EVALUATION AND PERFORMANCE CHARACTERISTICS OF TWO DTN ROUTING PROTOCOLS DURING A TRAIN ACCIDENT USING ONE

Anasuya Dutta

20594248

COMP4032

REPORT

Under guidance of

Dr Milena Radenkovic

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Department of Computer Science

December 14, 2023

Abstract

Over the past decade, there has been a notable surge in the occurrence of train accidents, stemming from various causes. To address this escalating issue, an innovative approach involves the integration of sensors capable of detecting objects, including humans and animals, through the use of electromagnetic waves. The intent is to provide timely awareness to train drivers or control systems, thereby potentially mitigating the impact of these accidents.

The goal of this study is – to assess and compare performances of two opportunistic routing protocols within the ONE simulator, a platform designed for realistic train accident scenarios. The proposed opportunistic routing protocols have been deployed and built to enhance communication within Vehicular Delay Transport Networks (VDTN). The focus is on evaluating their effectiveness in diverse and realistic settings, where nodes exhibit dynamic movements, and communication patterns are influenced by the urgency to alert the train system upon detecting objects in its path.

The evaluation process will entail a comprehensive analysis of two opportunistic routing protocols. The experiments will be meticulously designed, taking into account the intricacies of train accident scenarios, node mobility patterns, and communication dynamics. The heart of the project lies in the performance evaluation phase. The simulations will generate data on various criteria, such as latency, reliability, and energy efficiency, providing a holistic view of each protocol's capabilities. This information will be instrumental in determining which opportunistic routing protocol is most adept at meeting the specific communication demands within VDTN, especially in the context of alerting the train system to potential obstacles.

The concluding section of the report will present a thorough analysis of the experimental results, drawing insights into the strengths and weaknesses of each opportunistic routing protocol. By understanding their performance under realistic conditions, the project aims to contribute valuable knowledge to the field, guiding future implementations and improvements. Thus, this project serves as a comprehensive exploration into the realm of opportunistic routing protocols within VDTN, with a specific focus on their application in realistic train accident scenario.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

In recent years, the transportation sector has witnessed a surge in technological advancements aimed at enhancing safety measures and mitigating the impact of unforeseen events. Among these, railway systems play a pivotal role in the global transportation network, connecting regions and facilitating the movement of goods and people. However, the unfortunate occurrence of train accidents remains a persistent challenge, necessitating innovative solutions to understand, analyse, and ultimately prevent such incidents.

The global transportation landscape relies heavily on the efficiency and safety of railway systems, making it imperative to continually innovate and enhance the understanding of potential risks, especially in the context of train accidents. In recent years, the advent of advanced technologies has opened new avenues for comprehensive simulations, offering a unique opportunity to delve into the intricacies of these incidents. Our project is motivated by the critical need to leverage Vehicular Delay Transport Network (VDTN) technology to simulate and analyse train accidents, contributing to the development of effective preventive measures and improved safety protocols.

The occurrence of train accidents poses a significant challenge to the reliability and safety of railway systems. Traditional investigation methods often fall short in capturing the dynamic nature of these incidents, necessitating a paradigm shift towards advanced simulation techniques. VDTN, with its capacity to model vehicular interactions and delays within a network, emerges as a powerful tool to recreate realistic scenarios in a controlled virtual environment.

The primary motivation behind our project is to bridge the gap between theoretical understanding and practical application in the domain of train accident analysis. By employing VDTN, we aim to create a sophisticated simulation platform that accurately mirrors real-world conditions, allowing for the exploration of diverse accident scenarios and their associated variables. Moreover, the integration of VDTN in our simulation project aligns with the broader goal of advancing transportation safety. The ability to replicate and study train accidents in a virtual space not only facilitates a deeper comprehension of the contributing factors but also enables the formulation of data-driven solutions to enhance railway system resilience.

1.2 Scenario of the Project

There has been a noticeable increase in the number of train accidents over the last decade for various reasons. Most of us are aware of the head-on collision between the passenger and freight trains in central Greece has resulted in a significant loss of life, injuries, and has brought attention to the state of railway safety in the country. The collision involved a passenger train, carrying more than 350 people, and a freight train. The impact of the head-on collision resulted in a devastating loss of life, with at least 38 people confirmed dead, and numerous others sustaining injuries. Emergency response teams, including paramedics and firefighters, were deployed to the crash site to manage the aftermath and address the immediate needs of the victims. The incident has had a profound impact on the nation, prompting discussions about the need for comprehensive reforms in railway safety and infrastructure. The loss of lives and injuries has sparked public grief and demands for accountability and swift action to prevent such tragedies in the future.

Figure 1.2 : Train Accident in central Greece

Considering the above scenario, I think there is an immediate need of an awareness object that can detect objects with and make the train drivers aware of it so that they can take some actions. Thus, the scenario focuses on the communications between the trains or between the trains and pedestrians or so when there is an accident across Helsinki. The number of accidents may increase in areas where there is a lot of congestion and at the messages may not reach the destination. Here, we have considered the collided trains that are tc1 and tc2 as stationary nodes. The reason behind this is that once there is an accident, the trains should not move and communicate hereafter.

Two protocols Epidemic and Spray and Wait will be implemented as the routing protocol used by the trains for communication after detecting the objects. Therefore, various effects on the delivery probability, average latency and buffer availability will be compared between the two protocols just to assess how each responds differently to the situation and which was affected the least.

Here, in the scenario there are 6 node types- Pedestrians, Trains, Ambulances, Cars, Communication Towers and Control Centres where the collided trains (tc1) and (tc2) are the source nodes and control centres are the destination nodes. Messages are passed through all the other node types through high-speed network and sent to the control centre which monitor and assist the trains.

1.3 Structure of the project

This study will evaluate the performances of two DTN routing protocols using various metrics:

The first chapter consists of the motivation, an introduction to the routing technique in DTN and explanation of the scenario.

Chapter Two includes related works literature review of opportunistic networks and a brief introduction to the DTN simulator.

Chapter Three includes on the methodology, providing an overview, movement models, routing modules, events generation, report modules, simulation, and scenario

Chapter Four includes experiment aim and setup followed by evaluation and discussed results on various metrics for both the protocols

Chapter Five includes the pros and cons of various opportunistic networks in various scenarios

Chapter Six presents the conclusion and any suggestions for future work.

2 Literature Review

2.1 Introduction

There has been a huge increase in the number of train accidents due to various reasons in the past decade. These accidents could be reduced to a bit if we can make use of some sensors which can detect the any objects, be it human beings or animals, with the help of electromagnetic waves so as to make the train drivers or the control systems aware of the object. The aim of this project is to compare performances of multiple criteria of two opportunistic routing protocols within the ONE simulator for realistic train accidents scenarios that has been proposed, deployed and built. Here, I will compare multiple criteria simulation results in order to better understand what the most efficient protocol is for sending messages within VDTN, where simulations are basically configured with my scenario, diverse node movements, and communication patterns where the train must be made alerted as soon as there is an object. This report will provide a detailed description and analysis of opportunistic routing protocols, how and why they are used in our scenario. A description of my experiment will be provided, in addition to a full performance evaluation and an analysis of my findings.

2.2 Opportunistic Networks

Due to possible lack of connectivity between sources and destinations, opportunistic networks have complex topologies that can be modelled as complex temporal graphs with partitions. Because they operate without interacting with or utilising infrastructure, they minimise any overhead brought on by contact at the infrastructure level. As the cost of computing steadily decreases each year, there is a corresponding increase in the volume of data that new devices need to transfer over networks. This trend is leading to the integration of networks into our daily lives. However, not all networks are equally equipped to handle the diverse challenges posed by the wide range of communication-enabled devices available to consumers. The primary reason for this discrepancy lies in the network topology, which is influenced by the varying nature of different devices.[5]

For example, a typical desktop computer demands a steady and reliable stream of information. Protocols designed for such interactions may struggle in scenarios where data transmission is frequently disrupted, and the data's destination is not consistently within reach. To address these issues, numerous new protocols and systems have been developed to tackle common challenges in today's society.

Opportunistic networks are a common term for networks designed to handle such challenges. These networks facilitate intercommunication among devices even when a static route does not exist. Typically wireless, these systems require no infrastructure to operate, as each device in the network serves as a node for data transmission. Any device with the ability to communicate with neighbouring devices can function as a node, and the choice of how a node sends data depends on the selected protocol. These nodes should be self-assembling and self-organizing, minimizing the need for human intervention.
In the 21st century, many systems commonly require an end-to-end connection, and protocols like TCP/IP were developed with static connections in mind. However, such protocols may attempt to find the destination of a message only a few times before giving up and dropping the data. This approach is suitable for networks where a single message among thousands is not critical, and extensive wiring can be established for static machines. In contrast, in areas like battlefields or disaster zones, this approach is far from optimal. In such cases, traditional networks and MANETs may struggle with frequent delays and large network partitions, increasing the likelihood of important messages, potentially involving life or death scenarios, not reaching their destination.[8]

In opportunistic networks, nodes serve two primary functions. Firstly, they engage in node discovery, actively searching their proximity or communication range for other nodes available for communication. The second function involves message exchange, where two nodes can mutually forward messages to each other.

To accommodate disconnections and delays, Delay-Tolerant Networks (DTNs) adhere to the 'Store-Carry-Forward' approach where a node receives messages from the source or transmitting node and stores it within its buffer. The message remains in the buffer until a relay node comes within the communicating range. At that point, the node forwards the message to some other node. This paradigm is well-suited for unpredictable communication patterns, as the transmitting node lacks information about when or where the next contact with another node will occur. Additionally, this approach fosters dynamically formed routes, allowing any node to potentially serve as the next hop if it positively contributes to the message reaching its intended destination.[10]

2.3 MANETs

These are networks comprised of wireless mobile nodes, lacking a fixed infrastructure. To overcome the absence of infrastructure, messages traverse a series of nodes to reach their intended destination. MANETs consist of wireless, mobile nodes that communicate without relying on a standard network infrastructure. Instead of routing messages through a centralized structure, like a router or server, messages are directly transmitted to nearby nodes within the sender's transmission range. MANETs are designed to accommodate the unpredictable connections and disconnections inherent in mobile environments.

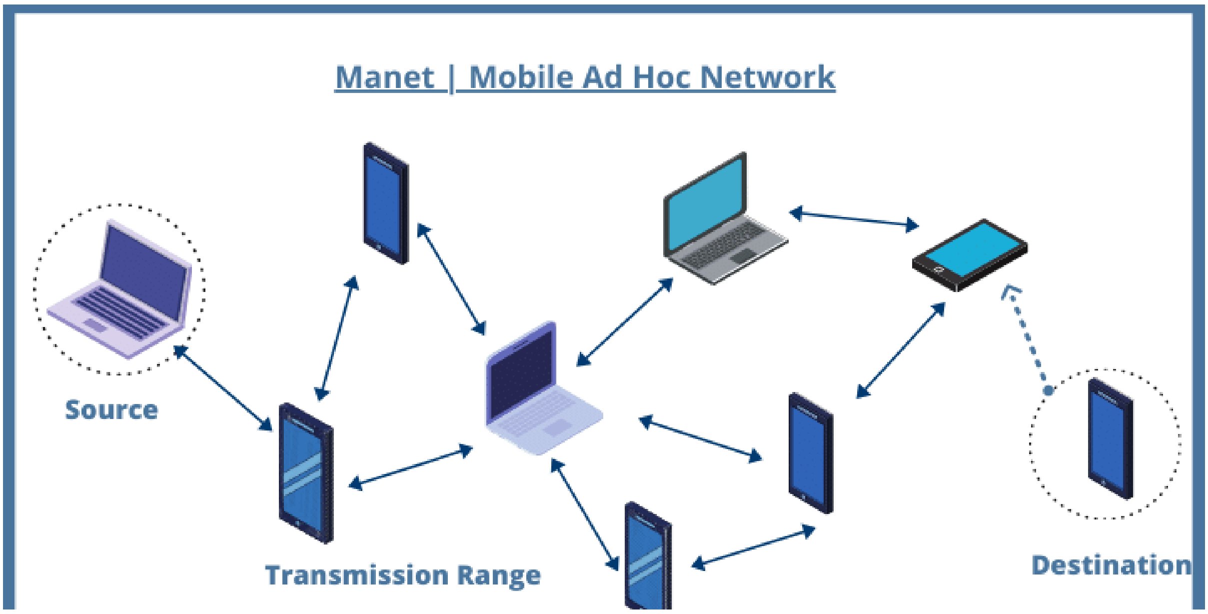

Figure 2.3: MANET

Distinguishing themselves from opportunistic networks, MANETs focus on establishing end-to-end routes between two nodes before initiating communication. Conventional routing protocols are ill-suited for MANET conditions, leading to the existence of various MANET-specific routing protocols, each with its unique approach. Proactive routing protocols maintain routing tables that store information on every node in the network, updating periodically to track changes. Analysing these tables helps determine the most efficient path for forwarding nodes.

Routing in MANETs typically involves discovering an end-to-end path before actual data transmission. In sparse networks with intermittent connections, sustaining complete paths for a sufficient duration becomes challenging, highlighting the necessity for Delay Tolerant Networks.

2.4 VANETs

Vehicular Ad-hoc[3] are a specialised type of Ad-Hoc Network (a decentralised kind of wireless network that operates independently of an existing infrastructure, such access points in managed (infrastructure) wireless networks or routers in wired networks) which are wireless and self-organised communication networks. They enable vehicle-to-vehicle and vehicle-to-infrastructure communications where the nodes are basically free-moving and are made up of vehicles and fixed infrastructure so ad to form a connection and deal with the constant change in topology due to the fact that these vehicles are often in motion. Thus, there is a need to find the optimal and most efficient algorithm to find the best route to pass data through the network since there is increase in the constant streams of information that has to be passed between vehicles for various safety reasons such as accident detection, collision avoidance, etc. Because

DTNs do not presume end-to-end connectivity or feedback, VANETs are more susceptible to congestion and disconnections than DTNs. As a result, VANETs must be expanded to include the DTN paradigm of communications.

2.5 DTN

Delay Tolerant Networks (DTNs) have become indispensable due to challenges arising from nodes' mobility and limited transmission range in a network, leading to unstable connections. To tackle these issues as we have seen in Section 2.1, Disruption Tolerant Networks (DTNs) were devised, implementing a store-carry-forward approach using routing protocols to determine message propagation without requiring a pre-established path.

The challenges in DTNs extend to congestion management, requiring various protocols to modify data packet paths through the network. Additionally, frequent communication among nodes is vital to providing real-time information about each node's reliability, enabling the identification of sudden surges and facilitating data rerouting to alleviate stress in specific areas.

Security is a paramount concern in DTNs, as nodes are required to share information about neighbouring nodes. The presence of malicious or faulty nodes can manipulate traffic or prevent data transfer, posing potential disasters, particularly if critical safety messages are compromised. Nodes causing such disruptions are commonly referred to as "black holes" due to their behaviour akin to galactic structures that devour light.

A drawback of this ad hoc network style is the absence of trusted nodes or infrastructure, heightening the challenge of identifying and mitigating these hazards. The dynamic and unpredictable nature of DTNs necessitates continuous efforts to address these issues, enhancing the network's overall robustness and reliability. Because DTNs do not presume end-to-end connectivity or feedback, VANETs are more susceptible to congestion and disconnections than DTNs. As a result, VANETs must be expanded to include the DTN paradigm of communications.

2.6 VDTN

Vehicular Delay Tolerant Networks are a part of family of self-organized, opportunistic and autonomous networks. Because DTN’s do not assume end to end connectivity which VANETSs does, it therefore led to the expansion of DTN’s for vehicular nodes (VDTN). Due to its mobile (and possibly disconnection prone) nature, the VDTN network lacks consistent network connectivity; instead of waiting for the source node to encounter the destination, messages are stored by nodes as data bundles and moved in hops throughout the network until they reach their destination. This raises the likelihood that a message will be delivered, and opportunistic protocols control which nodes get a message and to whom. VDTNs are more resistant to extended periods of congestion or isolation as they do not require end-to-end connection between the source and destination nodes.

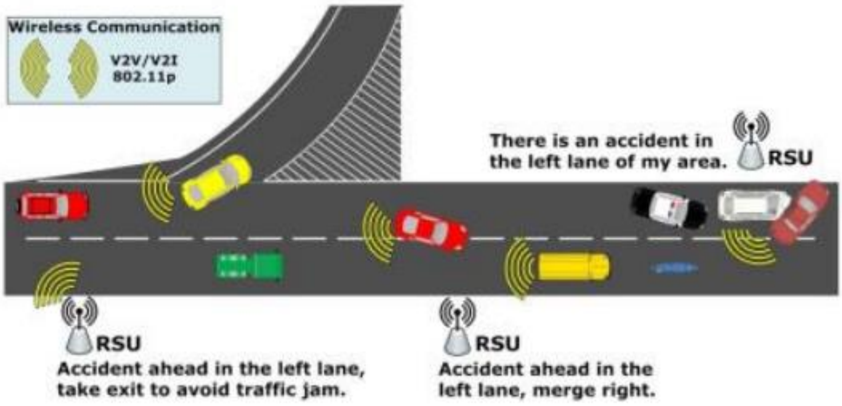

Figure 2.6: VANET example where other users are notified when car send packets

2.7 DTN Routing Protocols

2.7.1 Routing in DTN

Routing protocols [17] in DTNs fall into two categories based on their decision-making processes: flooding-based and forwarding-based protocols. Flooding-based protocols involve duplicating a message and spreading it across the network to boost delivery probability. Conversely, forwarding-based protocols leverage heuristics about nodes within the network to identify the most likely path to the destination swiftly, reducing resource demands.

The forwarding-based method entails a node acting as the custodian of a message until the destination node becomes visible or another node, capable of enhancing the probability of finding the destination node, is reached. While effective, this method comes at the cost of each node needing extra buffer space to hold messages before forwarding, introducing the risk of buffer overflow and packet drops.

DTNs normally follows store-carry forward approach which is a fundamental concept in Delay-Tolerant Networking (DTN) routing protocols. This paradigm is designed to address communication challenges in scenarios where traditional end-to-end paths are disrupted, intermittent, or non-existent. In the Store-Carry-Forward paradigm, nodes in the network play a crucial role in storing, carrying, and forwarding messages.

- **Store:** When a node receives a message from the source or transmitting node, it stores the message in its buffer or storage. The buffer serves as a temporary repository for messages, allowing nodes to retain information even in the absence of a continuous communication path.
- **Carry:** Nodes, equipped with stored messages in their buffers, move around within the network. The mobility of nodes is a key aspect of the paradigm. As nodes travel, they

carry the stored messages with them, creating opportunities for encounters with other nodes.

- **Forward**: When a node comes into contact with another node, the stored messages in its buffer can be forwarded to the neighboring node. This forwarding process occurs opportunistically whenever nodes are within communication range of each other. The messages are relayed from one node to another until they reach their intended destination.

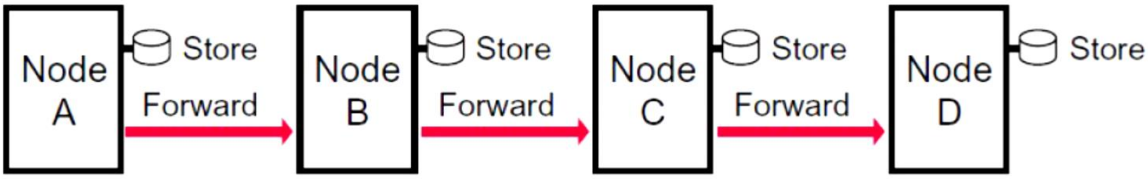

Figure 2.7.1: Store-carry forward paradigm

2.7.2 Epidemic Routing Protocol

The Epidemic protocol[6] is a replication-based protocol, one of the earliest and simplest for disseminating information. Operating on the concept of flooding, Epidemic floods the network with copies of messages to facilitate messaging. When two nodes form a connection, they exchange copies of messages from their buffers, specifically those that have nothing in common. This process repeats as subsequent nodes encounter each other, propagating message copies throughout the network.

Epidemic Routing supports eventual message delivery to arbitrary destinations without relying on minimal assumptions about the network's topology and connectivity. The flooding nature of messaging in Epidemics benefits from increased flow interest, resulting in lower delivery delays compared to other routing protocols. However, its effectiveness is contingent on minimal competition for shared network resources. Epidemic is a Delay Tolerant Network (DTN) protocol, employing the store, carry, and forward approach, focusing on message delivery rate achieved through replication.

One challenge with message delivery in Epidemic is the potential loss of a bundle of data sent by a node, as the destination node remains unaware if its received message has been lost. Unlike end-to-end systems with acknowledgment messages, Epidemic relies on the replication principle, increasing the probability of a message reaching its destination with more copies.

Replication events occur when nodes encounter each other in this mobile network, with nodes constantly moving within a limited radius. During these events, nodes compare messages and exchange copies, becoming custodians of the messages. While it offers a higher chance of success, it comes at the cost of significant resource wastage, causing congestion in the system.

The Epidemic protocol, proposed by Vahdat and Becker, floods the network with messages, aiming to increase the chances of messages reaching their destination. It operates by having encountering nodes exchange copies of messages from their buffers, resembling the spread of a disease epidemic. While ensuring high delivery probabilities, Epidemic does not consider nodes with a low chance of reaching the destination, leading to inefficient use of resources like buffer storage. Despite efforts to enhance efficiency, the protocol still faces challenges, primarily resource wastage and buffer congestion. In the following section, we will explore how changing buffer sizes affects the delivery ratio of messages, as buffer space significantly influences the performance of the Epidemic protocol.

Working principle:

The diagram below illustrates the entire process of message exchange within the Epidemic Routing protocol. When host A encounters host B, the two hosts establish a connection and initiate an anti-entropy session. In the initial step of the message exchange, host A sends its summation vector SV_A to host B. Subsequently, B performs a logical operation involving the negation vector Summation and selects a different set of message buffers along with locally buffered messages. B then transmits a vector to request these messages from A. Lastly, B sends the requested messages. This entire process is replicated when B establishes a connection with a new neighbour. With adequate buffer space and time, these anti-entropy sessions guarantee eventual message delivery through the pairwise exchange of information.

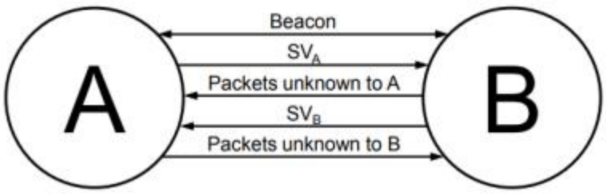

Figure 2.7.2 : Working principle of Epidemic

Epidemic Routing Protocol revolves around a replication-based approach for disseminating information in a network. Here's a breakdown of its working mechanism based on the provided information:

- **Replication-Based Protocol:** Epidemic Routing is a replication-based protocol, meaning it relies on creating multiple copies of messages to increase the chances of successful delivery.
- **Flooding Concept:** The protocol employs a flooding concept, intending to enable messaging by flooding the network with copies of messages. When a message needs to be disseminated, it is replicated and sent across the entire network.
- **Node Encounter and Message Exchange:** When two nodes encounter each other in the network, they compare the messages present in their respective buffers. Copies of messages are then exchanged between the nodes, specifically those messages that have nothing in common.
- **Propagation of Message Copies**: Subsequent encounters with nodes lead to the repetition of the comparison and exchange process. This causes copies of messages to be propagated throughout the entire network.
- **Eventual Delivery to Arbitrary Destinations:** This routing supports the eventual delivery of messages to arbitrary destinations without relying on minimal assumptions about the network's topology and connectivity. This flexibility allows messages to reach their intended destinations over time.
- **Increased Flow Interest:** The flooding nature of messaging in Epidemic benefits from increased flow interest, resulting in lower delivery delays compared to other routing

protocols. This is because the likelihood of delivering one of the copies of a message is higher than with other routing methods.

- **Store, Carry, and Forward Approach:** Epidemic is a Delay Tolerant Network (DTN) protocol that follows the store, carry, and forward approach. This means that nodes act as carriers of messages, storing and forwarding them as they encounter other nodes in the network.
- **Buffer Usage and Resource Management**: The protocol triggers replication events when nodes encounter each other. Nodes compare messages, exchange copies, and become custodians of the messages. However, this method leads to buffer space usage, and nodes may become "custodians" even when it is unlikely that they can efficiently pass on the data.
- **Limitations and Inefficiencies:** Despite its high delivery probabilities, Epidemic Routing has limitations, including resource wastage and inefficient usage of buffer space. Redundant message copies may persist in the network even after reaching their destination, contributing to inefficiencies.

Advantages and Disadvantages:

Advantages:

- High Delivery Probabilities: Epidemic Routing ensures very high delivery probabilities by flooding the network with message copies. This increases the likelihood of messages reaching their destination.
- Simple Implementation: The Epidemic protocol is relatively simple to program and easy to understand. Its lower complexity makes it accessible for deployment in emergency situations where highly specified solutions may be lacking.
- No Assumptions About Network Topology: Epidemic Routing supports the eventual delivery of messages to arbitrary destinations without relying on minimal assumptions about the topology and connectivity of the underlying network.
- Increased Flow Interest: The flooding nature of messaging in Epidemic benefits from increased flow interest, achieving lower delivery delays compared to other routing protocols.

Disadvantages:

- Resource Wastage: Epidemic floods the network with message copies, leading to significant resource wastage. This includes buffer space, which is used up to hold data for destinations that may not efficiently pass on the data, causing congestion in the system.
- Buffer Congestion: The protocol's reliance on replicating messages across the network can lead to buffer congestion. Nodes require space in their buffer to hold bundles of data, and the flooding-based technique exacerbates congestion issues.
- Redundant Message Copies: After a message reaches its destination, redundant copies may persist in the buffers of nodes. This results in the continued propagation of unnecessary copies, wasting potential resources.
- Lack of Feedback Mechanism: In contrast to end-to-end systems, Epidemic lacks an easy method of utilizing acknowledgment messages. The destination node remains

unaware if a received message has been lost, limiting feedback mechanisms for efficient communication.

- Inefficient Resource Usage: The protocol does not consider nodes with a low chance of reaching the destination, leading to inefficient use of resources like buffer storage. This oversight can impact the overall efficiency of the network.

2.7.3 Spray and Wait Routing Protocol

Spray and Wait[13] stands out as an effective protocol addressing challenges commonly associated with replication-based approaches. Being a Delay-Tolerant Network, it employs the store, carry, and forward mechanism, which involves working with buffers and within limited node ranges.

The fundamental idea behind Spray and Wait is to optimize the replication of messages in a network by imposing an upper limit. This strategy aims to reduce resource consumption while maintaining a high probability of message delivery.

Working Principle:

The protocol consists of two main phases- Spray and Wait, and there are two versions, with the primary difference lying in how messages are disseminated during the spray phase.

In the initial "spraying" stage, a predetermined number of message copies are sent from the source node to neighbouring nodes within communication range. The subsequent "wait" stage activates if the spraying phase fails to reach the destination. In the wait phase, relay nodes that received a copy of the message initiate direct transmissions. These nodes retain the message and the responsibility to deliver it to the destination, and the phase concludes when the message reaches its destination or when the message's Time To Live (TTL) expires.

- **Message Creation:** The process begins when a message is created for transmission within the network.
- **Limiting Replication:** Each message is associated with a predetermined limit value, representing the maximum allowed number of replications.
- **Encounter and Comparison:** When two nodes come into proximity, a comparison takes place between the messages stored in their respective buffers.
- **Message Exchange:** Nodes swap the required data during encounters. The limit value associated with the message is decremented by one after successful transmission.

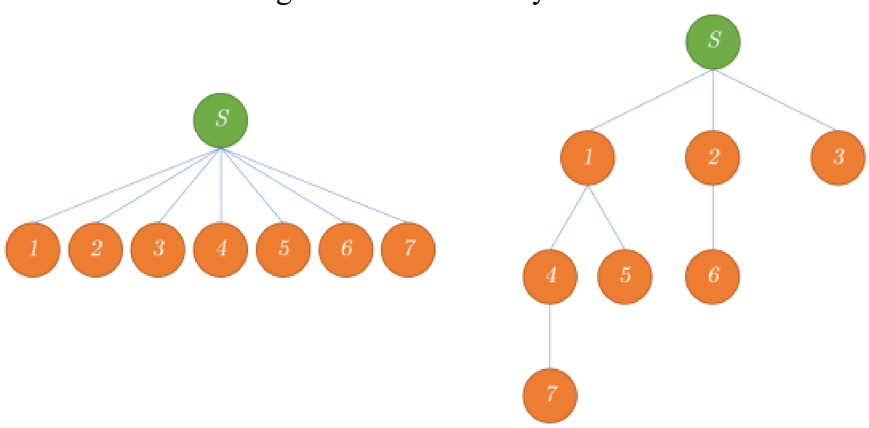

Figure 2.7.3 : Source node spreading message copies by Vanilla which is in the left and Binary which is in the right variants of Spray and Wait protocol

- **Vanilla:** In this simpler version, each message is associated with a limit value representing the maximum allowable replications. When two nodes meet, a comparison occurs, and necessary data is exchanged. The limit value decreases by one after the transmission. When a node's limit value reaches 1, it enters the wait phase and can no longer create additional replicates.
- **Binary:** Similar to the Vanilla version, this method involves creating messages with a limit value. However, in this version, a node passes half of its available copies to the receiving node during each encounter. This process repeats until a node has only one copy remaining, at which point it enters the wait phase. This approach allows for a faster spread of data from the source node.

While these versions offer improvements over the epidemic approach, their resource efficiency depends on the size of the limit parameter. If the limit is too large in a small node cluster, it leads to wasted resources. Conversely, if it is too small, the probability of message delivery decreases. Hence, an intelligently selected limit parameter based on the node's knowledge of its environment is crucial for the protocol's effectiveness.

During the wait phase, when a node's replication limit reaches one, it can only use direct transmission to pass the message to its destination. Direct transmission significantly reduces resource usage but introduces unbounded latency, as it relies on the chance encounter of the carrier node with the destination. The delivery probability is also contingent on the random meeting of the two nodes, potentially resulting in undelivered messages if their paths do not align.

Advantages and Disadvantages:

Advantages:

- **Resource Efficiency**: Spray and Wait significantly reduces resource consumption by limiting the number of message replications, optimizing the utilization of network resources.
- **Probability of Message Delivery**: The protocol maintains a high probability of message delivery, especially during the wait phase where custodian nodes take responsibility for direct transmissions to the destination.
- **Simplicity:** The protocol is relatively simple to understand and implement, making it accessible for deployment in various scenarios without extensive complexity.
- **Adaptability:** Spray and Wait is adaptable to different network environments and scenarios, allowing for effective communication in diverse settings, including those with intermittent connectivity.
- **Reduced Replication Overhead:** By employing a limit parameter and custodian-based direct transmissions, the protocol reduces the replication overhead compared to more aggressive replication-based approaches.

Disadvantages:

- Latency: The reliance on custodian-based direct transmissions introduces unbounded latency, as the successful delivery depends on the chance encounter of the carrying node with the destination.

- Delivery Probability Challenges: The probability of successful message delivery is contingent on the random meeting of nodes, and if the carrier node is traveling away from the destination for an extended period, there is a risk that the message may never be delivered.
- Parameter Selection Sensitivity: The effectiveness of the protocol is sensitive to the selection of the limit parameter. If the parameter is too large in a small node cluster, it leads to wasted resources, while if it is too small, the probability of message delivery decreases.
- Dependence on Encounter Patterns: The success of the protocol relies on encounter patterns between nodes. In scenarios with sparse encounters, the effectiveness of the protocol may be compromised.
- Limited Scalability: The protocol may face challenges in highly dynamic and large-scale networks where encounters are sporadic and network topology is frequently changing.

2.8 DTN Protocol Simulation

Simulation software offers a convenient, rapid, and cost-effective platform for assessing Delay-Tolerant Networking routing protocols. The ONE simulator is specifically designed for the evaluation of these protocols. This simulator facilitates routing among mobile nodes, assesses energy consumption, and enables the incorporation of mobile traces. It consists of six pre-configured, state-of-the-art DTN routing protocols, including First Contact, Direct Transmission or Delivery, Spray and Wait, Epidemic routing, MaxProp and PRoPHET, simplifying setup and simulation. Additionally, it features a user-friendly graphical interface that provides real-time visualization of node mobility.

3 Methodology

3.1 Introduction

Here, we introduced the ONE (Opportunistic Network Environment) Simulator as our chosen method for assessing and evaluating Delay-Tolerant Networking routing protocols and also elucidate the reason behind our choice.

3.2 DTN Simulator

Simulation software plays a crucial role in analysing the behaviour of Delay-Tolerant Networking (DTN) routing protocols. Designing an effective simulator for DTNs presents challenges due to the opportunistic nature of such networks, characterized by mobile nodes, sparse populations, and limited transmission ranges. While simulators like dtnsim and dtnsim2 have been created for DTN routing, they come with certain limitations. These simulators primarily focus on routing messages and lack the capability to inform routing protocols about established or broken network links between nodes. Additionally, the presentation of simulation progress in a console as text requires expertise for interpretation.

To address these limitations, the ONE simulator was developed, aiming to offer a practical and comprehensible simulation environment. Unlike its counterparts, ONE integrates routing simulation, movement modelling, visualization, and reporting into a single package. Notably, ONE utilizes a real map from Helsinki, Finland, providing actual movement patterns for nodes, including pedestrians, cars, and trams. The simulation's Graphic User Interface (GUI) captures node movement, facilitating user observation of movement patterns and enhancing the simulation's realism. ONE adopts the store-carry-forward approach for routing messages between nodes, enabling the evaluation of message delivery probability across various DTN routing protocols.

It is developed in the Java programming language, ONE allows for extension and flexibility, empowering users to modify the simulator's behaviour to investigate routing protocol performance. This design choice enables an in-depth exploration of routing protocols within the ONE simulator framework.

3.3 Overview of the ONE Simulator

The use of simulation software is crucial in evaluating the behaviour of DTN routing protocols. A group of computer scientists, supported by Nokia, developed the Opportunistic Network Environment (ONE) Simulator[2] to simulate the impact of networking protocols on the city of Helsinki. This software, based on Java, is highly complex, offering a range of tools for examining specific details about protocol performance, enabling accurate evaluations.

The ONE simulator utilizes a real map from Helsinki, incorporating actual movement patterns for nodes such as pedestrians, cars, and trams. The GUI captures node movement, providing users with a realistic observation of movement patterns.

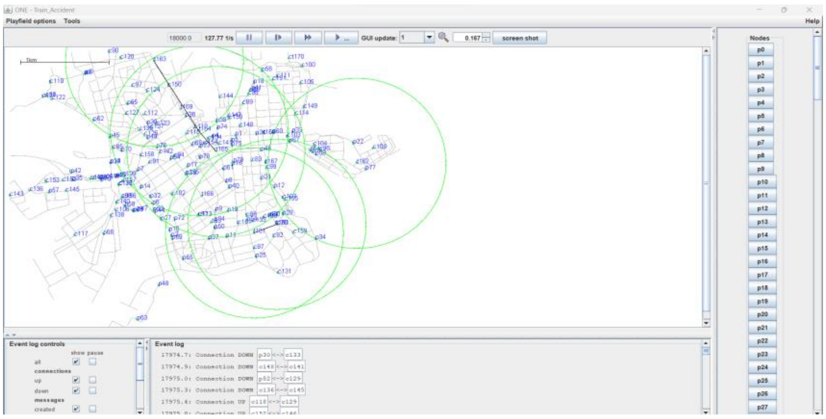

Figure 3.3.1: GUI mode of ONE Simulator

Developed in Java, ONE allows for extension and flexibility, granting users the ability to modify the simulator's behaviour for an in-depth exploration of routing protocols. The software's libraries can be imported into an Eclipse IDE, providing full control over the program. For those less experienced with Java, the default version can be used with settings files allowing significant customization. These files enable users to alter factors such as node ability, behaviour, and number within the environment, facilitating the comparison of protocol effects under various conditions.

The simulator features a robust reporting system that can be extended with additional options for examining new protocols. This flexibility allows users to potentially develop and compare new protocols to existing standards in the same environment for fair assessments. The report of protocol performance offers detailed information about scenario outcomes, including a separation of priority and non-priority messages for focused analysis.

At its core, ONE operates as event simulation engine, updating modules for node movement, inter-node contacts, routing, and message handling at each simulation step. Simulation results are collected through reports, visualizations, and post-processing tools, providing a comprehensive understanding of protocol behaviour.

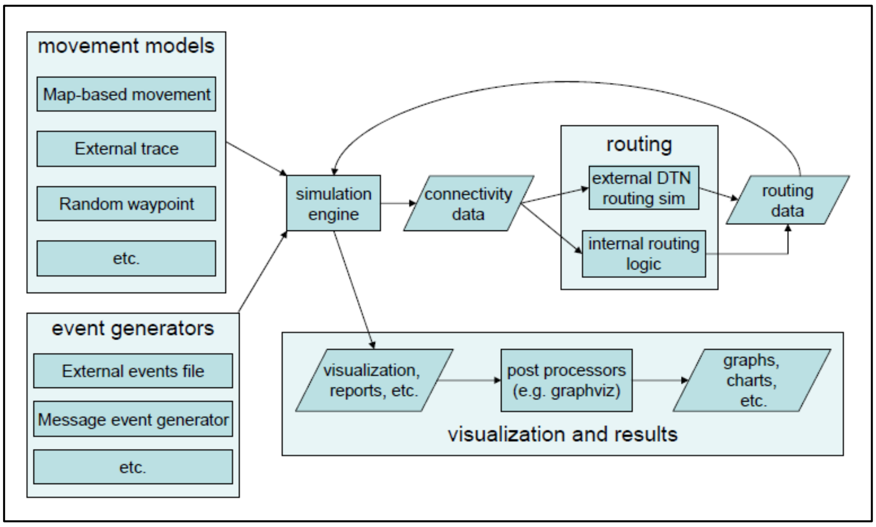

Figure 3.3.2: Overview of One Simulator

The ONE simulator's main structure includes:

Core: It contains the fundamental classes of the simulator, defining DTN hosts and their inter-host connections, along with interfaces. These classes are essential for every simulator run.

Routing: It contains classes related to routing, containing the available routing protocols within the simulation program.

Test: Although not directly linked to the simulator, it includes a set of individual system tests designed to ensure the system's proper functionality.

Reports: Serves as the repository for software simulation results, encompassing simulation statistics, Graph viz outputs, and compatibility data. Throughout the simulation, routing and movement modules act as the data source for the reporting module.

Movement: Facilitates the construction of movement models for nodes.

GUI: Holds classes for the user graphical interface. Additionally, it includes a sub-package for the playfield, defining classes that has to be displayed in the playfield's view.

UI: Encompasses basic user interface classes and text-based output classes.

Input: Provides interfaces and classes for accessing external resources.

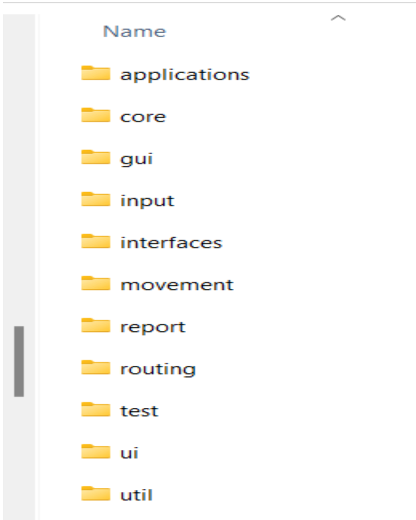

Figure 3.3.3: Main Structure of ONE

3.4 Movement Models

To create a realistic environment for evaluating the performance of DTN routing protocols, the ONE simulator incorporates movement models that constrain node mobility to predefined paths. Within ONE, the Map-Based Model, Shortest Path Map-Based Movement Model, and Route-Based Model emulate real-world movement scenarios. These models restrict node movements to specific paths defined in map data, with the default simulator settings utilizing the Helsinki map for simulation control.

The Map-Based Model (MBM) stands as the most straightforward movement model in ONE. Nodes are randomly positioned on the Helsinki map based on their type (e.g., pedestrians on walkways, cars and trams on roads). During simulation, nodes progress unidirectionally along their paths until reaching an endpoint or intersection, at which point they change direction. Importantly, nodes do not backtrack to their starting point but instead opt for random new directions, offering the advantage of delineating distinct paths for various node types.

The Shortest Path Map-Based Movement Model (SPMBM) is an enhancement of the MBM. Nodes, randomly placed on the map during simulation, determine the shortest path to their destination using the "Dijkstra's shortest path algorithm". Upon reaching their destination, nodes pause before selecting a new destination. The map data can include Points of Interest (POIs), acting as destination points. Nodes belonging to a specific group, such as pedestrians, can choose a POI as their subsequent destination.

The Route-Based Model (RBM) provides predetermined paths for a group of nodes to traverse on a map. These paths consist of stop points or POIs where nodes pause before advancing to the next stop point, employing the shortest path approach. RBM is particularly valuable for simulating nodes like cars, buses, and trams.

3.5 Routing Modules

Routing or forwarding of messages, often referred to as bundles, from a source to a destination is orchestrated by routing modules within the simulator. These modules oversee the replication, storage, and forwarding of messages in the ONE simulator. The ONE simulator offers six well-known routing protocols for DTNs, including First Contact, Direct Delivery, Epidemic, Spray and Wait, etc where I have used Epidemic and Spray and Wait in my scenario. While these routing modules share a commonality in allowing contact nodes to exchange messages until they reach a predefined destination, variations exist in their behaviour concerning message replication and forwarding.

Additionally, the ONE simulator incorporates a fundamental energy module, a derivative of the Epidemic Router, designed to quantify the energy consumed during node activities such as transmission and scanning. However, it's important to note that this protocol does not account for the energy consumed when a node receives a message and does not advance the state of making routing protocols energy-aware. In the subsequent chapter, the simulator's functionality will be expanded to introduce an energy-aware routing protocol for DTNs, along with the incorporation of a metric to measure dead nodes in a simulation.

3.6 Events Generation

The simulation process involves the generation of events, a task managed by event generators. In the case of the ONE simulator, an external events framework is incorporated, allowing the import of messages through trace files or event generator modules from external sources. Trace files, typically derived from other programs, are text documents containing event timestamps such as the establishment of new links, creation or removal of messages. Event generator modules, on the other hand, dynamically generate these events, and their configurations can be adjusted using the simulator settings. Notably, the simulator supports the simultaneous simulation of multiple event generators.

3.7 Report Modules

Simulation results in ONE are produced through report modules, which receive input from the event generators, capturing data dropped messages or node movements during the simulation. These report modules then either write or store event-related information in an output file, generating a comprehensive summary at the simulation's conclusion. Specifically, the simulator creates a MessageStatsReport file for each simulation within a designated reports folder. This file encompasses a summary of metrics such as messages created, dropped, delivered, average latency, and more. The purpose of this report is to assess the performance of DTN routing protocols.

```
Message stats for scenario Epidemic_Bf_5M
sim_time: 3600.1000
created: 146
started: 1350997
relayed: 1350954
aborted: 0
dropped: 1350503
removed: 0
delivered: 25
delivery_prob: 0.1712
response_prob: 0.0000
overhead_ratio: 54037.1600
latency_avg: 9.0720
latency_med: 7.8000
hopcount_avg: 6.8400
hopcount_med: 6
buffertime_avg: 1.4700
buffertime_med: 0.6000
rtt_avg: NaN
rtt_med: NaN
```

Figure 3.7 MessageStatsReport

3.8 Simulation Modes

The ONE simulator provides users with the flexibility to execute simulations in two distinct modes: the Graphical User Interface (GUI) mode and the batch mode. In GUI mode, the simulation unfolds in real time, offering users the ability to monitor node movement and message exchanges, making it particularly valuable for testing and debugging. On the other hand, batch mode is employed for running extensive simulations with varying configurations. In both modes, the simulator generates reports, which can be examined using post-processing tools to generate graphs and plots.

3.8.1 GUI Mode

The ONE simulator's GUI is structured into three main sections. The most prominent section is the playfield graphics, which showcases the simulation time in seconds. It includes control buttons like play, pause, and step forward, a dropdown menu for GUI update speed, a zoom button, and a screenshot capture button. The graphical representation on this playfield depicts the geographical simulation area. It illustrates the node locations on the map, their communication range, movement paths, and the quantity of messages. Nodes are distinguishable by their names, and the green rings surrounding them indicate their communication range. Nodes in contact for communication are denoted by the touching or overlapping of their rings. Messages are represented by blue and green blocks atop nodes, while red blocks signify dropped messages.

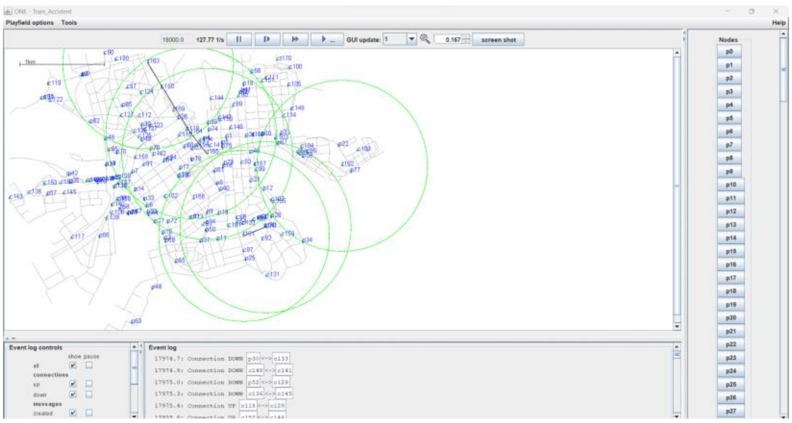

Figure 3.8.1: ONE's GUI mode

The second section of the GUI is dedicated to Event log controls and the Event log panel. All events are presented in the Event log panel on the map. Displayed just above the event log are the geographical coordinates of the node and the message count. The "event log controls" enable the user to manage the information displayed in the event log.

Finally, the third section of the GUI features a list of nodes in the simulation. Nodes are categorized by group names (e.g., p for pedestrian, c for car, t for tram) in the list. The number of nodes in the list is contingent on the simulator settings. Clicking on a node in the list reveals its path on the map.

3.8.2 Batch Mode

The batch mode operates simulations through a command line interface, lacking real-time visualization. To initiate batch mode in the command line, users type the command "one.bat –

b n," where n denotes the number of runs. The simulation commences by displaying the scenario name (default_scenario) and the remaining number of scenarios in the batch. Report modules gather simulation results. An advantageous feature in batch mode is "run indexing," facilitating the straightforward configuration of settings for various simulations using a single configuration file. The efficiency of batch mode lies in its faster execution, attributed to its omission of graphics processing.

Figure 3.8.2: ONE's Batch mode

3.9 Scenario Settings

To simulate a scenario, the ONE simulator is set up by specifying scenario settings, which involve defining parameters such as the number of nodes, groups, routing models, movement models, and more. Configuration is achieved through a text-based configuration file, typically named default_settings, which serves to establish various parameters for the simulator. While the behaviour of modules in ONE during simulation is determined by Java code implementation, the intricate details of these behaviours are articulated through the configuration file. Upon startup, the simulator dynamically incorporates movement models, routing modules, event generators, and report modules by parsing the configuration file. Additionally, optional files are loaded if specified as parameters in the configuration file.

4 Evaluation and Discussion

4.1 Introduction

In this chapter we will compare and analyse the performance of the two protocols that have been used in our scenario depending upon various performance metrics using the map of Helsinki using One Simulator. For comparison I have taken three metrics i.e. Buffer Size, Message size and Network Density.

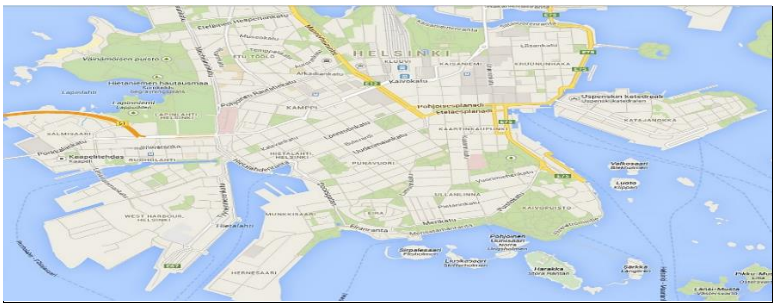

Figure 4.1: Helsinki, Finland Map used in the experiment

4.2 Experiment Aim and Setup

We have conducted experiments with the aim of analysing and evaluating the performance of the two protocols in our scenario depending upon the three metrics that we have taken. The first is the buffer size, where it is expected to increase the number of messages delivered with the increase in the amount of buffer size and the second metric is the message size, where if the message size is less, then the probability density of the message being delivered is more and vice versa. Our final metric is network density, where if the network density is more, the congestion increases and the number of messages delivered will be less. Therefore, it is expected to get the results in this way from the defined settings that we have considered for our scenario in the table below.

Parameters	Settings
Number of node types	6
Node Types	Pedestrians, Cars and Trams
Message TTL	5 hours
Interface	Bluetooth and High-Speed Interface
Simulation Time	12h
Movement	Map Based Movement using Helsinki map

Table 4.2.1: Default settings of the experiment

Parameters	Settings
Number of node groups	11
Node Types	Pedestrians, Trains, Ambulances, Cars, Communication Towers and Control Centres
Message TTL	300 minutes (5 hours)
Interface	High-Speed Interface
Simulation Time	3600s
Map	Helsinki
Movement	Map Based Movement
Transmission Range	800m
Transmission Speed	10M

Table 4.2.2: Defined settings of the experiment

The below figure is the simulation of our scenario considering the above parameter settings. It can be clearly seen from the map that there is an accident that has happened and the information has been sent to the control centres. In my scenario tc1 and tc2 are the trains that has been collided and marked them as stationary nodes. The message has been delivered to cc (control centres) that there is an accident.

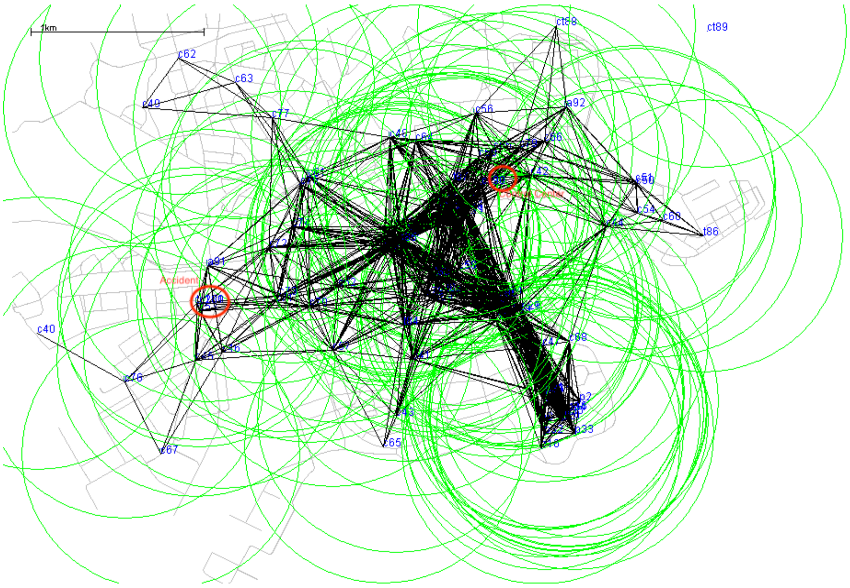

Figure 4.2: Emergency Scenario on the ONE Simulator

4.3 Observed Metrics

We have evaluated the performance of DTN routing protocols using the following metrics -

- **Buffer Size** - it corresponds directly to the quantity of messages that a node can store in its cache memory at any given time and send to other nodes. A node's ability to hold messages is determined by its buffer sizes. The various parameters based on this metrics which were used to compare the two protocols are:
    1. **Messages Delivered –** when the has reached the destination node, it is said to be delivered
    2. **Messages Dropped –** when the nodes could not reach the destination nodes due to various reasons and has been dropped in between
    3. **Delivery Probability- i**s the total number of messages that has been delivered over the total number of messages that are created, which also means as success ratio
    4. **Overhead Ratio-** this refers to the number of redundant messages that has been relayed to deliver a single packet
    5. **Average Latency-** the average time between a message that has been generated and when it is received at the destination node
    6. **Average hop count-** the average number of devices that has been used as an intermediate through which the message has been forwarded from source to destination
- **Message size-** In real-world situations, the size of messages can vary depending on the medium. Larger message sizes occupy more space in the buffer, leading to quicker saturation and a higher likelihood of dropped messages. Consequently, the delivery rate of messages is significantly reduced when the buffer becomes full in a shorter timeframe. The various parameters based on this metrics which were used to compare the two protocols are:

1. **Messages Dropped -** when the nodes could not reach the destination nodes due to various reasons and has been dropped in between
2. **Delivery Probability - i**s the total number of messages that has been delivered over the total number of messages that are created, which also means as success ratio
3. **Average Latency -** he average time between a message that has been generated and when it is received at the destination node

- **Network Density**- Network density depends on the number of hosts that are present in the network. If the number of nodes is more it means that the network is dense and vice versa. This directly affects the congestion inside the network and as we know that if congestion increases the probability of delivery decreases so we can consider network density as a parameter to evaluate the performance of both the protocols. The various parameters based on this metrics which were used to compare the two protocols are:

1. **Messages Dropped -** when the nodes could not reach the destination nodes due to various reasons and has been dropped in between
2. **Delivery Probability -** the total number of messages that has been delivered over the total number of messages that are created, which also means as success ratio
3. **Average Latency -** he average time between a message that has been generated and when it is received at the destination node

4.4 Simulation Results

We have taken out the results and with those taking as inputs we have plotted charts in excel for a better visualisation and clear understanding for the metrics Buffer size. Table Epidemic and Spray and Wait shows the message statistics report of the two protocols for the following nodes.

4.4.1 Statistical overview against Buffer sizes 5M, 25M and 50M in both the protocols

The table below shows the variation of various metrics of both the protocols by taking the buffer sizes as 5M, 25M and 50M so as to evaluate their performances.

	Epidemic (5M)	Epidemic (25M)	Epidemic (50M)
Messages Created	146	146	146
Messages Started	1350997	889237	852303
Messages Relayed	1350954	889194	852260
Messages Dropped	1350503	886300	846332
Messages Delivered	25	56	82
Delivery_prob	0.1712	0.3836	0.5616
Overhead_ratio	54037.1600	15877.4643	10392.4146
Latency_avg	9.0720	20.9268	30.4720
Hopcount_avg	6.8400	5.1607	5.3415

Table 4.4.1.1: Epidemic

	SnW (5M)	SnW (25M)	SnW (50M)
Messages Created	146	146	146
Messages Started	769	840	846
Messages Relayed	769	840	846
Messages Dropped	786	529	317
Messages Delivered	39	110	116
Delivery_prob	0.2671	0.7534	0.7945
Overhead_ratio	18.7179	6.6364	6.2931
Latency_avg	491.8256	656.4900	695.3414
Hopcount_avg	2.7949	2.7455	2.7069

Table 4.4.1.2: Spray and Wait

- **Messages Delivered against buffer sizes in Epidemic and SnW**

The figure below shows the message delivered graph, that indicates the number of messages that has reached the destination successfully. As seen in the graph below, the number of messages that gets delivered completely increases linearly with the buffer sizes. Both the Epidemic and Spray and Wait gradually increases with increase in the buffer sizes and successful delivery of the messages.

The gradual increase in the number of delivered messages with larger buffer sizes suggests that an ample buffer capacity positively impacts the overall efficiency of these DTN protocols. A larger buffer size allows for a more extensive storage capacity, accommodating a greater number of messages as they traverse the network. This scalability is particularly beneficial in scenarios where temporary disruptions in connectivity or delays in message forwarding occur, as a sufficiently sized buffer can store messages until opportune moments for transmission arise.

Although in both the protocols its increasing linearly but the messages delivered the most is in Spray and wait and the lowest in Epidemic. Considering this, it can be summarised that spray and wait gives better results when compared to epidemic in this metric.

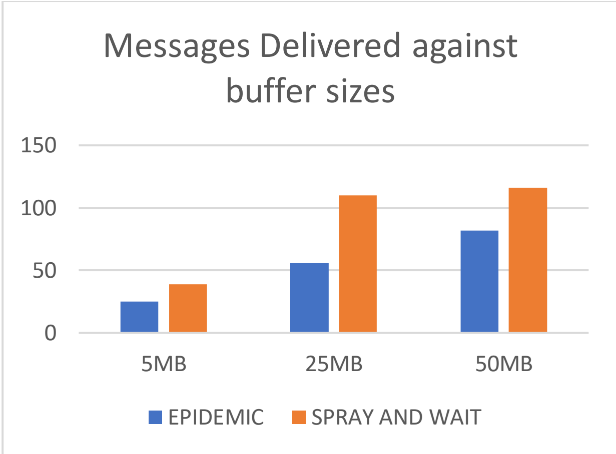

Figure 4.4.1.1: Messages Delivered

- **Messages Dropped against buffer sizes in Epidemic and SnW**

The presented data outlines the relationship between the number of dropped messages and varying buffer sizes for both the Epidemic and Spray and Wait (SnW) protocols. A clear pattern emerges, indicating that as buffer sizes increase, there is a corresponding decrease in the number of dropped messages. This inverse relationship suggests that larger buffer sizes contribute to a reduction in packet loss for both the Epidemic and SnW protocols. The observed trend underscores the significance of appropriately dimensioned buffers in mitigating message loss within opportunistic networks employing these DTN protocols. Larger buffer capacities allow for improved message retention during periods of network disruption or delays, thereby minimizing the likelihood of dropped messages.

The inverse proportionality between buffer sizes and dropped messages emphasizes the critical role that buffer size plays in optimizing the performance of both Epidemic and SnW protocols. It implies that, by increasing the buffer size, the network can better cope with intermittent connectivity issues, ensuring a higher success rate in message delivery. This analysis provides valuable insights into the relationship between buffer sizes and message drop rates, guiding decisions on buffer dimensioning for enhanced performance in DTN scenarios. Thus Spray and Wait is far better than that of Epidemic.

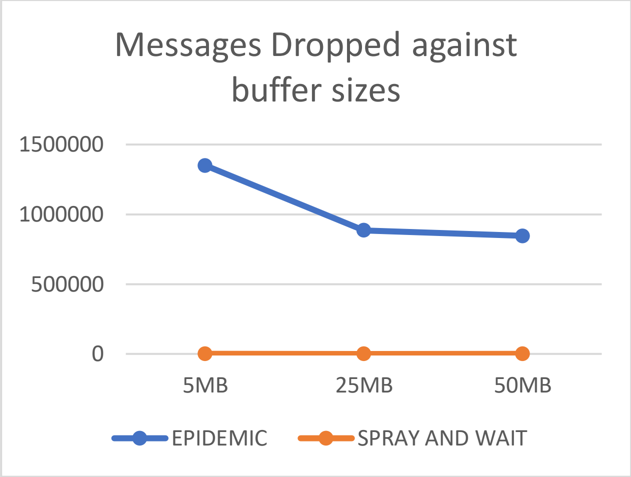

Figure 4.4.1.2: Messages Dropped

- **Delivery Probability against buffer sizes in Epidemic and SnW**

The provided chart illustrates the delivery probability of both the Epidemic and Spray and Wait (SnW) protocols across varying buffer sizes. The delivery probability serves as a key metric, representing the success ratio of the protocols, indicating the proportion of messages successfully delivered relative to the total number created. A noticeable trend emerges from the data, revealing that as the size of the buffer increases, there is a corresponding improvement in the success ratio or delivery probability for both protocols. The positive correlation between buffer size and delivery probability suggests that larger buffers positively impact the overall efficiency of the DTN protocols. A larger buffer provides the network with increased storage capacity, enabling better handling of intermittent connectivity issues or delays in message forwarding. This leads to a higher success rate in delivering messages to their intended destinations. Furthermore, the observation that the delivery probability in the Spray and Wait protocol is better than that in the Epidemic protocol indicates the comparative effectiveness of these two protocols in the context of opportunistic networks.

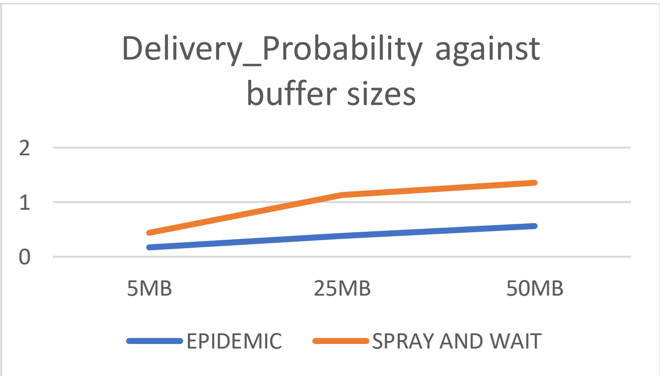

Figure 4.4.1.3: Delivery Probability

- **Overhead Ratio against buffer sizes in Epidemic and SnW**

The below chart depicts the overhead ratio of those two protocols. From the figure we can see that Spray and Wait has the lowest overhead ratio than that of Epidemic and thus better than Epidemic.This protocol uses the Spray and Wait approach to sprays a limited number of messages in the network and hence reduce the number of redundant messages. Epidemic routing protocol has the highest overhead ratio which increases with an increase in message replication, a direct relation to increase in node size. Utilizing the Spray and Wait approach, this protocol disperses a finite number of messages within the network, effectively minimizing redundant messages. In contrast, the Epidemic routing protocol exhibits the highest overhead ratio, and this ratio escalates with the replication of messages, establishing a direct correlation with the increase in the node size.

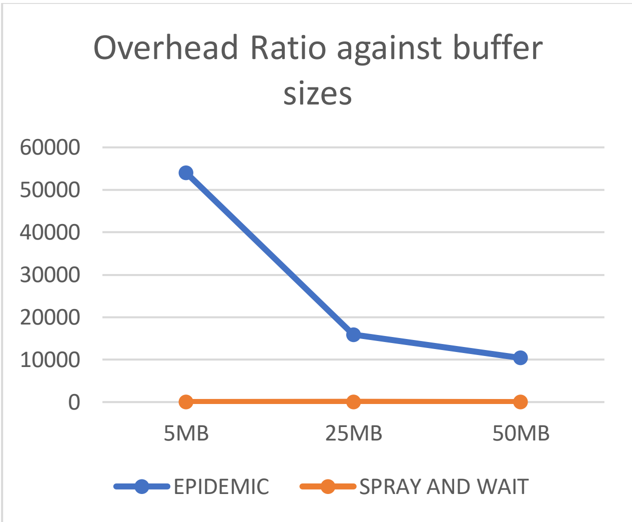

Figure 4.4.1.4: Overhead Ratio

- **Average Latency against buffer sizes in Epidemic and SnW**

The analysis conducted indicates a notable difference in the average latency between the Epidemic and Spray and Wait (SnW) protocols, with SnW exhibiting considerably higher latency compared to Epidemic. This implies that the time taken to deliver messages is significantly prolonged in the case of Spray and Wait when compared to the Epidemic protocol. The observed variance in average latency suggests that the two protocols operate with distinct efficiency in terms of message delivery time. The higher latency in the Spray and Wait protocol may be attributed to its specific approach of spraying a limited number of messages and subsequently waiting for potential encounters with other nodes for message forwarding. This method, while reducing redundancy, may introduce additional delays in reaching the destination.

Understanding the trade-offs between latency and other performance metrics is crucial for selecting the most suitable protocol for a given network scenario. The findings emphasize that in situations prioritizing lower latency for message delivery, the Epidemic protocol may be a more suitable choice compared to the Spray and Wait protocol.

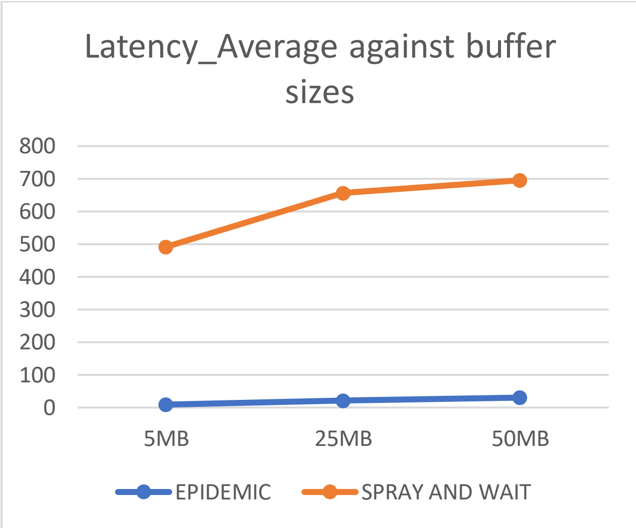

Figure 4.4.1.5: Average Latency

- **Average hop count against buffer sizes in Epidemic and SnW**

The provided figure illustrates a notable difference in the average hop count between the Epidemic and Spray and Wait (SnW) protocols, with Epidemic exhibiting a lower average hop count compared to SnW. This indicates that Epidemic requires a smaller number of intermediate nodes to successfully deliver messages from the source to the destination, in contrast to the Spray and Wait protocol. The observed disparity in average hop count underscores a key distinction in the routing efficiency of the two protocols. A lower average hop count in the Epidemic protocol suggests a more direct and efficient route for message transmission, minimizing the number of intermediate nodes involved in the delivery process. On the other hand, the higher average hop count in the Spray and Wait protocol may be attributed to its strategy of spraying messages and waiting for encounters with other nodes, potentially leading to a more circuitous route.

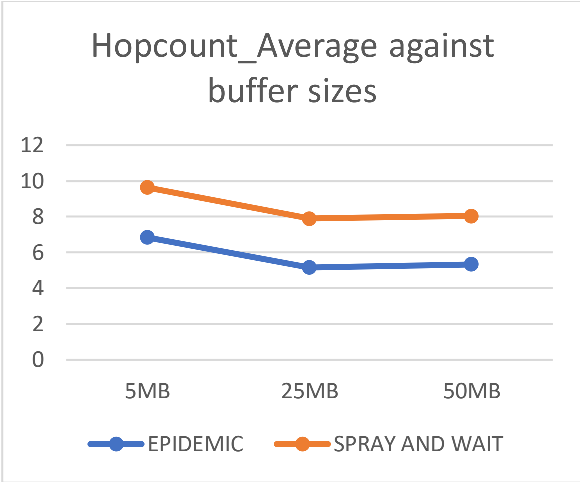

Figure 4.4.1.6: Average hop count

4.4.2 Statistical overview against Network Density 20, 60 and 80 number of hosts in both the protocols

The table below shows the variation of various metrics of both the protocols by taking the network density as 20 (low density), 60(medium density) and 80 (high density) as the number of hosts so as to evaluate their performances.

	SnW (Less Density)	SnW (Medium Density)	SnW (High Density)	Epidemic (Less Density)	Epidemic (Medium Density)	Epidemic (High Density)
Messages Created	146	146	146	146	146	146
Messages Started	842	842	841	604641	1927551	2343529
Messages Relayed	842	842	841	604619	1927488	2343446
Messages Dropped	568	491	503	603691	1925069	2340130
Messages Delivered	119	112	111	46	49	33
Delivery_prob	0.8151	0.7671	0.7603	0.3151	0.3356	0.2260
Overhead_ratio	6.0756	6.5179	6.5766	13142.8913	39335.4898	71012.5152
Latency_avg	388.2798	250.4143	475.9739	29.4304	18.6388	10.6394
Hopcount_avg	2.5630	2.5357	2.5676	5.2174	5.1020	4.9091

Table 4.4.2: Statistical Overview against Network Density

- **Delivery Probability against Network Density in Epidemic and Spray and Wait**

The figure below depicts that the delivery probability is more in Spray and Wait as that of Epidemic and is decreasing with increase in the number of nodes. The more the network density is, the less delivery probability it will be due to the increase in the congestion. Because there are high chances that messages might get dropped in between due to more congestion and vice versa where if there is less congestion then the nides will get delivered better.

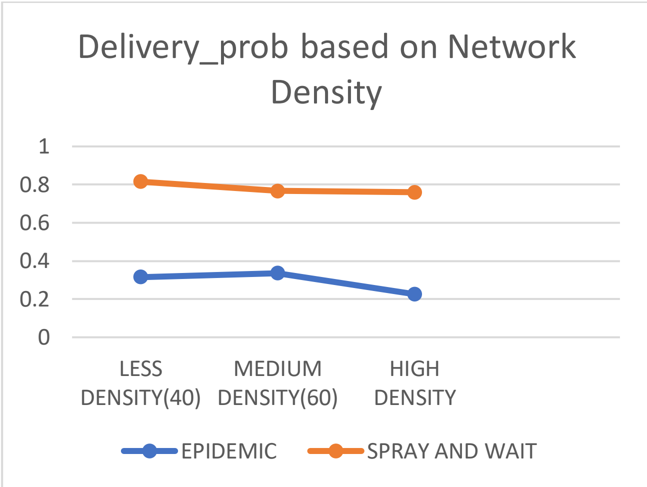

Figure 4.4.1.7: Delivery probability

- **Average Latency against Network Density in Epidemic and Spray and Wait**

The analysis of average latency against network density for the Epidemic and Spray and Wait protocols reveals interesting trends. As the number of nodes in the network increases, the latency rate decreases. This phenomenon is attributed to the higher probability of contact with potential relay nodes, subsequently enhancing the overall delivery probability of messages.

In the case of the Epidemic protocol, the graph initially exhibits a decreasing trend in latency as network density rises. This suggests that with a growing number of nodes, the Epidemic protocol benefits from increased opportunities for contact with potential relay nodes, leading to more efficient message delivery.

On the other hand, the graph for the Spray and Wait protocol shows a consistent and slow reduction in latency as network density increases. This more gradual decrease in latency is attributed to the fact that the number of messages being sent only increases marginally. Thus we can say is Epidemic is better in that case.

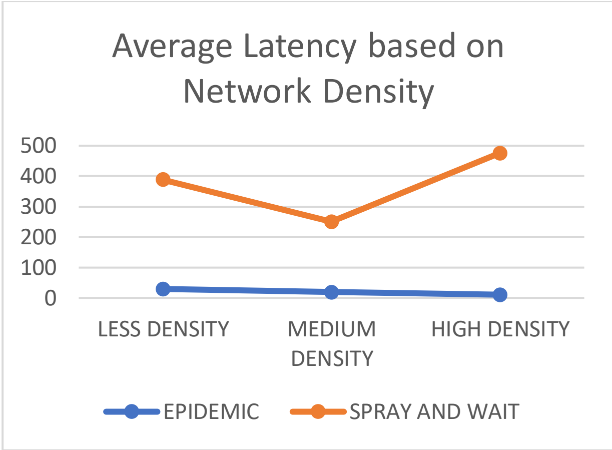

Figure 4.4.1.8: Average Latency

- **Messages Dropped against Network Density in Epidemic and Spray and Wait**

The chart below depicts that Epidemic has more number of messages dropped with the increase in the network density and for Spray and Wait it is less than that of Epidemic which clearly says that Spray and Wait outperforms Epidemic. It has a very consistent and lower drop rate than the other which makes it more scalable.

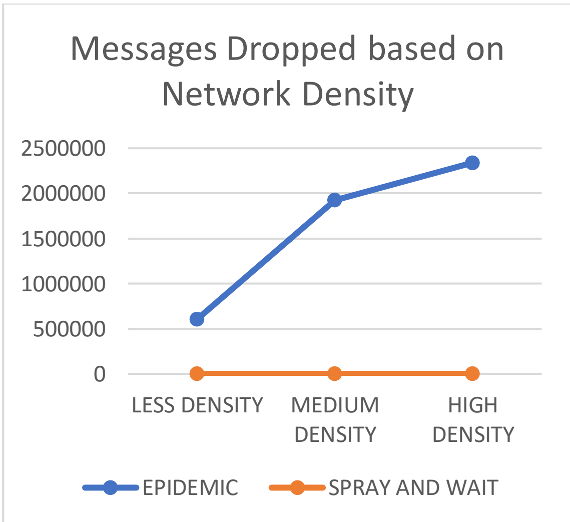

Figure 4.4.1.9: Messages Dropped

4.4.3 Statistical overview against Message Size between 300k to 2M in both the protocols

The table below shows the variation of various metrics of both the protocols by taking message size as 300-800k(Small), 900-1.5M(Moderate) and 1.6M-2M (Large) as the number of hosts so as to evaluate their performances.

	SnW (Small Message Size)	SnW (Moderate Message Size)	SnW (Large Message Size)	Epidemic (Small Message Size)	Epidemic (Moderate Message Size)	Epidemic (Large Message Size)
Messages Created	146	146	146	146	146	146
Messages Started	854	824	783	633660	352305	330515
Messages Relayed	854	824	783	633638	352188	330360
Messages Dropped	496	669	704	632333	351679	330080
Messages Delivered	131	104	86	44	33	33
Delivery_prob	0.8973	0.7123	0.5890	0.3014	0.2260	0.2260
Overhead_ratio	5.5191	6.9231	8.1047	14399.8636	10671.3636	10009.9091
Latency_avg	414.0634	386.7038	339.6919	14.7159	29.0576	25.3606
Hopcount_avg	2.5496	2.5577	2.6047	4.3864	4.6970	4.5758

Table 4.4.3: Statistical Overview against Message Size in Epidemic and Spray and Wait

- **Delivery Probability against Message size in Epidemic and Spray and Wait**

With delivery probability, the probability seems to decrease with respect to increase in the message size. Although in Spray and Wait it begins at a higher probability, but it seems to decrease at a steeper way than that of Epidemic. The reason might be that the destination nodes buffer must have filled with huge messages and thus dropping other messages. Thus, both the protocols follow almost a similar type of pattern.

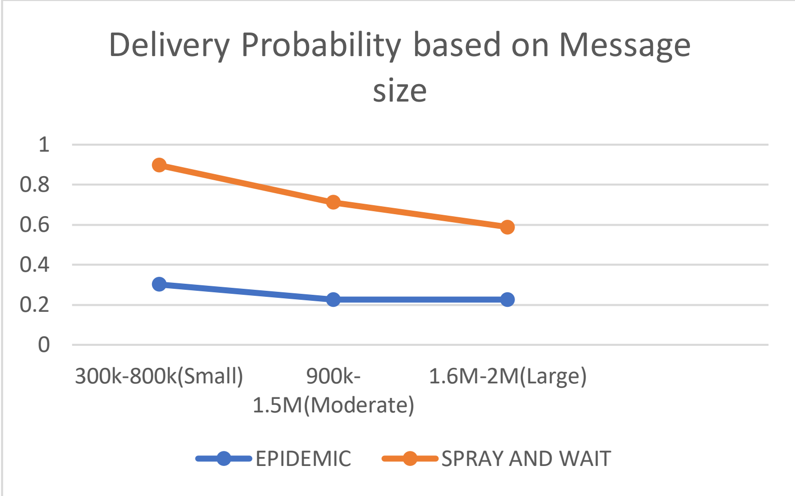

Figure 4.4.1.10: Delivery Probability

- **Messages Dropped against Message size in Epidemic and Spray and Wait**

From the chart below, we can sum up two conclusions with this metric i.e. Spray and Wait was not affected significantly by the increase in the message size. But in Epidemic we can see the gradual decrease in the message loss. Whether in Epidemic the messages dropped rapidly and made a significant impact. Thus, the loss of messages in Spray and Wait is comparatively better than that of Epidemic.

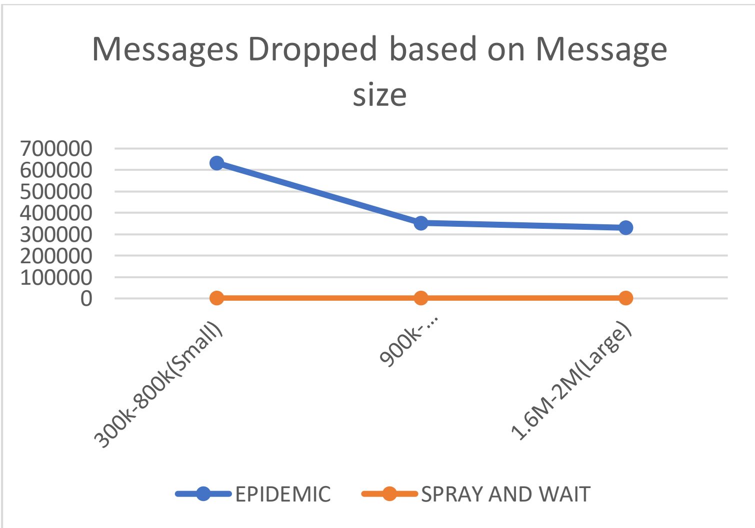

Figure 4.4.1.11: Message Dropped

- **Average Latency against Message size in Epidemic and Spray and Wait**

The below chart depicts the average latency between the two protocols based on the message size and it is observed that although the average latency of Spray and Wait is more, but it decreases gradually with the increase in the message size. The reduction in latency, despite the potentially negative impact of larger message sizes, can be attributed to fewer messages being sent from each node. The conservative spread of messages in the network is a result of the size of the messages, contributing to the observed decrease in latency. While in Epidemic, the average latency increase with increase in the message size, which is a positive side of this.

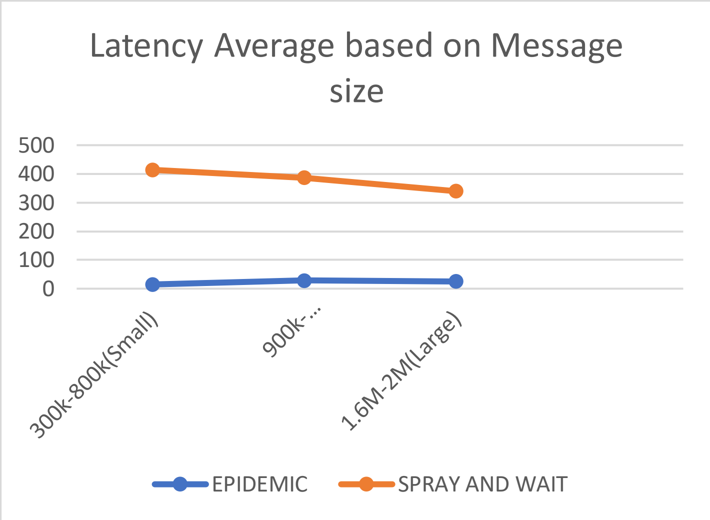

Figure 4.4.1.12: Average Latency

5 Pros And Cons of Opportunistic Networks in various scenarios

In the context of a train accident as an emergency scenario, Delay-Tolerant Networking (DTN) opportunistic protocols present both advantages and challenges. One notable advantage is the ability to facilitate communication in areas with disrupted or absent network infrastructure. In a train accident, where traditional communication channels may be compromised, DTN protocols allow for message dissemination through opportunistic encounters between mobile nodes. This resilience in the face of infrastructure failures is a crucial benefit, enabling communication to persist even in challenging conditions. However, one significant challenge lies in the potential delay and increased latency associated with opportunistic communication. As nodes may need to opportunistically encounter each other for message relay, the time taken for critical information to reach its destination can be extended, which may impact emergency response times.

In another emergency scenario, such as a natural disaster or a remote location without reliable infrastructure, DTN opportunistic protocols offer similar advantages and face analogous challenges. These protocols excel in scenarios where traditional networks are unavailable or unreliable, allowing communication to occur through intermittent connections. The flexibility of DTN protocols is particularly beneficial in situations where infrastructure damage or geographical constraints limit traditional communication methods. However, a common challenge in such scenarios is the potential for increased message duplication and latency. The lack of continuous connectivity may lead to delays in message delivery, and the opportunistic nature of encounters can result in duplicate messages, affecting the efficiency of communication and potentially overwhelming the network. Therefore, while DTN opportunistic protocols provide valuable solutions for communication in emergency scenarios, careful consideration is required to address the associated challenges.

6 Conclusion and Future Enhancemets

6.1 Conclusion

To sum up, opportunistic networks like DTNs represent a novel network paradigm that enhances the end-to-end connectivity compared to traditional networks. DTNs have the capacity to establish connections in areas and among devices that are underserved by conventional networks. Communication in DTNs is facilitated through temporary links for data routing. This paper provides an overview of Opportunity Networks, introduces the ONE simulator, and delves into two DTN protocols: the epidemic protocol and the Spray and Wait protocol. Firstly, the trains collision scenario has been created along with various communications happening between the pedestrians, cars, ambulances, communication towers and the control centres in this emergency situation. Then configuration files are set up to simulate in ONE. Lastly, the performance of the two protocols is analysed and compared based on the simulation results in generated reports.

Firstly, the analysis was made based on the performance of the two protocols at different buffer sizes, network density and message sizes using various metrics such as delivery probability, messages delivered, messages dropped, average latency, overhead ratio, and average hop count. After analysing, its seen that Spray and Wait protocol outperforms Epidemic considering the default setting experiments in the scenario.

6.2 Future Enhancements

In future implementations, the integration of a mechanism for buffer removal could be considered to eliminate duplicate messages. This has the potential to decrease both latency and the incidence of message loss. Even in emergency scenarios, a priority-based queuing system could be introduced, where high-priority messages are given precedence. Messages categorized as low priority or non-emergency could be directed along a less efficient route, ensuring that critical communications take a more efficient and expedited path.

Since this scenario is very much related to real-world emergency scenarios, therefore it would also be used in some other real-world situations where the main priority would be sending messages in case of such emergencies when there is a train accident and thus the ambulances can be called up to help out.

References

[1] M. Radenkovic and C. Oham , "Congestion Aware Spray and Wait Protocol: A Congestion Control Mechanism For The Vehicular Delay Tolerant Network," International Journal of Computer Science & Information Technology, vol. 7, no. 6, 6 December 2015.

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

[3] S. Yousefi, M.S. Mousavi, and M. Fathy. "Vehicular ad hoc networks (VANETs): challenges and perspectives". In: 2006 6th International Conference on ITS Telecommunications. IEEE. 2006, pp. 761–766.

[4] J. Rodrigues, J. Isento, J. Dias, V. Soares, B. Silva, N. Magaia, P. Pereira, A. Casaca, C. Pastor and J. Gallego, "The Vehicular Delay-Tolerant Networks (VDTN) Euro-NF Joint Research Project," 2011.

[5] L. Pelusi, A. Passarella, and M. Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. 2006.

[6] A. Vahdat and D. Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. 2000.

[7] P. Jadhav and R. Satao, "A Survey on Opportunistic Routing Protocols for Wireless Sensor Networks," Procedia Computer Science, vol. 79, pp. 603-609, 2016.

[8] Chiara Boldrini, Kyunghan Lee, Melek Önen, Jörg Ott, "Opportunistic Networks", p2-5, July 2014

[9] JAIN, S., FALL, K., AND PATRA, R. Routing in a Delay Tolerant Network. In Proc. of ACM SIGCOMM (2004).

[10] KERÄNEN, A. Opportunistic Network Environment Simulator. Special Assignment report, Helsinki University of Technology, Department of Communications and Networking, May 2008.

[11] LINDGREN, A., DORIA, A., AND SCHELEN, O. Probabilistic routing in intermittently connected networks. In The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR) (2004).

[12] PITKÄNEN, M. J., KERÄNEN, A., AND OTT, J. Message Fragmentation in Opportunistic DTNs. In Proc. of the 2nd WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC) (2008).

[13] SPYROPOULOS, T., PSOUNIS, K., AND RAGHAVENDRA, C. S. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In Proc. of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN) (2005).

[14] ZHANG, Z. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. IEEE Communications Surveys and Tutorials 8, 4 (January 2006), 24–37.

[15] Isaac Woungang et al. Routing in Opportunistic Networks. Springer, 2013.

[16] Athanasios V. Vasilakos, Yan Zhang, and Thrasyvoulos Spyropoulos. Delay Tolerant Networks: Protocols and Applications. CRC Press, 2011.

[17] Khalil Massri et al. "Routing Protocols for Delay Tolerant Networks: A Reference Architecture and a Thorough Quantitative Evaluation". In: (2016). MDPI.

[18] Pietro Manzoni et al. "DTN Protocols for Vehicular Networks: An Application Oriented Overview". In: (2015). IEEE.

[19] Dongkyun Kim et al. "Vehicular Delay Tolerant and Sensor Networks: Protocols and Applications". In: (2015). SAGE Journals.

[20] Monika Aneja and Vishal Garg. "Simulation of Epidemic, Spray and Wait and First Contact Routing Protocols in Delay Tolerant Network". In: (2015), pp. 30– 34. IOSR Journal of Electronics and Communication Engineering