Master Thesis

Opportunistic Routing with Minimum Latency for MANETs with Intermittent Link Failures

Professor Satoshi Fujita

A thesis presented for the degree of
Master of Informatics and Data Science

M200580 Gurunathan Vijayalakshmi

Graduate School of Advanced Science and Engineering
Hiroshima University
15 August 2022

ACKNOWLEDGMENTS

I would like to express my heartfelt thanks to my supervisor Professor Satoshi Fujita and my sub-supervisors, Professor Okamura Hiroyuki, and Associate Professor Kitasuka Teruaki, for their passionate guidance, endless support, and instructive suggestions not only during the thesis preparation process, but also throughout my studies at Hiroshima University.

I would like to thank my lab mates, especially Iwao Taiki, Kobayashi Kakeru, and Tao Tao for their friendly assistance and insightful conversations. Finally, I would like to thank my family for their moral support, as well as all of my friends for their generous support.

ABSTRACT

PROPHET is a probabilistic routing protocol for Delay Tolerant Networks (DTN). In such networks, there is no guaranteed routing path from a source node to a receiving node so, most of the connections between the nodes are temporary, and rendering traditional routing protocols unable to deliver messages between hosts. This work considers the problem of routing a message from source node a to destination node d as quickly as possible by changing the mobile nodes as needed by using the bipartite graph with minimum hop count to destination. An AODV approach is used to realize the message delivery towards the given destination along a path with the minimum expected latency.

Keywords- Delay tolerant networks, Prophet, bipartite graph, AODV

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The Delay Tolerant Network (DTN) is an ad-hoc wireless network that tolerates intermittent connectivity. DTN was derived from the interplanetary Networks (IPN) project, which began in the 1970s. A DTN's goal is to provide reliable communication in environments with frequent network disruptions and extremely long transmission delays, and it has the potential to interconnect devices in regions where current networking technology cannot reach [27].

To realize the DTN vision, routes must be found over multiple unreliable, intermittently connected hops. Intermittent connectivity is defined as a sudden change in the state (up/down) of any communication link between nodes. Determine routes through the network without ever having an end-to-end connection, or even knowing which "routers" will be connected at any given time, is a critical challenge for DTNs. DTNs, like the postal network, use temporary connections to relay data rather than relying on end-to-end network connectivity. These networks could be useful for everything from connecting sensors to connecting remote regions of the world. DTN is appropriate for all networks with significant latency. On the other hand, DTN uses scarce resources while ensuring delivery in a terrestrial and heterogeneous environment.

DTN can prevent data loss by employing a store-carry-forward strategy. In DTN, each node sends a message using a store-carry-forward technique. When a node receives a message, it stores and carries it when it moves. The message is then forwarded to other relay nodes or a destination node. Message forwarding based on hop-by-hop routing decisions is more practical than finding a fully connected end-to-end path. The key issue is thus how to make effective forwarding decisions in order to ensure that messages are carried by relays that have the best chance of reaching their destinations. DTN is used in difficult environments such as military battlefields, deep under water communication, natural disaster affected areas, remote area social networking etc.

In other traditional networks if the communication link goes down at any point of time of communication then the data loss is assured but in DTN, if the communication is in down then the data get stored in the sender/receiver node's buffer rather being lost. This is because, in DTN the message forwarding is done by replicating the message to other nodes i.e. the sender and receiver both the nodes can have the copy of the same message stored in their buffer. A node in these networks can send data to another if they are within transmission range of each other. Because these networks are active, there is no guarantee that a direct connected path from a given source to a given destination exists at any time.

DTN routing enables communication in constrained environments by replicating messages from source nodes to relay nodes and delivering data to the destination by one of the nodes proposed to increase the message delivery ratio over such intermittently connected networks as Epidemic, Prophet, Spray, and Wait.

Prophet Protocol

PROPHET [9] (Probabilistic ROuting Protocol using Hisory of Encounters and Transitivity) is used in the proposed method which is a message routing protocol proposed for DTNs. The basic idea of PROPHET is to determine the next node on the delivery route of a message with the notion of delivery predictability of the nodes, where for each message transmission, the receiver of the message must be within the transmission radius of the transmitting node to successfully forward the message to the next node. Let $P_{a,b}$ denote the delivery predictability of node a concerned with destination b, where it should be noted that it is not an attribute of messages, but an attribute of nodes. In PROPHET, the value of $P_{a,b}$ is dynamically updated according to the following three types of events. At first, when node a encounter with node b, it is updated as

$$P_{a,b} = P_{a,b} + (1 - P_{a,b})\alpha \qquad (1)$$

where α is a parameter which is typically fixed to 0.75 [9].

Since the second term in the formula is positive, and it holds $P_{a,b} < 1.0$ for any $0 < \alpha < 1$, the value of $P_{a,b}$ geometrical approaches to 1.0 as repeating encounters. Next, if a does not encounter with b for a certain time period, $P_{a,b}$ is updated as

$$P_{a,b} = P_{a,b} \times \gamma \qquad (2)$$

where γ is a parameter called aging which is typically set to 0.98 [9].

Note that aging can occur several times for a long duration of time; namely, it gradually converges to zero. The above two update rules merely consider the direct encounter of a and b, but in actual mobile communication, the message delivery is realized by repeating message transfer through intermediate nodes, since even if a does not meet b in person, it is possible to hand a message to b through a common friend c. The third update rule reflects the transitive nature of such data transfer. Specifically, when node a encounters with node c($\neq$ b), $P_{a,b}$ is updated as

$$P_{a,b} = P_{a,b} + (1 - P_{a,b} \times P_{a,c} \times P_{c,b}) \times \beta \qquad (3)$$

where β is an appropriate parameter.

In PRoPHET, as shown in Figure 1.1, knowledge obtained from past encounters with other nodes is used to optimize both packet delivery and delivery performance for forecasting future contacts and determining the next suitable hops for a given packet. The performance evaluation of such protocols in terms of message delivery ratio and probability of deliverance is a difficult task due

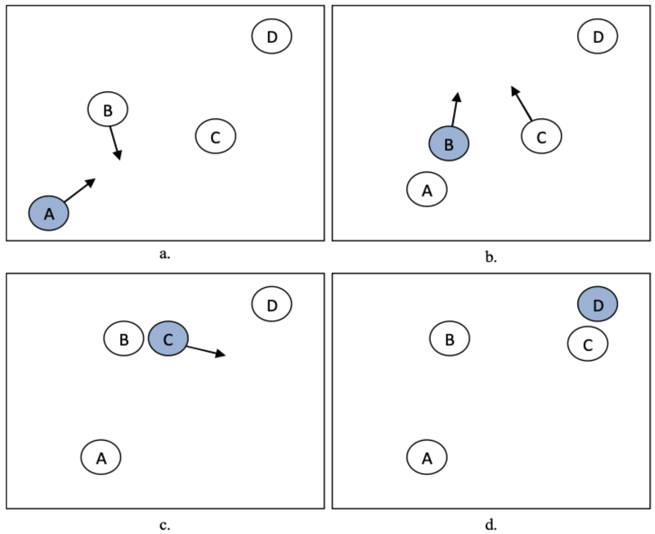

Figure 1.1: Node Encounters in prophet

to the complexity to drive mobile network simulations. Several efforts have been made to use simulations to evaluate the performance of routing schemes. The ONE simulator is now a reference tool in this field, having been approved by Helsinki University of Technology, particularly the Networking Laboratory.

This work describes the Opportunistic Networking Environment (ONE) simulator, which was created specifically for testing DTN routing and application protocols. It enables users to create scenarios based on various synthetic movement models and real-world traces, and it includes a framework for implementing routing and application protocols (already including six well-known routing protocols). Experiment evaluation is supported by interactive visualization and post-processing tools, and an emulation mode allows the ONE simulator to become a part of a real-world DTN testbed [19]. The ONE simulator's primary functions are to model node movement, inter-node contacts, routing, and message handling. Visualization, reports, and post-processing tools are used to collect and analyze results [20].

Chapter 2

RELATED WORKS

Existing adhoc routing schemes are mainly concerned with finding a path from source to destination. They are typically classified into three categories based on their concepts: 1) based on distance vectors; 2) based on link states; and 3) based on demand. The first two types are directly derived from wired network routing protocols because they typically consume a significant amount of resources and these algorithms become infeasible as the size of the network grows. On-demand protocols are introduced mainly to reduce maintenance costs in mobile environments by querying routines reactively when users require them. However, if there is no connection at the time of querying, communication fails. Several routing algorithms have been proposed, resulting in various types of DTN routing.

Priority-enhanced PROPHET [1] prioritizes messages exchanged among nodes so as to allow urgent messages to reach their destination faster, while it is still realized in the best effort manner. The urgency of messages is acquired through natural language processing, and an important use case of this extension is the information sharing in an area suffer from disaster. In the original PROPHET, each relay node deletes message contained in the buffer after passing enough time (e.g., few minutes) so as to guarantee the reach of the message to the destination.

There are many previous works concerned with the selection policy of the next node. Energy is a very important resource in battery operated mobile devices and available free buffer is very important in nodes in DTNs as they have to store message. PRoPHET does not take into account node energy consumption. Reducing node energy consumption and thus increasing network life time, as well as increasing message delivery probability in DTN, is an important issue to be considered. Since it has been considered in EA-PROPHET, which is an Energy Aware Prophet [2], it takes into account the buffer availability and the lifetime of the node's battery in addition to the delivery predictability to the destination.

The majority of previous works used the PROPHET protocol but did not provide an efficient buffer management strategy, and works dealing with buffer management strategies fail to take into account for all types of changes that occur in a DTN network that cannot be predicted in advance. Wang et al. [14] proposed a technique to increase the size of free space in the buffer by forcing the destination of messages to explicitly returns an ACK message to relay nodes. PROPHET+ [6] considers the buffer size, battery capacity, node location, and the node popularity were used to create a weighted function. The consideration of the battery capacity is also done in [3].

PROPHET-CLN [15] consider the congestion level of nodes, which could be detected by monitoring the free space of the buffer. Di PROPHET (Distance-based PROPHET) [13] considers the physical distance to the destination by adopting a cross-layer approach to acquire the RSSI value of received messages. PROPHETv2 [8] consider the encounter intervals in updating the delivery predictability. This allows us to effectively eliminate misleading cases in which a node repeatedly encounters with another node in a very short time period as in the car movement in a parking area. Finally, [16] reduce the variance of jitter.

In order to improve the Prophet protocol's message delivery rate even further, a new weighted metric for message forwarder selection was introduced in [26], which takes into account both the contact information of immediate encounters and the two-hop neighbors. Based on the forwarder selection strategy, an improved probabilistic routing algorithm (IPRA) with a higher chance of delivering messages to the destinations has also been proposed. When compared to existing schemes, the results show that can explore the benefits of two-hop contact information to select a more suitable forwarder and has the best performance of average delivery rate while achieving a better or comparable performance in terms of average overhead, average delay, and total energy consumption.

Epidemic algorithms guarantee that if a sufficient number of random data exchanges occur, all nodes will eventually receive all messages. In epidemic routing, a number of duplicate messages are sent in each hop so that at least one reaches the destination. Epidemic Routing is relatively simple because it requires no network knowledge. This provides a high level of redundancy while ensuring that all nodes receive every message, making this strategy extremely resistant to node and network failures. It tries every possible path and delivers each message in the shortest amount of time. The disadvantage of epidemic routing is that it requires a large amount of buffer space, bandwidth, and power to deliver multiple copies of the message, which increases the overhead due to the large consumption of bandwidth. Many previous works on the Epidemic Routing Protocol have been proposed which can be good comparison to the Prophet Routing Protocol have been published.

An enhanced version of the Epidemic Routing Protocol [23] is deployed, allowing the node to consider battery energy level and buffer space when making routing decisions, increasing message delivery ratio, and decreasing network traffic. To reduce message delivery delays, a cluster movement model with coordinates restricted to a circular area defined by a central point and range was used. The minimum estimated expected delay (MEED) path metric [17], which estimates waiting times for each contact based on observed data and demonstrated an epidemic protocol for propagating topology updates in a delay-tolerant network.

A new routing metric called Expected Path Length [22] guides messages to the nodes with the shortest expected path, reducing the number of unnecessary message copies while increasing message delivery rate. An improved opportunistic routing protocol [25] was used, in which the context information of average distance travelled, and average time elapsed from message reception to delivery to the destination node was used. When varying buffer size, message generation interval, and number of nodes, the average distance and average time were updated whenever a message was delivered to a destination node and had better performance on delivery ratio, overhead ratio, and delivery latency.

Chapter 3

NETWORK MODEL

There are various techniques for describing and understanding data communications networks. To transfer a message from sender to receiver, all networks provide the same basic functions. A network model is a database model that is intended to describe objects and their relationships in a flexible manner. The network model's structure, which is represented as a graph with relationship types as lines and object types as nodes, is an unique characteristic. Apart from other database models, the network model's structure is not limited to a lattice or hierarchy; instead, a graph replaces the hierarchical tree, allowing for more basic relationships with the nodes.

In the proposed method, let V be a set of fixed-location stationary nodes and U be a set of mobile nodes that can move freely in a given region. As long as their locations are close enough, stationary and mobile nodes can directly communicate with each other via short-range wireless communication such as Wi-Fi and Bluetooth (e.g., few meters). The time required for message transmission is negligibly short, and the moving speed of mobile nodes is considered to be constant for simplicity. A message sent from node i to node j is picked up by a mobile node traveling near i followed by the mobile node, and dropped near j.

Each message sent from a source node can go through other stationary nodes and change to other mobile nodes along the way to the destination. This is accomplished by repeating the store-and-forward operation, in which each node has a local buffer large enough to store all incoming messages, so no need to worry about the message drops due to buffer full. Each link becomes unavailable if the distance between two incident nodes exceeds the BLE transmission radius (e.g., 20m), and such intermittent disconnection occurs with a particular probability distribution. In this work, we consider the problem of sending a given message to its destination with as small maximum delivery time as possible.

3.1. Contact Possibility Graph

In this work, we assume that any two stationary nodes are located far apart so that the direct communication between them is not possible, and the contact of two mobile nodes could not be predicted beforehand, although it enables the direct communication between them. Each mobile node v has a unique range of activity (e.g., around the residence for retired person and between residence and school for primary school children), and let R(v) denote the set of stationary nodes contained in the activity range of v.

Such a possibility of contacts between stationary and mobile nodes is modeled by a bipartite graph G = (A, B, E) with vertex set A B and edge set E so that: u $\in$ A and v $\in$ B are connected by an edge in E iff u $\in$ R(v). In the following, we call such G the contact possibility graph for the given wireless network which is shown in figure 3.1.

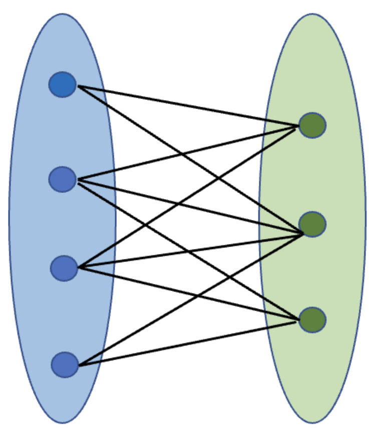

Figure 3.1: Bipartite Graph

Chapter 4

PROPOSED METHOD

4.1. Directed Bipartite Graph with Edge Weight

We are concerned about the issue of routing a message as quickly as possible from source node s to destination node t in this study. Such message routing can be understood as the process of forwarding messages to destination t by changing mobile nodes as needed. If the contact possibility graph G is explicitly defined, we may realize the minimal hop count routing to t by having each node forward the received message in the direction that reduces the minimum hop count to t (similar to AODV).

On the other hand, if we want to reduce the actual latency of message delivery (rather than the number of hops), we should consider the waiting time for a mobile node to arrive at a particular stationary node as well as the travel time to the next stationary node from the given stationary node. Although a generic integer programming solver might be used to calculate an optimal scheduling based on complete understanding of the behavior of individual mobile nodes.

We would prefer to take a macroscopic approach in this work and consider it as a stochastic process, as this could better reflect changes in travel routes and travel times (remind that primary school children often take a detour even if the school dismisses at a fixed time). To express the above two characteristics (i.e., waiting time and travel time), we shall extend G to a directed graph G as follows:

1. For each v ∈ B, replace mobile node v by a collection of $|R(v)|$ virtual nodes so that each virtual node corresponds to a stationary node u ∈ R(v).

2. For each v ∈ B and u ∈ R(v), connect u and all virtual nodes w derived from v by a directed edge (u, w) with the weight corresponding to the expected waiting time for v at node u.

3. For each v ∈ B and u, u' ∈ R(v), connect virtual node w corresponding to u' and stationary node u by a directed edge (w,u) with the weight corresponding to the expected travel time from u' to u.

We then use an AODV-like technique to ensure message delivery to the specified destination along a path with the minimum expected latency (note that this calculation depends on the linearity of expectation).

Theorem (linearity of expectation):

A bipartite graph is a graph whose vertices can be divided into two disjoint sets V1 and V2 such that every edge connects a vertex in V1 to one in V2. Every graph G = (V,E) has a bipartite sub-graph with at least $|E|/2$ edges.

Proof:

1. We construct a random sample space by choosing for every vertex in which set (V1 or V2) it belongs at random, equiprobably for the two sets and independently for each vertex. Thus, the points are random "bipartitions" of V .

2. We define the r.v. X that corresponds to the number of "crossing" edges (joining vertices in different parts).

3. Let g be an edge. We have that

$$X = \Sigma_{g \in E(G)} X_g$$

where:

$$X_g = \begin{cases} 1 & g\ is\ crossing \\ 0 & otherwise \end{cases}$$

4. We have,

$$E[X_g] = Pr\{X_g = 1\} = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = 2 \cdot \frac{1}{4} = \frac{1}{2}$$

5. By linearity of Expectation,

$$E[X] = E(\Sigma_{g \in E(G)} X_g) = \Sigma_{g \in E(G)} E[X_g] = |E| \cdot \frac{1}{2}$$

6. Thus, there must exist a bipartite sub-graph which has at least $|E|/2$ edges.

4.2. Estimation of Edge Weight

The remaining task is to estimate the expected waiting and travel times as accurately as possible. The followings are hints to improve the accuracy of estimation.

1. In general, the sample size is very small.

2. If two mobile nodes have similar range of activities, we could merge them into a single mobile nodes with more sample data and shorter waiting time.

Chapter 5

DISTRIBUTED IMPLEMENTATION

Different networks have special individual needs and different routing protocols have been designed to meet the individual needs of the networks. Network routing protocols can be roughly divided into two categories called distance-vector type and link-state type.

5.1. Distance Vector Routing Protocols

Distance Vector routing protocols make decisions based on the shortest path to a particular destination. Distance is usually measured in hops, however the distance metric could be latency, packet loss, or something else. If the distance metric is hop, then a hop is regarded to have been traversed each time a packet passes through a router. The route with the least hops to a given network is determined to be the best route to that network. The vector indicates the direction to that specific network. Distance vector protocols send their entire routing table to neighbors that are directly connected.

5.2. Link State Routing Protocols

Shortest-path-first protocols are another name for link state protocols. Link state routing protocols have an accurate representation of the network structure. As a result, they are more knowledgeable about the entire network than any distance vector protocol. Each link state routing enabled router generates three distinct tables. One table stores information on directly connected neighbors, another stores the topology of the complete inter network, and the last one stores the actual routing table. All routers in the network receive information about directly connected links using link state protocols.

The Dijkstra's algorithm is appropriate for message routing in the link-state protocol, in which link weights are collected to a specific location, such as a server, and each source finds the route of a message based on them. However, in general DTNs which are the target of PROPHET, the state of links changes dynamically, so that it often changes even during aggregation, or an efficient aggregation itself is difficult.

As a result, in this study, we will consider distance-vector message routing, in which the next node on the delivery route is adaptively selected by each relay node by referring to the routing table kept locally by each node. In a wireless network, one commonly used protocol for maintaining such a routing table is AODV. The proposed method obtains the weight of each link in a distributed manner similar to AODV, as follows:

• Based on local observations, each node computes the link encounter probability p for each neighbor and takes the inverse of the probability as the link weight (if we want to account for message transfer time, we can increase the link weight by that amount. Please keep in mind that time is a dimension of weight).

• The source node floods request packets to each destination and fills table entries by aggregating the destination's reply packets (or its neighboring nodes). We can try to leave the next hop on the route with the lowest cumulative weight during the aggregation.

5.3. AD-HOC ON-DEMAND VECTOR (AODV)

The proposed method uses AODV (Adhoc On-Demand Distance Vector) to find a route from the source to the final destination. AODV is a reactive routing protocol designed for mobile ad-hoc network (MANETs) [21].

• Routing protocols in mobile networks are subdivided into two basic classes. Proactive routing protocols (e.g. OLSR) are table-driven. They usually use link-state routing algorithms flooding the link information. Link-state algorithms maintain a full or partial copy of the network topology and costs for all known links.

• The reactive routing protocols (e.g. AODV) create and maintain routes only if these are needed, on demand. They usually use distance-vector routing algorithms that keep only information about next hops to adjacent neighbors and costs for paths to all known destinations. Thus, link-state routing algorithms are more reliable, less bandwidth-intensive, but also more complex and compute- and memory-intensive.

• In on-demand routing protocols a fundamental requirement for connectivity is to discover routes to a node via flooding of request messages. The AODV routing protocol is one of several published reactive routing protocols for mobile ad-hoc networks, and is currently extensively researched.

• The Bellman-Ford distant vector algorithm is related to the AODV algorithm, which has been modified to operate in a mobile setting. Only when a node wishes to send a packet to a particular destination does AODV decide on a route to that location. Routes are kept up for as long as the source requires them. Sequence numbers guarantee loop-free routing and the freshness of routes.

5.3.1. Routing tables

The destination, next hop, number of hops, destination sequence number, active neighbors for this route, and the expiration period for this route table entry are all details that are contained in each routing table entry. Every time the route is utilized, the expiration time, also known as lifetime, is reset. The new expiration time is the sum of the current time and a parameter called active route timeout. The duration after which the route is considered invalid and the nodes not on the route indicated by RREPs delete their reverse entries is determined by this parameter, also known as the route caching timeout. Route repairs will maintain routes if the active route timeout is significant enough. The route discovery process is shown in figure 5.1.

5.3.2. Control messages

5.3.2.1. Routing request

• A route request packet (RREQ) floods throughout the network when a route is not available for the destination. The following fields are included in the RREQ:

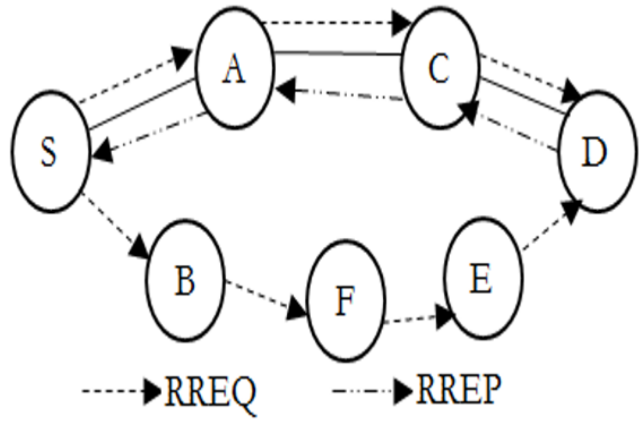

Figure 5.1: Route discovery process

• The pair (source address, request ID) uniquely identifies an RREQ since the request ID is increased each time the source node delivers a new RREQ. Each node examines the source address and the request ID upon receiving an RREQ message. The new RREQ packet will be discarded if the node has already received an RREQ with the same pair of parameters. If not, an RREP message will be sent in response to the RREQ, either broadcast or unicast.

• The RREQ will be rebroadcasted with an increased hop count if the node does not have a route entry for the destination or if it has but it is no longer an updated route. A RREP message will be generated and sent back to the source if the node has a route with a sequence number greater than or equal to that of RREQ. The number of RREQ messages that a node can send per second is limited.

• When flooding RREQ messages, AODV is optimized using the expanding ring (ESR) method. The time to live (TTL) value that each RREQ carries indicates how many times this message should be re-broadcast. When no responses are received, retransmissions take place. Historically, such floodings used a TTL large enough to reach every node in the network and ensure successful route discovery in just one round of flooding. This TTL was larger than the network's diameter. This low delay time method, however, results in excessive overhead and pointless broadcast messages.

• Later, it was demonstrated that the minimal cost flooding search problem may be solved by a sequence of flooding with a well selected set of TTLs.

...	Source address	Destination address	Destination sequence no	Hop count	Next hop IP address	Expire time	...

Figure 5.2: Route Request Packet

5.3.2.2. Routing response

• A route reply message (RREP) is unicast back to the source if a node is the destination or has a valid route to the destination. RouteReply packet records the number of hops to the destination and is delivered to the originator of the corresponding RouteRequest packet along the route of the packet in the reverse direction.

• After receiving RouteReply packets from neighbors, the node selects the route with the smallest number of hops and updates the routing table so that the next node along the selected route is recorded for each destination. The format of this message is as follows:

...	Source address	Destination address	Destination sequence no	Hop count	...

Figure 5.3: Route Reply Packet

To implement the the proposed method, we added some fields in the routing table of AODV, such as ARRIVAL TIME and RESPONSE TIME in the RREQ message shown in figure 5.4. The AODV selects the route according to the routes that calculate the RESPONSE TIME in each routing table of the nodes.

....	Source address	Destination address	Destination Sequence no	Hop count	Next hop IP address	Expire Time	**Time arrival**	**Time response**	....

Figure 5.4: Proposed Route Request Packet

A node that receives a RREQ must fill the ARRIVAL TIME field to determine when the message arrives at the node. Each node updates its routing table along the paths and stores the response time information in the routing table to ensure that the destination node can find the fastest route within a short time. Also, the MAX RESPONSE TIME field is added in the RREP message shown in figure 5.5.

....	Source address	Destination address	Destination sequence no	Hop count	**Time response**	....

Figure 5.5: Proposed Route Reply Packet

To calculate the RESPONSE TIME, the response time between each node and its neighbors should be calculated first, and the results should be recorded in the node's routing table. In route discovery process, when the RESPONSE TIME is greater than MAX RESPONSE TIME, it will update the routing table and the destination will receive the RREQ packets. If RESPONSE TIME is lesser than MAX RESPONSE TIME, then directly the destination will receive the RREQ packets. Then, the destination selects a route with minimum RESPONSE TIME and sends unicast RREP packet towards the source.

Chapter 6

THE ONE SIMULATOR

ONE is an agent-based discrete event simulation engine at its core. The engine updates a number of modules that implement the primary simulation functionalities at each simulation step. The ONE simulator's main characteristics are to represent node movement, inter-node connections, routing, and message handling. Visualization, reporting, and post-processing tools are used to collect and analyze results. Figure 6.1 represents the elements and their interactions which is the overview of the ONE simulation environment [20].

Report modules receive events (such as message or connectivity events) from the simulation engine and generate findings based on them. The output can be event logs that are then analyzed by external post-processing tools, or aggregate statistics calculated in the simulator. Second, the graphical user interface (GUI) displays a visualization of the simulation state, including the nodes' positions, active contacts, and messages.

The simulations can include any number of various sorts of agents, such as wireless nodes. The nodes are organized into node groups, and each group has a set of common parameters such as message buffer capacity, radio range, and mobility model. Because separate groups can have different configurations, creating a simulation including pedestrians, cars, and public transportation is possible. All movement models, report modules, routing algorithms, and event generators are dynamically loaded into the simulator, making it simple for users and developers to extend and configure the simulator with various types of plugins: simply creating a new class and defining its name in the configuration file is usually sufficient [28].

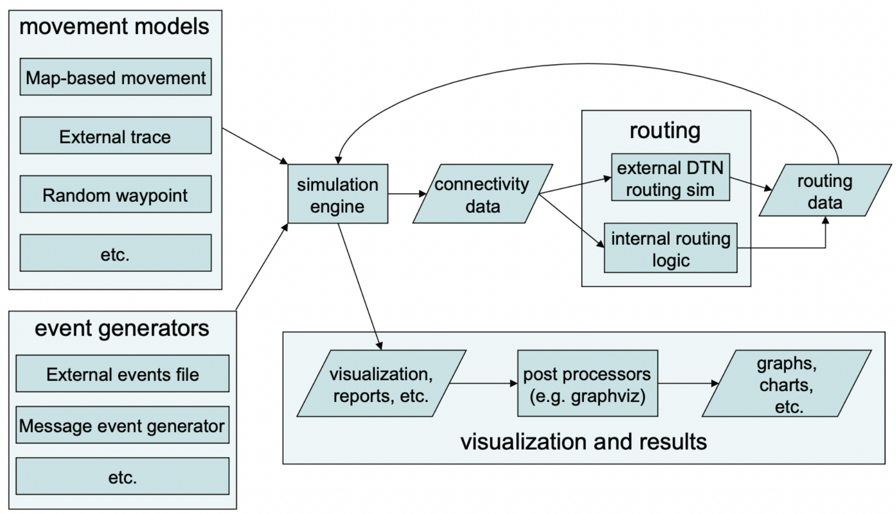

Figure 6.1: Overview of the ONE simulation environment

6.1. Application Support

The ONE simulator provides two methods for generating application messages within the simulation: message generators and external event files. Messages can be unidirectional or create responses when received, similar to a request-response application. Messages may also include application-specific information via generic (name, value) pairs connected to them.

Messages having a random or predetermined source, destination, size, and interval are generated by the built-in message generator. There is also a separate tool for creating message event files. In simulations, any number of such message event sources can be used concurrently. Messages are either unidirectional or tagged to expect a response, with the response size controlled separately. Application-specific headers and payloads can be attached to messages, and nodes can be extended to handle checking message headers and contents along the route, enabling application aware forwarding, such as for content distribution.

6.2. Reporting and Visualization

The ONE may view simulation results in two ways: through an interactive Graphical User Interface (GUI) and by generating images from the data obtained during the simulation. The main window displays node locations, current paths, connections between nodes, the number of messages carried by a node, and so on. If a map-based movement model is used, all map paths are displayed. If available, an extra background image (e.g., a raster map or a satellite image of the simulation area) is displayed below the map pathways. The view enables for zooming and interactive simulation speed adjustment.

The GUI produces a filtered log of simulation events, such as contacts and message transfers. Filters are used to show only interesting events, or to pause the simulation when a particular type of event occurs. Selecting a node from a list or a log message opens it for closer inspection. This enables for the retrieval of additional information about the messages carried by a node as well as the state of the routing module. While the GUI provides an intuitive general image of what is happening during the simulation, post-processed report files give more rigorous ways to visualize node relations, message pathways, and performance statistics. The simulator provides a message statistics report module that collects overall performance information (amount of created messages, message delivery ratio, how long messages stay in node buffers, etc.).

6.3. Routing simulation

While mobility models determine where nodes should move next, routing modules determine where messages, or bundles, end up. The ONE includes six implementations of well-known routing algorithms, as well as a passive routing module for interacting with external DTN routing simulators. The ONE includes the following active routing modules: First Contact, Direct Delivery, Spray and Wait (normal and binary), Epidemic, PRoPHET, and MaxProp.

When two (or more) nodes meet and there is an opportunity to exchange messages, all routing modules check to see if they have any messages destined for the other node and try to send them. If the node has already received the message, it declines to receive it, and alternative messages can be tried. After all, if such messages are exchanged, the routing algorithm determines how the rest of the messages behave [28].

The simplest routing modules are the Direct Delivery, Epidemic and First Contact. After exchanging deliverable messages, the Direct Delivery module does not initiate any more transactions since it will only transmit messages if it is in contact with the final recipient. This reduces buffer space and bandwidth, but it is obviously not the best approach in many circumstances if a high message delivery probability is needed.

The Epidemic routing module has a different method in that after two nodes exchange deliverable messages, it attempts to exchange all other messages until both nodes have the identical set of messages or the connection breaks. If we had unlimited buffer space and bandwidth, we would have maximum message distribution throughout the nodes and thus maximum delivery probabilities. However, if buffer space and/or bandwidth are restricted, Epidemic routing is likely to waste a significant amount of resources. For example, a message that was delivered quickly after it was created could be relayed and retained by all nodes for a long period of time even if it is no longer needed. In this manner, the message consumes resources that could be better used for messages that have yet to be delivered.

While the First Contact module also transmits as many messages as it can to the other node, it deletes the local copy of the message after a successful transfer. As a result, every message in the network have only one copy. To avoid two nodes that have been in contact for a long period from exchanging the same messages, the receiving node receives a message only if it has not previously passed through it. Unfortunately, there are no guarantees that the first node encountered is a stronger candidate than the last node carrying the message, hence First Contact is unlikely to reach extremely high delivery probabilities.

The Spray and Wait routing module works similarly to the Epidemic but is a bit more complex because it limits the number of copies disseminated across the network. This is accomplished by allowing each produced message to replicate just a certain number of times. A node with more than one copy of the message can give one copy (the regular mode) or half of the copies to another node (the binary mode). If the node only has one copy of the message left, it is only sent to the final recipient. Spray and Wait can balance high message dissemination and high resource utilization by using a varying number of initial copies.

PRoPHET and MaxProp increase the complexity by keeping track of which nodes have been in contact with which nodes. This information can be used to determine whether a particular node is a good candidate for delivering a message to another node, based on the assumption that if two nodes have previously met, they are more likely to meet (soon) again. While PRoPHET determines whether another node is more likely to meet the final recipient, MaxProp extends this concept by using Dijkstra's method to calculate entire paths from node to node based on meeting probabilities. MaxProp also use acknowledgments of delivered messages to aid in the removal of redundant messages from the network.

The passive routing module can connect to other DTN routing simulators. If a DTN routing simulator can generate timestamped data regarding message-related events (creating, relaying, and removing messages), this information can be input into ONE for analysis and visualization. For example, the debug trace from dtnsim2 can be transformed to ONE-compatible format. If the ONE is used to create the contact schedule for dtnsim2's input, the message routing and mobility modeling can be displayed and inspected in the GUI in the same way that the routing was done by the ONE.

6.4. Simulation setup

The simulation scenario is taken from a real map of Helsinki downtown area covering a 4500 m × 3400 m region with different numbers of nodes, which is developed by the University of Helsinki to investigate the Opportunistic Networks in real life. The map information includes streets, shops, parks, bus stops, and trams of Helsinki. All nodes are divided into 6 groups. A node group has a common set of simulation parameters like moving speed and pause time distribution, data transmission rate, message buffer size, and communication range.

Parameters	**Values**
Simulation time	43200 sec
Interface	Bluetooth
Interface type	Simple broadcast
Transmit speed	2 Mbps
Transmit range	10m
Movement model	Shortest Path Map Based Movement
Buffer size	5 MB
Total no. of nodes	40
Speed of nodes	0.5 - 1.5 m/s
Message size	500 KB to 1 MB
Message interval	25 to 35 sec
Message TTL	300 minutes

Table 6.1: Parameters and their values

In the scenario, Group 1 and Group 3 are pedestrian groups; Group 2 is automobile group; Group 4, Group 5, and Group 6 are trolleybus groups. The pedestrians and cars randomly choose destinations in their reach on the map and follow the shortest path to the destination. The trolley buses follow predetermined routes to match trolleybus routes. With the exception of Group 4, other groups use Bluetooth devices. IEEE 802.11b WLAN devices have been used by Group 4. The source and destination for each message are randomly chosen from all nodes. The message size varies from 500 Kb to 1 MB, and the message lifetime is set to 18000 seconds. The duration of simulation is 43200 seconds. The default setting parameters and their values is described in the table 6.1.

Chapter 7

EXPERIMENTS

For performance analysis, Opportunistic Network Environment (ONE) simulator is used to analyze the performance of delivery predictability, overhead ratio, and delivery latency.

Experiment 1

The bipartite graph in figure 3.1 represents the possibility of contacts between stationary and mobile nodes is modeled by a bipartite graph G = (A,B,E) with vertex set AB and edge set E so that: $u \in A$ and $v \in B$ are connected by an edge in E iff $u \in R(v)$, which is called as the Contact Possibility Graph G for the given wireless network.

We can realize the minimum hop count routing to d using the contact possibility graph G by each node forwarding the received message in the direction that reduces the minimum hop count to d (like AODV). The output shown in figure 7.1 represents the Request as [Source, Source Sequence, ID, Destination, Destination Sequence, Hop count] and the Response as [Source, Destination, Destination Sequence, Hop count].

Experiment 2

In AODV, when TTL reduced to 0, the RREQ packet should be dropped even the destination could not be found. So, this may lead to unwanted packet loss and will obtain less delivery predictability.

To solve this, the TTL has been dynamically reduced according to the node speed if the speed exceeds the threshold value in AODV. So that, it reaches 0 quickly when the packet passes through some of the fast nodes. By this, we can find stable path with less hop count and less delay. The experiments have done with including the following conditions,

Condition 1: If the source S sends RREQ towards the destination D, the forwarding node will check for the speed of the node, if speed exceeds the threshold value it will reduce speed value from the TTL.

Condition 2: If the forwarding node speed does not exceed the threshold value it will reduce TTL value by 1 by general flooding rule.

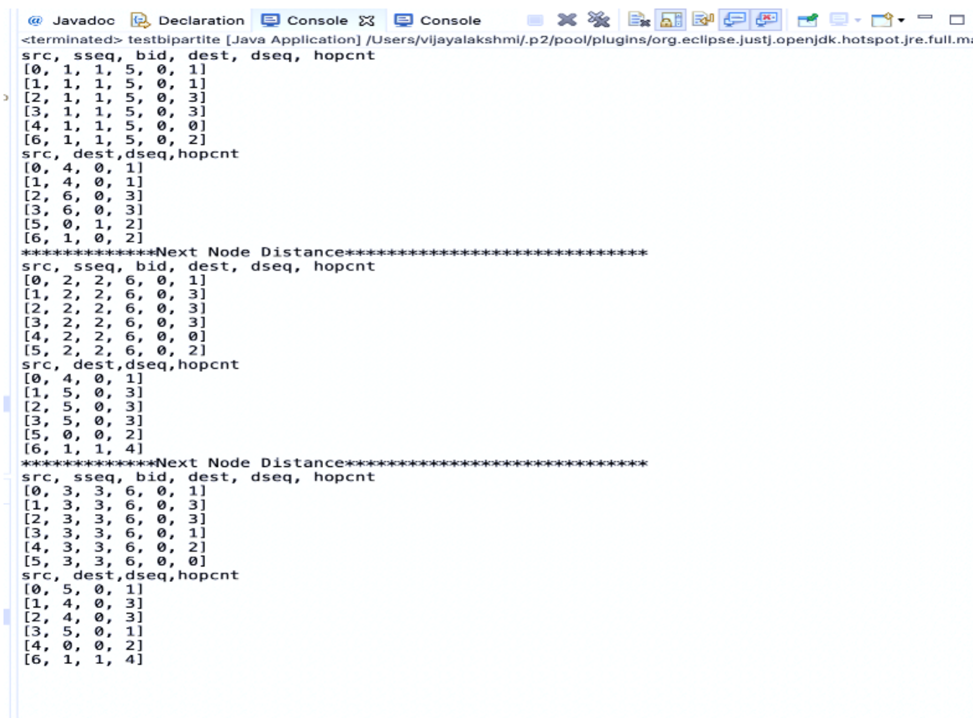

Figure 7.1: The output shows that the bipartite graph is linked with AODV and sends Requests and receives responses for all node combinations in the given graph.

Chapter 8

CONCLUSION

By reducing the number of hops in the network, we presented an improved PRoPHET routing protocol for DTNs. In the proposed method, we consider using the expected number of hops to the destination as the edge weight where the previous work focused on delivery predictability as the edge weight. The proposed method which could make a significant improvement for the AODV protocol and despite of having a large number of hops in the routing table, the proposed protocol selects a path and sends a response in a short amount of time. It improves the performance of original AODV and meets the route selection requirements. However, the results are expected in the sense that AODV in the proposed method would help to select the paths based on the minimum response time in each routing table of the nodes with the least number of hops, which could be more efficient than the previous work. In terms of message delivery and overhead ratio, the enhanced PRoPHET outperforms the original PRoPHET.

BIBLIOGRAPHY

[1] S. Bhattacharjee, S. Basu, S. Roy, and S. D. Bit. “Best-effort delivery of emergency messages in post-disaster scenario with content-based filtering and Priority-enhanced PROPHET over DTN.” in Proc. 8th Int’l Conf. on Communication Systems and Networks (COMSNETS), 2016.

[2] B. B. Bista and D. B. Rawat. “EA-PROPHET: An Energy Aware PROPHET-Based Routing Protocol for Delay Tolerant Networks.” In Proc. 31st Int’l Conf. on Advanced Information Networking and Applications (AINA), IEEE, 2017, pages 670–677.

[3] S. J. Borah, S. K. Dhurandher, S. Tibarewala, I. Woungang, and M.S. Obaidat. “Energy-Efficient PROPHET-PRoWait-EDR Protocols for Opportunistic Networks.” in Proc. GLOBECOM, IEEE, 2017.

[4] T. W. Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. in Proc. IEEE Int’l Conf. on Communications, 1998. p. 171–5.

[5] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann. The optimized link state routing protocol, evaluation through experiments and simulation. in Proc. IEEE Symp. on wireless personal mobile communications, 2001.

[6] T.-K. Huang, C.-K. Lee, and L.-J. Chen. “PROPHET+: An Adaptive PROPHET-Based Routing Protocol for Opportunistic Network.” In Proc. 24th Int’l Conf. on Advanced Information Networking and Applications (AINA), 2010, pages 112–119.

[7] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. Mobile Computing, 1996:153–81.

[8] H.-J. Lee, J.-C. Nam, W.-K. Seo, Y.-Z. Cho, and S.-H. Lee. "Enhanced PROPHET routing protocol that considers contact duration in DTNs." in Proc. Int'l Conf. on Information Networking (ICOIN), 2015.

[9] A. Lindgren, A. Doria, and O. Schelen. "Probabilistic Routing in Intermittently Connected Networks." in Dini P., Lorenz P., de Souza J.N. (eds) Service Assurance with Partial and Intermittent Resources. SAPIR 2004, Lecture Notes in Computer Science, vol 3126. Springer, Berlin, Heidelberg.

[10] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. Mobile Networks and Applications. 1:183–97, 1996.

[11] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. ACM SIGCOMM Computer Communication Review, 24:234–44, 1994.

[12] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. in Proc. 2nd IEEE Workshop on mobile computing systems and applications (WMCSA'99), 1999. pages 90–100.

[13] P. Sok, S. Tan, and K. Kim. "PROPHET Routing Protocol Based on Neighbor Node Distance Using a Community Mobility Model in Delay Tolerant Networks." in Proc.10th Int'l Conf. on High Performance Computing and Communications and Int'l Conf. on Embedded and Ubiquitous Computing, IEEE, 2013.

[14] P. Wang and F. Shen. "Method to Improve the Performance of PROPHET Routing Protocol." in Proc. 2nd Int'l Workshop on Education Technology and Computer Science, 2010.

[15] G.Wang,J.Tao,H.Zhang,andD.Pan."A improved PROPHET routing based on congestion level of nodes in DTN." in Proc. 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), IEEE, 2017.

[16] J. Xue, J. Li, Y. Cao, and J. Fang. “Advanced PROPHET Routing in Delay Tolerant Network.” in Proc. Int’l Conf. on Communication Software and Networks, 2009.

[17] Evan P.C. Jones; Lily Li; Paul A.S. Ward. ”Practical Routing in Delay Tolerant Networks” in SIGCOMM’05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.

[18] K. Fall. ”A delay-tolerant network architecture for challenged internets.” In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (pp.27-34). ACM, 2003, August.

[19] Ari Keranen, Teemu Karkkainen, and Jorg Ott. ”Simulating Mobility and DTNs with the ONE” in Journal Of Communications, Vol. 5, No.2, February 2010.

[20] Ari Keranen, Teemu Karkkainen, and Jorg Ott.”The ONE Simulator for DTN Protocol Evaluation”.http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5674

[21] Prashant Kumar Maurya, Gaurav Sharma, Vaishali Sahu, Ashish Roberts, Mahendra Srivastava. “An Overview of AODV Routing Protocol “ in International Journal of Modern Engineering Research (IJMER) Vol.2, Issue.3, May-June 2012 pp-728-732.

[22] Kun Tan, Qian Zhang, Wenwu Zhu.”Shortest Path Routing in Partially Connected Ad Hoc Networks.” in IEEE GLOBECOM 2003.

[23] T. Ahmad, S. Irfanullah, W.U. khan. ”An Enhance Mechanism for PRoPHET ROUTING PROTOCOL to Achieve Qos in DTN” in ILMA Journal of Technology and Software Management - IJTSM Vol. 1 Issue.1, 2021.

[24] Prachi Garg, Hemang Kumar, Rahul Johari, Prabhanshu Gupta and Riya Bhatia. ”Enhanced Epidemic Routing Protocol in Delay Tolerant Networks”

in 5th International Conference on Signal Processing and Integrated Networks (SPIN), 2018.

[25] Kyung Min Baek, Dong Yeong Seo, Yun Won Chung. "An Improved Opportunistic Routing Protocol Based on Context Information of Mobile Nodes". Special Issue on Applied Sciences Based on and Related to Computer and Control, 2018.

[26] Xu Wang, Rongxi He, Bin Lin, and Ying Wang. "Probabilistic Routing Based on Two Hop Information in Delay/Disruption Tolerant Networks" in Hindawi Publishing Corporation Journal of Electrical and Computer Engineering Volume 2015, Article ID 918065, 11 pages http://dx.doi.org/10.1155/2015/918065

[27] "Mobile Ad-Hoc and DTN Networks at IPCAS Lab" online available: http://www.ece.gatech.edu/research/labs/WCCL/DTN2.html

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