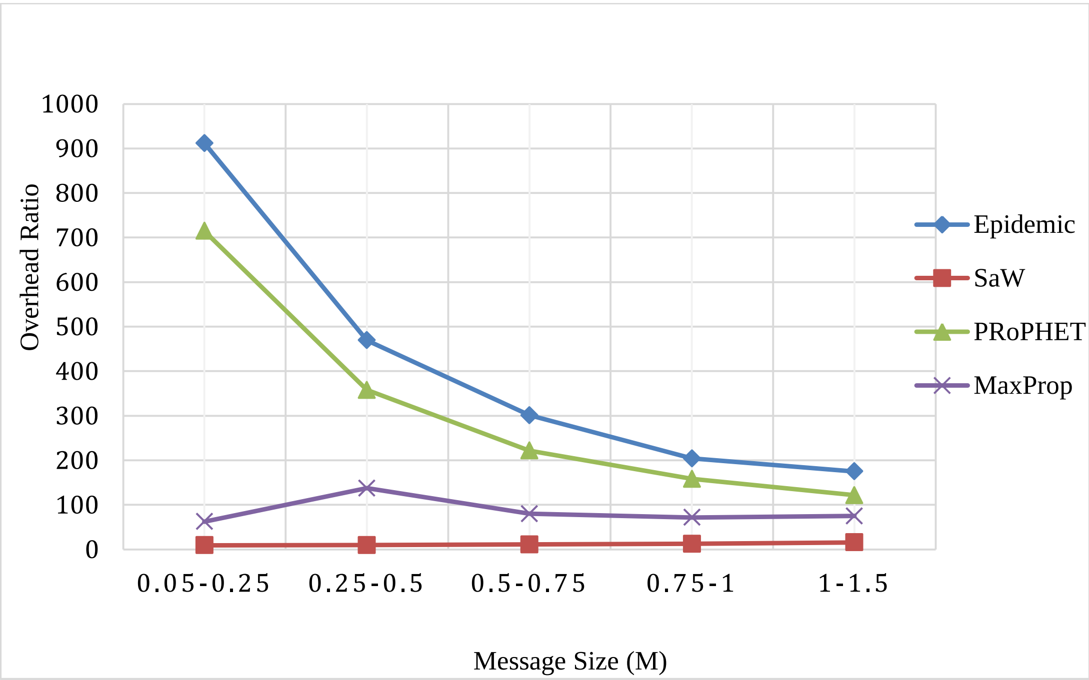

Figure 4.17: Impact of message size in bytes on overhead ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message generation interval = 100 – 150 s)

Table 4.4 shows overall results of node's average remaining energy and number of dead nodes with varying message size.

Table 4.4: Summary of varying message size

Protocols	**Node's Average Remaining Energy**		**Number of Dead Nodes**	
	Message size (MB)			
	0.1 – 0.5	**2 – 2.5**	**0.1 – 0.5**	**2 – 2.5**
Epidemic	Very low	Very low	Medium	Medium
SaW	Very high	Very high	Very low	Very low
PRoPHET	Low	Low	Low	Low
MaxProp	Very high	Very low	Very low	Medium

4.3.3 Impact of Message Generation Interval

We applied a set of ranges of message generation interval as 0-10, 10-20, 20-30, 30-40, 40-50, 100-150, 150-250, 250-350 and 550-650 seconds to evaluate the impact of message generation interval whereas other parameters are fixed as TTL is 5 hours, message size is 0.5 - 1 MB, number of nodes as 80, buffer size with two values as 3 MB and 9 MB and node's speed range as 0.5 – 1.5 m/s for P, 1.5 – 2.5 m/s for C and 2.5 – 12.5 m/s for V .

Figures 4.18 and 4.19 clearly show that by raising the number of message generation interval the average remaining energy and delivery ratio go up as well. And this applies to all protocols because increasing message generation interval means the number of message creations are reduced so the energy consumption decreases. Also the number of dead nodes goes down to zero as indicated in Figures 4.20 and 4.21.

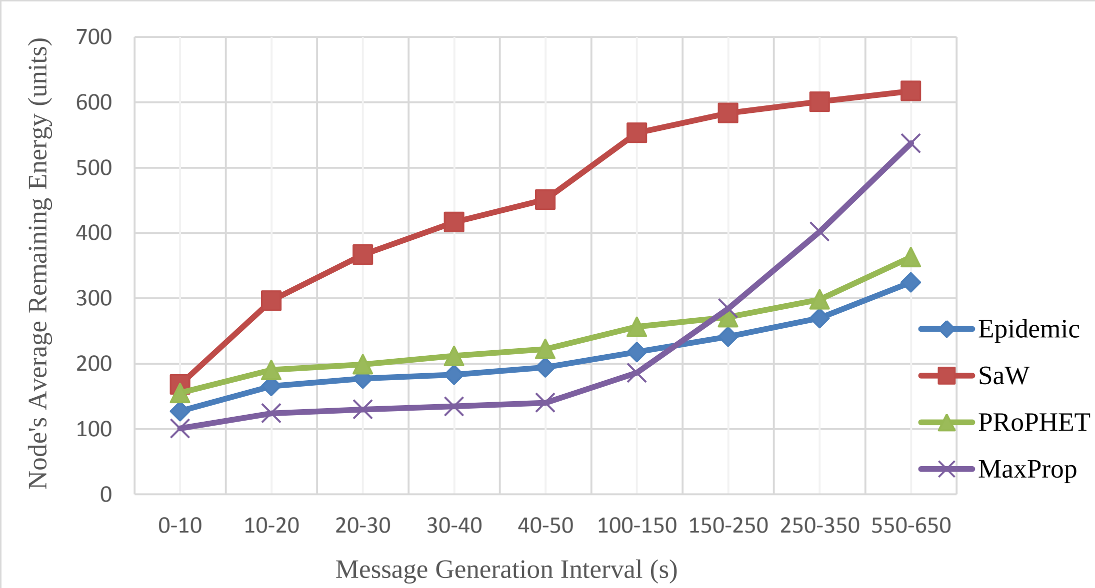

Figure 4.18: Impact of message generation interval on node’s average remaining energy using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

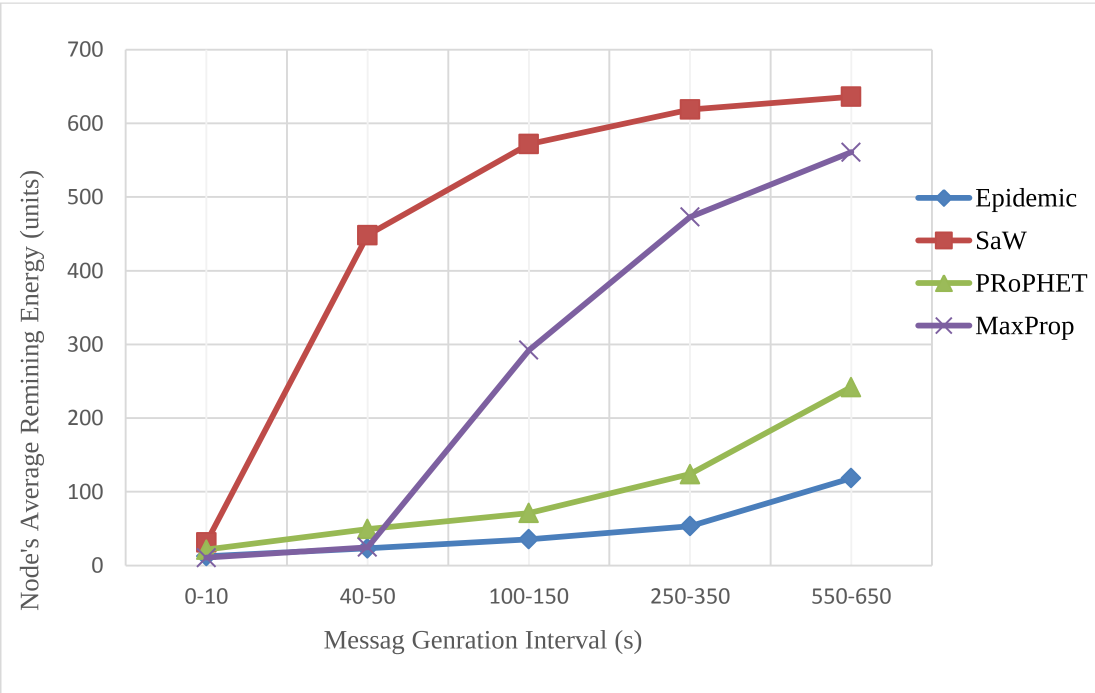

Figure 04.19: Impact of message generation interval on node’s average remaining energy using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

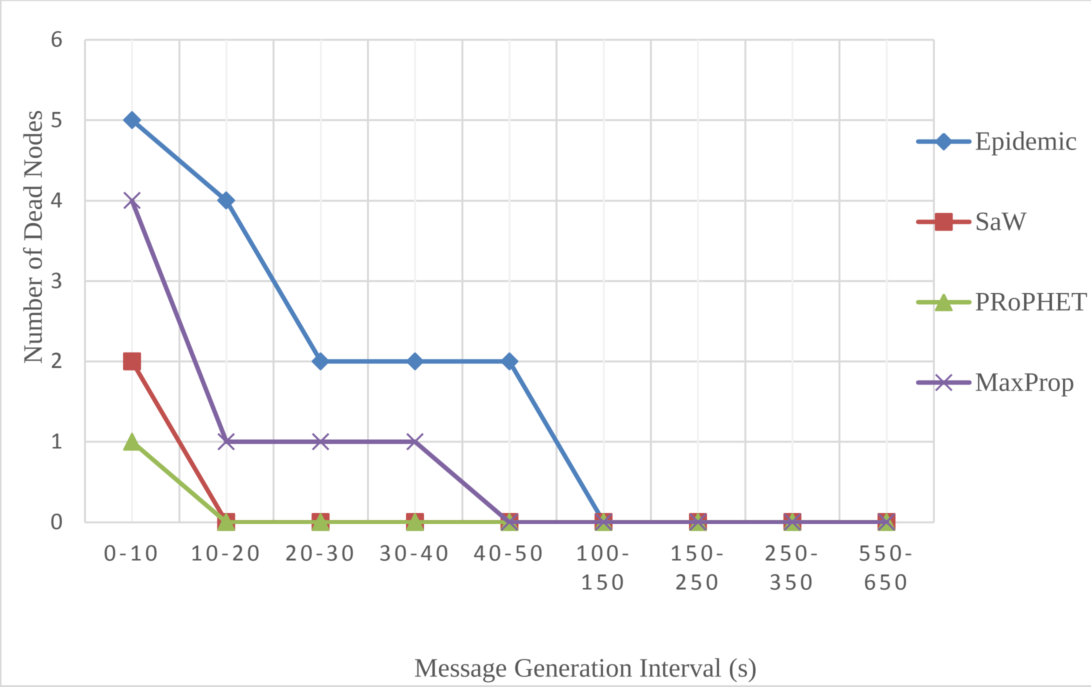

Figure 4.20: Impact of message generation interval on number of dead nodes using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

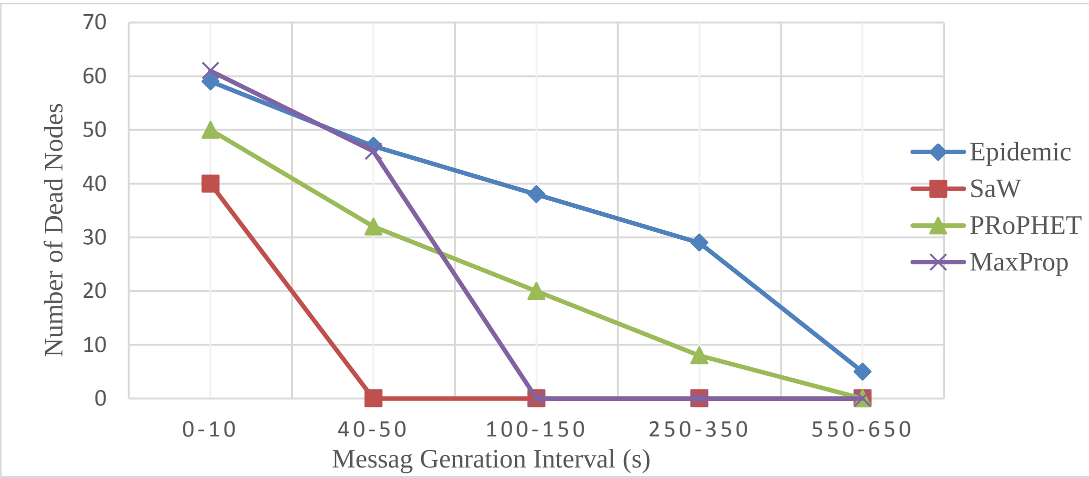

Figure 4.21: Impact of message generation interval on number of dead nodes using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

Extending the traffic load in the network with a steady number of host results in the overburdening of buffers and an increase in the dropping proportion. According to this, the delivery ratio goes up considerably. Hence, increment in the message generated interval leads to decrease in the number of generated messages. The delivery proportion for all protocols increases as shown in Figures 4.22 and 4.23.

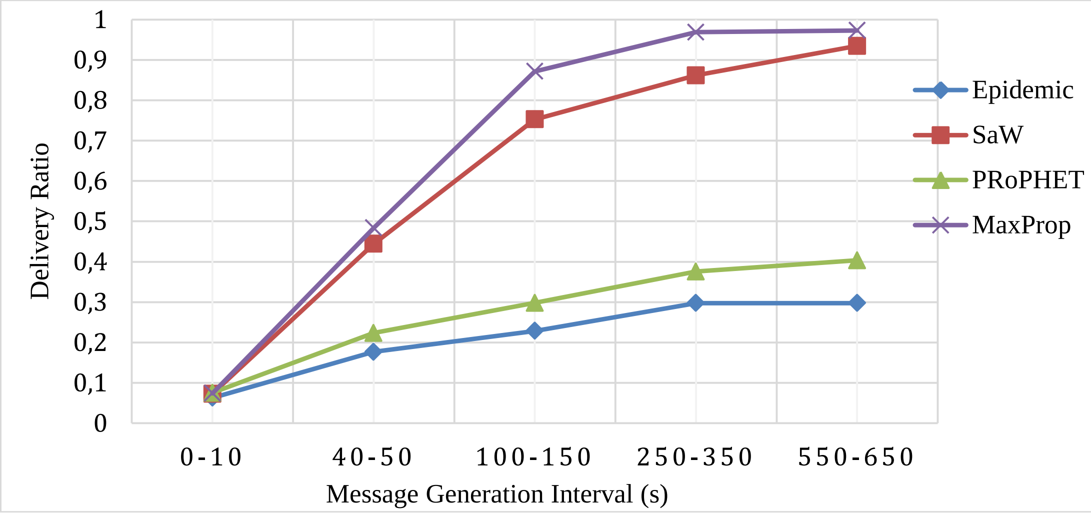

Figure 4.22: Impact of message generation interval on delivery ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

The overhead ratio of the unrestricted protocols rise as well while the restricted protocols do not rise and this is shown in Figure 4.24. Moreover, Figure 4.25 indicate that average latency of all protocols goes up but with the exception of the MaxProp protocol which shows the worst results while using message generator range as 40-50. However, it should be noted that these critical results occur when using the range as 550-650 as the best.

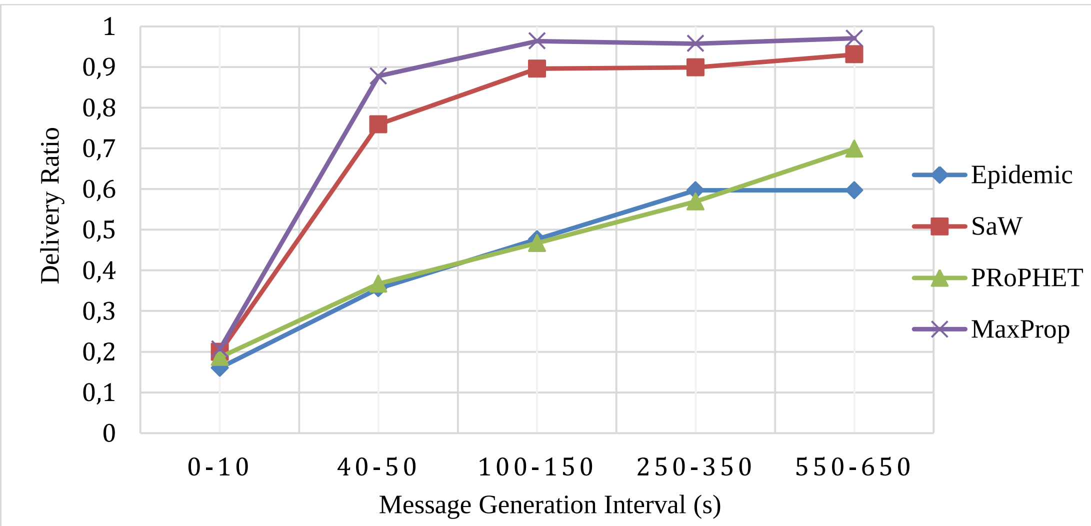

Figure 4.23: Impact of message generation interval on delivery ratio using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

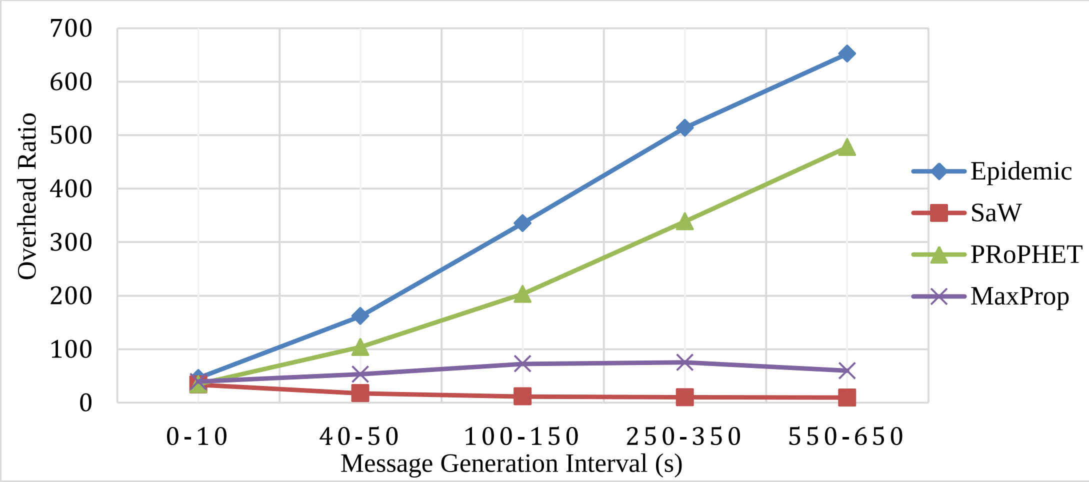

Figure 4.24: Impact of message generation interval on overhead ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

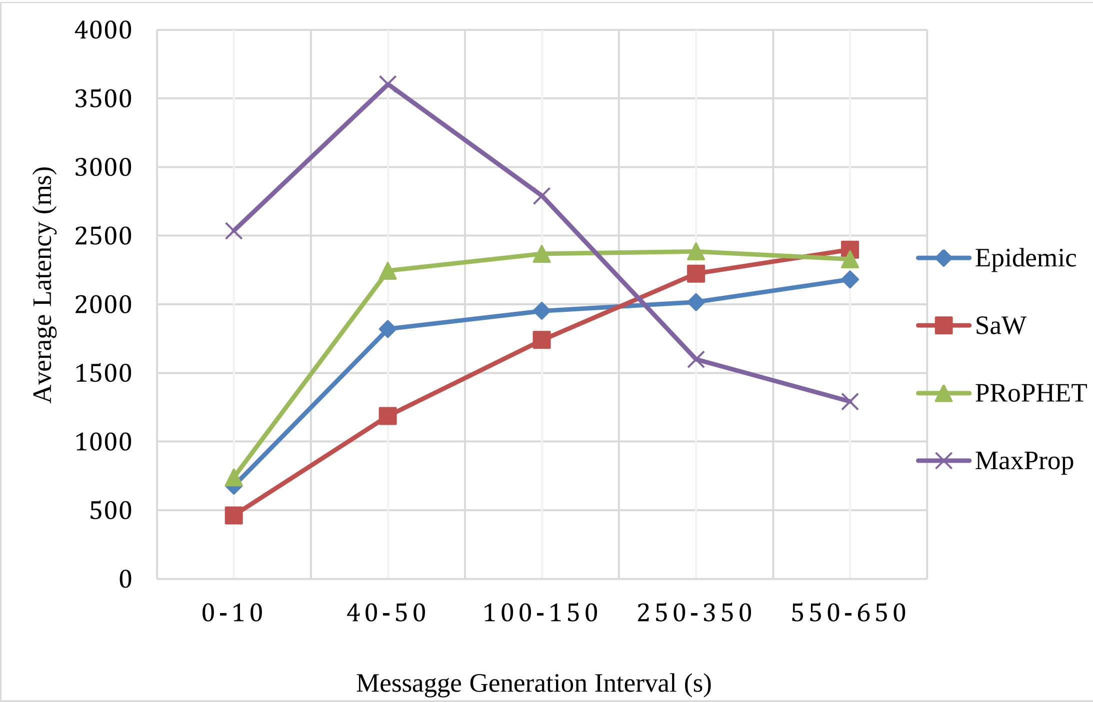

Figure 4.25: Impact of message generation interval on overhead ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB)

Table 4.5 shows overall results of node’s average remaining energy and number of dead nodes with varying message generation interval.

Table 4.5: Summary of varying message generation interval

Protocols	Node’s Average Remaining Energy		Number of Dead Nodes	
	Message Generation Interval (s)			
	0-10	550-650	0-10	550-650
Epidemic	Low	High	Low	Very low
SaW	Low	Very high	Very low	Very low
PRoPHET	Low	High	Very low	Very low
MaxProp	Low	Very high	Low	Very low

4.3.4 Impact of Node’s Speed

We applied a set of ranges of node’s speed as 0-2.5, 2.5-5, 5-7.5, 7.5-10 and 10-12.5 m/s to evaluate the impact of varying node’s speed whereas other parameters are fixed as TTL is 5 h, number of nodes as 80, message size is 0.5 - 1 MB, message generation interval is 100-150 seconds and buffer size with two values as 3 MB and 9 MB.

Basically increments in node speed decreases the contacting period of time which causes more scanning and transferring messages between nodes. And that in turn leads to increase in energy consumption as well as an increased number of dead nodes as shown in Figures 4.26 - 4.29 respectively. It is obvious that the range speed as 2.5 - 5 m/s presents the best results overall other ranges on delivery ratio and average latency while the range speed as 5 - 7.5 m/s has the lowest overhead ratio as shown in Figures 4.30 - 4.33.

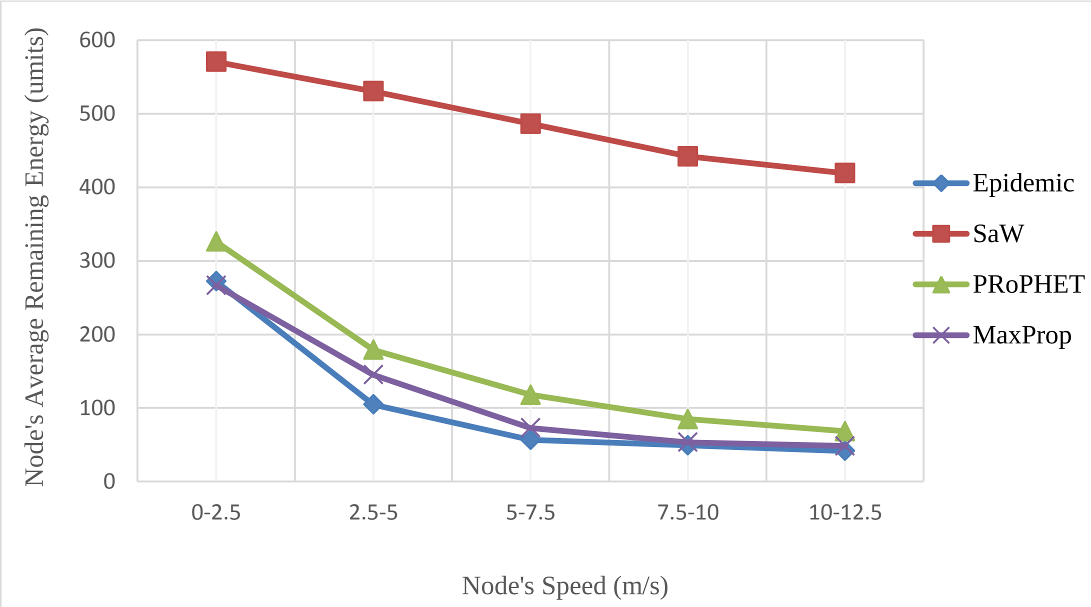

Figure 4.26: Impact of node’s speed on node’s average remaining energy using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

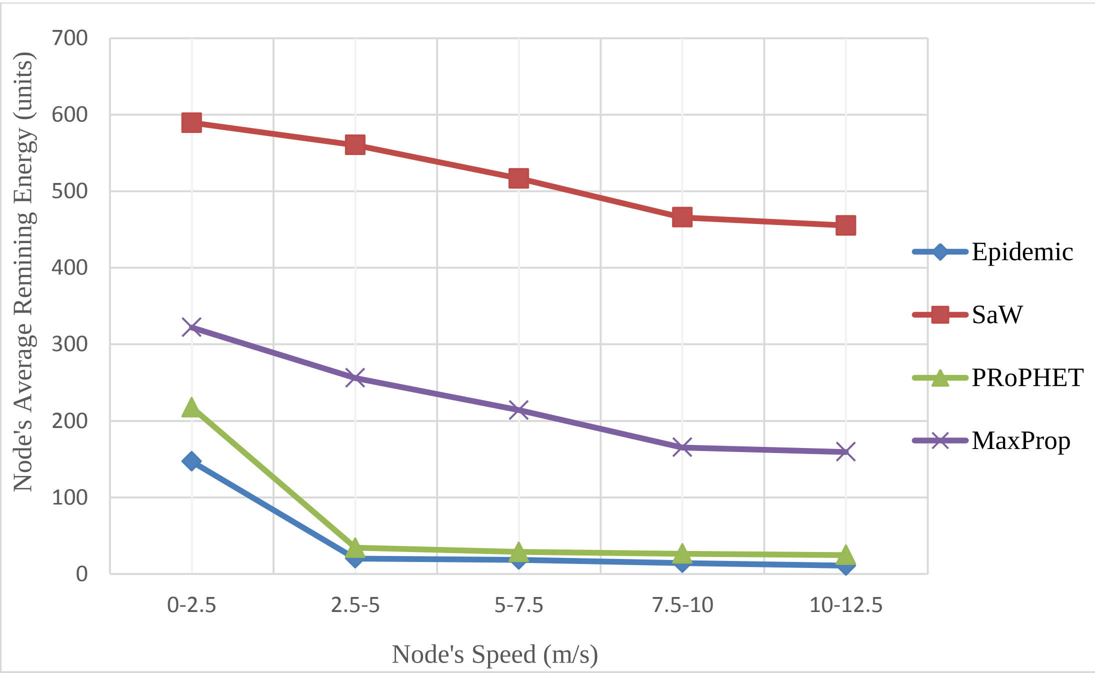

Figure 4.27: Impact of node’s speed on node’s average remaining energy using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

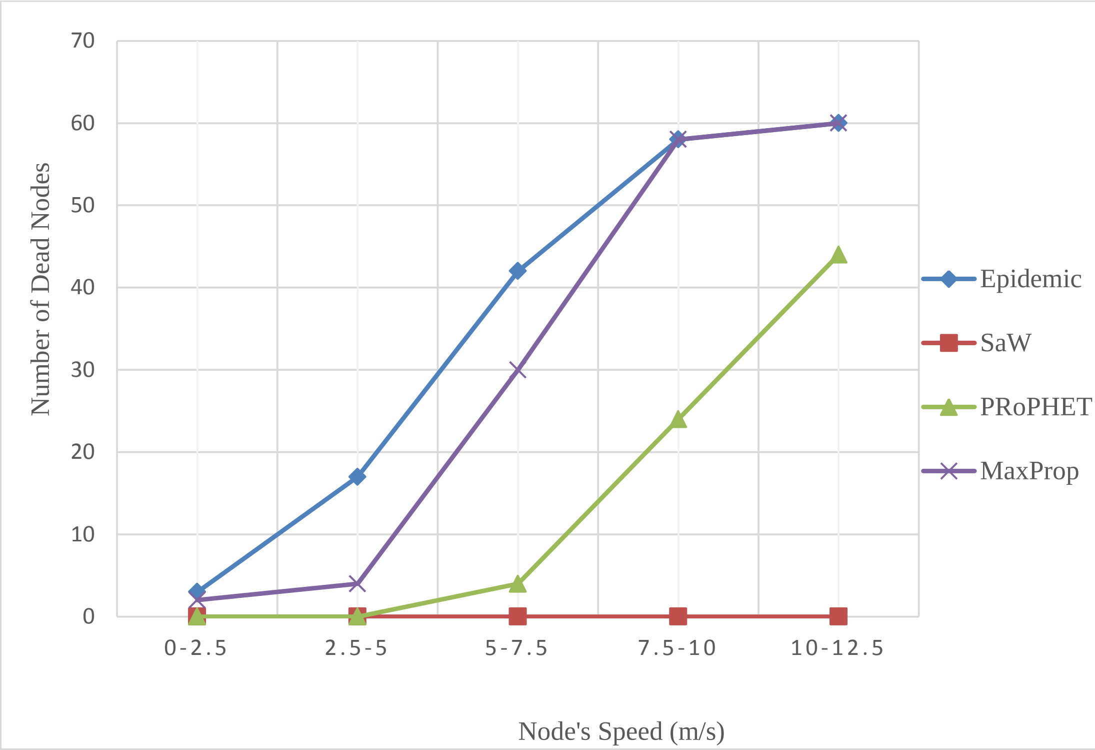

Figure 4.28: Impact of node’s speed on number of dead nodes using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

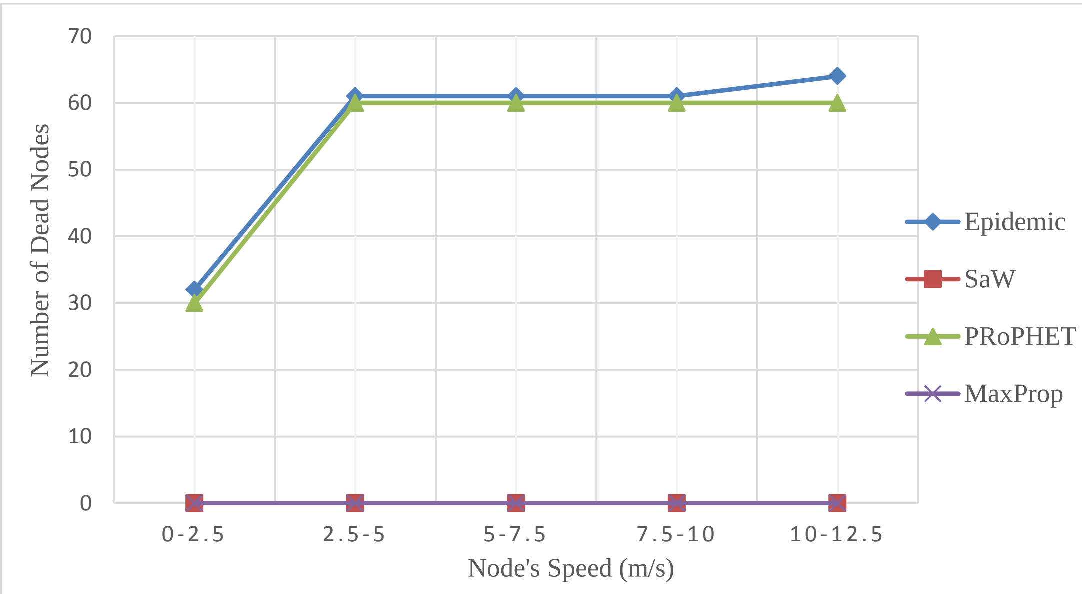

Figure 4.29: Impact of node’s speed on number of dead nodes using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

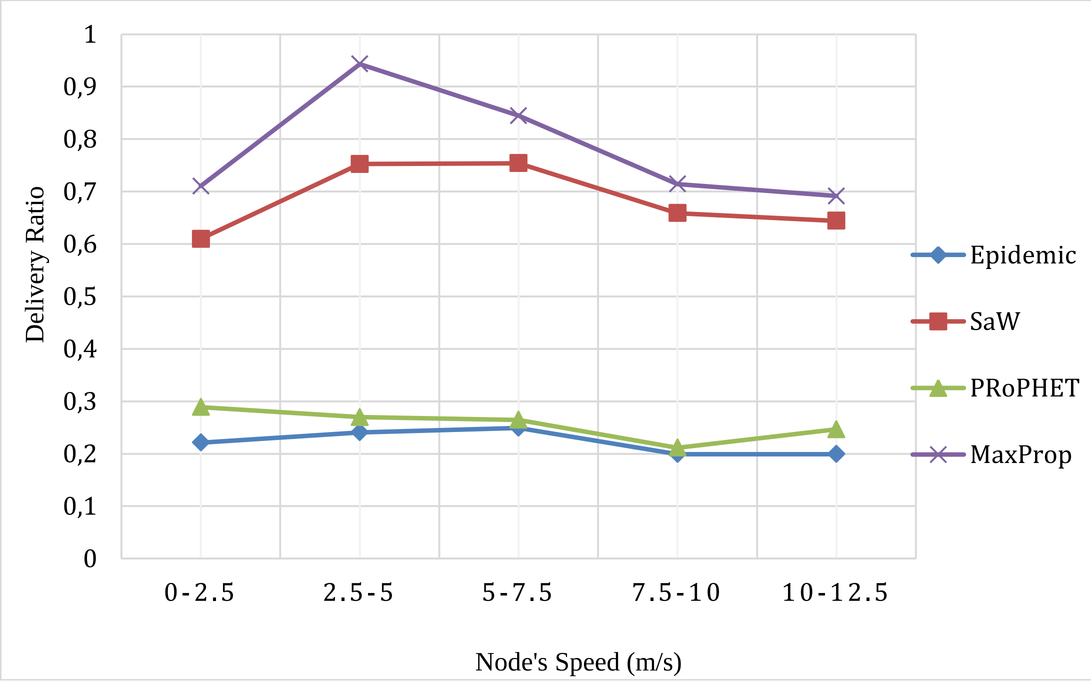

Figure 4.30: Impact of node’s speed on delivery ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

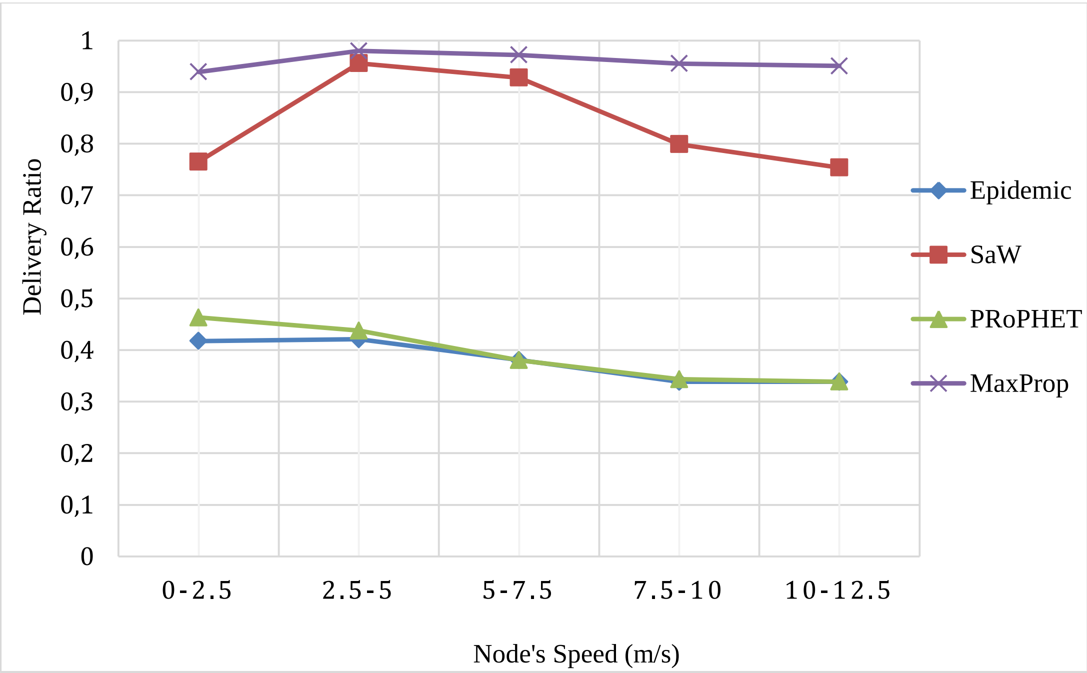

Figure 4.31: Impact of node’s speed on delivery ratio using buffer size as 9 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

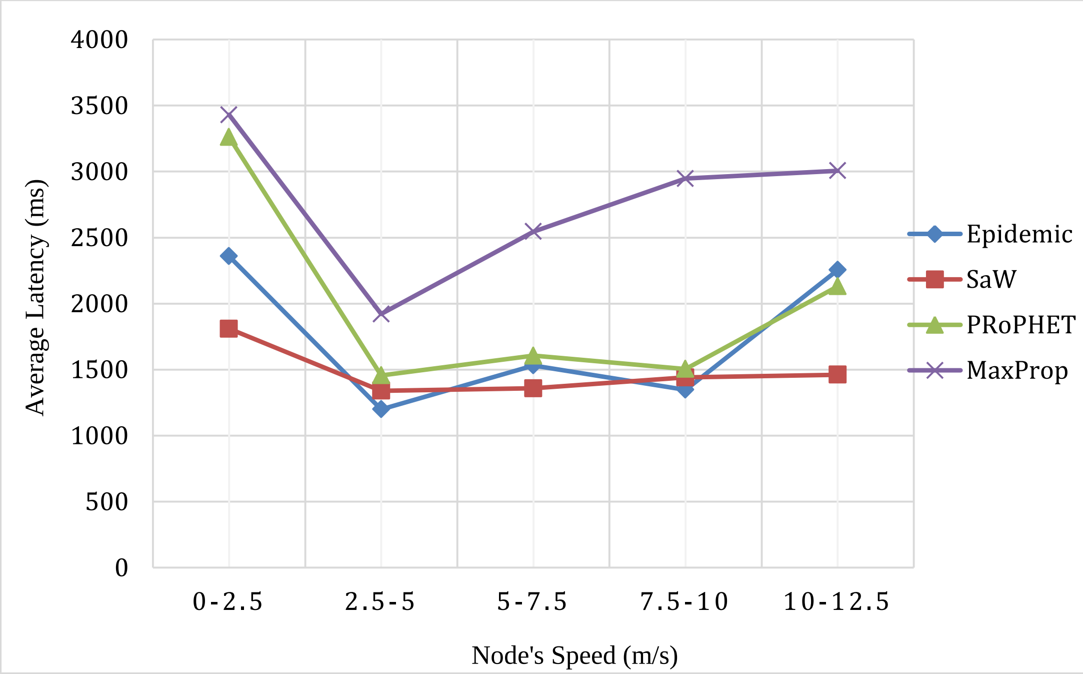

Figure 4.32: Impact of node's speed on average latency using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

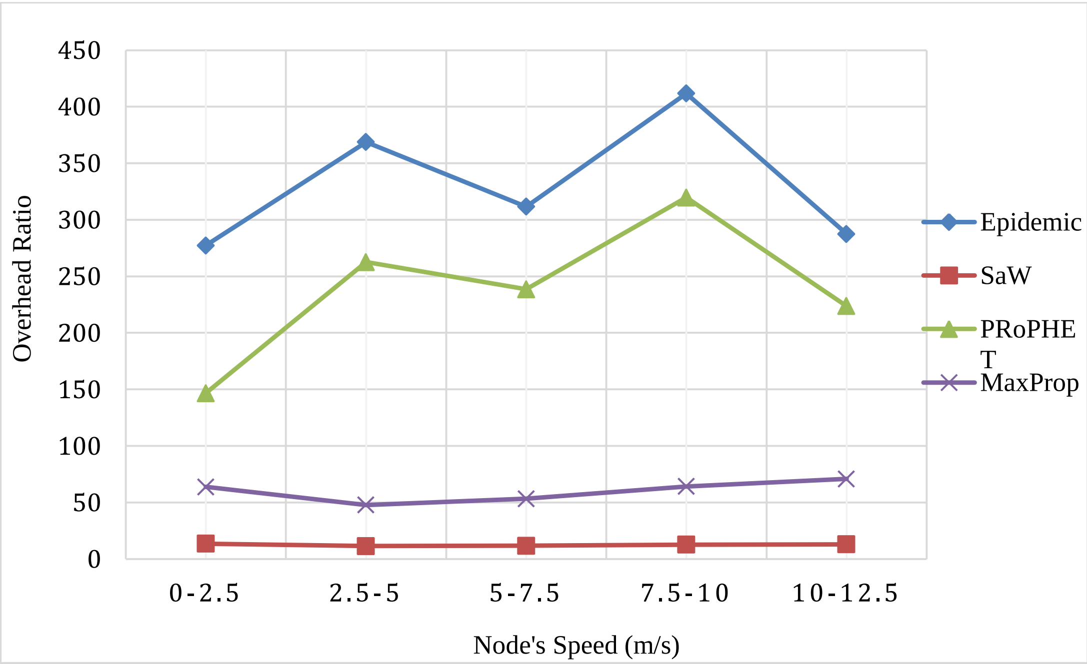

Figure 4.33: Impact of node's speed on overhead ratio using buffer size as 3 MB (number of nodes = 80, TTL = 5 h, message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

Table 4.6 shows overall results of node's average remaining energy and number of dead nodes with varying node's speed.

Table 4.6: Summary of varying Node's speed

Protocols	**Node's Average Remaining Energy**		**Number of Dead Nodes**	
	Node's Speed (m/s)			
	0-1.5	**2.5-12.5**	**0-1.5**	**2.5-12.5**
Epidemic	Medium	Low	Very low	Very high
SaW	Very high	High	Very low	Very low
PRoPHET	Medium	Low	Very low	High
MaxProp	Medium	Low	Very low	Very high

4.3.5 Impact of Time to Live

A set of TTL values is applied as 1, 3, 5, 7 and 9 h to evaluate the impact of varying TTL on performance of energy consumption, delivery ratio, average latency and overhead ratio whereas other parameters are fixed as number of nodes as 80 and 160 as a medium and large networks, message size ranged as between 0.5 – 1 MB, message generation interval set as between 100-150 seconds, buffer size with two values as 3 MB and 9 MB and node's speed range as 0.5 – 1.5 m/s for P, 1.5 – 2.5 m/s for C and 2.5 – 12.5 m/s for V.

The impact of varying TTL on energy consumption divides the protocols into two groups. It is clearly shown in Figures 4.34 - 4.38 that the node's average remaining energy of the first group (MaxProp and Saw) remains stable with all settings without any dead nodes except in large network where protocols such as MaxProp has some dead nodes but also remains stable. On the other hand, the node's average remaining energy of the other group (Epidemic and PRoPHET) reduces slightly by increasing TTL to 5 h then becomes stable without any effect as the number of dead nodes increase rapidly.

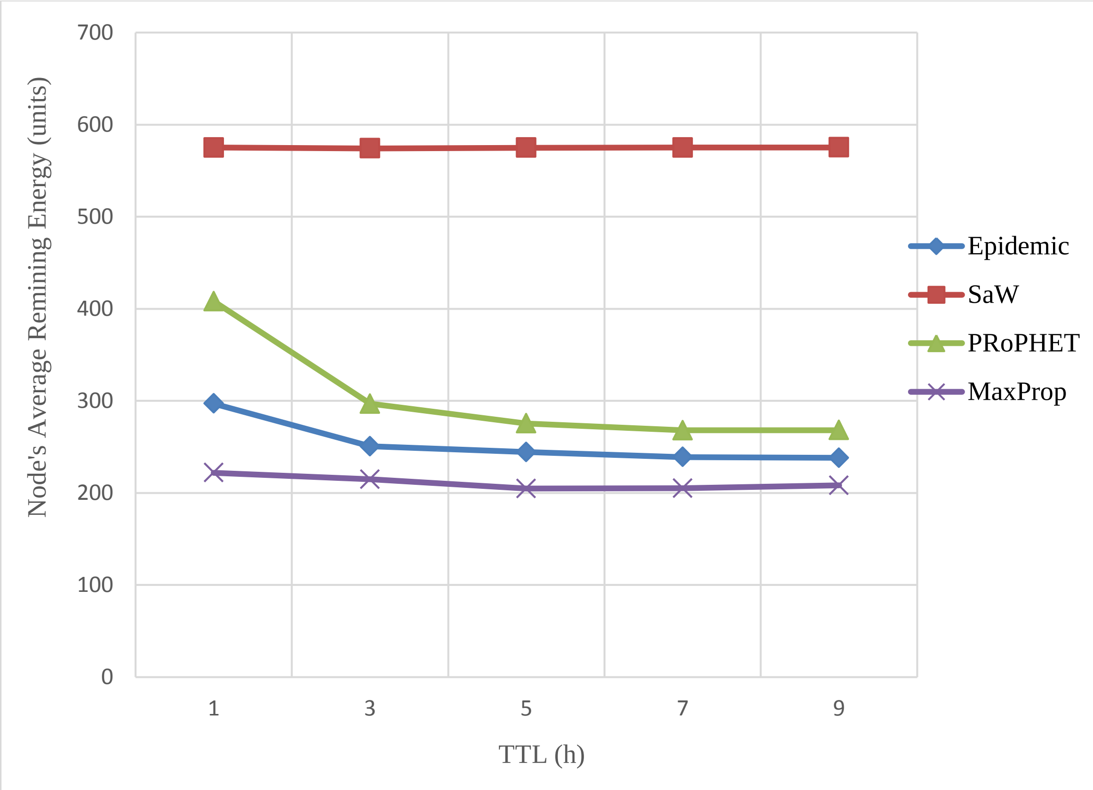

Figure 4.34: Impact of TTL on node's average remaining energy using 80 nodes and buffer size as 3 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

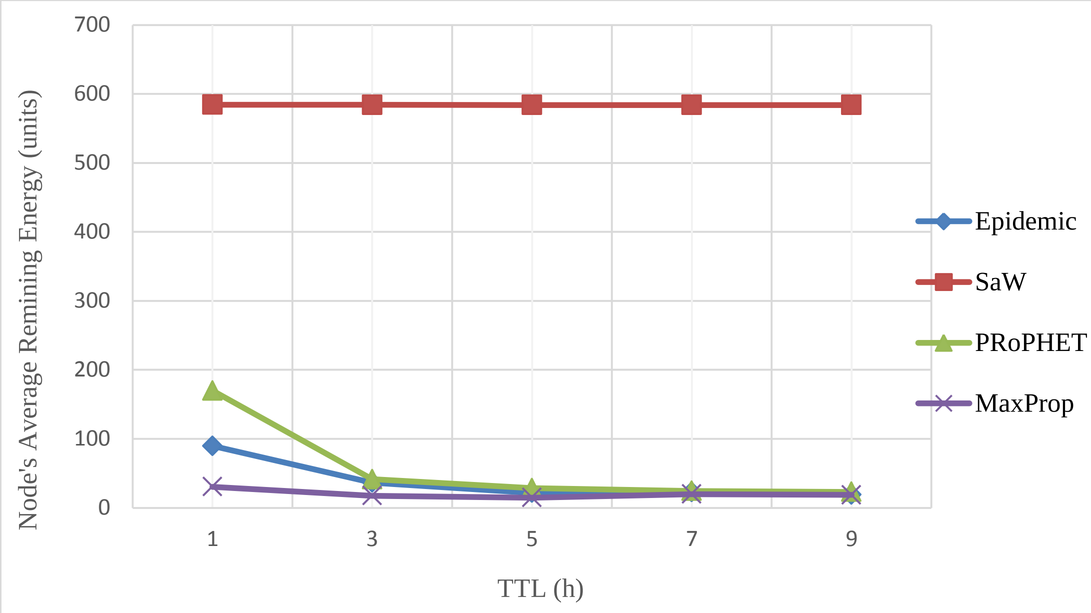

Figure 4.35: Impact of TTL on node’s average remaining energy using 160 nodes and buffer size as 3 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

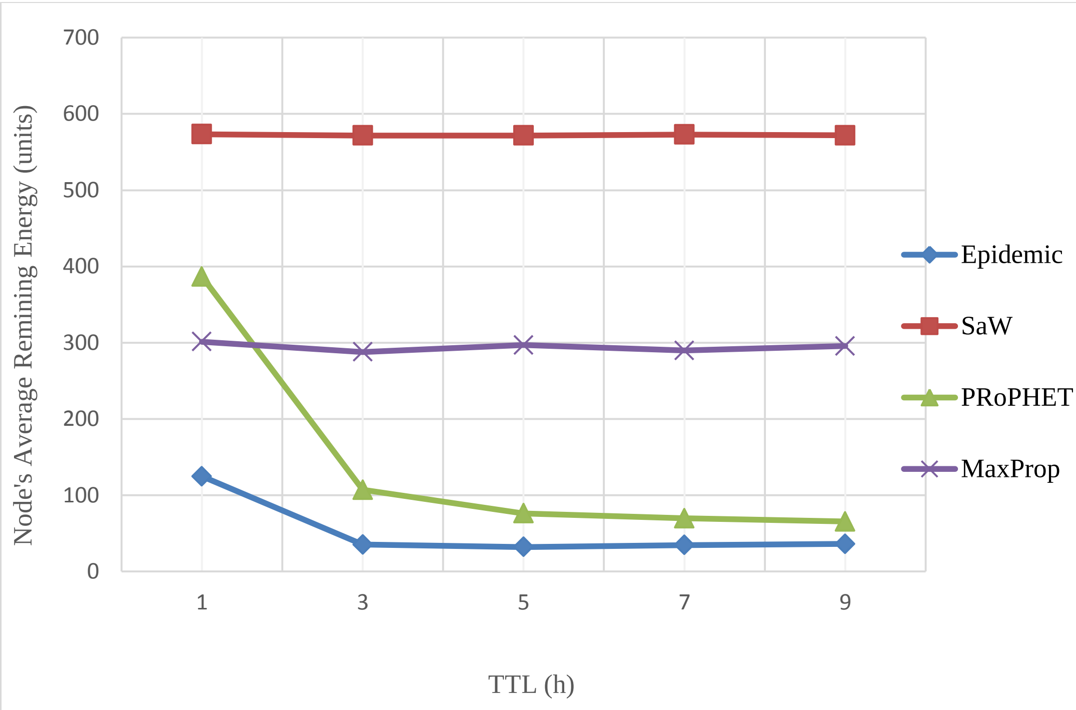

Figure 4.36: Impact of TTL on node’s average remaining energy using 80 nodes and buffer size as 9 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

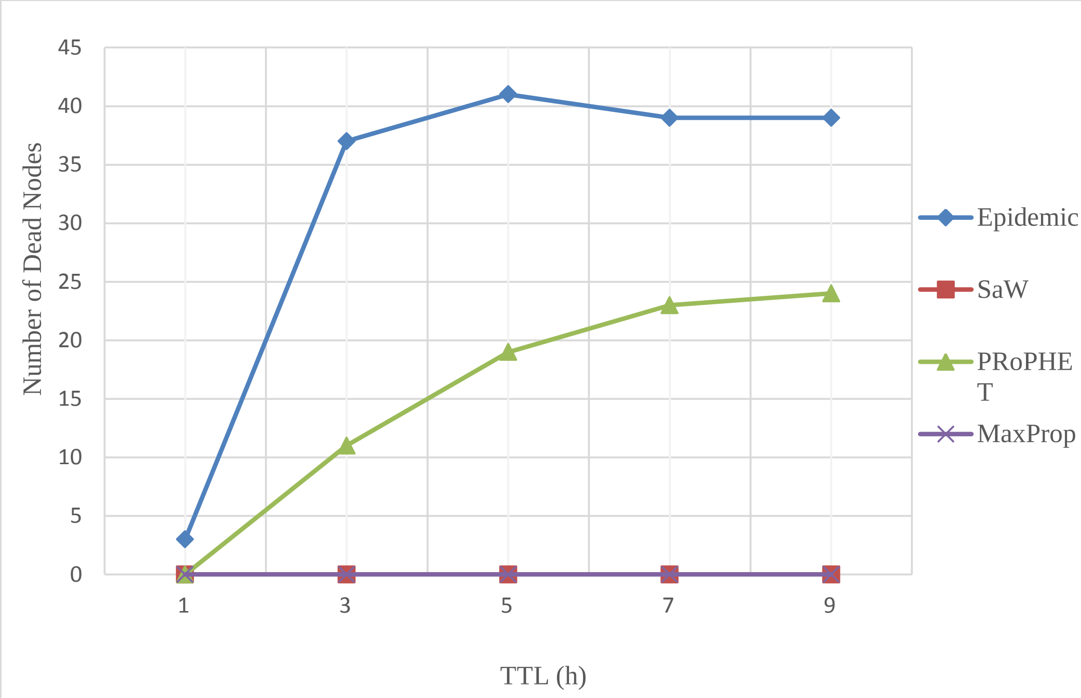

Figure 4.37: Impact of TTL on number of dead nodes using 80 nodes and buffer size as 9 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

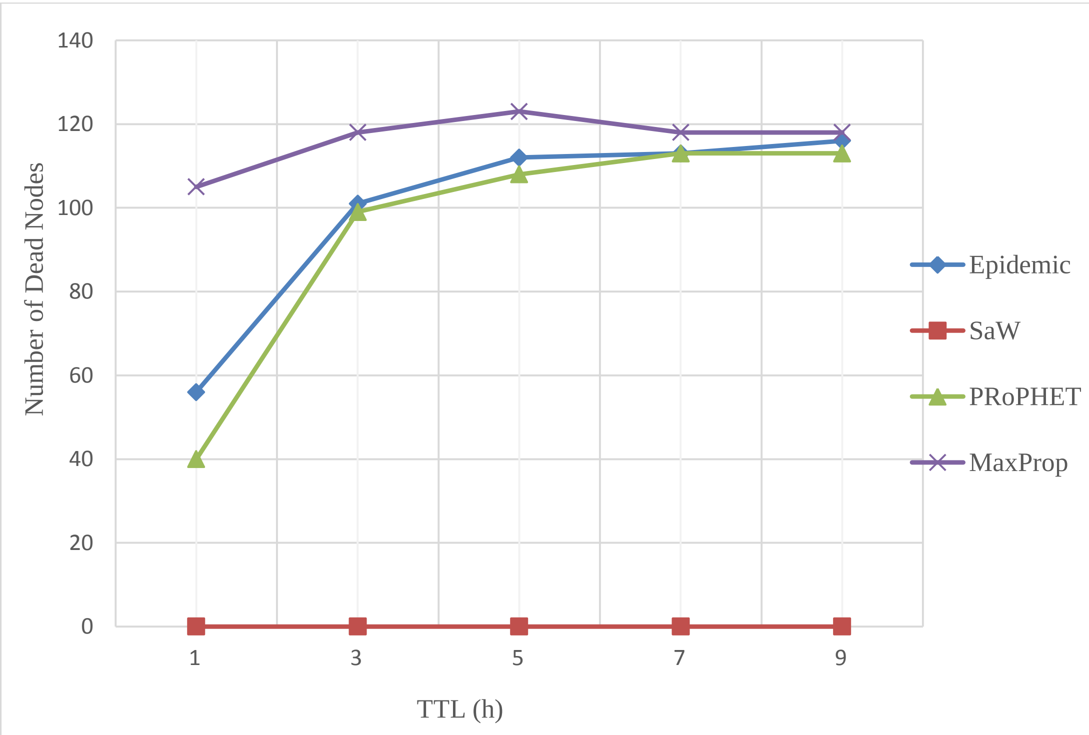

Figure 4.38: Impact of TTL on number of dead nodes using 160 nodes and buffer size as 3 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

For delivery ratio, the protocols (MaxProp and SaW) go up slightly at the beginning then stabilize due to sending of a restricted number of message duplicates. So therefore the increasing of TTL causes decreasing of total dropped message. Vice versa, the unrestricted protocols (Epidemic and PRoPHET) result that when TTL is increased, the delivery ratios are decreased as presented in Tables 4.43 and 4.44 and Figures 4.39 and 4.40. The overhead ratio of MaxProp and SaW protocols remain stabilized whereas those of the other groups (PRoPHET and Epidemic) increase as depicted in Figure 4.41.

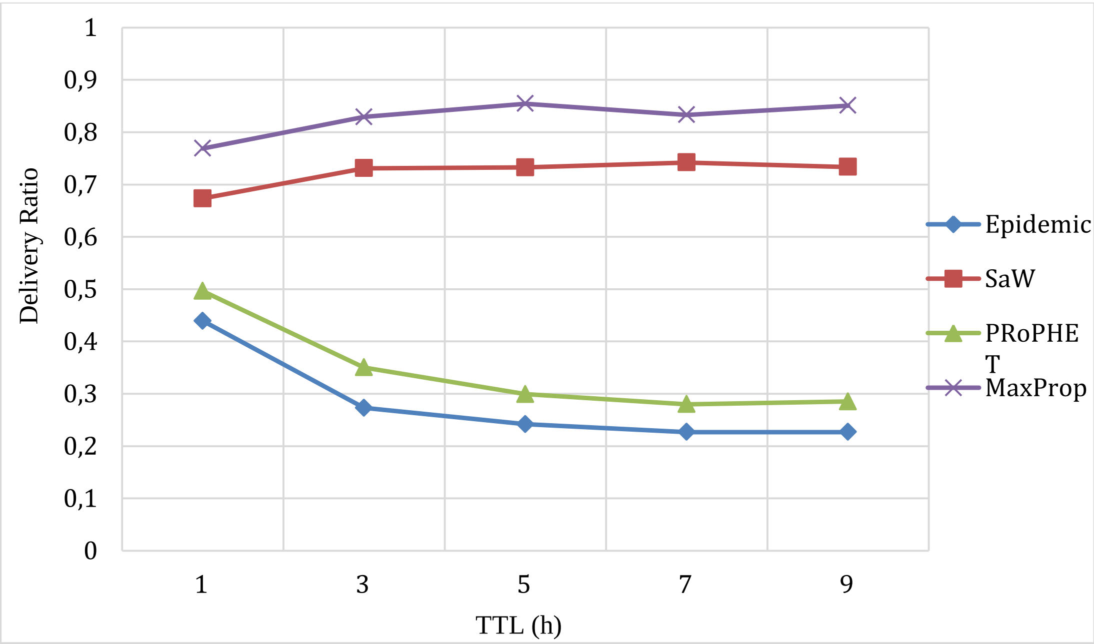

Figure 4.39: Impact of TTL on delivery ratio using 80 nodes and buffer size as 3 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

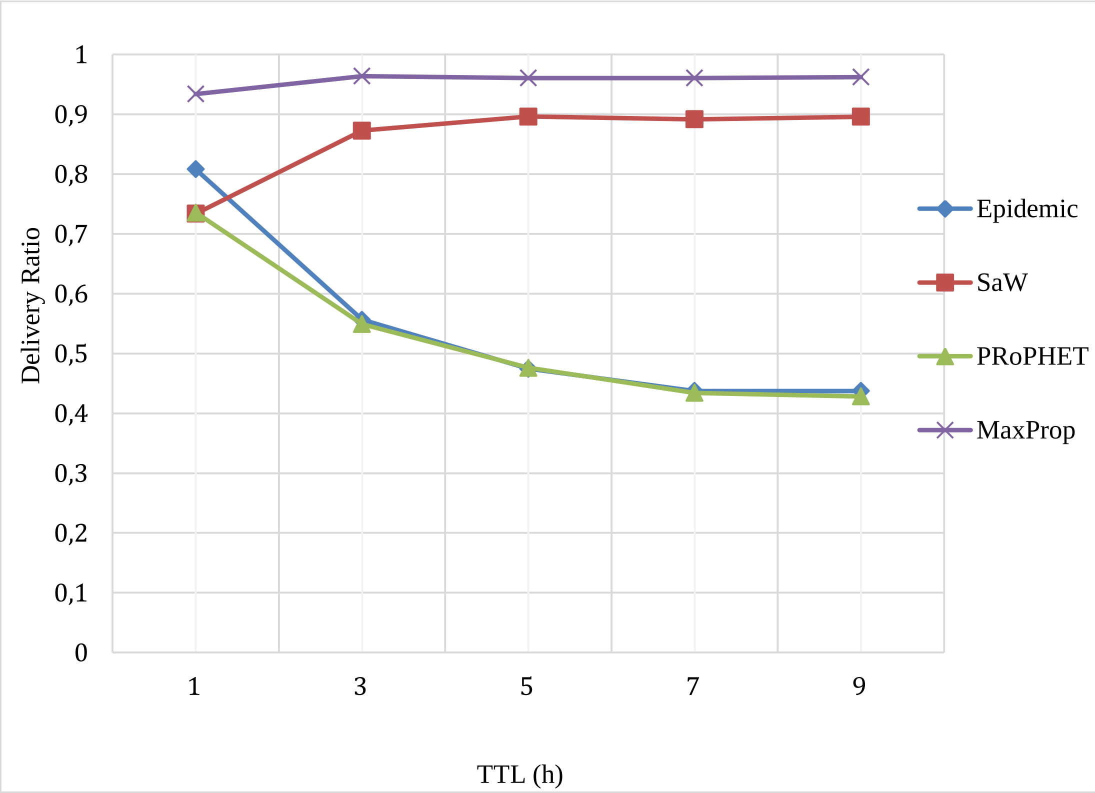

Figure 4.40: Impact of TTL on delivery ratio using 80 nodes and buffer size as 9 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

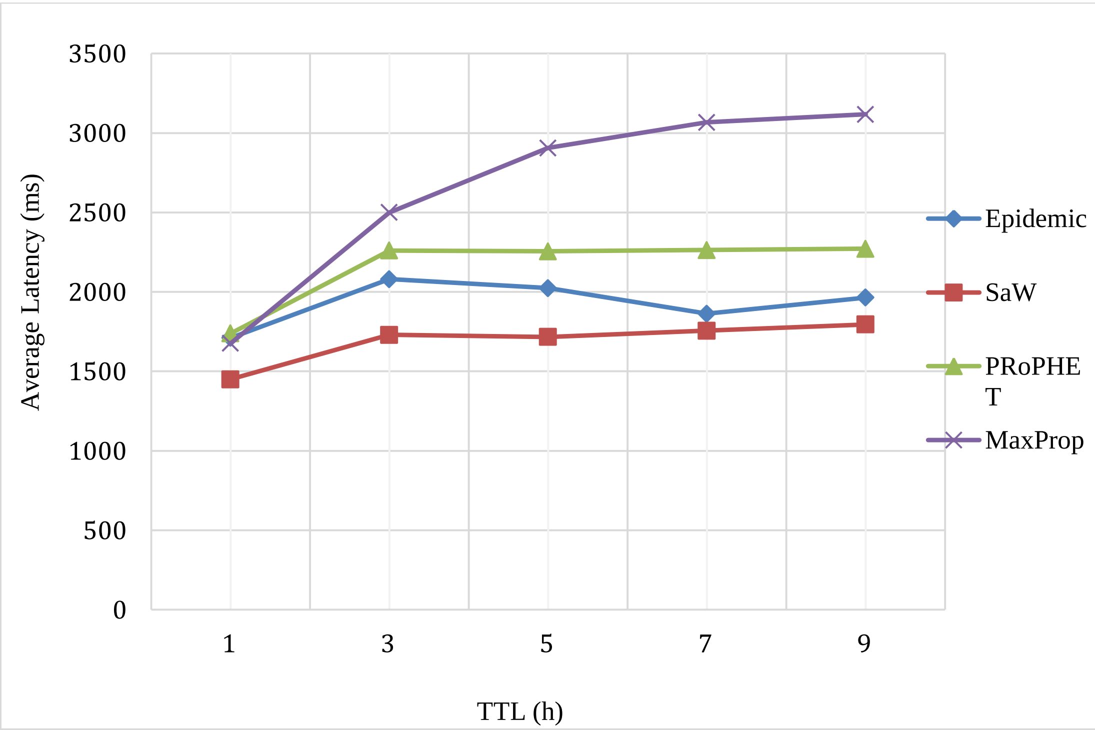

Figure 4.41: Impact of TTL on average latency using 80 nodes and buffer size as 3 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

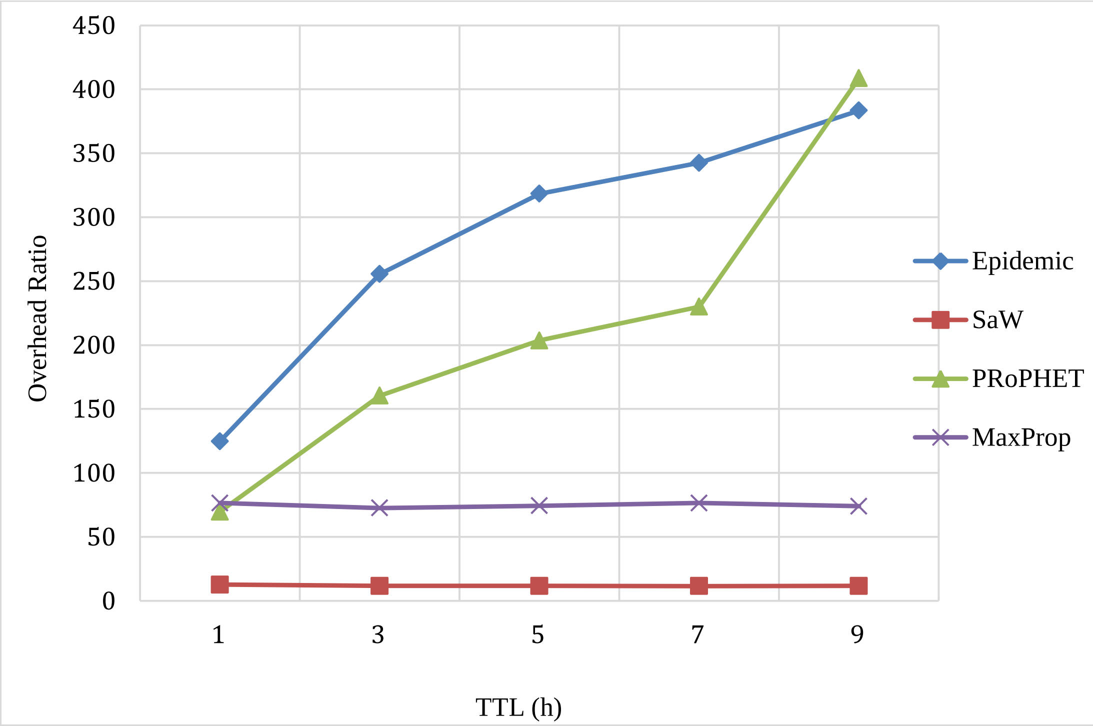

Figure 4.42: Impact of TTL on overhead ratio using 80 nodes and buffer size as 3 MB (message size = 0.5 – 1 MB, message generation interval = 100 – 150 s)

Table 4.7 shows overall results of node's average remaining energy and number of dead nodes with varying TTL.

Table 4.7: Summary of varying TTL

Protocols	**Node's Average Remaining Energy**		**Number of Dead Nodes**	
	TTL (h)			
	1	**9**	**1**	**9**
Epidemic	Medium	Medium	Very low	Medium
SaW	Very high	Very high	Very low	Very low
PRoPHET	High	Medium	Very low	Low
MaxProp	Medium	Medium	Very low	Very low

4.3.6 Impact of Buffer Size

We use a number of buffer size values as 3, 5, 7, 9 and 11 MB to evaluate the impact of varying buffer size on performance of energy consumption, delivery ratio, average latency and overhead ratio while other parameters are fixed as number of nodes as 80 and 160 for medium and large networks respectively, message size as 0.5 – 1 MB, message generation interval as 100-150 seconds and TTL as 5 h and node's speed range as 0.5 – 1.5 m/s for P, 1.5 – 2.5 m/s for C and 2.5 – 12.5 m/s for V.

Increasing the buffer size increases the number of messages which automatically leads to more energy consuming as the number of nodes dies out, especially when using unrestricted protocols, whilst the number of dead nodes goes down using restricted protocols according to their behavior that do not forward all the messages on the network blindly. Accordingly, it is illustrated in Figures 4.43 - 4.46 that varying buffer size has a reverse effect on node's average remaining energy in unrestricted protocols (Epidemic and PRoPHET) which decreases during rising buffer capacity particularly in large networks and the number of dead nodes increases sporadically especially in large networks. This results in all nodes dying off when buffer size is 5 MB. On the other hand, MaxProp protocol increases slightly in both networks without any dead nodes in medium network and the number of dead nodes goes down in large network. Whereas SaW protocol remains stable with all settings without any dead nodes.

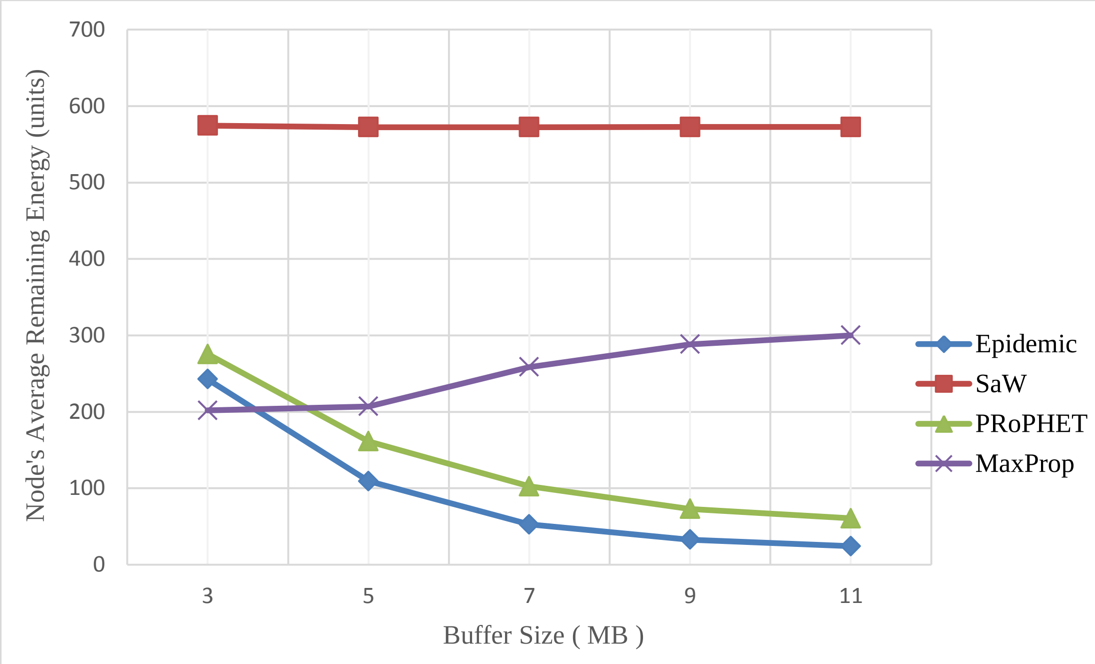

Figure 4.43: Impact of buffer size on node’s average remaining energy using 80 nodes (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

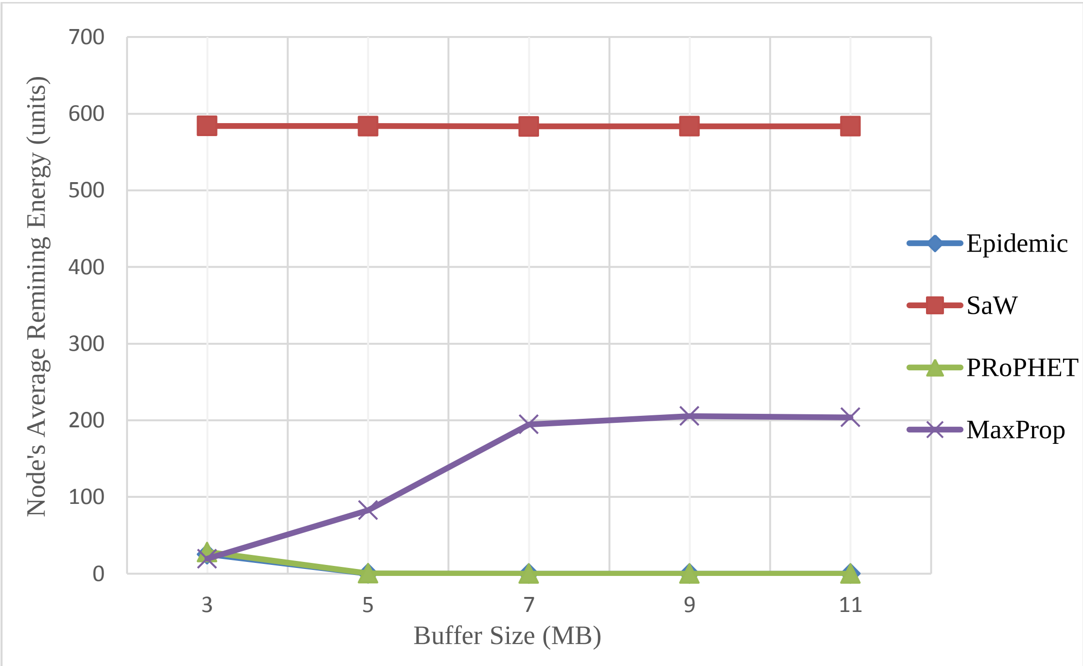

Figure 4.44: Impact of buffer size on node’s average remaining energy using 160 nodes (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

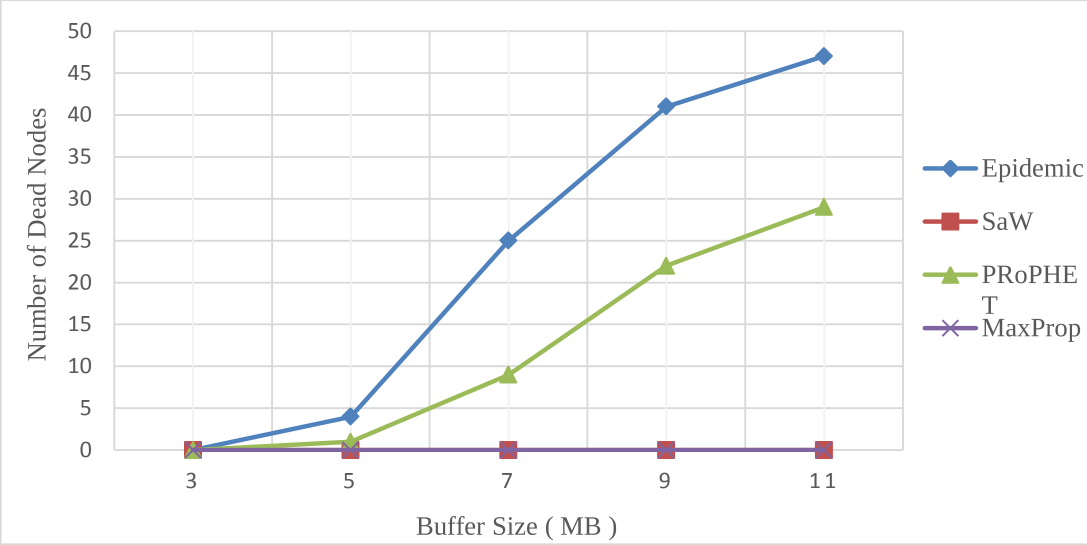

Figure 4.45: Impact of buffer size on number of dead nodes using 80 nodes (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

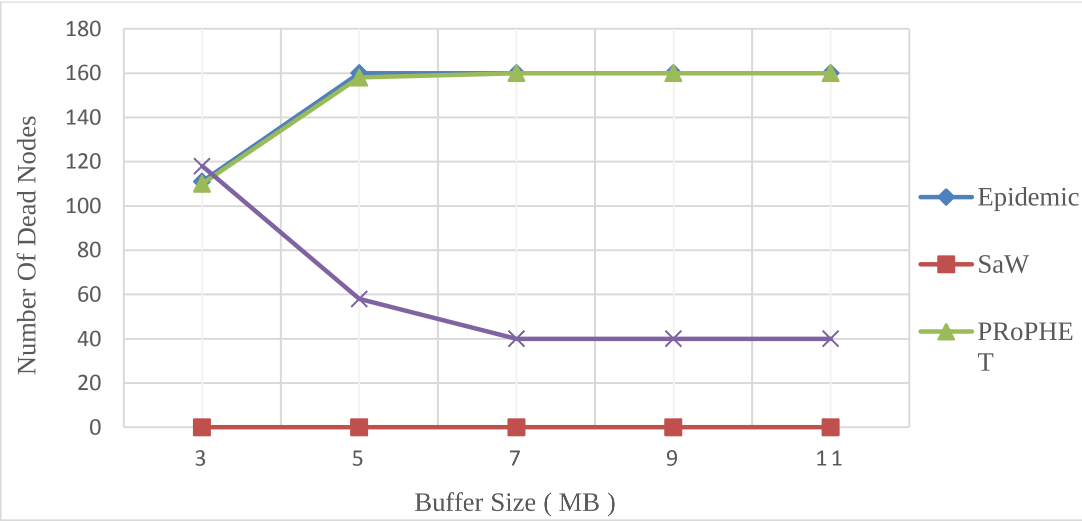

Figure 4.46: Impact of buffer size on number of dead nodes using 160 nodes (message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

According to the same reason, the delivery ratio of all protocols increases but the average latency of Epidemic and PRoPHET protocols increase whereas SaW and MaxProp protocols decrease. Moreover, the overhead ratio of all protocols decreases except SaW which shows no effect as shown in Figures 4.47 - 4.49 respectively.

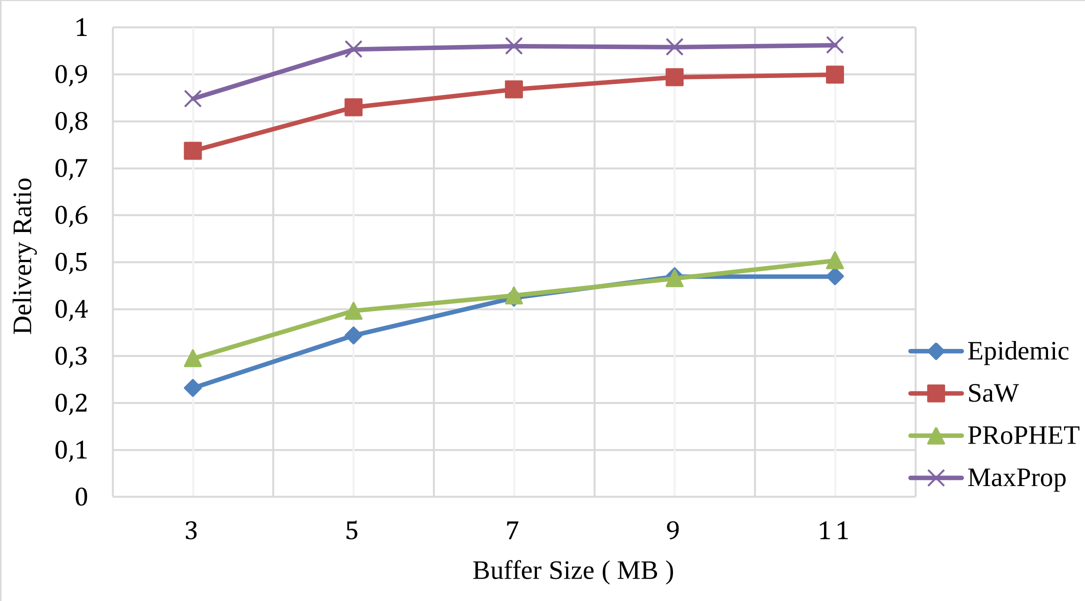

Figure 4.47: Impact of buffer size on delivery ratio (number of nodes = 80, message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

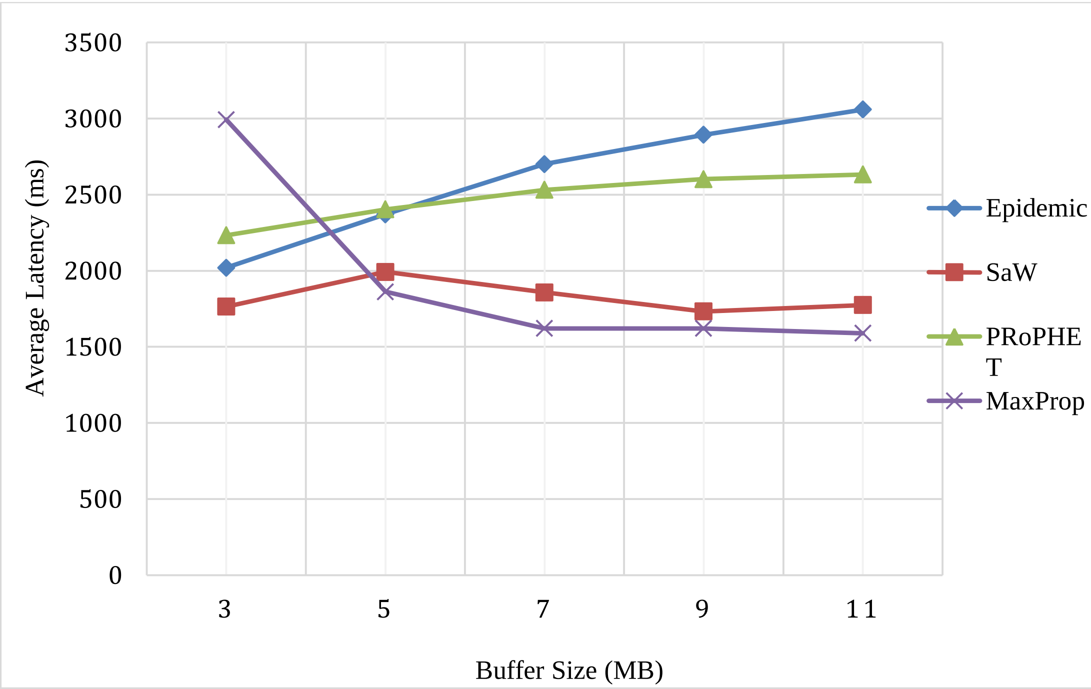

Figure 4.48: Impact of buffer size on average latency (number of nodes = 80, message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

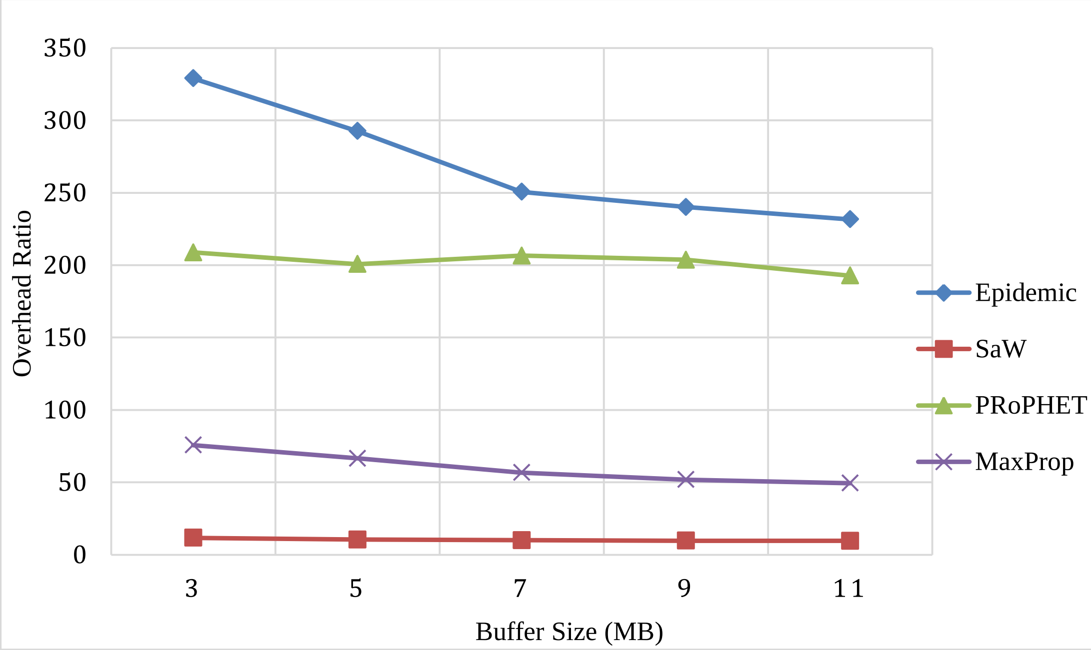

Figure 4.49: Impact of buffer size on overhead ratio (number of nodes = 80, message size = 0.5 – 1 MB, TTL = 5 h, message generation interval = 100 – 150 s)

Table 4.8 shows overall results of node's average remaining energy and number of dead nodes with varying buffer size. Average values of all simulation results are provided in Appendix C.

Table 4.8: Summary of varying buffer size

Protocols	**Node's Average Remaining Energy**		**Number of Dead Nodes**	
	Buffer size (MB)			
	3	**11**	**3**	**11**
Epidemic	Medium	Low	Very low	High
SaW	Very high	Very high	Very low	Very low
PRoPHET	Medium	Low	Very low	Medium
MaxProp	Medium	Medium	Very low	Very low

Chapter 5

CONCLUSION AND FUTURE WORK

DTNs, also referred to as opportunistic networks are scatter host networks in which an end-to-end connection between source and its destination may not exist. The hosts in DTNs are required to transmit and hold on data bundles until it comes in contact with another host. Holding this data may take quite a while and this dismisses one of the main assumptions of classic routing protocols. Moreover, this issue motivates researchers and scientists to find innovative solutions to solve the problem.

In this research, it is provided a study of four well-known DTN routing protocols. Epidemic is an instance of a blind routing protocol which uses the flooding strategy approach. MaxProp is an instance of a guided routing protocol which favors messages with minimum hops. PRoPHET is an instance of a guided routing protocol with a first in first out message selection mechanism. SaW is an instance of a partial-flooding blind-routing protocol which also applies the flooding strategy.

This thesis investigates and compares all these routing protocols in terms of energy consumption, delivery ratio, average latency and overhead ratio performance metrics. Parameters such as number of nodes, message size, node's speed and message generation interval are varied to observe their impact on performance. We have used ONE simulator where the hosts travel in a network using the Helsinki map which is available in the simulator. It can be concluded from the results that SaW protocol outclasses other protocols in terms of energy consumption whereas MaxProp protocol has the highest delivery ratio. Epidemic, on the other hand, seems to show the worst performance. It is observed that the performance of routing protocols is affected by the varying the message generation interval, number of nodes, node's speed, message size, TTL and buffer size. It is shown that the node's average remaining energy reduces and dead nodes increases with increments in node's speed, number of nodes, message size, TTL and buffer size and node's average remaining energy rise and dead nodes reduce when message generation interval increase. Moreover, the delivery ratio goes up by increasing of buffer size and message generation interval whereas decreases by rising TTL while using flooding protocols and remains stable while using forwarding protocols also decreases by increasing message size. Furthermore, the average latency increases by increasing message generation interval, message size, TTL and buffer size whereas goes down by rising number of nodes. Moreover, the overhead ratio reduces by increasing buffer size and rises by increasing number of nodes and remains stable with the forwarding protocols by increasing message generation interval, message size and TTL. On the other hand it increases using flooding protocols by increasing TTL and message generation interval and decreases by increase message size.

Recommendations for future research and development studies regarding DTN must aim towards developing new energy efficient routing protocols which take into deliberation the rate of energy remaining in various nodes in the network based on the results provided here and compared it with the existing protocols.

REFERENCES

[1] Archana, R. & Gracy, T. (2015, September). Survey on Attacks in MANET. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(9), 125-31.

[2] Ramesh, S., & Indira, R. (2016). Ant Routing Protocol with Location Services in Intermittently Connected MANETs, *Circuits and Systems*, 7(7), 1087.

[3] Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad Hoc On-Demand Distance Vector (Aodv) Routing (RFC 3561).

[4] Johnson, D., Hu, Y. C., & Maltz, D. (2007). The Dynamic Source Routing (DSR) Protocol For Mobile Ad Hoc Networks for Ipv4 (RFC 4728).

[5] Chawla, M. (2013). Comparing Delay Tolerant Network Routing Protocols for Optimizing L-Copies in Spray and Wait Routing for Minimum Delay. *Atlantis Press*, 239-244.

[6] Bista, B. B., & Rawat, D. B. (2017). EA-Epidemic: An Energy Aware Epidemic-Based Routing Protocol for Delay Tolerant Networks. *Journal of Communications*, 12(6).

[7] Fall, K. (2003, August). A delay-tolerant Network Architecture for Challenged Internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 27-34.

[8] Mangrulkar, R. S., & Atique, M. (2010, October). Routing Protocol for Delay Tolerant Network: A Survey and Comparison. In Communication Control and Computing Technologies (ICCCCT), *IEEE International Conference*, 210-215.

[9] Shen, J., Moh, S., & Chung, I. (2008). Routing Protocols in Delay Tolerant Networks: A Comparative Survey. In the 23rd International Technical Conference on Circuits/Systems, *Computers and Communications, (ITC-CSCC 2008)*, 6-9.

[10] Khabbaz, M. J., Assi, C. M., & Fawaz, W. F. (2012). Disruption-tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges. *IEEE Communications Surveys & Tutorials*, 14(2), 607-640.

[11] Fall, K., Scott, K. L., Burleigh, S. C., Torgerson, L., Hooke, A. J., Weiss, H. S., ... & Cerf, V. (2007). Delay-tolerant Networking Architecture, 15-18.

[12] Caini, C., Cornice, P., Firrincieli, R., & Lacamera, D. (2008). A DTN Approach to Satellite Communications. *IEEE Journal on Selected Areas in Communications*, 26(5), 820-827.

[13] Pentland, A., Fletcher, R., & Hasson, A. (2004). Daknet: Rethinking Connectivity in Developing Nations. *Computer*, 37(1), 78-83.

[14] Abdel-kader, T. A. M. M. (2012). Design and Performance Analysis of Opportunistic Routing Protocols for Delay Tolerant Networks.

[15] Qiao, J., Shen, X. S., Mark, J. W., Shen, Q., He, Y., & Lei, L. (2015). Enabling Device-To-Device Communications in Millimeter-Wave 5G Cellular Networks. *IEEE Communications Magazine*, 53(1), 209-215.

[16] Cheng, M. X., Ye, Q., & Cai, L. (2015). Rate-adaptive Concurrent Transmission Scheduling Schemes for WPANS with Directional Antennas. *IEEE Transactions on Vehicular Technology*, 64(9), 4113-4123.

[17] D’souza, R. J., & Jose, J. (2010). Routing Approaches in Delay Tolerant Networks: A Survey. *International Journal of Computer Applications*, 1(17), 8-14.

[18] Ntareme, H., & Domancich, S. (2011, October). Security and Performance Aspects of Bytewalla: A Delay Tolerant Network on Smartphones. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, 449-454.

[19] Jones, E. P., & Ward, P. A. (2006). Routing Strategies for Delay-Tolerant Networks. *Submitted to ACM Computer Communication Review (CCR)*, 1-10.

[20] Benamar, N., Singh, K. D., Benamar, M., El Ouadghiri, D., & Bonnin, J. M. (2014). Routing Protocols in Vehicular Delay Tolerant Networks: A Comprehensive Survey. *Computer Communications*, 48, 141-158.

[21] Liu, M., Yang, Y., & Qin, Z. (2011). A Survey of Routing Protocols and Simulations in Delay-Tolerant Networks. Wireless Algorithms, Systems, and Applications, 243-253.

[22] Vahdat, A., & Becker, D. (2000). Epidemic Routing For Partially Connected Ad Hoc Networks. Duke University, Durham, NC 27708.

[23] Bloom, B. H. (1970). Space/Time Trade-Offs in Hash Coding With Allowable Errors. *Communications of the ACM*, 13(7), 422-426.

[24] Fan, L., Cao, P., Almeida, J., & Broder, A. Z. (1998, October). Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. *In ACM SIGCOMM Computer Communication Review ACM*, 28(4), 254-265.

[25] Terry, D. B., Theimer, M. M., Petersen, K., Demers, A. J., Spreitzer, M. J., & Hauser, C. H. (1995). Managing Update Conflicts In Bayou, A Weakly Connected Replicated Storage System. *In ACM SIGOPS Operating Systems Review ACM*, 29(5) 172-182.

[26] Burgess, J., Gallagher, B., Jensen, D. D., & Levine, B. N. (2006). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In Infocom. Massachusetts University.

[27] Lindgren, A., Doria, A., & Schelén, O. (2003). Probabilistic Routing in Intermittently Connected Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3), 19-20.

[28] Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2005). Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. 0*In Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking ACM*, 252-259.

[29] Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2008). Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case. *IEEE/ ACM Trans. Networking (ToN)*, 16(1), 77-90.

[30] Friedman, R., Kogan, A., & Krivolapov, Y. (2013). On Power and Throughput Tradeoffs of Wifi and Bluetooth in Smartphones. *IEEE Transactions on Mobile Computing*, 12(7), 1363-1376.

[31] Cabacas, R. A., Nakamura, H., & Ra, I. H. (2014). Energy Consumption Analysis of Delay Tolerant Network Routing Protocols. *International Journal of Software Engineering and Its Applications*, 8(2), 1-10.

[32] Bista, B. B., & Rawat, D. B. (2016, January). Energy Consumption and Performance of Delay Tolerant Network Routing Protocols under Different Mobility Models. In *Intelligent Systems, Modelling and Simulation (ISMS), 2016 7th International Conference on*, 325-330.

[33] Socievole, A., & Marano, S. (2012, August). Evaluating the Impact of Energy Consumption on Routing Performance in Delay Tolerant Networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, 481-486.

[34] Abdelkader, T., Naik, K., Nayak, A., Goel, N., & Srivastava, V. (2016). A Performance Comparison of Delay-Tolerant Network Routing Protocols. *IEEE Network*, 30(2), 46-53.

[35] Cuka, M., Shinko, I., Spaho, E., Oda, T., Ikeda, M., & Barolli, L. (2017). A Simulation System Based on ONE and SUMO Simulators: Performance Evaluation of Different Vehicular DTN Routing Protocols. *Journal of High Speed Networks*, 23(1), 59-66.

[36] Samyal, V. K., & Sharma, Y. K. (2017). Performance Evaluation of Delay Tolerant Networks Routing Protocols under varying Time to Live. *International Journal*, 8(1), 299-302.

[37] Alajeely, M., Doss, R., & Ahmad, A. (2017). Routing Protocols in Opportunistic Networks–A Survey. *IETE Technical Review*, ISSN: 0256-4602, 1-19.

[38] Keranen, A. (2008). Opportunistic Network Environment Simulator. Special Assignment report, Helsinki University of Technology, Department of Communications and Networking.

[39] Keränen, A and Ott, J. (2013, November 4), The ONE. Retrieved from https://www.netlab.tkk.fi/tutkimus/dtn/theone/javadoc_v141/.

[40] Abdelkader, T., Naik, K., Nayak, A., Goel, N., & Srivastava, V. (2013). SGBR: A Routing Protocol for Delay Tolerant Networks Using Social Grouping. IEEE Transactions on Parallel and Distributed Systems, 24(12), 2472-2481.

[41] Keränen, A. (2017, April 17), The-one. Retrieved from https://github.com/akeranen/the-one

APPENDICES

Appendix A: Algorithms of Protocols

A1: Epidemic Routing Protocol Algorithm

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
if ContactDuration > 0 then
 pkt=GetPacket(a)
 if pkt then
     if NotReceivedBefore(pkt,b) then
         if IsDestination(pkt,b) then
                 SendPacket(pkt,a)
                 ConsumePacket(pkt,b)
             else
                 SendPacket(pkt,a)
                 StorePacket(pkt,b)
             end if
             ContactDuration=ContactDuration-size(pkt)
         end if
     end if
end if
```

A2: PRoPHET Routing Protocol Algorithm

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
UpdateDeliveryPredictability()
if ContactDuration > 0 then
 pkt=GetPacket(a)
```

```
    if pkt then
        if NotReceivedBefore(pkt,b) then
                if IsDestination(pkt,b) then
                    SendPacket(pkt,a)
                    ConsumePacket(pkt,b)
                else
                    DPn1=DeliveryPredictability(pkt,a)
                    DPn2=DeliveryPredictability(pkt,b)
                    if DPn2 > DPn1 then
                        SendPacket(pkt,a)
                        StorePacket(pkt,b)
                endif
            endif
            ContactDuration=ContactDuration-size(pkt)
        endif
    endif
endif
```

A3: MaxProp Routing Protocol Algorithm

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
UpdateDeliveryPredictability()
SortPackets() /* Using MAXPROP sorting criteria */
if ContactDuration > 0 then
 pkt=GetPacket(a)
 /* pkt is the packet with the minimum hop count, or higher delivery predictability */
    if pkt then
        if NotReceivedBefore(pkt,b) then
            if IsDestination(pkt,b) then
                SendPacket(pkt,a)
```

```
                ConsumePacket(pkt,b)
            else
                SendPacket(pkt,a)
                StorePacket(pkt,b)
            endif
            ContactDuration=ContactDuration-size(pkt)
        endif
    endif
endif
```

A4: Spray and Wait Routing Protocol Algorithm

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
if ContactDuration > 0 then
pkt=GetPacket(a)
if pkt then
    if NotReceivedBefore(pkt,b) then
        if IsDestination(pkt,b) then
            SendPacket(pkt,a)
            ConsumePacket(pkt,b)
        else
            NrOfCopies=GetNrOfCopies(pkt,a)
            if NrOfCopies > 1 then
                SendPacket(pkt,a)
                StorePacket(pkt,b)
                SetNrOfCopies(pkt,a,NrOfCopies/2)
                SetNrOfCopies(pkt,b,NrOfCopies/2)
            endif
        endif
        ContactDuration=ContactDuration-size(pkt)
    endif
```

```
    endif
endif
```

All previous algorithms are presented in [35]

Appendix B: Screenshots of the ONE Simulator

B1: GUI Interface

We can run the ONE simulator in GUI mode by run its patch file which is “one.bat” that exist in the simulator folder. Figure B.1 illustrates the interface of the simulator.

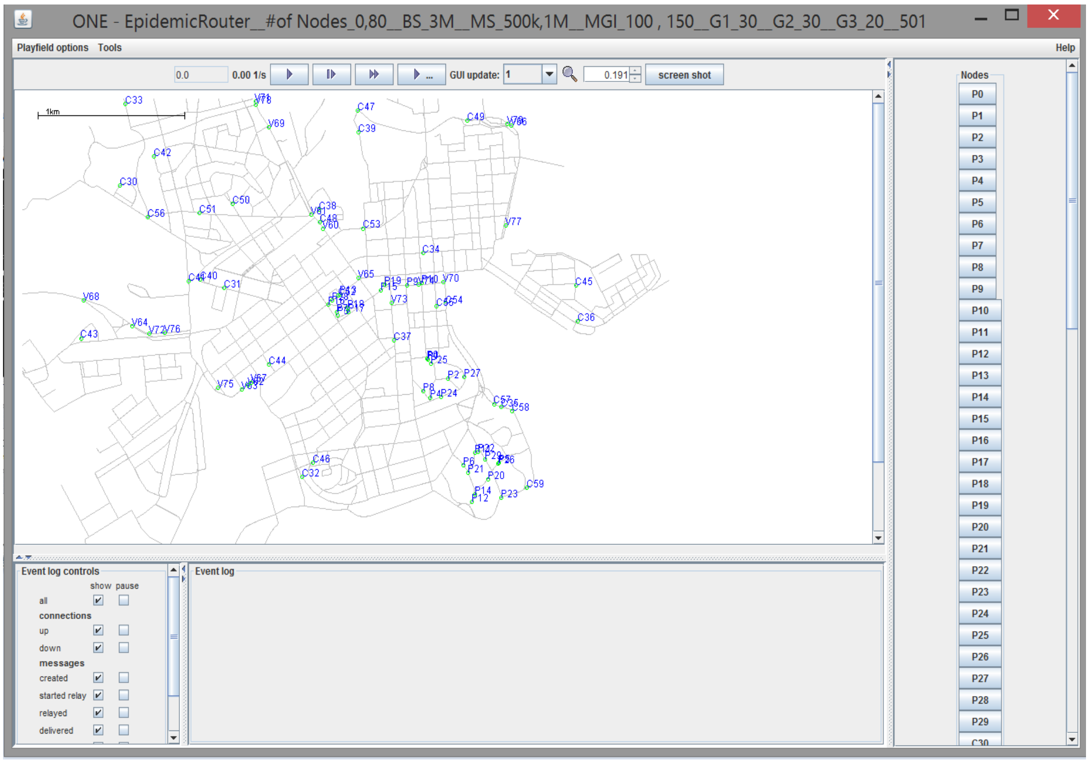

Figure B.1: The ONE simulator GUI interface

B2: Setting File

The ONE simulator takes the setting of the network parameters for each scenario from a text file with “txt” extinction that must be saved in the same folder with the patch file of the simulator. This files includes a scenarios names and the path of the results reports along with the values of each parameter as follow:

Scenario settings
Scenario.name = %%Group.router%%__#of Nodes_%%Events1.hosts%%__BS_%%Group.bufferSize%%__MS_%%Events1.size%%__MGI_%%Events1.interval%%__G1_%%Group1.nrofHosts%%__G2_%%Group2.nrofHosts%%__G3_%%Group3.nrofHosts%%__%%MovementModel.rngSeed%%
Scenario settings
Scenario.simulateConnections = true

```
Scenario.updateInterval = 0.1
# 43200s == 12h
Scenario.endTime = 43200
## Interface-specific settings:
# type : which interface class the interface belongs to
# For different types, the sub-parameters are interface-specific
# For SimpleBroadcastInterface, the parameters are:
# transmitSpeed : transmit speed of the interface (bytes per second)
# transmitRange : range of the interface (meters)
# "Bluetooth" interface for all nodes
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10
#Energy Settings
# initial amount of energy units
Group.initialEnergy = 5000
# the amount of energy taken by each scan
Group.scanEnergy = 0.1
# the amount of energy taken each second when transferring
Group.transmitEnergy = 0.2
# the amount of energy taken each second when receiving
Group.scanResponseEnergy = 0.1
# the amount of energy taken each second when the node is idle
Group.baseEnergy = 0.01
## Group-specific settings:
# groupID : Group's identifier. Used as the prefix of host names
# nrofHosts: number of hosts in the group
# movementModel: movement model of the hosts (valid class name from movement
package)
# waitTime: minimum and maximum wait times (seconds) after reaching destination
# speed: minimum and maximum speeds (m/s) when moving on a path
# bufferSize: size of the message buffer (bytes)
# router: router used to route messages (valid class name from routing package)
# activeTimes: Time intervals when the nodes in the group are active (start1, end1,
start2, end2, ...)
# msgTtl : TTL (minutes) of the messages created by this host group, default=infinite
#      for ShortestPathMapBasedMovement
# okMaps : which map nodes are OK for the group (map file indexes), default=all
#         for all MapBasedMovent models
## Message creation parameters
# How many event generators
Events.nrof = 1
# Class of the first event generator
Events1.class = MessageEventGenerator
# Message ID prefix
Events1.prefix = M
# (following settings are specific for the MessageEventGenerator class)
# Creation interval in seconds (one new message every 100 to 150 seconds)
Events1.interval =  100 , 150
```

# Message sizes (500k - 1M)
Events1.size = 500k,1M
# range of message source/destination addresses
Events1.hosts = 0,40
Group.bufferSize = [3M; 5M; 7M; 9M; 11M]
# Message TTL of 240 minutes (4 hours)
Group.msgTtl = 300
# Define 3 different node groups
Scenario.nrofHostGroups = 3
# Common settings for all groups
Group.movementModel = ShortestPathMapBasedMovement
# All nodes have the bluetooth interface
Group.nrofInterfaces = 1
Group.interface1 = btInterface
# World's size for Movement Models without implicit size (width, height; meters)
MovementModel.worldSize = 4500, 3400
Group.router = [EpidemicRouter; SprayAndWaitRouter; ProphetRouter; MaxPropRouter]
# group1 (pedestrians) specific settings
Group1.groupID = P
# pedestrians can walk only on pedestrian's paths
Group1.okMaps = 3
# Walking speeds
Group1.speed = 0.5, 1.5
Group1.nrofHosts = 15
# group2 specific settings
Group2.groupID = C
Group2.okMaps = 3
# Cyclist speeds
Group2.speed = 1.5, 2.5
Group2.nrofHosts = 15
# group3 specific settings
Group3.groupID = V
# Vehicle can drive only on roads
Group3.okMaps = 2
# Vehicle speeds
Group3.speed = 2.5, 12.5
Group3.nrofHosts = 10

Map based movement -movement model specific settings
MapBasedMovement.nrofMapFiles = 3
MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
Movement model settings
# World's size for Movement Models without implicit size (width, height; meters)
MovementModel.worldSize = 4500, 3400
# seed for movement models' pseudo random number generator (default = 0)
MovementModel.rngSeed=[501;502;503;504;505;506;507;508;509;510;511;512;513
;514;515;516;517;518;519;520;521;522;523;524;525;526;527;528;529;530;531;532;

```
533;534;535;536;537;538;539;540;541;542;543;544;545;546;547;548;549;550;551;
552;553;554;555;556;557;558;559;560;561;562;563;564;565;566;567;568;569;570;
571;572;573;574;575;576;577;578;579;580;581;582;583;584;585;586;587;588;589;
590;591;592;593;594;595;596;597;598;599;500;601;602;603;604;605;606;607;608;
609;610;611;612;613;614;615;616;617;618;619;620;621;622;623;624;625;626;627;
628;629;630;631;632;633;634;635;636;637;638;639;640;641;642;643;644;645;646;
647;648;649;650;651;652;653;654;655;656;657;658;659;660;661;662;663;664;665;
666;667;668;669;670;671;672;673;674;675;676;677;678;679;680;681;682;683;684;
685;686;687;688;689;690;691;692;693;694;695;696;697;698;699;701;702;703;704;
705;706;707;708;709;710;711;712;713;714;715;716;717;718;719;720;721;722;723;
724;725;726;727;728;729;730;731]
## Reports - all report names have to be valid report classes
# how many reports to load
Report.nrofReports = 2
Report.report1 = EnergyLevelReport
Report.report2 = MessageStatsReport
# length of the warm up period (simulated seconds)
Report.warmup = 0
# report once every 43200 minutes
EnergyLevelReport.granularity = 43200
# round to two decimal's precision
EnergyLevelReport.precision = 2
Report.reportDir = reports/1
## Default settings for some routers settings
ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 10
SprayAndWaitRouter.binaryMode = true

## Default for Contact router
ContactRouter.nrOfCopies = 10
ContactRouter.deadline = 5.0
ContactRouter.totalNodes = 40
## Optimization settings -- these affect the speed of the simulation
## see World class for details.
Optimization.connectionAlg = 2
Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true
MaxPropRouter.PROB_SET_MAX_SIZE_S= "probSetMaxSize"
MaxPropRouter.probSetMaxSize= 500
## GUI settings
# GUI underlay image settings
GUI.UnderlayImage.fileName = data/helsinki_underlay.png
# Image offset in pixels (x, y)
GUI.UnderlayImage.offset = 64, 20
# Scaling factor for the image
```

GUI.UnderlayImage.scale = 4.75
# Image rotation (radians)
GUI.UnderlayImage.rotate = -0.015
# how many events to show in the log panel (default = 30)
GUI.EventLogPanel.nrofEvents = 100
# Regular Expression log filter (see Pattern-class from the Java API for RE-matching details)
#GUI.EventLogPanel.REfilter = .*p[1-9]<->p[1-9]$

B3: Report Files

The ONE simulator creates a report files that contain the results of performance metrics and the results are shown as in Figures (B.2 and B.3) where each file named based on the protocol name and the scenario name.

المفكرة - EpidemicRouter__#of Nodes_0,40__BS_3M__MS_500k,1M__MGI_100 , 150__G1_15__G2_15__G3_10__501_MessageStatsReport.txt

ملف تحرير تنسيق عرض تعليمات

Message stats for scenario EpidemicRouter__#of Nodes_0,40__BS_3M__MS_500k,1M__MGI_100 , 150__G1_15__G2_15__G3_10__501 sim_time: 43200.1000
created: 347 started: 9883 relayed: 6354 aborted: 3529
dropped: 6443 removed: 0 delivered: 113 delivery_prob: 0.3256
response_prob: 0.0000 overhead_ratio: 55.2301 latency_avg: 2254.0796
latency_med: 1851.2000 hopcount_avg: 3.3540 hopcount_med: 3 buffertime_avg: 903.5796 buffertime_med: 687.0000 rtt_avg: NaN rtt_med: NaN

Figure B.2: ONE simulator Message Stats Report file

..._EpidemicRouter__#of Nodes_0,40__BS_3M

[43200]
P0 489.13
P1 468.35
P2 503.07
P3 469.15
P4 460.53
P5 518.53
P6 487.19
P7 505.11
P8 502.93
P9 529.75
P10 506.61
P11 463.41
P12 512.83
P13 494.97
P14 481.55
C15 430.07
C16 417.23

..._EpidemicRouter__#of Nodes_0,40__BS_3M

P13 494.97
P14 481.55
C15 430.07
C16 417.23
C17 431.69
C18 427.33
C19 433.79
C20 424.69
C21 447.33
C22 438.99
C23 466.41
C24 425.27
C25 434.69
C26 418.39
C27 423.19
C28 442.89
C29 434.51
V30 395.15

..._EpidemicRouter__#of Nodes_0,40__BS_3M

C22 438.99
C23 466.41
C24 425.27
C25 434.69
C26 418.39
C27 423.19
C28 442.89
C29 434.51
V30 395.15
V31 394.73
V32 403.45
V33 415.21
V34 403.81
V35 414.15
V36 400.11
V37 428.51
V38 403.83
V39 399.27

Figure B.3: ONE simulator Energy Level Report file

Appendix C: Simulation Results Tables

C.1 The impact of number of nodes

Table C.1: Impact of number of nodes on node’s average remaining energy using buffer size as 3MB

Node’s Average Remaining Energy (Units)	Protocols	Number of Nodes				
		40	80	120	160	200
	Epidemic	252.47	245.27	92.27	23.19	3.03
	SaW	553.76	574.29	584.61	583.71	575.39
	PRoPHET	482.77	278.49	106.17	27.74	4.2
	MaxProp	435.49	205.32	53.38	21.45	9.01

Table C.2: Impact of number of nodes on node’s average remaining energy using buffer size as 9 MB

Node’s Average Remaining Energy (Units)	Protocols	Number of Nodes				
		40	80	120	160	200
	Epidemic	323.11	37.27	0.27	0	0
	SaW	532.74	571.79	583.50	583.39	575.38
	PRoPHET	380.97	73.01	1.30	0	0
	MaxProp	385.47	292.51	226.74	205.24	190.04

Table C.3: Impact of number of nodes on number of dead nodes using buffer size as 3 MB

Number of Dead Nodes	Protocols	Number of Nodes				
		40	80	120	160	200
	Epidemic	0	0	38	111	187
	SaW	0	0	0	0	0
	PRoPHET	0	0	35	109	184
	MaxProp	0	0	60	114	169

Table C.4: Impact of number of nodes on number of dead nodes using buffer size as 9 MB

Number of Dead Nodes	Protocols	Number of Nodes				
		40	**80**	**120**	**160**	**200**
	Epidemic	0	37	119	160	200
	SaW	0	0	0	0	0
	PRoPHET	0	21	117	160	200
	MaxProp	0	0	17	40	50

Table C.5: Impact of number of nodes on delivery ratio using buffer size as 3 MB

Delivery Ratio	Protocols	Number of Nodes				
		40	**80**	**120**	**160**	**200**
	Epidemic	0.287	0.235	0.201	0.187	0.187
	SaW	0.541	0.737	0.776	0.853	0.866
	PRoPHET	0.343	0.304	0.263	0.248	0.231
	MaxProp	0.639	0.849	0.924	0.926	0.909

Table C.6: Impact of number of nodes on delivery ratio using buffer size as 9 MB

Delivery Ratio	Protocols	Number of Nodes				
		40	**80**	**120**	**160**	**200**
	Epidemic	0.51	0.479	0.418	0.382	0.382
	SaW	0.821	0.894	0.909	0.923	0.915
	PRoPHET	0.54	0.477	0.408	0.368	0.345
	MaxProp	0.911	0.96	0.971	0.971	0.96

Table C.7: Impact of number of nodes on overhead ratio using buffer size as 3 MB

Overhead Ratio	Protocols	Number of Nodes				
		40	80	120	160	200
	Epidemic	62.38	317.89	231.62	1462.07	2202.96
	SaW	13.02	11.61	11.17	10.36	10.23
	PRoPHET	40.56	196.58	540.98	951.92	1492.28
	MaxProp	27.25	75.14	143.67	214.90	293.27

Table C.8: Impact of number of nodes on average latency (ms) using buffer size as 3 MB

Average Latency (ms)	Protocols	Number of Nodes				
		40	80	120	160	200
	Epidemic	2539.6	2093.2	1768.8	1708.0	1490.0
	SaW	1779.8	1758.0	1795.9	1873.9	1952.0
	PRoPHET	2864.8	2197.2	2022.2	1984.4	1824.0
	MaxProp	4103.9	2958.0	2064.8	1505.7	1262.3

C.2 The Impact of message size

Table C.9: Impact of message size in bytes on node's average remaining energy using buffer size as 3 MB

Node's Average Remaining Energy (Units)	Protocols	Message Size (MB)				
		0.05 – 0.25	0.25 – 0.5	0.5 – 0.75	0.75 – 1	1 – 1.5
	Epidemic	222.402	222.511	232.436	252.339	300.789
	SaW	642.572	622.956	591.833	557.782	521.878
	PRoPHET	277.533	256.676	267.834	283.925	319.911
	MaxProp	587.141	428.691	227.485	188.171	218.776

Table C.10: Impact of message size in bytes on node's average remaining energy using buffer size as 9 MB

Node's Average Remaining Energy (Units)	Protocols	Message Size (MB)				
		0.1 – 0.5	**0.5 – 1**	**1 – 1.5**	**1.5 – 2**	**2 – 2.5**
	Epidemic	50.095	37.275	37.339	34.636	38.475
	SaW	629.27	571.793	493.118	420.048	364.821
	PRoPHET	107.921	76.364	79.826	87.199	105.26
	MaxProp	510.27	297.05	46.858	30.893	36.128

Table C.11: Impact of message size in bytes on number of dead nodes using buffer size as 9 MB

Number of Dead Nodes	Protocols	Message Size (MB)				
		0.1 – 0.5	**0.5 – 1**	**1 – 1.5**	**1.5 – 2**	**2 – 2.5**
	Epidemic	34	37	36	38	36
	SaW	0	0	0	0	0
	PRoPHET	12	21	19	16	10
	MaxProp	0	0	31	44	40

Table C.12: Impact of message size in bytes on delivery ratio using buffer size as 3 MB

Delivery Ratio	Protocols	Message Size (MB)				
		0.05 – 0.25	**0.25 – 0.5**	**0.5 – 0.75**	**0.75 – 1**	**1 – 1.5**
	Epidemic	0.5046	0.3636	0.2762	0.2329	0.2329
	SaW	0.9422	0.9077	0.7931	0.6722	0.474
	PRoPHET	05054	0.3958	0.3293	0.2788	0.2137
	MaxProp	0.9739	0.9712	0.9401	0.7628	0.4385

Table C.13: Impact of message size in bytes on delivery ratio using buffer size as 9 MB

Delivery Ratio	Protocols	Message Size (MB)				
		0.1 – 0.5	**0.5 – 1**	**1 – 1.5**	**1.5 – 2**	**2 – 2.5**
	Epidemic	0.6366	0.4788	0.3681	0.2975	0.2975
	SaW	0.9425	0.8949	0.7517	0.614	0.4735
	PRoPHET	0.6217	0.4672	0.385	0.3218	0.2393
	MaxProp	0.9755	0.9599	0.8756	0.6453	0.4564

Table C.14: Impact of message size in bytes on average latency (ms) using buffer size as 3 MB

Average Latency (ms)	Protocols	Message Size (MB)				
		0.05 – 0.25	**0.25 – 0.5**	**0.5 – 0.75**	**0.75 – 1**	**1 – 1.5**
	Epidemic	1842.6	1918.07	2055.61	2360.55	2139.68
	SaW	1517.01	1427.96	1769.37	1722.61	1646.79
	PRoPHET	1838.83	2029.51	2045.37	2556.62	2641.14
	MaxProp	1046.94	1168.55	2252.19	3395.13	3776.44

Table C.15: Impact of message size in bytes on overhead ratio using buffer size as 3 MB

Overhead Ratio	Protocols	Message Size (MB)				
		0.05 – 0.25	**0.25 – 0.5**	**0.5 – 0.75**	**0.75 – 1**	**1 – 1.5**
	Epidemic	912.074	469.208	300.949	204.255	175.059
	SaW	9.245	9.565	10.904	12.576	15.615
	PRoPHET	714.761	357.925	227.852	158.062	121.543
	MaxProp	62.421	137.22	80.066	71.651	75.006

C.3 The Impact of message generation interval

Table C.16: Impact of message generation interval on node’s average remaining energy using buffer size as 3 MB

Node’s Average Remaining Energy (Units)	Protocols	Message Generation Interval (s)								
		0-10	10-20	20-30	30-40	40-50	100-150	150-250	250-350	550-650
	Epidemic	127.36	165.65	177.18	183.01	194.33	217.83	241.36	269.74	324.48
	SaW	168.14	296.76	366.78	416.89	451.3	553.43	583.74	600.95	617.45
	PRoPHET	155.11	190.37	199.12	212	222.4	256.56	271.16	298.27	363.33
	MaxProp	100.81	124.15	129.96	134.66	140.24	186.02	284.68	402.18	537.41

Table C.17: Impact of message generation interval on node’s average remaining energy using buffer size as 9 MB

Node’s Average Remaining Energy (Units)	Protocols	Message Generation Interval (s)				
		0-10	40-50	100-150	250-350	550-650
	Epidemic	12.558	23.214	35.541	53.437	118.562
	SaW	31.190	448.443	572.03	618.877	636.298
	PRoPHET	21.481	49.344	71.219	123.947	242.141
	MaxProp	10.481	24.755	292.297	472.884	560.921

Table C.18: Impact of message generation interval on number of dead nodes using buffer size as 3 MB

Number of Dead Nodes	Protocols	Message Generation Interval (s)								
		0-10	10-20	20-30	30-40	40-50	100-150	150-250	250-350	550-650
	Epidemic	5	4	2	2	2	0	0	0	0
	SaW	2	0	0	0	0	0	0	0	0
	PRoPHET	1	0	0	0	0	0	0	0	0
	MaxProp	4	1	1	1	0	0	0	0	0

Table C.19: Impact of message generation interval on number of dead nodes using buffer size as 9 MB

	Protocols	Message Generation Interval (s)				
		0-10	**40-50**	**100-150**	**250-350**	**550-650**
Number of Dead Nodes	Epidemic	59	47	38	29	5
	SaW	40	0	0	0	0
	PRoPHET	50	32	20	8	0
	MaxProp	61	46	0	0	0

Table C.20: Impact of message generation interval on delivery ratio using buffer size as 3 MB

	Protocols	Message Generation Interval (s)				
		0-10	**40-50**	**100-150**	**250-350**	**550-650**
Delivery Ratio	Epidemic	0.063	0.177	0.228	0.298	0.298
	SaW	0.072	0.444	0.752	0.862	0.935
	PRoPHET	0.075	0.223	0.298	0.376	0.403
	MaxProp	0.074	0.484	0.872	0.969	0.973

Table C.21: Impact of message generation interval on delivery ratio using buffer size as 9 MB

	Protocols	Message Generation Interval (s)				
		0-10	**40-50**	**100-150**	**250-350**	**550-650**
Delivery Ratio	Epidemic	0.160	0.355	0.476	0.597	0.597
	SaW	0.199	0.759	0.895	0.899	0.931
	PRoPHET	0.186	0.367	0.467	0.569	0.699
	MaxProp	0.206	0.877	0.964	0.958	0.97

Table C.22: Impact of message generation interval on overhead ratio using buffer size as 3 MB

Overhead Ratio	Protocols	Message Generation Interval (s)				
		0-10	**40-50**	**100-150**	**250-350**	**550-650**
	Epidemic	45.734	161.460	335.321	513.693	652.003
	SaW	33.654	17.514	11.421	10.081	9.314
	PRoPHET	34.552	103.804	203.131	338.672	477.355
	MaxProp	39.192	53.017	72.587	75.029	59.700

Table C.23: Impact of message generation interval on average latency (ms) using buffer size as 3 MB

Average Latency	Protocols	Message Generation Interval (s)				
		0-10	**40-50**	**100-150**	**250-350**	**550-650**
	Epidemic	677.64	1820.236	1951.423	2015.808	2181.462
	SaW	462.025	1186.439	1739.795	2222.917	2397.545
	PRoPHET	738.801	2244.684	2367.145	2385.218	2327.549
	MaxProp	2535.464	3604.068	2789.711	1598.65	1291.873

C.4 The Impact of node's speed

Table C.24: Impact of node's speed on node's average remaining energy using buffer size as 3 MB

Node's Average Remaining Energy (Units)	Protocols	Node's speed (m/s)				
		0-2.5	**2.5-5**	**5-7.5**	**7.5-10**	**10-12.5**
	Epidemic	272.234	104.713	56.622	49.127	41.447
	SaW	570.827	530.639	486.443	442.036	419.319
	PRoPHET	326.174	179.077	117.819	84.765	68.335
	MaxProp	266.707	145.146	72.898	53.235	48.295

Table C.25: Impact of node's speed on node's average remaining energy using buffer size as 9 MB

Node's Average Remaining Energy (Units)	Protocols	Node's speed (m/s)				
		0-2.5	2.5-5	5-7.5	7.5-10	10-12.5
	Epidemic	146.823	20.187	18.616	14.387	10.760
	SaW	589.338	560.383	516.687	456.92	455.158
	PRoPHET	217.488	34.349	28.825	26.519	24.779
	MaxProp	321.962	256.068	213.983	165.387	159.417

Table C.26: Impact of node's speed on number of dead nodes using buffer size as 3 MB

Number of Dead Nodes	Protocols	Node's speed (m/s)				
		0-2.5	2.5-5	5-7.5	7.5-10	10-12.5
	Epidemic	3	17	42	58	60
	SaW	0	0	0	0	0
	PRoPHET	0	0	4	24	44
	MaxProp	2	4	30	58	60

Table C.27: Impact of node's speed on number of dead nodes using buffer size as 9 MB

Number of Dead Nodes	Protocols	Node's speed (m/s)				
		0-2.5	2.5-5	5-7.5	7.5-10	10-12.5
	Epidemic	32	61	61	61	64
	SaW	0	0	0	0	0
	PRoPHET	30	60	60	60	60
	MaxProp	0	0	0	0	0

Table C.28: Impact of node’s speed on delivery ratio using buffer size as 3 MB

	Protocols	Node’s speed (m/s)				
		0-2.5	**2.5-5**	**5-7.5**	**7.5-10**	**10-12.5**
Delivery Ratio	Epidemic	0.221	0.241	0.249	0.199	0.199
	SaW	0.61	0.752	0.754	0.659	0.644
	PRoPHET	0.289	0.27	0.265	0.211	0.246
	MaxProp	0.71	0.943	0.844	0.714	0.691

Table C.29: Impact of node’s speed on delivery ratio using buffer size as 9 MB

	Protocols	Node’s speed (m/s)				
		0-2.5	**2.5-5**	**5-7.5**	**7.5-10**	**10-12.5**
Delivery Ratio	Epidemic	0.418	0.421	0.381	0.338	0.338
	SaW	0.765	0.956	0.928	0.799	0.754
	PRoPHET	0.463	0.437	0.380	0.343	0.339
	MaxProp	0.939	0.98	0.972	0.955	0.951

Table C.30: Impact of node’s speed on average latency (ms) using buffer size as 3 MB

	Protocols	Node’s speed (m/s)				
		0-2.5	**2.5-5**	**5-7.5**	**7.5-10**	**10-12.5**
Average Latency (ms)	Epidemic	2358.804	1199.537	1531.604	1347.625	2254.459
	SaW	1809.358	1339.796	1358.291	1440.463	1461.767
	PRoPHET	3260.996	1456.218	1605.154	1505.287	2130.375
	MaxProp	3429.900	1921.046	2545.954	2946.521	3006.772

Table C.31: Impact of node's speed on overhead ratio using buffer size as 3 MB

Overhead Ratio	Protocols	Node's speed (m/s)				
		0-2.5	**2.5-5**	**5-7.5**	**7.5-10**	**10-12.5**
	Epidemic	277.271	368.795	311.629	411.809	287.379
	SaW	13.526	11.425	11.618	12.630	12.896
	PRoPHET	146.593	262.58	238.497	319.73	223.603
	MaxProp	63.698	47.758	53.215	64.091	70.79

C.5 Impact of Time to Live

Table C.32: Impact of TTL on node's average remaining energy using 80 nodes and buffer size as 3 MB

Node's Average Remaining Energy (Units)	Protocols	TTL (h)				
		1	**3**	**5**	**7**	**9**
	Epidemic	297.135	250.559	244.426	239.07	238.061
	SaW	575.150	574.258	574.939	575.138	575.193
	PRoPHET	408.112	296.856	275.542	268.075	268.159
	MaxProp	221.897	214.834	204.747	205.16	208.233

Table C.33: Impact of TTL on node's average remaining energy using 160 nodes and buffer size as 3 MB

Node's Average Remaining Energy (units)	Protocols	TTL (h)				
		1	**3**	**5**	**7**	**9**
	Epidemic	89.604	36.488	21.77	22.873	18.984
	SaW	584.494	584.125	584.064	584.083	583.873
	PRoPHET	169.857	41.313	28.431	24.567	23.038
	MaxProp	30.571	17.646	14.703	19.873	18.59

Table C.34: Impact of TTL on node's average remaining energy using 80 nodes and buffer size as 9 MB

Node's Average Remaining Energy (Units)	Protocols	TTL (h)				
		1	3	5	7	9
	Epidemic	124.852	35.342	32.227	34.695	36.079
	SaW	573.460	571.694	571.721	572.928	572.027
	PRoPHET	386.702	107.107	76.324	69.832	65.603
	MaxProp	301.316	287.871	297.021	289.718	295.571

Table C.35: Impact of TTL on number of dead nodes using 80 nodes and buffer size as 9 MB

Number of Dead Nodes	Protocols	TTL (h)				
		1	3	5	7	9
	Epidemic	3	37	41	39	39
	SaW	0	0	0	0	0
	PRoPHET	0	11	19	23	24
	MaxProp	0	0	0	0	0

Table C.36: Impact of TTL on number of dead nodes using 160 nodes and buffer size as 3 MB

Number of Dead Nodes	Protocols	TTL (h)				
		1	3	5	7	9
	Epidemic	56	101	112	113	116
	SaW	0	0	0	0	0
	PRoPHET	40	99	108	113	113
	MaxProp	105	118	123	118	118

Table C.37: Impact of TTL on delivery ratio using 80 nodes and buffer size as 3 MB

Delivery Ratio	**Protocols**	**TTL (h)**				
		1	**3**	**5**	**7**	**9**
	Epidemic	0.439	0.273	0.242	0.227	0.227
	SaW	0.673	0.731	0.732	0.742	0.734
	PRoPHET	0.496	0.350	0.299	0.280	0.285
	MaxProp	0.769	0.829	0.854	0.833	0.851

Table C.38: Impact of TTL on delivery ratio using 80 nodes and buffer size as 9 MB

Delivery Ratio	**Protocols**	**TTL (h)**				
		1	3	5	7	9
	Epidemic	0.808	0.556	0.474	0.437	0.437
	SaW	0.734	0.873	0.896	0.892	0.896
	PRoPHET	0.736	0.549	0.476	0.434	0.428
	MaxProp	0.934	0.964	0.961	0.961	0.962

Table C.39: Impact of TTL on average latency (ms) using buffer size as 3 MB

Average Latency (ms)	**Protocols**	**TTL (h)**				
		1	**3**	**5**	**7**	**9**
	Epidemic	1711.02	2079.90	2024.45	1862.59	1964.99
	SaW	1449.80	1729.96	1716.93	1755.92	1795.56
	PRoPHET	1739.41	2260.26	2256.11	2263.89	2271.87
	MaxProp	1676.44	2500.53	2905.60	3067.22	3117.54

Table C.40: Impact of TTL on overhead ratio using buffer size as 3 MB

Overhead Ratio	Protocols	TTL (h)				
		1	3	5	7	9
	Epidemic	124.731	255.527	318.401	342.498	383.357
	SaW	12.722	11.724	11.689	11.486	11.632
	PRoPHET	69.778	160.454	203.586	229.958	408.668
	MaxProp	76.471	72.583	74.304	76.518	73.943

C.6 Impact of Buffer Size

Table C.41: Impact of buffer size on node's average remaining energy using 80 nodes

Node's Average Remaining Energy (Units)	Protocols	Buffer Size (MB)				
		3	5	7	9	11
	Epidemic	242.732	109.205	52.804	32.877	24.215
	SaW	574.608	572.416	572.396	572.544	572.596
	PRoPHET	275.518	161.470	102.503	73.123	60.801
	MaxProp	201.918	207.175	258.722	288.401	300.06

Table C.42: Impact of buffer size on node's average remaining energy using 160 nodes

Node's Average Remaining Energy (Units)	Protocols	Buffer Size (MB)				
		3	5	7	9	11
	Epidemic	25.315	0	0	0	0
	SaW	584.110	583.820	583.516	583.644	583.648
	PRoPHET	28.230	0	0	0	0
	MaxProp	19.480	82.690	194.600	205.608	203.839

Table C.43: Impact of buffer size on number of dead nodes using 80 nodes

Number of Dead Nodes	Protocols	Buffer Size (MB)				
		3	**5**	**7**	**9**	**11**
	Epidemic	0	4	25	41	47
	SaW	0	0	0	0	0
	PRoPHET	0	1	9	22	29
	MaxProp	0	0	0	0	0

Table C.44: Impact of buffer size on number of dead nodes using 160 nodes

Number of Dead Nodes	Protocols	Buffer Size (MB)				
		3	**5**	**7**	**9**	**11**
	Epidemic	111	160	160	160	160
	SaW	0	0	0	0	0
	PRoPHET	110	158	160	160	160
	MaxProp	118	58	40	40	40

Table C.45: Impact of buffer size on delivery ratio

Delivery Ratio	Protocols	Buffer Size (MB)				
		3	**5**	**7**	**9**	**11**
	Epidemic	0.232	0.344	0.424	0.469	0.469
	SaW	0.737	0.83	0.868	0.894	0.899
	PRoPHET	0.295	0.396	0.429	0.465	0.503
	MaxProp	0.848	0.954	0.96	0.958	0.962

Table C.46: Impact of buffer size on average latency (ms)

Average Latency (ms)	Protocols	Buffer Size (MB)				
		3	5	7	9	11
	Epidemic	2019.65	2370.55	2700.71	2893.29	3059.90
	SaW	1763.64	1992.50	1857.94	1732.65	1774.98
	PRoPHET	2233.08	2403.31	2531.94	2602.31	2632.84
	MaxProp	2993.52	1862.44	1621.16	1621.17	1589.33

Table C.47: Impact of buffer size on overhead ratio

Overhead Ratio	Protocols	Buffer Size (MB)				
		3	5	7	9	11
	Epidemic	329.002	292.603	250.672	240.084	231.675
	SaW	11.623	10.426	9.951	9.630	9.585
	PRoPHET	208.749	200.755	206.610	203.700	192.761
	MaxProp	75.678	66.554	56.713	51.798	49.412

Appendix D: Survey of DTN Researches

Table D. 1: Survey of DTN Researches

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	**Value(s)**	**Metrics**	**Results**
SGBR: A Routing Protocol for Delay Tolerant Networks Using Social Grouping (2013)	• SGBR (proposed) • OPT • MAXPROP • SNW • PROPHET • EPIDEMIC	The ONE simulator	Hosts	10	Delivery Ratio	Five well known protocols are compared with the proposed protocol **SGBR** in the table. Three performance metrics are measured by varying the buffer size and packet TTL.
			Speed	0.5 – 1.5		
			Buffer Size	2 – 10 MB		
			Packet Interval	250 – 350	Average Delay	
			Movement	SPMBM		
			Packet TTL	2 – 10 hours	Delivery Cost	
			Transmission Speed	5 MB/s		
			Simulation Time	12 hours		
A Simulation Based Study of Well Known Routing Protocols for Delay Tolerant Network (2013)	• EPIDEMIC • PROPHET • SPRAY AND WAIT	The ONE simulator	Simulation area	4500 x 3400 m	Delivery Ratio	Three protocols are compared in graphs that show the protocols performance with three metrics by varying the number of nodes and the transmission rang.
			Time	43 K Sec		
			Number of nodes	50, 100, 125, 200	Overhead Ratio	
			Mobility Models	SPMBM		
			Transmission rang (Meter)	10, 50, 100	Average Latency	
			Buffer size	5 MB		

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	Value(s)	Metrics	Results
A Hop Count Based Heuristic Routing Protocol for Mobile Delay Tolerant Networks (2014)	• HCH • SPRAYAND WAIT • EPIDEMIC • PROPHET	The ONE simulator	Number of nodes	120	Delay Rate	Four protocols are compared in a graphs that demonstrate three performance metrics by varying the buffer size and packet TTL.
			World size (Meter)	4500 x 3000		
			Tickets for S &W	13		
			Message TTL (Seconds)	200 - 500		
			Simulation time (hours)	12	Average Latency	
			Message size (KB)	500 - 1024		
			Pedestrian buffer (MB)	15 - 55		
			Tram buffer (MB)	500		
			Bluetooth range (m)	10		
			High-speed range (m)	1000	Overhead Ratio	
			Bluetooth bandwidth (KBps)	250		
			High-speed bandwidth	10		
			Pedestrian speed (m/s)	0.5 - 1.5		
			Message interval (s)	35 - 45		
Probability Routing Algorithm Based On Historical Throughput In DTN Network (2013)	• HT-PROPHET (proposed) • PROPHET • EPIDEMIC	The ONE simulator	Nodes	60 (pedestrian) (car)	Delay Rate	The proposed protocol **HT-PROPHET** compared with tow protocols in the graphs that show the performance during the time of simulation.
			Node's Speed	0.5-1.5 (pedestrian) 1-16.7(car)	Average Latency	
			Transmission range	10m (pedestrian) m (car)	Overhead Ratio	

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	Value(s)	Metrics	Results
A MULTI-SCHEME ADAPTIVE ROUTING ALGORTHM BASED ON SPRAY AND WAIT FOR DALAY TOLERANT NETWORKS (2015)	• ASW (Proposed) • SPRAYAND WAIT • PROPHET • EPIDEMIC	The ONE simulator	Number of nodes	50	Delay Rate	The proposed protocol **ASW** compared with three other in graphs that demonstrate the performance evaluation by varying three parameters: message interval, buffer size, and message TTL.
			World size	1000(m^2)		
			Initial Tickets' Number	18		
			Initial Topology	Uniform	Average Latency	
			Transmit Range	20		
			Message Size	500 KB		
			Message Interval	20 - 60		
			Transmit Speed	200	Overhead Ratio	
			Moving Speed	0.5 – 1.5 m/s		
			Node Buffer Size	2 – 10 MB		
			Message TTL	60 – 300 Minutes	Hop count	
			Simulation Time	12 hours		
			Mobility Model	Random Waypoint		
			a	0.5		
Simulation of Epidemic, Spray and Wait and First Contact Routing Protocols in Delay Tolerant Network *Conference*	• EPIDEMIC • SPRAY AND WAIT • FIRST CONTACT	The ONE simulator	Simulation Area	4500 x 3400 m	Delay Probability	Three DTN protocols are compared in the graphs that show the impact of the varying of buffer size.
			Simulation Time	20000 s		
			Mobility Model	SPMBM	Average Delay	
			Transmission Range	10 m		
			TTL	300 s		
			Buffer Size	5, 10, 15, 20, 25	Overhead Ratio	
			Warm Up Period	1000 Seconds		

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	**Value(s)**	**Metrics**	**Results**
DirMove: direction of movement based routing in DTN architecture for post-disaster scenario (2015)	● DIR MOVE (Proposed) ● EPIDEMIC ● MAXPROP ● PROPHET ● SPRAYAND WAIT ● RAPID ● EBR ●	The ONE simulator	simulation time	6 hours	Delay Probability	The proposed protocol **Dir Move** compared with six other protocols in graphs that show the performance evaluation by varying the movement model and the number of nodes.
			Number of nodes	30, 60, 100		
			Node buffer size	50 MB – 1GB		
			Interface transmit range	(Wi-Fi) 50m, (Bt) 10m	Overhead Ratio	
			Interface transmit speed (B/s)	1M (Wi=Fi), 250 K (Bt)		
			Node Movement speed	- Fixed destination - MapBAsed - RWPM		
			Message creation rate	25 – 35 Seconds	Average Latency	
			Message size	500 KB – 2MB	No of Packet Drops	
			Movement model	0.5–1.5 m/s for pedestrians, 9–10 m/s for Police Cars, 11–13 m/s for ambulance		
Performance of Efficient Routing Protocol in Delay Tolerant Network: A Comparative Survey (2014)	● EPIDEMIC ● PROPHET ● SPRAY AND WAIT	Not Mentioned	Not Mentioned		They introduced delay tolerant network with their features such as intermittent connectivity, resource limitation and high delay. They also introduced the open routing issues in Delay Tolerant Network's security. The existing routing protocols in DTNs are classified to their strategies for controlling message copies and making forwarding decision.	

Source	Protocol	Simulation	Simulation settings			performance	
			Parameter	Value(s)		Metrics	Results
A Performance Comparison of Delay-Tolerant Network Routing Protocols (2016)	• EPIDEMIC • PROPHET • MAXPROP • SPRAY AND WAIT	The ONE simulator		Pedestrian	vehicles	Delivery Ratio	Four of DTN protocols are compared in a table and graphs that show the impact of varying of four parameters (No of nodes, speed, TTL, Packet inter-arrival time) on the protocols performance.
			Hosts	5,20,30, 45,60	5,10,15, 25,30		
			Speed m/s	0.5-1.5	2.7-13.9		
			Movement	SPMBM		Delivery Cost	
			Buffer size	2 MB – 10 MB			
			TTL	2 – 10 hours			
			Packet inter-arrival time	10, 30, 60, 300, 600 sec		Average Delay	
			Transmission speed				
			Simulation time				
A Probabilistic Misbehavior Detection Scheme toward Efficient Trust Establishment in Delay Tolerant Networks (2014)	• iTrust	The ONE simulator	Simulation area	2880 m^2			The proposed a probabilistic misbehavior detection scheme (iTrust) could reduce the detection overhead.
			Time Interval	10800 Sec			
			Number of nodes	50, 80, 100			
			Rounds	100			

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	Value(s)	Metrics	Results
Performance Evaluation of Delay Tolerant Network Routing Protocols (2015)	• EPIDEMIC • MAXPROP • PROPHET • SPRAY AND WAIT • DD • FF	The ONE simulator	Simulation Time(min)	720	Delivery Probability	Six of DTN routing protocols are compared in the graphs that show the impact of varying the movement model on the performance.
			Number of Nodes	50		
			Transmit Range(m)	250		
			Transmit speed (Mbps)	2	Average Latency	
			Node Speed (km/h)	10 - 50		
			TTL (Minutes)	120		
			Buffer size	Infinite	Overhead Ratio	
			Movement Model	-RWM -MBM -SPMBM -RW		
On Social Delay-Tolerant Networking: Aggregation, Tie Detection, and Routing (2014)	• CSAR (Proposed) • EPIDEMIC • PepoleRank • BubbleRAp •	The ONE simulator	Duration (Days)	0.125, 3, 19, 79, 246	Delivery Ratio	The proposed protocol **CSAR has** compared with three protocols in the graphs that show the performance of three matrices.
			Network Type	Bluetooth		
			Devices	iMote, Phone, Pmtr, T-mote	Overhead Ratio	
			Granularity (Seconds)	6.6,15, 120, 300	Delivery Delay	
			No of Nodes	27, 44, 62, 97, 98		

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	Value(s)	Metrics	Results
Energy Consumption Ana of Delay Tolerant Networ Routing Protocols (2014)	• EPIDEMIC • SPRAY AND WAIT • PROPHET • MAXPROP •	The ONE simulator	Simulation Area	4500 x 3400 m	Node's Average Remaining Energy	Four protocols are compared in a graphs that demonstrate two performance metrics of energy consumption.
			Interface	Bluetooth		
			Interface Data Rate	2 Mbps		
			Radio Range	10 m		
			Simulation Time	12 Hours		
			Initial Energy		Number of Unavailable Nodes	
			Scan Energy			
			Transmit Energy			
			Scan Response Energy			
			Base Energy			
SEBAR: Social-Energy-Based Routing for Mobile Social Delay-Tolerant Networks (2017)	• SEBAR • SEBAR-AU • EPIDEMIC • SPRAY AND WAIT • Greedy Total • Bupple Rap	MIT reality mining project	Time Window	165600 sec	Delivery Ratio	Six of DTN protocols are compared in the graphs that shows the performance of four matrices
			No of Nodes	78		
			K value of k-clique	4	Average hops	
			Threshold w of k	4500 s		
					Delay	
			No of communications	3		
					Number of forwarding	
			No of replication	5		

Source	Protocol	Simulation	Simulation settings		performance	
			Parameter	Value(s)	Metrics	Results
Performance Comparison of RAPID, Epidemic and Prophet Routing Protocols for Delay Tolerant Networks (2012)	• RAPID • PROPHET • EPIDEMIC	The ONE simulator	Simulation Time	12 Hours	Delivery Ratio	Three DTN protocols are compared in the graphs that show the impact of the varying of TTL.
			World Size	4500*3400 m		
			Movement Model	SPMBM		
			Node Buffer Size	5 M	Average Latency	
			No of Nodes	126		
			Interface transmit speed	2 Mbps		
			Interface Transmit Range	10 m	Delivery Cost	
			TTL	60, 120, 180, 240, 300, 360		
			Node Movement Speed	0.5 – 1.5		
			Message Creation Rate	25 – 35 Sec		
			Message Size	500 KB – 1 MB		