Technische Universität Ilmenau

Controlling the Mobility and Enhancing the Performance of Multiple Message Ferries in Delay Tolerant Networks

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

vorgelegt in der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von Herrn M.Sc. Mehdi Harounabadi
geboren am 07.07.1983 in Teheran

Datum der Einreichung: 28.02.2020

Datum der Verteidigung: 21.10.2020

Gutachter:

1. Professor Dr.-Ing. habil. Andreas Mitschele-Thiel,
Ilmenau University of Technology

2. Professor Dr. rer. nat. Jochen Seitz,
Ilmenau University of Technology

3. Professor Dr. Edison Pignaton de Freitas,
Federal University of Rio Grande do Sul

urn:nbn:de:gbv:ilm1-2020000472

Thanks

First and foremost I would like to thank my advisor Professor Andreas Mitschele-Thiel for the continuous support of my research, for his patience, guidance and encouragements. My sincere thanks also go to Professor Jochen Seitz from Ilmenau University of Technology and Professor Edison Pignaton de Freitas from Federal University of Rio Grande do Sul for their precious time reviewing my thesis and their insightful comments. I would like to express my deepest appreciation to all my colleagues in Integrated Communication Systems (ICS) group of TU Ilmenau.

Finally, my deep gratitude to my wife, Bita, for her continuous love, help and support. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am.

Kurzfassung

In einem drahtlosen Netzwerk mit isolierten und stationären Knoten können Adhoc und verzögerungstolerante Netzwerk Routing-Protokolle nicht verwendet werden. Message Ferry Netzwerke sind die Lösung für diese Fälle, in denen ein (oder mehrere) Message Ferry Knoten den store-carry-forward Mechanismus verwendet und zwischen den Knoten reist, um Nachrichten auszutauschen. In diesem Fall erfahren die Nachrichten für gewöhnlich eine lange Verzögerung. Um die Performance der Message Ferry Netzwerke zu verbessern, kann die Mobilität der Message Ferry Knoten gesteuert werden. In dieser Doktorarbeit werden zwei Strategien zur Steuerung der Mobilität der Message Ferry Knoten studiert. Die Strategien sind das on-the-fly Entscheidungsverfahren in Ferry Knoten und die offline Wegplanung für Ferry Knoten. Für die on-the-fly Strategie untersucht diese Arbeit Decision-maker in Ferry Knoten, der die Entscheidung auf Grundlage der lokalen Observation eines Ferry Knoten trifft. Zur Koordinierung mehrerer Ferry Knoten, die keine globale Kenntnis über das Netzwerk haben, wird eine indirekte Signalisierung zwischen Ferry Knoten vorgeschlagen. Zur Kooperation der Ferry Knoten für die Zustellung der Nachrichten werden einige Ansätze zum Nachrichtenaustausch zwischen Ferry Knoten vorgeschlagen, in denen der Decision-maker eines Ferry Knotens seine Information mit dem verzögerungstoleranten Router des Ferry Knoten teilt, um die Effizienz des Nachrichtenaustauschs zwischen Ferry Knoten zu verbessern. Umfangreiche Simulationsstudien werden zur Untersuchung der vorgeschlagenen Ansätze und des Einflusses verschiedener Nachrichtenverkehrsszenarien vorgenommen. Außerdem werden verschiedene Szenarien mit unterschiedlicher Anzahl von Ferry Knoten, verschiedener Geschwindigkeit der Ferry Knoten und verschiedener Ansätze zum Nachrichtenaustausch zwischen Ferry Knoten studiert. Zur Evaluierung der offline Wegplanungsstrategie wird das Problem als Multiple Traveling Salesmen Problem (mTSP) modelliert und ein genetischer Algorithmus zur Approximation der Lösung verwendet. Es werden verschiedene Netzwerkarchitekturen zur Pfadplanung der Ferry Knoten vorgestellt und studiert. Schließlich werden die Strategien zur Steuerung der Mobilität der Ferry Knoten verglichen. Die Ergebnisse zeigen, dass die Performance der Strategien in Bezug auf die Ende-zu-Ende-Verzögerung von dem Szenario des Nachrichtenverkehrs abhängt. In Szenarien, wie Nachrichtenverkehr in Sensor-Netzwerken, in denen ein Knoten die Nachrichten zu allen anderen Knoten sendet oder von allen anderen Knoten empfängt, zeigt die offline

Wegplanung, basierend auf der mTSP Lösung, bessere Performance als die on-the-fly Strategie. Andererseits ist die on-the-fly Stratgie eine bessere Wahl in Szenarien wie Nachrichtenaustausch zwischen Rettungskräften während einer Katastrophe, in denen alle drahtlose Knoten die Nachrichten austauschen müssen. Zudem ist die on-the-fly Strategie flexibler, robuster als offline Wegplanung und benötigt keine Initialisierungszeit.

Abstract

In a wireless network with isolated and stationary nodes, ad hoc and delay tolerant routing approaches fail to deliver messages. Message ferry networks are the solution for such networks where one or multiple mobile nodes, i.e. message ferry, apply the store-carry-forward mechanism and travel between nodes to exchange their messages. Messages usually experience a long delivery delay in this type of network. To improve the performance of message ferry networks, the mobility of ferries can be controlled. In this thesis, two main strategies to control mobility of multiple message ferries are studied. The strategies are the on-the-fly mobility decision making in ferries and the offline path planning for ferries. To apply the on-the-fly strategy, this work proposes a decision maker in ferries which makes mobility decisions based on the local observations of ferries. To coordinate multiple ferries, which have no global view from the network, an indirect signaling of ferries is proposed. For cooperation of ferries in message delivery, message forwarding and replication schemes are proposed where the mobility decision maker shares its information with the delay tolerant router of ferries to improve the efficiency of message exchange between ferries. An extensive simulation study is performed to investigate the performance of the proposed schemes and the impact of different traffic scenarios in a network. Moreover, different scenarios with different number of ferries, different speed of ferries and different message exchange approaches between ferries are studied. To study the offline path planning strategy, the problem is modeled as multiple traveling salesmen problem (mTSP) and a genetic algorithm is applied to approximate the solution. Different network architectures are proposed and studied where the path of ferries are planned in advance. Finally, the strategies to control the mobility of ferries are compared. The results show that the performance of each strategy, in terms of the average end-to-end delay of messages, depends on the traffic scenario in a network. In traffic scenarios same as the traffic in sensor networks, where only a single node generates messages to all nodes or receives messages from all node, the offline path planning based on mTSP solution performs better than the on-the-fly decision making. On the other hand, in traffic scenarios same as the traffic in disaster scenarios, where all nodes in a network may send and receive messages, the on-the-fly decision making provides a better performance. Moreover, the on-thy-fly decision making is always more flexible, more robust and does not need any initialization time.

Contents

1 Introduction

Nowadays, ad hoc networks find more applications due to their flexibility, fast deployment, and low cost. They are independent of network infrastructure and provide end-to-end communication through message relaying. The end-to-end path of a message flow is established in a self-organized manner by the cooperation of existing nodes in a network. The nodes can be mobile or stationary concerning the application of such networks. Wireless sensor networks, vehicular ad hoc networks, and post-disaster networks are some examples of ad hoc networks. In some cases, ad hoc networks face challenges to provide end-to-end paths for message flows due to the mobility of nodes, bad channel quality, and isolation of nodes. This kind of networks are called ”challenged networks” [1], [2]. However, the term ”challenged networks” does not refer only to ad hoc networks but any type of networks which fail to provide end-to-end paths for message flows. Messages face a long delivery delay and a high drop rate in challenged networks. Therefore, only delay tolerant messages can be delivered in this type of network.

Delay Tolerant Networks (DTN) are challenged networks where messages are delivered to their destinations experiencing long delays. The routing protocols of ad hoc networks such as Ad hoc On-demand Distance Vector (AODV) [3] and Optimize Link State Routing (OLSR) [4] fail in such cases while no end-to-end paths can be found for message flows. Ad hoc routing protocols find a (multiple) path(s) from the source to the destination of messages, initially and then utilize the path as long as it is available. In case of any disconnection, they try to discover a new route. In a DTN, it may be impossible to find any route for a message. For this reason, different types of routing protocols are used in DTNs. The store-carry-forward paradigm is the key enabler for message delivery in DTNs and employed by DTN routing protocols. In a DTN, nodes do not discover the end-to-end path from the source to the destination of a message. They decide to forward a message or a replication of it when they visit other nodes. In case of no connectivity, nodes store messages and carry them until meeting a new node. Therefore, the success of store-carry-forward mechanism to deliver messages depends on the mobility of nodes. It should be noted that messages must tolerate long delays while they usually have to wait in the buffer of nodes for a long time. There are a number of scenarios where DTNs can be employed. Some of the scenarios to apply DTNs are as follows:

Sparse sensor networks: sensor nodes are placed sparsely to take measurements from their environment in mountains, forests, etc. The sensor data are collected in a sink node for further analysis. Placing relay nodes to forward the messages from sensor nodes to the sink or a direct communication of sensors with the sink is not possible in sparse networks. For this reason, messages must wait in the buffer of sensors to have a new contact, which is a wireless node. When the contact is available, messages can be forwarded to it.

Military networks: in military scenarios, network infrastructures cannot be established. Moreover, ad hoc networking faces many disconnections due to the mobility of nodes (soldiers and vehicles). In this case, nodes must keep their messages and carry them to find an opportunity to forward them. Applications with strict quality of service requirements should be refrained in military scenarios but some delay tolerant traffic like map data, mission commands, and text messages can be exchanged among nodes.

Post disaster networks: network infrastructures may be damaged and broken after natural hazards such as earthquakes, tsunamis, floods, etc. Rescue teams are sent to the spot to save lives. They need to communicate with each other or injured people. Ad hoc networking may fail due to the limited radio transmission range of nodes and their mobility. Store-carry-forward mechanism makes the communication possible with long delays. Having long delays, rescue teams can exchange messages and some vital information may be sent to the victims of a disaster.

In all of the above mentioned scenarios, nodes may be stationary (without mobility) or have only a limited mobility. The store-carry-forward mechanism relies on the mobility of nodes and fails if all nodes are stationary. To overcome this challenge in DTNs, mobile nodes are employed to collect data from disconnected nodes and carry them to their destinations. The mobile nodes are called "data collectors" and they are the solution for message exchange in such networks. They are mobile wireless nodes with a high capacity of memory. They travel among nodes, collect messages from them, carry and forward them. Data collectors provide the communication for isolated and stationary wireless nodes. Moreover, they can accelerate the message delivery in DTNs where nodes are not stationary. Different types of data collectors can be employed in a DTN such as ground and aerial data collectors. The aerial data collectors are usually faster because they face fewer physical barriers. On the other hand, the aerial data

Figure 1.1: A disaster scenario with multiple UAVs as message ferries

collectors face some limitations such as flight time (due to limited energy resources), reliability and capacity of memory.

Message ferry networks are DTNs where a message exchange between two nodes is possible only through data collectors. The data collectors are called also message ferries. A single message ferry can make the communication possible in a message ferry network. However, if there are long distances between nodes or high loads of messages, a long delay of message delivery may occur. To overcome this challenge, multiple message ferries can be employed. Message delivery in a network with multiple message ferries is faster, but efficient coordination of message ferries is required. Furthermore, the message delivery can be accelerated, if message ferries cooperate in message delivery by exchanging messages.

An Unmanned Aerial Vehicle (UAV) can be employed as a message ferry in disconnected networks. A UAV is a mobile wireless node which flies between isolated nodes and delivers their messages. Figure 1.1 illustrates a disaster scenario where rescue teams are sent to different spots and need to communicate with each other or receive/send some messages from/to a command center. UAVs are employed in this scenario to fly between rescue teams and the command center and deliver their messages. However, ground vehicles can be employed in such networks as message ferries, but the mobility of such ground vehicles is limited due to the natural obstacles.

1.1 Problem statement

The main problem in disconnected networks where nodes are isolated is the communication between nodes. A message ferry is a mobile wireless node which travels between disconnected wireless nodes and provides the communication between them. In some cases, there are performance requirements in addition to the basic communication between nodes. Based on the requirements in a disconnected network, single or multiple message ferries are employed to deliver messages. The mobility of message ferries is one of the effective factors on the delay of message delivery. Therefore, a new question arises: **What is the most efficient mobility strategy for message ferries?**

Random mobility of message ferries is the most straightforward solution to provide communication between isolated nodes, but it may impose an unbounded latency for the delivery of messages. The other option is to control the mobility of message ferries.

In this thesis, different strategies are studied to control the mobility of message ferries. Applying each strategy, a set of optimizations is performed and the results are investigated. Finally, the strategies to control the mobility of message ferries are compared. The main strategies to control the mobility of message ferries are as follows:

- **Static trajectory**: the travel path of message ferries is planned in advance applying this strategy. An optimization algorithm must run offline to find the optimal paths for all message ferries before they start their mission in the network. Different optimization objectives such as the traveled distance of message ferries or the delay of messages can be considered. After the path planning, the paths are given in message ferries and they only follow the given paths. A path planner is needed to plan the path of message ferries. This strategy is called offline path planning in this thesis.

- **Dynamic trajectory**: applying this strategy, message ferries are self-organized and decide the next wireless node to visit within their travel (on-the-fly). They do not follow any given path and there is no initial path planning. In this case, a message ferry visits an isolated (disconnected) node, exchanges messages with it and decides the next node to visit. Employing multiple message ferries, coordination of them is necessary to enhance the efficiency of decision making. Moreover, message ferries can cooperate in message delivery to accelerate a mission accomplishment. No central entity is needed in this case and message ferries make all decisions in a self-organized manner. Self-organized message ferries have no global view from the state of the network and make their decisions based on their

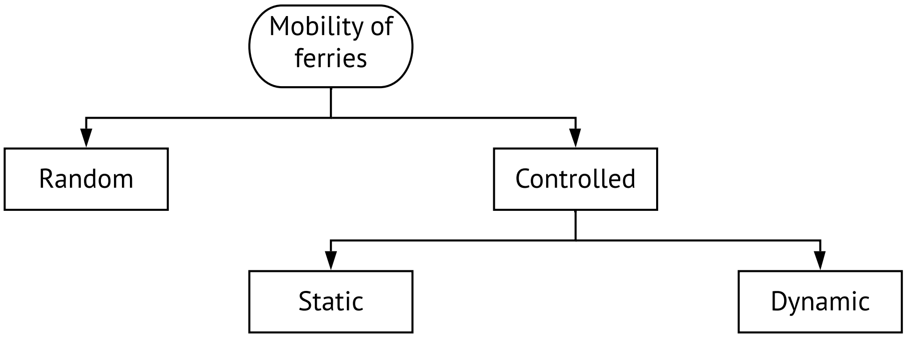


Figure 1.2: Classification of the mobility strategies of message ferries

local observations which they obtain from the network. This strategy is called on-the-fly decision making in this thesis.

Figure 1.2 shows different mobility strategies of message ferries.

This work answers several questions considering different strategies to control the mobility of multiple message ferries. The questions are as follows considering an on-the-fly mobility decision making in message ferries:

1. Which metrics must be considered for the on-the-fly mobility decision making of message ferries?
2. How can message ferries be coordinated having only a local observation?
3. How can message ferries cooperate to accelerate the delivery of messages?

Considering an offline path planning, following questions are answered:

1. How can the path of multiple message ferries be planned in a reasonable time?
2. How can the performance of message ferry networks be evaluated?
3. What is the best network architecture when the offline path planning is applied?

And finally, the main question by comparing both strategies is:

- What is the best strategy to control the mobility of message ferries?

In the next section, the contributions of this thesis are described which consists of a set of studies, optimizations, and schemes to answer the above-mentioned questions.

1.2 Contribution

The contributions of this thesis can be classified into three main branches with respect to the strategies to control the mobility of message ferries.

In the first branch, a set of algorithms and schemes are proposed and studied considering the on-the-fly mobility decision making in self-organized message ferries. The proposed approaches in this branch are as follows:

- An on-the-fly mobility decision maker in message ferries which works only based on the local observations of message ferries
- A self-organized mechanism for coordination of message ferries to avoid redundancies
- Message forwarding and replication schemes for cooperation message ferries to accelerate delivery of messages

In the second branch, the offline path planning for message ferries is considered where message ferries follow a static trajectory. In this branch, the path planning problem and the network architecture to apply the planned path of message ferries are the main concerns. The contributions of this thesis for this branch are as follows:

- Modeling the path planning problem as multiple Traveling Salesman Problem (mTSP)
- Applying an evolutionary algorithm as a heuristic to plan the path of message ferries
- A performance metric to evaluate the planned path of message ferries and the network architecture

Finally, the strategies to control the mobility of message ferries are compared in the third branch.

Figure 1.3 shows the three branches of studies in this thesis.

1.3 Structure

In Chapter 2, the background knowledge is recalled and the literature is reviewed for message delivery in ad hoc and delay tolerant networks. Message ferry networks are explained in detail and different trajectory types of message ferries such as the static

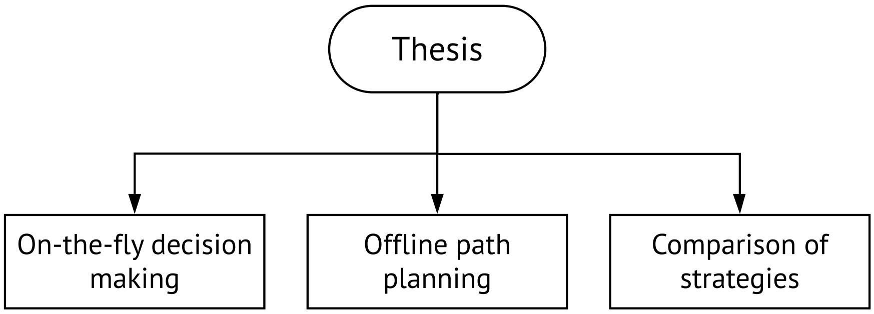


Figure 1.3: Main branches of studies in the thesis

and dynamic trajectories are reviewed from the state of the art. The network model and main assumptions are defined in Chapter 3. In Chapter 4, the on-the-fly mobility decision maker of message ferries is proposed. Different metrics to apply in the mobility decision maker are studied. The chapter proposes the concept of stigmergy in the form of an indirect signaling between message ferries for their coordination. The main assumption in this chapter is the absence of any message exchange between message ferries. A message is delivered to its destination by the message ferry which collects the message from its source node. Chapter 5 proposes a set of schemes for the cooperation of self-organized message ferries in message delivery. The cooperation of message ferries is based on the direct or indirect exchange of data messages between them to establish a multi-hop communication and accelerate the delivery of messages. The direct message exchange occurs in the form of a message forwarding or replication when message ferries meet each other. The difference between forwarding and replication of a message is on the number of instances of the message in a network. By message forwarding, a single instance of a message exists in a network, but there can be multiple copies of a message employing message replication. First, the direct message exchange between message ferries is studied. Then, an indirect message exchange through nodes is proposed for scenarios where message ferries do not meet each other to exchange messages directly.

Figure 1.4 shows the structure of studies in the branch of ”on-the-fly decision making” in Chapters 4 and 5. The studies are classified into two main classes with respect to the existence or absence of the message exchange between message ferries.

The offline path planning as the other strategy to control the mobility of message ferries is presented in Chapter 6 and a genetic algorithm is applied to find the path of multiple message ferries. Several architectures for message ferry networks are studied where the offline path planning is applied.

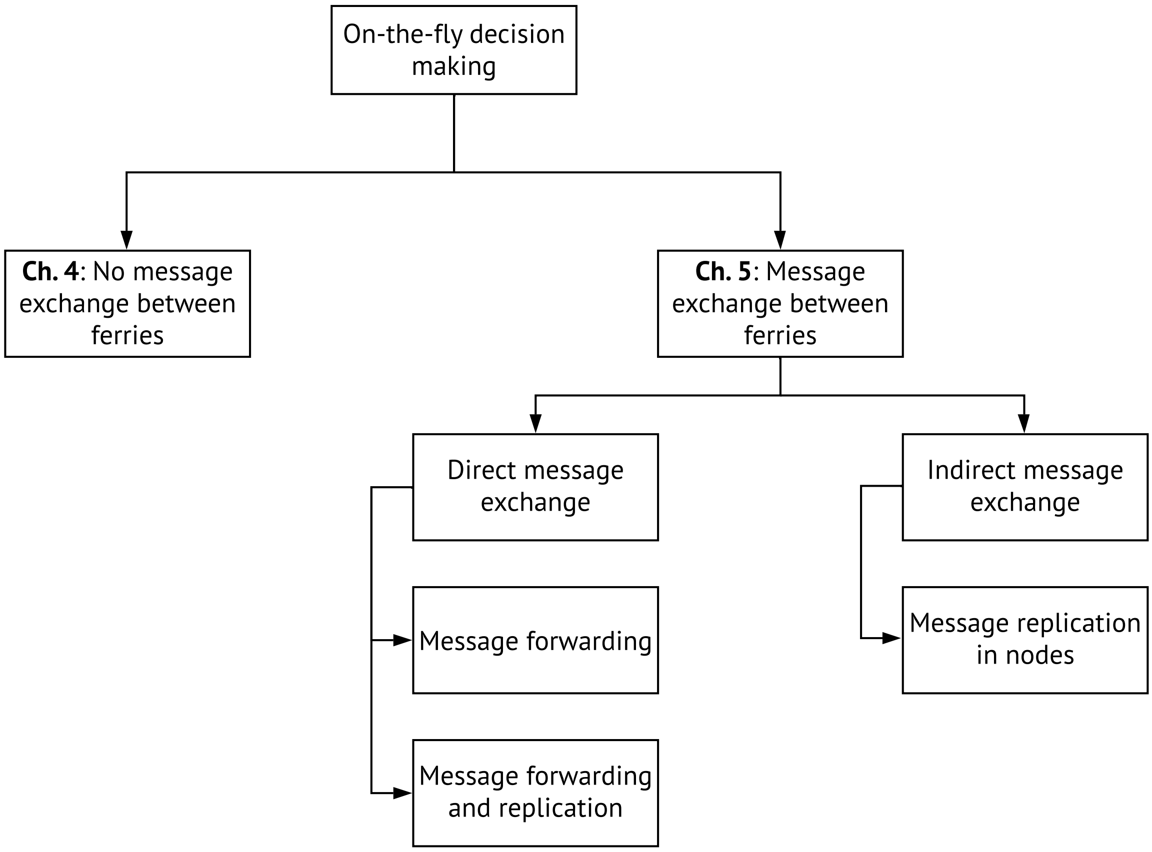


Figure 1.4: Classification of studies on self-organized message ferries with an on-the-fly mobility decision maker

Chapter 7 compares the on-the-fly and offline strategies to control the mobility of message ferries. Finally, Chapter 8 is the summary of the thesis, the most important lessons learned from the studies and suggestions for future work.

1.4 Publications

Several scientific contributions of this thesis have been published in peer reviewed journals and conferences as follows:

- Journal papers
  - The indirect signaling mechanism for coordination of self-organized message ferries with a dynamic trajectory was published in ACM/Springer "*Mobile Networks and Applications*" [5].

  - The direct message forwarding and replication schemes for cooperation of self-organized message ferries were published in ACM/Springer "*Mobile Networks and Applications*" [6].

- Conference papers
  - The on-the-fly mobility decision making algorithm in message ferries was published in *Adhocnets* conference and received the **best paper award** [7].
  - The cooperation of self-organized message ferries in the form of message forwarding between them was published in *Adhocnets* conference [8].
  - The genetic algorithm for offline path planning of multiple message ferries was published in *IEEE PIMRC* [9].
  - Network architectures to apply the planned path of message ferries and their comparison were published in *IEEE ICUFN* [10].

2 Background and State of the Art

In this chapter, the required background knowledge about wireless ad hoc networks, delay tolerant networks and message ferry networks are recalled. The state of the art is reviewed for message ferry networks since they are the main focus of the current thesis.

2.1 Wireless ad hoc networks

Wireless ad hoc networks are self-organized wireless networks consisting of autonomous nodes that can make decisions independently to deploy a wireless network without the support of any infrastructure and the need for any central administration. Each node communicates directly with other wireless nodes in its neighborhood through the wireless link. Moreover, a node can communicate with distant nodes applying a multi-hop communication. All ad hoc nodes act as routers and forward messages between the source and destination of a message.

The wireless ad hoc networks with mobile nodes are called Mobile Ad hoc NETworks (MANETs) [11] where each node decides about forwarding a message based on its routing table or discovers a new route to the destination of the message. MANETs have several applications such as Vehicular Ad hoc NETworks (VANETs) [12], [13] where vehicles are wireless ad hoc nodes and need to exchange messages between each other or with a roadside unit having a stringent quality of service requirement, battlefield networks [14] where a fast deployment of the network is needed, post-disaster networks [15], [16] where the communication is a crucial requirement even with a high latency in delivery of messages, and wireless sensor networks [17], [18], [19] where limited resources and capability of nodes are main challenges.

Ad hoc networks are self-organized, flexible and quickly deployable. However, they face several challenges in terms of addressing of nodes, radio access control, routing of messages, energy management, etc. The mobility of nodes in MANETs leads to a dynamic topology of the network and causes a non-deterministic behavior of wireless links. Besides, the lack of a central entity in ad hoc networks makes the message routing procedure more complicated than conventional wireless networks. In this section, routing protocols in the network layer of ad hoc nodes are reviewed to have an insight on message delivery schemes in this type of network.

Generally speaking, there are two main categories of routing protocols in ad hoc networks which are different in terms of the information they need to build a route or forward a message. In the following sections, the categories are explained briefly.

2.1.1 Topology based routing

Topology based routing protocols in ad hoc networks utilize topology information of a network to establish a route between the source and destination of a message. To discover a route between two nodes, control messages are flooded to obtain the topology information of a network. There are sub-classes of topology based routing protocols which perform differently to maintain a routing table in ad hoc nodes and they are as follows:

- Proactive: each node maintains a routing table containing routes to all other nodes (destinations) in the network. Proactive routing protocols run the route discovery procedure periodically to keep all the routes up-to-date. They generate a big amount of overhead in the network. However, there are always ready to use routes to all destinations. In ad hoc networks, traditional proactive routing protocols from wired networks are adapted to limit the overhead of periodic route discoveries [20], [4], [21], [22].

- Reactive: nodes do not start the route discovery periodically. Whenever a new route is needed, the source node initiates the route discovery procedure to find the route. For this reason, the reactive routing protocols are also called on-demand protocols. A discovered route is valid only for a limited time due to dynamics in the topology of the network. The overhead is less than proactive routing protocols, but there is an initial delay for route discovery [3], [23], [24], [25], [26].

- Hybrid: the idea of proactive and reactive routing protocols are applied in this class of routing protocols. A network is divided into zones where a proactive approach is applied in them and routes between zones are discovered on-demand [27].

2.1.2 Position based routing

In this category of routing protocols, the message forwarding is performed based on position (geographical) information of nodes. No end-to-end route is established between the source and destination of a message. There is a per-hop behavior to choose the next hop for a message.

To apply a position based routing protocol, each node needs to know its position using Global Positioning System (GPS) or any other positioning approach such as [28], [29], [30]. Nodes exchange beacon messages periodically to inform their neighbors about their current location. Moreover, a location server is needed in the network to obtain the location information of a destination node.

Greedy Perimeter Stateless Routing (GPSR) [31] is the most well-known position based routing protocol where a node forwards a message to the neighbor node which is the closest to the destination of the message. There are several routing protocols in the literature such as [32], 33, [34] which each of them defines a set of conditions to choose the next hop for a message. The main challenge in position based routing protocols is the local minima problem where a node has no possibility to forward the message to a neighbor which is closer than itself to the destination. An extensive survey on position based routing protocols can be found in 35.

2.1.3 Ad hoc routing protocols in disconnected networks

Neither topology based routing protocols such as proactive, reactive or hybrid protocols, nor position based routing protocols can be employed directly in networks with disconnected typologies.

Topology based routing protocols need a connected topology for a route discovery. In case of a link failure, a new route is established. If the route establishment takes longer than usual, the message is discarded from the buffer of a node.

Position based routing protocols have per-hop behavior and do not establish an end-to-end route. The case of a disconnected network for a node that runs a position based routing protocol is similar to the local minima problem. However, in the case of local minima, nodes try to find a new neighbor to forward the message, even without making any progress in message forwarding toward the destination. If the node fails to find a neighbor, the message is discarded from the buffer of the node due to its lifetime.

Therefore, ad hoc routing protocols cannot be applied without modifications to networks where nodes are isolated and there is no route between the source and destination of a message. However, some of the features of ad hoc routing protocols such as per-hop behavior of position based routing protocols can be utilized in disconnected networks.

2.2 Delay tolerant networks

Delay Tolerant Networking (DTN) [36] is a solution for challenged networks where nodes are only intermittently connected. In DTNs, the multi-hop delivery of messages is not possible in the same way as ad hoc networks. Mobility is a challenge for routing protocols in ad hoc networks. However, it is taken as an opportunity in DTNs and exploited for message delivery. The applied mechanism in DTNs is the store-carry-forward. Applying store-carry-forward, each node stores messages, also called bundles, in its memory and carries them until a new contact comes to its radio transmission range. Then, the node may forward messages or again keep them in its buffer.

Zebranet [37] to monitor the wildlife, Daknet [38] to provide internet access in isolated regions and delay tolerant sensor networking for a roadside noise monitoring [39] are some applications of delay tolerant networking.

While a node carries a message and contacts a new node, there are three possibilities for the message:

- Don't forward: a node decides to do nothing and keeps the message in its buffer.
- Replicate: a copy of the message is sent to the new contact. The (carrier) node keeps the message after sending it.
- Forward: the message is sent to the new contact and is discarded from the memory (of carrier node) after forwarding.

The most simple case for delay tolerant routing is the direct delivery [40] where a node stores a message in its buffer and carries it until visiting the message destination. This approach neglects other nodes than the message destination and does not forward the message to any node except the destination. It does not generate any overhead in the network but may impose long delays to the delivery of messages. A node behaves the same as the first and the third above-mentioned possibilities for a message.

Regarding the second possibility i.e. message replication, there are several works in the literature. [41] is an epidemic routing for DTNs where a node replicates a message in all new contacts. The epidemic routing arises flooding of a message in a network. This strategy leads to the fastest delivery of the message but consumes lots of resources in the network. [42] performs similar to the epidemic routing but limits the number of message replications in a network to limit the overhead.

Another approach for DTN routing protocols is to select a contact or set of contacts to replicate a message. In this approach, a utility value is calculated for each contact using the knowledge which a node has (or obtains) about the network. The knowledge can be:

1. Historical information: PROPHET [43] is the most well-known utility based routing protocol which employs historical knowledge of nodes to calculate the encounter probability for a new contact. The encounter probability is calculated using a utility function and it is the probability for a node to visit the destination of a message. PROPHET replicates a message to a new contact if it has a higher probability to visit the destination of the message than the carrier node. [44], [45], [46], [47], [48], [49] also employ historical knowledge in their decision making to choose contacts to replicate a message.

2. Context information: in [50], [51] and [52] context information like speed, mobility, direction, residual energy, etc. are applied instead of historical information to calculate the utility value of a new contact.

3. Social information: is another type of information that is applied to find the social ties between nodes. The nodes with higher degrees of social ties are more probable to visit each other. [53], [54], [55], [56] apply social knowledge of nodes in their utility function.

An extensive literature review about routing protocols in delay tolerant networks can be found in [57], [58], [59], and [60].

2.3 Message ferry networks

Connectivity of nodes is the main requirement to apply ad hoc routing protocols for a multi-hop communication between two nodes. In the case of a disconnection, an ad hoc routing protocol fails to deliver messages. DTN routing protocols, which exploit the mobility of nodes for delivery of messages, cannot deliver messages if all nodes are stationary. In such scenarios, the solution is to employ mobile nodes to travel in the network and deliver messages between stationary nodes. There are different terminologies for the mobile nodes that are ”messages ferries”, ”data ferries”, ”data mules”, and ”data collectors”. In this work, they are called ”ferries”. A network where messages are delivered employing a ferry is called a ”message ferry network”. In a message ferry network, same as DTNs, disconnected nodes have to store messages when no connection is available. A message waits in the buffer of a node until a ferry visits the node. While visiting a node, the ferry collects all the waiting messages in the node. Then, messages wait in the ferry buffer. When the ferry visits the destination, messages are delivered to the destination node.

A ferry is employed to utilize the store-carry-forward mechanism in a DTN with stationary nodes.

The mobility of a ferry is an important and influencing parameter on the performance of a message ferry network [7]. A ferry can have a random or a controlled mobility. A ferry with a random mobility makes the communication in a DTN with stationary nodes possible, but no upper bound can be realized for the latency of messages. A message ferry network is a DTN and messages have to tolerate long delays. However, reducing the delay of message delivery means the better performance of the network. A solution to reduce the message delivery delay in message ferry networks is to control the mobility of ferries. There are two main strategies to control the mobility of a ferry (or ferries). In one strategy, the path of a ferry is planned in advance (offline path planning) and the ferry follows a static path. In the other strategy, a ferry (or ferries) does not follow a planned path, but it decides about its mobility within its travel (on-the-fly decision making).

2.3.1 Static path of ferry

The idea of message ferrying for disconnected stationary nodes was proposed in [61]. They assumed that nodes are stationary and their location is known. They modeled the problem of path planning for a ferry to visit the stationary disconnected nodes as the Traveling Salesman Problem (TSP).

In TSP, the goal is to find the shortest path for a salesman to visit a set of given cities. The constraint of TSP is that the salesman must visit each of the cities exactly once. Figure 2.1 illustrates the TSP problem and a solution for it. The salesman starts its tour from city C_1 and follows the shortest path to visit all cities and returns back to node C_1. TSP has several applications such as scheduling, circuit wiring [62], gene ordering [63] and many more [64]. TSP is a NP-hard problem and its solution is usually approximated using heuristic algorithms such as ant colony [65], genetic algorithm [66] [67] [68], particle swarm optimization [69] or hybrid approaches [70].

The authors of [61] found a path for a ferry using the TSP solution. The objective

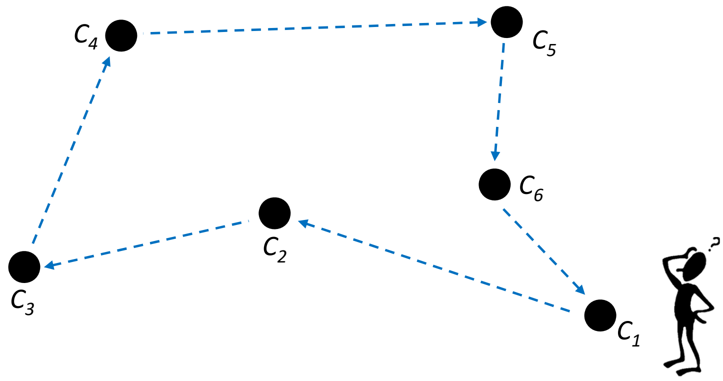


Figure 2.1: A solution for TSP

in TSP is to minimize the traveled distance of a ferry to visit all nodes. To minimize the delay of message delivery, they applied the 2-opt algorithm [71] which does a local search by swapping the order of nodes in a path.

Another work for path planning of a ferry in disconnected networks with stationary nodes is [72]. The location of nodes is also known in their path planner. They modeled the problem of path planning for the ferry as TSP similar to [61]. To find the solution of TSP, they considered the buffer overflow in nodes. If there is a buffer overflow in some of the nodes, they can be visited more than once by the ferry in a tour.

[73] employs a ferry in sensor networks as the sink node to collect the measured values from sensor nodes. This work addresses the buffer overflow problem in sensor nodes by clustering the nodes using a k-dimensional tree algorithm based on their buffer overflow probability and their location. A TSP solution is found for each cluster of nodes and TSP paths are concatenated to build an overall path for the ferry.

Single ferry networks face several challenges such as limited resources of a ferry and long latency of messages. In the case of wide networks, the long travel time of a ferry will cause a long waiting time of messages. To overcome challenges in message ferry networks, multiple ferries can be employed in message ferry networks. Ferries can cooperate in message delivery and improve the performance of communication. However, an efficient mechanism for coordination and cooperation of ferries is required to optimally exploit their potentials.

Several network architectures were proposed in [74] for networks with multiple message ferries. The architectures are different in terms of the cooperation of ferries. In their work, a path or paths are planned for ferries. Ferries travel either the same path with a time difference (time shift) or travel different paths. In the case of dif-

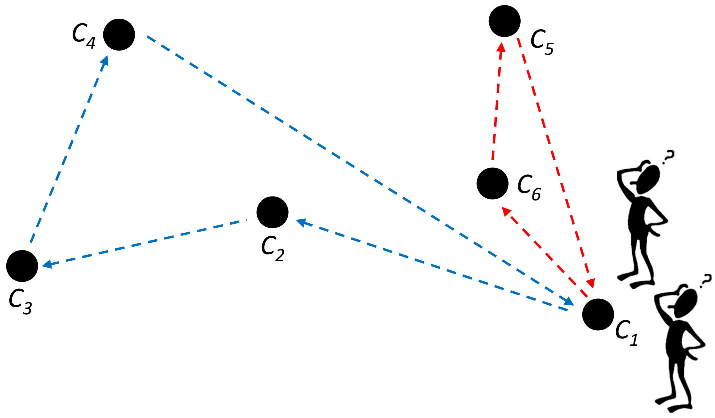


Figure 2.2: A solution for mTSP

ferent paths, several clusters of nodes are built where the inter-cluster communication takes place through direct exchange of messages between ferries or through a (or several) relay node(s). The focus of their work was to compare different architectures of multi-ferry networks.

To plan the path for multiple ferries, the problem of path planning can be modeled as a multiple Traveling Salesman Problem (mTSP).

mTSP is a combinatorial optimization problem [75] and a generalization of TSP. It aims to find the travel path of multiple salesmen to visit a given set of cities. Each salesman visits a subset of cities and each city must be visited exactly once by a salesman. However, all salesmen start/finish their travel in the same city and this city is visited by all salesmen. The objective in mTSP is to find travel paths of salesmen in such a way which the total traveled distance of them is minimum. Figure 2.2 shows a solution for mTSP having 6 cities and 2 salesmen. mTSP is formulated as follows [75]:

$$\min \quad \sum_{i}^{N} \sum_{j}^{N} x_{ij} d(i,j) \tag{2.1a}$$

$$\text{subject to} \quad \sum_{j=2}^{N} x_{1j} = m \tag{2.1b}$$

$$\sum_{j=2}^{N} x_{j1} = m \tag{2.1c}$$

$$\sum_{i=1, i \neq j}^{N} x_{ij} = 1, j = 2, ..., N \tag{2.1d}$$

$$\sum_{j=1, i \neq j}^{N} x_{ij} = 1, i = 2, ..., N \tag{2.1e}$$

where N is the set of cities, $d(i,j)$ is the euclidean distance between city i and j, x_{ij} is a binary variable which is 1 when a path from the city i to j exists in the solution and m is the number of salesmen. The constraints in 2.1b and 2.1c are about the starting city where m salesmen start/finish their tour from/in that city. The constraints 2.1d and 2.1e limit the number of visits to each city to exactly 1 visit.

There are several applications for mTSP such as scheduling [76] [77] [78], workforce balancing [79], robot and machine path planning [80] [81] and vehicle routing problem [82]. mTSP is even more complex than TSP. It is only possible to find a solution for a mTSP with small scales. Therefore, heuristics such as neural networks [83], genetic algorithm [84], ant colony [85], tabu search and simulated annealing [86] are applied to approximate the optimal solution for mTSP.

The authors in [87] and [88] proposed solutions for path planning of multiple data collectors in sensor networks. All sensor nodes generate messages for a sink node. The mobile data collectors visit sensor nodes and deliver their messages to the sink node. Each data collector visits a subset of nodes and the goal is to minimize the traveled distance of mobile data collectors same as mTSP.

2.3.2 Dynamic path of ferry

Another strategy to control the mobility of ferries is to decide their trajectories dynamically. In [89] and [90], authors proposed a dynamic trajectory planning for multiple ferries that serve several mobile disconnected nodes. They assume that ferries have only a local observation from the state of the network. The state of a network is defined by the location of nodes and ferries. They modeled the problem with Partial Observable Markov Decision Process (POMDP) [91] which is the generalization of MDP. MDP is used to model a decision making problem where the outcome of an action by a decision maker is partly random and partly under the control of the decision maker. The decision maker takes an action and observes the outcome of its action. In their work, the mobility decision in ferries is based on the probabilities for the mobility of nodes. The probabilities are calculated based on the observations of a ferry from the last location of nodes.

[92] and [93] assume that isolated nodes are mobile, but ferries have a full observation about the actual position of nodes through a location management system or an out-of-band communication with a central entity. The full observation in a ferry is also assumed in [94]. They assume stationary nodes but the message generation in nodes is dynamic. A ferry has a full observation of the buffer of nodes. The problem of mobility decision of a ferry is formulated as Markov Decision Process (MDP) [95] and is solved by a reinforcement learning algorithm [96].

Due to the exponential increase in the state space of MDP, authors in [97] applied a solution for a ferry visit to the stationary isolated nodes based on the deficit round-robin algorithm which is applied in the queue scheduling problem [98]. They assume a full observation of ferry on the buffer of nodes.

An on-the-fly decision making in a ferry was proposed in [99] and [100] to control the mobility of the ferry. They assume stationary nodes and a local observation in the ferry. The ferry observes the state of the network when it visits a node. There are no cyclic visits to nodes by the ferry. The ferry visits a node and gains a new observation about the messages in the node. Then, it selects another node to visit.

2.4 Conclusion

This chapter explained different approaches for message delivery in ad hoc networks, DTNs and message ferry networks.

In wireless networks, ad hoc networking is applied if the network infrastructure is not available. Nodes cooperate in message forwarding and act as routers. The multi-hop communication in ad hoc networks needs a fully connected network topology. In the case of any disconnection, the multi-hop delivery of messages is not applicable anymore. By frequent disconnections in a network, DTN routing protocols can exploit the mobility of nodes to deliver messages. In the worst case, where networks are fully disconnected and nodes are not mobile, neither ad hoc routing, nor DTN routing

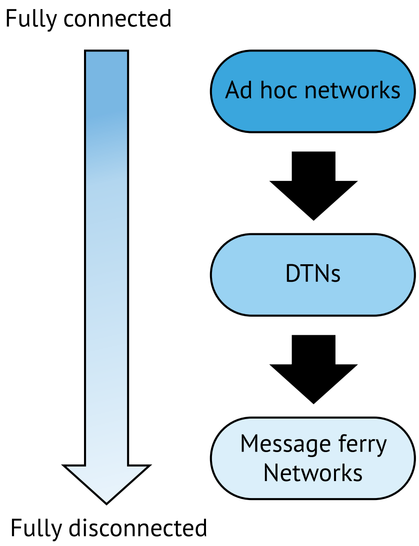


Figure 2.3: Connectivity level and wireless networking solutions

Table 2.1: List of parameters in the comparison

Abbreviation	Definition
S/D	**S**tatic or **D**ynamic route of ferry
Trm (K/U)	**Tr**affic of **m**essages (**K**nown/**U**nknown)
Nm (St/Mo)	**N**odes **m**obility (**St**ationary or **Mo**bile)
MF/SF	**M**ulti or **S**ingle **F**erry network
Obs (Fu/Pa)	**Obs**ervation (**Fu**ll/**Pa**rtial)
Ce	**Ce**ntral entity in the network

protocols can be employed. The only solution for such cases is a message ferry network. Figure 2.3 shows the relation between the connectivity level in a network and the possible networking solution for each level.

Table 2.1 defines the parameters of the comparison in Table 2.2 which existing work for message delivery in message ferry networks are compared. Different metrics are taken into consideration for the comparison which are shown by abbreviations.

Looking at the comparison table, the following points can be concluded:

- [61], [72], [72], [73], [87] and [88] were proposed for single ferry or multi-ferry networks, but the path of a ferry (or ferries) are static and cannot are not adapted based on the dynamics of a network such as changes in the message generation rate in nodes or changes in the traffic flows.

- [89] and [90] are message ferry approaches with dynamic paths of ferries and mobile isolated nodes. The path of ferries is adapted based on the mobility of nodes and a prediction of ferries about their mobility. They assume that the traffic of messages is constant and known for ferries, but ferries have only a partial observation about the location of nodes.

- [92], [93], [94] and [97] assume a full observation in a ferry to adapt the path of a ferry. They assume a long range communication in ferries and nodes to have such observations.

- [99], [100] propose an on-the-fly decision maker in a ferry to decide the next node to visit dynamically. The works were proposed for single ferry networks where a ferry has only a partial observation about the traffic model. However, mechanisms for coordination of ferries and cooperation among them in message delivery has not been studied in them.

This work studies the main strategies to control the mobility of ferries. For the on-the-fly decision making strategy, this work studies and proposes a self-organized multi-ferry network where nodes are stationary. Ferries have only a local observation from the state of the network. The state of a network is defined by the number of messages in nodes at a given time, location of nodes, and location of ferries. The traffic model is not known by ferries. They obtain a local observation from the state of the network when they visit a node. The only global knowledge in ferries is the location of nodes which is given in all ferries. Ferries cooperate with each other in message delivery and are coordinated through a self-organized mechanism. For the offline path planning strategy, this work applies the genetic algorithm to plan the path of ferries. With the studies on both strategies, this work compares the strategies under different scenarios. In the next chapter, the network model and main assumptions of this work are explained.

Table 2.2: Comparison of existing work

Scheme	S/D	Trm	Nm	MF/SF	Obs	CE	Goal	Remarks
[61]	S	K	St	SF	Fu	✓	Traveled distance + delay	TSP based
[72]	S	K	St	SF	Fu	✓	Traveled distance, avoiding buffer overflow	TSP based
[73]	S	K	St	SF	Fu	✓	Avoiding buffer overflow	Clustering
[74]	S	K	St	MF	Fu	✓	Evaluation of architectures	-
[87], [88]	S	K	Mo	MF	Fu	✓	Total traveled distance	mTSP, sensor network all to sink scenario
[89], [90]	D	K	Mo	MF	Pa	✗	Based on mobility of nodes	POMDP
[92], [93]	D	K	Mo	MF	Fu	✗	Based on mobility of nodes	Out of band communication of ferries to find nodes
[94]	D	K	St	MF	Fu	✗	Minimize delay	MDP, Reinforcement learning
[97]	D	K	St	MF	Fu	✗	Minimize delay	Deficit round robin based solution
[99], [100]	D	U	St	SF	Pa	✗	Minimize delay	On-the-fly decision making

3 Network Model and Assumptions

This chapter describes the main assumptions and network model for the studies in this thesis. There are assumptions in our studies which help us to investigate the proposed strategies and find answers to our questions.

3.1 Message ferry network

A message ferry network consists two types of wireless nodes (N); isolated nodes ($R \subset N$) and message ferries ($F \subset N$).

$$System = (R \subset N) \cup (F \subset N) | R \cap F = \emptyset \tag{3.1}$$

Isolated nodes, that are called "nodes" in this work, are wireless transmitters/receivers which are carried by rescue teams or integrated in sensor nodes. Nodes are located far from each other such that they cannot communicate. There is no infrastructure such as access points, base stations, equipment for satellite communications or any wired networks. Due to the isolation of nodes in the modeled network, ad hoc networking is not applicable, either.

Nodes have no mobility or their mobility is neglectable. Thus, the store-carry-forward mechanism cannot be applied to have a delay tolerant network. Though messages can tolerate a delivery delay for several minutes or more, the delivery delay of messages in such networks is non-deterministic.

Messages ferries, that are called "ferries" in this thesis, are mobile wireless nodes that travel among nodes and deliver their messages. Ferries employ the store-carry-forward mechanism to exchange messages between nodes. They travel always with a constant velocity. For simplicity, we neglect any acceleration and deceleration in the mobility of ferries. Moreover, there is no physical obstacle to limit ferries to travel directly between nodes. Based on the assumptions for the ferries, the mobility of ferries is controlled. To control the mobility of ferries, either a path is planned offline for each ferry or ferries decide their trajectories on-the-fly. Moreover, a ferry has only a local observation in our network. It obtains an observation from the state of a network when it visits a node or meets other ferries. Therefore, a ferry has no knowledge about the waiting messages in nodes, the location of other ferries and their trajectories. The only global knowledge in ferries is the static location (geographical coordinates) of nodes. As mentioned for nodes, they are stationary or semi-stationary. For this reason, the location of nodes can be given to ferries in the beginning of a mission and there will be no need for any update of location information in ferries.

To have a delay tolerant network, ferries and nodes should store messages for a limited time in their buffers until they forward messages. For simplicity, the capacity of data buffers is assumed to be unlimited in ferries and nodes. With existing memory technologies, it is not an unrealistic assumption to have an unlimited buffer in wireless nodes. However, the number of waiting messages in nodes and ferries is an important metric which should be studied.

A UAV with a wireless interface, a flight and communication controller can be considered as a ferry.

Figure 3.1 illustrates an internal architecture of a ferry. A mobility controller is responsible for the path which ferry travels. The mobility controller may decide the travel path of the ferry or just follows a given path. A delay tolerant router decides about the message exchange with a node or other ferries. The figure shows the exchange of data messages between different entities with a solid arrow and the exchange of control messages with dashed arrows. The delay tolerant router and mobility controller may exchange some control messages to improve the efficiency of their decisions. The internal architecture of a node is same as a ferry with a single difference on the mobility controller which is missing in nodes while they are assumed to be stationary.

In next chapters, the details of shown entities are explained.

The scale of distances between nodes is much bigger than their radio transmission ranges. The radio transmission range of wireless nodes is usually limited to several meters in low power technologies which are suitable for battery-based devices. There are some long range wireless technologies with a low power consumption, but they suffer from very low data rates which cannot be employed to exchange bulk messages, that are called bundles in delay tolerant networks, such as map information, etc. On the other hand, the distance between nodes is in a scale of thousand meters.

$$d(i, j \in R) \gg tx_{range} \tag{3.2}$$

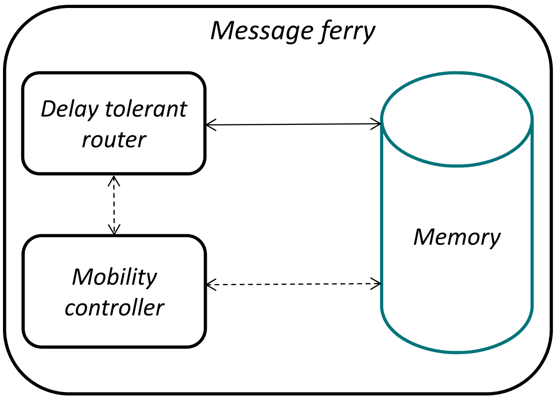


Figure 3.1: An internal architecture of a ferry

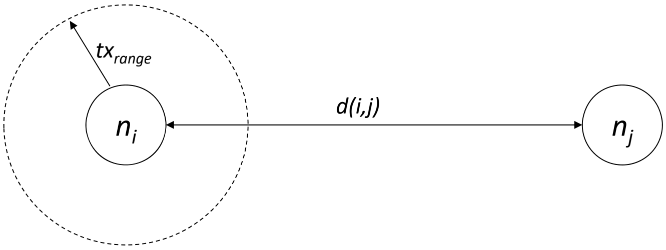


Figure 3.2: The radio transmission range of a node and the distance between two nodes

Therefore, we neglect the radio transmission range of nodes and consider it as zero ($tx_{range} = 0$). With a zero meter radio transmission range of nodes and the lack of mobility in them, the network is fully disconnected and no direct or multi-hop communication may take place without ferries. Having such an assumption, a ferry should travel to the exact coordinates of a node to visit it and exchange messages with it. Figure 3.2 illustrates such circumstances.

The travel time of ferries between nodes in our model is much longer than the required time for a message transmission between a node and ferry. Existing wireless technologies such as IEEE 802.11 allow us to send and receive messages with hundreds of megabytes per second. If we assume that the distance between nodes is thousand meters and the speed of ferry is limited to several meters per second, a ferry must travel several minutes from a node to another one ($T_{travel}(i, j \in R)$). On the other hand, ferries and nodes need several seconds to exchange messages. Therefore, the required time for message exchange between nodes and ferries is neglected ($T_{tx} = 0$) in this work for simplicity and being independent from wireless technologies. Moreover, it has no tangible impact on the end-to-end delay of messages in a DTN

$$T_{travel}(i, j \in R) \gg T_{tx} \tag{3.3}$$

3.2 Traffic of messages

Message are generated only in nodes from the time 0 for a limited time to model a message delivery mission for ferries. Messages are generated to be consumed in nodes. Therefore, no data message is generated to ferries. The format of generated messages and considered traffic scenarios are explained in this section.

3.2.1 Message format

A message consists of a header and a payload. The payload of a message contains data with respect to the application of the message. The payload of messages is out of our focus, but the header contains information which is required for our studies on message ferry networks. Table 3.1 illustrates fields in the header of a message in our network model.

Table 3.1: The format of a message header in our network model

Source ID
Destination ID
Message ID
Generation time
Collection time
Delivery time
Original message
... Payload ...

The fields in the header of a message are as follows:

- Source, destination and message ID: they are unique IDs in the network to identify source and destination nodes and a message. The ID of nodes is given in advance and the message ID is a counter in each node which is a unique value if it is considered along with the source ID.

- Generation time: it is the time stamp for the generation of a message. It is required for calculation of the wait time of a message in its source node and the end-to-end delay of the message. It is assumed that a global time is available in all nodes.

- Collection time: it is the time stamp of the collection a message by a ferry. It is also used to calculate the wait time of a messages in its source node and its travel time in a ferry or ferries.

- Delivery time: it is the time when the message is delivered to its destination node. This field is used to calculate the travel time a message in a ferry or ferries and its end-to-end delay.

- Original message: it is a Boolean value which defines if a message is an original message or a replication of it. This field is used to differentiate between original and replicated messages in the network.

3.2.2 Traffic scenario

All studies of this thesis are done considering three different traffic scenarios. The traffic scenarios are as follows:

- **All to sink**: all nodes generate messages periodically which should be delivered to a sink node. The sink node does not generate any message itself. The message generation rate of nodes are different. 80% of nodes generate a single message to the sink at each message generation interval. The average message generation interval in nodes is 5 seconds. The rest of nodes, which are 20% of all nodes, generate 5 times more messages to the sink. It means, they generate 5 messages to the sink in each message generation interval. This traffic scenario models a sensor network where several sensors need to send messages to a sink node periodically. Figure 3.3 shows an all to sink traffic scenario.

- **Broadcast**: a specific node generates messages which should be delivered to all nodes. The node which broadcasts messages does not receive any message from other nodes. It generates different number of messages to different nodes at each message generation interval. The average message generation interval is again 5 seconds. At each interval, the broadcaster node generates a single message to 80% of the nodes and 5 messages to 20% of the nodes. It models sensor networks,

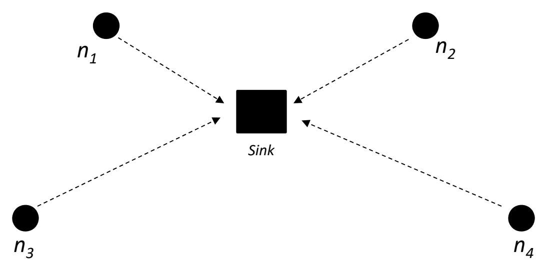


Figure 3.3: All to sink traffic scenario

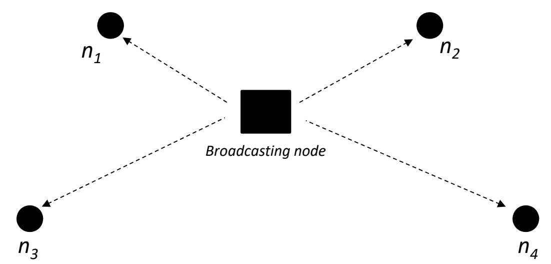


Figure 3.4: Broadcast traffic scenario

where a gateway or a controller may need to broadcast updates for the firmware of sensor nodes. It is illustrated in Figure 3.4.

- **All to all**: all nodes generate and receive messages. Nodes generate messages with different rates. 60% of nodes generate a single message to another node and 40% of nodes generate more than one message (2-10 messages) to more than one node (2-5 nodes) in each message generation interval. The average message generation interval is 5 seconds for this traffic scenario in all nodes. The traffic scenario models a post-disaster communication where rescue teams need to exchange messages. Figure 3.5 shows this traffic scenario.

In the considered network model, nodes generate messages with different rates which models a non-uniform message generation rate of nodes. The non-uniform message generation rate is considered while all nodes do not generate messages with the same rate in a realistic scenario.

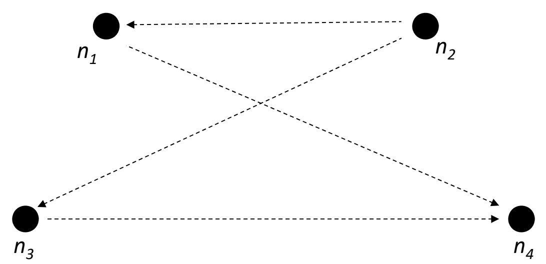


Figure 3.5: All to all traffic scenario

3.3 Performance metrics

In this section, a set of performance metrics are defined which are applied in this thesis to evaluate the proposed schemes. The metrics are as follows:

- End-to-end delay: It is the time difference between a message generation in the source node and the message delivery in its destination node and is calculated as follows:

$$delay_{e2e} = t_{delivery} - t_{generation} \tag{3.4}$$

  $t_{delivery}$ and $t_{generation}$ are exploited from the header of a message considering the format in Table 3.1 for the messages in our message ferry network. The end-to-end delay of a message consists of two parts:

$$delay_{e2e} = wait + travel \tag{3.5}$$

- *wait* time: is the waiting time of a messages in the source node after its generation until its collection by a ferry.

$$wait = t_{collection} - t_{generation} \tag{3.6}$$

- *travel* time: is the traveling time of a message in the buffer of a ferry (or ferries) after its collection from the source node until its delivery to the destination node.

$$travel = t_{delivery} - t_{collection} \tag{3.7}$$

- Traveled distance: is the traveled distance of a ferry from $t = 0$ until the delivery of the last message in the network. It reflects the cost for each ferry to complete a given message delivery mission in a message ferry network.

- Buffer length: is the number of messages in the buffer of a node or a ferry. An infinite capacity of buffer is considered for both ferries and nodes, but this metric reflects the cost of message delivery applying different schemes.

3.4 Conclusion

Table 3.2 summarizes the assumptions in our network model. All the assumptions are considered in such a way to simplify the performance evaluation of the proposed network and avoid the impact of irrelevant parameters on the evaluations.

Table 3.2: Assumptions in the proposed network model

Parameter	Assumption
Topology of nodes	Fully disconnected
Radio transmission range of nodes	0 meters
Mobility of nodes	Stationary
Transmission delay	0 seconds
Capacity of buffer (nodes and ferries)	∞
Message traffic scenarios	All to sink/ Broadcast/ All to all
Message generation rate	Non-uniform
Observation of ferries	Local
Mobility of ferries	Controlled mobility
Velocity of ferries	Constant
Acceleration/deceleration time	0 seconds

4 On-the-fly Mobility Decision Making of Self-Organized Ferries

In this chapter, on-the-fly mobility decision making of ferries is studied. Main contributions of this chapter are as follows:

- Study and comparison of different metrics to apply in the mobility decision maker of ferries
- A novel approach to coordinate ferries in which an indirect signaling between ferries takes place
- Study on the impact of different parameters such as the speed and number of ferries on the performance and costs of a message ferry network

Ferries are ”autonomous” in this chapter for the reason that they decide about their mobility applying an internal process and based on their local observations. Not only the decision about mobility but also all decisions of ferries are made in the same way. This behavior of ferries leads us to the definition of self-organized systems, where numerous interactions of lower level components in a self-organized system emerge patterns at a global level. In our case, ferries are the lower level components of the self-organized network. They employ the on-the-fly decision making as an internal process to make decisions about their mobility.
The flowchart in Figure 4.1 shows steps which are taken by a ferry in a self-organized network.
The steps are taken in a cyclic manner. Each step of the flowchart is as follows:

- **Visit a node**: a ferry arrives at a node after traveling for a limited amount of time. Arrival at a node means that the node is within the radio transmission range of the ferry and both are able to exchange messages. Based on our assumptions for the radio transmission range of nodes, arrival of a ferry at a node means that the ferry is exactly at the same two dimensional coordinates as the

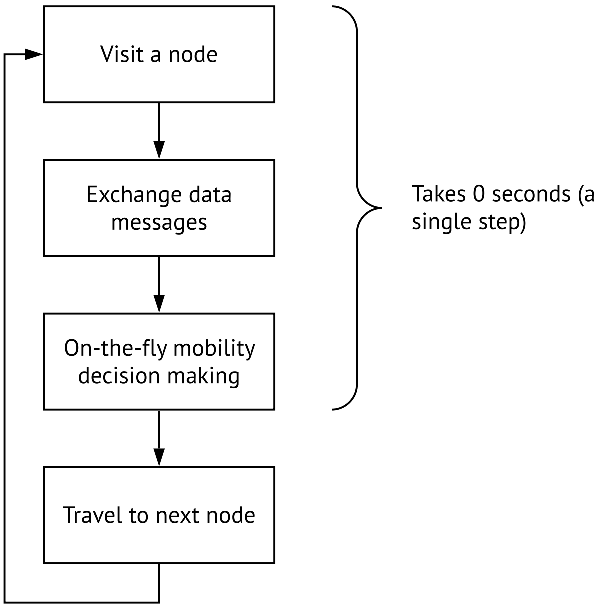


Figure 4.1: Cyclic steps of an autonomous ferry with an on-the-fly mobility decision maker

node. The third dimension, which is the altitude, is neglected for simplicity. If the ferry is an aerial vehicle such as a UAV, all ferries are assumed to be at the same altitude. If a UAV and a node are at the same two dimensional coordinates, they are able to communicate based on our assumptions.

- **Exchange data messages**: during a visit, the ferry exchanges data messages with the node. The ferry delivers messages to the node for which the node is their destination and collects all waiting messages in the node buffer.
- **On-the-fly mobility decision making**: a node is selected as the next node (hop) to visit based on the local observations of the ferry and applying an on-the-fly decision maker.
- **Travel to next node (hop)**: after the selection of a node as the next node to visit, the ferry travels toward the node.

In the next section, an on-the-fly mobility decision maker of ferries is proposed and explained in detail.

4.1 On-the-fly next-hop decision maker for multi-ferry networks (ODMF)

In this section, an On-the-fly next-hop Decision maker for Multi-Ferry networks (ODMF) is proposed. ODMF selects the next node to visit from the set of existing nodes in the network. It applies observations of a ferry in its decision making. To coordinate ferries in the network, an indirect signaling between ferries through nodes takes place. Through the indirect signaling, a ferry expands its observation from the state of the network. The expanded observation is applied as one of the inputs to ODMF to select the next node to visit. In the next section, the decision function in ODMF is explained and later the indirect signaling of ferries is proposed.

4.1.1 Decision function of ODMF

To choose the next node to visit applying ODMF, a score value is calculated by a ferry for all nodes in the network. The node with the highest score value is selected as the next node to visit. The "*Score*" function of ODMF is as follows:

$$Score(r) = \frac{fb(r) + lvt(r)}{distance(c, r)} \tag{4.1}$$

where $fb(r)$ returns a normalized value for node r based on the number of messages to it in the ferry buffer. fb serves messages in the ferry buffer by giving priority to nodes which have more waiting messages in the buffer of a ferry. Therefore, fb in the decision function aims to reduce the *travel* time of messages. fb is calculated as follows:

$$fb(r) = \frac{msg.count(r)}{\max_{i \in R}\{msg.count(i)\}} \tag{4.2}$$

$msg.count(r)$ is the number of messages to node r in the ferry buffer and $\max_{i \in R}\{msg.count(i)\}$ is the number of messages to the node with the maximum number of messages in the ferry buffer.

To serve messages in nodes and reduce their *wait* time, ODMF applies the last visit time (lvt) which is based on a history of visits to nodes by a ferry. Each ferry keeps a history of its last visit time to each node and applies it in its mobility decision making. Table 4.1 shows the history table which a ferry keeps in its memory. The table stores the time of the last visit to nodes.

$lvt(r)$ in Equation 4.1 returns a normalized value for node r based on the last visit

Table 4.1: Last visit history in a ferry

Node ID	Last Visit Time
r_1	t_1
r_2	t_2
...	...

time of the ferry to it. The value of lvt for node r is calculated as follows:

$$lvt(r) = \frac{t_c - t_{visit}(r)}{t_c - min_{i \in R}\{t_{visit}(i)\}} \tag{4.3}$$

where $t_{visit}(r)$ is the time of the last visit to node r by the ferry and t_c is the current time. Node r has a bigger value for $lvt(r)$, if it has been visited a longer time ago. lvt in the decision maker serves messages in nodes and avoids a visit starvation in them. Visit starvation degrades performance of a message ferry network because messages experience a long *wait* time in a starved node. It also prevents timely close visits to a node. Frequent visits to a node may waste resources in a message ferry network when the visit rate to a node is higher than its message generation rate.
$distance(c, r)$ in Equation 4.1 returns a normalized value based on the distance between the current node c (which is visited by the ferry in the time of decision making) and node r. The decision maker tends to select a node in the neighborhood of the current location of the ferry to improve the performance of message delivery.
In the next section, the indirect signaling of ferries through nodes is explained which is applied to coordinate ferries.

4.1.2 Indirect signaling of ferries through nodes

Based on the assumptions of this thesis, each ferry has only a local observation. It obtains an observation when it visits a node or meets another ferry. In a system with multiple autonomous agents, a mechanism can be applied to coordinate agents to improve the efficiency of the system. In our problem, each ferry decides on-the-fly about the next node to visit based on its buffer state and a history which it keeps in its memory about its previous decisions independent of other ferries. For example, a ferry may decide to visit node r_1. At the same time, other ferries may make the same decision. In this case, all of the visits except the first visit to node r_1 can be useless. The first ferry, which visits the node before others, collects all messages from the node. Thus, there will be an empty buffer in the node for other ferries. To overcome this challenge, an indirect signaling of ferries is applied in this thesis. We assume that ferries have not a long range communication interface to exchange their historical knowledge. Moreover, they do not meet each other within their travel. Therefore, the idea of stigmergy from self-organized systems is employed in this thesis to exchange historical knowledge between ferries.

Stigmergy [101], [102], [103], [104] is the indirect communication of agents through their environment. It is an instance of self-organization. The indirect communication of agents leads to the coordination of agents and stimulates their next actions. Agents will be coordinated and collaborate without any need for planning and control. Each agent leaves traces in its environment and traces will stimulate the actions of other agents. Stigmergy can be found in biologic organisms such as bees, termites, ants, etc. As an example of stigmergy, paths are built by ants using traces (pheromone) between their nests and a food source.

In our problem, ferries are self-organized agents which employ nodes as their environment for an indirect communication. They exchange the last visit time history with nodes and update their tables. For this reason, each node in the network keeps a history table same as Table 4.1. The extended flowchart of a ferry steps with a signaling step is shown in Figure 4.2 in which the update of the history table in a ferry is highlighted. The update of the history table takes place through a signaling between a ferry and a node.

As it is seen in Figure 4.2, the update of the history table in a ferry occurs before the mobility decision making. Therefore, the signaling between a ferry and a node impacts on the next decision of the ferry.

In the proposed message ferry network, two types of signaling exist and they are as follows:

- **Direct signaling**: it takes place when a ferry visits a node. The ferry and node exchange their last visit history table and mutually update their tables to have the most up-to-date information.
- **Indirect signaling**: it emerges through direct signaling between ferries and nodes. The last visit time history of nodes are exchanged indirectly between ferries through nodes. Nodes act as relays to exchange information between ferries. However, the exchange of information between ferries takes place with high delays. This can invalidate information inside the tables.

As mentioned for the on-the-fly next-hop decision maker of ferries, a ferry considers

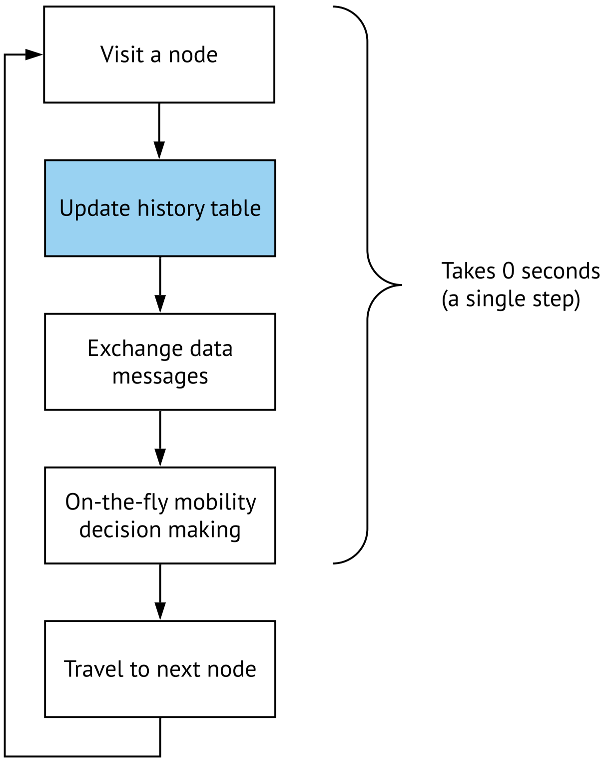


Figure 4.2: Cyclic steps of an autonomous ferry with an on-the-fly mobility decision maker and a signaling step

its past decisions in its next decision. History of the past decision of a ferry is the list of nodes which the ferry has visited until now and the time stamp of visits (shown in Table 4.1).

Figure 4.3 shows the emerged indirect signaling between two ferries through a single node. Ferries f_1 and f_2 visit node n_1 in two different times t_1 and t_2. If we assume that $t_1 \leq t_2$, then f_1 visits the node earlier and may leave information in it which is applied in the decision of f_2. In this case, f_2 can consider the visit time of f_1 to a node and avoid a redundant visit to it. However, the emerged indirect signaling through nodes is imperfect and ferries do not have access to the real up-to-date information about the state of the network. The stigmergy is sub-optimal comparing with the case in which all ferries have a global view from the state of a network. However, it may improve the efficiency of decision making in ferries with local observations.

In the next section, the performance of the on-the-fly decision maker and the proposed indirect signaling of ferries is studied.

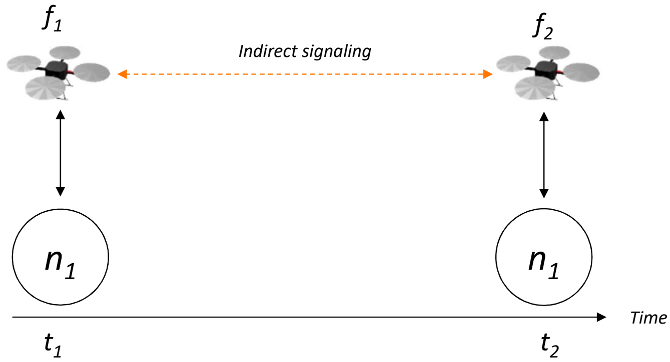


Figure 4.3: Emerged indirect signaling of ferries through a node

4.2 Study on the performance of ODMF

In previous sections, ODMF was proposed which decides on-the-fly the next node to visit for ferries and applies an indirect signaling of ferries for their coordination. In this section, the performance of ODMF in terms of the end-to-end delay of message delivery and its constituent elements such as *travel* and *wait* time of messages are studied.

For comparisons, different on-the-fly decision makers are selected to study the impact of decision metrics. The studied on-the-fly decision makers are as follows:

- ODMF-NC (ODMF-No Coordination): it uses the same decision function as ODMF, but ferries do not apply the indirect signaling. ODMF-NC is compared with ODMF to investigate the impact of the indirect signaling between ferries.

- SOMF (Self-Organized Message Ferry): it is an on-the-fly decision maker for single ferry networks [99], [100] where a ferry applies the following decision function to choose the next node to visit:

$$Score(r) = \frac{fb(r) + nv(r)}{distance(c, r)} \tag{4.4}$$

  $fb(r)$ and $distance(c, r)$ are same as the decision function in Equation 4.1. $nv(r)$ is the normalized value of the number of visits to node r. As SOMF has been proposed for single ferry networks, it does not apply any mechanism for coordination of ferries.

- SOMF-MH (SOMF with a Message generation History): it is similar to SOMF, but considers the message generation rate of nodes. A ferry keeps the number of messages which has collected from a node during its last visit to the node and the time between visits to calculate the message generation rate of the node. SOMF-MH applies following decision function:

$$Score(r) = \frac{fb(r) + nv(r) + mr(r)}{distance(c, r)} \tag{4.5}$$

where $mr(r)$ is the normalized value for the message generation rate of a node which is calculated by a ferry to apply in its decision function. The rest of the elements in the decision function of SOMF-MH are same as SOMF.

- FB (Ferry Buffer): it is an on-the-fly decision maker which considers only messages in the ferry buffer. It does not apply any other metric in its decision making and does not apply any mechanism for coordination of ferries. The decision function of FB is as follows:

$$Score(r) = fb(r) \tag{4.6}$$

where $fb(r)$ is the same function as in ODMF, ODMF-NC, SOMF, and SOMF-MH. In a case of an empty buffer in a ferry, FB chooses a node randomly to break the tie.

- FBD (Ferry Buffer and Distance): it is similar to FB, but considers the distance between the current location of a ferry and nodes. It is studied to find the impact of distance metric in mobility decision. The decision function of FBD is as follows:

$$Score(r) = \frac{fb(r)}{distance(c, r)} \tag{4.7}$$

where are $fb(r)$ and $distance(c, r)$ are same as functions in ODMF. FBD chooses a node randomly same as FB, if the ferry buffer is empty.

Figure 4.4 shows all approaches in our studies. The simulation setup is shown in Table 4.2.

In following sections, the performance of decision makers are studied under 3 different traffic scenarios which have been explained in Chapter 3. The results are shown in boxplots which illustrate the maximum, minimum and dispersion of values (5 to 95 percentile). The red line inside a box is the median of values and the red square is the mean value.

Table 4.2: Simulation setup of studies on the performance of on-the-fly decision makers

Parameter	Value
Number of nodes	20
Number of ferries	10
Distribution of nodes	Random Uniform
Speed of ferries	5 m/s
Network size	1000 x 1000 meter
Message generation	1000 seconds
Number of runs	10 runs

4.2.1 All to all traffic

All approaches are compared considering the all to all traffic scenario. The end-to-end delay of messages, their travel and wait time are shown in Figure 4.5. The lessons that we learn from the results are as follows:

- Indirect signaling between ferries decreases the end-to-end delay of messages while the only difference between ODMF and ODMF-NC is the indirect signaling of ferries through nodes. To find the reasons for the impact of the indirect signaling on message delivery, the travel time and wait time of messages will be studied.
- The indirect signaling of ferries through nodes, in favor of their coordination, impacts on the travel and wait time of messages. It provides recent information about the state of the network for ferries. A ferry avoids a useless visit a node while it receives up-to-date information from other ferries about a recent visit to the node. In such a case, the ferry can serve messages in its buffer that leads to a shorter travel time of them or it can visit other nodes to reduce the wait time of messages in them. However, the amount of improvement on the travel and wait time of messages depends on the traffic scenario in the network. In the all to all traffic, the impact of indirect signaling is mostly on the travel time of messages.
- The ”last visit time” metric in ODMF and ODMF-NC has a better performance than the ”number of visits” to nodes which is applied in SOMF, and SOMF-MH. This is concluded by comparing ODMF-NC and SOMF. Their only difference is the metric in their decision function to provide fairness among nodes and serve waiting messages in them. The reasons for the better performance of ”last visit time” will be discussed later.

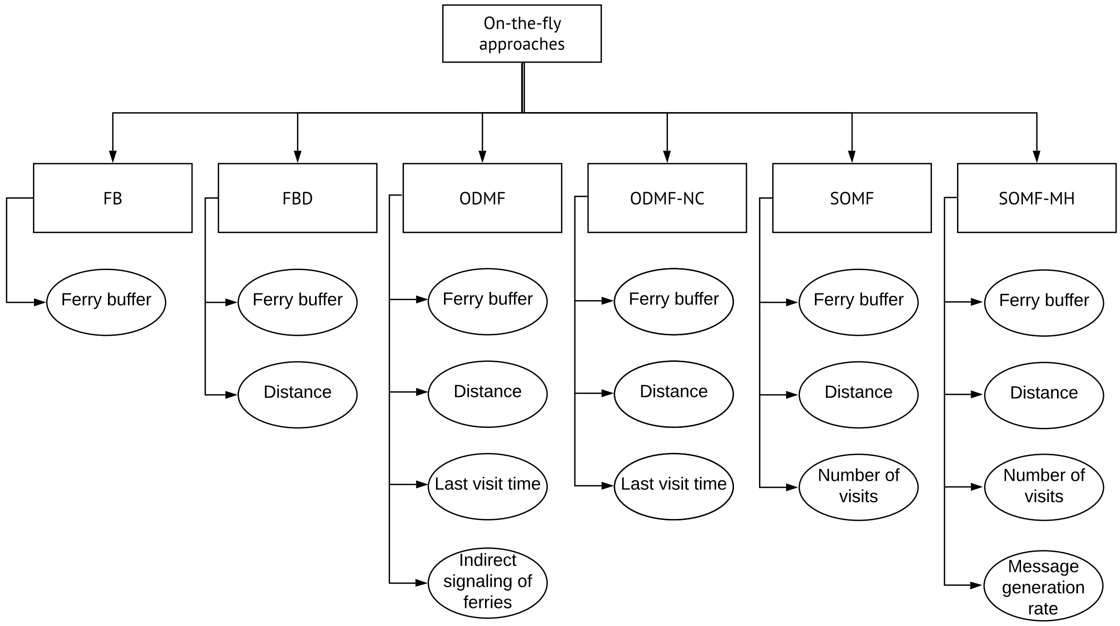


Figure 4.4: On-the-fly decision makers and their considered metrics

- Taking the message generation rate of nodes into consideration for the next-hop decision of ferries is useless in a multi-ferry network. Learning in a system with multiple active agents where each agent impacts on the state of the system needs an efficient coordination of them. In SOMF-MH, each ferry learns the message generation rate of nodes independently. The learned knowledge by a ferry is invalid or becomes invalid when other ferries visit the same node.

- The distance metric in FBD has a tangible impact on the faster delivery of messages in the all to all traffic scenario when it is considered along with the state of buffer in a ferry. Due to the distance metric, FBD shows a better performance than FB.

- FB and FBD have the best travel time within all approaches while they serve only messages in the ferry buffer. On the other hand, they show the longest wait time of messages because they neglect waiting messages in nodes.

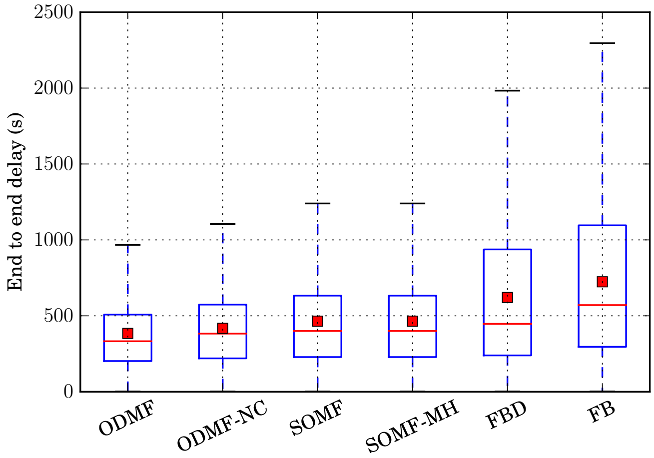
- The distance metric in FBD improves both travel and wait time of messages. The travel time is improved since a ferry can deliver more messages to closer nodes instead of traveling to a distant node to deliver its messages. The total number of messages which could be delivered in the vicinity is usually more than the number of messages which are delivered to a distant node.

- The same reason is true for the wait time of messages when a distance metric is applied in the mobility decision function. More messages can be collected from nodes in the vicinity than a distant node.
- The ”number of visits” to nodes in SOMF and SOMF-MH makes the travel time of messages longer in comparison with the ”last visit time” in ODMF and ODMF-NC. When a message flow exists between two close nodes or some nodes receive messages from several nodes, the number of visits to nodes becomes unbalanced. In this case, the ”number of visits” to nodes in SOMF and SOMF-MH avoids visiting some of the nodes for some time and pushes the ferry to visit other nodes to provide fairness and balance the number of visits to nodes. This will increase the travel time of messages in the ferry to the nodes which have been visited more often. The same problem may occur in ODMF and ODMF-NC with the ”last visit time” metric, but a single visit to other nodes solves the problem. Therefore, the travel time in ODMF and ODMF-NC is less than SOMF, and SOMF-MH.
- Despite different metrics to serve waiting messages in nodes applying ODMF and SOMF, the wait time of messages is same with ODMF, ODMF-NC, SOMF and SOMF-MH. It shows that the ”last visit time” metric can provide fairness among nodes same as the ”number of visits” to nodes, while it has a less adverse impact on the travel time of messages.

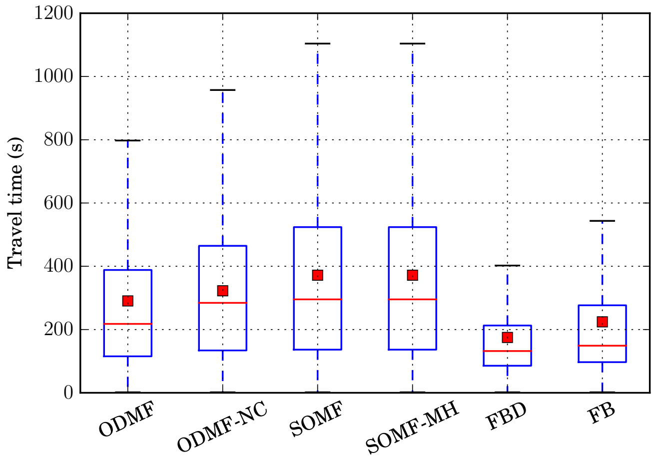
4.2.2 All to sink traffic

All approaches are compared considering the all to sink traffic scenario. The end-to-end delay of messages, their travel and wait time are shown in Figure 4.6. The lessons that we learn from the results are as follows:

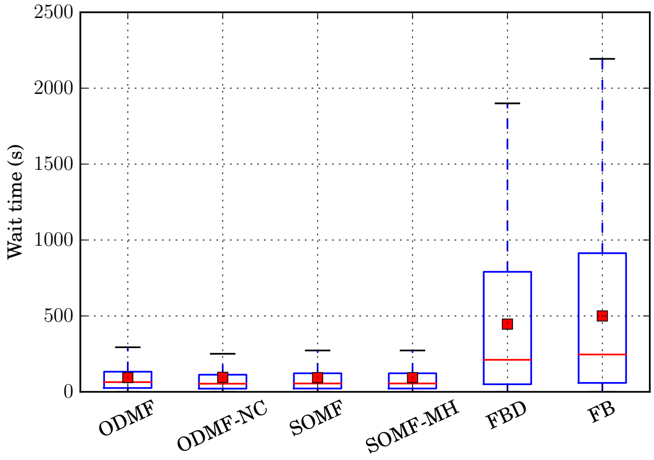
- Indirect signaling in ODMF impacts on the travel and wait time of messages in the all to sink traffic scenario. The reasons for the reduction in the travel and wait time of messages in this traffic scenario is same as the reasons which were mentioned for the all to all traffic scenario.
- The distance metric in FBD has no impact on the performance of message delivery since all messages should be delivered to a sink node. There is no message which can be delivered to a node in the vicinity of the ferry in this traffic scenario.



(a) End to end delay of messages



(b) Travel time of messages in ferries



(c) Wait time of messages in nodes

Figure 4.5: All to all traffic scenario with 20 nodes and 10 ferries

- FBD and FBD have the shortest travel time and longest wait time. A ferry travels directly to the sink node when it collects a message from a node. Therefore, the travel time of messages is the minimum in this case. However, FB and FBD neglect messages in nodes. This causes the longest wait time of messages applying them.

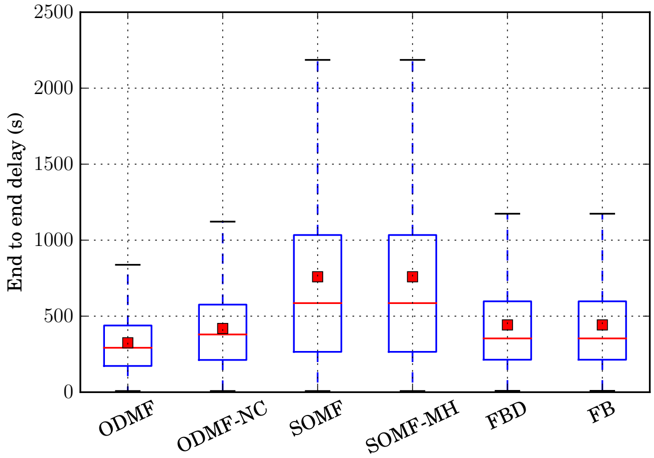
- The "number of visits" to nodes in SOMF and SOMF-MH performs even worse in this traffic scenario than the all to all traffic for the reason that the sink node is visited more often and the number of visits to the sink increases rapidly. Then, the ferry neglects the sink for some time to visit other nodes to provide fairness in terms of the number of visits. This problem is called "sink avoidance" in this thesis. This behavior increases the travel time of messages.

- The wait time in SOMF and SOMF-MH is more than ODMF and ODMF-NC. The reason is the waiting messages in nodes which are very close to the sink. Due to the distance metric in SOMF and SOMF-MH, the nodes in the proximity of the sink are visited more than others since the sink is the destination for all messages and is visited frequently. After some time, there will be a difference in the number of visits to close and far nodes from the sink. Again to provide fairness, the close nodes to the sink are neglected for a while. This problem is called "close to sink avoidance" in this thesis. This problem increases the wait time of messages while all nodes generate messages to the sink node.

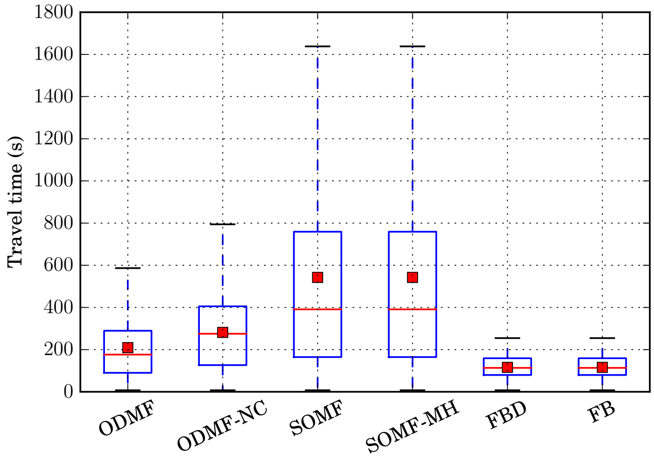
4.2.3 Broadcast traffic

All approaches are compared considering the broadcast traffic scenario. The end-to-end delay of messages, their travel and wait time are shown in Figure 4.7. The lessons that we learn from the results are as follows:

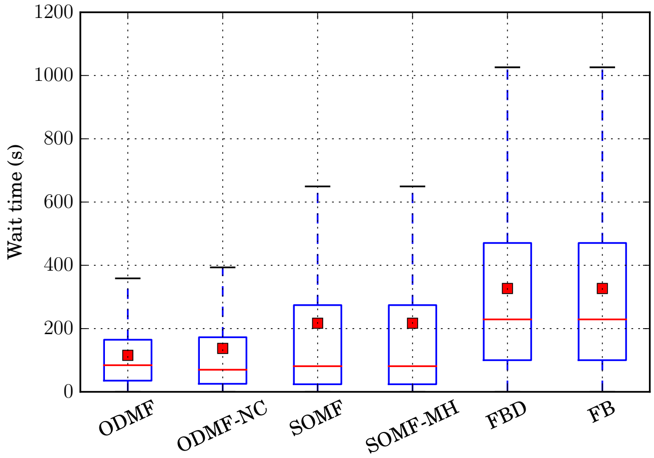
- The indirect signaling of ferries in ODMF has no tangible impact on message delivery delay. The performance of ODMF and ODMF-NC are same in terms of the end to end delay, travel and wait time of messages. The travel time is not improved because a ferry has messages in its buffer to all nodes. If it does not visit a node to visit others, the average delay of messages does not change. Moreover, the wait time is not reduced while only a single node generates messages. Therefore, the change in the order of nodes to visit which is caused by an indirect signaling of ferries has no tangible impact on the wait time of messages in the broadcaster node.



(a) End to end delay of messages



(b) Travel time of messages in ferries



(c) Wait time of messages in nodes

Figure 4.6: All to sink traffic scenario with 20 nodes and 10 ferries

- The distance metric highly improves the performance of FBD compared to FB. It reduces the travel time of messages by delivering more messages to nodes in the vicinity than delivering less messages to a far node.
- In this traffic scenario, a ferry has messages to all nodes in the network. For this reason, the travel time of messages is the main factor in their end-to-end delay. Any improvement in the travel time is more effective than an improvement in the wait time. The wait time of different approaches are different, but all approaches except the FB show similar travel time of messages. Altogether, the end-to-end delay of all approaches except FB is similar.
- To find out the reasons for the difference in the wait time of messages in FB and FBD, the results for this traffic scenario with different number of ferries is studied.

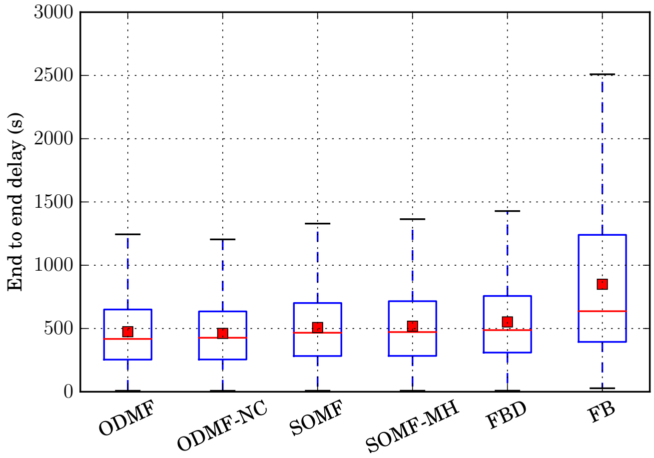
4.3 Study on the impact of the number of ferries

In this section, the number of ferries in a message ferry network is increased and its impact on the average end-to-end delay, travel and wait time of messages is studied. The number of ferries is increased from 1 to 20 ferries to study different ratios for the number of ferries to nodes. The studies are done considering the three traffic scenarios which were explained in Chapter 3. The simulation setup is same as Table 4.2 with a difference in the number of ferries which is increased from 1 ferry to 20 ferries. The SOMF-MH is omitted from our studies in this section while no difference has been observed in the results of SOMF and SOMF-MH in our previous studies.

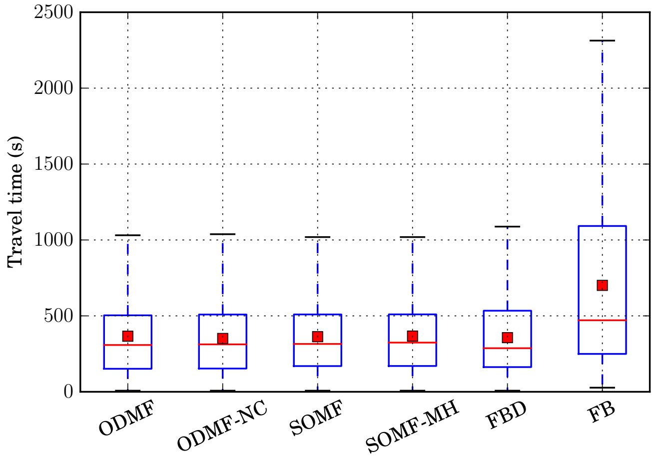
4.3.1 All to all traffic

All approaches are compared considering the all to all traffic scenario. The average end-to-end delay of messages, their average travel and wait time are shown in Figure 4.8. The lessons that we learn from the results are as follows:

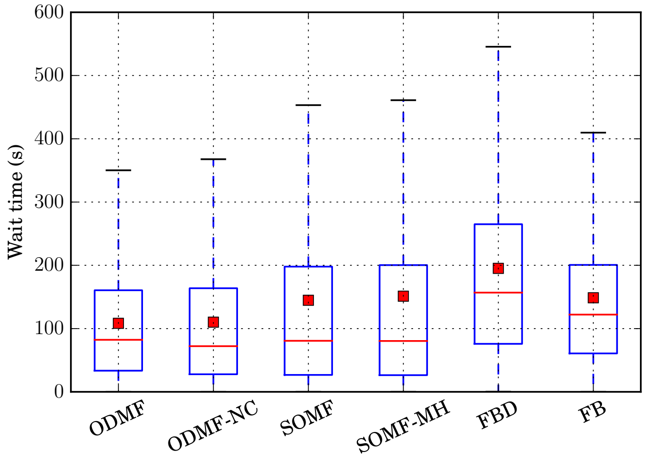
- Clearly, the indirect signaling has no impact employing a single ferry in the network. By increasing the number of ferries, more impact is seen comparing ODMF and ODMF-NC.
- The effect of indirect signaling between ferries is mostly on the travel time of messages considering the all to all traffic scenario.



(a) End to end delay of messages



(b) Travel time of messages in ferries



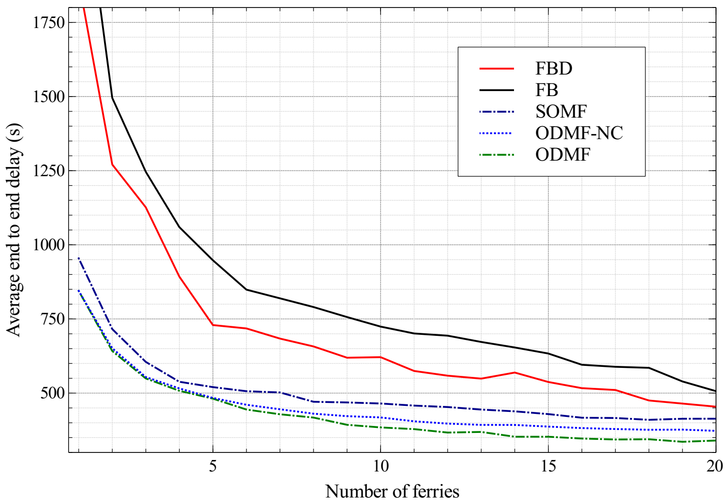
(c) Wait time of messages in nodes

Figure 4.7: Broadcast traffic scenario with 20 nodes and 10 ferries

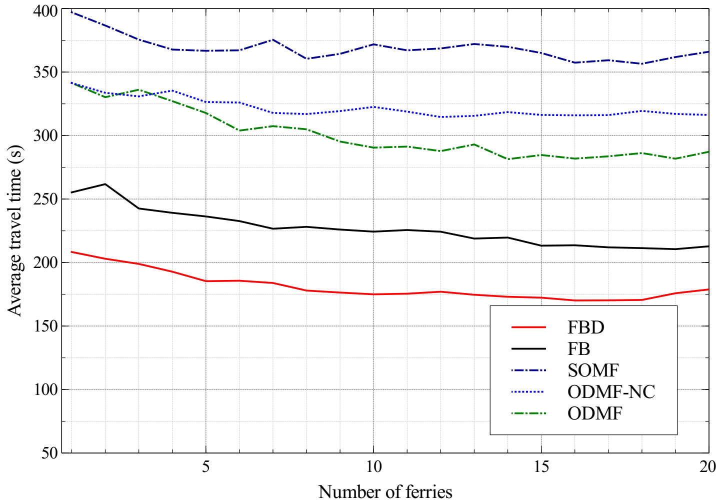
- FBD has always the best travel time while it serves only messages in the ferry buffer and gives priority to messages which can be delivered to closer nodes.
- All approaches which have a metric to serve waiting messages in nodes (or fairness metric) have the same wait time and it tends to zero by increasing the number of ferries.
- FBD shows less wait time than FB while it can deliver messages in the ferry buffer faster than FB. When there is no message in the ferry buffer, FB and FBD start to visit nodes randomly. FBD starts this procedure earlier. As all nodes generate messages, the wait time in FBD is less than FB since a ferry can collect messages earlier. With more ferries, the difference between FB and FBD decreases while there are fewer messages in the buffer of ferries. This causes a similar behavior of FB and FBD.
- Increasing the number of ferries reduces the end-to-end delay. The reduction occurs mostly in the wait time of messages and slightly in the travel time. The reduction of the wait time is due to the more frequent visits to nodes by more ferries. All nodes are message generators and more often visits to them reduce the wait time of messages. However, if the frequency of the visits is more than the frequency of message generation in nodes, the more visits to nodes will be useless and will not impact on the wait time of messages. The more ferries in the network impacts on the average number of messages in the buffer of a ferry and reduce it. By the reduction of the average number of messages in the buffer of a ferry, the travel time is also reduced while the ferry should serve fewer messages. However, there is a slight improvement in the travel time of messages in this traffic model since a node may generate messages to several destinations at each message generation interval. In this case, the increase in the number of ferries does not impact the travel time while all messages are collected by a ferry and should be delivered to their destinations by the same ferry.

4.3.2 All to sink traffic

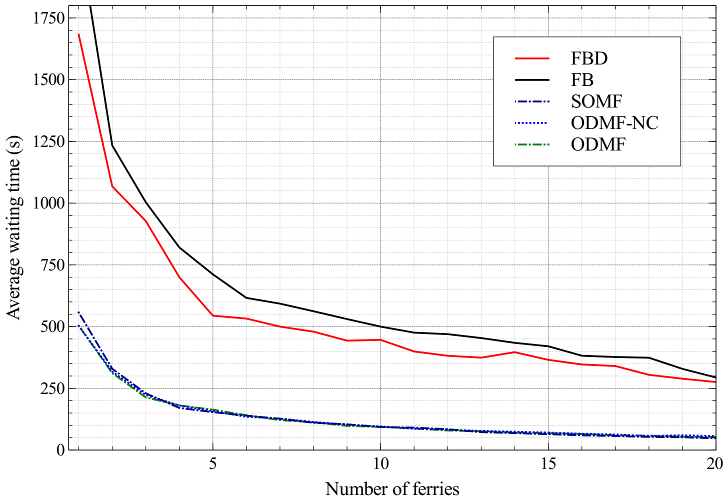
All approaches are compared considering the all to sink traffic scenario. The average end-to-end delay of messages, their average travel and wait time are shown in Figure 4.9. The lessons that we learn from the results are as follows:



(a) Average end to end delay of messages



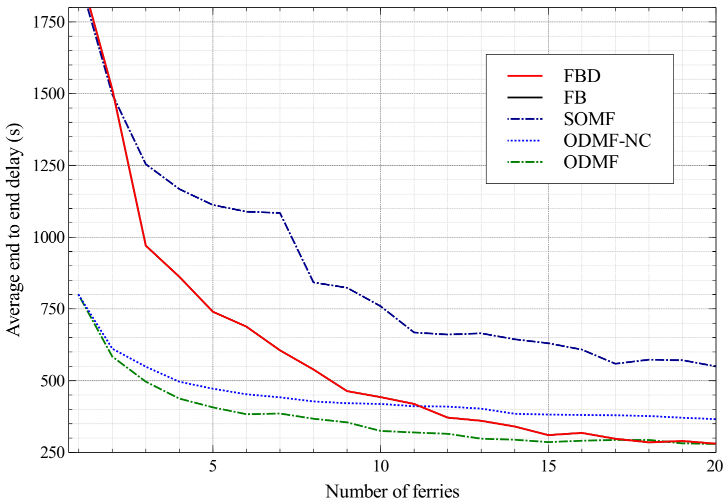
(b) Average travel time of messages in ferries



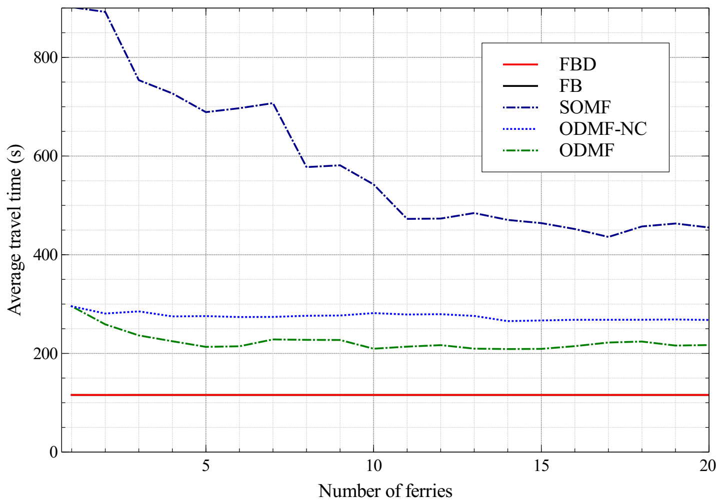
(c) Average wait time of messages in nodes

Figure 4.8: All to all traffic scenario with 20 nodes and 1 to 20 ferries

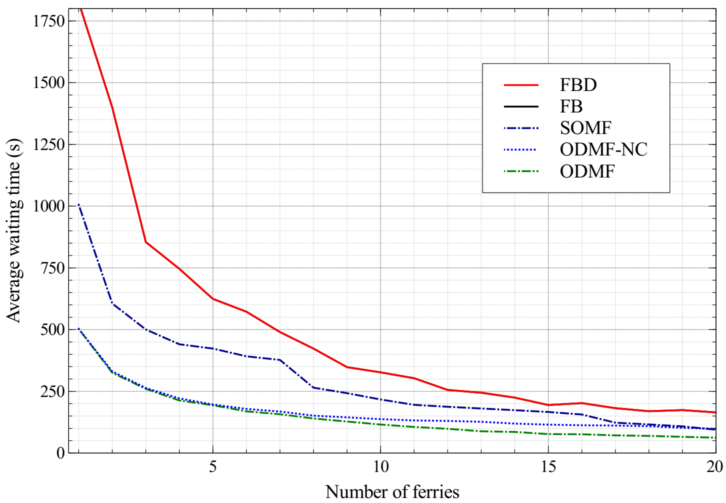
- The ”number of visits” to nodes which is applied in SOMF causes a long travel time of messages in this traffic scenario due to the ”sink avoidance” problem which was mentioned earlier.
- The wait time of messages in SOMF is also worse than ODMF and ODMF-NC due to the ”close to sink avoidance” problem. In ODMF and ODMF-NC, the same issue may occur, but it is resolved by a single visit to nodes which are far from the sink. Therefore, the sink and nodes in its vicinity are visited more than others and the ferry does not try to balance the number of visits for all nodes.
- FB and FBD always have the minimum and the same travel time while a ferry travels directly to the sink after the collection of a message. An increase in the number of ferries does not have any impact on the travel time of messages.
- The performance of FB and FBD improves faster than other approaches by increasing the number of ferries. With 20 ferries, they have the same performance as ODMF and better performance than ODMF-NC. The weak aspect of FB and FBD is the high wait time of messages which is not the case with a high number of ferries. Remember that the travel time does not change for FB and FBD by increasing the number of ferries.
- Increasing the number of ferries has no impact on the travel time of FB, FBD, ODMF-NC while each ferry works independently. In ODMF, the travel time decreases by increasing the number of ferries due to the coordination of ferries. The indirect signaling between ferries is more important when there are more ferries in the network because the probability of redundant visits to nodes is more and the indirect signaling of ferries avoids the redundancies. The emerged coordination of ferries helps to avoid useless visits to nodes and improves the travel time of messages.
- In SOMF, ferries also work independently from each other. However, increasing the number of ferries improves the travel time while the task of message delivery is distributed among more ferries. As ferries start from different nodes, they have different views from the network. In this case, the problem of ”sink avoidance” is less important while each ferry has fewer messages in its buffer. Whenever a ferry avoids visiting the sink node, other ferries visit the sink because their view from the state of the network is different.



(a) Average end to end delay of messages



(b) Average travel time of messages in ferries



(c) Average wait time of messages in nodes

Figure 4.9: All to sink traffic scenario with 20 nodes and 1 to 20 ferries

4.3.3 Broadcast traffic

All approaches are compared considering the broadcast traffic scenario. The average end-to-end delay of messages, their average travel and wait time are shown in Figure 4.10. The lessons that we learn from the results are as follows:

- Increasing the number of ferries does not impact on the travel time of messages while all messages to all nodes are in the buffer of a single ferry. A ferry collects all messages from the broadcaster and delivers them to their destinations.
- The distance metric shows a noticeable impact on the travel time of messages. This can be interpreted by comparing FB and FBD.
- All approaches except FB have a similar travel time of messages since all nodes are receivers of messages and a change in the order of visits to nodes does not have a big impact on the average travel time of messages.
- The wait time of messages is reduced by increasing the number of ferries because all nodes are visited more often. This is true for all traffic scenarios.
- With a high number of ferries, 20 ferries in our case, the wait time of messages is very similar for all approaches. Even FB and FBD, which only consider messages in the buffer of a ferry, provide a similar wait time as ODMF, ODMF-NC, and SOMF, which have a metric to serve waiting messages in nodes in their decision function.
- When there are two close nodes in a network, a ferry may travel back and force several times between them due to the distance metric, if it is applied in the mobility decision maker. In such a case, SOMF behaves rigorously with those nodes by ignoring them for some time to provide fairness. If one of these nodes is the broadcaster, the average wait time increases and if not, the average travel time increases. The reaction of ODMF and ODMF-NC is much smoother to such a problem. However, the problem is less important when more ferries are employed in a network.
- Performance of FB and FBD in terms of the average wait time of messages has fluctuation employing different number of ferries. The fluctuations are due to the random behavior of FB and FBD in the case of an empty buffer of a ferry. In the case of random behavior of a ferry, the broadcaster node may be visited frequently or rarely. For this reason, the average wait time of messages fluctuates

having different number of ferries. However, FB and FBD show similar results due to the uniform distribution of the random function in them.

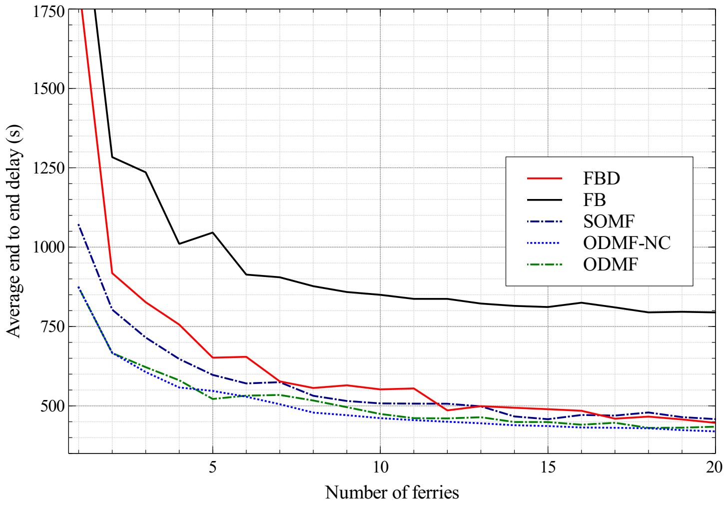
4.3.4 Average traveled hops of messages

The average traveled hops of messages in the buffer of ferries reflects the number of nodes which are visited by a ferry after the collection of a message from its source node until its delivery to its destination node. All approaches are compared in terms of the average traveled hops of messages to find the impact of the number of ferries on it and the relationship between its value and the travel time of messages. Figure 4.11 shows the average traveled hops of messages considering different traffic scenarios. The following lessons are learned from the results of the all to all traffic scenario:

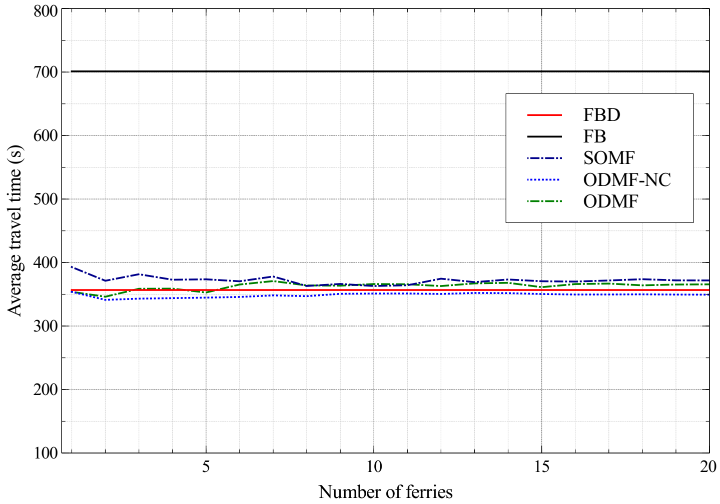
- The average traveled hops of messages in ODMF is always (except when there is a single ferry) less than ODMF-NC due to the indirect signaling between ferries. Applying indirect signaling between ferries, a ferry avoids visiting some of the nodes which have been visited by other ferries recently. This decreases the number of traveled hops for messages and consequently the average travel time of messages.
- The impact of indirect signaling between ferries is increased by increasing the number of ferries. The need for coordination of ferries is more when there are more ferries in a network.
- FB and FBD have the least traveled hops of messages since they serve only the messages in the ferry buffer.

Considering the all to sink traffic scenario, following lessons are learned from results:

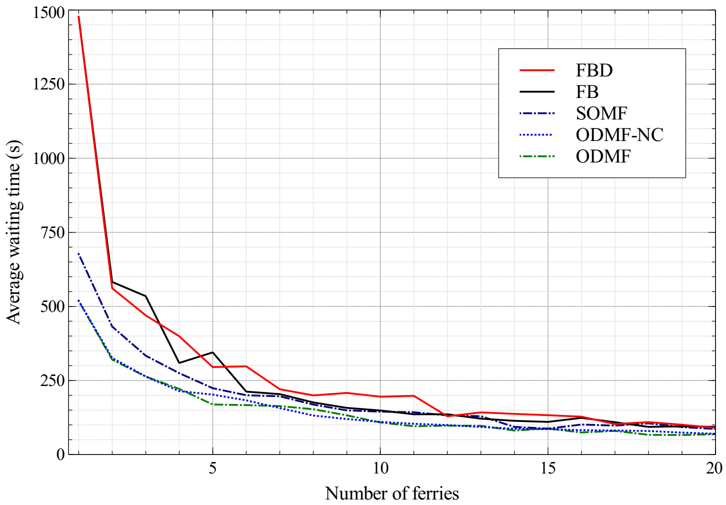
- With FB and FBD, messages travel a single hop between the source and sink. They provide the minimum travel time of messages.
- The same reason is valid for the difference between ODMF and ODMF-NC as mentioned for the all to all traffic.
- The reason for the high number of traveled hops in SOMF is the ”sink avoidance” problem.



(a) Average end to end delay of messages



(b) Average travel time of messages in ferries



(c) Average wait time of messages in nodes

Figure 4.10: Broadcast traffic scenario with 20 nodes and 1 to 20 ferries

The lesson learned from the simulation results considering the broadcast traffic scenario is as follows:

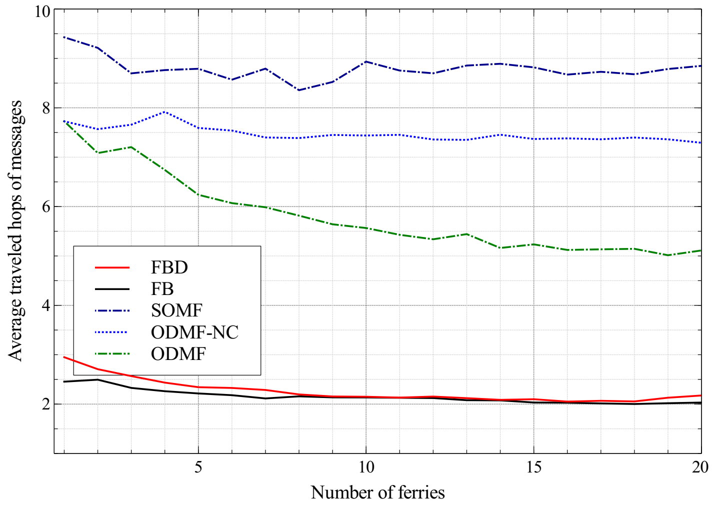
- The most interesting result for this traffic scenario is the comparison of FB and FBD. The distance metric in FBD causes more traveled hops of messages, but the less travel time. Applying the distance metric, FBD delivers messages to closer nodes first. For this reason, it increases the number of traveled hops of messages in a ferry buffer. However, this behavior decreases the average travel time of messages due to earlier delivery of messages to closer nodes.
- In all other approaches (than FB and FBD), the less traveled hops is equivalent to a shorter travel time of messages.

4.4 Study on the cost vs. performance

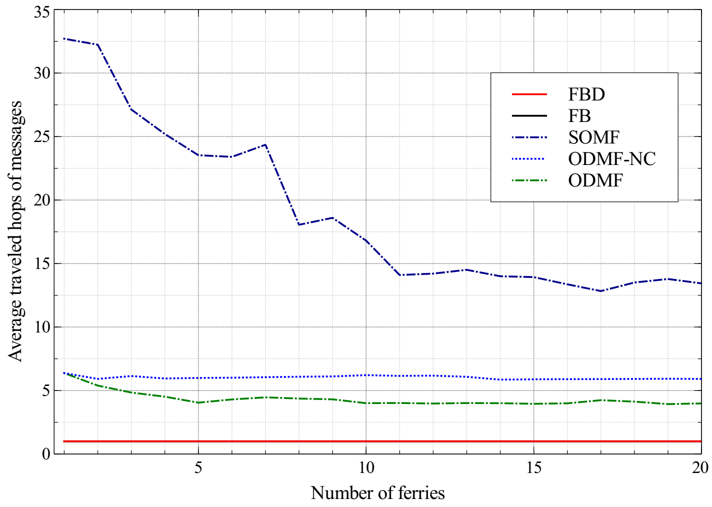
The traveled distance of ferries is an important parameter in a message ferry network which reflects the cost of message delivery in a network. Ferries have a limited source of energy and therefore a limited operation time. If we employ a UAV as a message ferry which flies among disconnected nodes, the limited lifetime of UAV should be taken into consideration. Therefore, the total traveled distance of ferries to complete a message delivery mission, which is the delivery of all messages that have been generated by nodes, is an important metric to study. Figure 4.12 illustrates the total traveled distance of all ferries and the average end-to-end delay of messages for different setups in message ferry networks with 20 nodes considering different traffic scenarios. The total traveled distance of ferries is the summation of all traveled distances by all ferries and reflects the total cost of the system. To achieve a performance goal which is an average end-to-end delay of messages in a network, there are two options in designing a message ferry network with multiple ferries. The options are as follows:

1. Increasing the speed of ferries
2. Increasing the number of ferries

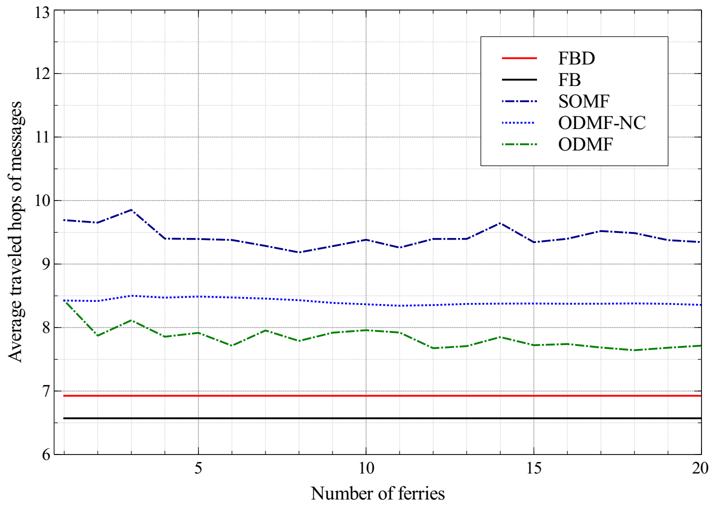
In Figure 4.12, a network setup is shown as a pair of (number of ferries, ferry speed). For instance, (4,2) is a network setup with 4 ferries where each ferry travels with the speed of 2 m/s. The simulation setup of this study is shown in Table 4.3. Based on the results of simulation studies, the following points are inferred:



(a) All to all traffic



(b) All to sink traffic



(c) Broadcast traffic

Figure 4.11: Average number of traveled hops of messages

Table 4.3: The simulation setup of the study on the cost vs. performance of message ferry networks

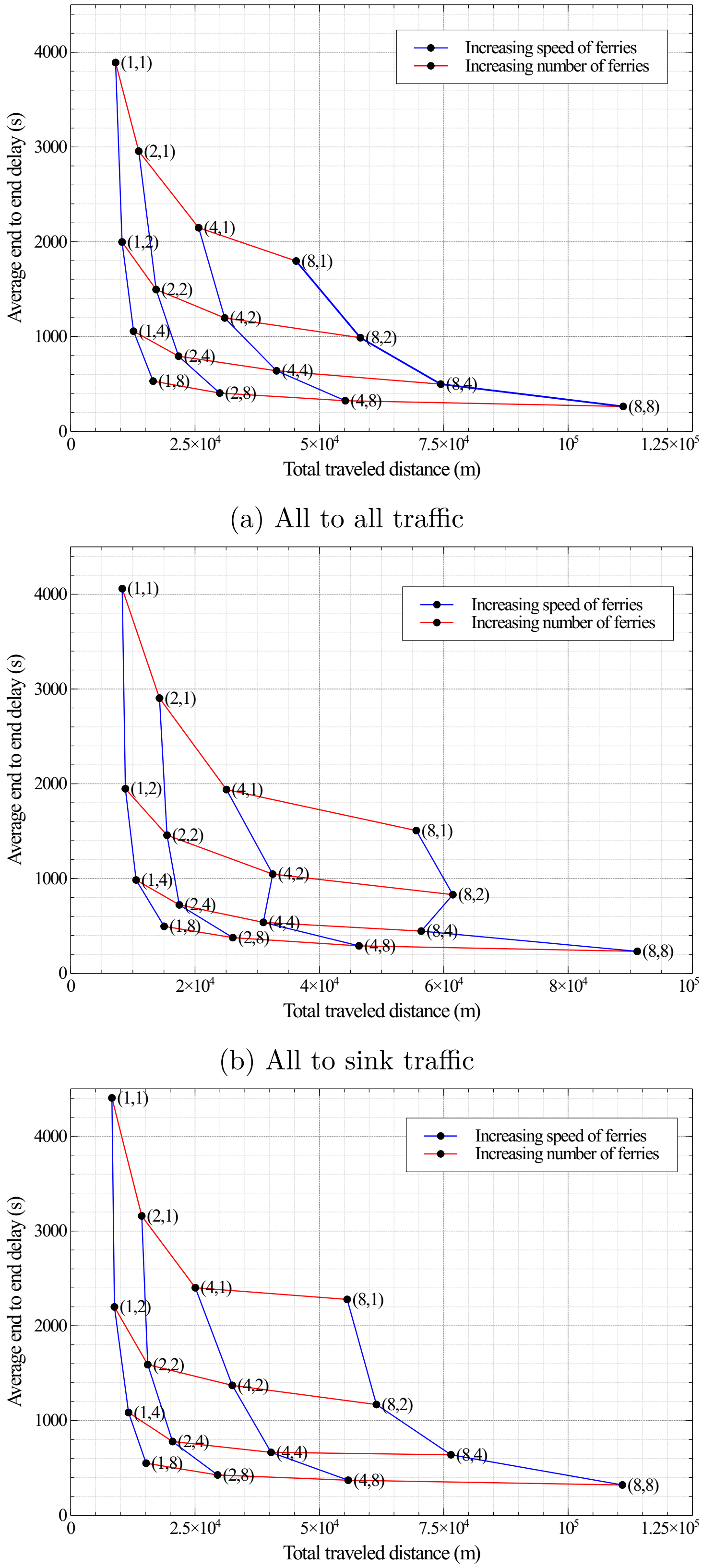
Parameter	Value
Number of nodes	20
Number of ferries	1,2,4,8
Distribution of nodes	Random Uniform
Speed of ferries	1,2,4,8 m/s
Network size	1000 x 1000 meter
Message generation interval	5 seconds
Message generation	1000 seconds
Number of runs	10 runs

- For all considered traffic models, a Pareto curve is seen. The least total traveled distance of ferries belongs to the setup which has the worst performance and vice versa.
- Increasing the speed of a ferry or ferries is a better choice than increasing the number of ferries with respect to the end-to-end delay of messages. It imposes also less cost in terms of the total traveled distance of ferries.
- Increasing the speed of ferries is not always an option due to the limitation of ferries. For this reason, increasing number of ferries should be considered as a solution for performance improvement.

By increasing the speed of ferries, the travel time of messages in ferries and their wait time in source nodes decreases. Regarding the travel time, it is obvious that faster ferries deliver messages faster. The wait time decreases since nodes are visited more frequent employing faster ferries.

4.5 Conclusion

In this chapter, a self-organized message ferry network was proposed where ferries employ an on-the-fly mobility decision maker. The on-the-fly decision maker applies only local observations of a ferry and ferries are coordinated through an indirect signaling. The performance of the proposed on-the-fly decision maker and the indirect signaling of ferries were studied considering three different traffic scenarios. The results showed that:



(a) All to all traffic

(b) All to sink traffic

(c) Broadcast traffic

Figure 4.12: Average end-to-end delay of messages vs. total traveled distance of ferries for different network setups

- Learning the message generation rate of nodes is never helpful in multi-ferry networks.
- The indirect signaling of ferries can reduce the travel and wait time of messages depending on the traffic scenario.
- The indirect signaling is beneficial having the all to all or all to sink traffic scenarios. With the broadcast traffic scenario, the indirect signaling of ferries has no advantage.
- The distance metric in the on-the-fly decision maker improves efficiency in all to all and broadcast traffic scenarios. It has no benefit considering the all to sink scenario
- A metric in the decision maker to reduce the wait time of messages is needed to serve waiting messages in nodes. However, due to more frequent visits to nodes in a network with a high number of ferries, this metric is not important anymore.
- The last visit time metric in ODMF is a better metric than the number of visits to nodes in SOMF to reduce the wait time of messages since the latter one causes the ”sink avoidance” and ”close to sink avoidance” problems in the all to sink traffic scenario.
- Increasing the number of ferries reduces the end-to-end delay of messages in all to all, all to sink and broadcast traffic scenarios. The reduction in the end-to-end delay of messages can be due to the reduction in the travel time or the wait time of messages and depends on the traffic scenario.
- Increasing the number of ferries reduces the wait time of messages in all considered traffic scenarios. However, it has no impact on the travel time in the broadcast scenario and a slight impact in the all to all traffic scenario.
- In the all to sink scenario, increasing the number of ferries has no impact on the travel time when FB and FBD are applied. With SOMF, there is a noticeable reduction in travel time by increasing the number of ferries.
- Increasing the speed of ferries is a better choice than increasing the number of ferries in terms of the end-to-end delay of messages.

5 Cooperation of Self-Organized Ferries

In the previous chapter, a message ferry network was proposed where a message is delivered to its destination traveling only in one ferry and there was no message exchange between ferries. However, ferries were cooperating implicitly since all of them were serving the same network. Figure 5.1 is a message ferry network with implicit cooperation of ferries. A message is collected by a ferry and delivered by the same ferry to its destination. This case was studied in Chapter 4. An arrow with a solid line represents a communication between entities and an arrow with a dashed line represents a travel path of a ferry. In the travel of a ferry from the source of a message to its destination, a ferry does not always travel directly to the destination of the message. It may visit other nodes based on the decision of the on-the-fly mobility decision maker. In Figure 5.1, ferry f_i visits node n_i and collects a message which should be delivered to node n_j. f_i travels directly or indirectly to n_j and delivers the message to it.

Another option for the cooperation of ferries is explicit cooperation where ferries exchange messages between each other to accelerate their deliveries. In this chapter, a message ferry network is proposed and studied where ferries can exchange messages directly or indirectly. By explicit cooperation of ferries in the delivery of messages, a message may travel in several ferries to be delivered to its destination. Therefore, a delay tolerant multi-hop communication among ferries takes place for the delivery of a message. Figure 5.2 shows a classification of different types of message exchange between ferries. In the class of direct message exchanges between ferries, ferries forward messages between each other or exchange replications of messages. A replication of a message is a copy of the message. In the case of a message replication between ferries, there are multiple copies of the message in a network.

The other class of message exchange between ferries is the indirect message exchange between ferries where ferries do not meet each other to forward or replicate messages, but they leave replications of messages in nodes. In this case, nodes act as relays to exchange the replication of messages between ferries.

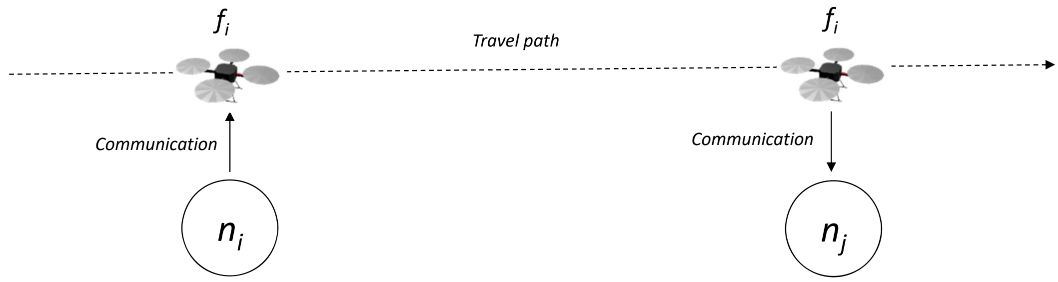


Figure 5.1: A message ferry network with an implicit cooperation of ferries

The motivation of message replication in nodes or between ferries in this chapter is to provide an opportunity for the replication of a message to be delivered earlier than the message itself.
As the first step, the explicit cooperation of ferries employing message forwarding between them is studied in the next section.

5.1 Direct message exchange of ferries by message forwarding

As mentioned earlier in this chapter, ferries can accelerate the message delivery procedure by explicit cooperation. To do this, they can forward messages between each other to establish a delay tolerant multi-hop communication. Each ferry visits nodes and collects messages from them. In its travel path to the next node, which is decided by the on-the-fly mobility decision maker, the ferry may meet other ferries. During a meeting to another ferry, some messages can be exchanged between ferries to accelerate their delivery. Figure 5.3 depicts two ferries which meet each other and exchange their messages. The message exchange can be in the form of a message forwarding or a message replication. In this section, the study is limited to the message forwarding between ferries and the message replication is studied later. In Figure 5.3, ferry f_i visits node n_i and collects a message which should be delivered to node n_j. f_i meets another ferry, f_j, in its travel path. f_i forwards the message to f_j to accelerate its delivery.
The procedure of message forwarding from a ferry to another one is shown in Figure 5.4. By message forwarding, ferry f_i sends message M_i to ferry f_j and discards it

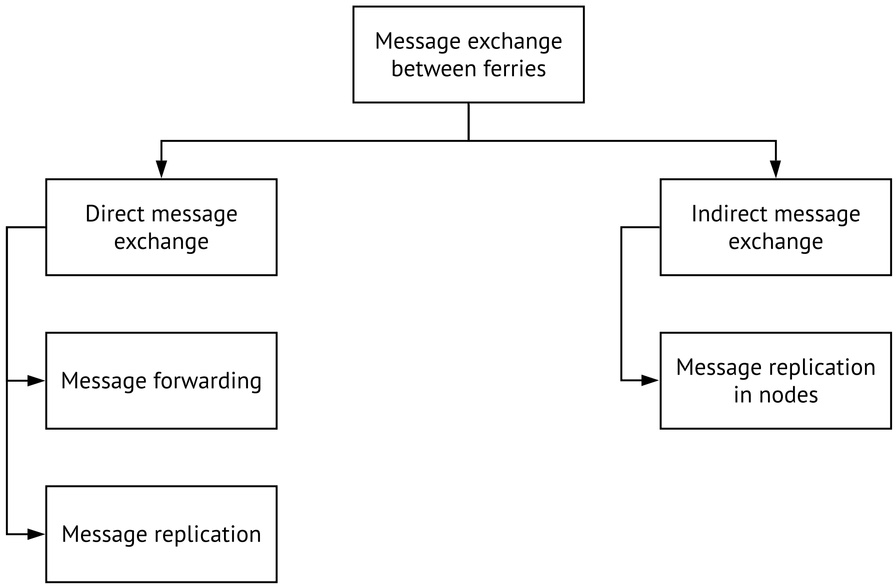


Figure 5.2: Classification of different types of message exchange between ferries for explicit cooperation in delivery of messages

from its memory. In this case, there is always a single instance of the message in the network.

Now, the following question should be answered:

"When to forward a message to accelerate its delivery?"

To make a decision on a message forwarding, some criteria can be defined. Considering the criteria, a ferry decides to forward the message, when it meets another ferry. An efficient message forwarding between ferries accelerates the message delivery. On the other hand, an inefficient message forwarding can impose extra travel delay for the delivery of a message.

In this thesis, signaling between ferries is proposed. The signaling occurs when ferries meet each other. Based on our assumptions, ferries have no global observation. Therefore, signaling between ferries in the time of a meeting provides a local observation for them. Ferries apply their local observations to make efficient decisions about the forwarding of messages. Table 5.1 shows the content of signaling between ferries. The signaling between ferries provides a view for a ferry about the current position of a neighbor ferry, the next node that it will visit (trajectory information) and its visits history.

Figure 5.5 is the flowchart which shows the steps of a meeting between two ferries. Signaling precedes any decision about the forwarding of messages or mobility of the

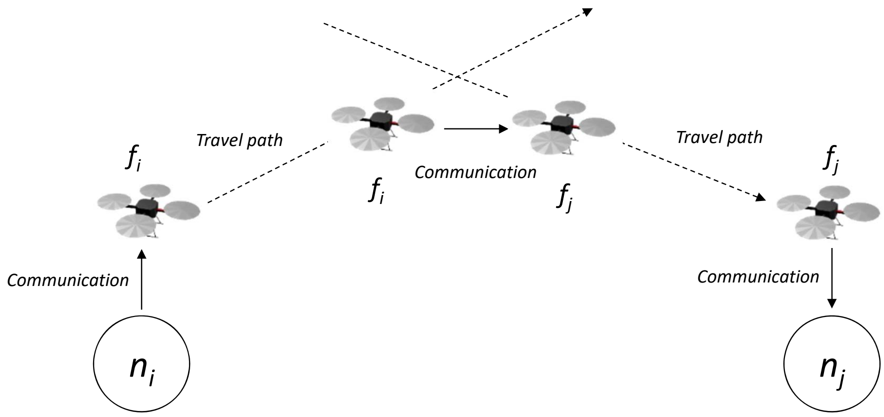


Figure 5.3: A message ferry network with explicit cooperation of ferries

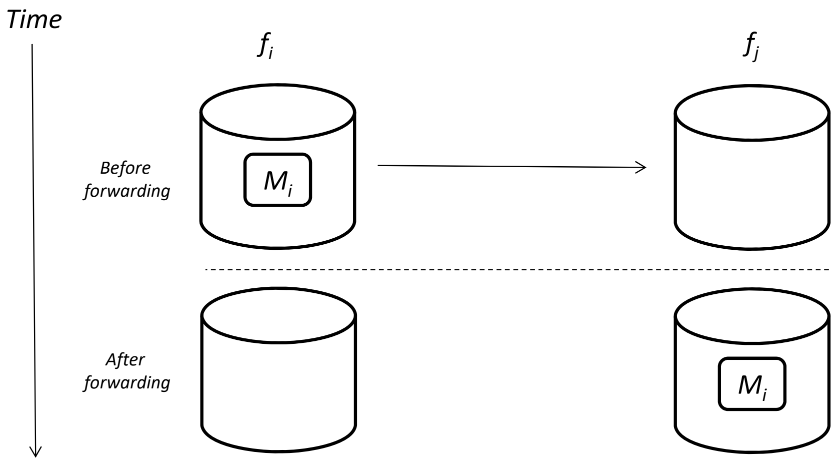


Figure 5.4: Message forwarding between two ferries

ferry. It should be noted that the message forwarding does not take place in every meeting between two ferries. In some cases, a ferry may decide to keep a message rather than forwarding it.

In this section, different schemes to forward messages between ferries are studied.

A set of variables to formulate the message forwarding schemes is defined in this section.

The variables store binary values based on the state of a ferry and its neighbor ferry for a message or set of messages to the same destination.

The following variables explain the state of ferry i which has a message to the destination node n_d in its buffer.

$$X = \{x_d, x_t, x_p\} \in \{0, 1\} \tag{5.1}$$

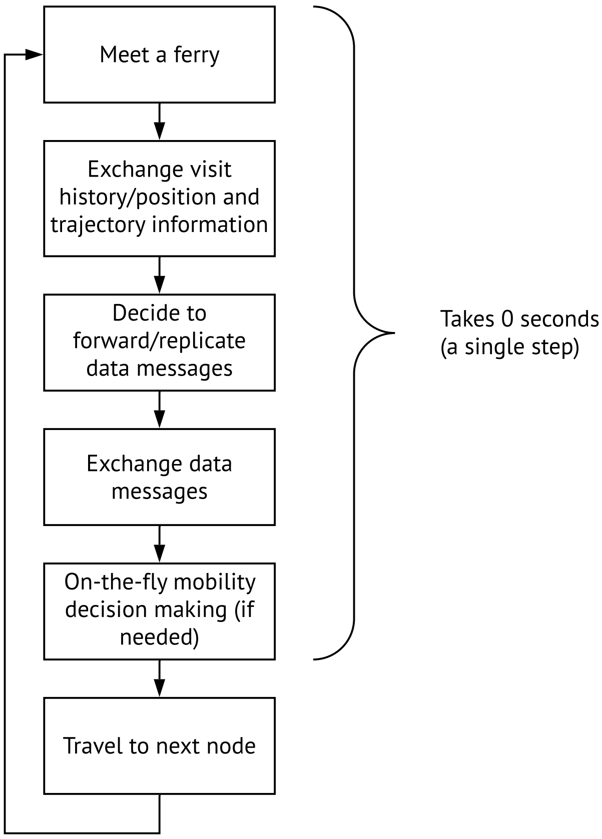


Figure 5.5: Steps in a meeting of ferries

All the variables in the following tables have the value 0, if the relevant condition for them is not met, otherwise they are 1. Table 5.2 defines the conditions.
The variables in set Y describe the state of a neighbor ferry j which is in a meeting with ferry i and a message to node n_d can be forwarded to it.

$$Y = \{y_d, y_t, y_p\} \in \{0, 1\} \tag{5.2}$$

Table 5.3 defines the conditions which the value of a variable is 1 if the relevant

Table 5.1: Content of a signaling between two ferries during their meeting

Ferry ID
Current position
Next node to visit
Visit history of nodes

Table 5.2: State variables of a ferry and their relevant condition

Variable	Condition
x_d	ferry travels directly to n_d
x_t	ferry travels towards n_d
x_p	ferry is closer to n_d than the neighbor

Table 5.3: State variables of a neighbor ferry and their relevant conditions

Variable	Condition
y_d	neighbor ferry travels directly to n_d
y_t	neighbor ferry travels towards n_d
y_p	neighbor ferry is closer to n_d than ferry i

condition is met.
The multiplication of the variables is same as a logical AND operation. The summation of the variables is same as a logical OR operation.
Following describes the studied message forwarding schemes applying the defined state variables.

5.1.1 Greedy forwarding (GF)

A greedy forwarding of messages between ferries exploits the idea of greedy geographical routing protocols in ad hoc networks [31]. A ferry forwards a message to its neighbor ferry if the neighbor is closer to the message destination. The decision is made in a ferry based on the received signaling information from the neighbor ferry. The message is forwarded to the neighbor ferry if the value of variable M_d^f is equal to 1. The value of the variable is calculated as follows:

$$M_d^f = y_p \qquad (5.3)$$

5.1.2 Destination aware forwarding (DAF)

A message is forwarded to a neighbor ferry if the neighbor ferry travels directly to the message destination. If both ferries travel to the message destination, the message is forwarded if the neighbor ferry is closer to the message destination. In this case, the neighbor ferry can deliver the message earlier and the goal is achieved. A message is

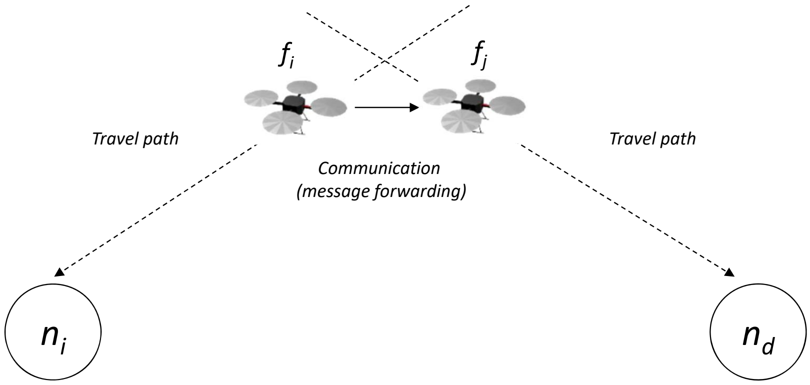


Figure 5.6: Destination aware message forwarding (DAF)

forwarded to a neighbor ferry if the value of variable M_d^f is equal to 1. The value of the variable is calculated as follows:

$$M_d^f = \bar{x}_d y_d + x_d y_d y_p \tag{5.4}$$

Figure 5.6 illustrates DAF where ferry f_i has a message to deliver to node n_d. f_i is traveling to node n_i. It meets another ferry, f_j, which is traveling directly to n_d. In this case, f_j can deliver the message earlier. Therefore, f_i forwards the message to f_j.

5.1.3 Trajectory aware forwarding (TAF)

The trajectory aware message forwarding is a proposed scheme in this work to make a decision on a message forwarding between ferries. TAF considers the trajectory information of ferries which is exchanged through signaling between ferries when they meet each other.

Applying TAF, a message is forwarded to a neighbor ferry, if the neighbor travels to a node which is closer to the message destination, while the ferry itself travels away from the message destination. This condition is taken into account since the distance metric impacts on the on-the-fly mobility decision maker of ferries. When a ferry visits a node, it may visit other nodes in its vicinity with a high probability. A message to n_d is forwarded, if variable M_d^f is equal to 1. The value of M_d^f is calculated as follows:

$$M_d^f = \bar{x}_t y_t \tag{5.5}$$

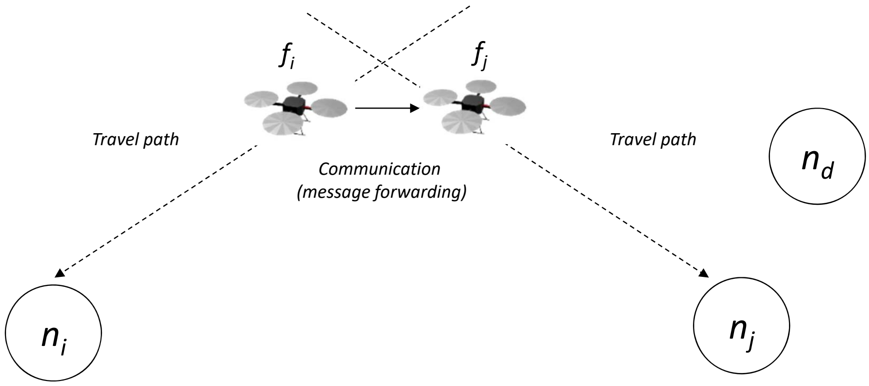


Figure 5.7: Trajectory aware message forwarding (TAF)

Figure 5.7 demonstrates TAF where ferry f_i has a message to deliver to node n_d. f_i is traveling to node n_i. It meets ferry f_j, which is traveling to node n_j. n_j is closer to n_d than n_i ($d(n_j, n_d) < d(n_i, n_d)$). Therefore, f_j **may** deliver the message earlier and f_i forwards the message to f_j.
It should be taken into consideration that a message forwarding condition in TAF does not guarantee the earlier delivery of a message since the decision of a message forwarding is only based on local observations of ferries and they cannot predict the future decisions of each other.

5.1.4 Performance evaluation of message forwarding strategies

The performance of message forwarding schemes is compared in terms of end-to-end delay of messages in this section. As the mobility decision of ferries can impact on the message forwarding decision, different combinations of on-the-fly mobility decision making schemes and message forwarding schemes are studied. Following on-the-fly mobility decision makers from Chapter 4 are chosen for our studies:

- FB: it is taken as the most simple mobility decision maker which only considers the state of a ferry buffer in its decision.
- FBD: it is taken to study the impact of the distance metric.
- ODMF: it is taken as a mobility decision maker which considers messages in a ferry buffer, messages in nodes and the distance. It was also the best approach based on our studies in Chapter 4 and is taken to achieve the best performance.

The message forwarding schemes in our studies are shown in Table 5.4. Same as

Table 5.4: Message forwarding schemes

Forwarding scheme	Definition
GF	Greedy Forwarding
DAF	Destination Aware Forwarding
TAF	Trajectory Aware Forwarding

Table 5.5: The simulation setup of studies in this section

Parameter	Value
Number of nodes	20
Number of ferries	10
Distribution of nodes	Random Uniform
Speed of ferries	5 m/s
Network size	1000 x 1000 meter
Radio transmission range of ferries	20 meter
Message generation interval	5 seconds
Message generation	1000 seconds
Number of runs	10 runs

Chapter 4, all studies are done considering all to all, all to sink and broadcast traffic scenarios, which were defined in Chapter 3. The simulation setup of the studies in this section is shown in Table 5.5.

5.1.4.1 All to all traffic

The performance of different combinations of mobility decision makers and message forwarding schemes is studied considering the all to all traffic scenario. Figure 5.8 shows the results of this study and the lessons learned from the results are as follows:

- Applying FB, message forwarding highly improves the performance of message delivery because a ferry delivers messages to a node with the highest number of messages in its buffer even if the node is far from the current location of the ferry. In this case, other traveling messages in the ferry buffer can be forwarded and have an opportunity to be delivered earlier.
- DAF has always the minimum improvement comparing with other approaches due to its forwarding conditions which are not met most of the time.

- GF and TAF always improve the end-to-end delay in this traffic scenario. However, TAF always performs better since it takes trajectory information instead of temporary location information into consideration. In GF, a message may be forwarded to a ferry which is closer to the destination but travels away from the destination. In this case, the travel time of forwarded messages is increased.

5.1.4.2 All to sink traffic

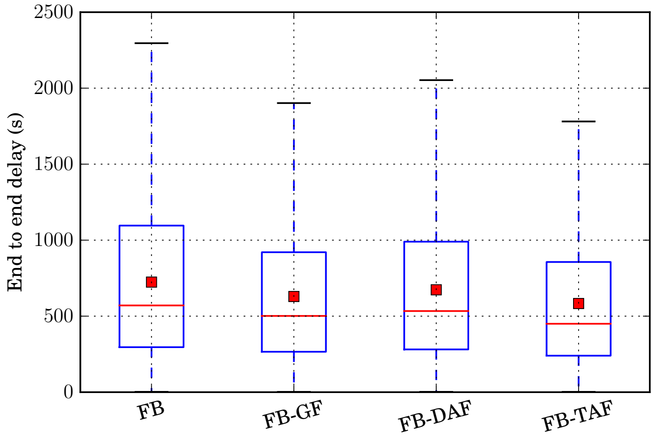
The performance of different combinations of mobility decision makers and message forwarding schemes is studied considering the all to sink traffic scenario. Figure 5.9 shows the results of this study and the lessons learned from the results are as follows:

- Message forwarding is not beneficial when ferries apply FB and FBD for the next-hop decision making. Applying FB and FBD, ferries travel directly from the source of a message to the sink node. Therefore, any message forwarding for earlier delivery of messages is useless.
- GF increases the end-to-end delay of messages with FB and FBD for the reason that a neighbor ferry which is closer to the sink may travel to other nodes than the sink if it has no message in its buffer.
- Applying ODMF, the message forwarding improves the end-to-end delay slightly while a ferry does not always travel directly from the source to the sink node and may visit some other nodes in between due to their last visit time.
- The impact of message forwarding in ODMF-DAF is even less than other forwarding approaches since there are very rare cases in which a ferry travels to other nodes than sink but the neighbor ferry travels directly to the sink node.

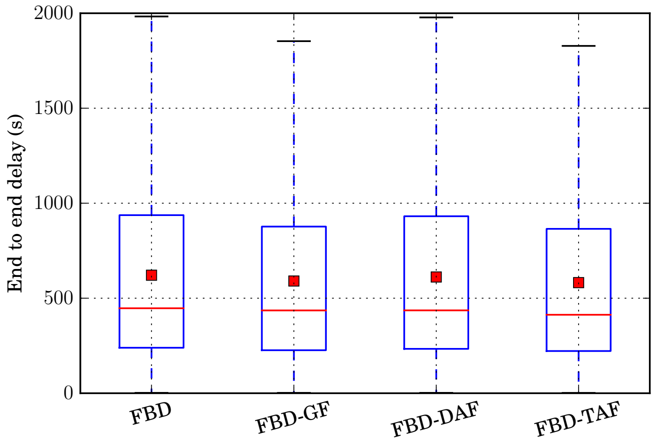
5.1.4.3 Broadcast traffic

The performance of different combinations of mobility decision makers and message forwarding schemes is studied considering the broadcast traffic scenario. Figure 5.10 shows the results of this study and the lessons learned from the results are as follows:

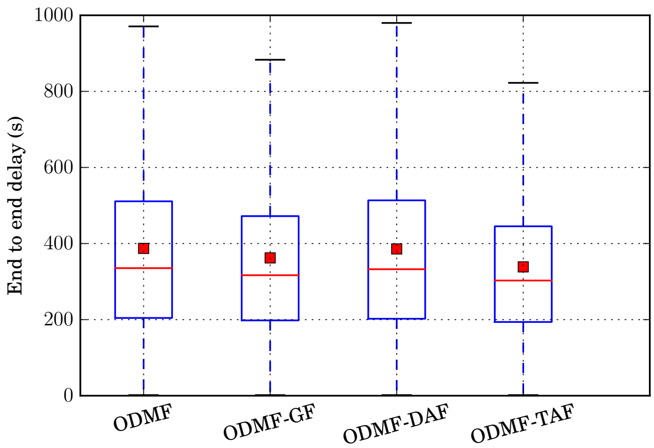
- Any type of message forwarding in combination with any type of next-hop decision making is highly effective on the end-to-end delay of messages while a single ferry collects messages from the broadcaster node and should deliver them. Therefore, message forwarding helps to distribute this task among ferries.



(a) FB as mobility decision maker

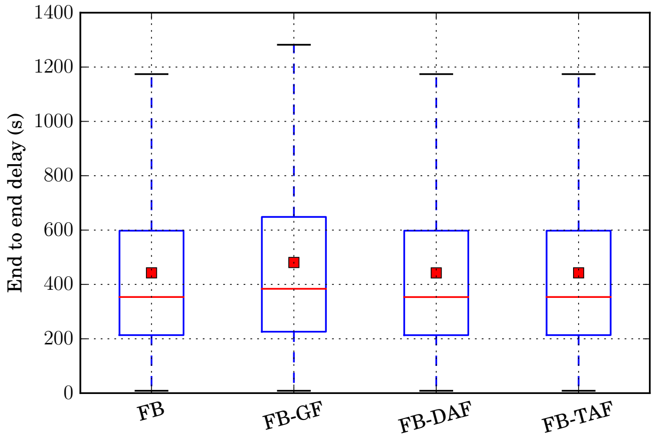


(b) FBD as mobility decision maker

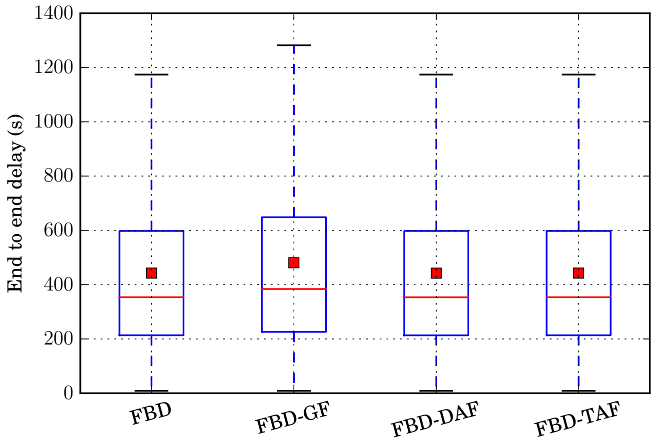


(c) ODMF as mobility decision maker

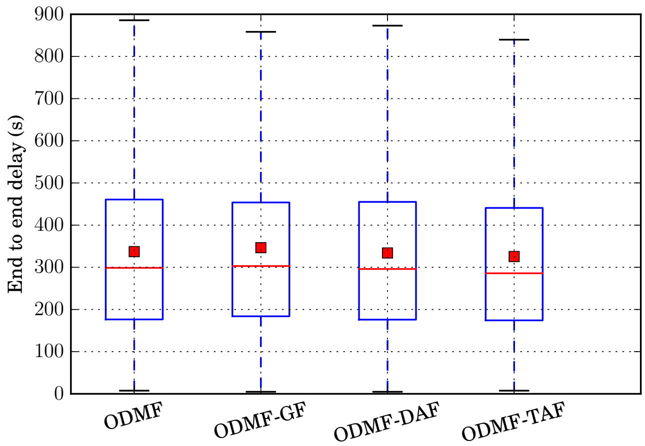
Figure 5.8: Study on message forwarding schemes applying different mobility decision makers considering the all to all traffic scenario



(a) FB as mobility decision maker

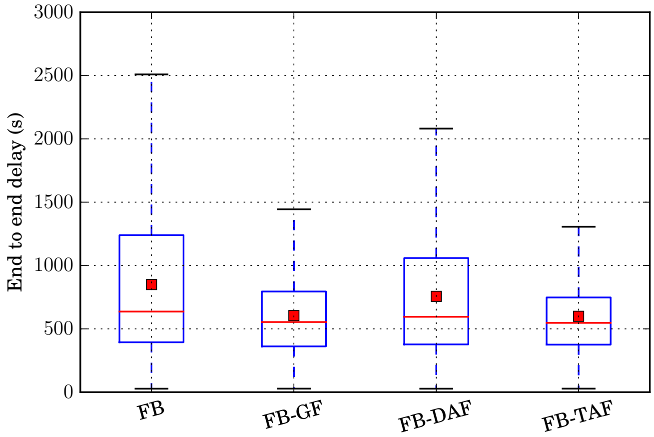


(b) FBD as mobility decision maker

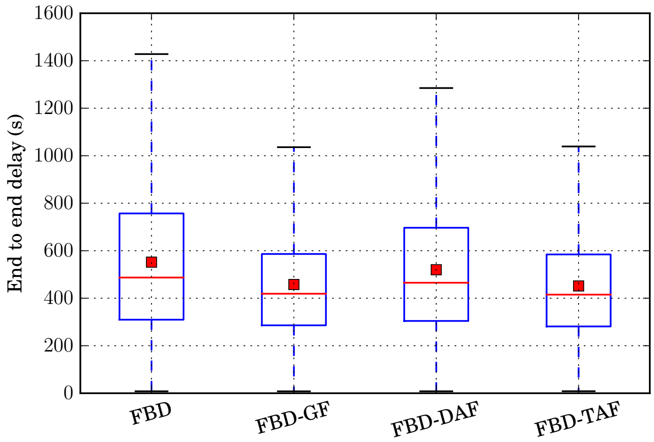


(c) ODMF as mobility decision maker

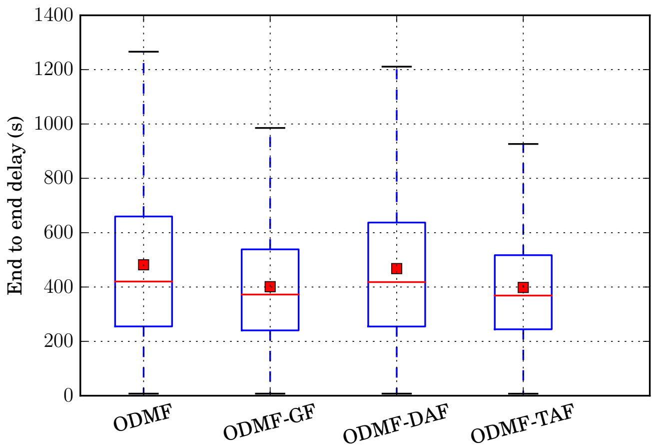
Figure 5.9: Study on message forwarding schemes applying different mobility decision makers considering the all to sink traffic scenario



(a) FB as mobility decision maker



(b) FBD as mobility decision maker



(c) ODMF as mobility decision maker

Figure 5.10: Study on message forwarding schemes applying different mobility decision makers considering the broadcast traffic scenario

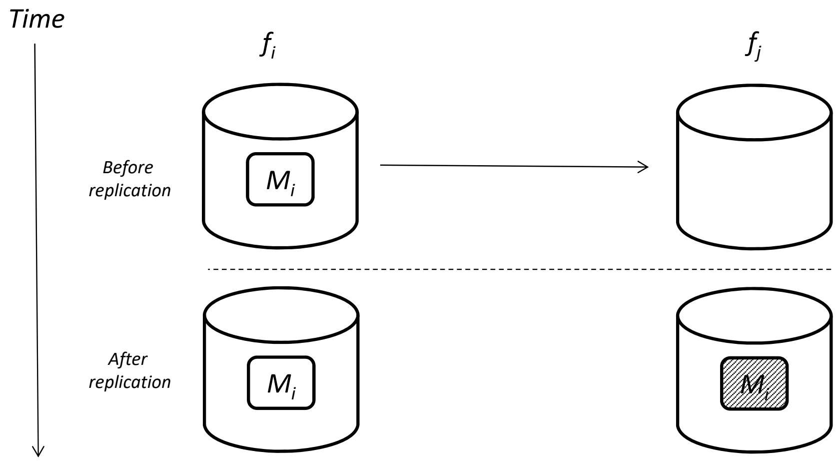


Figure 5.11: Message replication between two ferries

- The comparison of DAF, GF, and TAF in terms of performance is same as the all to all traffic scenario and lessons learned are still valid for this traffic scenario.

5.2 Direct message exchange of ferries by message forwarding and replication

In this section, another option for direct message exchange between ferries is studied. During a meeting between ferries, they can forward messages or exchange replication of messages. The difference between a message forwarding and replication is that a copy of the message is sent from a ferry to another one with replication of the message. There can be several copies of a message in a network when ferries replicate messages. Figure 5.11 shows a message replication procedure between two ferries. f_i meets f_j during its travel to its next node. f_i decides to replicate message M_i to f_j. f_i makes a copy from the message and sends the copy of the message to the neighbor ferry and keeps the original message in its buffer.
As the on-the-fly mobility decision maker in ferries considers the state of a ferry buffer in its decisions, multiple copies of a message cause multiple deliveries of the message. This imposes unnecessary redundancies to a network and wastes resources. To avoid redundancies in the delivery of messages, the original messages and their replications are differentiated in this thesis. To do this, the on-the-fly decision maker considers only original messages in a ferry buffer in its decisions and neglects the replication of messages. The original messages and the replication of messages are stored in two

different buffers as follows:

- Original messages buffer: it stores messages which a ferry collects from their source node or receives during message forwarding from other ferries. Messages in this buffer are considered in the on-the-fly mobility decision maker and influence the future mobility of the ferry.

- Replicated messages buffer: it stores messages which a ferry receives from other ferries during meetings with other ferries. They are replicas of original messages and the on-the-fly mobility decision maker of the ferry does not consider them for its mobility decision. Replication of messages in this buffer may be delivered to their destinations opportunistically earlier than the original messages.

In the proposed network of this thesis, only one ferry is responsible to travel to the destination of a message. Therefore, the redundancy in delivery of messages is avoided in our work.
The motivation of message replication is to accelerate message delivery by opportunistic delivery of replicated messages. Therefore, the following question arises:

When to replicate a message?

Several strategies are proposed and investigated in this section which a ferry decides to make a replication of a message. Although, message replication may reduce message delivery delay but replicating of messages impose extra memory usage in ferries. Therefore, the decision about when to replicate a message impacts on the performance and cost of a message ferry network.
To limit our studies to the different message replication strategies, it is assumed that the message forwarding scheme in ferries is the proposed trajectory aware message forwarding (TAF) scheme. In the following subsections, different message replication strategies are explained.

5.2.1 Pure replication (PR)

Following a pure replication strategy, a ferry replicates all messages in its buffers to a neighbor ferry. However, a ferry does not send a replication of a message to a neighbor ferry, if it decides to forward the message. The replicated messages are stored in the replicated messages buffer of the neighbor ferry. PR does not consider any criteria to select messages for replication. A message is replicated to a neighbor ferry if the value

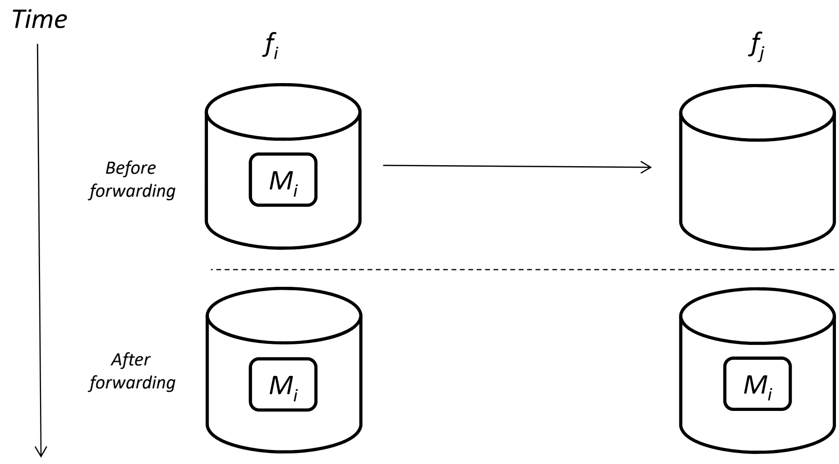


Figure 5.12: Forward and keep strategy (Keep)

of variable M_d^r is equal to 1. The value of this variable is calculated as follows:

$$M_d^r = \bar{M}_d^f \tag{5.6}$$

where M_d^f is the variable for the condition of the message forwarding.

5.2.2 Forward and keep (Keep)

This strategy is different from the message replication procedure which was explained in Figure 5.11. In the forward and keep strategy, a ferry forwards a message to a neighbor ferry applying TAF scheme but it does not discard the message from its original messages buffer. Applying this scheme, more than one instance of a message exists in the network as an original message. This scheme is studied to find the impact of the emerged redundancy from multiple instances of a message and the importance of the differentiation between original and replicated messages which is proposed in this thesis. Figure 5.12 illustrates Keep strategy where f_i forwards message M_i to f_j and keeps M_i as an original message in its buffer.

5.2.3 Trajectory aware replication (TR)

TR exploits the local observations of a ferry which are obtained from the signaling between ferries during a meeting. TR selects a message for replication if the following conditions are met for the message:

1. The message has not been selected for forwarding to a neighbor ferry.

2. The ferry is not traveling directly to the destination of the message.

3. The neighbor ferry is not traveling away from the destination of the message.

A message to destination n_d is replicated if the variable M_d^r has the value 1 and its value is calculated as follows:

$$M_d^r = \bar{M}_d^f \bar{x}_d y_t \tag{5.7}$$

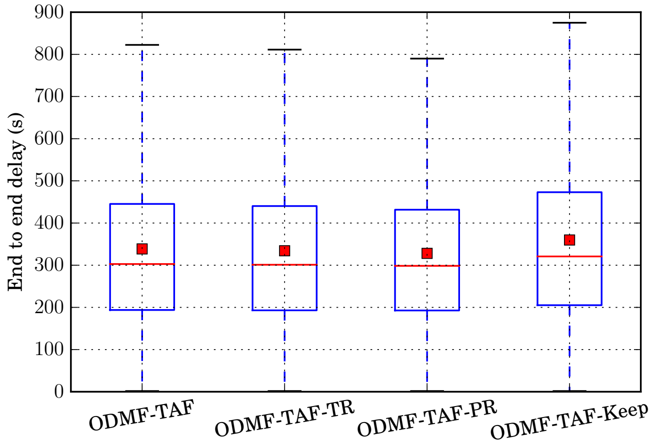
5.2.4 Performance evaluation of message replication strategies

In this section, the performance of message replication strategies is studied. In our studies, ODMF is taken as the on-the-fly mobility decision maker and TAF is taken as the message forwarding scheme since both showed the best performance in our previous studies. Figure 5.13 shows the results considering the all to all, all to sink and broadcast traffic scenarios. The lessons learned from the results are as follows:

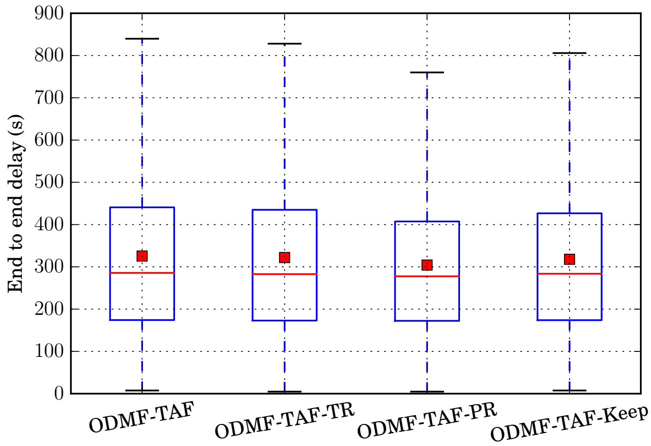
- Message replication between ferries slightly improves the end-to-end delay of messages. Even ODMF-TAF-PR which replicates all messages, except messages which are forwarded, to neighbor ferries does not show a noticeable improvement. This shows the effectiveness of the message forwarding strategy (TAF) which performs similar to the PR (flooding of replicas).

- The performance of TAF-TR is slightly better than TAF. It can be concluded that any message replication strategy than PR is not beneficial.

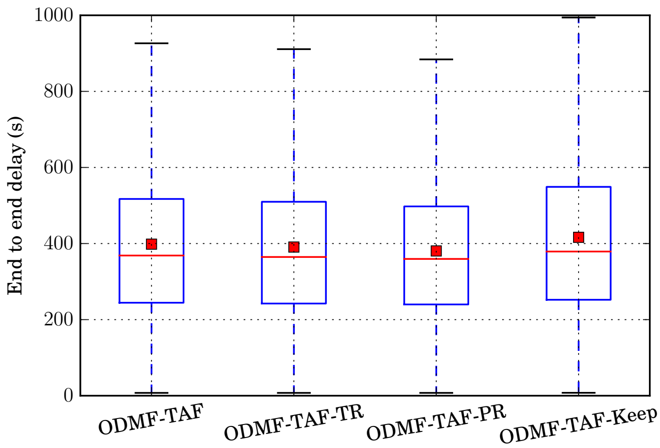
- Applying ODMF-TAF-Keep, a ferry forwards a message and keeps the message in its buffer. This behavior degrades the performance in all to all and broadcast traffic scenarios. More than a single instance of a message in a network causes redundant delivery of it. The redundant delivery of a message wastes resources and harms the delivery of other messages in the ferry buffer. However, ODMF-TAF-Keep has not any negative impact on the performance and slightly improves the maximum delay in the all to sink traffic scenario. The reason for this exception is the nature of the all to sink traffic where any redundancy does not have any negative impact on the delivery of other messages while all messages should be delivered to a sink node.



(a) All to all traffic



(b) All to sink traffic



(c) Broadcast traffic

Figure 5.13: Study on message replication schemes applying TAF as message forwarding and ODMF as mobility decision maker

5.3 Indirect message exchange of ferries by message replication in nodes

The message exchange between ferries aims to accelerate the delivery of messages. Ferries exchange messages in the form of a message forwarding or replication when they meet each other. A ferry can meet other ferries if they come inside its radio transmission range. Based on the assumptions of this work, there is no predefined location for ferries to meet each other. Therefore, any meeting of ferries is not predictable. When a ferry meets other ferries, messages are exchanged and a multi-hop communication is established for delivery of messages. The meeting of ferries is crucial to apply message forwarding and replication schemes. In wide networks, ferries may meet each other rarely. In this case, the message forwarding and replication schemes which were proposed earlier in this section cannot be applied to accelerate the delivery of messages.

In this section, a solution for an indirect message exchange between ferries is proposed. The proposed approach makes the multi-hop delivery of messages possible and a message is delivered to its destination traveling in several ferries without any need for a meeting of ferries.

In the proposed network of this thesis, nodes are employed as delay tolerant relays to exchange messages between ferries. Ferries leave the replication of messages in nodes. A node stores the replication of a message for a limited time. During this time, if another ferry visits the node, it receives the message replica from the node. Therefore, ferries do not need to meet to exchange messages. The messages are exchanged indirectly among them through nodes. Figure 5.14 shows node n_i which acts as a delay tolerant relay between ferries f_i and f_j. n_i receives a replication of message M_i at time t_1 from f_i and stores it in its buffer. At time t_2 another ferry, f_j, visits the node and receives the replication of M_i. In this way, the replication of M_i is sent indirectly from f_i to f_j.

The idea of employing nodes as relays between ferries is similar to the indirect signaling of ferries which was proposed in Chapter 4. The difference is on the goal of them. In the indirect signaling of ferries, nodes are employed to exchange control information between ferries. The control information is applied in the mobility decision maker of ferries to avoid redundant visits to nodes. On the other hand, the main objective here is to form an indirect message exchange between ferries to accelerate the delivery of data messages.

The reason for the replication of a message in nodes instead of forwarding them is the

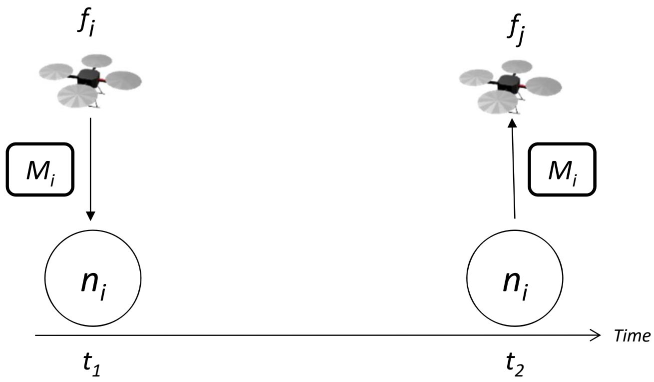


Figure 5.14: A node as a relay between two ferries for indirect communication

uncertainty about the future in this type of network. Ferries are assumed to have a local observation. They have no knowledge about other ferries and their future decisions. A ferry cannot forward messages to relay (intermediate) nodes since messages may have to wait in them for a long time to be collected by other ferries. Therefore, a replication of messages is sent to nodes. The replication of a message can be delivered to its destination earlier than the original message or later than it.
As mentioned earlier for differentiation of original and replicated messages, the replicated messages are stored in a separate buffer in nodes and a timer is set for each of them to calculate their buffering time. If the timer goes to zero for any message, it is discarded from the buffer of the relay node. It should be noted that only original messages are considered in the mobility decision maker of ferries.
In following sections, different schemes to select relay nodes for messages are studied. A ferry can choose a node as the relay for a message and send a replication of the message to it. Figure 5.15 shows the decision on message replication to a node in a flowchart of a ferry visit to a node. The message replication decision takes place after the mobility decision making of a ferry. For this reason, the ferry can apply the mobility information in its decision to select the node to replicate a message. Different strategies can be applied for this selection which are described and studied in this work.

5.3.1 Flooding in all nodes (Flood)

The most straightforward solution to replicate messages in relay nodes is flooding. Applying this strategy, a ferry sends a replication of all waiting messages in its buffers to any node which it visits. Therefore, there is no condition to limit a ferry in relay

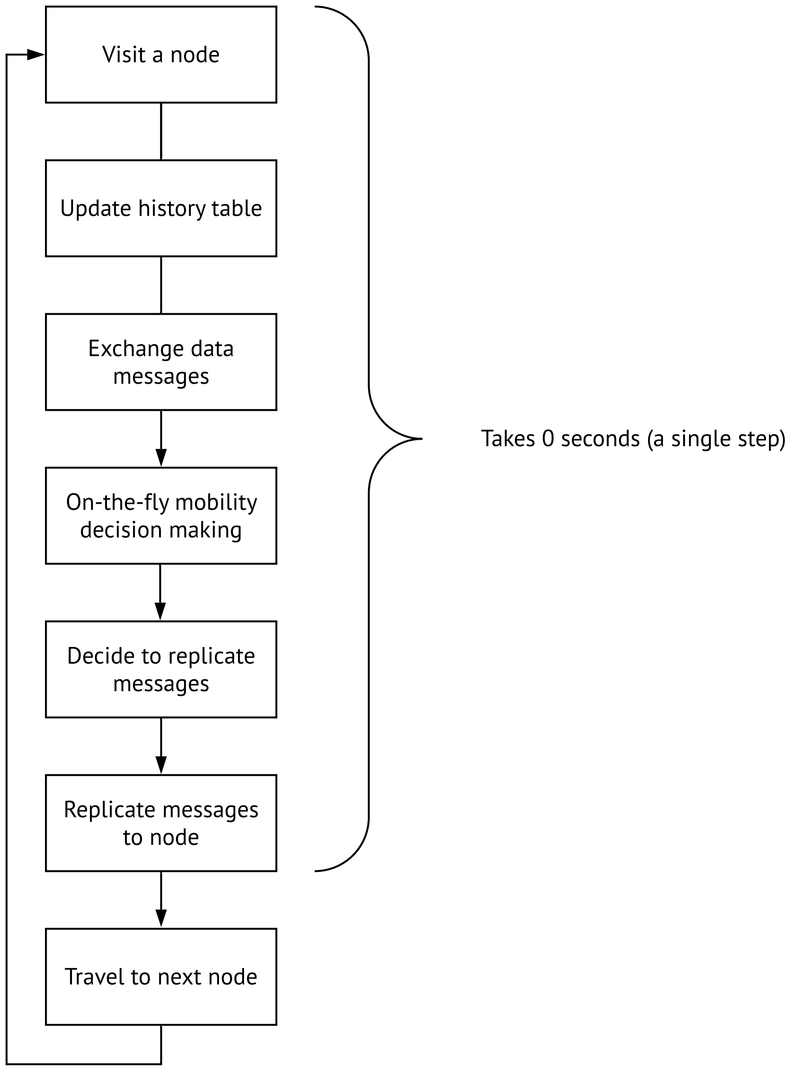


Figure 5.15: Steps in a ferry visit to a node including a message replication in the node

selection. It is obvious that this approach increases the probability of faster delivery for the replication of a message. However, it increases the costs in a message ferry network in terms of the number of message transmissions, the lengths of buffers in nodes and ferries and the processing time.

5.3.2 Zone-based replication (Zone25/50)

To limit the number of message replications in the Flood strategy, a zone-based approach is proposed. In the zone-based strategy, a zone around the destination of a message is considered and the message is replicated into all nodes inside the zone.

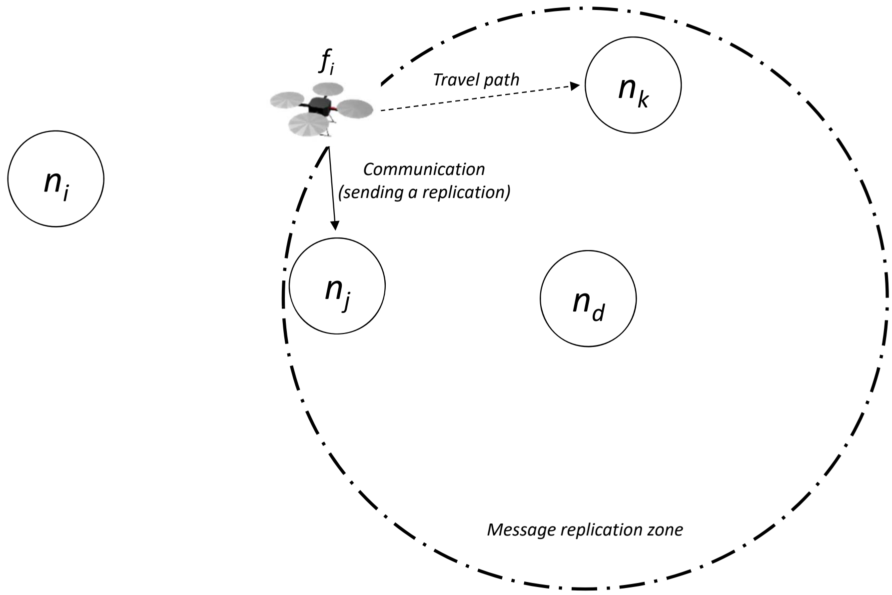


Figure 5.16: Zone-based replication of messages in nodes (Zone25/50)

The zone is a circle in which the location of the message destination is considered as the center of it and the radius of the circle is a percentage of the distance between the source and destination of the message. In this thesis, two different sizes of zones are studied. The radius sizes of zones are 25% and 50% of the distance between the message source and destination. It is assumed that if the replica of a message is stored in close nodes to the message destination, it may be collected by other ferries and be delivered to the destination earlier than the original message. On the other hand, nodes outside a zone are far from the message destination. Even if a ferry collects the replication of a message from a node which is far from the message destination, the probability of its earlier delivery is low. Figure 5.16 illustrates a zone-based approach. Ferry f_i collects a message from node n_i which must be delivered to node n_d. f_i visits n_j, which is inside the defined zone, and decides to visit n_k as the next node. Thus, it replicates the message in n_j.

5.3.3 Trajectory aware replication (TAR)

A trajectory aware relay selection algorithm is proposed in this work which replicates a message in a node if a trajectory condition is met for a ferry. In this approach, information from the on-the-fly mobility decision maker is provided to TAR strategy

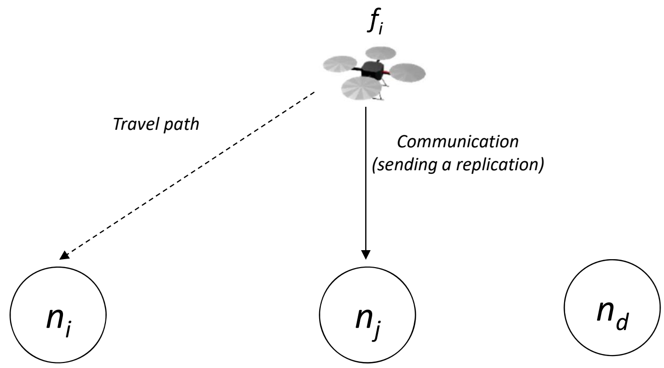


Figure 5.17: Trajectory aware replication of messages in a node (TAR)

in a ferry. A ferry replicates a message in a node if following condition is met:

$$d(n_i, n_d) > d(n_j, n_d) \tag{5.8}$$

where n_i is the next node for the ferry to visit, n_j is the current node and n_d is the destination of a message. A ferry replicates a message to a node if it will travel away from the message destination. On the other words, the next node which the ferry will visit is farther than the current node to the destination of a message. Therefore, it is reasonable to replicate the message in the current node and allow the replica to be delivered to the destination by other ferries. Figure 5.17 shows this scenario about TAR. n_j, n_i and n_d are current visiting node, next node to visit and the destination of a message, respectively.
In the next section, the performance and imposed cost of each strategy for relay selection are studied.

5.3.4 Performance evaluation of relay selection strategies

The end-to-end delay of messages and the length of buffers in nodes are studied in this section as performance and cost metrics to compare different strategies for relay selection. To measure the end-to-end delay of messages, the generation time of a message and the delivery time of the message or its replica is taken into account. The one which is delivered earlier is considered in the calculation of the end-to-end delay to study the performance of each relay selection strategy. Moreover, to find the impact of the mobility decision maker in ferries on the relay selection strategies, different combinations of on-the-fly mobility decision makers and relay section schemes are studied.

Table 5.6: Simulation setup of studies on the relay selection strategies

Parameter	Value
Number of nodes	20
Number of ferries	10
Distribution of nodes	Random Uniform
Speed of ferries	5 m/s
Network size	1000 x 1000 meter
Message generation	1000 seconds
Number of runs	10 runs

The simulations setup is shown in Table 5.6. The considered traffic scenarios are the all to all, all to sink and broadcast same as all previous studies in this work. The studied on-the-fly mobility decision makers are limited to FB, FBD, and ODMF.

5.3.4.1 All to all traffic

The performance of different relay selection strategies is studied considering the all to all traffic scenario. Figure 5.18 shows the results of our studies. The following lessons are learned from the results:

- All strategies for message replication in nodes have a minor impact on the end-to-end delay if the next-hop decision maker is FB or FBD. Applying FB and FBD, the average traveled hops of messages is almost two hops based on our studies in Chapter 4. Therefore, a ferry visits only one node between the source and destination of a message on average. Leaving a replication of a message in an intermediate node is not beneficial most of the time. Even Flood has a small impact on the performance.
- Applying ODMF, the average number of traveled hops for a message is almost five hops based on our previous studies. Therefore, there is an opportunity for the replication of a message to be collected by other ferries from an intermediate node and delivered to its destination earlier than the original message.
- The zone-based approaches do not show any noticeable impact on the performance of message delivery due to the distance metric in ODMF. When a node is visited, which is located close to the destination, the destination will be the next to visit with a high probability. Therefore, leaving a replication of a message in a zone close to the destination is not always helpful.

- TAR has a similar performance to Flood. To find the gain of this approach, the buffer length in nodes will be studied.

5.3.4.2 All to sink traffic

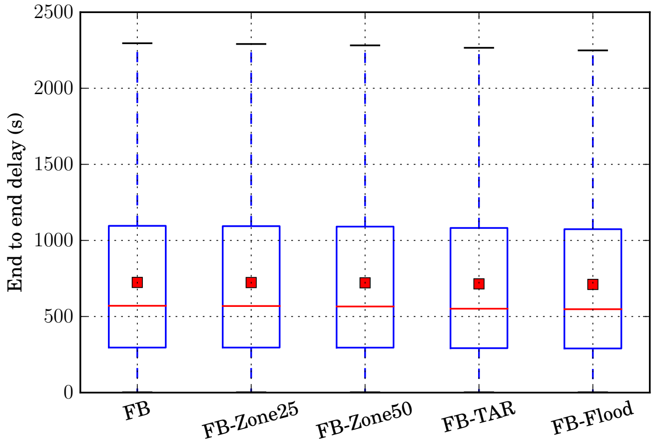
The performance of different relay selection strategies is studied considering the all to sink traffic scenario. Figure 5.19 shows the results of our studies. The following lessons are learned from the results:

- Applying FB and FBD, message replication in nodes does not occur at all while a ferry travels directly from the source of a message to the sink node. The average traveled hops of messages is always one hop applying FB and FBD.
- Applying ODMF, message replication in nodes is helpful while a ferry does not always travel directly from the source to the sink node and may visit some nodes in between due to the last visit time history which it considers in its decision making.
- Applying ODMF, zone-based approaches have no impact on the performance and any message replication in nodes is useless. If a node is visited in the zone, the next-hop is the sink node with a high probability due to the state of a ferry buffer and the distance metric in ODMF.
- TAR shows a good performance which is similar to Flood. Flooding can be considered as the best case since messages are replicated in all nodes and can be delivered to the sink through the shortest path.

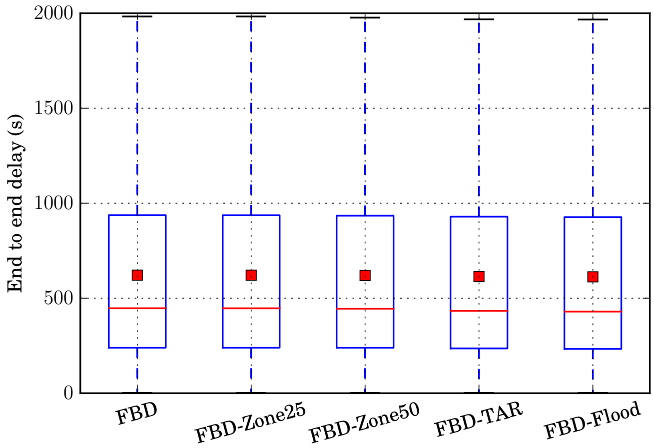
5.3.4.3 Broadcast traffic

The performance of different relay selection strategies is studied considering the broadcast traffic scenario. Figure 5.20 shows the results of our studies. The following lessons are learned from the results:

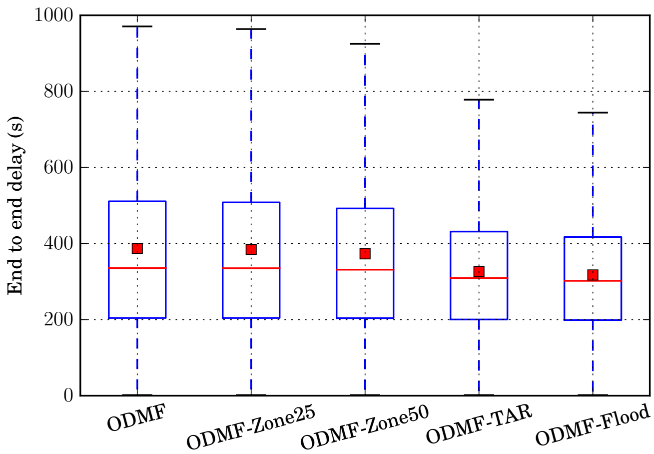
- Message replication in nodes is helpful most of the time in this traffic scenario regardless of the mobility decision maker since a single ferry must deliver all messages to all nodes and messages experience a long travel time. Message replication in nodes causes cooperation of ferries and reduces the travel time of messages.



(a) FB as mobility decision maker

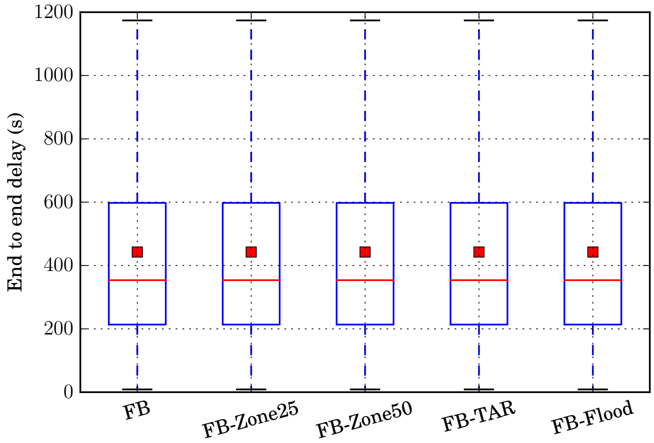


(b) FBD as mobility decision maker

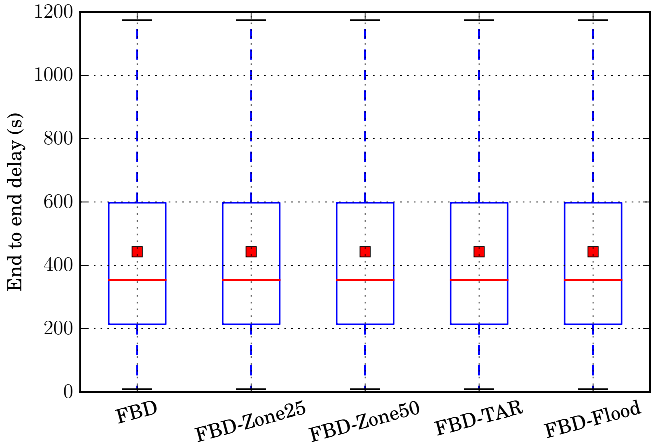


(c) OMDF as mobility decision maker

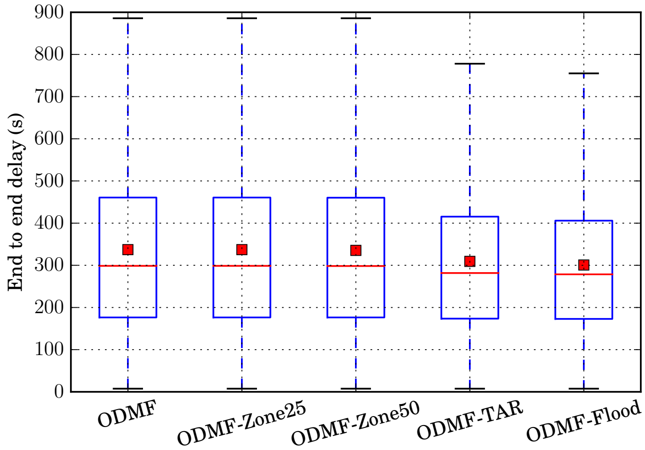
Figure 5.18: Study on different combinations of relay selection strategies and mobility decision makers considering the all to all traffic scenario



(a) FB as mobility decision maker



(b) FBD as mobility decision maker



(c) OMDF as mobility decision maker

Figure 5.19: Study on different combinations of relay selection strategies and mobility decision makers considering the all to sink traffic scenario

- TAR is again the closest approach to Flood.
- Applying FB, even zone-based approaches show some improvements in the end-to-end delay while a ferry can be very close to the destination of a message but decides to travel to a far node.
- Zone-25 is not helpful applying FBD because of the distance metric in FBD. If a ferry is too close to the destination, the next node to visit is the destination. Thus, there is no benefit to replicate the message in a node.

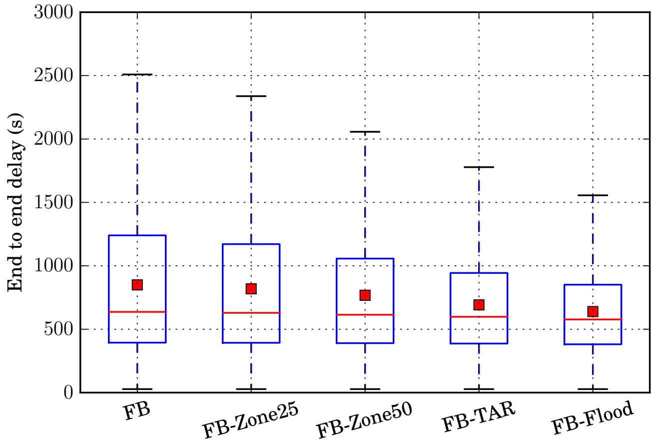
5.3.4.4 Study on the buffer length in nodes

The buffer length in nodes is an important metric which shows the cost of each relay selection strategy. The buffer length in nodes is studied considering all to all, all to sink and broadcast traffic scenarios. The simulation setup is same as Table 5.6. Figure 5.21 shows the buffer length in nodes applying different relay selection strategies. The applied mobility decision maker in all studies is ODMF. The following lessons are learned from the results:

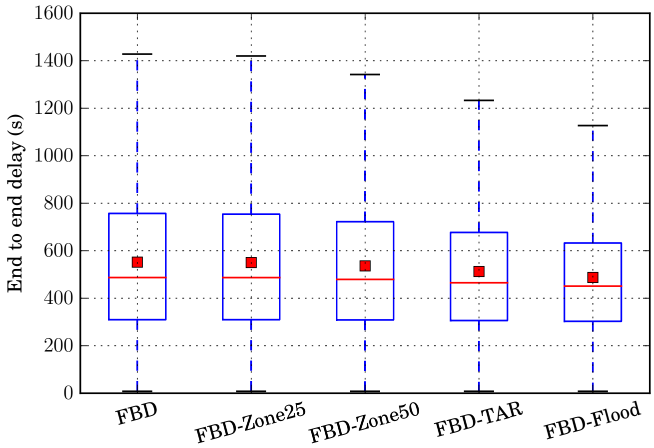
- Zone25 always causes the shortest buffer length in nodes and the minimum improvement on the performance.
- Zone50 causes 3 to 5 times longer buffer length in nodes than Zone25 but provides a slight improvement on the performance in the all to all scenario and almost no improvement in the all to sink scenario comparing with Zone25.
- TAR which performs similar to Flood causes about 50% shorter buffer length in nodes.

5.4 Conclusion

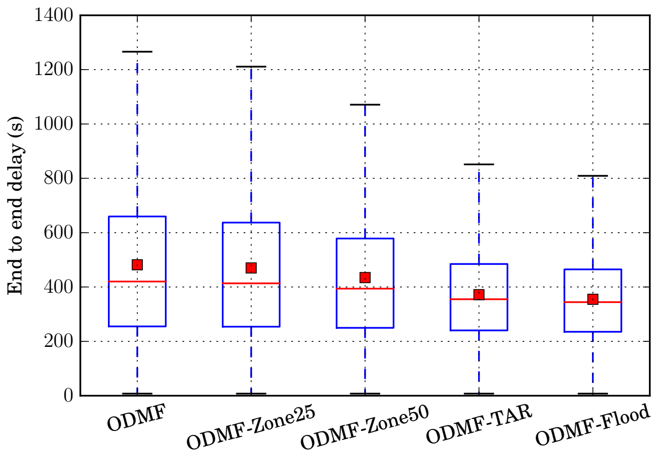
In this chapter, different approaches for the cooperation of ferries were proposed and studied. The cooperation of ferries occurs when messages are exchanged between ferries and a multi-hop delivery of messages is formed. The message exchanges between ferries are done directly or indirectly. By a direct message exchange between ferries, they can forward messages to each other or send the replication of messages when they meet. Different strategies for a direct message forwarding and replication between ferries were proposed and studied. Different mobility decision makers from Chapter 4 were



(a) FB as mobility decision maker

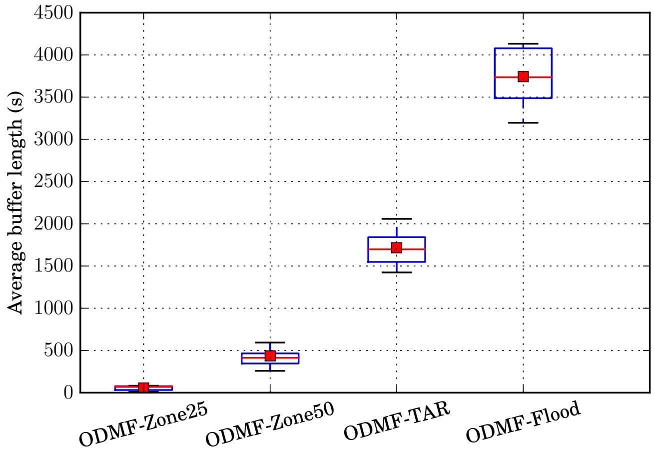


(b) FBD as mobility decision maker

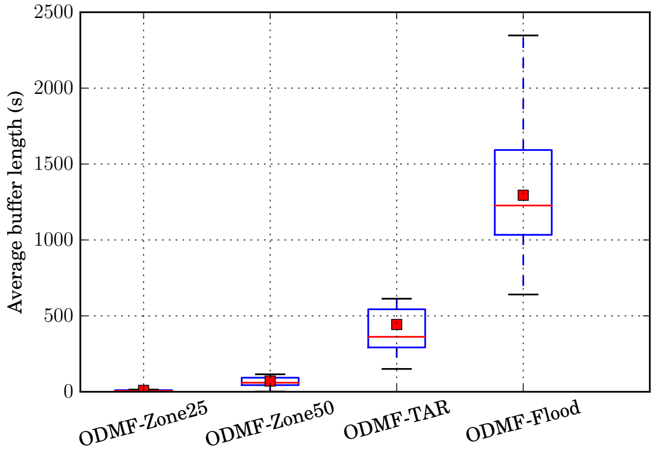


(c) OMDF as mobility decision maker

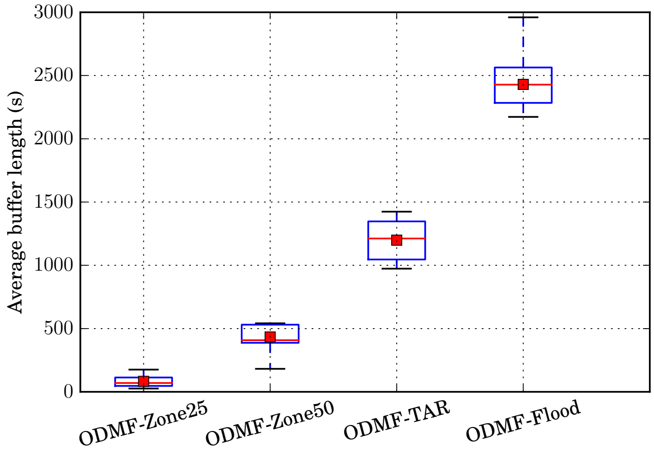
Figure 5.20: Study on different combinations of relay selection strategies and mobility decision makers considering the broadcast traffic scenario



(a) All to all traffic



(b) All to sink traffic



(c) Broadcast traffic

Figure 5.21: Study on the buffer length in nodes applying ODMF as the mobility decision maker and different relay selection strategies

considered to be applied in ferries to find the impact of the mobility decision maker on the performance of message forwarding schemes. Moreover, an indirect message exchange between ferries was proposed where ferries do not need to meet for message exchange. They employ nodes as relays to exchange the replication of messages. Same as message forwarding schemes, the impact of the mobility decision maker of ferries on different relay selection strategies for message replication in nodes was studied. The following points are the conclusion of our studies:

- Message forwarding between ferries is more beneficial in the broadcast scenario than all to sink and all to all traffic scenarios.
- The on-the-fly mobility decision maker has a big impact on the performance of message forwarding schemes.
- Message replication between ferries does not reduce significantly the end-to-end delay of messages. It can be concluded that TAF is an efficient scheme for message forwarding which any message replication strategy is almost useless.
- The differentiation of original messages and their replications is necessary while considering replicated messages for the mobility decision making causes a degradation of the performance in all to all and broadcast traffic scenarios. However, the differentiation is not important in the all to sink traffic scenario since any redundancy in the delivery of messages is harmless due to the nature of the traffic scenario.
- The message replication in nodes is an efficient approach for indirect message exchange between ferries in all traffic scenarios, it can achieve the same performance as the direct message exchange between ferries. The advantage of message replication in nodes for indirect message exchange between ferries is that they do not need to meet each other to exchange messages.
- TAR for message replication in nodes achieves similar performance as Flood with about 50% less overhead. It can be concluded that considering the trajectory information of ferries for the relay selection saves resources in a message ferry network.

6 Offline Path Planning for Multiple Ferries

As mentioned in Chapter 1, the mobility of ferries can be controlled either on-the-fly or offline. An on-the-fly approach was proposed earlier in this thesis and several schemes were applied to improve the performance of message delivery. An offline approach to control the mobility of ferries is studied in this chapter.

The path of ferries can be planned in advance in a path planner instead of making an on-the-fly decision in ferries. The path planner finds the (optimal) path of each ferry. After the start of a mission, each ferry only visits nodes based on the given path by the path planner.

To find the path of ferries in the path planner, the problem of path planning for multiple ferries is modeled as mTSP. In Chapter 2, mTSP was described as a generalization of TSP which is applied to model the path planning problem for multiple ferries. In mTSP, all ferries start/finish their tour from/in a depot (or a node). A subset of nodes is visited by each ferry and each node is visited exactly once. The objective of the optimization in mTSP is to minimize the total traveled distance of ferries. Therefore, a solution for mTSP must cluster the nodes and find the path of ferries inside the clusters.

Same as TSP, mTSP is a NP-hard problem which the optimal solution for it cannot be found in a reasonable time for a big number of nodes and ferries. For this reason, a heuristic algorithm is usually applied to approximate the optimal solution faster. In the next section, a genetic algorithm is described which is applied to find the solution for the path planning problem.

6.1 Genetic algorithm

The genetic algorithm is an evolutionary algorithm in computer science which has been derived from the biological evolution. It can search in complex problems for a high quality solution with respect to the objectives of the search problem using bio-inspired

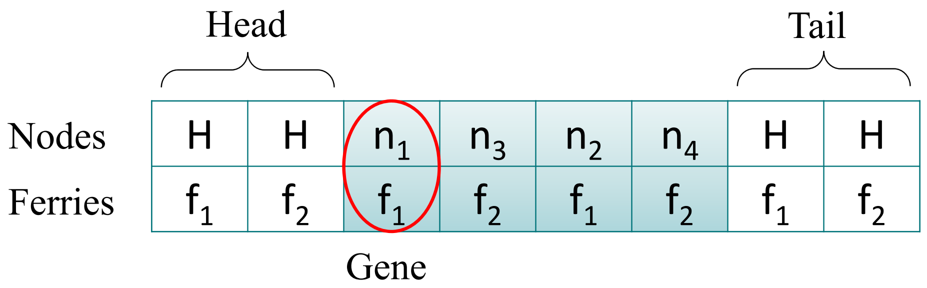


Figure 6.1: A chromosome which represents a solution for the path planning of multiple ferries

operations. To apply the bio-inspired operations, a solution for a given problem must be encoded into a chromosome. A chromosome is a solution for the given problem which is evolved into the optimal solution after several (or many) iterations using the bio-inspired operations. The genetic operators search the solution space and the quality of each chromosome is evaluated by a measure with respect to the objectives of the optimization. A selection procedure pushes the evolution towards the optimal solution.

Algorithm 6.1 represents the steps in the genetic algorithm:

1: *Initial_population_generation*
2: *Fitness_calculation*
3: *Selection*
4: *Cross_over*
5: *Mutation*
6: **if** $Stop\_condition \neq True$ **then**
7: *Goto_step_2*
8: **end if**

Algorithm 6.1: Steps in the genetic algorithm

In next sections, all steps of the genetic algorithm to approximate the solution of the path planning problem are described.

6.1.1 Initialization

The steps in the genetic algorithm start with the generation of a random population from the possible solutions of a given problem. To cover the solution space, the solutions are generated randomly with a uniform distribution. Each solution is generated in the form of a chromosome. A chromosome is an encoded solution for the path plan-

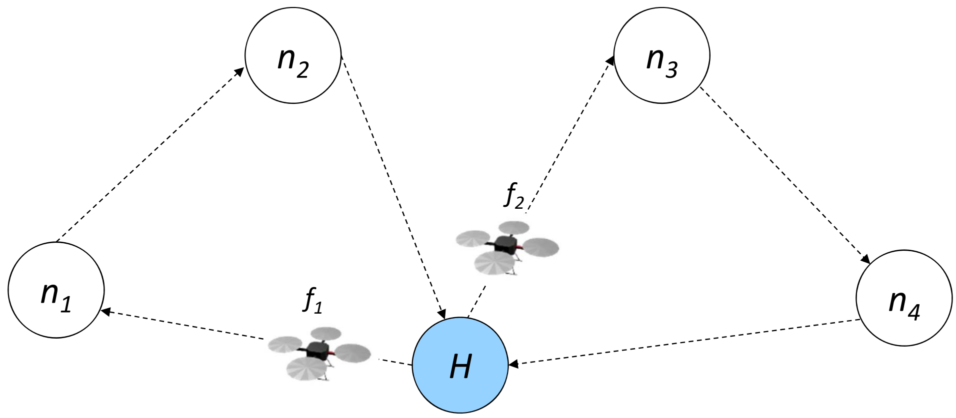


Figure 6.2: The illustration of the network encoded in the chromosome

ning problem. It consists a set of genes which build a chromosome. Encoding solutions to chromosomes makes the genetic operations possible.

Figure 6.1 illustrates a chromosome for the path planning problem of multiple ferries in a message ferry network. The first row of a chromosome is the set of nodes and the second row is the set of ferries. A gene in this problem is a combination of a node and a ferry which builds a column of a chromosome and represents an allocation of a node to a ferry. The allocation of a node to a ferry means that the node is visited by a specific ferry. Each chromosome starts and ends with a head and tail, respectively. The head and tail of a chromosome represent the start and end point for a ferry tour. A tour of a ferry starts from a depot which is shown with H in the chromosome and ends in the depot.

Each chromosome represents a solution for the path planning problem which consists of two parts. The first part of the solution is clustering of nodes. The solution divides nodes to subsets, which are called clusters, and allocates a ferry to each cluster. In the second part, the solution defines the order of nodes in each cluster where a ferry must visit the nodes. In the illustrated chromosome, all genes which have the same ferry index in their second row belong to the same cluster. Moreover, the order of genes appearance from a cluster in a chromosome represents the order of nodes inside a cluster or in other words, scheduling of nodes inside a cluster to be visited by the ferry. Figure 6.2 shows the corresponding network for the encoded solution in the chromosome of Figure 6.1. Both ferries start their tours from the depot H and each ferry visits a subset of nodes.

6.1.2 Fitness function

The quality of each solution, which is a chromosome, should be evaluated to let the evolution to take place. For this reason, a fitness function is defined for a given problem to the genetic algorithm.
In the given problem for path planning of multiple ferries, the main objective is to minimize the delivery delay of messages between isolated nodes. However, to evaluate the average end-to-end delay of messages for a path planning solution, knowledge about the message flows between nodes is needed which is missing based on our general assumptions for the message ferry network. Therefore, the only possible optimization for such a scenario is the minimization of the total traveled distance of ferries same as mTSP. The fitness function in the path planning problem is as follows:

$$F(s) = \sum_{i}^{N} \sum_{j}^{N} x_{ij} d(i,j) \tag{6.1}$$

where s is a solution which is encoded in a chromosome, N is the set of the nodes, x_{ij} is binary value which is 1 when a path exists from node i to j in the solution and $d(i,j)$ is the distance between the nodes.

6.1.3 Selection

To evolve into the optimal solution, a set of solutions should be selected in each iteration, i.e. a new generation of the genetic algorithm. The algorithm applies the fitness value for each solution to make the selection. There are different strategies to make the selection such as deterministic and probabilistic. In the deterministic approach, the solutions are sorted and a subset of them starting from the best solution is selected to be kept for the next iteration of the algorithm. In the probabilistic approach, a probability is assigned to each solution and solutions are selected based on their probabilities. The assignment of probabilities to the solutions is based on their fitness values.
In this work a probabilistic selection approach is applied where the probability of each solution to be selected for the next generation is calculated as follows:

$$p_s = 1 - \frac{F(s)}{\sum_{i=1}^{k} F(i)} \tag{6.2}$$

where s is a solution in a generation and k is the total number of solutions in the generation.

6.1.4 Genetic operators

In the genetic algorithm, the new generation of solutions is derived from the existing generation. To do this, genetic operators are applied. The solutions in the current generation of the algorithm are called parents and the generated solutions for the next generation are called children or offspring. There are two genetic operators which generate new children from the existing solutions in our problem. The genetic operators are as follows:

- **Crossover**: it takes two parents and generates a new child by taking parts from parents. In our specific problem, the constraint for a new child is the uniqueness of each node in the generated offspring. There are several crossover operators in the literature for mTSP which consider this constraint [105]. In this work, the cyclic crossover [106], [107], [108], [109] is applied which is fast and guarantees that all nodes are unique in the generated offspring.

  The motivation of the crossover is to take good genes from selected parents to generate better children.

  The cyclic crossover is shown in Figure 6.3. It starts the operation from one of the parents and moves back and forth between parents to find a cycle. The cycle is a set of genes that starts from one node in the first parent and ends in the same node in the second parent. Then, genes from the first parent which are in a cycle will be taken for the new offspring and the rest of the genes are taken from the second parent. The head and tail of the chromosomes are excluded from the crossover operation since they are always the same for all chromosomes and must be the same in the next generation.

  In the opposite way, the genes from the first parent which are not in the cycle are taken and the genes from the second parent which are in the cycle are taken for another offspring.

  The crossover operator generates new solutions for the path planning problem which can be different from their parents in terms of clustering of nodes and their scheduling inside the clusters.

  Figure 6.4 shows the network of the parents and the generated child from the crossover operation of Figure 6.3.

- **Mutation**: is another genetic operator which aims to search locally in the solution space for a better solution. The mutation operator takes an offspring and

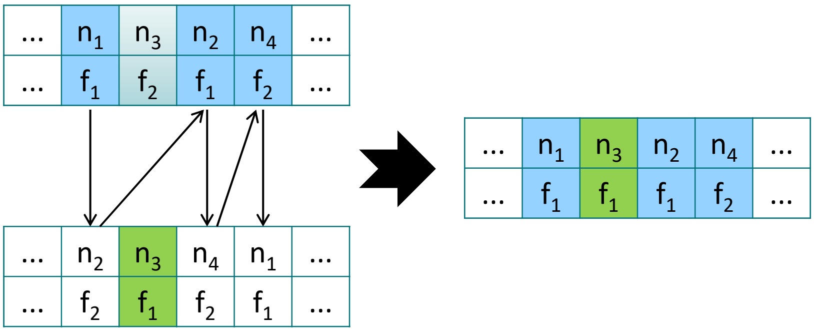


Figure 6.3: A crossover operation on two parent chromosomes to generate a new child (offspring)

searches for solutions in the vicinity of the given solution. In this work, a random swap mutation scheme [110], [111], [112] is applied which randomly selects genes and swaps them in a chromosome. The outcome of random swap mutation is a change in the scheduling of nodes inside a cluster. The mutation operation does not change the clustering of nodes. Same as the crossover, the head and tail of a chromosome are excluded from the mutation operation.

The number of swaps for a given chromosome to generate a new offspring is a controllable parameter in the mutation operation and is shown by α. The α parameter is calculated as follows:

$$\alpha = \frac{2 * |gene\_swaps|}{len(chromosome)} \tag{6.3}$$

where $len()$ function returns the length of a chromosome and $|gene\_swaps|$ is the number of gene swaps in a mutation operation. While there are always 2 genes in a swap, the number of swaps are multiplied by 2. The α parameter defines how far from the given solution, the mutation operation must search.

Figure 6.5 shows the mutation operation on a given chromosome and its corresponding network. It changes only the order of nodes inside one of the clusters.

6.1.5 Termination

The genetic algorithm is an iterative algorithm which starts with an initial generation consisting a set of individuals and evolves during iterations to the optimal solution using the genetic operators and the fitness calculation. The search procedure must be

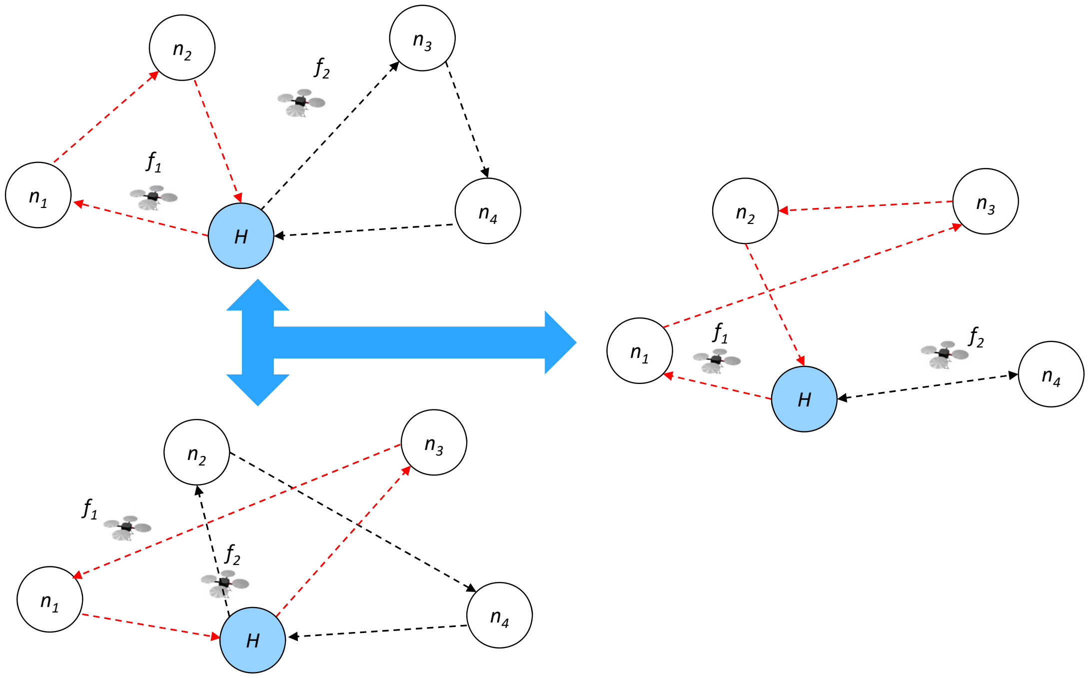


Figure 6.4: Illustration of a crossover operation on two parent chromosomes to generate a new child (offspring)

terminated by finding the optimal solution. As the optimal solution in problems with huge solution spaces is not known, the iterations of the algorithm can be terminated when the algorithm provides a reasonable solution with respect to the requirements of the problem or at a deterministic number of iterations. As an example, the requirement can be the total traveled distance for ferries in the path planning problem. When the genetic algorithm finds a solution which satisfies the requirements of the problem, the algorithm can be terminated. Other than, the algorithm can iterate to a specific number of generations. In this work, a maximum number of generations is assumed for termination of the algorithm. Moreover, it is terminated when the fitness value of the solutions is saturated. The saturation occurs when the improvement in the quality of new solutions is below a defined threshold. This approach helps to accelerate the execution of the algorithm.

$$Threshold \geq F(s_{i-1}) - F(s_i) \tag{6.4}$$

where i is the index for a generation, F is the fitness function and s is a solution for the given problem.

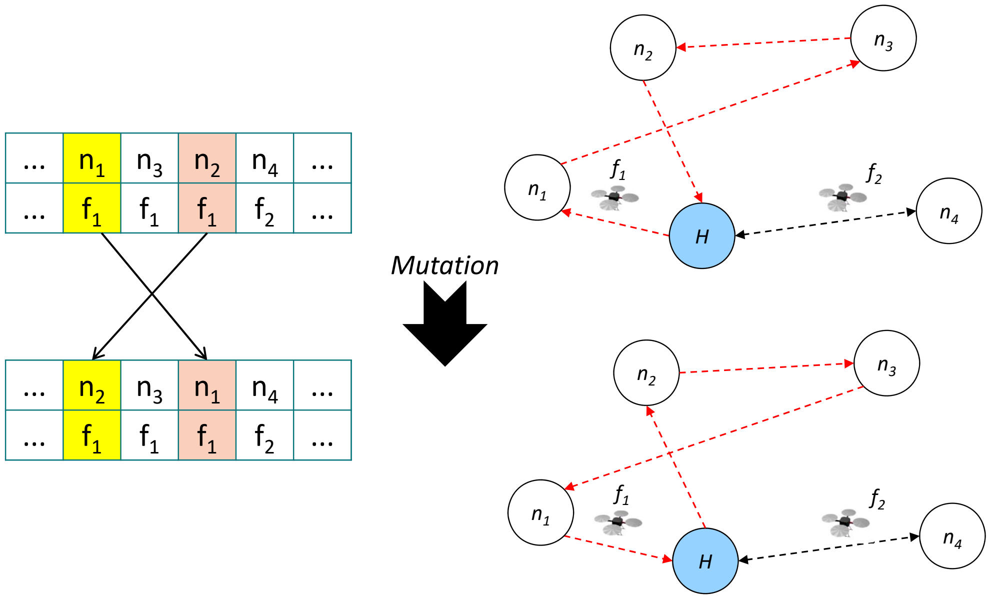


Figure 6.5: Mutation operation on a new child to generate a better solution

6.2 Network architectures

Different network architectures can be employed in a message ferry network when the path of ferries are planned. The architectures are different in terms of message delivery between isolated nodes and the role nodes and ferries in this procedure. In this section, 4 different architectures are introduced and studied where the path of ferries are planned in advance. To plan the path of ferries, the problem of path planning is modeled as either TSP or mTSP. In both cases, the objective is to minimize the traveled distance of ferries. Based on the general assumptions in this work, no information about the message traffic is available for the path planner algorithm. However, the simulator considers the traffic information to calculate the arithmetic mean of message delivery delay to compare the architectures. The following sections are the description of architectures.

6.2.1 Single route for multiple ferries (SRMF)

To deploy the SRMF architecture, the problem of path planning is modeled as TSP. The genetic algorithm finds the shortest path for a ferry to visit all the nodes only once. Therefore, there is no clustering of nodes in this architecture. All nodes are inside a single cluster and the problem is limited to the scheduling of nodes inside the

single cluster.
As multiple ferries are employed in the message ferry network, all ferries will follow the same path which starts from a node (depot) and ends at the same node. To utilize ferries better and avoid useless visits to nodes, ferries start their mission at different times. In this work, ferries start with equal time shifts. Therefore, they are distributed uniformly in a path. A path which is defined for a ferry to visit nodes is called a ferry tour in this work.
The genetic algorithm which solves the TSP, finds the shortest path to visit all nodes without considering message flows while there is no knowledge about them in the path planner.
In this architecture, messages are delivered to their destination traveling in a single ferry. The ferry, which collects a message, delivers it to its destination. There is no explicit cooperation of ferries in terms of message forwarding. However, they cooperate implicitly while serving the same network.
Figure 6.6 shows the SRMF architecture in a network with 5 nodes and 2 ferries. The schedule of nodes in the planned path is n_1, n_2, n_3, n_4, n_5. As an example, the message m_{23}, which must be delivered from n_2 to n_3, travels in one of the ferries directly between two nodes. On the other hand, the ferry which collects the message m_{32} visits n_4, n_5, n_1 and finally delivers the message in n_2.
To calculate the average delay of messages in a network with SRMF architecture, the end-to-end delay is divided into the average wait time of messages in their source nodes and the average travel time of messages in the ferries buffer.

$$avg\_delay = T_{avg\_wait} + T_{avg\_travel} \tag{6.5}$$

To calculate the average wait time of messages, the best and worst cases for the wait time are considered. In the best case, a message is collected by a ferry immediately after its generation. In this case, there is no wait time, $T_{wait} = 0$, for the message in its source node.
In the worst case, a message is generated immediately after a ferry visit to the source node. In this case, the message has no chance to be collected by the ferry and must wait in the buffer of the source node for the next ferry. The worst case for the wait time is calculated as follows:

$$T_{wait} = \frac{len(tour)}{v * f} \tag{6.6}$$

where the $len(tour)$ is the length of the planned path for a ferry, which is the length

of the path for the TSP, v is the velocity of ferries, which is same in all ferries and f is the number of ferries in the network. By increasing the number of ferries in a network, the wait time of messages in SRMF will tend to zero.
Therefore, the average wait time in a network is as follows:

$$T_{avg\_wait} = \frac{max(T_{wait}) + min(T_{wait})}{2} \tag{6.7}$$

The average travel time depends on the planned path for ferries and the message flows between nodes.

$$T_{avg\_travel} = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} t_{ij} \tag{6.8}$$

where t_{ij} is the time to travel from node i to j by a ferry following the planned path and is calculated as follows:

$$t_{ij} = \frac{d_{path}(i,j)}{v} \tag{6.9}$$

where $d_{path}(i,j)$ is the distance between node i and j inside the planned path and v is the velocity of ferries.
The w_{ij} is the weight for the message flow between node i and j. It is calculated as follows:

$$w_{ij} = \frac{|m_{ij}|}{\sum_{x=1}^{N}\sum_{y=1}^{N}|m_{xy}|} \tag{6.10}$$

where $|m_{ij}|$ is the number of messages from node i to j and N is the set of nodes.

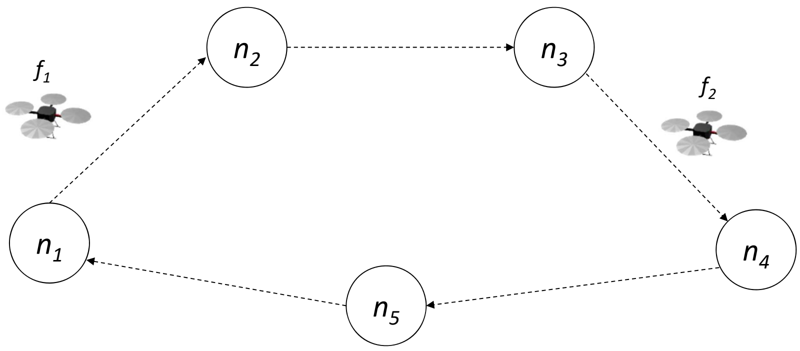


Figure 6.6: Illustration of SRMF architecture

6.2.2 Multiple routes for multiple ferries (MRMF)

To plan the path for multiple ferries, the problem is modeled by mTSP and a solution is found applying the genetic algorithm. The genetic algorithm divides nodes into clusters and assigns a ferry to each cluster. Moreover, it defines the optimal scheduling of nodes in a cluster. The Multiple Route for Multiple Ferries (MRMF) architecture applies the solution for mTSP. A message is delivered to its destination traveling in a single ferry if the source and destination of the message are in the same cluster. In case which the source and destination of a message are in different clusters, the message is delivered to its destination traveling in two ferries and through a hub node. In MRMF, all ferries start their travel from a node (depot), visit a subset of nodes and return to the same node. This node belongs to all clusters and is visited by all ferries. It is called the hub node in MRMF.

In case of an inter-cluster message, the message is delivered first to the hub node. Then, the ferry in the destination cluster collects the message from the hub node and delivers it to the destination node.

The hub node can be a special node or any of the nodes in the network which is considered as the starting point of all ferries by the path planner.

Figure 6.7 illustrates the MRMF architecture where all ferries start their tour from node n_5. n_5 is the hub node in this network and acts as a relay between clusters. For example, the message m_{13} from node n_1 to n_3 is collected by f_1 and is delivered first to node n_5. Then, the message must wait in n_5 to be collected by f_2. After the collection by f_2, the message travels in f_2 the path to its destination node which is n_3. Therefore, the traveled path of a message in the buffer of ferries is n_1, n_2, n_5, n_3.

To calculate the average delay of messages in MRMF, the constituent elements in the end-to-end of messages are calculated.

In case of inter-cluster messages in MRMF, there are two wait time for a message. The first wait time may occur in the source node and the second wait time may occur in the hub node.

$$T_{wait} = T_{wait\_source} + T_{wait\_hub} \tag{6.11}$$

The wait time in the source has a best case and a worst case. To calculates the average wait time in the source node, both cases are summed up and divided by two. In the best case, same as SRMF, there can be no wait time, $T_{wait\_source} = 0$, at source and a message is collected by a ferry as soon as it is generated. In the worst case, the message misses the ferry and must wait until the next visit of the ferry. A ferry visits all nodes in a cluster in a cyclic manner to exchange messages between them. The worst case for wait time in the source node in MRMF architecture is calculated as follows:

$$T_{wait\_source} = \frac{len(tour_f)}{v} \tag{6.12}$$

where is $len(tour_f)$ is the length of the ferry tour and v is the velocity of the ferry. The wait time in the hub node depends on the difference between the length of ferry tours. Moreover, the T_{wait_hub} for intra-cluster messages is zero.
The travel time of a message from node i to node j also depends on the location of j and is calculated as follows:

$$t_{ij} = \begin{cases} \frac{d_{path}(i,j)}{v} & i, j \in C_n \\ \frac{d_{path}(i,hub)+d_{path}(hub,j)}{v} & else. \end{cases} \tag{6.13}$$

where the $i, j \in C_n$ is the condition to find intra-cluster messages, $d_{path}(i, hub)$ is the distance between the source and the hub following the planned path and $d_{path}(hub, j)$ is the distance between the hub and destination node following the planned path.
To calculate the average travel time, the message flows between nodes should be considered same as Equation 6.8.

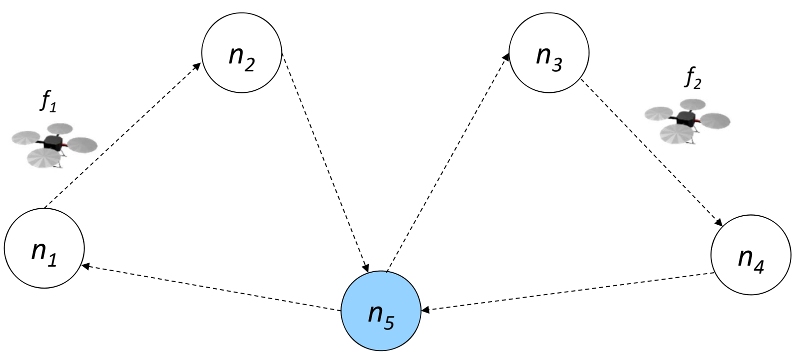


Figure 6.7: Illustration of MRMF architecture with a hub node for inter-cluster communication

6.2.3 MRMF with a rendezvous (MRMF-R)

It is similar to the MRMF where the path of ferries is planned applying the solution for mTSP. The difference is in the exchange of messages between clusters. In Multiple Routes for Multiple ferries with Rendezvous (MRMF-R), the inter-cluster messages are exchanged directly between ferries and the hub node does not act as a relay between

clusters. As a short range communication in ferries is assumed, ferries should visit each other at a predefined location to exchange their messages. The rendezvous location is defined by the path planner. Based on mTSP solution, one of the nodes is visited by all ferries. The rendezvous location can be at the same location as the location of the common node between clusters. Moreover, each ferry should visit all other ferries in a rendezvous. A ferry which arrives earlier than others to the rendezvous point must wait until the arrival of the last ferry. When all ferries are in the rendezvous location, they exchange their messages and start a new tour at the same time.
Figure 6.8 shows a MRMF-R architecture where ferries make a rendezvous at the location of n_5. As an example, the message m_{13} travels the path n_1, n_2 in f_1 and then is forwarded to f_2 at the rendezvous and continues its travel in f_2 to n_3.
To calculate the delay of messages in the MRMF-R architecture, Equation 6.5 is valid for the average end-to-end delay in the network. The wait time of messages in MRMF-R is calculated as follows:

$$T_{wait} = T_{wait\_source} + T_{wait\_rend} \tag{6.14}$$

where T_{wait_rend} is 0 for intera-cluster messages and for the inter-cluster messages is calculated as follows:

$$T_{wait\_rend} = max(len(tour_f)_{f \in F}) - min(len(tour_f)_{f \in F}) \tag{6.15}$$

where f is a ferry and F is the set of ferries. The rendezvous duration depends on the difference between the shortest and longest tour of ferries.
The travel time of message m_{ij} is calculated same as Equation 6.13. Instead of a hub in the traveled path of a message, the rendezvous location exists where can be same as the location of the common node between clusters or any other predefined location.

$$t_{ij} = \begin{cases} \frac{d_{path}(i,j)}{v} & i, j \in C_n \\ \frac{d_{path}(i,rend)+d_{path}(rend,j)}{v} & else. \end{cases} \tag{6.16}$$

6.2.4 MRMF-R with a balanced travel path of ferries (MRMF-R-B)

It is the same architecture as MRMF-R and the only difference is on the fitness function of the genetic algorithm to cluster the nodes and find the path of ferries inside

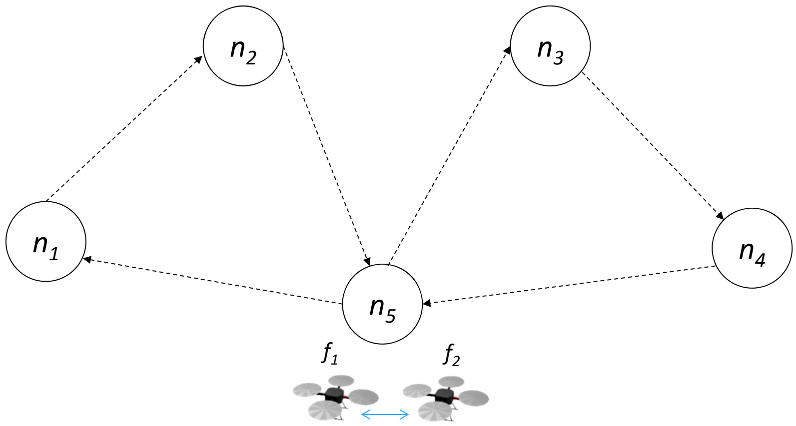


Figure 6.8: Illustration of MRMF-R architecture with a rendezvous point for inter-cluster communication

the clusters. For this network architecture, the genetic algorithm search for a solution which causes the minimum rendezvous time for ferries. To have the minimum rendezvous time, the difference between the minimum and maximum travel path of the ferries must be minimized. For this reason, the genetic algorithm may find a solution in which the total traveled distance of ferries is not minimum but the rendezvous time of ferries is minimum. Except this difference, MRMF-R and MRMF-R-B are same.

6.3 Performance evaluation of network architectures

In this section, the performance of different network architectures is evaluated. All architectures are based on an offline path planning for ferries in a central entity. The assumptions for the nodes and ferries are the same as all other studies in this work. The exception is the existence of an entity which plans the path of ferries in advance. Ferries follow a given path in a cyclic manner. The problem of path planning is modeled as TSP or mTSP. To plan the path of ferries, the genetic algorithm which was explained earlier in this chapter is applied.

It worth emphasizing that **the path planner has no knowledge about the message flows** in the network and only minimizes the total traveled distance of ferries in MRMF, MRMF-R and the rendezvous time in MRMF-R-B. The goal of the genetic algorithm in SRMF is to minimize the traveled distance of a single ferry.

The simulation setup is shown in Table 6.1.

Table 6.1: The simulation setup of studies on the network architecures with an offline path planning of ferries

Parameter	Value
Number of nodes	20
Number of ferries	1-10
Distribution of nodes	Random Uniform
Speed of ferries	5 m/s
Network size	1000 x 1000 meter
Number of runs	10 runs
Size of initial population	100
Initial population distribution	Uniform
Maximum number of generations	100

6.3.1 All to all traffic

In this section, the performance of different network architectures is studied considering the all to all traffic scenario. Figure 6.9 shows the results and the lessons learned from the results are as follows:

- MRMF is the best architecture while it has the shortest average travel time for intra-cluster messages. For inter-cluster messages, the unbalanced travel path of ferries may impose a long wait time of messages in the hub node.
- MRMF-R-B shows good performance as MRMF because of the minimum wait time for the rendezvous when there are less than 5 ferries in the network. In MRMF-R-B, only the wait time of inter-cluster messages is minimized. This may lead to a longer travel time of messages while the path planner does not care about the traveled distance of ferries. With more ferries, the inter-cluster messages are more due to the less populated clusters. The wait time for rendezvous may be less in MRMF-R-B but a message should travel a longer path in ferries of the source and destination clusters.
- MRMF-R is worse than MRMF since rendezvous is not always necessary and may cause a useless wait time of ferries and a longer travel time of messages.
- SRMF is the worst approach since each ferry visits all nodes which is not necessary.

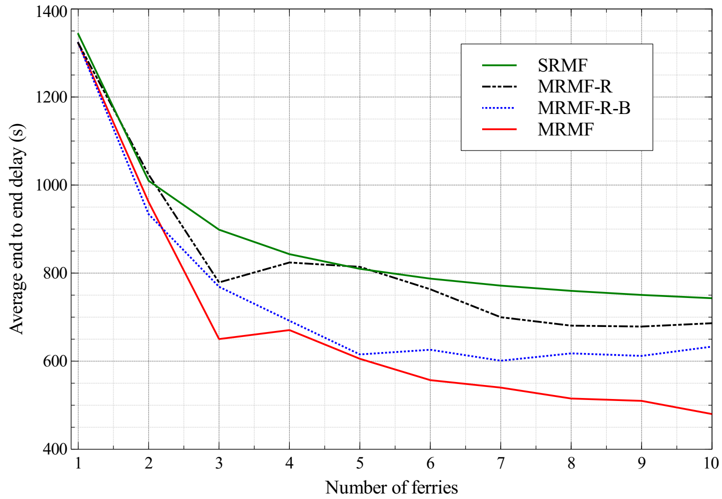


Figure 6.9: Comparison of different network architectures applying the genetic algorithm for offline path planning considering the all to all traffic scenario

6.3.2 All to sink and broadcast traffic

In this section, the performance of different network architectures where the genetic algorithm plans the path of ferries are studied in terms of the average end-to-end of messages considering the all to sink and broadcast traffic scenarios. The performance of each network architectures is the same in the all to sink and broadcast traffic scenarios while the only difference between the traffic scenarios is the direction of message flows. Figure 6.10 shows the results for both traffic scenarios and the lessons learned from the results are as follows:

- SRMF has the worst performance due to the long travel time of messages. In this architecture, each ferry visits all nodes which is unnecessary. All nodes follow the same path since the genetic algorithm plans the path of a single ferry and all other ferries must follow the same path with a time shift.
- MRMF-R and MRMF-R-B impose extra wait time for unnecessary rendezvous. The wait time for rendezvous is less in MRMF-R-B due to the balanced travel path of ferries in clusters.
- MRMF has the best performance since messages only experience a minimum travel time and the path planner minimizes the traveled distance of ferries in clusters. This leads to the minimum average travel time of messages.

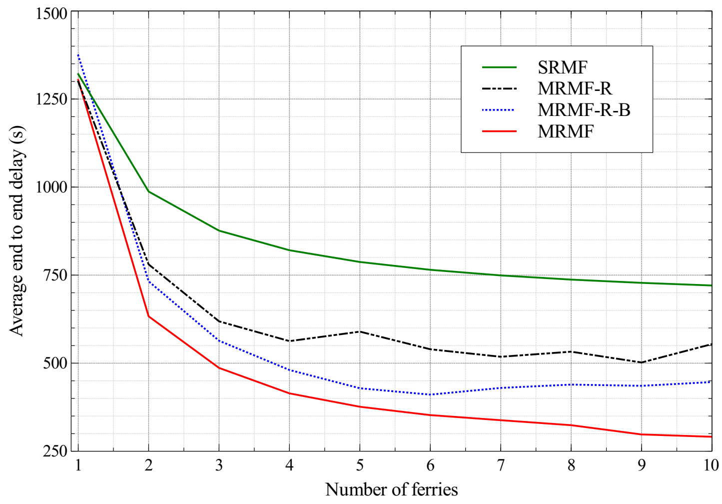


Figure 6.10: Comparison of different network architectures applying the genetic algorithm for offline path planning considering the all to sink/broadcast traffic scenarios

7 Comparison of Different Strategies to Control the Mobility of Ferries

In this chapter, the on-the-fly mobility decision making and offline path planning are compared as the main strategies to control the mobility of ferries in terms of the end-to-end delay of messages, robustness, initialization time and flexibility.

7.1 Average end-to-end delay of messages

In this section, the average end-end delay of messages applying different strategies to control the mobility of ferries, which were studied in this thesis, are compared. For comparison, the following strategies are considered:

- On-the-fly decision making: two different cases based on the on-the-fly mobility decision making in ferries are considered for the comparison as follows:
  - Non cooperative ferries: in this case, ferries decide their mobility on-the-fly and are coordinated applying an indirect signaling through nodes. There is no message exchange between ferries in terms of message forwarding or replication. A message is collected by a ferry and delivered by the same ferry to its destination. This approach was proposed and studied in Chapter 4. It is called ”On-the-fly next-hop Decision maker for Multi-Ferry networks” (ODMF) in this study.
  - Cooperative ferries: ferries apply not only the on-the-fly mobility decision making and the indirect signaling, but also have explicit cooperation. They exchange messages in the form of message forwarding or replication. Therefore, a multi-hop delivery of a message may occur in this case. ODMF-TAF-PR and ODMF-TAR represent this case. Applying ODMF-TAF-PR, ferries exchange messages directly when they meet each other and apply the trajectory aware message forwarding (TAF) and the pure replication (PR) to make decision on the message forwarding or replication. Applying

Table 7.1: The simulation setup of studies on different strategies to control the mobility of ferries

Parameter	Value
Number of nodes	20
Number of ferries	10
Distribution of nodes	Random Uniform
Speed of ferries	5 m/s
Network size	1000 x 1000 meter
Number of runs	10 runs

ODMF-TAR, ferries select nodes as relays to leave replication of messages in them. To select the relay nodes, the trajectory aware messages replication (TAR) scheme is applied. ODMF-TAF-PR and ODMF-TAR schemes were proposed and studied in Chapter 5.

– Offline path planning: mTSP is applied to model the path planning problem for multiple ferries. The genetic algorithm is applied in a path planner to minimize the total traveled distance of ferries. Finally, the solution of the path planning problem is applied in the Multiple Routes for Multiple Ferries (MRMF) architecture. MRMF is chosen since it had the best performance within the network architectures based on the studies in Chapter 6. The path of each ferry is planned in this strategy and all ferries follow the given path.

It worth to mention that neither offline path planning, nor the on-the-fly approaches have no knowledge about the traffic scenario and message flows in the network. The only available information is the location of nodes. The traffic scenarios for this study are the all to all, all to sink and broadcast scenarios, which were explained in Chapter 3. The simulations setup is shown in Table 7.1. The parameters of the genetic algorithm such as initialization, number of generations, and termination of the algorithm is same as Chapter 6.

7.1.1 All to all traffic

In this section, the all to all traffic scenario is considered and all approaches are compared in terms of the average end-to-end delay of messages. The results are shown in Figure 7.1 and the lessons learned from the results are as follows:

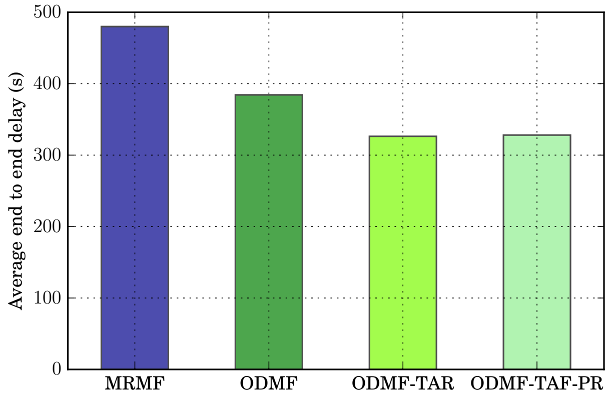


Figure 7.1: Comparison of different strategies to control mobility of ferries considering the all to all traffic scenario

- The on-the-fly mobility decision maker, ODMF, outperforms the offline approach. Applying MRMF, the source, and destination of a message can be in the same cluster or in different clusters. In the case of intra-cluster messages, they should travel the path of a ferry inside the cluster. In the case of inter-cluster messages, they experience a wait time in the hub node, which is the intersection of all clusters. Applying ODMF, there is no wait time in any node except the source node. Moreover, ferries consider the state of their buffer in their mobility decision maker. This increases the chance of faster delivery of messages.
- By message replication in nodes in ODMF-TAR and message forwarding and replication between ferries in ODMF-TAF-PR, messages are delivered to their destinations not only by a single ferry but multiple ferries. This emerges the cooperation of ferries and reduces the end-to-end delay of messages.
- By message replication in nodes applying ODMF-TAR, the same performance is achieved as ODMF-TAF-PR, which needs the meeting of ferries.
- The genetic algorithm minimizes the total traveled distance of ferries which is not helpful considering the all to all traffic scenario.

7.1.2 All to sink traffic

In this section, the all to sink traffic scenario is considered and all approaches are compared in terms of the average end-to-end delay of messages. The results are shown in Figure 7.2 and the lessons learned from the results are as follows:

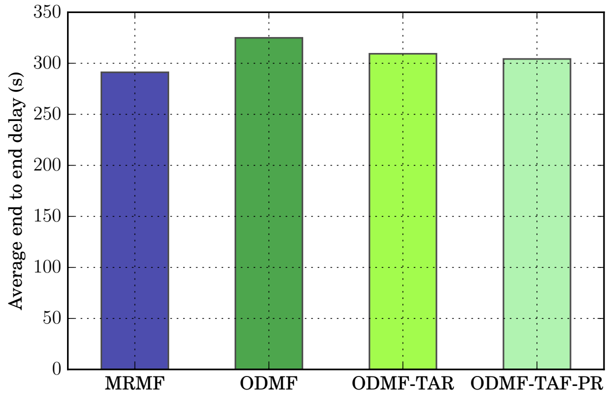


Figure 7.2: Comparison of different strategies to control the mobility of ferries considering the all to sink traffic scenario

- In this traffic model, the offline approach outperforms the on-the-fly strategies. In MRMF, nodes are clustered and paths of ferries are optimized to provide the minimum traveled distance of ferries. While all nodes generate messages to the sink node, clustering of nodes and finding the shortest path to visit them provides the minimum average travel time of messages. For this reason, the offline approach is the best approach for this traffic model.

- The on-the-fly approaches, even with the cooperation of ferries, lack global knowledge. For this reason, the distribution of ferries (resources) is done imperfectly in the network and there can be lots of redundancies in ferries visits to nodes.

- The path planner of the offline approach should know the location of the sink node. Other than, the path planner cannot cluster the nodes optimally and the average end-to-end delay of messages will not be minimum.

- Considering the all to sink traffic, the improvement in the average end-to-end delay of messages is achieved with the cost of time for path planning. Applying the on-the-fly mobility decision maker, there is no initialization time since nothing is planned in advance.

- To have the best performance in MRMF, the path planner should know about the sink node. Other than, the results will be worse than the existing results for MRMF. Comparing MRMF with ODMF, the later one does not need to know about the sink node or any other message flow in the network.

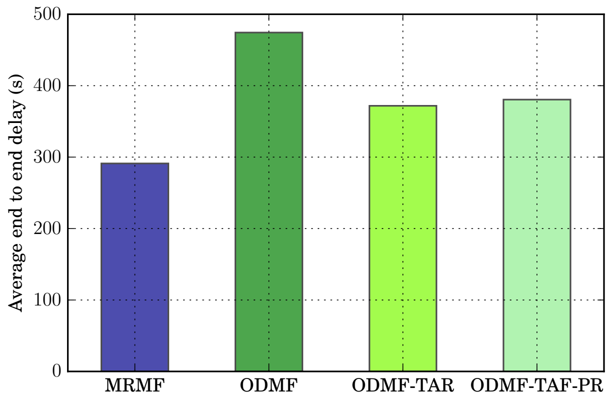


Figure 7.3: Comparison of different strategies to control the mobility of ferries considering the broadcast traffic scenario

7.1.3 Broadcast traffic

In this section, the broadcast traffic scenario is considered and all approaches are compared in terms of the average end-to-end delay of messages. The results are shown in Figure 7.3 and the lessons learned from the results are as follows:

- The results for MRMF is the same as the all to sink traffic scenario since the only difference is the direction of message flows.
- Offline path planning in MRMF and the shortest path to visit all nodes provides the minimum end-to-end delay.
- Same as the all to sink traffic scenario, the path planner should know the location of the broadcaster node to consider it as the intersection of all clusters. Other than, the results will be worse than the existing results for MRMF.
- Another assumption in MRMF is about the message collection from the broadcaster node. It is assumed that each ferry collects only messages for the cluster where it visits. Extra travel time and wait time will be added to all messages if a ferry collects all messages from the broadcaster node.
- The difference between offline and on-the-fly approaches is bigger in this traffic scenario because the message collection in MRMF is smarter than ODMF and its variants where a ferry collects all messages from the broadcaster node and messages experience a long travel time.

- Even with cooperation of ferries in ODMF-TAR and ODMF-TAF-PR, the performance is not as good as MRMF since the distribution of the message delivery task is done optimally in MRMF by optimal clustering of nodes.

7.2 Robustness

With respect to the robustness of networks applying offline or on-the-fly strategies, the static and dynamic nature of the strategies and their capability for the toleration of failures should be studied. The following points compare the robustness of strategies:

- In the offline strategy, a single ferry is assigned to each cluster. By the failure of the ferry, all nodes inside the cluster will suffer visit starvation and the wait time of messages will go to infinity.
- The only robust architecture against failures of ferries, applying the offline path planning, is SRMF where a single cluster exists in the network and all ferries visit the same cluster. If a ferry fails in SRMF, the wait time of messages will increase but communication does not fail in the network.
- The on-the-fly strategy is a self-organized approach which is not dependent on a single ferry. By failure of a ferry, there will be no visit starvation in nodes. The delay of messages may increase due to the less number of ferries in the network but communication is still possible.

7.3 Initialization time

Initialization time is also different in offline and on-the-fly strategies. The offline path planner needs time to solve the path planning problem. This is considered as the initialization time. To find the exact solution for the path planning, an initialization time in the scale of several seconds to several hours may exist for the offline strategy. The initialization time of the offline strategy increases by increasing the number of nodes or ferries in a network. Table 7.2 shows the execution time of the genetic algorithm for a network with 10 nodes and different number of ferries. By increasing the number of ferries, the execution time increases linearly. On the other hand, the on-the-fly strategy do not need any initialization since nothing is planned in advance and ferries make all decisions about their mobility or message forwarding (or replication) on-the-fly.

Table 7.2: Execution time of the offline planner

Number of ferries	Execution time (seconds)
2	1,9
3	2,12
4	2,33
5	2,71
6	3,03
7	3,26

However, the initialization time in the scale of several seconds for small networks worth to have a better average end-to-end delay in the all to sink and broadcast scenarios applying the offline strategy.

7.4 Flexibility

The flexibility of a strategy depends on its reactions to the dynamics of a network. Dynamics of a message ferry network can be changes in the location of nodes, changes in the number of ferries or nodes and changes in the message traffic.

In case of any change in the location of nodes, the offline approach must run the path planner algorithm again to find paths of ferries. In the proposed on-the-fly mobility decision maker, isolated nodes are assumed to be stationary. Assuming mobile isolated node, the on-the-fly mobility decision making would be a better choice than running periodically the path planner algorithm. An enhanced on-the-fly approach can track the mobility of nodes and predict their future locations.

A change in the number of ferries does not have any impact on ferries which apply the on-the-fly strategy. Ferries can continue to serve the nodes as before. However, the average message delivery delay may change. In the offline strategy, mTSP based architectures where nodes are clustered, the path planner must run again to find a new solution with an increase or decrease in the number of ferries. Only the TSP based architecture, which is SRMF and all ferries follow the same cluster, is flexible and does not need to find the new solution for the path planning problem.

The change in the number of nodes makes the path of ferries invalid applying offline path planning. In all network architectures based on the offline strategy, the path planner must find a new solution. Applying the on-the-fly strategy, ferries can continue their mission without a need for any adaptation by removing a node from the network.

By adding a new node, the location information of the new node must be informed to one of the ferries. Then, this information is received by other ferries through indirect signaling of ferries.

The on-the-fly strategy is completely flexible with respect to the changes in the message traffic since it applies the state of a ferry buffer in its decision. Any change in the traffic flows will impact the decisions of ferries and ferries adapt themselves to the new circumstances. The offline strategy does not react to any change in the message traffic in the network because minimizing the delay of messages is not the goal of its optimization. Any change in the message traffic may cause longer message delivery delays in offline approaches.

8 Conclusion

In this chapter, the thesis is summarized and the most important lessons learned are mentioned. Then, the possible improvements to the current work are explained as future work.

8.1 Summary of the thesis and lessons learned

In this thesis, a message ferry network with multiple ferries was studied where the mobility of ferries is controlled to improve the performance of message delivery. First, the following strategies to control the mobility of ferries were considered to be studied:

1. On-the-fly mobility decision making: a ferry decides about its mobility within its travel based on its local observations. There is no central entity to control the network and a ferry does not have a global knowledge about other ferries and nodes. The only global knowledge in ferries is about the location of stationary nodes.
2. Offline path planning: a path planner decides about the path of ferries in advance and ferries follow the given path.

Then, the strategies were compared. All studies and comparisons were done considering different traffic scenarios such as all to all, all to sink and broadcast scenarios.
To study the on-the-fly decision making strategy, an on-the-fly mobility decision maker for multi-ferry networks, ODMF, was proposed which considers the state of a ferry buffer, a visit history and the distance to nodes in its decision. Ferries apply an indirect signaling through nodes to update their visit history and avoid redundant visits to nodes. The studies on the on-the-fly decision making can be classified into two main classes:

1. Message ferry networks where ferries do not exchange messages: a message is collected by a ferry and is delivered to its destination by the same ferry.

2. Message ferry networks where ferries exchange messages: a message may travel in multiple ferries to be delivered to its destination. The message exchange between ferries can be done as following:
   - Direct message exchange: ferries forward or replicate messages between each other.
   - Indirect message exchange: ferries leave replication of messages in nodes. Nodes act as relays to exchange the replication of messages between ferries.

In the beginning, the first class was considered to study the performance of ODMF. The study was done to find the impact of different metrics on the on-the-fly decision maker. For this reason, different decision makers were compared to ODMF. The following lessons are learned from studies:

- The indirect signaling between ferries which was proposed to avoid timely close visits to nodes by different ferries is beneficial for the all to all and all to sink traffic scenarios but has no benefit in the broadcast scenario.
- The indirect signaling of ferries can avoid useless visits to nodes which have been visited by other ferries. In this case, a ferry can serve messages in its buffer to reduce their travel time or visit other nodes to reduce the wait time of messages in them. In the broadcast scenario, a ferry has messages to all nodes in its buffer. Therefore, the ferry should visit all nodes to deliver their messages. Avoiding to visit a node due to the indirect signaling of ferries reduces the travel time for some of the messages but increases the travel time for others. For this reason, the average travel time of messages does not change.
- The distance metric in the decision function of ferries improves the efficiency of message delivery and reduces the travel time of messages in the all to all and broadcast scenarios. Employing the distance metric, ferries deliver messages to close nodes first and improve the travel time of messages.
- There is no benefit to use the distance metric in the all to sink scenario while all messages should be delivered to the sink and there is no message to be delivered to any close node.
- The last visit time history is a better choice than the number of visits to nodes. Both of them are applied to serve messages in nodes but the number of visits to nodes performs poorly in the all to sink scenario. It causes the avoidance of the

sink node by ferries in the all to sink scenario and increases the travel time of messages.

- Considering the number of visits to nodes in the on-the-fly decision maker performs also poorly when there are close nodes in a network which generate messages for each other. The number of visits to such nodes increases rapidly. For this reason, a ferry avoids the nodes for some time to balance the number of visits to nodes. This increases the travel time and wait time of messages.
- There is no benefit to keep a history of collected messages from nodes to find their message generation rate. While multiple ferries serve the same network, the collected history is invalid or becomes invalid after a short time.

To find the impact of the number of ferries on the end-to-end delay, travel time and wait time of messages having different metrics in the decision function of ferries, the number of ferries was increased from 1 to 20 ferries. The following lessons are learned from this study:

- Increasing the number of ferries reduces the average wait time of messages in all considered traffic scenarios. The reason is the more often visits to nodes by more ferries.
- The impact of the increase in the number of ferries on the average travel time of ferries depends on the traffic scenario. In the broadcast scenario, it has no impact while a single ferry delivers messages to all nodes. It also depends on the mobility decision function of ferries. In the all to sink scenario, having more ferries do not reduce the travel time if a ferry applies only the state of its buffer. In this case, the ferry travels directly from the source of a message to the sink node and the increase in the number of ferries cannot reduce the travel time of messages since it is the minimum travel time.
- When there are lots of ferries in a network, the wait time of messages is only a small fraction of the end-to-end delay. Therefore, applying the visit history of nodes is not necessary anymore to serve waiting messages in nodes. The reason is more often visits to nodes employing more ferries which reduces the wait time of messages.

To achieve a better performance of message delivery in message ferry networks, two options were compared. The options and the lessons learned from the results are as follows:

1. Increasing the speed of ferries: it is a better choice than increasing the number of ferries in terms of the end-to-end delay of messages. Increasing the speed of ferries impacts on the travel and wait time of messages. The travel time is reduced while ferries travel faster and travel time between nodes is shorter. The wait time is also reduced while nodes are visited more often. This option imposes also less cost in terms of the total traveled distance of ferries to achieve a performance goal.

2. Increasing the number of ferries: it has less benefit than increasing the speed of ferries. The wait time of messages is reduced with more ferries but there are always redundancies in visits to nodes and the redundancies are increased having more ferries. Redundancies decrease the efficiency of having more ferries. The travel time is not always reduced employing more ferries and depends on the traffic scenario. In the broadcast scenario, there is no reduction in the travel time employing more ferries. In the all to all and all to sink scenarios, there is a slight improvement (applying ODMF) in the travel time of messages due to the fewer messages in a ferry buffer and the indirect signaling of ferries.

In the next step, message ferry networks where ferries exchange messages were studied. To do this, two message exchange approaches were proposed and investigated. In the first approach, ferries forward or replicate messages between each other when they meet. Different schemes for message forwarding and replication between ferries and the impact of the mobility decision maker in ferries on them were studied. In the second approach, an indirect message exchange between ferries through nodes occurs. In this case, ferries do not need to meet each other to exchange messages. They leave the replication of messages in nodes. Different strategies to select a relay node for a message and the impact of the mobility decision maker in ferries on them were studied. The lessons learned from the studies are as follows:

- The impact of a message forwarding scheme (between ferries) depends on the mobility decision maker in ferries and the traffic scenario. Therefore, message forwarding between ferries is not always helpful.

- Message forwarding between ferries has no benefit in the all to sink traffic scenario if the mobility decision maker in ferries serves only messages in the ferries buffer (eg. FB, FBD).

- The message forwarding between ferries shows a major benefit in the broadcast traffic scenario since the travel time of messages is long and any message forwarding improves the travel time of messages.

- Message replication between ferries slightly improves the end-to-end delay of messages in all traffic scenarios. This shows that the proposed message forwarding scheme is an efficient approach and there is no need for a message replication between ferries. Even flooding of replicas does not show a significant improvement in the performance.
- The differentiation of original and replicated messages is necessary when there are multiple instances of a message in a network. Only original messages must be taken into account for the mobility decision making in ferries. Other than, there are redundancies in a message delivery and the redundancies degrade the performance of message delivery in the network. Only in the all to sink traffic scenario, the redundancies do not show any adverse impact since all messages should be delivered to a sink node.
- If the mobility decision maker serves only messages in the ferry buffer (FB, FBD), leaving a replication of messages in relay nodes has a slight and no benefit in the all to all and all to sink traffic scenarios, respectively.
- Message replication in nodes is always helpful when the traffic scenario is the broadcast scenario independent from the mobility decision maker in ferries.
- Message replication in nodes for indirect message exchange between ferries achieves similar performance as the direct messages exchange between ferries without any need for the meeting of ferries in all traffic scenarios.
- Leaving a replication of a message in a node, which is close to the destination of the message, is useless when the distance metric is considered in the mobility decision maker of ferries having the all to sink traffic scenario.
- The more replications in a network, the better performance is achieved. However, considering trajectory information of ferries for the selection of nodes as relays reduces the number of replicas about 50% comparing with the flooding of replicas but it achieves similar performance to flooding.

The offline path planning of ferries was also studied as another strategy to control the mobility of ferries. Different network architectures for message ferry networks were studied considering different traffic scenarios. The lessons learned from the studies are as follows:

- The solution of mTSP always achieves a better performance than the solution of TSP for path planning of multiple ferries.
- Using a node as a relay (hub node) to exchange messages between clusters of nodes is a better choice than having a rendezvous of ferries to exchange inter-cluster messages between ferries.
- Balancing the size of clusters in the path planning process minimizes the wait time of messages due to the rendezvous. However, the average travel time of messages is increased since the objective of the path planner is the minimization of the rendezvous time and not the traveled distance of ferries.

Finally, the main strategies to control the mobility of ferries were compared considering the all to all, all to sink and broadcast scenarios. The lessons learned from the comparisons are as follows:

- On-the-fly mobility decision making is a better choice for the all to all traffic scenario since there is only a wait time for messages in their source node. On the other hand, messages may experience extra wait time in the hub node applying the offline strategy. The offline strategy also does not consider the traffic flows in path planning. Whereas, the on-the-fly strategy considers the ferry buffer and tries to adapt the mobility based on the local observations from the buffer of a ferry. Moreover, the cooperation of ferries reduces the travel time of messages applying the on-the-fly strategy.
- Offline path planning is a better choice when there is an all to sink or broadcast traffic scenario since the average travel time of ferries is the minimum. The minimum travel time of ferries leads to minimum average travel and wait time of messages in these traffic scenarios.
- However, to have better performance applying the offline strategy in the all to sink and broadcast traffic scenarios, the path planner should have knowledge about the location of the sink and broadcaster nodes.

8.2 Future work

In this thesis, two main strategies to control the mobility of multiple ferries were studied. For each of the strategies, an improvement is considered as future work.

8.2.1 Adaptive on-the-fly mobility decision maker in ferries

In the proposed mobility decision function in Chapter 4, the state of a ferry buffer, the visit history and the distance to the next node are considered and a normalized value is calculated for each of them. All the normalized values have the same weight in the mobility decision function. To achieve a better performance in a message ferry network, the following weighted function can be applied in ferries:

$$Score(r) = \frac{w_1 * fb(r) + w_2 * lvt(r)}{w_3 * distance(c, r)} \tag{8.1}$$

where w_1, w_2, w_3 are the weights for the influential values in the mobility decision function and can be adapted based on the state of a network instead of having the same value for all them. As an example, based on our studies on the number of ferries in a message ferry network, the last visit time history is not a necessary metric in the mobility decision function when there are many ferries in a network. with a high number of ferries, the wait time of messages tends to zero and there is no need to consider the waiting messages in nodes for the mobility decision making. In such circumstances, the value for the state of a ferry buffer must have a bigger weight than the value for the last visit time history. With a bigger weight for the state of the ferry buffer, messages in the ferry buffer are served more and their travel time is reduced. To define the best weight for each metric in the decision function, ferries should apply their local observations. For this case, the ratio between the average wait time of messages and their end-to-end delay can be observed by ferries to find out the importance of each metric in the decision function. Base on this observation, ferries can adapt the weights.

8.2.2 Adaptive path planning

In this thesis, an offline path planning for multiple ferries based on mTSP solution was studied where the total traveled distance of ferries was minimized while the path planner had no knowledge about the message traffic. The planned paths were static and not optimized to provide the minimum delay of messages. Based on our studies, the best approach to plan the path of ferries was to cluster the nodes and find the shortest path for each ferry in its cluster.

An improvement of the existing approach is to adapt the path of ferries after visiting nodes by ferries. Ferries can observe the traffic and message flows inside their clusters and provide this information to the path planner. The path planner can update the path of ferries inside each cluster to minimize the end-to-end delay of messages. Moreover, the path planner can change the clustering of the nodes to achieve its goal. The adaptation of the planned paths and clustering of nodes can be done after an exploration phase to let the ferries complete their observation from the message flows between nodes. The exploration and path planning can be done periodically to react to the changes in the message traffic in a network.

Bibliography

[1] 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.

[2] A. V. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay tolerant networks: Protocols and applications*. CRC press, 2016.

[3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," tech. rep., RFC 3561, 2003.

[4] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," tech. rep., RFC 3626, 2003.

[5] M. Harounabadi and A. Mitschele-Thiel, "Stigmergic communication for self-organized multi ferry delay tolerant networks," *Mobile Networks and Applications*, 2017.

[6] M. Harounabadi and A. Mitschele-Thiel, "Applying message forwarding and replication to multi-uav message ferry networks," *Mobile Networks and Applications*, 2018.

[7] M. Harounabadi, A. Rubina, and A. Mitschele-Thiel, "Cooperative on-the-fly decision making in mobility-controlled multi ferry delay tolerant networks," in *International Conference on Ad hoc Networks*, 2016.

[8] M. Harounabadi and A. Mitschele-Thiel, "Trajectory and buffer aware message forwarding for multiple cooperating uavs in message ferry networks," in *International Conference on Adhoc Networks*, 2017.

[9] M. Harounabadi, M. Bocksberger, and A. Mitschele-Thiel, "Evolutionary path planning for multiple uavs in message ferry networks applying genetic algorithm," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018.

[10] M. Harounabadi, M. Bocksberger, and A. Mitschele-Thiel, "Study on the network architectures for message ferry networks with multiple uavs," in *IEEE International Conference on Ubiquitous and Future Networks*, 2018.

[11] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, 2005.

[12] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.

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

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

[15] D. Reina, M. Askalani, S. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A survey on multihop ad hoc networks for disaster response scenarios," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 647037, 2015.

[16] K. Miranda, A. Molinaro, and T. Razafindralambo, "A survey on rapidly deployable solutions for post-disaster networks," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 117–123, 2016.

[17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[18] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

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

[20] G. He, "Destination-sequenced distance vector (dsdv) protocol," *Networking Laboratory, Helsinki University of Technology*, pp. 1–9, 2002.

[21] S. Murthy and J. J. Garcia-Luna-Aceves, “An efficient routing protocol for wireless networks,” *Mobile Networks and applications*, vol. 1, no. 2, pp. 183–197, 1996.

[22] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A distance routing effect algorithm for mobility (dream),” in *4th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 76–84, ACM, 1998.

[23] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, pp. 153–181, Springer, 1996.

[24] C.-K. Toh, “A novel distributed routing protocol to support ad-hoc mobile computing,” in *15th Annual International Phoenix Conference on Computers and Communications*, pp. 480–486, IEEE, 1996.

[25] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *IEEE INFOCOM*, vol. 3, pp. 1405–1413, IEEE, 1997.

[26] Y.-C. Hu, D. B. Johnson, and D. A. Maltz, “Flow state in the dynamic source routing protocol for mobile ad hoc networks,” tech. rep., IETF MANET, 2001.

[27] Z. J. Haas, “A new routing protocol for the reconfigurable wireless networks,” in *6th International Conference on Universal Personal Communications*, vol. 2, pp. 562–566, IEEE, 1997.

[28] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in *7th Annual International Conference on Mobile Computing and Networking*, pp. 166–179, ACM, 2001.

[29] A. Savvides, H. Park, and M. B. Srivastava, “The n-hop multilateration primitive for node localization problems,” *Mobile Networks and Applications*, vol. 8, no. 4, pp. 443–451, 2003.

[30] D. Niculescu, “Positioning in ad hoc sensor networks,” *IEEE Network*, vol. 18, no. 4, pp. 24–29, 2004.

[31] B. Karp and H.-T. Kung, “Gpsr: Greedy perimeter stateless routing for wireless networks,” in *6th Annual International Conference on Mobile Computing and Networking*, pp. 243–254, ACM, 2000.

[32] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *11th Canadian Conference on Computational Geometry*, Citeseer, 1999.

[33] H. Li and M. Singhal, "An anchor-based routing protocol with cell id management system for ad hoc networks," in *14th International Conference on Computer Communications and Networks (ICCCN)*, pp. 215–222, IEEE, 2005.

[34] Y. Zhao, Y. Chen, B. Li, and Q. Zhang, "Hop id: A virtual coordinate based routing for sparse mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1075–1089, 2007.

[35] F. Cadger, K. Curran, J. Santos, and S. Moffett, "A survey of geographical routing in wireless ad-hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 621–653, 2013.

[36] "Delaytolerant networking research group, "http://www.dtnrg.org"."

[37] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *ACM SIGARCH Computer Architecture News*, vol. 30, no. 5, pp. 96–107, 2002.

[38] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[39] "Sensor networking with delay tolerance, "http://down.dsg.cs.tcd.ie/sendt"."

[40] M. Grossglauser and D. Tse, "Mobility increases the capacity of adhoc wireless networks," in *IEEE/ACM Transactions on Networks*, 2002.

[41] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-200006, Duke University*, 2000.

[42] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 252–259, ACM, 2005.

[43] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *ACM International Symposium on Mobilde Ad Hoc Networking and Computing (MobiHoc)*, 2003.

[44] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *4th ACM International Symposium on Mobile Ad hoc Networking and Computing*, pp. 257–266, ACM, 2003.

[45] K. Tan, Q. Zhang, and W. Zhu, "Shortest path routing in partially connected ad hoc networks," in *IEEE Global Telecommunications Conference*, vol. 2, pp. 1038–1042, IEEE, 2003.

[46] B. Burns, O. Brock, and B. N. Levine, "Mora routing and capacity building in disruption-tolerant networks," *Ad hoc Networks*, vol. 6, no. 4, pp. 600–620, 2008.

[47] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network*, vol. 34. ACM, 2004.

[48] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, 2007.

[49] E. C. De Oliveira and C. V. D. Albuquerque, "Nectar: a dtn routing protocol based on neighborhood contact history," *ACM Symposium on Applied Computing*, 2009.

[50] M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, 2009.

[51] T. F. Leguay, Jérémie and V. Conan, "Evaluating mobyspace-based routing strategies in delay-tolerant networks," *Wireless Communications and Mobile Computing*, 2007.

[52] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic dtn routing with window-aware adaptive replication," in *4th ACM Asian Conference on Internet Engineering*, 2008.

[53] B. Link, J. Ágila, and et al, "Simbetage: utilizing temporal changes in social networks for pocket switched networks," *1st ACM Workshop on User-provided Networking: Challenges and Opportunities*, 2009.

[54] J. C. Hui, Pan and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, 2011.

[55] M. M. C. D. Mtibaa, Abderrahmen and M. Ammar, “Peoplerank: Social opportunistic forwarding,” in *IEEE INFOCOM*, 2010.

[56] G. C. S. Jahanbakhsh, Kazem and V. King, “Social-greedy: a socially-based greedy routing algorithm for delay tolerant networks,” in *2nd International Workshop on Mobile Opportunistic Networking*, 2010.

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

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

[59] Y. Cao and Z. Sun, “Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 654–677, 2013.

[60] Y. Zhu, B. Xu, X. Shi, and Y. Wang, “A survey of social-based routing in delay tolerant networks: Positive and negative social effects,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 387–401, 2013.

[61] W. Zhao and M. H. Ammar, “Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks.,” in *9th IEEE International Workshop on Future Trends of Distributed Computing Systems*, vol. 3, p. 308, 2003.

[62] T. Hirogaki, E. Aoyama, K. Ogawa, N. Hashimoto, and M. Matsumura, “Cam systems based on traveling salesman problem from time perspective for high density through-hole drilling,” in *ASME Pacific Rim Technical Conference and Exhibition on Integration and Packaging of MEMS, NEMS, and Electronic Systems*, pp. 1783–1789, American Society of Mechanical Engineers, 2005.

[63] S. S. Ray, S. Bandyopadhyay, and S. K. Pal, “Gene ordering in partitive clustering using microarray expressions,” *Journal of Biosciences*, vol. 32, no. 1, pp. 1019–1025, 2007.

[64] G. Gutin and A. P. Punnen, *The traveling salesman problem and its variations*, vol. 12. Springer Science & Business Media, 2006.

[65] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, 1997.

[66] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999.

[67] Z. H. Ahmed, "Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator," *International Journal of Biometrics & Bioinformatics (IJBB)*, vol. 3, no. 6, p. 96, 2010.

[68] S. Yuan, B. Skinner, S. Huang, and D. Liu, "A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms," *European Journal of Operational Research*, vol. 228, no. 1, pp. 72–82, 2013.

[69] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and Q. Wang, "Particle swarm optimization-based algorithms for tsp and generalized tsp," *Information processing letters*, vol. 103, no. 5, pp. 169–176, 2007.

[70] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Applied Soft Computing*, vol. 30, pp. 484–490, 2015.

[71] J. J. Bentley, "Fast algorithms for geometric traveling salesman problems," *ORSA Journal on Computing*, vol. 4, no. 4, pp. 387–411, 1992.

[72] J.-H. Chang, R.-H. Jan, *et al.*, "Message ferry routing algorithm for data collection in partitioned and buffer-limited wireless sensor networks," *Journal of information science and engineering*, vol. 28, no. 4, 2012.

[73] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, pp. 395–410, 2007.

[74] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1407–1418, IEEE, 2005.

[75] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

[76] J. K. Lenstra and A. R. Kan, "Some simple applications of the travelling salesman problem," *Journal of the Operational Research Society*, vol. 26, no. 4, pp. 717–733, 1975.

[77] K. C. Gilbert and R. B. Hofstra, "A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem," *Decision Sciences*, vol. 23, no. 1, pp. 250–259, 1992.

[78] L. Tang, J. Liu, A. Rong, and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in shanghai baoshan iron & steel complex," *European Journal of Operational Research*, vol. 124, no. 2, pp. 267–282, 2000.

[79] C. Okonjo-Adigwe, "An effective method of balancing the workload amongst salesmen," *Omega*, vol. 16, no. 2, pp. 159–163, 1988.

[80] J. Li, M. Zhou, Q. Sun, X. Dai, and X. Yu, "Colored traveling salesman problem," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2390–2401, 2015.

[81] L. J. G. G. Z. R. Yu, Zhong and Y. Haiyan, "An implementation of evolutionary computation for path planning of cooperative mobile robots," in *4th World Congress on Intelligent Control and Automation*, 2002.

[82] P. Toth and D. Vigo, *The vehicle routing problem.* SIAM, 2002.

[83] E. Wacholder, J. Han, and R. Mann, "A neural network algorithm for the multiple traveling salesmen problem," *Biological Cybernetics*, vol. 61, no. 1, pp. 11–19, 1989.

[84] A. E. Carter and C. T. Ragsdale, "A new approach to solving the multiple traveling salesperson problem using genetic algorithms," *European Journal of Operational Research*, vol. 175, no. 1, pp. 246–257, 2006.

[85] P. Junjie and W. Dingwei, "An ant colony optimization algorithm for multiple travelling salesman problem," in *First International Conference on Innovative Computing, Information and Control*, vol. 1, pp. 210–213, IEEE, 2006.

[86] C.-H. Song, K. Lee, and W. D. Lee, "Extended simulated annealing for augmented tsp and multi-salesmen tsp," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 2340–2343, IEEE, 2003.

[87] Y. Y. Ma, Ming and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, 2013.

[88] J. X. He, Laing and Y. Yu, "Optimize multiple mobile elements touring in wireless sensor networks," in *IEEE International Symposium on Parallel and Distributed Processing with Applications*, 2009.

[89] A. S. He, Ting and K.-W. Lee, "Dispatch-and-search: dynamic multi-ferry control in partitioned mobile networks," in *Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2011.

[90] K.-W. L. He, Ting and A. Swami, "Flying in the dark: controlling autonomous data ferries with partial observations," in *11th ACM international Symposium on Mobile Ad hoc Networking and Computing*, 2010.

[91] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[92] J. Wu, S. Yang, and F. Dai, "Logarithmic store-carry-forward routing in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 6, pp. 735–748, 2007.

[93] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 187–198, ACM, 2004.

[94] D. Henkel and T. X. Brown, "Towards autonomous data ferry route design through reinforcement learning," in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–6, IEEE, 2008.

[95] E. A. Feinberg and A. Shwartz, *Handbook of Markov Decision Processes: Methods and Applications*, vol. 40. Springer Science & Business Media, 2012.

[96] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[97] A. Mansy, M. Ammar, and E. Zegura, "Deficit round-robin based message ferry routing," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pp. 1–5, IEEE, 2011.

[98] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on Networking*, no. 3, pp. 375–385, 1996.

[99] T. Simon and A. Mitschele-Thiel, "A self-organized message ferrying algorithm," in *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pp. 1–6, IEEE, 2013.

[100] T. Simon and A. Mitschele-Thiel, "Next-hop decision-making in mobility-controlled message ferrying networks," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pp. 9–14, ACM, 2015.

[101] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 851–871, 2000.

[102] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 97–116, 1999.

[103] R. Beckers, O. Holland, and J.-L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Artificial Life IV*, vol. 181, p. 189, 1994.

[104] P. Valckenaers, M. Kollingbaum, H. Van Brussel, *et al.*, "Multi-agent coordination and control using stigmergy," *Computers in Industry*, vol. 53, no. 1, pp. 75–96, 2004.

[105] N. Kumar, R. K. Karambir, *et al.*, "A comparative analysis of pmx, cx and ox crossover operators for solving traveling salesman problem," *International Journal of Latest Research in Science and Technology*, vol. 1, no. 2, pp. 98–101, 2012.

[106] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Addison Wesley Reading, MA*, 1989.

[107] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

[108] Y. Fu and Y. Yi, "Vehicle scheduling in parking station with modified cycle crossover ga," in *International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 1312–1315, IEEE, 2015.

[109] M. A. Al-Omeer and Z. H. Ahmed, "Comparative study of crossover operators for the mtsp," in *International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–6, IEEE, 2019.

[110] M. Gen and L. Lin, "A new approach for shortest path routing problem by random key-based ga," in *8th Annual Conference on Genetic and Evolutionary Computation*, pp. 1411–1412, ACM, 2006.

[111] D. Lei, "Solving fuzzy job shop scheduling problems using random key genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 1-4, pp. 253–262, 2010.

[112] K.-M. Lo, W.-Y. Yi, P.-K. Wong, K.-S. Leung, Y. Leung, and S.-T. Mak, "A genetic algorithm with new local operators for multiple traveling salesman problems," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 692–705, 2018.

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition	Notes
AODV	Ad hoc On-demand Distance Vector	ad hoc routing protocol
Ce	Central entity	a node which manages the wireless network
d	distance	metric in the mobility decision function of ferries
DAF	Destination Aware Forwarding	message forwarding between ferries
Fu	Full	-
fb	ferry buffer state	metric in the mobility decision function of ferries
FB	Ferry Buffer	on-the-fly mobility decision maker in ferries
FBD	Ferry Buffer Distance	on-the-fly mobility decision maker in ferries
Flood	Flooding in all nodes	message replication to relay nodes
GF	Greedy Forwarding	message forwarding between ferries
GPS	Global Positioning System	-
GPSR	Greedy Perimeter Stateless Routing	ad hoc routing protocol
K	Known	-
Keep	forward and Keep	message forwarding between ferries
len	length	-
lvt	last visit time	metric in the mobility decision function of ferries

Abbreviation	Definition	Notes
MANET	Mobile Ad hoc NETworks	ad hoc networks with mobile nodes
MDP	Markov Decision Process	-
MF	Multi-Ferry network	-
Mo	Mobile	-
mr	message generation rate of nodes	metric in the mobility decision function of ferries
MRMF	Multiple Route for Multiple Ferries	network architecture for ferries with offline path based on mTSP and a hub node for inter-cluster communication
MRMF-R	Multiple Route for Multiple Ferries with a Rendezvous	network architecture for ferries with offline path based on mTSP and a rendezvous for inter-cluster communication
MRMF-R-B	Multiple Route for Multiple Ferries with a Rendezvous and Balanced travel path of ferries	similar to MRMF-R with balanced length of ferries paths
mTSP	multiple Traveling Salesmen Problem	-
Nm	Node mobility	-
nv	node visit count	metric in the mobility decision function of ferries
obs	observation	-
ODMF	On-the-fly next-hop Decision maker for Multi-Ferry networks	on-the-fly mobility decision maker in ferries
ODMF-NC	on-the-fly next-hop Decision maker for Multi-Ferry networks-No Coordination	same decision function as ODMF without indirect signaling of ferries
OLSR	Optimized Link State Routing	ad hoc routing protocol
Pa	Partial	-

Abbreviation	Definition	Notes
POMDP	Partial Observable Markov Decision Process	-
PR	Pure Replication	message replication between ferries
S/D	Static/Dynamic	type of trajectory of ferries
SF	Single Ferry network	-
SOMF	Self-Organized Message Ferry network	on-the-fly mobility decision maker in ferries
SOMF-MH	Self-Organized Message Ferry network-Message generation History	on-the-fly mobility decision maker in ferries
SRMF	Single Route for Multiple Ferries	network architecture for ferries with offline path based on TSP
St	Stationary	type of mobility
TAF	Trajectory Aware Forwarding	message forwarding between ferries
TAR	Trajectory Aware Replication	message replication to relay nodes
TR	Trajectory aware Replication	message replication between ferries
Trm	Traffic of messages	-
TSP	Traveling Salesman Problem	-
UAV	Unmanned Aerial Vehicle	flying robot with a wireless interface
U	Unknown	-
VANET	Vehicular Ad hoc NETworks	-
Zone25/50	Zone-based replication to relay nodes (25/50: size of the zone)	message replication to relay nodes