Cooperative On-the-Fly Decision Making in Mobility-Controlled Multi Ferry Delay Tolerant Networks

Mehdi Harounabadi(✉), Alina Rubina, and Andreas Mitschele-Thiel

Integrated Communication Systems Group, Ilmenau University of Technology, Ilmenau, Germany
{mehdi.harounabadi,alina.rubina,mitsch}@tu-ilmenau.de

Abstract. This paper presents a cooperative On-the-fly Decision maker for Multi Ferry (ODMF) delay tolerant networks. ODMF chooses the next node to visit for mobility-controlled data ferries. In ODMF, each ferry keeps a history about its last visit time to nodes and applies it in its decision making. Moreover, a ferry shares and exchanges the history information with other ferries through an indirect signaling. While there is no direct communication among ferries in our assumptions, ferries employ nodes as relays for the indirect signaling of control information. The simulation results show that ODMF outperforms the TSP and on-the-fly approaches in terms of message latency. In scenarios with high number of ferries, it can be seen that the impact of indirect signaling on the performance is more notable. In ODMF, the travel time is reduced for messages that are in the buffer of ferries by the cooperative decision making of ferries. In addition, we study and discuss the impact of increasing number of ferries and increasing ferry speed on the performance and cost of a message ferry network. We show that a required performance can be achieved with less cost by increasing speed of ferries than increasing number of ferries.

Keywords: Delay tolerant networks · Mobility-controlled · Multi ferry · On-the-fly decsion maker

1 Introduction

In Delay Tolerant Networks (DTNs), nodes can be scattered over a vast area and far from the radio transmission range of each other. In such situations, a direct communication among nodes is not possible. A data ferry is a specific mobility-controlled wireless node that travels among disconnected nodes in a DTN and exchanges messages among them [1]. A message ferry network is a DTN where the communication among nodes is only possible employing a data ferry. The trajectory of a data ferry is called 'ferry path'. It impacts the performance of a message ferry network. In a network with an asymmetric traffic load of nodes where the traffic generation rate in nodes is different and variable in time,

Y. Zhou and T. Kunz (Eds.): ADHOCNETS 2016, LNICST 184, pp. 246–257, 2017.
DOI: 10.1007/978-3-319-51204-4_20

visiting nodes by a ferry based on offline ferry path planning approaches such as the solution for the Traveling Salesman Problem (TSP) may degrade the performance. The TSP solution provides the shortest path to visit all nodes without considering the traffic load in nodes and the flow of traffic. Therefore, applying an on-the-fly decision maker in a ferry to dynamically choose the next node to visit seems to be an efficient solution for such networks. An on-the-fly algorithm adapts the trajectory of a ferry to the data traffic in a message ferry network. However, a single ferry approach provides a limited resource that is not sufficient for large or highly loaded networks. In such scenarios, adding more ferries boosts the performance [6].

In this paper, we propose a cooperative On-the-fly Decision maker for Multi Ferry networks (ODMF) where ferries share their observations using nodes as relays (indirect signaling of control information among ferries). The main goal of ODMF is making an on-the-fly decision in a ferry about the next node to visit. In ODMF, each ferry keeps the history of its last visit to nodes. Keeping the track of last visit time in ODMF helps to reduce too frequent or too seldom visits to nodes. While there is no direct communication among ferries, they share this history information through an indirect signaling via nodes to cooperate in a message ferry network.

The simulation results show that ODMF outperforms existing on-the-fly decision makers and the TSP solution as an offline ferry path planner in multi ferry networks in terms of message latency. Moreover, we show the importance of sharing history information among ferries and its impact on the performance. A message latency consist of a message waiting time in a node buffer after its generation and the message travel time in a ferry buffer (travel delay) till it is delivered at its destination. We study the impact of increasing number of ferries on the constituent components of a message latency applying ODMF and existing on-the-fly approaches. In addition, we investigate the impact of ferry speed in multi ferry scenarios. To the best of our knowledge, this is the first work that applies and evaluates a cooperative on-the-fly decision making in multi ferry networks.

The remainder of this paper is organized as follows: First, we will discuss existing work for single and multi ferry networks in Sect. 2. In Sect. 3, we describe our network model. Our on-the-fly decision maker algorithm is presented in Sect. 4. Section 5 introduces the indirect signaling among ferries via nodes for the cooperative decision making. In Sect. 6, we evaluate the performance of ODMF in multi ferry scenarios and study the impact of the number and speed of ferries on the performance and costs of the network.

2 Related Work

Message ferry path planning is modeled as a TSP problem in [1,2]. Authors find the shortest path for a ferry to visit all nodes. However, some visits may be unnecessary and a waste of resources. Moreover, the TSP solution do not consider the traffic load and the flow of traffic in a network. In [3], authors present an approach to control the mobility of a ferry. The problem is modeled as a Markov Decision Process (MDP) and solved by a heuristic algorithm to find a sequence of nodes to visit in a Round-Robin (RR) fashion. A self-Organized Messages Ferrying (SOMF) algorithm in [4,5] was proposed which employs the on-the-fly decision making for mobility of a ferry in single ferry scenarios. The on-the-fly decision maker selects a node to visit based on some weighted functions. The weight for each function is learned by a ferry for a specific network topology. Learning process takes a long time and becomes useless with variations in the network. It also applies a fairness function that distributes the number of visits among all nodes uniformly which may cause frequent and useless visits of a node.

As single ferry approaches cannot be applied in large and highly loaded networks due to the performance needs. To overcome limitation of a single ferry network, authors in [6] proposed different architectures to deploy a multi ferry network. However, they did not focus on the ferry path planning. In [7], authors modeled multi ferry path planning as a Partial Observable Markov Decision Process (POMDP). In their assumptions, nodes are located in clusters and ferries visit the cluster heads to exchange messages. From a ferry point of view, the clusters are stationary. However, cluster heads are mobile within a cluster. The contribution of the algorithm is to plan the ferries paths based on the mobility model of cluster heads. The mobility model of cluster heads is given as an input for the POMDP.

None of the existing approaches take the advantages of cooperation among ferries. There is a lack of an on-the-fly decision maker for multi ferry networks in the literature which a ferry can adapt its trajectory based on the traffic of the network and the decision of other ferries.

3 Network Model

3.1 Assumptions

In our work, the network is modeled as follows: wireless nodes (N) are of two types; regular nodes ($R \subset N$) and ferry nodes ($F \subset N$).

$$System = (R \subset N) \cup (F \subset N) | R \cap F = \emptyset \tag{1}$$

From now on, we call regular nodes only 'nodes'. Nodes are assumed to be disconnected stationary wireless nodes. They can generate and consume (receive) messages. Message generation in nodes is variable in time with a random intermessage arrival time. Ferries are mobility-controlled wireless nodes that travel among nodes and transfer their messages. Location of nodes is known by ferries. A ferry itself does not generate or consume any messages. It only collects messages from nodes and delivers them to their destinations. Ferries travel always with a constant velocity. In our network, there is no obstacle to limit the direct movement of ferries between any pair of nodes. Moreover, we assume that there is no limit in the size of buffers in both ferries and nodes.

The scale of distances between nodes are considered much bigger than their radio transmission range.

$$d(i, j \in R) \gg tx_{range} \tag{2}$$

Therefore, we neglect the radio transmission range for both nodes and ferries and consider it zero ($tx_{range} = 0$). Moreover, the required time for a ferry to travel among nodes is much bigger than the required message transmission time (T_{tx}) between ferries and nodes. Thus, we neglect T_{tx}.

$$T_{travel}(i, j \in R) \gg T_{tx} \tag{3}$$

We assume that our network is a pure message ferry network and communication among nodes is only possible via ferries. Moreover, there is no direct communication among ferries. A ferry can only obtain an observation whenever the ferry visits a node. In our network, ferries plan on-the-fly for their trajectory. A ferry visits a node, obtains an observation and chooses the next node to visit. Therefore, there is no predefined path for ferries.

Next, we introduce steps for a ferry when it visits a node.

3.2 Steps of a Ferry Visit to a Node

In our multi ferry DTN, whenever a ferry visits a node, a set of functions is triggered and run sequentially in the ferry. The functions are:

1. Exchange of messages with the node
   (a) The ferry collects all messages from the node's buffer
   (b) The ferry delivers all messages for which the current node is the destination
2. Exchange the history information with the node (indirect signaling among ferries)
3. Decide about the next node to visit using ODMF
4. Traveling towards the next (decided) node

In the next section, we describe our on-the-fly decision maker in ferries that chooses the next node to visit.

4 On-the-Fly Decision Maker for Multi Ferry Networks

In this section, we propose a novel cooperative On-the-fly Decision maker for Multi Ferry (ODMF) networks. The main goal of ODMF is to make on-the-fly decisions in a ferry about the next node to visit. ODMF works only based on the local observations of a ferry and the history of nodes that a ferry saves in its memory. In our multi ferry network, there is no direct communication between ferries (no long range communication), but ferries share control information for better cooperation using an indirect signaling. Indirect signaling among ferries is done by employing nodes as relays. ODMF can be applied in single and multi ferry networks without any modifications in the algorithm.

In our network, the ODMF in a ferry decides about the next node to visit applying a *Score* function. The score function is calculated in a ferry for each node r and a node with the maximum $Score(r)$ value is selected as the next node to visit. The *Score* for each node r is calculated as follows:

$$Score(r) = \frac{fb_{norm}(r) + lvt_{norm}(r)}{d(c, r)} \tag{4}$$

where $fb_{norm}(r)$ is a function that returns a normalized value for a node r based on the number of waiting messages in the ferry buffer. A candidate node will have a bigger fb_{norm} value if it is the destination for more messages. It is calculated as follows:

$$fb_{norm}(r) = \frac{msg.count(r)}{msg.count(r_{max})} \tag{5}$$

$msg.count(r)$ is the number of messages for the node r in the ferry buffer and $msg.count(r_{max})$ is the number of messages for the node with the maximum number of messages in the ferry buffer.

The second function is based on the history of nodes in the memory of a ferry. Each ferry keeps the history of its last visit time to all nodes and applies it in its decision maker. $lvt_{norm}(r)$ returns a normalized value for the node r based on the last visit time of ferry to node r. The value for each node r is calculated as follows:

$$lvt_{norm}(r) = 1 - \frac{last.visit.time(r)}{current.time} \tag{6}$$

lvt_{norm} in the decision maker avoids any visit starvation in nodes. Visit starvation degrades the performance of a message ferry network because messages in a starved node must wait for a long time to be collected by a ferry. It also prevents frequent visits of a node in a short time window. Frequent visits of a node may waste the resource in a message ferry network, when the visit rate for a node is higher than its message generation rate.

$d(c, r)$ in *Score* function is the distance between the current node c and a possible next node r.

In the next section, we introduce the indirect signaling among ferries for a cooperative decision making. Ferries exchange their history of nodes ($last.visit.time$), that they save in their memory, using nodes as relays.

5 Cooperation of Ferries by Sharing History Information Through Indirect Signaling

As mentioned before, each ferry keeps the history of its last visit time to all nodes to apply it in its decision maker. The last visit time history of nodes in a ferry is updated when the ferry visits a node.

In our multi ferry network, each node acts as a relay for signaling among ferries. The signaling information is the last visit time history of nodes in ferries.

Similar to ferries, each node keeps this history in it's memory. The history information in a node does not refer to the history of any specific ferry. All ferries can update the history information in nodes to share their information with each other. Whenever a ferry visits a node, the ferry exchanges the history information with the node. Older history information is updated with more up-to-date information in both sides. Therefore, a ferry receives history information of other ferries through an indirect signaling via nodes. Moreover, the ferry shares its own history by updating it in the node, if the ferry has more up-to-date information.

As mentioned in Sect. 3.2, the history information exchange between a ferry and a node occurs before the decision making of a ferry about its next node to visit. Therefore, the ferry can apply more up-to-date information to its decision maker if it is available in the node. Indirect signaling among ferries can lead to better cooperation of ferries. For instance, a ferry can receive an information about a node that has been visited by other ferries recently. Thus, the ferry avoids visiting the node since it could likely be a waste of resource.

Algorithm 1 describes the update of last visit time history when a ferry visits a node. First, the ferry receives the history table that is in the node's memory. Then, the ferry updates its own history table and the history table of the node by comparing values in both tables for all nodes in the network. Finally, ferry sends the updated table to the node and the latter saves the updated history table in its memory.

Algorithm 1. Update of the last visit time (lvt) history in a ferry

receive(*lvt_hist_in_node*) ▷ *last.visit.time* is the history in the ferry
for each node r **do**
    if *last.visit.time*(r) < *lvt_hist_in_node*(r) **then**
        last.visit.time(r) ← *lvt_hist_in_node*(r)
    else
        lvt_hist_in_node(r) ← *last.visit.time*(r)
send(*lvt_hist_in_node*)

6 Simulation Study

In this section, we evaluate and study the performance of the proposed ODMF and existing approaches. In our comparisons, we evaluate two versions of ODMF. In the first version, indirect signaling of control information among ferries exists. We name this version 'ODMF'. In the second version, there is no signaling among ferries (no sharing of history information among ferries) but the decision function in a ferry is same as ODMF and we call it 'ODMF-NC' (non-cooperative). To do this, we extended the Python based single ferry simulator introduced in [4]. The main objectives of our simulations are as follows:

1. Comparison of the ODMF versions with existing on-the-fly decision makers and the TSP path planner as an offline approach in terms of message latency and the constituent components of a message latency
2. Study on the impact of increasing number of ferries on the average message latency in the network applying existing on-the-fly decision makers
3. Study on the impact of ferry speed on the performance and costs of single and multi ferry networks

In our simulations, nodes are placed randomly in each simulation run. The position of a node is restricted to a $1000 \times 1000\,\mathrm{m}^2$ area. Message generation in nodes is variable in time. It starts at $t = 0$ and runs for 1000 s. Then, the simulation is continued till delivery of all messages. We consider an asymmetric traffic load in our network. Nodes can be categorized in our model according to their message generation rates to very high rate (10% nodes), high rate (10% of nodes), normal rate (70% of nodes) and no message generation (10% of nodes) with mean inter-message arrival time of 1 s, 5 s, 10 s, ∞, respectively.

6.1 Comparison of ODMF with Existing Approaches

In the first simulation, we model the network with 20 nodes and 15 ferries and compare two versions of ODMF with existing on-the-fly decision makers and the TSP path planner as an offline approach. We run the simulation 10 times for each algorithm. In each run the topology of the network, i.e. the placement of nodes is different. Ferries start their travel from different nodes with a constant velocity of 10 m/s. The on-the-fly decision makers (other than ODMF versions) that we apply in our comparisons are 'fb' (ferry buffer) and 'SOMF' (Self-Organized Message Ferrying) [4,5]. fb is a basic on-the-fly decision maker that applies a ferry buffer function to choose a node to visit based on the number of messages in a ferry buffer for a destination. SOMF applies a visit count function in addition to the ferry buffer function and a distance function. Visit count function in SOMF distributes the number of ferry visits among nodes.

Figure 1 shows the end to end latency of messages in 5 different approaches. The end to end latency of a message refers to the time difference between a message generation in its source and the delivery of message at its destination.

In terms of the median of end to end latencies, TSP has the worst result as it is an offline path planner and does not consider the load of messages in nodes and the flow of traffic in the network. However, regarding to the dispersion of latency values and the maximum latency in the network fb is the worst approach. fb only considers messages in a ferry buffer for its decision. This causes long waiting of messages in nodes.

It can be observed that ODMF outperforms all existing approaches. Two lessons is learned from the ODMF results. First, applying the last visit time history leads to better end to end message latency in a network comparing with other metrics such as the visit count that is applied in SOMF. This can be seen by comparing ODMF-NC and SOMF which both approaches do not share any control information and their difference is in their decision functions. Second, sharing history

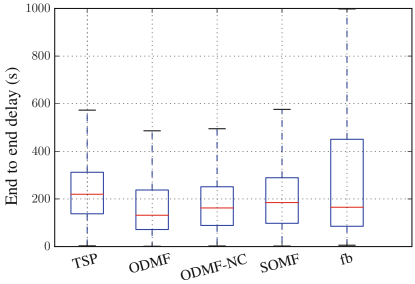

Fig. 1. The end to end latency of messages

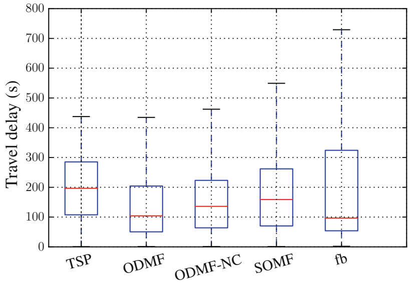

Fig. 2. Messages travel time (waiting time in a ferry buffer).

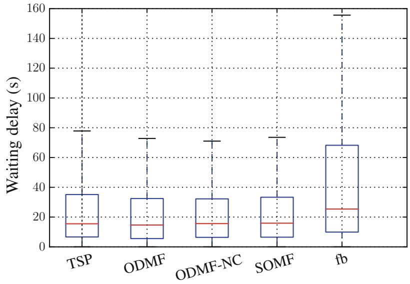

Fig. 3. Messages waiting time (in a node buffer).

information through indirect signaling in ODMF is an efficient solution for the cooperation of ferries in a multi ferry network. This can be seen by comparing ODMF and ODMF-NC. To have a deeper insight about the performance of all approaches, we divide the end to end message latency to its constituent elements. The end to end message latency consists of two parts:

1. Message waiting delay in a node buffer after its generation till it is collected by a ferry- $delay_{wait}$
2. Message waiting delay in a ferry buffer after the ferry collected it and before it is delivered at its destination. It is the time that a message travels in a ferry buffer- $delay_{travel}$.

Therefore the end to end delay of a message can be calculated as following:

$$Delay_{e2e} = delay_{wait} + delay_{travel} \tag{7}$$

Figures 2 and 3 illustrate the messages waiting delay '$delay_{wait}$' in nodes buffer and messages travel delay '$delay_{travel}$' in ferries buffer, respectively. TSP has the highest median of travel delays, as it does not consider the destination of messages in a ferry buffer. It always visits a predefined sequence of nodes in order to ensure that all the nodes are visited and the path length is minimal.

The best median value for $delay_{travel}$ occurs in fb while it chooses a node to visit only based on the waiting messages in a ferry buffer. However, it always chooses a node with maximum number of waiting messages in a ferry buffer and has no other metrics. This results in increasing the $delay_{travel}$ of messages for a destination that are in minority. ODMF has similar results to fb in terms of median value for $delay_{travel}$ but ODMF has the least maximum $delay_{travel}$ and better dispersion of values. In ODMF, cooperation of ferries by sharing the history information leads to better strategy in visiting nodes. Due to the indirect signaling of ferries, a ferry avoids visiting a node that has been visited by other ferries. It causes faster delivery of messages that are waiting in the buffer of a ferry. This is also the reason why ODMF is better than ODMF-NC. In ODMF-NC, no signaling among ferries occurs. Therefore, a node may be visited by different ferries in a short time window which impacts on the $delay_{travel}$ of messages in the buffer of ferries. In SOMF frequent visits of a node in a short time window occurs not only by several ferries, but also it may occur by one ferry while it tries to visit all nodes equally. The visit count function in SOMF may force a ferry to visit a node several times in a short time window that impacts the $delay_{travel}$ of messages in the ferry buffer.

Looking at waiting delay of messages in nodes '$delay_{wait}$', it can be seen that ODMF, ODMF-NC, SOMF and TSP have similar results. However, waiting delays in ODMF versions is slightly better due to the last visit time history in nodes. fb has the worst $delay_{wait}$ for messages while it only serves the messages in a ferry buffer. Messages in the buffer of nodes may wait for a long time to be visited by a ferry that applies fb. In fb, a ferry visits a node only if it has any message for the node or the ferry buffer is empty and in this case fb chooses a node randomly to visit.

Comparing constituent elements of end to end latency, we can conclude that the last visit time history of nodes and indirect signaling in ODMF impacts mostly on the travel delay of messages '$delay_{travel}$' while in multi ferry networks the waiting delay of messages '$delay_{wait}$' are not so different applying different strategies to visit nodes.

6.2 Study on the Impact of Increasing Number of Ferries on the Performance of Message Ferrying

In this section, we study the impact of increasing the number of ferries in a message ferry network applying ODMF and existing on-the-fly decision makers. We run the simulation 100 times and in each new run, the number of ferries is increased by one. Therefore, the number of ferries is increased from 1 to 100 and we measure the average latency of messages after each run. Moreover, we evaluate the impact of the number of ferries on the constituent elements of the average end to end latency in the network which are the average waiting delay ($delay_{wait}$) and the average travel delay ($delay_{travel}$) of messages. The network consist of 20 nodes and the placement of nodes are kept identical for all 100 runs to see only the impact of the number of ferries in the network. Ferries start their travel from one specific node (a depot) with a constant velocity of 10 m/s and

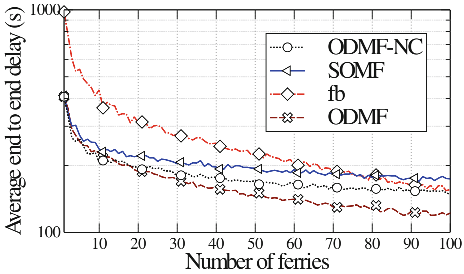

Fig. 4. Impact of the number of ferries on the average end to end latency of messages

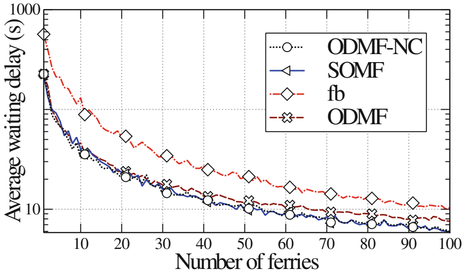

Fig. 5. Impact of the number of ferries on the average waiting delay of messages (waiting in a node buffer).

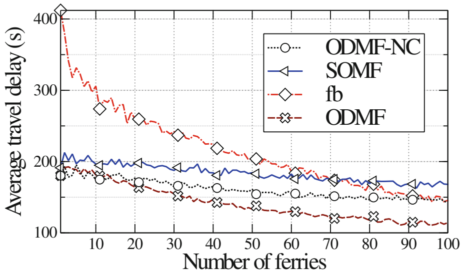

Fig. 6. Impact of the number of ferries on the average travel latency of messages (waiting in a ferry buffer).

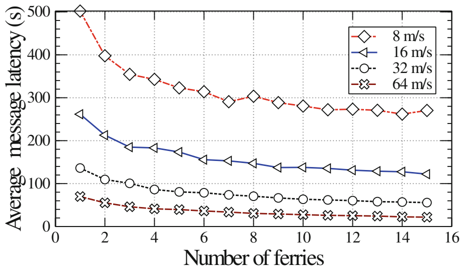

Fig. 7. Increasing number of ferries with different speeds (ODMF).

keep it till delivery of all messages in the network. The traffic generation model in nodes is same as in Sect. 6.1 for all 100 runs.

Figure 4 demonstrates the average end to end latency of messages in a network employing 1 to 100 ferries. The average end to end latency decreases by increasing the number of ferries in the network. ODMF is always the best approach and shows better performance than ODMF-NC employing more ferries in the network. However, both versions of ODMF have similar performances having few number of ferries. The difference between versions of ODMF illustrates the importance of indirect signaling in a network with high number of ferries. Sharing the history information among ferries results in better cooperation and coordination of ferries when there are high number of ferries in the network.

SOMF and ODMF-NC show a saturation in decreasing the average message latency by adding 20 ferries or more in the network. On the other hand, the average end to end latency in fb is not saturated and always decreases. fb is a better decision maker than SOMF having more than 80 ferries. However, it is the worst approach with less number of ferries. To find the reasons for the behavior of different algorithms, we look at the constituents of a message latency. Figures 5 and 6 show the average ($delay_{wait}$) and the average ($delay_{travel}$), respectively.

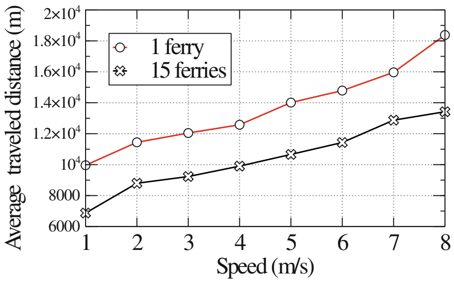

Fig. 8. The average traveled distance of a ferry (cost per ferry).

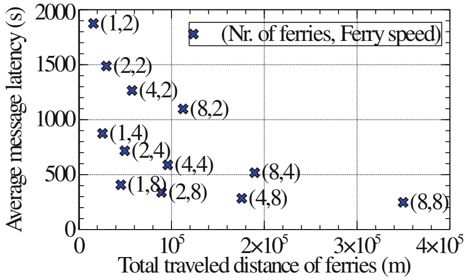

Fig. 9. Performance and total cost of different networks setups.

With more ferries in the network, the $delay_{wait}$ tends to zero in all approaches. Therefore, having 80 ferries or more, the $delay_{travel}$ has the main impact on the average end to end latency. In fb, increasing the number of ferries is more effective in $delay_{travel}$ comparing with SOMF and ODMF-NC because it only serves the waiting messages in a ferry buffer.

6.3 Study on the Impact of Ferry Speed

In this section, we study the impact of ferry speed on the performance and cost of message ferry networks. To do this, we increase the number of ferries from 1 to 15 for different speeds of ferries. The decision maker in ferries is ODMF and 20 nodes exist in the network. Traffic model in the network is same as in Sect. 6.1. Figure 7 shows the average latency of messages in the network for different number of ferries and different speeds. The average message latency is decreased by speeding up ferries. However, we can observe that a saturation in the network occurs by increasing the number of ferries. Saturation means that the increasing number of ferries does not have a tangible impact on the average end to end latency in the network. We can see that the saturation occurs earlier employing faster ferries.

Figure 8 illustrates the impact of ferry speed on the traveled distance of each ferry in the network to complete a message ferry mission. Increasing the ferry speed causes more traveled distance for each ferry. This is the cost of improvement in message latency by increasing the ferry speed.

Figure 9 shows the total traveled distance of all ferries and the average message latency for different setups in networks with 10 nodes. Each network setup is shown in a pair of (number of ferries, ferry speed). For instance, (4,8) is a network setup with 4 ferries that each ferry travels with the speed of 8 m/s. To achieve a required average message latency in a network, there are possibilities to increase the speed of ferry(ies) or number of ferries. For instance, to achieve the average message latency below 500 s, we can have both (1,8) or (2,4) setups. The best setups are those which have the least average message latency and the least total traveled distance of ferries (total cost) in a network. Therefore, setups closer to the lower left corner are desirable setups in a network. Looking at different setups, it can be inferred that adding up the number of ferries in a network imposes more total cost to a network than speeding up less number of ferries to achieve the required performance.

7 Conclusion

In this paper, we proposed a cooperative On-the-fly Decision maker for Multi Ferry networks (ODMF) where each ferry keeps a history of nodes and shares it with other ferries through an indirect signaling. The results show that ODMF outperforms existing approaches (offline and on-the-fly) in terms of message latency. Sharing the last visit time history through an indirect signaling in ODMF causes cooperative decisions in ferries and decreases the travel time of messages that are in the buffer of ferries. The impact of indirect signaling is more in multi ferry networks with high number of ferries. Moreover, we studied the impact of number and speed of ferries in message ferry networks. It can be seen from results that with increasing the number of ferries, the saturation in a network occurs earlier if we employ faster ferries. Increasing number of ferries after the saturation of a network does not improve the performance as before. Besides, increasing number of ferries imposes more cost to a network than increasing the speed of ferries to achieve a required performance. In addition to the cooperative decision making, ferries my cooperate in message forwarding by using nodes as relays to improve the performance of a message ferry network. This is our future work.

References

1. Zhao, W., Ammar, M., Zegura, E.: 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 (2004)
2. Heimfarth, T., de Araujo, J.P.: Using unmanned aerial vehicle to connect disjoint segments of wireless sensor network. In: 28th International Conference on Advanced Information Networking and Applications (AINA) (2014)
3. Mansy, A., Ammar, M., Zegura, E.: Deficit round-robin based message ferry routing. In: IEEE Global Telecommunications Conference (GLOBECOM) (2011)
4. Simon, T., Mitschele-Thiel, A.: A self-organized message ferrying algorithm. In: 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6 (2013)
5. Simon, T., Mitschele-Thiel, A.: Next-hop decision-making in mobility-controlled message ferrying networks. In: First Workshop on Micro Aerial Vehicle Networks Systems, and Applications for Civilian Use (2015)
6. Zhang, Z., Fei, Z.: Route design for multiple ferries in delay tolerant networks. In: IEEE Wireless Communications and Networking Conference (WCNC) (2007)
7. He, T., Swami, A., Lee, K.-W., Dispatch-and-search: dynamic multi-ferry control in partitioned mobile networks. In: Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing (2011)