Integration of Fixed and Mobile Infrastructure for Message Passing in Opportunistic Networks

Sanjay K. Dhurandher, Deepak K. Sharma, and Sahil Gupta
Netaji Subas Institute of Technology, University of Delhi, New Delhi, India
Email: dhurandher@rediffmail.com, dk.sharma1982@yahoo.com, sahilgupta221@gmail.com

Isaac Woungang
Department of Computer Science, Ryerson University, ON, Toronto, Canada
Email: iwoungan@ryerson.ca

Mohammad S. Obaidat
3Department of Computer and Information Science, Fordham University, Bronx, New York 10458, USA
Email: mobaidat@fordham.edu

Abstract—In an Opportunistic networks (OppNets), an established end-to-end connected route between the source and destination nodes may never exist, making the routing of messages a challenge. However, using some infrastructures, it is possible to create an OppNet architecture similar to that of a well-connected network. This paper focuses on combining the benefits of infostation and message ferry to design a novel routing scheme called Horizontal and Vertical ferry (HVF), which also includes an acknowledgment method for message delivery that has not been considered in any of the existing approaches that use infostation or message ferry. Through simulations, the performance of the proposed HVF scheme is evaluated and compared against that of the Global Ferry scheme (GFS), chosen as a benchmark, in terms of number of messages delivered, overhead ratio, average latency, and average buffer time, showing that HVF outperforms the GFS in both the normal and ferry failure scenarios.

Index Terms—Opportunistic Networks (OppNet); Infostation; Message Ferry; Infrastructure-Based Protocols; Horizontal and Vertical Ferry (HVF); Global Ferry Scheme (GFS); ONE Simulator

I. INTRODUCTION

OppNets [1] are the most recent evolutions of the traditional mobile ad-hoc networks (MANETs) [2], which are used to provide communication facilities among the heterogeneous devices in sparse network scenarios. In OppNets, nodes may be of different types and capacities. They may be fixed such as access points or mobile such as vehicles or pedestrians. In addition, they can communicate with each other by all kinds of communication media such as WiFi [3], Cellular [4], Bluetooth [5]. The nodes are often out of range of a network, but may be connected to other devices. These connections between the nodes are utilized for data delivery. To determine the usefulness of nodes, a protocol can use several parameters such as probability of meeting the destination, battery life, stability, to name a few, then figure out the best next hop for message transmission purpose. OppNets also exploit the data transmission capability of small handheld devices such as mobile phones and personal digital assistants (PDAs). These devices form an OppNet when they come in close proximity of other wireless devices to exchange the data. For example, when people travel in buses and trains, they come in contact with other passengers, and their mobile devices can utilize these contacts with others to exchange the information of their interests, such as a word document, a photograph, an MP3 file, to name a few.

OppNets are also considered as a subclass of Delay tolerant networks (DTNs) [6,7] since they have many characteristics that are similar to that of DTNs. Some of their most important characteristics are as follows: (a) The network is generally divided into several partitions (called regions), thus a complete path between sender and receiver rarely exists; (b) The network may experience frequent disconnections and re-connections due to the mobility of nodes, node failure, and node power saving efforts to conserve its energy; (c) Network contacts are intermittent, i.e. the contact opportunities and contact durations between the nodes are very few; (d) The link performance is highly variable due to the inconsistency and unstable behaviors of the wireless links; and (e) Intermediate nodes use a store-carry-and-forward technique for message passing. 

Traditional MANETs and Internet routing protocols fail to work in OppNets since they rely on the assumption that a complete source to destination path must be established prior to message passing, which is not the case of OppNets due to network partitions. The source node can choose any node from a group of neighbors as next hop, which has the highest probability of delivering the message closer to the destination, or to the destination itself. OppNets have already been implemented successfully in various projects such as ZebraNet [10] and SWIM [11], DakNet [12], Sammi Network Connectivity, interplanetary communications [16], to name a few.
Routing and forwarding are compelling tasks in OppNets due to the fact that they are based on the contact opportunity between nodes, which arises due to the mobility of nodes and the store-carry-and- forward technique used to pass the messages. Due to the problem of network partition, an intermediate node may not find a suitable node to forward the message towards the destination, thus, it may have to keep the message in its buffer for a longer period of time as long as there is no forwarding opportunity towards the destination. Nodes are required to have an adequate amount of buffer to store all the messages until they are transferred to the next suitable node. Thus, messages in OppNets can suffer longer delays and this is the reason why they are considered as delay-tolerant in nature.

The routing protocols in OppNets can be broadly classified into two categories: infrastructure-less and infrastructure-based protocols [17, 34]. Infrastructure-less protocols have been designed for flat ad-hoc networks. They make use of the contact opportunity between the nodes as well as the mobility of nodes to forward the messages. Nodes themselves can transfer the messages using suitable techniques. In the routing and forwarding process, no categorization in terms of priority or hierarchy is assigned to nodes in order to distinguish them. Examples of infrastructure-less protocols include: Epidemic Routing [18], Spray and Focus [19], HiBoP [20], HBPR[21, 22], GAER [23], SPRINT [24], ContentPlace [25], to name a few.

On the other hand, in infrastructure-based protocols, some form of infrastructure is exploited to opportunistically forward the messages to the destination. In this case, some reliable agents are designed for carrying the messages towards the destination. These agents are held responsible for efficient message delivery with minimum delivery delays and maximum delivery ratio. Examples of agents include infostations [26], message ferries [27], data MULEs [28], to name a few. In addition, infrastructure-based routing protocols can be further classified into fixed infrastructure and mobile infrastructure [17]. In the former, nodes are kept fixed at some geographical locations (e.g.: Infostations), while in the later, nodes can move around in the network, following either a fixed predetermined path or a random path (e.g.: message ferries).

In this paper, a novel method for routing in infrastructure-less OppNets that uses a combination of fixed and mobile infrastructures for message passing between the nodes is proposed. This infestation-ferry combination is meant to ensure that the message copy remains at the infostation in case the ferry fails, which is not possible if only ferries are used in the network. Further, the infostation also helps in preventing the message loss in the network. The message may get dropped at few concentrated nodes since it has to be stored for a longer period of time until a promising next hop node is encountered. This is due to the fact that the infostation acts as a permanent storage device and can pass the message copy to some other ferry it comes in contact with. The ferry is also necessary as it acts as a transporter that takes the message from one part of the network to the other in case of sparse network scenarios. This type of architecture can find applications in urban city scenarios where trains, metros, buses, to name a few, follow a predefined fixed path, and hence can act as ferries. In this case, radio towers and access points can act as infostations; and smart phones, laptops, and PDAs, can act as regular nodes that generate the message in the network.

The rest of the paper is organized as follows. In Section II, some background and related work on routing protocols for infrastructure-based OppNets are presented. In Section III, the proposed protocol is described. In Section IV, the proposed algorithm and sub-routines used in this work are presented. In Section V, the simulation results are presented. Section VI concludes the paper and provides some insights on the future work.

II. BACKGROUND AND RELATED WORK

In the literature, most of the proposed routing protocols for OppNets have been for the class of infrastructure-less OppNets, and few routing protocols have been proposed, that use dedicated infrastructures to assist in message passing between the nodes in OppNets. Since this paper focuses on routing protocols in infrastructure-based OppNets, only representatives such protocols are discussed.

A. Infostation Model

Infostations [26] are an array of isolated objects of high bandwidth connectivity that provide high bit rate services in core locations using separate radios. They can share a common channel and yield a transmission rate higher than few megabits per second if properly spatially separated. They can be installed along roadways, in buildings, at airports, and other known areas [26]. In addition, they allow significant amount of information such as fax, e-mail, large data files, to be transferred in few seconds. Individual infostations on highways and airports can provide special gateway to the Internet as well as access to remote servers for messages transfer. When any vehicle comes in contact with the infostation, it gets the required data packet. In the infostation model, the node that wishes to send a message moves closer to the infostation and uploads the message to it. The infostation keeps the message in its buffer and takes the responsibility to deliver it to the destination.

B. Shared Wireless Infostation Model

The Shared Wireless Infostation Model (SWIM) [11] is an integration of the infostation and the ad-hoc network. It comprises the infostations placed at fixed locations and the regular nodes, where each node is equipped with a large storage buffer. The nodes exchange the information whenever they come in contact with each other. Whenever any node comes in contact with the infostation, it offloads all its information to the infostation. While doing so, every node keeps track of the packets that it is carrying and the packets it has delivered to the infostation so that it does not accept the same packets again in the future. The SWIM model was studied with the help of a biological acquisition system in which the tagged whales...
act as nodes. The offloading stations called SWIM stations are placed on the sea surface, and whales are chosen because they return to the same place at regular intervals and spend minutes or hours on the sea surface, thus large information can be transferred. The whale offloads all the information to the infostation whenever it comes on the sea surface. The infostations then transmit the information to the shore either with the help of other infostations through an ad-hoc network or with the help of a satellite communication. To keep check of the information that needs to be deleted from the buffer of a node, two design strategies are followed [11]. The first method is through the invalidate packets. An invalidate packet is transferred to the node by the infostation whenever it offloads its information. This invalidate method is through the Service Request whenever a node wants to send a message, it generates a scheme, the ferry moves proactively to meet the node. The second method is based on the estimation of probability as a function of time in which a packet is delivered to any of the offloading stations. Every packet carries a timestamp, which denotes the time after which the packet is deleted from the storage of the node.

C. Message Ferry Scheme

The Message ferries (MFs) [27] are the special nodes in the network that are used to deliver the message between disconnected nodes. A ferry node has high energy, buffer, and speed compared to other nodes in the network. The ferries move in a region by using some fixed routes while communicating with other nodes that they meet. The ferry route should be designed in such a way that it can improve certain network characteristics such as average data delivery delay, message delivery rate, and data loss ratio. A ferry node moves around in its region and communicates with other nodes present in the region and establishes a connection with the GF to transfer the collected messages. The regular nodes cannot communicate with each other directly. The LF collects messages from the nodes present in the region and establishes a connection with the GF to transfer the collected messages when the GF comes in its region. The LF is also responsible for the intra-region communication between the nodes whereas the GF is meant to facilitate the inter-region communication among the nodes since it can carry and forward the messages between different regions. Both the LF and GF move with the same speed in the network. A sample network model used in this scheme is shown in Fig. 1.

D. Data Mobile Ubiquitous LAN Extensions Scheme

In [28], a three-tier architecture has been proposed. The lower tier comprises sensor nodes that collect the information from the surroundings at regular intervals. The middle tier consists of mobile ubiquitous LAN extensions (or data-MULEs for short). These are special nodes that move around the network and collect the information from the sensor nodes, then deliver it to the access points. The MULEs have large buffer compared to sensor nodes; they also have short wireless communication capabilities to communicate with the sensor nodes. The upper tier is composed of access points, which process the information received from the MULEs. These access points are connected to a data warehouse where further processing on the received data is performed. This architecture is energy efficient since the sensor nodes are not involved in the data forwarding. It suffers from data latency, which can be addressed by properly positioning the sensor nodes and the access points, and by using the MULEs mobility model.

E. Global Ferry Scheme

The Global ferry scheme (GFS) [29] uses multiple local ferries (LFs) along with a single global ferry (GF) to minimize the average message delivery delay in OppNets. In this scheme, the communication area is divided into square regions, each of which has a dedicated LF and N/M stationary nodes, where N is total number of nodes in the network. The regular nodes cannot communicate with each other directly. The LF collects messages from the nodes present in the region and establishes a connection with the GF to transfer the collected messages when the GF comes in its region. The LF is also responsible for the intra-region communication between the nodes whereas the GF is meant to facilitate the inter-region communication among the nodes since it can carry and forward the messages between different regions. Both the LF and GF move with the same speed in the network. A sample network model used in this scheme is shown in Fig. 1.
In Fig. 1, initially, all LFs are at the center point of their region and the GF can start moving from any region. The LF comes in contact with the GF only at the center point of the region. The LF remains at the center point until it meets with the GF for the first time. After meeting with the GF, the LF can start moving in its region. It moves on a path that passes through every regular node in the region and the center point in the region. Whenever two nodes come in contact with each other, they use the epidemic routing protocol [18] to forward the messages. The LF and GF are equipped with enough buffer space so as to avoid the dropping of the messages.

F. Dynamic Message Ferry Route Scheme

In this scheme [30], nodes in the network are divided into various groups called clusters. Each cluster has a node that works as head called cluster head. Each cluster region is called a head zone. Ferry nodes are also deployed in the network. They move from one head zone to another and are responsible for the inter-cluster communication. On arriving at a particular cluster, the ferry node will deliver the data to the cluster head and decides about its movement to the next head zone. As the members of a particular cluster are connected with each other, the message passing among them can be realized by means of any MANETs routing protocol. If the destination is present in another head zone, the ferry will transfer the message to the destination node’s cluster head.

III. PROPOSED ROUTING PROTOCOL

A. Motivation

As routing in OppNets is still a major concern, devising for a new routing protocol that guarantees the message delivery between nodes, with minimum delay, is always desirable. Due to the sparse connectivity, intermittent links, uncertainty in mobility behavior, and frequent disconnections between the nodes, it is highly desirable to employ some infrastructure in the form of special nodes that can help in providing the communication opportunities in such networks. Existing routing protocols for infrastructure-based OppNets either use infostation or message ferry as an infrastructure in the network. The infostation provides high bandwidth connectivity within a specified area and can be used as backup storage for the messages. But, in the cases that the network is sparse, the infostation may not work since the nodes may never come in the range of the infostation for a longer period of time. The message ferry is used to collect the messages from the nodes, and thus it provides a reliable source for communication and transmission of messages in sparse networks where network partition problem is very common. Furthermore, a ferry failure in the network can lead to the situation of message dropping and message delivery delays. Thus, there is a need to design a new infrastructure-based routing protocol for OppNets that can handle the above-mentioned weaknesses of the infostation and the message ferry. The motivation to design this protocol has come from the limitations of the existing GFS scheme used in OppNets, and the desire to take advantage of the features resulting from the integration of the infostation and message ferry into a single architecture.

The GFS scheme is highly dependent on a single global ferry employed in the network. With bigger network size and larger number of nodes, this single global ferry is inadequate to handle the massive increase in the network traffic, and thus a bottleneck can occur. Therefore, it cannot alone efficiently serve the increased demands of the network. Its failure can also lead to the situation of network partition, and message dropping as there will not be any medium available in the network to provide the inter-region communication between the nodes. This will result in catastrophic consequences on the network resources and delivery of messages. It is therefore necessary to design a network architecture that can provide permanent message storage such as infostation and utilizes multiple ferries for carrying messages between different regions of the network. This infostation-ferry architecture will provide reliable communication in case of ferry failure and also will decrease the load on each ferry. Further, the GFS scheme uses the epidemic protocol to forward the messages between two nodes. This causes some overhead in the network. Another drawback of the GFS scheme is that the local ferry moves to contact each ordinary node in its region. This will increase the routing length of the local ferry, which in turn will increase the message delivery delay if the number of ordinary nodes gets increased in the network. For this reason, this paper proposes a new routing protocol as an improvement of the GFS scheme in the sense that our scheme is meant to improve the message delivery while minimizing the incurred overhead and delivery delay in case of ferry failure.

B. Network Model and Components of the Proposed HVF Scheme

The network model and components used in HVF scheme are described in this subsection. The communication area for this protocol is considered to be equal-sized square grids. Each grid is denoted by its coordinate (i,j) where i denotes the column number and j denotes the row number of that grid. The number of grids can increase with an increase in the network size and can be taken as per the requirement. As the number of grids increases, the infrastructure (made of infostations and ferries) also increases in the network.

Figure 2. Network model.

Without loss of generality, let’s describe the working of the proposed protocol by considering only five square
Designing the protocol are as follows:

**D. Assumptions**

- Every node is equipped with a global positioning system (GPS) [31].
- Every ferry and the infostations are equipped with a short-range radio and a long-range radio. Regular nodes have only short-range radio, and cannot communicate with each other.
- The transmission time of a message from ferry to ferry and node to ferry is considered to be negligible compared to the message delivery delay from the source to the destination.
- Regular nodes are considered stable and not highly mobile.
- If there is a link between any two nodes, then the communication between them is assumed to be reliable.

**E. Working of the Protocol**

A regular node generates the message in the network. If the destination is present in its square grid then the message is passed to it, otherwise it detects the nodes (ferries or infostation) present in its neighborhood. If multiple nodes are detected, the message is given to the node that has the highest weight value and more chances to deliver the message towards the destination, where the weight $W(S)$ of a node $S$ (here an infestation or a ferry) is obtained as:

$$W(S) = w_1*v(S)+w_2*Power(S)+w_3*Buffer(S)$$  \(1\)

where $v(S)$, $Power(S)$, and $Buffer(S)$ are respectively the node $S$ velocity, power, and buffer, and $w_1$, $w_2$, $w_3$ are some weights. When an infostation is selected, the message is passed to it. Then, the infostation forwards this message in its buffer. When the ferry comes in the square grid, the infostation gives this message to the ferry. The ferry then selects some other suitable ferry or infostation to meets to pass the message towards the destination. Once the message is passed from one node to another node, the ACK procedure is activated to acknowledge the message sending node.

The ACK procedure works as follows: suppose that a node, say A, gives a message to another node, say B. Node A will then keep a copy of this message in its buffer. When node B will encounter a node C, it will transfer this message to node C and keep a copy of it in its buffer. Node B will then send an acknowledgement to node A indicating that it has further transferred the message. After receiving the acknowledgment from node B, node A will remove the copy of the message copy its buffer. Similarly, node C will acknowledge node B when it has transferred the message to some other node, say D, and this process will continued until the message reaches the destination. Thus, in case an infostation transfers the message to a ferry, it will invoke the ACK procedure to acknowledge the node from which it has received that message. The ferry will then keep the message copy in its buffer until it receives an acknowledgment from a ferry or infostation to which it has forwarded that message. This way, after some time, the message will be removed from the buffers of all nodes.

---

**C. Terminology**

- **Row ferry**: Ferry which is present in the row of the destination node.
- **Column ferry**: Ferry which is present in the column of the destination node.
- **Message-passing node**: Node that currently has the message copy to be routed in the network.
- **Round trip time**: Time taken by a ferry to reach the end of its path from its starting point and return back to the starting point again.
- **Φ**: This symbol is used to denote the NULL value. It represents the rejection of a node in the routing process.

**D. Assumptions**

The assumptions that are taken into account when designing the protocol are as follows:

- Every node is equipped with a global positioning system (GPS) [31].
- Every ferry and the infostations are equipped with a short-range radio and a long-range radio. Regular nodes have only short-range radio, and cannot communicate with each other.
- The transmission time of a message from ferry to ferry and node to ferry is considered to be negligible compared to the message delivery delay from the source to the destination.
- Regular nodes are considered stable and not highly mobile.
- If there is a link between any two nodes, then the communication between them is assumed to be reliable.

---

**E. Working of the Protocol**

A regular node generates the message in the network. If the destination is present in its square grid then the message is passed to it, otherwise it detects the nodes (ferries or infostation) present in its neighborhood. If multiple nodes are detected, the message is given to the node that has the highest weight value and more chances to deliver the message towards the destination, where the weight $W(S)$ of a node $S$ (here an infestation or a ferry) is obtained as:

$$W(S) = w_1*v(S)+w_2*Power(S)+w_3*Buffer(S)$$  \(1\)

where $v(S)$, $Power(S)$, and $Buffer(S)$ are respectively the node $S$ velocity, power, and buffer, and $w_1$, $w_2$, $w_3$ are some weights. When an infostation is selected, the message is passed to it. Then, the infostation forwards this message in its buffer. When the ferry comes in the square grid, the infostation gives this message to the ferry. The ferry then selects some other suitable ferry or infostation to meets to pass the message towards the destination. Once the message is passed from one node to another node, the ACK procedure is activated to acknowledge the message sending node.

The ACK procedure works as follows: suppose that a node, say A, gives a message to another node, say B. Node A will then keep a copy of this message in its buffer. When node B will encounter a node C, it will transfer this message to node C and keep a copy of it in its buffer. Node B will then send an acknowledgement to node A indicating that it has further transferred the message. After receiving the acknowledgment from node B, node A will remove the copy of the message copy its buffer. Similarly, node C will acknowledge node B when it has transferred the message to some other node, say D, and this process will continued until the message reaches the destination. Thus, in case an infostation transfers the message to a ferry, it will invoke the ACK procedure to acknowledge the node from which it has received that message. The ferry will then keep the message copy in its buffer until it receives an acknowledgment from a ferry or infostation to which it has forwarded that message. This way, after some time, the message will be removed from the buffers of all nodes.
IV. ALGORITHM AND SUB-Routines

A. Algorithm

1. Let $S_1$ be a source node and $D_1$ be a destination node.
2. If (there is any node in its neighborhood which is the destination node)
   Then call the ACK procedure and deliver the message to it and exit.
   Else
   Detect all the neighboring nodes (ferries and the infostation) present in its square grid.
3. Let $S_2$, $S_3$, $\ldots$, $S_k$ be the detected nodes, where $k$ is the number of nodes detected.
4. Select a node from the detected nodes by following the selection procedure.
5. In selection procedure, select a node $S$ which has the highest weight $W(S)$ from the detected nodes.
6. Let $S$ be the selected node.
7. Before passing the message to the selected node, the message-passing node stores the copy of the message in its buffer and waits for an acknowledgement from the selected node (with the help of the ACK procedure). This means that it has passed the message to some other suitable node (ferry or infostation).
8. If the message-passing node is the row or column ferry in which the destination resides, then it will not delete the message from its storage until it meets the destination and delivers the message to it.
9. Pass the message to the selected node $S$ along with the ID of the message-passing node that acts as a previous ID (used by the selected node for the ACK procedure). The message contains Source ID, message ID and Destination ID.
10. If the acknowledgment for a message is not received during the round trip time of the ferry, go back to Step 2.
11. Repeat Step 2 to Step 10 using the message-passing node as selected node, and follow the above algorithm till the message is delivered to the destination.

B. Description of Subroutines Used in the Algorithm

Weight function calculation subroutine

This subroutine calculates the value of the weight function $W(S)$. It further calls the power, velocity and buffer subroutines to calculate the final value of $W(S)$. */

```
Wi = \text{Get \ Power(S), the power of } S \ \text{.} \ \text{v}(S) \ \text{the velocity of } S, \ \text{Buffer(S) the buffer of } S\text{.} \ \text{If any of the Power(S), v(S), W is } \Phi\text{ then } W=\Phi; \ \text{Else} \ \text{W} = w_1 \ast v(S) + w_2 \ast \text{Power(S)} + w_3 \ast \text{Buffer(S)}; \ \text{// where } w_1, w_2 \text{ and } w_3 \ \text{are the weights assigned to } v(S), \ \text{Power(S) and Buffer(S) respectively. Return } Wt\text{.} \ \text{If } \Phi \ \text{then the message will not be given to that node.}
```

Power subroutine

This subroutine returns the normalized power $P$ of a node. If $P_{\text{remaining}}$ is the remaining power of the node, $P_{\text{threshold}}$ is the threshold value of the power below which no message passing will take place, and $P_{\text{total}}$ is the initial level of power of a node */

```
Power(S)/
if($P_{\text{remaining}}<P_{\text{threshold}}$)
Return ($P_{\text{remaining}}/P_{\text{total}}$,)
Else
Return $\Phi$;
```

Distance velocity subroutine

This subroutine finds out how closely a node is moving towards the destination. If a node, say $S$, is moving away from the destination, it will return $\Phi$, else it will return the $v(S)$ the velocity of $S$. Here, $\Phi$ denotes the NULL value, indicating that a message will not be given to other node */

```
v(S)/
if(S is an infostation)

\{
\text{Angle made by velocity vector with its position vector=0; // where, position vector } = \Phi - \Phi;
\}
Get Dt=(Distance between $D_t$ and $S$)(Distance between $D_t$ and $S$);
v=S(180-Angle made by velocity vector with its position vector) / 180;
if(V is less than 90 degree) then
$v(S)=x_1 \ast D + x_2 \ast V; // here, } x_1, x_2 \text{ are the weights assigned to } D \text{ and } V \text{ respectively.}
else
$v(S)=\Phi$;
Return $v(S)$;
```

Buffer subroutine

This subroutine calculates the normalized value of the remaining buffer space of a node. Here, remaining buffer space denotes the remaining space in the buffer of a node and total space denotes the initial total buffer space of a node */

```
Buffer(S)/
if(remaining_buffer_space!=0)
Return (remaining_buffer_space)/total_space;
else
Return $\Phi$;
```

ACK procedure subroutine

This subroutine is activated, which removes the message from the node with id = previous_node_id. */

```
ACK (previous_node_id, message_id); \text{Check the node with id=previous_node_id;} \text{ Remove the copy of message with id=message_id from the previous_node_id node;}
```

C. Scenario of the HVF Scheme

In this section, an example scenario of the HVF scheme is presented, which is based on the movement of the ferries in the network. In HVF, the ferries move in horizontal and vertical direction in each row and column. In general, multiple network configurations that use different numbers of ferries and different ferry paths are possible. For example, in a modified case of HVF, instead of one, two ferries in each row and column can move in opposite directions. In another configuration, there can be a horizontal ferry in each row and two diagonal ferries that move diagonally in the network. This increase in the number of ferries will improve the
network connectivity and reduce the message delay, but it may also increase the cost of infrastructure in the network.

The considered scenario of the HVF scheme is depicted in Fig. 3. Initially, all the ferries are present in the first grid of their corresponding rows or columns. After a certain time $t$, they move to the second grid of their rows or columns, and so on. Here, $t$ denotes the time taken by a ferry to move from one grid to another. In this scheme, the ferries are synchronized in such a way that the horizontal ferries meet with the vertical ferries only at the diagonal square grids as shown in Fig. 3.

![Figure 3. A scenario of the HVF scheme.](image)

The working of the HVF scheme at a particular instant of time is shown in Fig. 4. Here, $S_n$ is the source node and $D_n$ is the destination node. As per the algorithm, $S_n$ detects the nodes present in its square grid. The detected nodes can be either infostations or ferries (horizontal or vertical). These detected nodes can be selected as message-passing nodes. If a ferry which is moving towards the destination is selected, the message will be passed to it. This ferry will then take the message to the next square grid. If both ferries are not present in the square grid at that time, $S_n$ will transfer the message to the infostation present in the square grid at location $a$. This infostation will then become the message-passing node. This message from the infostation at $a$ can be taken either to $b$ if the horizontal ferry comes first or to $d$ if the vertical ferry comes first in the direction of the destination. Let’s assume that the horizontal ferry comes first at $a$ and take the message to the infostation at $b$. The infostation at $b$ will then acknowledges $S_n$ and will wait for an acknowledgment from this ferry. This acknowledgment will be given by the ferry when it will return back at $a$ after reaching the end of the row. From $b$, the message will go either at $c$ or $e$ depending on whichever ferry visits $b$ first moving in the direction of the destination.

![Figure 4. Unit step ladder in HVF.](image)

In Fig. 4, it is observed that the vertical ferry reaches $b$ (well before the horizontal ferry) since it has to travel a lesser distance to come back to $b$ in the direction of the destination compared to the horizontal ferry case.

Let the horizontal ferry at $bbe$ in grid $(i,j)$ where $i$ and $j$ are respectively the column number and row number of the grid. Then, the corresponding vertical ferry will be in grid $(i,i)$ and the horizontal ferry will have to travel a total of $D_v = (n-i) + (n-1) + (i-1) = 2*(n-1)$ grids distance to reach grid $(i,j)$ again in the direction of the destination. Here, $n$ denotes the number of grids in each row or column. The vertical ferry has to travel a total of $D_v = (n-i) + (n-j) = 2n-i-j$ grids distance to reach the grid $(i,j)$ in the direction of the destination. It can be observed that $D_v ≥ D_h$. Therefore, the vertical ferry carries the message from $b$ to $c$ i.e. to the diagonal grid from the original initial position $a$ (this is referred to as traversing a unit step ladder). The infostation at $c$ acknowledges the horizontal ferry when it passes the message to the vertical ferry and itself waits for an acknowledgment from this ferry. The infostation at $c$ then acknowledges the vertical ferry when it passes the message to the horizontal ferry of the upper row, and so on.

![Figure 5. Message path in HVF.](image)

Similarly, at location $a$, if the message is taken by the vertical ferry (instead of the horizontal ferry) and reaches $d$, then the horizontal ferry will take it from $d$ to $c$, resulting from a unit step ladder from $a$ to $c$. This process of moving diagonally towards the destination with a single step each time in the unit step ladder continues until the message reaches the row or column of the destination. At this point, the row or column ferry will be responsible for taking it to the destination. The path followed by message in HVF is shown in Fig. 5.

**D. Minimum Time/Maximum Time in Message Delivery in HVF Scheme**

The minimum time occurs when the source and destination are present in the same square grid. This is the transmission time from source to infrastructure and from infrastructure to destination. When the source and destination are present in different square grids, the total time taken in the delivery of the message can be divided into three parts: the time taken by a ferry to take the message from the source node; the time taken to satisfy all the steps of the unit step ladder; and the time taken to reach the row or column ferry of the destination; and
the time taken by the row or column ferry to reach the destination.

In order to traverse the unit step ladder, the message has to traverse horizontally and then vertically or vice-versa. Therefore, the time taken by the steps of the horizontal and vertical movement is more than the time taken by the row or column ferry to deliver the message to the destination. Thus, the maximum time will occur when the source and destination nodes are present at the extreme corner grids of the network as shown in Fig. 6.

The corresponding horizontal ferry moving left towards the message will be at the (j,j)th grid. The time taken by the horizontal ferry to reach the last grid in the jth row will be (j-1)\(t\). Therefore, the time taken by the horizontal ferry to reach the grid in which the message is currently present (i.e. (i,j)th grid) will be (i-1)\(t\), and the time taken by it to move the message one grid to the right will be t. Thus, the time taken by the horizontal ferry will be (j-1)\(t\) + (i-1)\(t\) + t. Since the message has reached the (i+1,j)th grid and the corresponding vertical ferry is at the (i+1,j+1)th grid and is moving downwards, the time taken by the vertical ferry to reach the grid in which the message is currently present (i.e. the (i,j)th grid) will be (i-1)\(t\), and the time taken by it to move the message one grid to the right will be t. Thus, the time taken by the vertical ferry will be (j-1)\(t\) + (n-1)\(t\) + t. Now, the message reaches the (i+1,j-1)th grid. Thus, to move one step in the unit step ladder from the (i,j)th grid to the (i+1,j-1)th grid, the total time taken is obtained as (i-1)\(t\) + (i+1)\(t\) + (n-1)\(t\) + (n-j+1)\(t\). Therefore, to reach from (i+k,j-k)th grid to next grid in the unit step ladder, the total time taken will be [(j-k+1) + (i+k-1)+1+ (n-i-k-1)+ (n-j+k+1)]\(t\).

In Fig. 6, each grid is represented by a coordinate (i,j) where i and j are respectively the column number and the row number of the grid. Let the source be in the (1,1)th grid and the destination be in the (n,n)th grid. For the purpose of calculation, the time taken to traverse a single step in the unit step ladder is divided into two parts. The first part deals with the situation when the message moves horizontally right and the second part deals the situation when it moves vertically up. Let’s assume that the message is currently taken to the (i,j)th grid in the first column and all the vertical ferries lie in the first row, the horizontal ferry is present in the (1,n)th grid, which also contains the source node. Thus, the ferry does not need to return from its current position to the left corner and will come back to the grid in which the message is currently present. Thus, \(T_{max}\) becomes:

\[
T_{max} = \sum_{j=0}^{\lfloor n/2 \rfloor} (2(n-1) * t) - n * t
\]

(3)

In the situation where all the ferries lie in the centre grids of the network, \(T_{max}\) becomes:

\[
T_{max} = \sum_{k=0}^{n-2} [(2n-1) * t] - \lfloor n/2 \rfloor * t
\]

(4)

V. SIMULATION RESULTS

A. Simulation Setup and Parameters

In this work, the performance evaluation of the HVF and GFS schemes is done using the ONE simulator [32]. The simulation area is divided into equal size square grids. Regular nodes are mobile in nature and move randomly.

In the ONE simulator [32], the MAC layer is abstracted to the bit rate and transmission range. The simulator has generic interfaces that are used to model nodes with multiple radios. Nodes can be configured with an arbitrary number of interfaces, as well as different transmission rates and ranges as per the requirement. For instance, nodes can be configured with high bit-rate but short range interfaces (e.g. Bluetooth) and with lower bit rate but longer range interfaces (e.g. cellular). The simulator also allows a subset of nodes to create a backbone network by using a long range interface. The nodes within this subset communicate with each other using this interface, and with other nodes using a short range interface.

Nodes are divided into five groups, namely infostation, horizontal ferries, vertical ferries, source nodes, and destination nodes. The source nodes and destination nodes groups form the category of regular nodes. A regular node has a single interface to communicate with the ferry and the infostation. The infostation has two interfaces; one to communicate with the regular nodes and other to communicate with the ferry. Similarly, the ferry has two interfaces; one to communicate with the regular nodes and other to communicate with each other and the infostation. The infostation covers the whole grid in which it lies. The nodes in the corner of the grid are not in the radio range of the infostation, and are covered by the ferry when it moves towards them. The speed of

\[
T_{max} = \sum_{j=0}^{\lfloor n/2 \rfloor} [(2(n-1) * t) - \lfloor n/2 \rfloor * t]
\]

The case when the source is located at the bottom left grid is treated in a similar way. To determine the total time taken to reach the top right grid, all the n-1 diagonal grids traversed in the unit step ladder should be covered, thus, the maximum time will be obtained as:

\[
T_{max} = \sum_{k=0}^{n-2} (2n-1) * t
\]

(2)

In the situation where all the horizontal ferries lie in the first column and all the vertical ferries lie in the first row, the horizontal ferry is present in the (1,n)th grid, which also contains the source node. Thus, the ferry does not need to return from its current position to the left corner and will come back to the grid in which the message is currently present. Thus, \(T_{max}\) becomes:

\[
T_{max} = \sum_{k=0}^{n-2} [(2n-1) * t] - n * t
\]

(3)

In the situation where all the ferries lie in the centre grids of the network, \(T_{max}\) becomes:

\[
T_{max} = \sum_{k=0}^{n-2} [(2n-1) * t] - \lfloor n/2 \rfloor * t
\]

(4)

V. SIMULATION RESULTS

A. Simulation Setup and Parameters

In this work, the performance evaluation of the HVF and GFS schemes is done using the ONE simulator [32]. The simulation area is divided into equal size square grids. Regular nodes are mobile in nature and move randomly.

In the ONE simulator [32], the MAC layer is abstracted to the bit rate and transmission range. The simulator has generic interfaces that are used to model nodes with multiple radios. Nodes can be configured with an arbitrary number of interfaces, as well as different transmission rates and ranges as per the requirement. For instance, nodes can be configured with high bit-rate but short range interfaces (e.g. Bluetooth) and with lower bit rate but longer range interfaces (e.g. cellular). The simulator also allows a subset of nodes to create a backbone network by using a long range interface. The nodes within this subset communicate with each other using this interface, and with other nodes using a short range interface.

Nodes are divided into five groups, namely infostation, horizontal ferries, vertical ferries, source nodes, and destination nodes. The source nodes and destination nodes groups form the category of regular nodes. A regular node has a single interface to communicate with the ferry and the infostation. The infostation has two interfaces; one to communicate with the regular nodes and other to communicate with the ferry. Similarly, the ferry has two interfaces; one to communicate with the regular nodes and other to communicate with each other and the infostation. The infostation covers the whole grid in which it lies. The nodes in the corner of the grid are not in the radio range of the infostation, and are covered by the ferry when it moves towards them. The speed of
the horizontal/vertical ferry (case of the HVF scheme) and that of the global/local ferries (case of the GFS scheme) are kept synchronized with each other. Table I below shows the values of various parameters taken during the simulation. The simulation parameters are given in Table 1.

### TABLE I. SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation area</td>
<td>7500m * 7500m</td>
</tr>
<tr>
<td>Size of each square grid</td>
<td>1500m * 1500m</td>
</tr>
<tr>
<td>Simulation time</td>
<td>18000 sec</td>
</tr>
<tr>
<td>Number of regular nodes</td>
<td>80</td>
</tr>
<tr>
<td>Ferry’s speed</td>
<td>9 m/s</td>
</tr>
<tr>
<td>Regular node’s speed range</td>
<td>0.5-1.5 m/s</td>
</tr>
<tr>
<td>Communication Interface</td>
<td>Bluetooth / Cellular</td>
</tr>
<tr>
<td>Regular node transmission range</td>
<td>30m</td>
</tr>
<tr>
<td>Ferry / Infostation transmission range</td>
<td>750m in radius</td>
</tr>
<tr>
<td>Regular node transmission speed</td>
<td>250Kbps</td>
</tr>
<tr>
<td>Ferry / Infostation transmission speed</td>
<td>10Mbps</td>
</tr>
<tr>
<td>Buffer capacity of regular node</td>
<td>5Mb</td>
</tr>
<tr>
<td>Buffer capacity of ferry</td>
<td>50Mb</td>
</tr>
<tr>
<td>Buffer capacity of Infostation</td>
<td>100Mb</td>
</tr>
<tr>
<td>Message size</td>
<td>500Kb to 1Mb</td>
</tr>
<tr>
<td>Message generation interval</td>
<td>25-35 sec</td>
</tr>
<tr>
<td>Message Time-to-live (TTL)</td>
<td>200 minutes</td>
</tr>
<tr>
<td>Regular node’s movement model</td>
<td>Random Waypoint</td>
</tr>
<tr>
<td>w₁</td>
<td>0.4</td>
</tr>
<tr>
<td>w₂</td>
<td>0.3</td>
</tr>
<tr>
<td>w₃</td>
<td>0.3</td>
</tr>
<tr>
<td>x₁</td>
<td>0.4</td>
</tr>
<tr>
<td>x₂</td>
<td>0.6</td>
</tr>
</tbody>
</table>

The following varying parameters are considered:

- **Varying the number of nodes**: The number of source and destination regular nodes in the network is varied from 40 to 200 with an increment of 40 nodes each time.
- **Varying the ferry speed**: The speed of the ferry is varied from 9 m/s to 45 m/s with an increment of 9 m/s each time.
- **Varying the number of messages generated in the network**: The number of messages created by the regular nodes varied from 200 to 1000 with an increment of 200 each time.
- **Varying the node speed**: The speed of regular nodes is varied from 1 m/s to 3 m/s with an increment of 0.5 m/s each time.
- **Varying the message size**: The message size is varied from 1Mb to 5 Mb with an increment of 1Mb each time.
- **Varying the Message Time-to-live (TTL)**: The message TTL is varied from 100 minutes to 300 minutes with an increment of 50 minutes each time.

### B. Performance Metrics

The following performance metrics are considered:

- **Number of messages delivered**: It is the number of messages successfully delivered to the destination during complete simulation run time.
- **Overhead ratio**: This is an assessment of the bandwidth efficiency. It is calculated as

\[
\text{Overhead ratio} = \frac{\text{NumberOfRelayedMessages}}{\text{NumberOfDeliveredMessages}}
\]

- **Average latency**: It is the average of the difference between the message creation time and the message delivery time.
- **Average buffer time**: It is the average time for which the messages stayed in the buffer at each node.

### C. Simulation Results

In this section, the HVF scheme is compared against the GFS scheme using the above-mentioned performance metrics. The simulations were carried out in two different scenarios. The first scenario is when both schemes work under normal operation (i.e. ferry failure). In this case, the HVF and GFS are respectively denoted as N-HVF and N-GFS. The second scenario is when both schemes operate under a ferry failure. In this case, they are respectively denoted as FF-HVF and FF-GFS. In this case, one horizontal ferry is disabled in case of FF-HVF and the global ferry is disabled in case of FF-GFS.

1) **Varying the Number of Nodes**

The number of source and destination regular nodes is varied and the impact of this variation on the studied performance metrics is analyzed. The results are captured in Fig. 7(a)-(d). In Fig. 7(a), it is observed that with an increase in the number of nodes, the total number of messages delivered increases. This is due to the fact that with an increase in number of nodes, more messages will be generated in the network, resulting to higher message delivery rate. HVF yields more number of messages delivered compared to GFS scheme in both normal and ferry failure scenarios. In case of ferry failure in GFS, only those messages whose sender and receiver lie in the same region are actually delivered by the local ferries. In Fig. 7(b), the overhead ratio increases with an increase in the number of nodes. This is due to the increase in the number of messages flowing in the network (i.e. the number of relayed messages). HVF yields a lesser overhead ratio compared to GFS in both the normal and ferry failure scenarios. This is attributed to the ACK procedure used in the HVF scheme, which imposes some limitation on the number of message copies in the network. The GFS scheme uses the epidemic routing to pass the messages between the nodes, therefore creates a lot of message copies, resulting to excessive overhead. In Fig. 7(c), it is observed that the average latency obtained with GFS is much higher than that obtained with HVF in both the normal and ferry failure scenarios. In the normal scenario, the result is attributed to the fact that with an increase in the number of nodes in the GFS scheme, the local ferry in each region has to visit more regular nodes. This increases the routing length of the local ferry, which in turn increases the time periods it takes to visit all the regular nodes, thereby the average delivery delay of messages also increases. In the ferry failure scenario, the result obtained is due to the fact that the failure of the global ferry leads to disconnected regions in the GFS scheme, thereby to an increase in delay (case of GFS);
whereas in the case of HVF, if either the horizontal or vertical ferry fails, then the other ferry can take the responsibility to deliver the messages to the destination. In Fig. 7(d), it is observed that when the number of regular nodes increases, the average buffer time also increases. This is due to the fact that with an increase in the number of nodes, more messages get created, which remains in the buffer of the nodes until they are delivered to the destination, hence the buffer time is increased.

HVF has lesser average buffer time compared to GFS in both the normal and ferry failure scenarios. This is due to the ACK procedure used in the HVF scheme, which removes the message from the previous node and thus limits the number of message copies stored in the buffer of the nodes in the network. On the other hand, the GFS scheme uses the epidemic routing, which generates lot of message copies in the network that are stored in the buffer of the nodes, and thus increase the average buffer time. In case of ferry failure, the average buffer time of GFS increases to a very high value due to the non-delivery of the messages to the destination. In addition, N-HVF performs 100.91% better than N-GFS whereas FF-HVF performs 158.74% better than FF-GFS in terms of number of messages delivered. The overhead ratio, average latency, and average buffer time of N-HVF are 50.47%, 40.87%, and 37.34% lesser than that of N-GFS respectively; whereas for FF-HVF, these values are 61.71%, 44.64%, and 65.79% lesser than that of FF-GFS.

2) Varying the Ferry Speed

The speed of the ferry is varied and the impact of this variation on the studied performance metrics is analyzed. The results are captured in Fig. 8(a)-(d). In Fig. 8(a), it is observed thatas the speed of ferry is varied from 9 m/s to 27 m/s, the number of messages delivered increases. This is due to the fact that with increase in speed, the ferry carries the messages towards the destination more frequently and thus it can deliver more messages per unit time. With further increase in ferry speed from 27 m/s to 45 m/s, the number of messages delivered decreases. This can be attributed to the fact that nodes/infostations do not get enough time to remain in contact with the ferry so as to exchange the messages with it. This causes the decrease in the number of messages delivered. In addition, HVF outperforms GFS in terms of number of messages delivered in both the normal and ferry failure scenarios. In Fig. 8(b), it is observed that the overhead ratio decreases with an increase in the ferry speed from 9 m/s to 27 m/s. It has a lowest value when the ferry speed is 27 m/s. This is attributed to the increase and decrease in the number of messages delivered, and thus in the number of messages relayed in the network when the ferry speed is varied from 9 m/s to 27 m/s and from 27 m/s to 45 m/s respectively. It is also observed that the overhead ratio of HVF is lesser than that of GFS in both normal and ferry failure scenarios. In Fig. 8(c) (resp. Fig. 8(d)), it is observed that the average latency (resp. average buffer time) first decreases and then increases with an increase in the ferry speed. This is justified by the fact that due to the high frequency of the ferry, the messages can be delivered to the destination in lesser time, and hence can be removed from the buffer of the nodes and the ferry, resulting to reduced latency and buffer time. With further increase in the ferry speed from 27 m/s to 45 m/s, some messages may not be fully transferred from the ferry to other ferry/infostation. This increases the number of incomplete messages transferred due to the shorter contact time with the ferry. The messages that are not completely transferred will remain in the buffer of the nodes until a ferry comes in contact with the node again.
Therefore, the increase in the latency and buffer time occur. It can also be observed that HVF outperforms GFS in terms of average latency and average buffer time. This is due to the availability of horizontal and vertical ferries in the HVF scheme, which result in multiple carriers that can take the message to the destination in lesser time. On the other hand, in GFS, only a single global ferry provides the inter-region communication among the nodes, resulting to longer delays compared to the HVF scheme. N-HVF performs 34.77% better than N-GFS, while FF-HVF performs 86.64% better than FF-GFS in terms of number of messages delivered. The overhead ratio, average latency, and average buffer time of N-HVF are 63.18%, 53.71%, and 57.79% lesser than that of N-GFS respectively; whereas for FF-HVF, these values are 75.33%, 57.36%, and 76.60% lesser than that of FF-GFS.

3) Varying the Number of Messages Generated in the Network

The number of messages generated by the regular nodes is varied and the impact of this variation on the studied performance metrics is analyzed. The results are captured in Fig. 9(a)-(d). In Fig. 9(a), it is observed that the number of messages delivered increases with an increase in the number of messages generated in the network. This is due to the fact that with an increase in the number of messages, the ferry will carry more messages towards the destination, which results in a higher rate of message delivery. In addition, HVF has more number of messages delivered than GFS and the performance of HVF in case of ferry failure is comparable to that of GFS in the normal operational mode. In Fig. 9(b), it is observed that the overhead ratio increases with an increase in the number of messages generated in the network. This is because of the increase in the number of messages flowing in the network (i.e., the number of relayed messages). The overhead ratio of GFS is much higher than that of HVF because of the use of epidemic routing in GFS to pass the messages between the nodes. In Fig. 9(c) (resp. Fig. 9(d)), it is observed that the average latency (resp. average buffer time) increases with an increase in the number of messages generated in the network. This is justified by the fact that due to the increase in the network traffic, some messages cannot be delivered immediately to the destination and need to be kept in the buffer of the nodes, resulting to an increased latency and buffer time, with a smaller rate of increment when the number of generated messages is below 600 messages and a larger rate of increment when the number of generated messages is beyond 600. HVF outperforms GFS in terms of average latency and average buffer time in both the normal and the ferry failure situations.

N-HVF performs 35.62% better than N-GFS whereas FF-HVF performs 120.86% better than FF-GFS in terms of number of messages delivered. The overhead ratio, average latency, and average buffer time of N-HVF are 59.87%, 29.92%, and 15.93% lesser than that of N-GFS respectively whereas for FF-HVF, these values are 59.48%, 56.93%, and 23.53% lesser than that of FF-GFS.
4) Varying the Node Speed

The speed of regular nodes is varied and the impact of this variation on the studied performance metrics is analyzed. The results are captured in Fig. 10(a)-(d). In Fig. 10(a), it is observed that as the speed of nodes is increased, the number of messages delivered increases as well. This is due to the fact that with an increase in node speed, the nodes move faster in the network and can come in contact with a ferry more frequently, resulting in more number of messages delivered to the destination. HVF has more number of messages delivered compared to GFS in both normal and ferry failure
scenarios. In Fig. 10(b), the overhead ratio decreases with an increase in the node speed. This is due to an increase in the number of delivered messages, which results in lesser number of messages relayed in the network. In addition, the overhead ratio of HVF is lesser than that of GFS. In Fig. 10(c) (resp. Fig. 10(d)), it is observed that the average latency (resp. average buffer time) decreases with an increase in the node speed. This is attributed to the fact that with an increase in node speed, that nodes can meet a ferry very frequently and pass their messages to it, thereby the message delivery time and buffer occupancy of the nodes get reduced. It is also be observed that the average latency and average buffer time are not affected much by the node speed as the rate of decrement is not very high for these two parameters. N-HVF performs 13.06% better than N-GFS whereas FF-HVF performs 64.45% better than FF-GFS in terms of number of messages delivered. The overhead ratio, average latency, and average buffer time of N-HVF are 54.49%, 32.96%, and 56.84% lesser than that of N-GFS respectively whereas for FF-HVF, these values are 68.08%, 41.78%, and 68.65% lesser than that of FF-GFS.

5) Varying the Message Size

The message size is varied and the impact of this variation on the studied performance metrics is analyzed. The results are captured in Fig. 11(a)-(d). In Fig. 11(a), it is observed that the number of messages delivered decreases with an increase of in the message size. This is due to the fact that with an increase in the message size, the ferry will carry less messages in its buffer, and thus will deliver less number of messages to the destination. In Fig. 11(b) (resp. Fig. 11(c)), it is observed that the overhead ratio (resp. average latency) increases with an increase in the message size. This is attributed to the observed decrease in the number of messages delivered. The average latency and average buffer time initially do not vary much with an increase in the message size from 1Mb to 3Mb. However, with further increase in the message size, these parameters increase at a rapid rate. This is justified by the fact that with further increase in the message size, the ferry will not be able to deliver the complete message to the destination in one go. It will deliver the message to the destination in the second round, which results in an increased average latency and increased average buffer time. Fig. 11(a)-(d) also show that HVF outperforms GFS in terms of all studied performance metrics when the message size is varied.

N-HVF performs 92.20% better than N-GFS whereas FF-HVF performs 171.34% better than FF-GFS in terms of number of messages delivered. The overhead ratio, average latency, and average buffer time of N-HVF are 55.06%, 16.42%, and 25.82% lesser than that of N-GFS respectively whereas for FF-HVF, these values are 61.19%, 29.43%, and 47.88% lesser than that of FF-GFS.

6) Varying the Time to Live

The message time-to-live (TTL) is varied and the impact of this variation on the studied performance metrics is analyzed. The results are captured in Fig. 12(a)-(d). In Fig. 12(a), it is observed that with an increase in the message TTL from 100 to 200, the number of messages delivered increases. This can be attributed to the fact that with further increase in the message TTL, a message can live in the network for a longer period of time, resulting in more chances of successful message delivery and increased delivery probability. If the TTL is further increased from 200 to 300, no significant increase in the aforementioned performance metrics is observed. This is
due to the fact that messages have lived enough to reach the destination before they have to be deleted from the network. In Fig. 12(b), the overhead ratio decreases with an increase in the message TTL. This is justified by the fact that an increase in the message TTL yields an increase in the number of messages successfully delivered, resulting to a decrease of the overhead ratio. In Fig. 12(c) (resp. Fig. 12(d), it is observed that the average latency (resp. average buffer time) increases with an increase in the message TTL. This is justified by the fact that with an increase in the message TTL, the rate of message dropping due to TTL expiration decreases. Thus, messages can remain active in the network and can reach the destination after a longer period of time, resulting to an increase in the average latency and average buffer time. In Fig. 12(a)-(d), it is also observed that HVF outperforms GFS in terms of all performance metrics when the message TTL is varied. N-HVF performs 18.81% better than N-GFS whereas FF-HVF performs 65.95% better than FF-GFS in terms of number of messages delivered. The overhead ratio, average latency, and average buffer time of N-HVF are 50.26%, 17.54%, and 16.39% lesser than that of N-GFS respectively; whereas for FF-HVF, these values are 58.28%, 35.01%, and 24.27% lesser than that of FF-GFS.

VI. CONCLUSION

In this paper, an infrastructure-based protocol (so-called HVF scheme) that uses a combination of infostations and message ferries for message transfer in OppNets has been proposed. The source node gives the message to the infostation or ferry and the message then flows in the network hop-by-hop through a combination of ferries and infostations that act as intermediate nodes. The architecture used in this work is fairly simple and ensures a guaranteed delivery of the message as each square grid region is connected to other grid region with the help of ferries. It also uses a weight function to break the tie if multiple ferries are present in the same region to carry the message. This weight function is calculated on the basis of three parameters, namely power, distance, velocity, and buffer values of the ferry. The HVF scheme is found to be robust in case of ferry failure as the message can still reach the destination with the help of other horizontal and vertical ferries. The HVF scheme is scalable as more ferries can be added to the route to enhance the reliability. The ferries may also be synchronized in some other way while keeping the same network topology. The acknowledgement procedure used ensures that at least one copy of the message remains in the network till it reaches the destination. It also ensures that the message is finally deleted from the buffer of other nodes after being delivered successfully at the destination. Through simulations, the performance of HVF scheme is compared against that of the GFS scheme, showing that HVF significantly outperforms GFS in terms of number of messages delivered, overhead ratio, average latency, and average buffer time.

As future work, more scenarios can be considered by designing different and optimal ferry routes so as to cover the whole network and ensure the guaranteed delivery of the message. The security threats caused to the proposed HVF scheme can also be explored. Other parameters such as energy efficiency, infrastructure reduction, cost reduction, delay minimization, can also be considered in the design of an enhanced HVF scheme.

REFERENCES


