Winter 1-31-2021

Enhanced Interest Aware PeopleRank for Opportunistic Mobile Social Networks

Yosra Saad Shahin
The American University in Cairo AUC, yosrasheen@aucegypt.edu

THE AMERICAN UNIVERSITY IN CAIRO

SCHOOL OF SCIENCES AND ENGINEERING

Enhanced Interest Aware PeopleRank for Opportunistic Mobile Social Networks

A Thesis submitted to the

Department of Computer Science and Engineering

In partial fulfillment of the requirements for the degree of

Master of Computer Science

By: Yosra Saad Shahin

ID: 800121025

Supervised by:

Professor: Sherif Aly

Professor: Soumaia Elayyat

November /2020

Abstract

Network infrastructures are being continuously challenged by increased demand, resource-hungry applications, and at times of crisis when people need to work from homes such as the current Covid-19 epidemic situation, where most of the countries applied partial or complete lockdown and most of the people worked from home. Opportunistic Mobile Social Networks (OMSN) prove to be a great candidate to support existing network infrastructures. However, OMSNs have copious challenges comprising frequent disconnections and long delays. we aim to enhance the performance of OMSNs including delivery ratio and delay. We build upon an interest-aware social forwarding algorithm, namely Interest Aware PeopleRank (IPeR). We explored three pillars for our contribution, which encompass (1) inspect more than one hop (multiple hops) based on IPeR (MIPeR), (2) by embracing directional forwarding (Directional-IPeR), and (3) by utilizing a combination of Directional forwarding and multi-hop forwarding (DMIPeR). For Directional-IPeR, different values of the tolerance factor of IPeR, such as 25% and 75%, are explored to inspect variations of Directional-IPeR. Different interest distributions and users' densities are simulated using the Social-Aware Opportunistic Forwarding Simulator (SAROS). The results show that (1) adding multiple hops to IPeR enhanced the delivery ratio, number of reached interested forwarders, and delay slightly. However, it increased the cost and decreased F-measure hugely. Consequently, there is no significant gain in these algorithms. (2) Directional-IPeR-75 performed generally better than IPeR in delivery ratio, and the number of reached interested forwarders. Besides, when some of the uninterested forwarders did not participate in messages delivery, which is a realistic behavior, the performance is enhanced and performed better generally in all metrics compared to IPeR. (3) Adding multiple hops to directional guided IPeR did not gain any enhancement. (4) Directional-IPeR-75 performs better in high densities in all metrics except delay. Even though, it enhances delay in sparse environments. Consequently, it can be utilized in disastrous areas, in which few people are with low connectivity and spread over a big area**.** In addition, it can be used in rural areas as well where there is no existing networks.

Table of Content

Table of Figures

Table of Tables

ACKNOWLEDGEMENTS

I would first like to thank my family, especially Mom and late Dad, for the continuous support they have given me throughout my entire life; I could not have done this research without them. My late Dad, I wish you were here. May God bless your soul in heavens. Second, my daughters who brought happiness and inspiration to every day. Thanks go out to my advisors, Professor Sherif Aly and Professor Soumaia Al Ayyat, for their incredible guidance (academic, scientific, and otherwise) through the course of this research. My fellow graduate students need to be acknowledged for making my experience in graduate school truly pleasurable.

CHAPTER 1: Introduction

Overview

Network infrastructures are experiencing notable challenges [1], especially with increased demand, and at times of worldwide crises when people work from home, and infrastructures are stretched to their limits [1] [2]. Ad hoc routing protocols connect the nodes without a pre-constructed network infrastructure or any related information about the network topology [1] [2] [3] [4] [5]. There are numerous terms related to the evolution of the Ad hoc networks. They are illustrated in FIGURE 1 and the following paragraphs elucidate them.

.

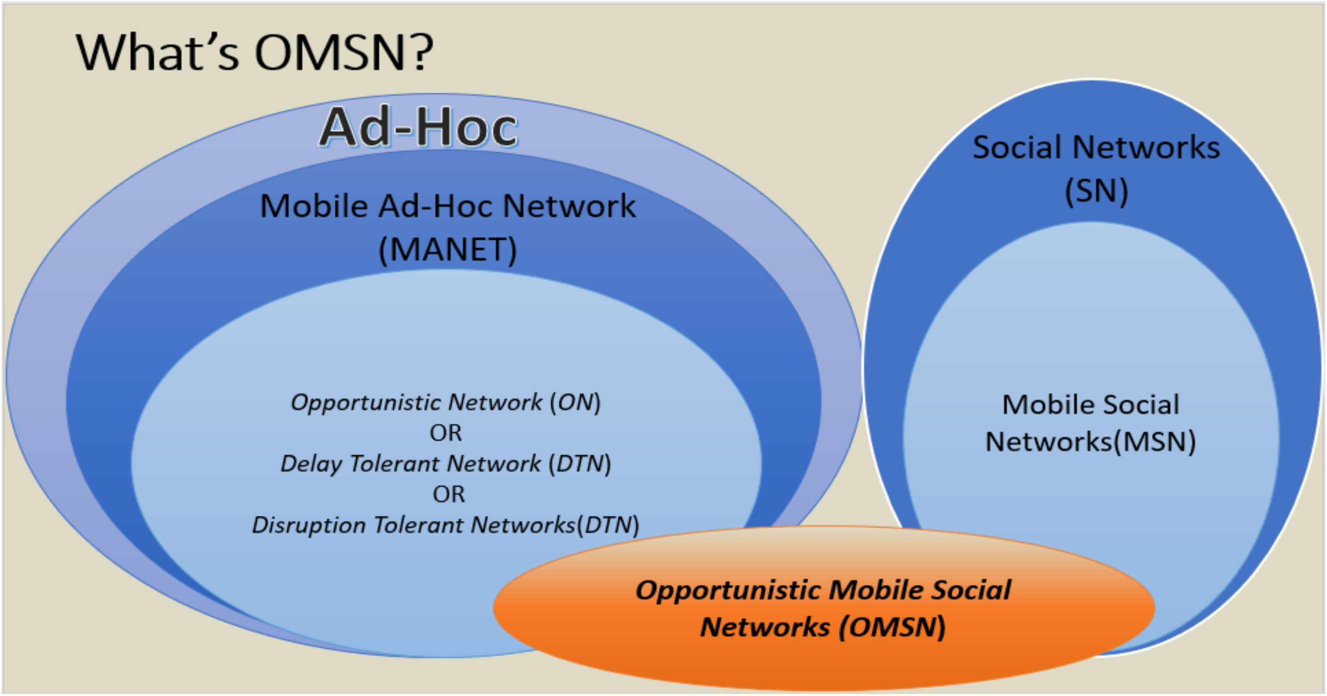

What's OMSN?
Ad-Hoc
Mobile Ad-Hoc Network (MANET)
Opportunistic Network (ON)
OR
Delay Tolerant Network (DTN)
OR
Disruption Tolerant Networks(DTN)
Social Networks (SN)
Mobile Social Networks(MSN)
Opportunistic Mobile Social Networks (OMSN)

FIGURE 1: Evolution of OMSN

Mobile Ad Hoc Network (***MANET***) *is an Ad Hoc network. However, in MANET the nodes are in motion*. Thus, the nodes may roam out from transmission range, when there is no data to be exchanged to preserve its limited power and may return. Accordingly, there may be a complete separation between the source and the destination at the time of sending the message, which results in the absence of an end-to-end path. Consequently, messages may get lost, which negatively affects the delivery ratio [6] [7] [8]. Examples of MANET include wild-animal tracking networks, pocket-switched networks, transportation networks, and battlefield networks [9].

Opportunistic Network (***ON***) *is MANET with frequent disconnections* [10]. In ON, there is no information about the network connection or the nodes' mobility patterns. To deliver the message to its destination, it uses some nodes as intermediate carriers to host the message and forward it to other nodes until it reaches the destination. In ONs, the node forward or store-and-carry the message at every hop [6] [9] [11] [12] [13] [14]. Consequently, the delay between being out of range and returning must be considered. Consequently, ON is also called ***Delay Tolerant Network*** or ***Disruption Tolerant Networks*** (***DTN***) [7] [15] [16] [17] [18].

There are various metrics to assist the performance of ONs protocols including the delivery ratio, delay, delivery cost, and others. The *delivery ratio* is the number of reached destination nodes to the total number of destination nodes that should receive the message. *Delay* is the average time consumed for a message to reach the destination node since it was sent from the source node. *Delivery cost* is the number of copies of a single message. Some researches consider the delivery cost as the product of the message size and the number of copies of a single message [4] [13] [7].

Two main characteristics are distinguishing ONs, replication, and forwarding. *Replication* describes the number of copies created for each message until it reaches its destination. Some protocols create only one copy of the message while others create several copies until the message reaches its destination. Some protocols send a copy of the message to each encountered node that does not have an extant copy such as the ***Epidemic*** protocol [1]. Other protocols send copies of the message to some of the encountered nodes based on specified constraints to eliminate the number of copies such as ***Spray and Wait (S&W)*** protocol [19]. The constraints encompass the number of copies, the time to live of the packet (TTL), or the probability of successful delivery to the destination such as ***PRoPHET*** protocol [20]. Others are creating only one copy of the message and sending it to an encountered node, such as Direct Forwarding and Randomized Forwarding, based on specific conditions for instance probability [8]. Message *forwarding* determines either to send the message or a copy of it, to the encountered node or not. It is centered on utility or probabilistic functions [13].

The interaction of a group of people constitutes a ***Social Network*** (***SN***). Thus, SN can be ***Online Network*** or ***Mobile Network*.** With the increasing usage of mobile devices, the ***Mobile Social Network*** (***MSN***) attracts more interest. The interactions in MSNs establish relationships or links among mobile devices [2]. These links can be physical contact, a shared interest, age, language, place, or many others [21] [15] [22] [23]. Such networks employ social advantages as well as the capabilities of smartphones like GPS, sensing features, and communication links [2]. To share the data efficiently, the users' information is utilized. This information includes mobility patterns, interest, local resources, and social interaction. Data sharing can be achieved throughout the entire network or in one aspect, which is called the community. The network could be divided into communities based on certain criteria such as location or interest [12] [24].

1.1. Opportunistic Mobile Social Networks

Opportunistic Mobile Social Networks (OMSN)s *are* Ad Hoc networks in which the nodes are in motion with recurring disconnections using mobile devices [11] [12] [10] [13] [15] [6] [7] [14] [9]. OMSN is the combination of ON and MSN [7]. They are instituted on creating virtual or physical social networks of users, defined by social similarities, and updating these networks with no urgency for internet connections [2] [21] [15] [22] [23] [7]

Numerous professional fields employed OMSN such as education, science, and disaster rescue. Scientific or educational groups employ OMSN to share information, establish discussion groups, and connect to other members or groups with a common interest. During disasters, when the electricity and the infrastructure networks are not available, OMSN engendered to be the merely way to communicate to get assistance or coordinate rescue procedures [2] [25] [26].

OMSN applications rely on specific pillars. First, **Platforms** represent the smart devices' specifications such as storage capacity, computation power, and battery life [27]. Second, **user information** incorporates files, locations, and calendar data, which can be employed to determine whether this node can be utilized for forwarding a message or not [28]. Third, **communication links** embrace using local wireless links such as Bluetooth and Wi-Fi. Fourth, **social profile** represents the social characteristics corroborate the interests, daily routines, and mobility patterns [29]. Fifth, **dissemination**

methods determines the next node to forward the message based on several criteria including the contact history [6] [7] [17] [19] [30] [31] [32] [33] [34] [35] [36], social relation [15] [37] [38] [39], mobility patterns or other features [16] [27]. Other features may comprise age [15], gender [40], or interest [41]. Dissemination methods can be combined to give a better performance [42] [43] [44].

1.2. Challenges

OMSNs have copious challenges comprising the deficiency of the absence of end-to-end guarantee to deliver the message to its destination [45] [46]. These challenges are demonstrated in the following paragraphs and in FIGURE 2.

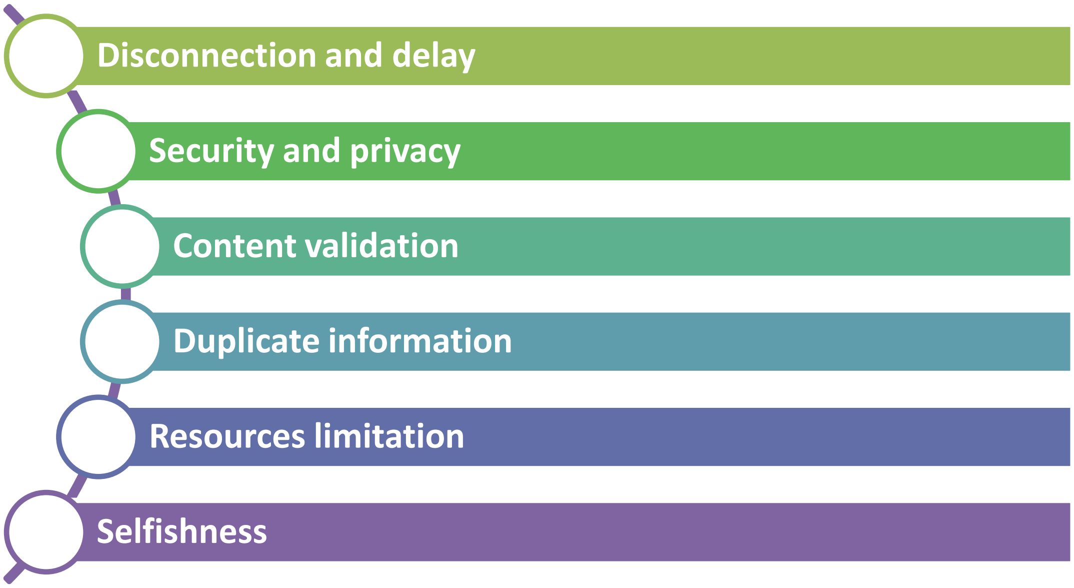


FIGURE 2: OMSNs Challenges

1. Disconnection and delay, which are essential features in OMSNs, are occurred as the nodes are frequently disconnected. Consequently, the messages might be lost [45] [46].

2. Security and privacy are desirable to create secure connections, encrypting sensitive data, authenticating, and authorizing the users [45] [46].

3. Content validation is required to certify that the received content is related to the user's preferences and benign [45] [46].

4. Duplicate information needs to be traced and deleted [45] [46].

5. Resources limitation, which encompasses power, computation, and network capabilities, constitutes a wise consumption of them [45] [46].

6. Selfishness is established when a node obtains content and is not willing to share it due to its battery or memory limitations. This problem can be handled by detecting the selfish nodes and stop sending any content to them until they start sharing their stored content with the other nodes [45] [46].

To resolve the first challenge, many researches utilized several features, which may combine to constitute improved results. They support contact history [3] [8] [6] [31] [32] [47] [48], centrality [15][71] [49], constituted communities [50] [51] [52] [53] [54], friendship [55] [56] [57] [58] [59] , nodes' mobility [60] [61], location [62] [63] [64], direction [65] [66], social [67] [42] [68] such as PeopleRank [43], and interest such as IPeR [44].

1.3. Motivation

Even with the absence of end-to-end communication between source and destination in OMSN, nodes communicate in multiple hops. The major dilemma of message forwarding is thus the selection of appropriate relay nodes, which can reduce delay without flooding the network with enumerable copies of the messages [66].

One suggestion is to pick up the most socially popular nodes, as they are more likely to meet the destination nodes. This popularity can be measured by ***PeopleRank***. PeopleRank is an algorithm that is built upon PageRank, which is an algorithm used in Google's search engine to define the relative importance of a Web page within a set of pages [43].

Furthermore, the characteristics of OMSNs define it as a good candidate for applications such as socially aware advertising [69]. In such an environment, the routing mechanism must forward the messages to the nodes, which have an interest in the content of the message [44].

Another solution is to create routing tables to control the exchanged copies of messages. Routing tables store information about the previous contact among nodes and share this information when they pass by each other. This way there is no need for extra control messages prior to data exchange [23]. This can be accomplished as two hops prediction [70] or more [49].

Another suggestion is that if a message is distributed to diverse locations, there is a higher probability to reach the destinations [66].

Another work is Interest Aware PeopleRank (IPeR), which was developed as an interest and socially aware algorithm that outperformed comparable algorithms [44]. In IPeR, the privacy of each node is maintained as each node calculates its value individually and they share a final value that is not reversible. Thus, other nodes cannot extract the private info of the node's profile.

1.4. Thesis statement

In this research, we contribute to enhancing the performance of some of the well-established algorithms used for OMSNs including, but not limited to, enhancing delivery ratio and reducing delay. Our work builds upon Interest Aware PeopleRank (IPeR), which was developed as an interest and socially aware algorithm that outperformed comparable algorithms [44]. We worked on three major fronts (1) explore the multi-hop variant, which is Multiple Hops Interest Aware PeopleRank (***MIPeR***) (2) embracing direction as a guiding criterion in ranking nodes in support for content forwarding decision making based on IPeR, Direction and Interest Aware PeopleRank (***Directional-IPeR***), and (3) utilizing the combination of MIPeR and Directional-IPeR in ranking nodes in support for content forwarding decision making Directional Multiple Hops Interest Aware PeopleRank (***DMIPeR***).

MIPeR is a variant of IPER that included the calculations of the ranks of neighboring nodes based on multiple hops. We experimented MIPeR using Two-hop and Three-hop variants of IPeR and calculated the ranks of neighboring nodes using the most used statistical measures, namely, average, maximum, and harmonic mean. For each experiment, different interest distributions as well as different user densities were employed. Based on the results, the harmonic mean 2-hop IPeR (2Har-MIPeR) algorithm performed better in terms of delivery ratio, the contacted interested forwarders ratio, and delay compared to IPeR. It is the best algorithm in terms of cost and F-measure compared to all MIPeR proposed algorithms. It even performed better than the 3-hope 3-Har-MIPeR in terms of F-measure and cost.

For Directional-IPeR, we utilize different values of what we call the tolerance factor to experiment with different ways of selecting forwarder nodes while keeping direction into consideration. The tolerance factor is a percentage, namely, 25%, 50%, and 75%. We multiply the IPeR value by one of these tolerance factors to come up with a new value which is less than the IPeR value to constitute a threshold below which we cannot send the data to nodes whose IPeR value is less than this threshold. For each experiment, different interest distributions as well as different user densities are employed. Based on the results of the simulation runs, adding direction guidance to IPeR with a 75% tolerance factor performs the best in terms of delay and delivery ratio compared to IPeR, MIPeR, and DMIPeR.

Our findings are defined as (1) adding multiple hops to IPeR enhances its delivery ratio, contacted interested forwarders ratio, and delay slightly. However, it increased the cost and significantly decreased F-measure. (2) using a 3-hop variation of the algorithm does not perform any significant gain. (3) 2Har-MIPeR is the best compared to the proposed algorithms in terms of cost and F-measure. (4) 2Max-MIPeR is the best in terms of delay, delivery ratio, and the ratio of contacted interested forwarders. (5) using 2-hop awareness can be useful for environments in which we can sacrifice cost to contact more destinations and interested forwarders such as disaster rescue. (6) the addition of direction awareness to a tolerant version of IPeR (Directional-IPeR-75) which improved delivery ratio and the number of reached interested forwarders. Even with the absence of some of the uninterested forwarders' support in delivering the messages, the performance is enhanced compared to IPeR. (7) Adding multiple hops to Directional-IPeR-75 does not gain any enhancement. (8) Directional-IPeR-75 performs better in high densities in all metrics compared to IPeR except delay. However, it enhances delay in less crowded environments. Consequently, it can be utilized in rural or disastrous areas, in which few people have internet access with low connectivity and spread over a big area

1.5. Thesis Layout

The literature review is in the following section, which is CHAPTER 2. Then, the Multiple Hops Interest Aware PeopleRank (***MIPeR***) is in CHAPTER 3. ***Directional-IPeR*** is in CHAPTER 4. Directional Multiple Hops Interest Aware PeopleRank (***DMIPeR***) is in CHAPTER 5. Adding Directional Guidance other related states of the art algorithms is in CHAPTER 6. Comparing Directional-IPeR to Social Cast is in CHAPTER 7. Finally, the conclusion and future work are illustrated in CHAPTER 8.

CHAPTER 2: Literature review

Overview

There are numerous protocols used in ***Opportunistic Mobile Social Networks (OMSN)***. To introduce some of them in details, they are classified into different categories. Some of them rely on only one category while the majority rely on two or more categories to achieve better performance. For this reason, the categorization of the protocols is not mutually exclusive as one protocol may belong to several categories. In this CHAPTER, the different categories of OMSN protocols are introduced in detail including flooding, random, contact, centrality, community, friendship, mobility, location, direction, social, interest, power and content-based protocols.

2.1. Flooding Protocol

It is one of the earlier attempts in this field. It is grounded on flooding the network with the message just like a disease, which spreads with no constraints [1]. The most popular flooding protocol is Epidemic routing protocol.

2.1.1. Epidemic routing protocol

It is a blind-routing full-flooding protocol. It is based on four assumptions. First, the sender is not connected to any network base station. Second, there is no information about the location or the route to the destination. Third, the receiver may be moving to different places. Fourth, other nodes than the source and the destination are moving and may come near to the communication range [1] [6] [47] [48]. In the Epidemic routing protocol, when two nodes meet, they compare their lists of messages, which enclose the IDs of these messages. If the message is not in the list of one of the two nodes, its buffer storage is checked to make sure it has space to store the message. If there is free space, the message is copied to the buffer of this node and its list is updated accordingly. This way, the Epidemic protocol can deliver the message to its destination with the absence of a connected path from the source to the destination.

The advantages of the Epidemic protocol are increasing the delivery rate of messages with decreased latency. However, it devours resources. To control resources consumption, the protocol maintains a hop count for each message as well as a controlled buffer space in each node. To be more specific, if the number of hops of the message exceeds the allowed hop count, the message would be removed. In addition, if the allowed buffer space in any node is full, the node will stop receiving any message. Epidemic routing protocol is useful especially when there is lack of information regarding network topology and nodes mobility patterns [1] [19] [71]. However, it floods the network with several copies of each single message. Consequently, it has a high delivery cost and thus consumes resources [8] [16].

2.1.2.Priority Based Forwarding for Epidemic Routing (PBFER)

Priority Based Forwarding for Epidemic Routing (PBFER) is an improved version of Epidemic routing. The aim is to prioritize the messages to enable the higher priority messages to be delivered to the destination with reduced delay compared to the other nodes. The priority is based on **urgency and security**. A message has a high urgency when it needs to reach its destination urgently. Considering security, the node, that has a high security needs, is to be copied to a smaller number of nodes compared to others. Hence, the message with high security has a low priority while the message with high urgency has a high priority. The priority of each one is scaled from 1 to 10 where 1 is the highest and 10 is the lowest value. The priority of the message is calculated as (urgency + security) / 2 [72].

When a node generates a message, its parameters such as creation time and destination are specified. Then, its urgency and security values are set. After that, the priority of the message is calculated. Then, the node searches its buffer to find the messages which can be delivered to their destinations. If many messages are found, the node sends the messages in order based on their priorities. If the destination is not reachable, all the neighboring nodes receive a copy of each remaining message. Through simulations PBFER performed better in terms of the number of high priority messages received, latency, delivery probability, overhead ratio, and average hop count compared to Epidemic routing [72].

2.1.3.Speed Epidemic Routing (SEd).

A recent enhancement of Epidemic Routing is ***Speed Epidemic Routing (SEd).*** It is based on using broadcast transmission as well as reducing the number of forwarding messages in the network to enhance the delivery time. It spreads the messages faster by skipping the time needed to wait for notifications as advertisement messages. The protocol in details is as follows: at the initial period, a node resets its duplicate set and sends a message using broadcast transmission that contains its message list. The duplicate set includes the key of each previously forwarded message. The aim is to tell all other nodes about its list of messages. When other nodes receive this broadcasted message list, each one of them sends a request for the messages that it does not have. Then the node, which sent initially the broadcasted message, checks its duplicate sets. If the key of this message is not in the duplicate set, it is inserted and then it sends the message to the node that requested it by broadcasting [71]. The simulation results show that SEd is a protocol providing low packet duplication rate and high delivery rate compared to Anti Entropy, n-Epidemic, Spray & Wait, and Epidemic Message with Message List Advertisement (EMMA) [71]

2.1.4.Random Routing Protocol

The opposite of the flooding protocol is the ***Random Routing Protocol.*** In this protocol when two nodes meet, it randomly generates a probability value between 0 and 1. When the generated random value is greater than a threshold, it sends a message to the encountered node. Otherwise, it does not send a message. It creates no copies of it. It sends the message itself. Here we count on the high mobility of the nodes in order to reach the destination before the message TTL expires [8].

Our point of view is that Flooding protocol enhances delay. However, it spams the network with copies of the message. Nowadays, with the heavy usage of the internet using such methods is undoable.

2.2. Contact History based Protocols

A contact is an opportunity to share data. Each node can detect the start and end time of a contact with a physically nearby node. Each node can be in more than one contact at a time. The contact history is the sequence of the contacts with other networking nodes [21]. The Epidemic routing has no consideration of the fact that the mobility of nodes in the network is not purely random. Some nodes tend to meet more compared to the rest of the nodes. This is employed to enhance the data routing in OMSN [3] [21] [8] [6] [47] [48].

Contact history is utilized in many applications. Yu et al. utilized the history of connections and the proximity to create an elastic architecture to support the social interaction in a university campus. They used a mobile prototype which is connected to a server backend. The server consolidates the social context that is gathered to create social connections among the users. Then, the social connections are utilized to provide social interaction services by using three social applications which are (1) Where2Study, (2) I-Sensing, and (3) BlueShare. Where2Study is an application that is used to help students to find a suitable place to study and locate his/her friends. I-Sensing is an application that gathers information of a place of interest on campus based on participatory sensing, which is deployed on mobile devices. BlueShare is an application that utilized Bluetooth devices to perform media sharing based on the opportunistic network. The aim is to share interesting media to the nearby users [26] .

2.2.1.PRoPHET

The ***Probabilistic Routing Protocol using History of Encounters and Transitivity*** (***PRoPHET***) protocol is grounded on using a set of probabilities to outline the successful delivery of the message to its destination. It is a guided-routing protocol, which was established on the history of past contact. A copy of a message is directed from the current node to the encountered node only if the encountered node has a better probability to meet the destination more than the current node. Initially, the value of the probability is set to zero. When two nodes meet, the probability is updated in the two nodes accordingly. The advantage is that the probability of meeting other nodes is recalculated at each opportunistic meeting [3] [8] [6] [47] [48].

2.2.1.1. PRoPHET+

PRoPHET+ is enhancement of PRoPHET Protocol. It utilizes a weighted function to calculate the probability value to determine the **shortest path**. The weight function considers the nodes' features such as the buffer size, power, bandwidth, location and popularity. PRoPHET+ reduces data loss ratio and latency. In addition, the weights can be updated due to the changes in the network. PRoPHET + performed better than PRoPHET in terms of delivery success rate and delay [3] [20].

2.2.1.2. Priority Enhanced PRoPHET protocol

One of the applications of OMSN is to provide a communication channel when disasters occur. In such situations the infrastructure and the power providers vanish. At the same time, disaster relief volunteers strive for a way of communication to fulfill their work. Using smartphones associated with OMSN can be the answer. After a disaster occurs, several agencies set up relief camps around the affected area and distribute volunteers and resources. Each camp has a defined number of volunteers and tasks to be done [25] [73]. A command and control station are created to organize the work of the different camps. However, due to the limited battery resources of mobile devices, some of the messages may be discarded. Based on this, a sentiment analysis may take place to filter the messages to define the important ones and forward them with higher priority. Bhattacharjee et al. designed a content-based filtering and **Priority Enhanced PRoPHET protocol**. It is based on two steps which are: (1) Applying natural language processing to filter and define the messages of high priority. (2) Sending these messages using PRoPHET Protocol which is enhanced with a priority property.

Bhattacharjee et al. (2016) assume that there are three types of nodes in the network which are: (1) Command and Control Node (*CC Node*), (2) Camp Node (*C Node*) and (3) Volunteer Node (V Node). A CC node is equipped with servers or powerful desktop to generate resource allocation messages which are to be sent to C and V nodes. A C Node is a mobile device such as a laptop that gathers the messages generated by the V nodes. Then, C Nodes consolidate the messages to define the needed resources and the overall information about the situation. A V node is a wireless handheld device that generates messages of the needed resources and partial situation description. Based on this there are five types of messages which are (1) resource requirement, (2) situation descriptive, (3) resource allocation, (5) conversation, and (4) sentimental messages. Based on this assumption, they put a priority property with each message that scaled from 1 to 5. The highest priority is 5 which are given to the resource requirement messages, then situational messages with priority of 4, then resource allocation messages with priority of 3, then conversation and finally the sentimental messages [25].

The proposed protocol starts with a classification module that classifies the messages to one of the categories mentioned above using Support Vector Machine (SVM). After that, the prioritization module adds the priority property for each message. Then, using PRoPHET routing algorithm associated with the priority property, the messages are sent. The protocol is evaluated using ONE simulator and the results proved that the proposed protocol performed better in terms of delivery ration and overhead ratio of the prioritized messages compared to PRoPHET, Spray-And-Wait, and MaxProp [25]

2.2.2.Spray and Wait Protocol

In ON, the practical way to route and deliver messages is to use store-carry-forward mechanism and to make copies of the message to increase the possibility of delivering the message to its destination (replication-based routing). These networks make use of the contact opportunities to spread these copies (forwarding-based routing) [6] [31] [32]. ***Spray and Wait*** (***S&W***) is considered the most appropriate store-carry-forward routing protocol [33]. S&W routing protocol provided a better solution compared to Epidemic routing protocol. It is a blind-routing partial-flooding protocol [47]. It defines L, which is a maximum number of message copies in a network. Initially the value of L is set, and then just few copies are allowed. The aim is to achieve a good delivery rate using limited resources and controlled overhead

[6] [32] . S&W protocol is composed of two phases: the spray phase and the wait phase. In Spray Phase, L copies of messages are initially created and forwarded from the source to other nodes until the message meets the destination. In Wait Phase, if the destination is unreachable in the spraying phase, each node holds a copy of the message until it reaches the destination directly. The goal is to reduce transmissions by reducing the total number of copies per message [34].

S&W protocol is compared to the congestion control version of the Epidemic protocol in terms of delivery ratio, average delay, hop count, average buffer usage. The results showed that the delivery ratio is increased with the increased node density. In addition, the usage of Epidemic protocol with TCP leads to increased average delay due to the need of sending acknowledgment packets to the source nodes [19]. However, the main problem in S&W protocol is to define the value of L, which is the number of copies of the message. If L is low, the average delay is increased. If L is high, it will flood the network like that of the Epidemic protocol. Another problem with S&W is that it forwards the messages blindly which may reduce the chance to reach the destination [17].

Al-Ghanimi, Radenkovic & Hassan conducted a study to test S&W protocol into London during the Olympic Games. The aim is to test the high network traffic over a short period of time. They tested two mobility patterns: Random Movement Scenario and Workday Movement Scenario. Random movement scenario represents six groups of people who move randomly around the city. Workday movement represents the people's daily routine such as getting up in houses at the morning, going to work, go shopping ...etc. Spray and Wait protocol has best performance with the Random Scenario but the worst performance with the Work Day Movement Scenario [74].

Samyal & Sharma analyzed the effect of Time to Live (TTL) on Epidemic, PRoPHET and S&W routing protocols. They are analyzed based on three different metrics which are delivery probability, average latency and overhead ratio. The results showed that the delivery ratio of S&W protocol is increased progressively with the increasing TTL. In addition, a better delivery ratio is achieved when the 20%-25% L copies of the total number of nodes in the network are created with less overhead and minimum average delay in S&W protocol [75].

2.2.2.1. Binary S&W (BSW)

One of the types of S&W is ***Binary S&W*** (***BSW***). It is used with the aim to enhance the performance by having a limited number of copies of the message in the network. In BSW, the source creates multiple copies of the message as M. Any node X may receive a copy of the message or more and store them which is N. When a node has only one copy of the message it is called direct transmission. If a node has more than one copy as N, it is called the active mode. If Node X meets another node Y which has no copies of the message and node X has an active mode, it keeps n/2 copies for itself and send n/2 to node Y. The performance of this protocol is compared to the Epidemic protocol and performed better [35].

2.2.2.2. S&W enhanced with PRoPHET

S&W is enhanced with PRoPHET routing protocol proposed by Kaur & Kaur [76]. S&W is used for controlling the replication whereas PRoPHET Protocol provides a routing strategy. The aim is to define the maximum number of copies of the message until it reaches the destination. If the number of copies is one and the destination is not reachable, the number of replicas is increased by one. This makes the node starts the spray mode and forward the message to other nodes. The PRoPHET routing protocol adds a predictability value to each node which depend on its hits with the destination. If the predictability value is high, it means this node is more probable to reach the destination and deliver the message. When two nodes meet, the current node checks the predictability value of the encountered node and compares it with a threshold which is calculated based on the network demands. If the predictability value of the encountered node is above the threshold, L/2 copies are forwarded to it. This way the probability of reaching the destination is higher compared to the classic S&W and the classic PRoPHET protocols, and the number of copies of the messages is controlled [76].

2.2.2.3. Improved Spray and Wait Routing (ISnW)

Improved Spray and Wait Routing (ISnW) protocol introduces improved spray phase and improved wait phase. In the improved spray phase, the source node initially has L number of message copies. When the source node meets another node, which has no copy of the message, the source node sends L/3 copies of the message to the encountered node and keeps 2L/3 for itself. Then, each node sends 2L/9 copies of the message to other encountered nodes and keeps remaining 4L/9 for itself. This process is repeated until each node has only one copy left. Hence, the improved spray phase is over, and the improved wait phase is to follow [36]. The improved wait phase takes place after the improved spray phase. Here the nodes start to move around to find the destination. In this phase, each node tries to route its copy to other nodes via a single-copy utility-based scheme. The copy is sent to the node that has a higher utility scheme to the destination and this value must be above a certain threshold. The performance of ISnW is compared with four routing protocols which are PRoPHET, Epidemic, spray and wait and NDDR in terms of delivery probability and overhead ratio. The simulation results showed that ISnW performed better than the four protocols [36].

2.2.2.4. Sharing Spray and Wait (SSW)

Sharing Spray and Wait (SSW) is a routing algorithm that enhances the performance of S&W. This protocol is considered an adaptive one [33]. In SSW, each node stores an encounter table that has two fields which are the ID of the node and the success probability. Also, each node stores a carrying timer for a message. At the beginning of the protocol each node has only one entry in its encounter table which is its ID and the value of "one" as its success probability. When two nodes meet for the first time, an entry is added to the table which is the ID of the encountered node and zero as its success probability. When these two nodes meet again the success probability is updated as in the PRoPHET method. For the timer, it is turned on for a message (1) once it is created by the source node, or (2) when it is obtained by an intermediary node. More copies are to be sent to the encountered nodes when the value of the timer of this message in the current node is big. When two nodes meet, the node that carries the copy or the copies of the message decides to send copies to the encountered node or not based on the success probability. Further, it defines the number of copies to be sent [33].

Firstly, when a source creates a message, it makes L number of copies of it. Secondly, it sends these copies to the encountered nodes (spray phase) and waits for a definite period to let the encountered nodes to send their success probability to the source node (wait phase). If one of the success probabilities is equal to one, it means it is the destination. Here the destination and source nodes are neighbors, the message is sent to destination and the transmission is over. In other cases, if the success probability of the encountered node is higher than that of the current node, a fraction of the message copies is sent to it and then to be sent to its neighbors. If it has only one copy, so this copy is sent to the encountered node. This process is repeated until the message reaches its destination (sharing phase). The performance of this protocol is compared to p-spray-wait, spray-focus-wait [77], and SW and proved to perform better in terms of delay average, the number of messages, and the delivery rate [33]

2.2.2.5. Enhanced Fuzzy Spray and Wait Routing (EFSW)

S&W is enhanced with buffer management in ***Enhanced Fuzzy Spray and Wait Routing (EFSW)*** to increase the delivery rate and to reduce the network overhead. It uses the Fuzzy logic to order the messages based on importance to be sent during the contact opportunities based on the size of the message, number of its copies, and remaining time-to-live. These three inputs are added to a fuzzy logic controller and the priority of this message is defined [31]. To define the number of the copies of a message an enhanced method is used in which each node stores two variables for each message stored in its buffer: the estimated total number of copies (ER) and the number of message copies created by the node itself and sent it to others (MF). When two nodes meet and both have a copy of the message, they update their ER values to be their old ER + MF of the other node. The messages that have small ETR value have a high scheduling priority to be sent during contacts [31]. EFSW protocol is compared to other variations of S&W routing protocol such as TBR and AFSnW using simulations. The results showed that EFSW performed better buffer management in terms of delivery ratio and overhead ratio [31].

2.2.2.6. S&W Encounter Based Routing (EBR)

Encounter Based Routing (EBR) protocol is added to S&W protocol to provide a reliable and efficient routing mechanism by (Chaubey & Mistri, 2016). Using EBR protocol, the past encounter is used to predict the future encounter rate. The encounter rate is used to determine the number of message copies a node may send during a contact opportunity. Each node stores two pieces of information to calculate their encounter rates which are encounter value (EV), and a current window counter (CWC). The past encounters are expressed as EV, while the current time interval is expressed as CWC. When two nodes meet, the CWC is incremented, the encounter value is updated and the CWC is reset to zero for each node. This is illustrated in the following equation. This way the highest EV denotes a higher probability of successful message delivery [34]

$$EV = weight * CWC + (1\text{-}weight) * EV$$

Using EBR and S&W, when two nodes meet, if the number of copies of the message is more than one, the number of copies is decreased to the half. Then for all cases the EV for each node is calculated. Based on the new EV if the maximum number of copies of the message is less or equal to one and the EV of the encounter node is greater than the current node, the current node will send its only copy directly to the destination using direct transmission. If the encounter node has less EV value than current node, the number of copies is increased by one and the process is repeated till it reaches to the destination [34]. The simulation results showed that EBR and S&W achieved better performance with respect to delivery probability, overhead ratio, dropping packets and good-put compared with that of Source S&W and Binary S&W under different network size. However, an efficient buffer strategy needs to be studied and implemented to achieve better performance [34].

2.2.2.7. OLSR-Opportunistic (OLSR-OPP)

Optimized Link State Routing Opportunistic (OLSR-OPP) protocol is a hybrid protocol which combines the efficiency of the OLSR and the flexibility of store, carry and forward routing and S&W protocol. The goals of the authors are (1) To introduce a protocol that works efficiently in high, medium, and low levels of network connectivity, and (2) Make it simple and requires no change to the format of the packets to avoid complexity and processing overhead. The protocol is adaptive whether to use the standard MANET routing or the S&W routing based on the connectivity state of the network. OLSR is a proactive routing protocol which provides and always maintains the topology information. It gathers the topology information using HELLO message as well as Topology Control messages. Also, it can detect link breaks by using HELLO messages combined with Link Layer feedback. OLSR has a global view of the topology which enables it to ship the message to its destination even if some routes are broken. Consequently, OLSR does not need to save a route to reach the destination. It simply just drops it. Adding the opportunistic support to OLSR is achieved by using a special buffer which is called OppQueue. So, the messages are not dropped anymore they are stored in a buffer and ship them to the destination using S&W routing mechanism. This way OLSR-OPP can provide end to end routing. When a message is created by the source and the destination is reachable, the OLSR routing will be used. Otherwise, where there is a break in the route from source to destination, the S&W routing will be used. Using different network connectivity models, OLSR-OPP is tested and compared to OLSR and S&W routings. It performed better than them in terms of packet delivery ratio [78].

2.2.2.8. Incentive-Compatible Routing

Opportunistic networking relies on the idea of using intermediary nodes to ship the message from the source to the destination. Some intermediary nodes may choose not to participate in such communication as it consumes its resources such as storage and power. This is called a selfish behavior of a node. Cia, Fan & Wen introduced an incentive-compatible routing (IC) combined with multiple copies for two-hop (ICRP) and based on sequential stopping rule and game theory algorithm. The protocol considers the contact probability and transmission cost. To select the best intermediary nodes, Vickrey–Clarke–Groves (VCG) auction strategy is used to choose them and give them their rewards. In addition, a bilinear map signature scheme is used to stop the nodes from tampering [79].

This protocol is used to deliver any packet that is at most two hops away from its destination. The protocol starts with the source when it chooses an intermediary node which has the maximum reward.

This is calculated using the optimal sequential stopping rule. Then, the source defines the reward value based on game theory algorithm. Only when the intermediary node truly reports the encounter probability and routing metrics, they can get the maximum reward. This reward is the incentive to make intermediate nodes participate in a communication. In addition, Vickrey–Clarke–Groves (VCG) auction strategy is used to choose the node which has the highest reward to be the intermediary node and to reward the second highest reward as well. In this case, when a node avoids forwarding messages, it will not receive any rewards as a punishment. To define the number of nodes which act as the intermediary nodes which represent the number of copies of a message in the network, an optimal stopping time threshold is defined adaptively based on a realistic probability model [79]. This protocol performed better in terms of delivery ratio and lower transmission cost compared to Epidemic, PRoPHET, and S&W protocols [79].

2.2.2.9. Optimal Copy Allocation (OPPO)

OPPO is an ***op***timal c***op***y allocation protocol which is utilizing past contact ratio of nodes and S&W protocol. Yuan & Wang observed that aggregate contact ratio of nodes can be used to define the allocation of copies of a message while transmission. In OPPO, during the copy allocation phase, every packet and its $K-1$ copies are spread by the source node, where K is the maximum of the allowed number of copies. Based on this, other nodes may carry more than one copy of different packets. For any node I with L copies where ($1 < L < K$), when it meets another node J without copies, it sends FLOOR (LBj / Bi + Bj) copies to node J and keeps CIEL(LBi / Bi + Bj) for itself, where Bj is the contact rate of J and Bi is the contact rate of I. When Bi = Bj, copies are sent to node J using the binary S&W. When Bi < Bj, node I send all its copies to node J. OPPO is compared to S&W and performed better in terms of packet delivery ratio [80] .

2.2.3.MaxProp

Another enhancement of the Epidemic routing is MaxProp, which considers the contact of nodes. It is a guided-routing protocol that gives more priority to packets with minimum number of hops [48]. It determines the priority of the message that can be sent to the encountered node taking into consideration that the contact time is unknown and can be limited. Added, it determines the priority of deleting the messages to manage the buffer efficiently. It uses an ordered queue based on the likelihood of the future contact with the encountered node. In MaxProp, each node stores a vector that represents the likelihood to meet each other node in the network. Initially, the value is set to one. Then, when two nodes meet, the value of the encountered node is incremented by one in the vector of each node. Using depth-first search, the shortest path to the destination can be computed and the messages are ordered to be sent or deleted based on the destination costs [13] [18] [47].

Abdelkader et. al conducted a comparison among the Epidemic, PRoPHET, MaxProp, and Spray-and-Wait in a procedural form. They defined delivery ratio, delivery cost, and average packet delay as the performance metrics. The experiments showed that MaxProp accomplished better in terms of delivery rate and delivery delay, while Spray and Wait, and PRoPHET outperform in delivery cost. In addition, they concluded the outlines of efficient protocols in DTN which are: (1) Build the routing mechanism on the easily quickly collected information in the network such as the topology, energy content, data content , or social relations. (2) Preserve the network resources by minimizing the number of copies of each packet in the network. (3) Reduce the end-to-end delay by using an efficient packet selection mechanism. (4) Delete the packets which have minimal delivery chances to free the buffer space. Integrating all these four outlines can bring efficiency to the DTN routing protocols [47].

Spaho et al. (2016) conducted the same comparison between the same four protocols which are Epidemic, PRoPHET, MaxProp, and Spray-and-Wait. However, they used two more performance metrics, which are average latency and the average number of hops, besides the delivery ratio and delivery cost. The average latency time is the average time passed from the creation of the messages at source until it reaches the destination. The average number of hops is the average number of hops counts between the source and the destination node. Spray and wait performs better than other protocols in terms of delivery cost and delivery ratio while MaxProp has the lowest average latency. When the distance between source nodes and the destination node is increased, the delivery ratio is increased by 10% and the average number of hops and average latency increased for all protocols [48].

2.2.4.DTN Routing Hierarchical Topology (DRHT)

DTN Routing Hierarchical Topology (DRHT) is an enhancement that aims to dynamically adapt to the changes in the OMSN. It is based on ferry routes, message ferries, and clusters. It is relied on an asynchronies protocol that reduce the number of copies of a message consequently uses the network resources efficiently. The clusters are defined based on contact which is the meeting of the nodes physically. This way exchanging the messages between nodes in the same cluster has low delay. The main feature in DRHT is to ensure the connection among different clusters by determining the number of ferries, which is the number of packets that pass from one cluster to another. Each cluster has a cluster-head (CH), center of cluster (CC), and the ordinary nodes (ON). CH is the main node in the cluster. CC is the point where CHs can exchange data by ferries. ON is not a main or CC node. The performance of this protocol is evaluated against MaxProp, Spray and Wait and Epidemic protocols. The results showed that MaxProp performed better in terms of the delivery rate and the delivery delay [30].

2.2.5.Two Hops Prediction

One of the predication protocols that are based on contacts is the ***Two Hops Prediction Protocol*** [70]. To consider the two-hop communication, initially the source node must have the information about its neighbors' contact probability with the destination. When two nodes meet, they exchange their unordered list of node IDs which have a contact probability above a threshold. Then, each node checks whether it has any message that is destined to any of the neighbors of the nodes in the list of the encountered node. If a message is found, a copy of the message is sent. This is repeated until the message reaches the destination. This protocol performed efficiently compared to Epidemic, Random and PRoPHET Protocols [70].

2.2.6.Hybrid Approach

Another Protocol is the ***Hybrid Approach*** [81]. In this protocol the probabilistic and Epidemic routing methods are combined to achieve better performance. It uses probability and priority-based routing strategy. Each node keeps on updating itself with information about its neighbors. The information includes the (1) delivery predictability, which is based on previous contacts, (2) summary vector information, which is vector that indicates the list of messages generated by this node as well as the other messages that it carries to be delivered to other nodes, (3) and priorities of the messages. When two nodes meet, they exchange the information and update their prediction vector. The node considers only the neighbors which have greater delivery predictability than its own value. Then, based on the summary vector, the node takes the decision of sending which messages to the encountered node by comparing the message priorities in the queue of each node. The node counts the number of messages that have a greater priority than the packet to be sent. Based on this, the packet is forwarded to two nodes only which have the least number of packets with a priority greater than that of the packet that needs to be sent. In order to be sure that there is no other copy of the packet to be sent in the buffer of the two selected nodes, the 'AND' operation between summary vectors of the node and the selected two nodes takes place. Then, these two nodes forward the message to successive two's in their range until it reaches the destination. The acknowledgment is sent to the source by using the same procedure. This way the source floods the message to all the nodes within its communication range. This protocol has a better performance compared to Epidemic and PRoPHET routing in terms of the message delivery and the resource overhead [81].

2.2.7.Snapshot

Snapshot is a protocol proposed by Lee at al. which is based on the accumulation and ***sharing of the contact information*** among the nodes to obtain the network topology. First, which is called the warmup period; each node collects the sequence of snapshots of its first order neighbors and aggregates them. The exchanging of the snapshots helps to obtain the change period of membership of a node. At the end of the warmup period each node gets the information of TTL of a message, pumping period, and k messenger carrier nodes from the obtained network features. Then, each node sends the message associated with the IDs of the message's carrier to its neighbors. This is repeated until the message reaches the destination. This protocol performed better than PRoPHET Protocol in terms of transmission delay time and the network overhead. However, it performed like Epidemic protocol in the same terms [82].

2.2.8.Contact Prediction-based Routing (CPR)

Zhang et al. introduced a contact prediction routing scheme. The aim is to increase delivery ratio by relying on the instant contact information, statistical contact information, and the contact transitivity. The authors claimed that any two successive contacts are independently distributed using normal distribution function. This means when two nodes have just finished a contact, the next contact will take a relative long time. The authors had two assumptions which are (1) each node can exchange messages with its neighbors and (2) each node stores a contact table that defines the indirect probabilities of contact with other nodes, exchanges them with its one hop neighbors using broadcasts, and the neighbor nodes update their own tables accordingly, and (3) the duration time of a contact and the time intervals between any two consecutive contacts of any two nodes are independently identically distributed. The proposed protocol is compared to Epidemic, PRoPHET, and S&W protocols and proved to perform better in terms of delivery ratio, reduces delivery latency, and delivery overhead [83]

2.2.9.Fair Contact Plan

Fraire & Finochietto proposed the usage of Contact Graph Routing (CGR) to create a contact plan to predict the future contacts of nodes. However, using such a mechanism consumes the devices' resources such as memory and power. Therefore, selecting only some of the nodes can save resources. There fair contact plan can be achieved by considering time and the previous link assignments. In addition, using heuristic optimization function can enhance the route delays [84].

A contact plan is based on a contact topology however a maximum number of concurrent active nodes are added to adapt to the limited resources of the devices in the DTN or OMN networks. Using store, carry and forward routing protocol and Floyd-Warshall algorithm which relies on the calculation of the shortest delay path, the CGR is created. Simulated annealing is used to enhance the overall performance of the CGR [84]. To evaluate this protocol, a set of random contact topologies are used, and the used contact plan is evaluated based on their metrics.

We see that contact-based protocols are much better than flooding protocols. However, it is not concerned about the incentive of making the nodes participate in transmission as intermediary nodes. Such incentive can be the interest. In other words, the nodes would like to save a copy of the message if the topic of the message is an interested topic to the intermediary node.

2.3. Centrality based Protocols

The centrality of a node represents its importance in the social network. The forwarding protocol can leverage the node centrality. Central nodes have more probability to meet the destination and to deliver the message to its destination. There are different methods to determine the centrality of the node. The most popular one is betweenness centrality. ***Betweenness* centrality** relies on the total number of the shortest paths, which uses intermediary nodes. In other words, it is the number of times this node is used in a shortest path between two other nodes. It enables the communication of other nodes as if it is a bridge for exchanging the messages between the nodes [49] [85] [52] [15].

2.3.1.Similarity and Betweenness (SimBet)

Central nodes are good candidates to send messages to other nodes in the network. Due the complexity of calculating the centrality in the OMSN, the expanded ego betweenness is employed to calculate the betweenness only with the local available information [71]. This can be performed using 1-hop neighbors and 2-hop neighbors. Consequently, the algorithm can discover the whole network and calculate betweenness. Daly & Haahr, 2007 introduced SimBet, which utilizes the betweenness centrality metrics and the local social similarity to choose the intermediary nodes and route the messages efficiently [49].

After two nodes become in contact, by exchanging Hi control messages, the message holder node verifies whether the met node is a new encountered node. If it is a new neighbor, any message that is directed to this new neighbor is sent immediately. Then, an encounter request is sent from node the message holder node to the new neighbor, which replies with its list of previously encountered nodes. Then, the betweenness centrality value, and the similarity value of the message holder node is updated using betweenness and similarity utilities. Next, the two nodes exchange a summary vector, which lists the destinations of their currently carrying messages associated with their own local betweenness and similarity values for each destination. Finally, the SimBet for each node is calculated using SimBet utility. If the message holder node has a higher SimBet value for a specific destination, it sends the message request list to the new neighboring node, which removes the entire related messages from its queue and forward them to the message holder node, which in turn add these messages to its own queue. SimBet is compared to Epidemic protocol based on the delivery performance and the results were close but with minimized network overhead. In addition, SimBet is compared to PRoPHET Protocol based on delivery ratio and proved to perform better. SimBet worked efficiently in low connectivity network models [49]

2.3.2.Betweenness of expanded Ego networks

Huijuan & Kai (2015) introduced a routing protocol that is based on an improved version of the known betweenness. Using the contact history, each node determines the best encountered node to send the message to. To create a social network, each node sends a Hi message to each encountered node. When I received a reply so it still in reach otherwise it moved away. So, the nodes keep on sending Hi messages to be sure they are still connected. Each node has a buffer that contains the contact history with every encountered node which defines the contact duration. To keep on storing recent contacts, a weight is calculated for each link to define if it is latest or old. Higher the value of the weight, stronger is the link between these two nodes [86]. Then, the expanded ego betweenness is used to calculate the betweenness only with the local available information [87]. For the routing strategy, first, the link weight is calculated. Then, the virtual network is built. Finally, the expanded ego betweenness centrality is calculated. Based on this the node decides whether to send a copy of the message to the encountered node or not. The experiments showed that this protocol has a better performance in terms of delivery ratio, delivery cost, and delivery efficiency compared to Epidemic and friendship [88] routing algorithms. [86] [89]

We see that centrality-based protocols are achieving more success as they rely on active intermediary nodes which have more probability to deliver the message to its destination. However, there is no motivation for the nodes to participate in the messages transmissions as intermediary nodes.

2.4. Community based Protocols

Clustering the nodes into groups or communities in ON can accelerate the messages delivery. Added, nominating a node to be a group owner and assign a replacement of this owner when the group is broken or not connected is studied and proved efficiency [50]. In social networks, when a group of people sharing the same location interacts with each other, this is called a community. The members of one community tend to meet with more probability compared to other members who are not in the same community. This can be employed to route the messages with an enhanced chance of delivery as such communities have non-trivial associations and specific scaling properties [15] [51].

2.4.1.Bubble RAP

It is a social network protocol, which is founded on community and centrality [52]. First, a contact graph is employed to demonstrate the mobility traces. It relies on the number of contacts and the contact duration, where the physical nodes are the nodes in the graph, the edges represent the contacts, and the weights on the edges represent the contacts duration and frequency. Then, in order to detect the communities of nodes in a social network, K-CLIQUE [51] and Weighted Network Analysis (WNA) [90] are applied [52].

K-CLIQUE is utilized because a node can be a member of several communities and K-CLIQUE can detect this automatically. K-CLIQUE community is the union of a series of adjacent k-cliques. As k increases, the number of communities shrinks, but the nodes become cohesive as each node tend to be a part of at least one k-clique. On the other hand, WNA works on weighted graphs without thresholding [52].

K-CLIQUE is utilized because a node can be a member of several communities and K-CLIQUE can detect this automatically. K-CLIQUE community is the union of a series of adjacent k-cliques. As K increases, the number of communities shrinks, but the nodes become cohesive as each node tend to be a part of at least one k-clique. On the other hand, WNA works on weighted graphs without thresholding [52].

For the forwarding algorithm, three paradigms can be used. (1) WAIT: using only one copy of the packet, the node waits until it has a contact with the destination to forward the packet to it. (2) FLOOD: send the packet to all the contacted nodes. (3) Multiple-Copy-multiple-hoP (MCP): using multiple copies of the packet in which each copy has time-to-live hop count (TTL), the packets are forwarded. Their experiment proved that 4-copy-4-hop MCP is the best in terms of delivery ratio [52].

In [52] four forwarding algorithms are assessed which are LABEL, RANK, DEGREE, and BUBBLE. (1) LABEL: Defined labels are used to choose the forwarding nodes that belong to the same k-clique community. (2) RANK: using the centrality value of the nodes, the message is forwarded to the nodes that have a higher centrality values than the current node. (3) DEGREE: the packet is to be sent to the nodes with the highest degree. The degree represents the average of the degree of a node over a certain time interval, which is centered on the last interval window (S-Window), or a long-term cumulative estimate (C-Window). (4) BUBBLE: the combination of LABEL and RANK constitute the BUBBLE protocol. LABEL is used to define the destination community and RANK is used to distribute the messages. Using the BUBBLE protocol, two assumptions are considered: (1) each node fits in at least one community. (2) Each node has a local centrality within its community as well as a global centrality in the whole system. A node, which is part of multiple communities, has multiple local centralities [52].

Bubble RAP performed better compared to flooding, control flooding, ***PRoPHET***, and SimBet in terms of delivery cost. It is evaluated using the flat community structure and due to the limited size of the dataset, it is assessed over the hierarchical community structure [52].

2.4.2.ROCD

Overlapping hierarchical Community Detection (ROCD) is a protocol that performed better than BUBBLE Rap. It is considering the average contact duration, encounter frequency, shortest separation period, and average separation period to define accurately the relationships between the nodes. When it defines the relationship between two nodes it uses the contact regularity (shortest separation period, and average separation period) as a penalty parameter. In addition, it is based on overlapping communities to construct a structure tree to be used for efficient forwarding. During the forwarding of the messages, the shared nodes of the overlapping communities act as bridges between those communities. Based on the experiments ROCD achieved better performance in terms of delivery rate compared to SimBet and Bubble Rap, without affecting the average delay [53]

2.4.3.CROP

Community-Relevance-Based Opportunistic Routing (CROP) is a community-based protocol. Here the network is divided into communities. It makes the forwarding based on the community relevance. Community relevance is the inverse of the random routing from the initial community to the destination community. When two nodes meet, the message is forwarded if the encountered node has a higher community relevance to the destination community. If they have the same value of community relevance, a highest degree criterion is considered. Highest degree criteria may rely on the number of neighbors a node has. To manage the buffer wisely, the packets with the lowest community relevance is deleted. This routing protocol makes only one copy of the message in the network. CROP is evaluated and compared to BubbleRap and SimBet. The results showed that CROP performed better in terms of average delivery ration [54].

2.4.4.CFBA and CDBA

Two protocols which are ***Contact Frequency Based Approach (CFBA)*** and ***Contact Duration Based Approach*** (CDBA) are combined to provide an enhanced delivery ratio. Both rely on contacts, centrality and community protocols. In these approaches the network is divided into communities based on the contact duration and centrality. When two nodes meet and the encountered node is not a member of the destination community, the message is sent to the highest centrality among all the nodes in the network which is called global centrality. If they are members of the destination community, the local centrality is to be considered. The local centrality within the community is the one that is based on contact frequency and contact duration. Only one copy of the message is allowed in this protocol. The proposed approaches perform well in terms of delivery ratio but with a little increase in delivery cost in the campus environment, where graph density and modularity are higher, compared to the local centrality of BubbleRap [18].

2.4.5.EER and CAR

Expected Encounter Routing (EER) and ***Community Aware Routing (CAR)*** utilize the history of contacts to route the messages in OMSN. In the first phase of the protocol EER takes place. EER has two steps: the multi-copy distribution step followed by the single copy forwarding step. In the multi-copy step, the elapsed time between the two nodes' last meeting and a message's TTL are used to calculate the Expected Encounter values (EV). The nodes that have a high EV have copies of the message. The single copy step starts for each node when it has only one copy of the message. Then, each node decides to forward the message to the other node based on the Minimum Expected Meeting Delay (MEMD) among the nodes. Community Aware Routing (CAR) does the same steps of EER. However, it considers the EV of a node when it meets another node from another community. Here, in the multi-copies phase, the copies are distributed based on the number of expected encountered communities. Once a replica of the message reaches the destination community, it then sends copies of it to other nodes in the same community based on the EVs of nodes. The protocol is evaluated in terms of delivery ratio, latency, and goodput. Goodput is calculated as the ratio of the number of successfully delivered messages to the total number of the copies of the message in the network. It is compared to BR [91] , MaxProp, S&W, Spray-and-Focus [77], and PRoPHET and performed better [37].

2.4.6.Wi-Fi Direct Multi-group Networks

Casetti et al. (2018), introduced intra- and intergroup communication methodologies associated with content-centric routing. For the sake of efficient resource utilization as well as full connectivity, the formation of groups is performed automatically. Using Android devices equipped with Wi-Fi Direct can provide Device-to-Device (D2D) communication if both the source and destination nodes exist in one single group.

Wi-Fi Direct is a recent protocol which aims to enable D2D communications between nodes within a single group. One of the nodes acts as the owner of the group (GO) while the others are called clients, associate to it. Starting wireless connections using Wi-Fi Direct is called P2P Discovery, which includes (1) Device Discovery, (2) Group Formation, and (3) Service Discovery. In device discovery stage, devices exchange their own information. In the group formation stage, when two nodes need to connect to each other they negotiate their roles whether it be GO or client, by exchanging a “GO Intent” which is an integer value. The node that has the higher value will be Go node. In the service discovery stage, each device defines higher-layer services to start the connection after that. The structure of a group keeps with no changes until the GO leaves the group; hence the other nodes are disconnected and another group is to be created [92].

Casetti et al. (2018) enhanced Wi-Fi Direct to work with multiple groups. To make D2D communication work with multiple groups, logical topology based on the application layer of the ISO model is constructed. In addition, group formation is done using a smart mechanism that considers the device energy, average throughput and network coverage. The performance of group formation mechanism is evaluated by topology creation time and coverage leveraging. Based on these metrics the proposed protocol is compared to Bluetooth technology and the results showed that the proposed protocol performed better [92].

We see that community-based protocols are superior compared to flooding, contact-based and centrality-based in delivery ratio with less network overhead. However, adding other metrics to define the proper intermediary node such as interest or friendship would be much better.

2.5. Friendship based Protocols

Friendship is another social metrics that describes how two persons are close. In OMSN, to define those two nodes are friends they need to have regular and longtime duration of contacts. Another way of determining friendship in OMSN is interest. Two nodes can be friends as they are sharing common interest. Either ways; contacts and interest can be used to define the friendships between the nodes [15] .

2.5.1.Social Relationship Adaptive Multiple Spray-And-Wait Routing Algorithm (SRAMSW)

Most of the proposed enhancements on Spray-and-Wait protocol did not solve the problem of dynamic adjustment of the number of copies of messages based on actual conditions. This may result as reduction of the message delivery rate and increase in cost. Using a ***Social Relationship Adaptive Multiple Spray-And-Wait Routing Algorithm (SRAMSW)*** is a better approach. It uses the messages residence time in the node to determine the re-transmission of them to provide a better buffer management. It also employs the social relationship to choose the forwarders [93]. In SRAMSW algorithm, the node that has only one copy of the message is the one to perform the spraying, which is considered a timeout retransmission mechanism. To eliminate the number of copies in the network, The ACK message is introduced to confirm the message delivery. In addition, a timeout threshold is applied on each message copy. If the copy has a longer time than the threshold and received no acknowledgment, it is timeout and sprayed to an adjacent node which has a social relationship with the current node. The social relationship is based on betweenness centrality, similarity, and friendship. The utilization of social relationships solves the problem of dead ends [93].

The simulation results with different scenarios show that the SRAMSW algorithm has a better performance compared to traditional spray routing algorithms and the Epidemic routing in terms of message delivery ratio and reduction in the caching time of the message which improves the buffer effectively [93].

2.5.2.Effective Information Multi- Controlling Node Transmission algorithm (EIMST)

Wu et al. (2017) introduced EIMST that is based on classifying the nodes into several communities based on their socialization. The socialization is based on real social models such as car network model, rescue sites, and map models. It uses the fact that the distribution of the time interval of an encounter node is the normal distribution of logarithmic normal distribution and the negative exponential curve could work with the distribution, too [55] [56] [57]. So, when the nodes have social ties, they become friends. Nodes tend to talk to friends rather than strangers. Accordingly, a node may forward a copy of the message when the encountered node has the maximum probability $t < h$ where h is a defined stop time. The node stop sending to the encountered node if $t > h$. EIMST is evaluated in terms of delivery ratio, overhead, average delay and energy consumption. It is compared to Epidemic, Epidemic with TTL=60, Spray-and-Wait with copy=10, and Spray-and-Wait with copy=30 and performed better in transmission delay and routing overhead [94].

2.5.3.Friendship-Based Routing Protocol (FBR)

FBR is a protocol that is based on community and friendship. Based on the accurate evaluation of the strength of the friendship between the nodes, communities of the close friend nodes are constituted. FBR can detect the direct and the indirect friendship between nodes. In real life close friends contact frequently, regularly, and in long-lasting sessions. This idea is the base for the three features considered in this protocol. The three features are regularity, frequency, and longevity of the contacts. Regularity indicates the variance of the intermeeting time. Frequency indicates the average intermeeting time. Longevity is the average of the time the meeting sessions. This protocol is compared to Epidemic, PRoPHET, SimBet, and FRESH [58]. And it performed better than them in routing efficiency which is routing efficiency is the ratio of the delivery ratio to the average cost [59].

2.5.4.Social Link Awareness Based Routing (SLABR) Protocol

Social Link Awareness Based Routing (SLABR) identifies a friendship that is built among the nodes based on the frequency and the duration of the contacts. Based on the friendship, the network is divided into communities. When the message is sent to other nodes in the same community, it makes only one copy of the data that is to be sent to the node that has the strongest friendship with the destination. Among the different communities a multi-copy based binary forwarding algorithm is applied to spread the data [38]. SLABR is compared to IFR algorithm and performed better in the network overhead [95]. However, it performed worse in delivery ratio. When SLABR is compared to Friendship-Based Routing Protocol (FBR) algorithm [59], it performed better in delay latency and deliver ratio [38].

2.5.5.Social-Based Single Copy Routing (SBSCR) Protocol

Social-Based Single Copy Routing (SBSCR) is a routing protocol that is based on contact, community, similarity and friendship. Initially, the network is classified into communities. Communities detection is implemented using Inter Contact Time (ICT) [39], which is based on the total number of contacts between each two nodes. Each node stores Community Structure (CT) and ICT with other nodes. If the CT or the ICT is above a threshold, the node is added to its community. The threshold value can be defined using the power law nature of CT and ICT considering the mobility traces. These communities are then updated in each opportunistic contact. Then, Social Based Utility (SBU) is used to route the messages. SBU is based on degree similarity and friendship. The degree similarity is calculated using the Jaccard’s similarity coefficient. The friendship is calculated using a friendship function that is updated in each contact in the network. Based on the values of similarity and friendship, a node sends a message to an encountered node or not. When two nodes meet, first the message holder node checks that the encountered one is in the same community. Then, SBU is calculated. If the encountered node has a higher SBU value, then the message holder node sends the message to the encountered node. This is repeated until the node reaches the destination. This protocol creates only one copy for each message. SBSCR is compared to PRoPHET, SimBet, and BubbleRap in terms of delivery ratio and delivery delay and proved to perform better [96].

2.5.6.Social Rank and Intermeeting Time (SRIT)

SRIT is a Social Rank and Intermeeting Time protocol [97]. It is based on contact and friendship. Firstly, the protocol identifies the active users. They are the users who transmit messages frequently and have long duration of communications. This way a smaller set of users are to be focused on and consequently the number of copies of a message is reduced as these active users are to be the intermediary nodes. They are identified based on historical encounter information utility. Then, the priority of the message is calculated based on the message's lifetime remaining which is based on replication utility. This utility controls replication redundancy to enhance the delivery ratio. Finally, the social rank utility is used to calculate the social relation between the nodes which is the social relation weight. Consequently, social graph is created and used to define the routing of a message from source to destination. Simulation results showed that SRIT performed better in terms of delivery ratio, average delivery latency, and overhead ratio compared to Spray-and-Wait and EBSR [98] [97]

2.5.7.Social-based Clustering and Routing scheme (SCR)

In OMSN, social relationships among nodes can be implicit or explicit. The explicit relationship is established based on the direct contact among nodes. Implicit relationship is established based on the "friends of friends" or "circles of friends" concept which is well known in communication networks such as Facebook, and Twitter. Consequently, adaptive weights of explicit (ExSRs) and implicit (ImSRs) relationships are assigned to the contact feature of nodes. Each node picks the nodes with close social relationships to create a local cluster. In addition, a self-control utility is employed to update relationships when changes occur. Each node can forward a copy of the message to nodes which are in the local cluster of the destination. IESR is based on contact, clustering, friendship routing protocol. Each node has an encounter table to store the node identification and encounter frequency during recent time as well as a set of common friends. When two nodes meet, they exchange their encounter table. For cluster updating, each node has a delete list of the nodes which will be excluded from its local cluster. SCR is compared with PRoPHET, DRAFT [99] and BubbleRap based on delivery ratio, transmission delay, and routing overhead. SCR performed better based on the simulation results [100].

We see that friendship-based protocols are superior compared to flooding, contact-based and centrality-based in delivery ratio with less network overhead. However, adding other metrics to define the proper intermediary node such as interest or mobility would be much better.

2.6. Mobility based Protocols

The mobility patterns of the nodes are schematic as people tend to have daily routines. In addition, the activity level of mobility, which can be determined by the speed of the node, needs to be considered for a specific period. GPS can obtain the speed of any node. So, the mobility of each node can be used to route message in OMSNs [16].

Hossen & Rahim made a comparison among Epidemic, PRoPHET, MaxProp, Resource Allocation Protocol for Intentional DTN (RAPID), Binary Spray-and-Wait (B-SNW), and Spray-and-Focus (SNF). They used different number of mobile nodes for three mobility models which are (1) Random Walk (RW), (2)Random Direction (RD), and (3)Shortest Path Map Based (SPMB) [60].

Random Walk (RW) movement model represents the movement of a node towards a new location randomly. It uses random direction and speed and each node is independent from the other nodes. Random Direction (RD) movement model represents the movement of a node up to the limits of the simulation area before it changes its speed and direction. Shortest Path Map Based (SPMB) movement model represents the movement of a node starts at a random location and chooses another random location on the map and then use Dijkstra's shortest path algorithm to find out the shortest path to that location from the current location [60]. Opportunistic Network Environment (ONE) simulator is used as an evaluation tool to define three metrics which are (1) delivery probability, (2) average latency, and (3) overhead ratio. The results showed that all the protocols performed better in the SPMB compared to other movement models. B-SNW and SNF offered the best delivery ratio. The authors declared that future work needs to be done to investigate other mobility models such as working days, office activity, map based, evening activity, and home activity [60].

Mobility patterns of the nodes keep on gaining interest. Hrabcak et al. (2017) introduced Students Social Based Mobility Model (SSBMM) social mobility simulation tool. SSBMM follows the routine of student's daily life by considering the mandatory parts of human life such as going to work or college at the morning. It produces design mobility models to be employed in evaluating the mobility-based routing protocols [61].

Kawecki & Schoeneich (2016) proposed a mobility-based protocol. In this protocol, two pieces of information are collected to control the forwarding of the messages: (1) mobility patterns of the nodes and (2) their contacts. When two nodes meet, the message holder node checks the mobility pattern of the encountered node. If the encountered node has a higher mobility pattern, the message holder node sends half copies of the message to it. In other words, it relies on the fact that nodes of high mobility are more expected to meet the destination compared to low mobility nodes. In addition, the nodes that have contact history with the destination are more probable to reach it. The performance of this protocol is compared to Epidemic routing, Spray-and-Wait, and PRoPHET and proved to be better in terms of overhead ratio and effectiveness [16]

We see that mobility-based protocols are superior compared to flooding, Spray-and-Wait, and PRoPHET based on less network overhead. However, adding other metrics to define the proper intermediary node such as interest would be much better.

2.7. Location based Protocols

Tracking peoples' locations is one of the research topics which gaining attention although it is there since decays. Two major services are related to location studies, which are location-tracking and position-aware services. Location tracking always involves that other parties tracking the user's location. Position aware is based on defining the device's awareness of its own current location. A case study is conducted by Barkhuus & Dey (2003) to examine people's attitude towards location privacy related to tracking and position awareness. They found out even of the various usages of such services, people are concerned with location tracking service as it attacks their privacy. At that time, researchers thought that developments would take place related to position awareness while location tracking will have fewer developments [62]. Location based routing protocols are considered greedy geographical based routing protocols. In such protocols, a node forwards data to its encountered node only if the encountered node is closer to the destination than its own current position [63] .

2.7.1.Hotspot based Forwarding Scheme (HFS)

Kim et al. (2013) observed that people have different interests, habits, and occupations. They visit places frequently such as bus stops, malls, railway stations and stay there for a while. These locations, in which people tends to gather, are Hotspots or Stop points. In ON, forwarding a message in hotspots can guarantee it reaches a big number of nodes. The authors start with the following assumptions (1) each node has an ID that is unique, (2) a node start position is its home community, (3) each node knows the nodes which belong to its home community, (4) each node has no global information about the network, (5) each node has a label to indicate its index in its home community, and (6) each node can record its location and speed. When a node moves it periodically measures its location to define its location and calculate its moving speed. At hotspots, a node stops for a while. HFS protocol starts with the warmup period in which each node gathers the needed information to identify the hotspots. At hotspot, the mobility pattern of the nodes will be the same as they are stopping. Based on this a community is created. The nodes start to communicate and flood the messages it is like the spray phase of S&W. Then, they carry the messages until they reach a destination or another hotspot which like the wait phase in S&W [64]. This protocol enhanced better in terms of overhead compared to Epidemic and performed better in delivery delay compared to S&W protocol [64].

2.7.2.History Based Prediction Routing (HBPR)

History Based Prediction Routing (HBPR) protocol is based on predicting the nodes' movements. The authors stared with a number of assumptions which are: (1)the nodes are moving into cells of size 100m * 100m, (2) each cell has a unique ID, (3) each node can get its location while moving related to the cells, and (4) each node records the cell that it has visited in its own history table. The protocol consists of three phases which are (1) Initializing the Home Locations, (2) generation of a message and selection of next hop and (c) Acknowledgement Phase [101].

In Home location initialization phase, a node defines its home location depending on the frequently visited location. The home location table defines the node's own home location as well as the home location of each encountered node. When two nodes meet, they share their home location tables and each one update its own table accordingly. If two nodes meet again and the home location is changed for any one of them, the old entry is to be deleted and the new data is inserted in the table. It is an update mechanism to ensure the freshness of the home location tables. In the second phase, a message is generated, and its destination is defined by an ID and is appended with the message. Then, a selection utility is used to define the next hop. In the third phase, the destination after receiving the message send an acknowledgment back to the source using the same hop nodes used for forwarding. In its way, it deletes the copies of messages which are stored in the intermediary nodes. HBPR maintains four tables. (1)home location table, (2)history table which stores its previous movements, (3)next cell table which maintains node’s next predicted movement, and (4) the speed table which records the different speed of the node based on fixed timer interval [101].

This protocol is evaluated based on delivery ratio, overhead ratio, average latency, average buffer time, average residual energy, and number of dead nodes. It is compared to Epidemic and PRoPHET Protocols and performed better except in average buffer time. It performed worse than both Epidemic and PRoPHET in the average buffer time [101]. In my opinion, the assumptions are hard to meet in realistic OMSN.

2.7.3.Two-level community-based routing (TLCR)

Xia et al. (2017) introduced the model of two-level community. The aim is to choose fewer intermediary nodes to enhance the packets’ forwarding and decrease the network overhead. In the two-level community model, each node belongs to multiple small communities and big communities. Each node is a friend with each node in the small community. In addition, a group of small communities can constitute a big community when several pairs of friend nodes are above a certain threshold. First, the closeness of two nodes is defined by calculating cumulative contact ratio (CCR) which is the total number of contacts between node K and J divided by the total number of contacts of node K with all nodes. If CCR is above a certain threshold, so nodes K and J are friends. Then, they become members of the same small community. Each node stores the friend graph of networks. It enables it to locate the small community in which the destination node exists. Finally, the message can be sent to any node that is a member of the small community of the destination node [102].

Each node has friends and the friends have friends which makes the small communities overlap. When several nodes, which above a defined threshold, are friends and they belong to different small communities, so these small communities are gathered in a one big community. There is no overlapping among the big communities. The forwarding algorithm works as follow: (1) If the source and destination nodes are in the same community, the message is delivered directly, (2) If the destination is a friend in the big community and not in the small community, the message is sent to friend node in the small community which is a friend with the destination or any node in the small community of the destination node, (3) If there are no common friends, the message is sent to the node which has the greatest number of friends [102].

The performance of this protocol is evaluate and compared to Epidemic, SGBR [103] and TotalCon [104] in terms of delivery ratio, average delay and overhead. The results showed that TLCR performed better based on overhead only [102]

We see that location-based protocols are superior compared to flooding and PRoPHET based on less network overhead and delay. However, adding other metrics to define the proper intermediary node such as interest would be much better.

2.8. Direction based Protocol

People are moving around with different directions, speeds, and visit different places. If a message is forwarded to nodes, which are travelling to different places where the source node cannot approach, then we have a good chance of reaching the message to its destination [66].

2.8.1. Location Prediction-based Forwarding for Routing using Markov Chain (LPFR-MC)

Dhurandher et al. (2018) presented LPFR-MC protocol that relies on the node’s current location and its angle as well as employing Markov chain to predict the node’s next location to define whether the node is moving towards the destination or not. Then, this information is used in the forwarding. If the encountered node has a high probability to meet the destination, a copy of the message is sent to it.

The authors made the following assumptions about the nodes in the network (1) they tend to cooperate to forward the messages, (2) they have sufficient energy to participate in data transmission, (3) they have sufficient buffer space to store messages, (4) there is no malicious messages or behaviors and (5) the locations of the destinations are predetermined [65].

LPFR-MC is compared to Epidemic, PRoPHET, HBPR [105], EDR [106], and ProWait [107] protocols. It performed better in terms of message delivery probability, hop count, number of messages dropped, message overhead ratio, and average buffer time. However, it performed worse compared to all the protocols in terms delay latency [65].

2.8.2.Direction Entropy Based Forwarding Scheme (DEFS)

Jeon, et al. (2014) suggested Direction Entropy Based Forwarding Scheme (DEFS). DEFS utilizes the main direction and the direction entropy to predict the nodes' direction. The aim is to identify the nodes that have high mobility and consequently high probability of meeting the destinations of a message to achieve a balance between the network overhead and delivery delay. The main direction, which is one of the four coordinal directions, represents the high frequency of nodes' travels in a particular direction. The direction entropy represents the certainty of the direction of the node. The lower the entropy is, the higher the certainty of going to a particular direction. The assumptions of the protocol for each node are (1) it has an unique ID, (2) it has a positioning system and periodically defines its own direction based on the changes of its positions, (3) it updates its direction based on location changes, and (4) the source node does not know the position of the destination nodes [66].

Each node computes its direction and direction entropy. Then, when two nodes meet, they exchange their direction information and direction entropy. If the encountered node (1) has a different direction than the current node or (2) has the same direction but with lower direction entropy, which indicates higher certainty, the message holder node sends the message to the encountered node. To evaluate DEFS, it is examined at different thresholds. The results show that the delivery ratio increases with increasing threshold, however, this resulted a high network overhead. Based on the experiments, DEFS perform better than Epidemic, PRoPHET, and S&W. In addition, it achieves the good balance between network overhead and delivery delay [66].

2.8.3.Post disaster Situation Analysis with Resource Management

After any disaster takes place, the available networks collapse. Shelter points or camps are constructed around the disaster's area. At that time, rescuers are striving to get a way of communication [25] [73]. Gupta et al. (2016) proposed a four tiers architecture. (Tier 1) Relief Workers (RW) are mobile nodes. (Tier 2) Throw Boxes (TB) are fixed locations representing the shelter points. (Tier 3) Data Mules (DM), like vehicles, offer connection among the shelter points. (Tier 4) Master Control Station (MCS) is the base station of rescue. RWs gather information about the situation and send them to TB. Then, the messages are sent to DMs and then to MCs [73].

For the forwarding mechanism, the recent directions of mobile nodes movements are tracked. The two past successive locations of a mobile node are compared to define its direction. If a node moves in different direction than destination's direction, no copy of the message is forwarded to this node. The fittest node is the one which receives a copy of the message for the next hop. The fitness utility is based on past delivery ratio, delivery latency, current direction of movement and the node's latest nearness to the destination [73].

This protocol is tested using ONE simulator with modified mobility models. It is compared to Epidemic, PRoPHET, MaxProp, Spray-and -Wait, RAPID [108] and Encounter Based Routing (EBR) in terms of delivery ratio, delivery latency, and network overhead. It performed better than all the protocols based on the three metrics [73]

We see that direction-based protocols are superior compared to flooding and PRoPHET based on less network overhead and delay. However, adding other metrics to define the proper intermediary node such as interest would be much better.

2.9. Social based Protocol

Similarity measures the separation between two entities. In social networks, the same idea can be implemented on social aspects such as the user interests, location, gender, age, or language [15]. Consequently, when two nodes meet, the data is forwarded to the node with the higher social vector because of its better chance of meeting other nodes [40].

2.9.1.Multi Social Similarity (SOSIM) Protocol

Social Similarity (SOSIM) is a routing scheme that is based on social features. It is a multicast algorithm in which each message has multiple destinations. Based on the past meetings, each node creates a vector of social features including affiliation, city, nationality, language, country, and position. To dynamically update the social victor to resemble the nodes' contact behaviors, SOSIM uses compare-split scheme [40]. Accordingly, a tree is structured to utilize the same route to be shared among several destinations until they reach a point of separation [67]. When two nodes meet, they compare their social features to find similarities. Then, the message is sent to the one with the more social similarity to the destinations. The Euclidean similarity matric is the one, which is employed to measure the similarity between two nodes [40]. This protocol is also implemented in other two variations (1) Multi-FwdNew in which the message holder forwards the message to a newly met node, and (2) Multi-Unicast in which multicast is done by multiple unicasts. Each unicast is based on dynamic social features. To evaluate the performance of SOSIM, it is compared to Epidemic, Social profile-based multicast routing algorithm (SPM), Multi-FwdNew, and Multi-Unicast. The simulation results showed that SOSIM performed better compared to SPM based on delivery ratio, latency, and the number of hops. It also performed better to its variations [40]. In this protocol, the social profile of the destination nodes must be known.

2.9.2.Hypercube Social Similarity Protocol

It is another social feature-based routing protocol that rely on the similarity of social features. Each node possesses a hypercube to represent its social features including gender, language, and position. In addition, it utilizes F-space 3D model to divide the nodes into communities. Accordingly, the features distance is calculated. Then, the data is forwarded to the node with the least distance until it reaches the destination, which has a zero distance. In this protocol, each group is a vertex on a cube. In addition, groups are connected if they are similar in two features. When two nodes meet, the message holder node forwards the data to the encountered node if it has a lower features distance from the destination. This is repeated until the message reaches its destination group. It uses Wait-and-Focus approach. To be more specific, a message holder node may wait until it meets the destination or focus, that is to send the message to another node within the group which has a higher activity level. When comparing the proposed protocol to Spray-and-Wait and Spray-and-Focus, it proves a better performance in terms of delivery ratio and delivery latency [42]. In this protocol, the social features of the destination nodes must be known.

2.9.3.MRH and SRH

Kim & Han (2018) classified each node’s contacts into social and pass-by contacts. That is, the pass-by contact is the physical proximity between the nodes. On the other hand, the social contact involves a social interaction between the nodes. The strength of social links between nodes is calculated by Grey Relational Analysis (GRA) method which uses several metrics. In more details, the social contact is defined by a social utility. Social utility has two components which are the social tie strength and social popularity. Social tie strength is determined by defining the contact duration sum and the inter-contact rate, which is low or high contact frequency. Here, GRA is used to define proper weights for the social utility. This weight represents the social popularity of a node that is defined by the number of nodes which have social relation with this node as well as how frequent they contact [68].

The routing of this protocol is taking place as follows. Each node structures its own social contact information based on past contacts, that is to include (1) social contact duration, (2) social inter-contact rates, and (3) pass-by inter-contact rate. When two nodes meet, they update their social contacts. If the encountered node has a higher social contact than the message holder node, the data is exchanged. This protocol has two variations “Multi-path Routing for Heterogeneous size data” **(MRH),** when it considers both social and pass-by contacts, and “Single-path Routing for Heterogeneous” **(SRH),** when it considers only the social contacts [68]. This protocol is evaluated using three metrics which are (1) delivery ratio, (2) overhead, and (3) transfer failure ratio. Transfer failure ratio is calculated by (the number of forwarding failures)/ (the number of forwarding attempts). It is compared to Prophet, BubbleRap, and others. The simulation results showed that the proposed protocol performed the second-best protocol after BubbleRap [68].

We see that social-based protocols are superior compared to flooding and Spray and Wait based on less network overhead and delay. However, adding other metrics to define the proper intermediary node such as interest and location would be much better.

Conclusion

In this CHAPTER, some of the dominant related researches in the field are illustrated. Basically, they utilized many features to enhance the performance of OMSN. They relied on flooding, contact, centrality, community, friendship, mobility, location, direction or social awareness. ***We see that combining several features to choose the most proper intermediary node, can enhance the delivery ratio, delay and cost***. In the following CHAPTER, Social and interest aware protocol IPeR is combined with multiple hops ***MIPER*** to achieve better performance.

CHAPTER 3: Multiple Hops Interest Aware PeopleRank (MIPeR)

Overview

In this thesis, we address the first challenge, which is disconnection and delay with the consideration of privacy and resource limitation. We aim to enhance evaluation metrics including but not limited to delivery ratio and reduce delay as compared to the most prominent algorithms sited and for most IPeR.

Our contribution embraces enhancing the delay and delivery ratio by:

1- Exploring more than one hop (MIPeR).

2- Embrace Direction as a guiding criterion (Directional-IPeR).

3- Utilize MIPeR and Directional-IPeR

Since our goal is to evaluate the efficiency and effectiveness with which the proposed algorithms can opportunistically reach users in OMSNs. We particularly use the following metrics: delivery ratio, the ratio of contacted interested forwarders, delay, F-measure, and delivery cost. Delivery ratio is the number of reached destination nodes to the total number of destination nodes that should receive the message. Delay is the average time consumed for a message to reach the destination node since it was sent from the source node. Delivery cost is the number of forwarded copies of the message. F-measure is the harmonic mean of the precision and recall. It is utilized to implement a type of penalty for reaching uninterested forwarders. Note that the targeted true set consists of the interested forwarder nodes in addition to the destination nodes, while the false set contains the uninterested nodes. FIGURE 3 defines the main contribution (roadmap).

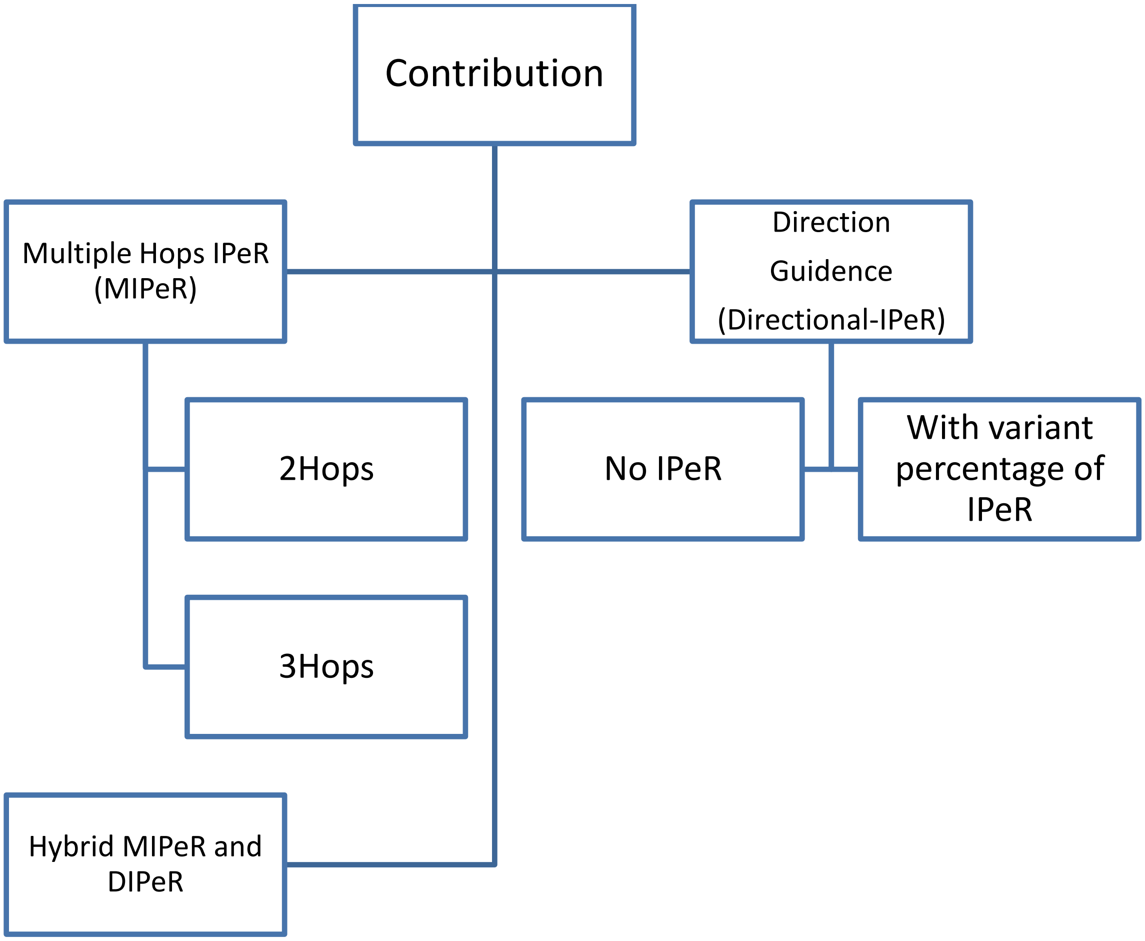

Contribution
Multiple Hops IPeR (MIPeR)
Direction Guidence (Directional-IPeR)
2Hops
3Hops
No IPeR
With variant percentage of IPeR
Hybrid MIPeR and DIPeR

FIGURE 3: Summary of Contribution (Roadmap)

3.1 General Experiment Parameters and Settings

We evaluate our proposed algorithms via the SAROS simulator [109] and validate our results using the SLAW mobility model [110]. We briefly describe our setup and present a subset of our results due to space limitations.

We used the SAROS simulator [109], which provides a wide variety of opportunistic forwarding algorithms and their related evaluation metrics. It correlates a diversity of interest distributions and social network integration associated with imported real traces. It generates random social profiles including interest for each user. In our experiments, SAROS was adjusted to work over an area of 1000m x 1000m on extracted user traces from the Self-similar Least Action Walk (SLAW) mobility model [110] .SAROS incorporates social contexts and interests among people in small scale communities such as malls.

Furthermore, the constructed friendship graph includes up to 20% of the available users in the friend list per user.

To get authentic results, each experiment is run 20 times and the average is calculated. Each run delivers 2 messages in an hour. Every 20 runs are applied with different user densities and different interest distribution, namely, discrete uniform, normal, and two disjoint interest subgraphs. The community is divided into 3 categories as per the Jaccard similarity index between their interest vector and the interest vector of the forwarded message.

In the discrete uniform interest distribution, the users are spread equally between 11 categories with varying interest ratios: namely, ranging from 0 to 1. Where each user has a constant interest vector. Accordingly, the destination set establishes 18% of the nodes while the interested forwarders cover 36%. In the normal interest distribution, the destination set embraces 2% of the community, the interested forwarders set comprises 48%, while the remaining 50% are uninterested nodes. In challenging environments such as the 2-disjoint subgraphs of interest distribution. The destination set embraces 2% of the community and 98% of the community are uninterested nodes. Thus, there are no interested forwarders in this community.

The following parameters and settings will be used across the different experiments in this research not only for MIPeR.

3.1.1 General Parameters

TABLE 1: General Experiments Parameters

Parameter	Value
No. of Users	50, 100, and 200
No. of iterations	20
No. of messages	2
Number of interests	10
Similarity interest distribution	Discrete uniform, discrete normal, and two disjoint subgraphs
Popularity Threshold	0.006
proximity	50
Initial Battery Distribution	Full Battery Distribution
SInt(source)	0.3
Max user move	1.42m / 1 sec.
Timeslots	3600
IPeR	1 - d 1.5d (relative contact ratio x sum (PeR of friends /|F(friends)|)) + 0.5 d (Sum((SameInterest(friends) - 0.5) / |F(friends)|)) [44]
Penalty	0.5
Divergence	0.13
Mobile models	4
Dataset	SLAW

3.1.2 General Settings

1. Each experiment is run 20 times and the average is calculated.
2. Each run delivers 2 messages.
3. Every 20 runs are applied with a different density which are 50 users, 100 users, and 200 users.
4. Every 20 runs are applied with different interest distribution which are discrete uniform, discrete normal, and two disjoint subgraphs. They are called setting 1, setting 2 and, setting 3, respectively.

Setting 1 (Discrete Uniform Distribution of Interest)

In the discrete uniform distribution of interest, the users are spread equally between 11 categories with varying interest rates ranging from 0 to 1. Accordingly, the destination set establishes 18% of the nodes while the interested forwarders cover 36%.

TABLE 2: Experiment Parameters of Settings 1

Parameter	Value
Destination Ratio	0.18
Interested FWD Ratio	0.36
Choice of interest distribution	1 (Discrete Uniform Interest)

Setting 2 (Discrete Normal Distribution of Interest)

In the normal distribution of interest, the destination set embraces 2% of the community, the interested forwarders set comprises 48%, while the remaining 50% are uninterested nodes.

TABLE 3: Experiment Parameters of Settings 2

Parameter	Value
Destination Ratio	0.02
Interested FWD Ratio	0.48
Choice of interest distribution	2 (Normal Distribution of Interest)

Setting 3 (Two Disjoint Subgraphs Distribution of Interest)

In two disjoint subgraphs distribution of interest, the destination set embraces 2% of the community while the remaining 98% are uninterested nodes.

TABLE 4: Experiment Parameters of Settings 3

Parameter	Value
Interested FWD Ratio	0
Destination Ratio	0.2
Choice of interest distribution	3 (**Two Disjoint Sub Graphs Distribution of Interest**)

In this CHAPTER, we introduce multiple-hop awareness to interest-aware social-based forwarding algorithms that recognize destination nodes by their interest profile. We build upon IPeR and perform our experimentations using the new variant that uses multi-hops, namely MIPeR.

3.2 IPeR (Interest aware PeopleRank)

IPeR is an interest-aware social forwarding algorithm [44] that introduces interest awareness in ranking the mobile nodes besides the typical social ranking and activeness used in the social-based ranking PeopleRank algorithm (PeR) [43]. Each node carries a PeopleRank value, which ranks nodes as per their social popularity, and a vector of interest values for each of the exchanged messages that collectively constitute its IPeR value. To elaborate, when a group of nodes has a higher or equal interest and PeopleRank values than the current node, this group receives copies of the message [43]. On the other hand, to make nodes other than the source and destination nodes, participate in delivering a message; a copy of the message is directed to the nodes, which have an interest in its content. It is a sort of an incentive for these nodes to sacrifice part of their storage and power to participate in delivering a message to other nodes. IPeR reduces the delivery cost, and delay in comparison to that of Epidemic and PeopleRank algorithms. As illustrated in FIGURE 4, the message holder node, indicated as “MH”, has an IPeR value of 0.3. The IPeR value is conveyed as two values which are the PeopleRank value and Interest value. In addition, There is a damping factor (d) to balance the two values. The equation [111] is illustrated below

$$\boxed{\begin{aligned} IPeR(i) &= (1-d) + \\ &d * \sum_{j \in F(i)} \frac{CA - PeR(j) * w_{i,j} * PSInt(j, Ad)}{|F(j)|} \end{aligned}}$$

[111]

MH has three contacts at this unit of time. Based on IPeR, it sends a copy of the message to only one node which has a higher IPeR value, which is 0.5. The contacted node that gets a copy of the message is illustrated with a double-lined border in FIGURE 4.

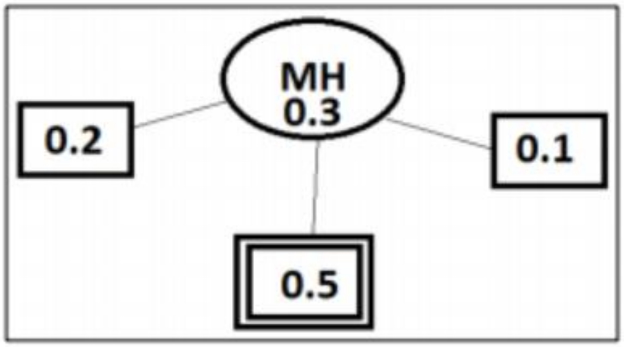


FIGURE 4: IPeR

3.3 Multiple Hops IPeR (2MIPeR)

As suggested by (Takasuka et al. 2018) [23], routing tables can be evolved to enhance the forwarding mechanism in OMSN to reduce network flooding. Creating routing tables reactively is more adequate for OMSNs. Reactive discovery is established on exchanging routing information when two nodes encounter. This way there are no extra control messages to be exchanged before sending data messages. Thus, the routing information that is measured here is the number of contacts with the encountered nodes, to cover 2 or 3 hop routing. MIPeR is using IPeR and accumulates its value by such routing tables. The routing information includes the node's PeopleRank and degree of interest in the message content combined in an IPeR value. The privacy of each node is maintained as each node calculates its value individually and they share a final value that is not reversible. Thus, other nodes cannot extract the private info of the node's profile.

Several experiments and settings are used to examine the performance of such an algorithm. The 2-hops experiments are illustrated in FIGURE 5.

2Hops	Average	Discrete Uniform	50 users
			100 users
			200 users
		Normal	50 users
			100 users
			200 users
		Two Disjoint Subgraphs	50 users
			100 users
			200 users
	Maximum	Discrete Uniform	50 users
			100 users
			200 users
		Normal	50 users
			100 users
			200 users
		Two Disjoint Subgraphs	50 users
			100 users
			200 users
	Top X% of Maximum	Discrete Uniform	50 users
			100 users
			200 users
		Normal	50 users
			100 users
			200 users
		Two Disjoint Subgraphs	50 users
			100 users
			200 users
	Harmonic Mean	Discrete Uniform	50 users
			100 users
			200 users
		Normal	50 users
			100 users
			200 users
		Two Disjoint Subgraphs	50 users
			100 users
			200 users

FIGURE 5: 2Hops-MIPeR Experiments

3.3.1 Steps of the algorithms (2-MIPeR)

2-hop experiments are applied to explore the effect of considering the number of contacts with the encountered nodes (2-MIPeR). We hypothesize that by having routing information that covers the next hop of the nodes we may have a better choice of relay nodes, which leads to better performance. Using harmonic means calculations, for example, of neighboring ranks, an extra piece of information is considered for the routing decision which is the number of current physical contacts with the encountered nodes. When two nodes meet, the IPeR rank of the encountered node is calculated based on the harmonic mean of its IPeR and IPeR values of its contacts. If the harmonic mean of IPeR values of the nodes currently in contact is higher than the IPeR of the node itself, the harmonic mean is considered as the new IPeR value of the node. Otherwise, the IPeR value of the node remains unchanged. Finally, the message holder node sends a copy of the message only if the harmonic IPeR value of the encountered node is higher than its value. As illustrated in FIGURE 6, the message holder node, indicated as “MH”, has an IPeR value of 0.3. MH has three contacts at this unit of time. Based on 2Har-MIPeR, it sends a copy of the message to two nodes which appear in the FIGURE as rectangles with a double border. The node that has an initial IPeR value of 0.2, when it meets the message holder node, it calculates a new IPeR value based on the harmonic mean of its IPeR value and the IPeR values of its four contacted nodes at this time slot, illustrated as diamonds. Its IPeR value is updated to be 0.5. Consequently, it gets a copy of the message.

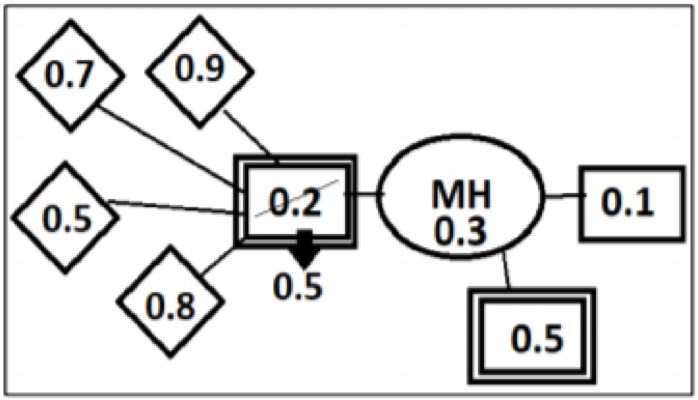
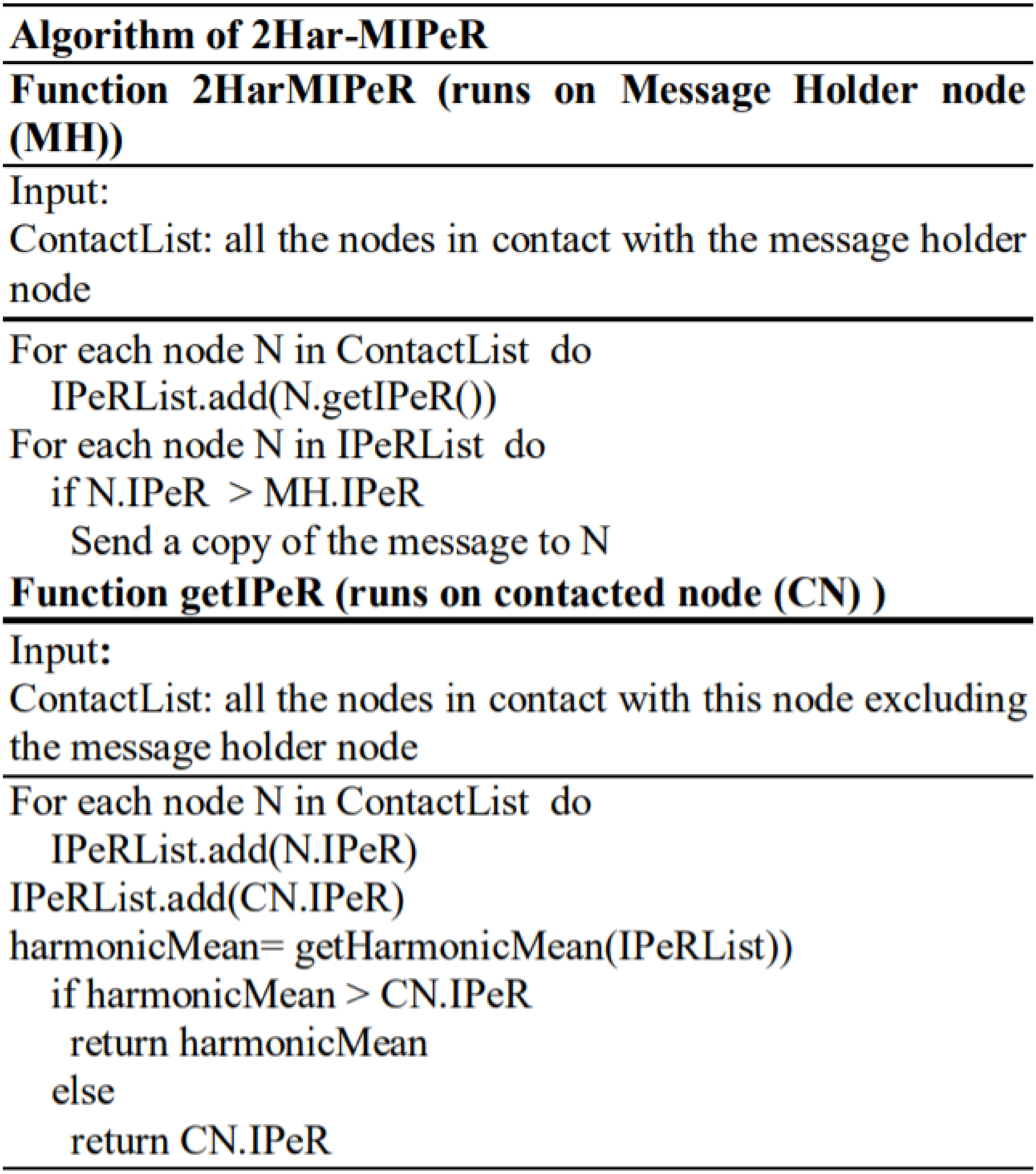


FIGURE 6: 2Har-MIPeR steps of algorithm

For 2MIPER, 6 experiments are different based on the statistical measure that is used. The statical measures that are used are average, maximum and harmonic mean. The experiments are 2Avg-MIPeR, 2Max-MIPeR, 2Har-MIPeR and 25%-2Max-MIPeR, 50%-2Max-MIPeR, and 75%-2Max-MIPeR. X%-2Max-MIPeR has three versions based on the percentage of the tolerance factor of the maximum value which are 25%, 50%, and 75%. To be more specific, for X%-2Max-MIPeR, the maximum value the IPeR values of the encountered nodes is calculated first, then a percentage of the maximum is considered by multiplying the maximum value by 25%, 50%, or 75%. Based on this, the message holder node sends a copy of the message to the nodes whose IPeR value is higher than the percentage of the maximum. The details of each experiment are in the following section.

3.3.2 2MIPeR Algorithm



Algorithm of 2Har-MIPeR

Function 2HarMIPeR (runs on Message Holder node (MH))

Input:

ContactList: all the nodes in contact with the message holder node

```
For each node N in ContactList  do
   IPeRList.add(N.getIPeR())
For each node N in IPeRList  do
   if N.IPeR  > MH.IPeR
     Send a copy of the message to N
```

Function getIPeR (runs on contacted node (CN))

Input**:**

ContactList: all the nodes in contact with this node excluding the message holder node

```
For each node N in ContactList  do
   IPeRList.add(N.IPeR)
IPeRList.add(CN.IPeR)
harmonicMean= getHarmonicMean(IPeRList))
   if harmonicMean > CN.IPeR
     return harmonicMean
   else
     return CN.IPeR
```

3.3.3 The experiments

3.3.3.1 Two Hops IPeR based on Harmonic Mean (2Har-MIPeR)

The harmonic mean is a sort of numerical average, which is calculated by dividing the number of observations by the reciprocal of each number in the series. Thus, it is the reciprocal of the arithmetic mean of the reciprocals. The harmonic mean is usually used with fractions like the values of PeopleRank and interest. This experiment is named, 2-hops based on Harmonic Mean Interest PeopleRank (2Har-MIPeR).

3.3.3.2 Two Hops IPeR based on Average (2Avg-MIPeR)

This experiment uses the same steps as 2Har-MIPeR. However, it uses average instead.

3.3.3.3 Two Hops IPeR based on Maximum (2Max-MIPeR)

The previous experiment is founded on calculating the average. Nevertheless, the average may miss a high PeopleRank or interest as only one value of the numerous values included in the calculation is too low or too high and affected the average. In this experiment, the contacts of the encountered nodes are examined in choosing the rely nodes. This time, the maximum is the basic measurement criterion, which is named 2-hops based on Maximum Interest PeopleRank (2Max-MIPeR).

3.3.3.4 Two Hops IPeR based on Percentage of Maximum (X%-2Max-MIPeR)

This time, the maximum is the basic measurement criteria. However, a percentage of the maximum is calculated and becomes the criteria for forwarding the message. When two nodes meet, the PeopleRank of the encountered node is calculated based on the maximum of its PeopleRank and PeopleRank values of its contacts as well as its interest. Then, the product of the maximum and a percentage is the new PeopleRank to be used in the comparison. The message holder sends a copy of the message only if the PeopleRank of the encountered node is higher than its own PeopleRank and its interest is higher as well. The parameters that can have different values is x% of 2Max-MIPeR which is considering a percentage of Max-MIPeR

1. It includes all the nodes which have IPeR more than 25% of the new calculated 2Max-MIPeR of the message holder node.

2. It includes all the nodes which have IPeR more than 50% of the new calculated 2Max-MIPeR of the message holder node.

3. It includes all the nodes which have IPeR more than 75% of the new calculated 2Max-MIPeR of the message holder node.

3.3.4 Results

Setting 1

50 Users

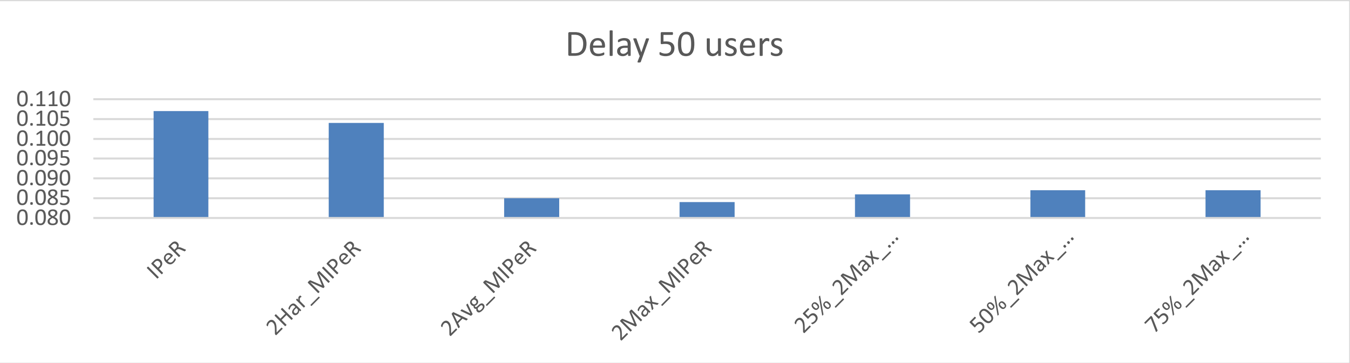


FIGURE 7: Delay of Setting 1, 2MIPeR, 50 Users

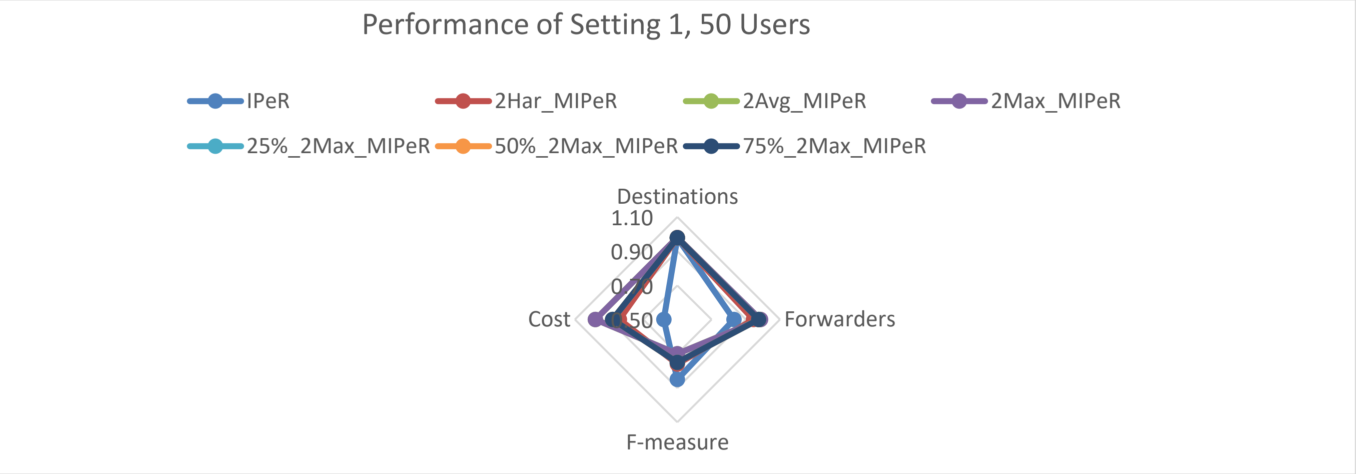


FIGURE 8: Delivery, F-measure & Cost of Setting 1, 2MIPeR, 50 Users

TABLE 5: Performance of Setting 1, 2MIPeR, 50Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.60	29.90	0.107	0.85	29
2Har_MIPeR	17.60	34.00	0.104	0.76	42
2Avg_MIPeR	17.65	35.45	0.085	0.70	49
2Max_MIPeR	17.65	35.50	0.084	0.70	49
25%_2Max_MIPeR	17.60	35.10	0.086	0.75	44
50%_2Max_MIPeR	17.60	35.10	0.087	0.75	44
75%_2Max_MIPeR	17.60	35.10	0.087	0.75	44

100 Users

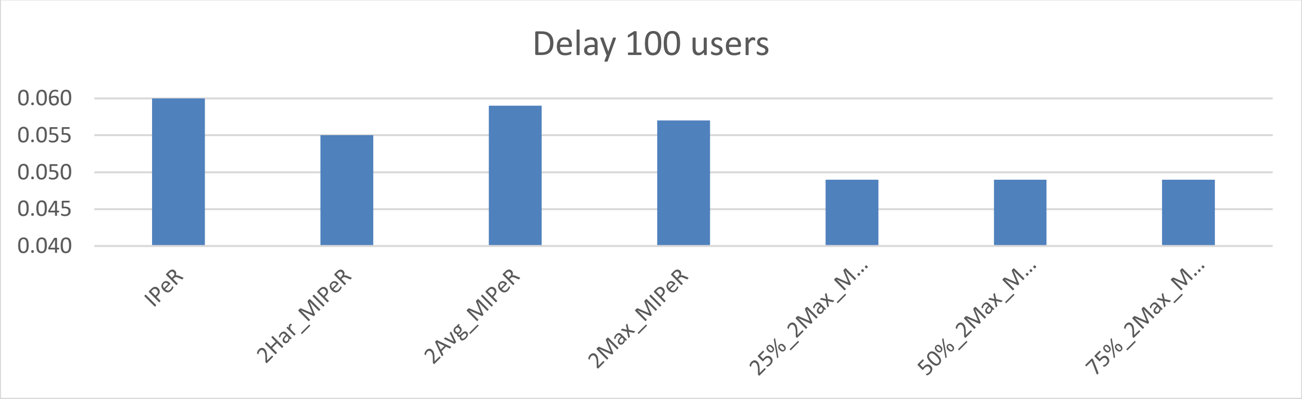


FIGURE 9: Delay of Setting 1, 2MIPeR, 100 Users

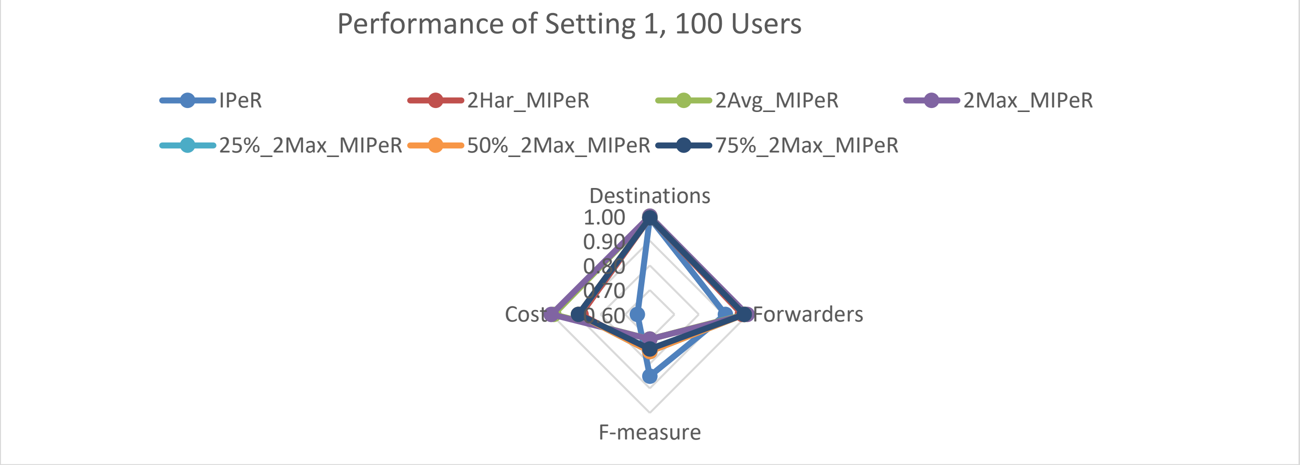


FIGURE 10: Delivery, F-measure & Cost of Setting 1, 2MIPeR, 100 Users

TABLE 6: Performance of Setting 1, 2MIPeR, 100Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.88	32.68	0.060	0.85	65
2Har_MIPeR	17.88	35.13	0.055	0.75	88
2Avg_MIPeR	18.00	35.53	0.059	0.70	99
2Max_MIPeR	18.00	35.83	0.057	0.70	100
25%_2Max_MIPeR	17.88	35.43	0.049	0.75	89
50%_2Max_MIPeR	17.88	35.43	0.049	0.75	89
75%_2Max_MIPeR	17.88	35.43	0.049	0.74	89

<u>200 Users</u>

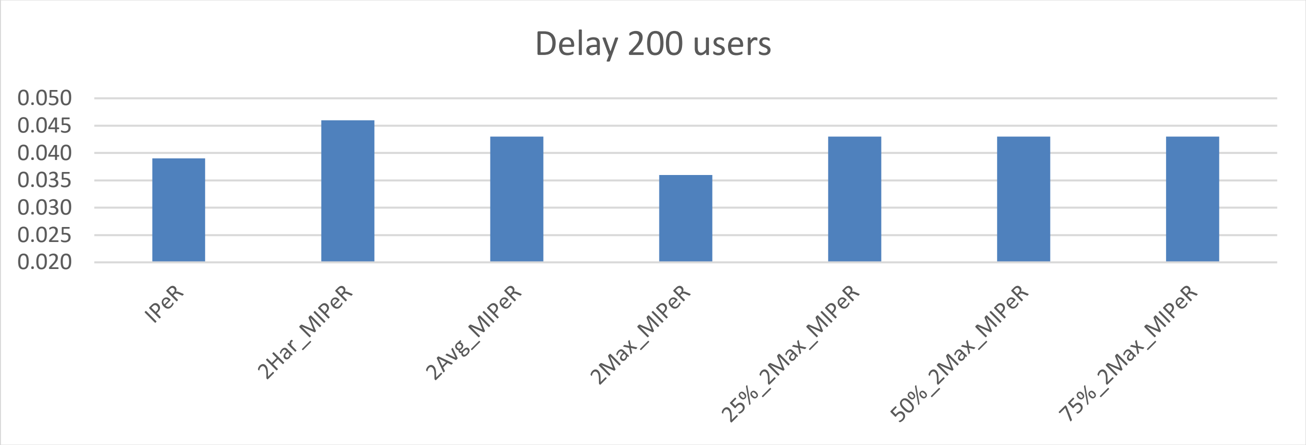


FIGURE 11: Delay of Setting 1, 2MIPeR, 200 Users

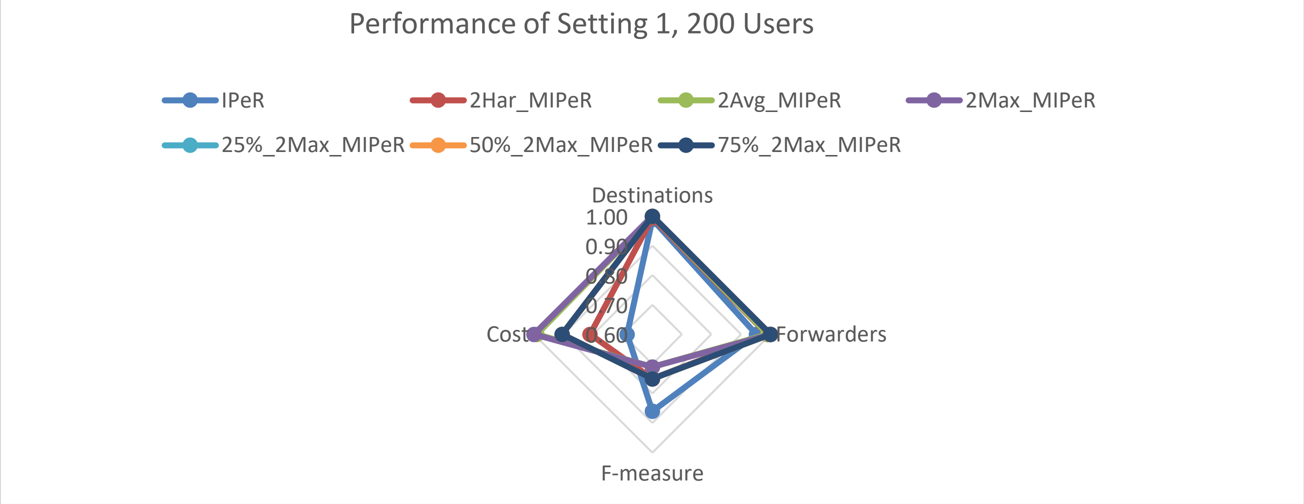


FIGURE 12: Delivery, F-measure & Cost of Setting 1, 2MIPeR, 200 Users

TABLE 7: Performance of Setting 1, 2MIPeR, 200Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.80	34.74	0.039	0.86	137
2Har_MIPeR	17.90	36.08	0.046	0.75	162
2Avg_MIPeR	17.99	36.14	0.043	0.71	198
2Max_MIPeR	18.00	36.49	0.036	0.71	200
25%_2Max_MIPeR	18.00	36.49	0.043	0.75	181
50%_2Max_MIPeR	18.00	36.49	0.043	0.75	181
75%_2Max_MIPeR	18.00	36.49	0.043	0.75	181

Discussion of Results

The three variations of X%-2Max-MIPeR almost act the same, so they are treated as one algorithm for all the variant users' densities.

For 50 users, 2Avg-MIPeR and **2Max-MIPeR** reached **0.3% more destination nodes**, while **X%-2Max-MIPeR and 2Har-MIPeR** reached the same number of destination nodes compared to IPeR.

For the number of reached **interested forwarders all the proposed algorithms performed better than IPeR. 2Avg-MIPeR** and **2Max-MIPeR reached 19%, X%-2Max-MIPeR reached 17 % and 2Har-MIPeR** reached 14% more interested forwarders in comparison to IPeR.

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **2Max-MIPeR** and **X%-2Max-MIPeR** reduced delay by **27%. 2Avg-MIPeR reduced it by 18% and 2Har-MIPeR diminished it by 9%.**

All the proposed algorithms perform worse in terms of cost and F-measure. The reason is that to perform better in delay they need to contact a larger number of uninterested forwarders which decreases the F-measure and increases cost. The best of them is **2Har-MIPeR, which diminished F-measure by 9%. Then, X% of 2Max-MIPeR decreased it by 12%. While F-measure decreased by 18%** related to **2Avg-MIPeR** and **2Max-MIPeR**.

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR, which increased it by 45%. Then, X% of 2Max-MIPeR increased by 52%. 2Avg-MIPeR** and **2Max-MIPeR increased cost by 69%.**

For 100 users, **2Avg-MIPeR** and **2Max-MIPeR** reached **0.7% more destination nodes**, while **X%-2Max-MIPeR and 2Har-MIPeR** reached the same number of destination nodes compared to IPeR.

For the number of reached **interested forwarders all the proposed algorithms performed better than IPeR. 2Max-MIPeR, 2Avg-MIPeR, X%-2Max-MIPeR and 2Har-MIPeR** reached **10%, 9%, 8% and 7%** more interested forwarders in comparison to IPeR, respectively.

For **delay, all the proposed algorithms slightly enhanced** it in comparison to IPeR. **The best algorithm** was **X%-2Max-MIPeR** which reduced delay by **17%.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them is **2Har-MIPeR** and **X%-2Max-MIPeR,** which diminished **F-measure by 12%.** Then**, 2Max-MIPeR** and **2Avg-MIPeR** decreased it by **18%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased **it by 35%. X%-2Max-MIPeR, 2Avg-MIPeR and 2Max-MIPeR** increased it by **37%, 52%** and **54%** compared to IPeR, respectively.

For 200 users, **all the proposed algorithms** reached **1% more destination nodes** compared to IPeR. For the number of reached **interested forwarders all the proposed algorithms performed better than IPeR. 2Max-MIPeR** and **X%-2Max-MIPeR** reached **5%,** while **2Har-MIPeR and 2Avg-MIPeR** reached **4%** more interested forwarders in comparison to IPeR.

For **delay, all the proposed algorithms slightly enhanced** it in comparison to IPeR. However**, 2Har-MIPeR** which increased delay by **25%.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them is **2Har-MIPeR** and **X%-2Max-MIPeR,** which diminished **F-measure by 13%.** Then**, 2Max-MIPeR** and **2Avg-MIPeR** decreased it by **17%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased **it by 18%.** Then, **X% of 2Max-MIPeR** increased it by **32%. 2Avg-MIPeR** increased it by **45%** and **2Max-MIPeR** increased cost by **46%** compared to IPeR.

For more details see appendix A.

Conclusion of Setting 1

In conclusion, for 2Avg-MIPeR, 2Max-MIPeR, and X%-2Max-MIPeR are performing insignificantly better compared to IPeR in the delivery ratio, the number of reached forwarders, and delay. However, they almost duplicate costs and decrease F-measure. ***2Har-MIPeR performs the best among the proposed algorithms as it minimized the increase of cost and decrease of F-measure while enhancing the number of reached interested forwarders in comparison to IPeR. In addition, it enhances delay in a low user***

density environment compared to IPeR. The main concern of these algorithms is that they are giving copies of the message of uninterested forwarders. They can be used in situations in which we sacrifice the cost to gain enhanced delay.

For the different densities of users using 2Har-MIPeR, 200 users' density has the best results in terms of delivery ratio, the number of reached interested forwarders, delay, F-measure, and cost. It is illustrated in theTABLE 8.

TABLE 8: Performance of 2Har-MIPeR for Setting 1

Algorithm	Destinations	Forwarders	F-measure	Cost	Delay
50 Users	**0.98**	**0.94**	**0.76**	**0.84**	**0.104**
100 Users	**0.99**	**0.98**	**0.75**	**0.88**	**0.055**
200 Users	**0.99**	**0.99**	**0.75**	**0.81**	**0.046**

This reveals that using 2Har-MIPeR at the discrete uniform distribution of interest, the denser the environment is the more delivery ratio, the more interested forwarders, the more F-measure, the less delay, and the less cost exerted by the algorithm.

Setting 2

50 Users

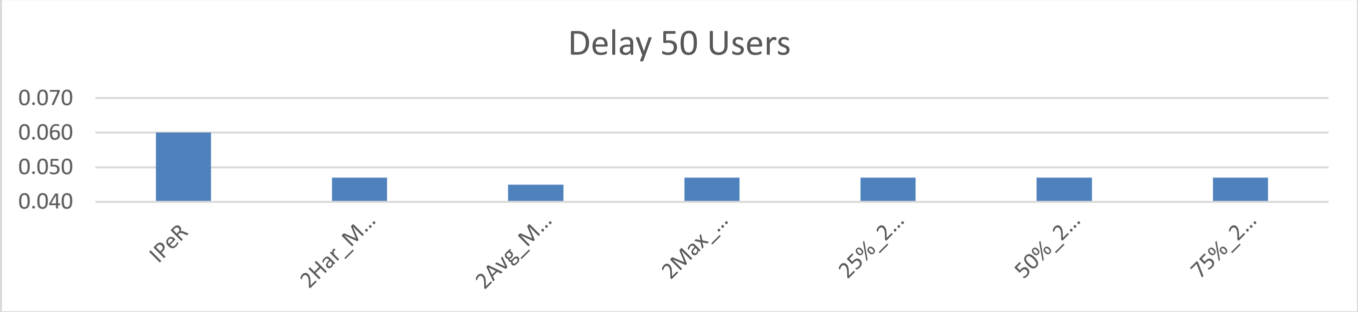


FIGURE 13: Delay of Setting 2, 2-MIPeR, 50 Users

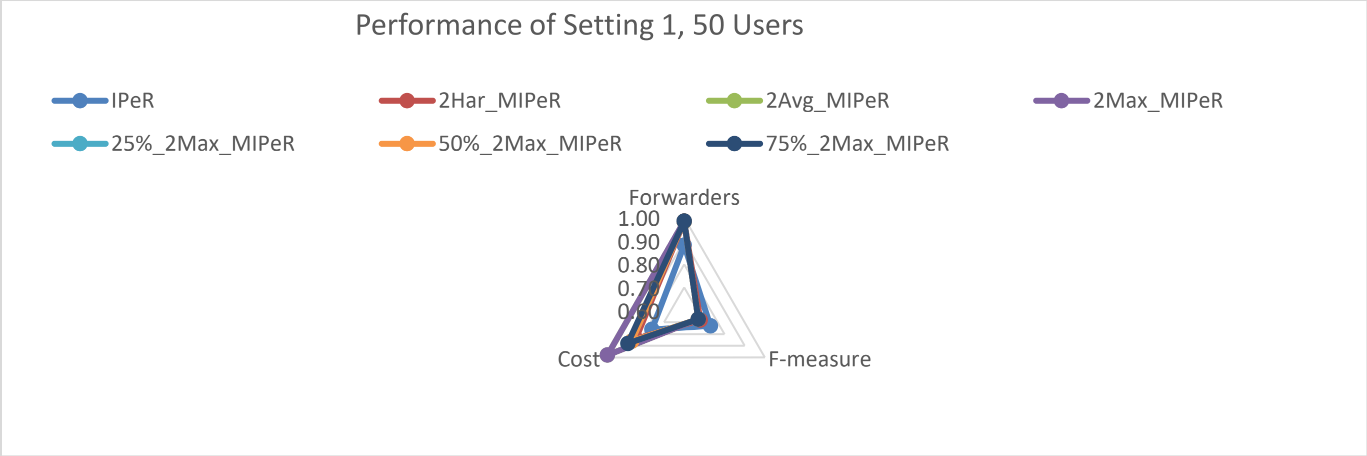


FIGURE 14: Delivery, F-measure & Cost of Setting 2, 2MIPeR, 50 Users

TABLE 9: Performance of Setting 2, 2MIPeR, 50Users

Algorithm	Forwarders	Delay	F-measure	Cost
IPeR	43.20	0.060	0.73	38
2Har_MIPeR	48.20	0.047	0.68	42
2Avg_MIPeR	48.20	0.045	0.67	49
2Max_MIPeR	48.25	0.047	0.67	49
25%_2Max_MIPeR	48.25	0.047	0.67	43
50%_2Max_MIPeR	48.25	0.047	0.67	43
75%_2Max_MIPeR	48.25	0.047	0.67	44

100 Users

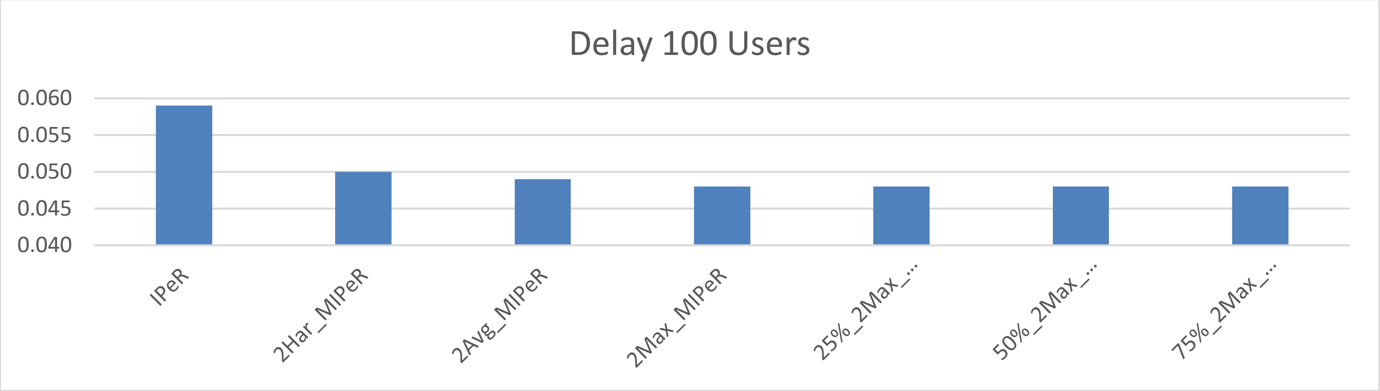


FIGURE 15: Delay of Setting 2, 2MIPeR, 100 Users

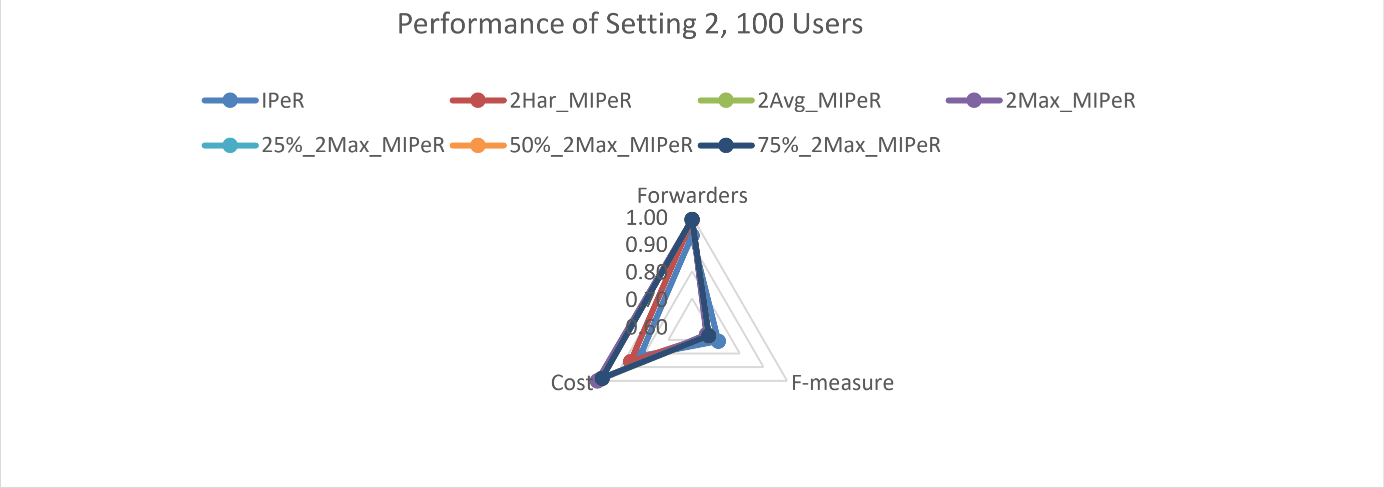


FIGURE 16: Delivery, F-measure & Cost of Setting 2, 2MIPeR, 100 Users

TABLE 10: Performance of Setting 2, 2MIPeR, 100Users

Algorithm	Forwarders	Delay	F-measure	Cost
IPeR	45.58	0.059	0.71	82
2Har_MIPeR	48.33	0.050	0.67	86
2Avg_MIPeR	48.38	0.049	0.67	99
2Max_MIPeR	48.48	0.048	0.66	100
25%_2Max_MIPeR	48.48	0.048	0.67	98
50%_2Max_MIPeR	48.48	0.048	0.67	98
75%_2Max_MIPeR	48.48	0.048	0.67	98

200 Users

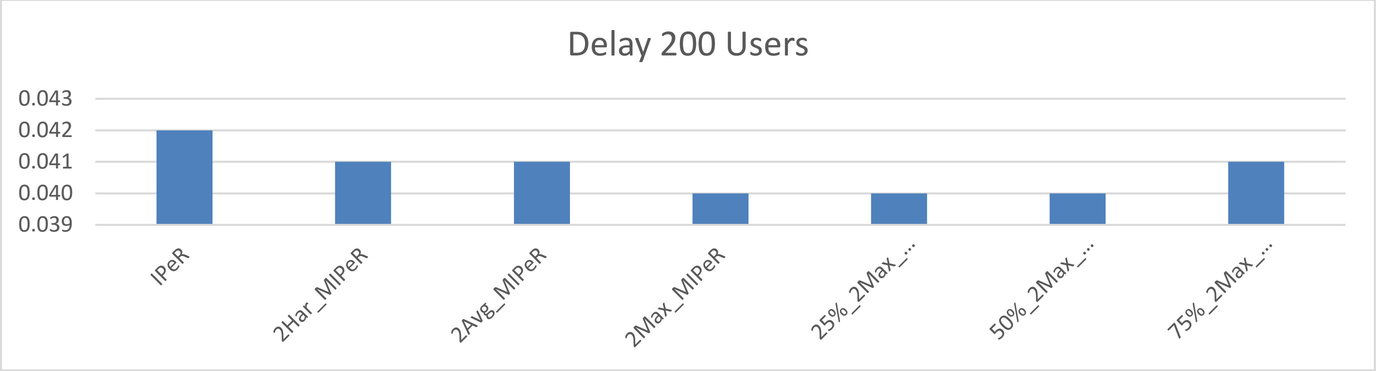


FIGURE 17: Delay of Setting 2, 2-MIPeR, 200 Users

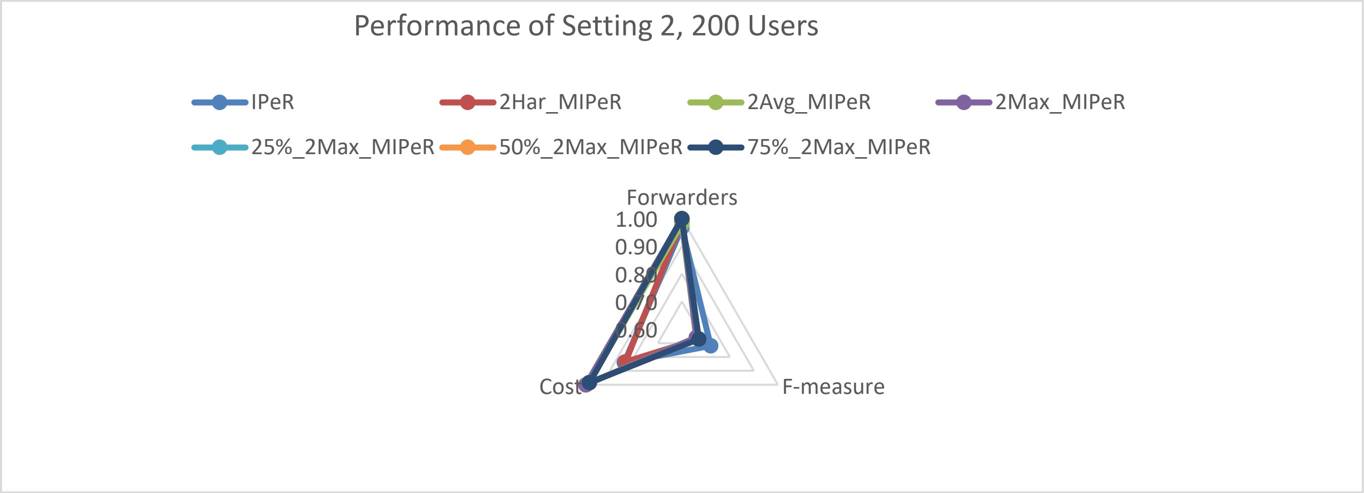


FIGURE 18: Delivery, F-measure & Cost of Setting 2, 2MIPeR, 200 Users

TABLE 11: Performance of Setting 2, 2MIPeR, 200Users

Algorithm	Forwarders	Delay	F-measure	Cost
IPeR	47.36	0.042	0.72	168
2Har_MIPeR	48.44	0.041	0.67	167
2Avg_MIPeR	48.00	0.041	0.66	198
2Max_MIPeR	48.95	0.040	0.66	200
25%_2Max_MIPeR	48.95	0.040	0.67	197
50%_2Max_MIPeR	48.95	0.040	0.67	197
75%_2Max_MIPeR	48.95	0.041	0.67	197

Discussion of Results

The three variations of X%-2Max-MIPeR almost act the same, so they are treated as one algorithm for all the variant users' densities.

For 50 users, for the number of reached **interested forwarders, all the proposed algorithms performed better than IPeR by 12%.**

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **2Avg-MIPeR is the best and** reduced delay by **25%. The rest of the proposed algorithm diminished it by 22%.**

All the proposed algorithms perform worse in terms of cost and F-measure. The reason is that to perform better in delay they need to contact a larger number of uninterested forwarders which decreases the F-measure and increases cost. The best of them is **2Har-MIPeR, which diminished F-measure by 7%. Then, the rest of the algorithms reduced F-measure by 8%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR, which increased it by 147%. Then, X%-2Max-MIPeR increased it by 153%. 2Max-MIPeR and 2Avg-MIPeR increased cost by 188%.**

For 100 users, for the number of reached **interested forwarders, all the proposed algorithms performed better than IPeR by 6%.**

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **X%-2Max-MIPeR and 2Max-MIPeR are the best and** reduced delays by **19%. 2Avg-MIPeR diminished it by 17%. 2Har--MIPeR enhanced it by 15%**

All the proposed algorithms perform worse in terms of cost and F-measure. **2Har-MIPeR, X%-2Max-MIPeR, and 2Avg-MIPeR diminished F-measure by 6%. 2Max-MIPeR reduced F-measure by 7%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased it by **5%. X%-2Max-MIPeR, 2Avg-MIPeR and 2Max-MIPeR** increased it by **20%**, **21% and 22%, respectively.**

For 200 users, **all the proposed algorithms** reached more **interested forwarders** compared to IPeR. **2Max-MIPeR** and **X%-2Max-MIPeR** reached **3%, 2Har-MIPeR reached 2%** and 2**Avg-MIPeR** reached **1%** more destination nodes.

For **delay, all the proposed algorithms performed better than IPeR. 2Max-MIPeR** and **X%-2Max-MIPeR** decreased it by **5%,** while **2Har-MIPeR and 2Avg-MIPeR** declined it by **2%** in comparison to IPeR.

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them is **2Har-MIPeR** and **X%-2Max-MIPeR,** which diminished **F-measure by 7%.** Then**, 2Max-MIPeR** and **2Avg-MIPeR** decreased it by **8%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased **it by 1%.** Then, **X%-2Max-MIPeR, 2Avg-MIPeR and 2Max-MIPeR** increased it by **17%, 18%** and **19%, respectively** compared to IPeR.

Conclusion of Setting 2

All the proposed algorithms are performing better compared to IPeR in the number of reached interested forwarders, and delay. However, they increased cost and decreased F-measure. ***2Har-MIPeR performs the best among the proposed algorithms as it minimizes the increase of cost and decrease of F-measure. For the different densities of users using 2Har-MIPeR, 200 users' density is the best in terms of delay and cost, while 50 users' density is the best in terms of F-measure. This is illustrated in TABLE 12.***

TABLE 12: Performance of 2Har-MIPeR for setting 2

Algorithm	Forwarders	F-measure	Cost	Delay
50 Users	0.98	0.68	0.84	0.047
100 Users	0.99	0.67	0.86	0.050
200 Users	0.99	0.67	0.84	0.041

This reveals that using 2Har-MIPeR in normal distribution of interest setting, the denser the environment is, the less delay, the less cost but the less F-measure exerted by the algorithm.

Setting 3

50 Users

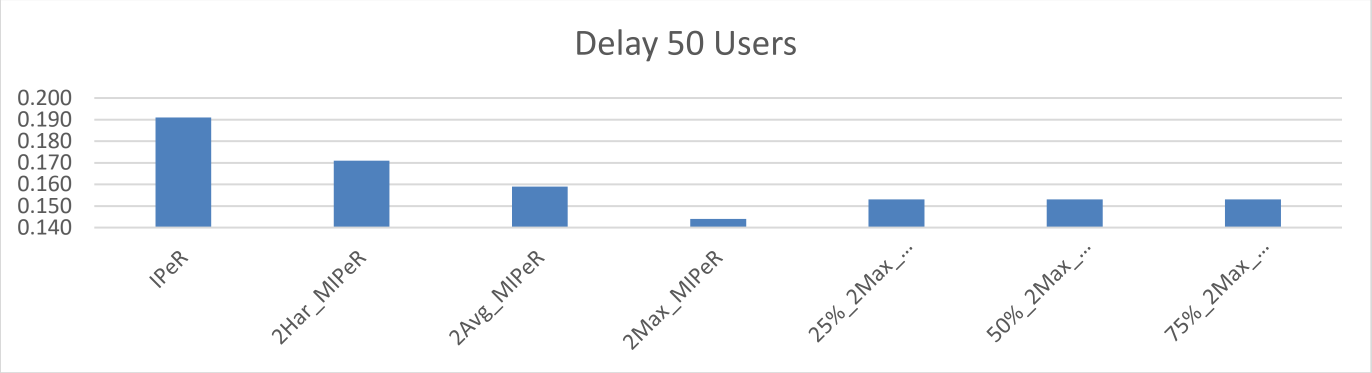


FIGURE 19: Delay of Setting 3, 2MIPeR, 50 Users

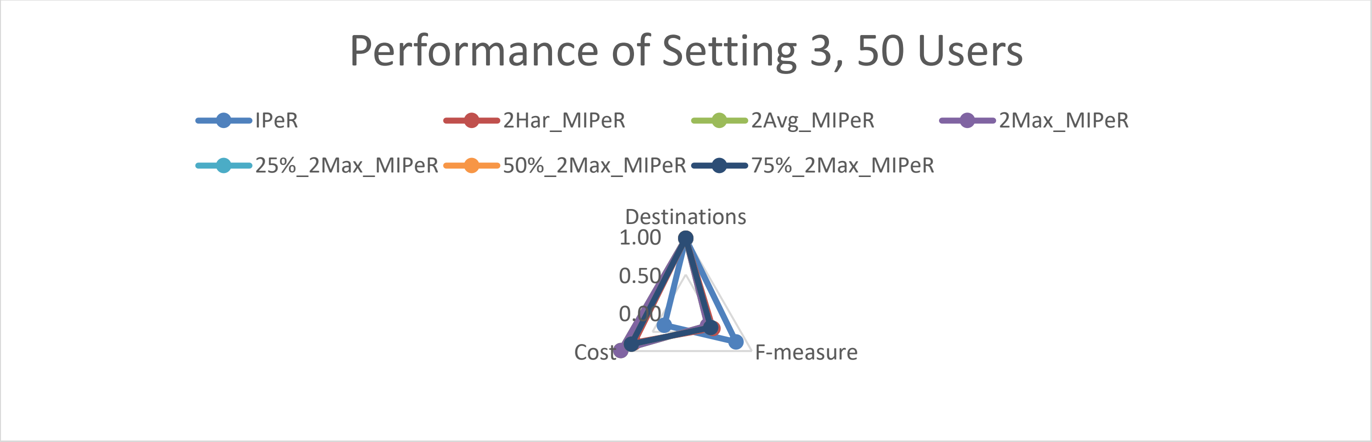


FIGURE 20: Delivery, F-measure & Cost of Setting 3, 2MIPeR, 50 Users

TABLE 13: Performance of Setting 3, 2MIPeR, 50Users

Algorithm	Destinations	Delay	F-measure	Cost
IPeR	19.40	0.191	0.76	16
2Har_MIPeR	19.50	0.171	0.41	39
2Avg_MIPeR	19.50	0.159	0.33	49
2Max_MIPeR	19.50	0.144	0.33	49
25%_2Max_MIPeR	19.50	0.153	0.38	41
50%_2Max_MIPeR	19.50	0.153	0.38	41
75%_2Max_MIPeR	19.50	0.153	0.38	41

100 users

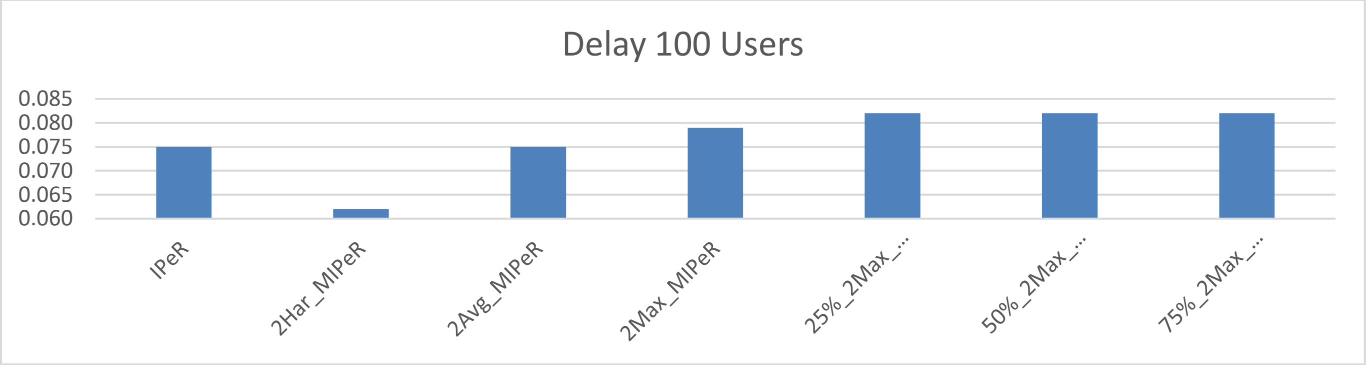


FIGURE 21: Delay of Setting 3, 2MIPeR, 100 Users

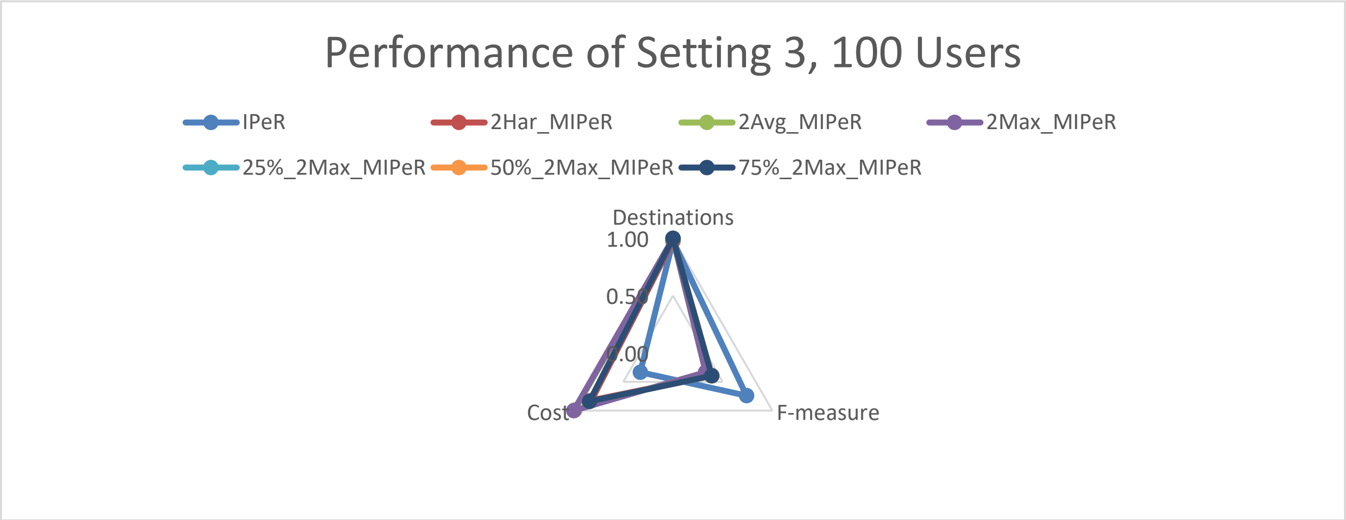


FIGURE 22: Delivery, F-measure and Cost of Setting 3, 2MIPeR, 100 Users

TABLE 14: Performance of Setting 3, 2MIPeR, 100Users

Algorithm	Destinations	Delay	F-measure	Cost
IPeR	19.55	0.075	0.74	33
2Har_MIPeR	19.55	0.062	0.39	82
2Avg_MIPeR	19.88	0.075	0.33	99
2Max_MIPeR	19.98	0.079	0.33	100
25%_2Max_MIPeR	19.98	0.082	0.39	84
50%_2Max_MIPeR	19.98	0.082	0.39	84
75%_2Max_MIPeR	19.98	0.082	0.39	84

200 users

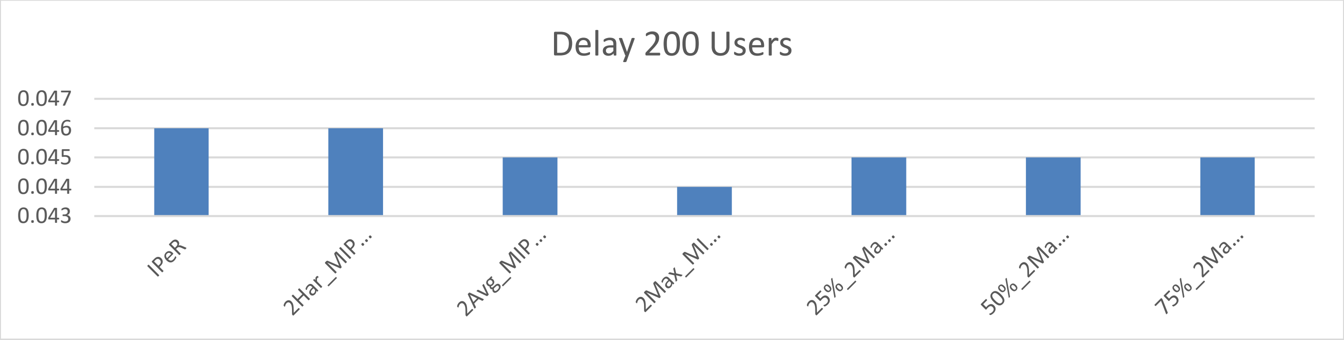


FIGURE 23: Delay of Setting 3, 2MIPeR, 200 Users

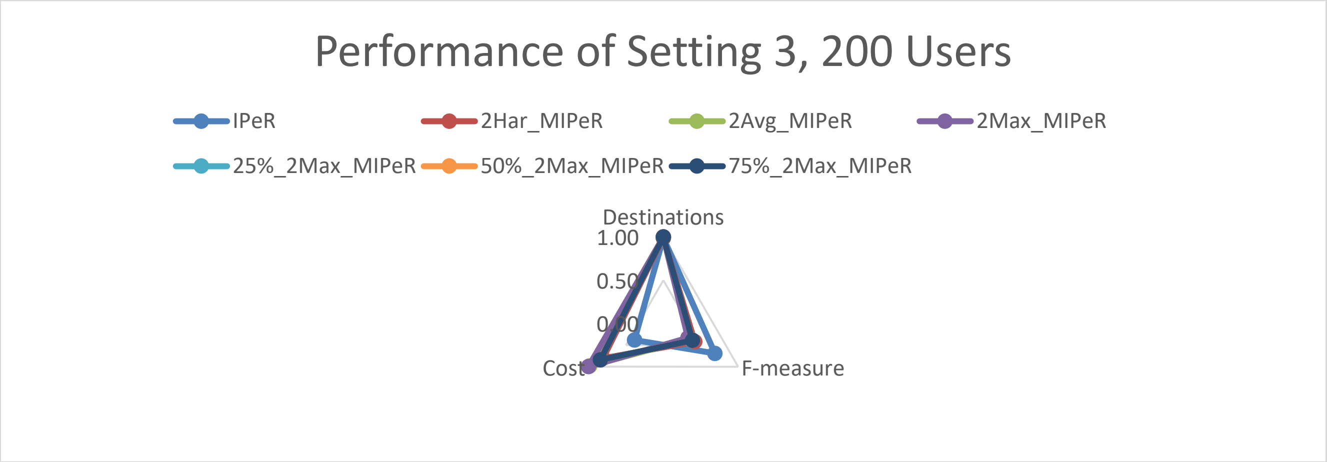


FIGURE *24*: Delivery, F-measure & Cost of Setting 3, 2MIPeR, 200 Users

TABLE 15: Performance of Setting 3, 2MIPeR, 200Users

Algorithm	Destinations	Delay	F-measure	Cost
IPeR	19.80	0.046	0.69	77
2Har_MIPeR	19.80	0.046	0.42	161
2Avg_MIPeR	19.99	0.045	0.36	196
2Max_MIPeR	19.99	0.044	0.33	200
25%_2Max_MIPeR	19.99	0.045	0.39	168
50%_2Max_MIPeR	19.99	0.045	0.39	168
75%_2Max_MIPeR	19.99	0.045	0.39	168

Discussion of Results

The three variations of X%-2Max-MIPeR almost act the same, so they are treated as one algorithm for all the variant users' densities.

For 50 users, for the number of reached **destinations, all the proposed algorithms performed better than IPeR by 1%.**

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **2Max-MIPeR is the best and** reduced delay by **25%. X%-2Max-MIPeR** reduced it by **20%. 2Avg-MIPeR** decreased it by **16%. Finally, 2Har-MIPeR** declined it by **10%**

All the proposed algorithms perform worse in terms of cost and F-measure. The best of them is **2Har-MIPeR, which diminished F-measure by 46%. X%-2Max-MIPeR reduced it by 50%. 2Avg-MIPeR** and **2Max-MIPeR decreased it by 57%**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased it by **144%. Then, X%-2Max-MIPeR increased it by 156%. 2Max-MIPeR and 2Avg-MIPeR increased cost by 206%.**

For 100 users, for the number of reached **destinations, all the proposed algorithms performed better than IPeR by 2% except 2Har-MIPeR reached an equal number of destination nodes compared to IPeR.**

For **delay, 2Har-MIPeR is the only algorithm that enhanced delay by 17%. 2Avg-MIPeR** performed equal to IPeR. **X%-2Max-MIPeR and 2Max-MIPeR increased delay by 5% and 9%. respectively.**

All the proposed algorithms perform worse in terms of cost and F-measure. The best of them is **2Har-MIPeR and X%-2Max-MIPeR which diminished F-measure by 47%. 2Avg-MIPeR** and **2Max-MIPeR decreased it by 55%**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased it by **148%. Then, X%-2Max-MIPeR increased it by 155%. 2Max-MIPeR and 2Avg-MIPeR increased cost by 203% and 200%, respectively.**

For 200 users, for the number of reached **destinations, all the proposed algorithms performed better than IPeR by 1% except 2Har-MIPeR reached an equal number of destination nodes compared to IPeR.**

For **delay, 2Max-MIPeR, X%-2Max-MIPeR** and **2Avg-MIPeR** enhanced delay by **4%, 2%** and **2%, respectively. 2Har-MIPeR performed equal to IPeR.**

All the proposed algorithms perform worse in terms of cost and F-measure. The best of them is **2Har-MIPeR which diminished F-measure by 39%. X%-2Max-MIPeR, 2Avg-MIPeR** and **2Max-MIPeR decreased it by 43%, 48% and 52%, respectively.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased it by **109%. Then, X%-2Max-MIPeR increased it by 118%. 2Avg-MIPeR and 2Max-MIPeR increased cost by 155% and 160%, respectively.**

For more details see appendix A.

Conclusion of Setting 3

In conclusion, in this challenging environment, all the proposed algorithms performed trivially better compared to IPeR in the delivery ratio, and delay. However, it drastically increases the cost and decreases the F-measure. ***2Har-MIPeR performed the best in terms of cost and F-measure.***

TABLE 16: Performance of 2Har-MIPeR Setting 3

Algorithm	Destinations	F-measure	Cost	Delay
50 Users	0.98	0.41	0.78	0.171
100 Users	0.98	0.39	0.82	0.062
200 Users	0.99	0.42	0.81	0.046

2Har-MIPeR performed better for 200 users in terms of F-measure, delay, and delivery ratio. This reveals that the denser the environment is, less delay, more delivery ratio, and the more f-measure exerted by the algorithm. This result is elucidated in theTABLE above.

Conclusion on 2MIPeR

For all different 2MIPeR experiments, 2Har-MIPeR is the best-proposed algorithm in terms of F-measure and cost. In addition, it performed better in terms of delivery ratio and contacted interested forwarders ratio compared to IPeR. However, all the proposed algorithms increased cost radically and diminished F-measure compared to IPeR.

For the uniform distribution, 2Har-MIPeR performs better than IPeR in terms of delivery ratio (up to 1% for 200 users) and contacted interested forwards (up to 14% for 50 users). In addition, it enhances the delay (up to 9% for 50 users). For the different densities of users, the denser the environment is, the more delivery ratio is, and the more contacted interested forwarders would be. For sparse environments, 2Har-MIPeR enhances delay. The reason is that the algorithm reaches more destination nodes and interested forwarders node compared to IPeR. It is noticed in the 200 users' experiment that it reaches these extra nodes in the last few seconds of the simulation, which increases the average delay.

For the normal distribution, the delivery ratio is ignored as it is equal in all algorithms. 2Har-MIPeR performs better than IPeR in terms of contacted interested forwards (up to 12% for 50 users). In addition, it enhances the delay (up to 15% for 100 users). The denser the environment is, the less cost and delay are.

For two disjoint subgraphs distribution, which is a challenging environment where there are no interested forwarders, 2Har-MIPeR performs better than IPeR in terms of delay (up to 10%) while performing equally in delivery ratio. The denser the environment is, the shorter delay, the more the delivery ratio and the more F-measure achieved by the algorithm.

Overall, 2HarMIPeR performed the best in the uniform distribution in terms of F-measure, the best in the normal distribution in terms of delay, and performed equally in terms of cost for uniform distribution and two disjoint subgraphs. This is illustrated in FIGURE 25.

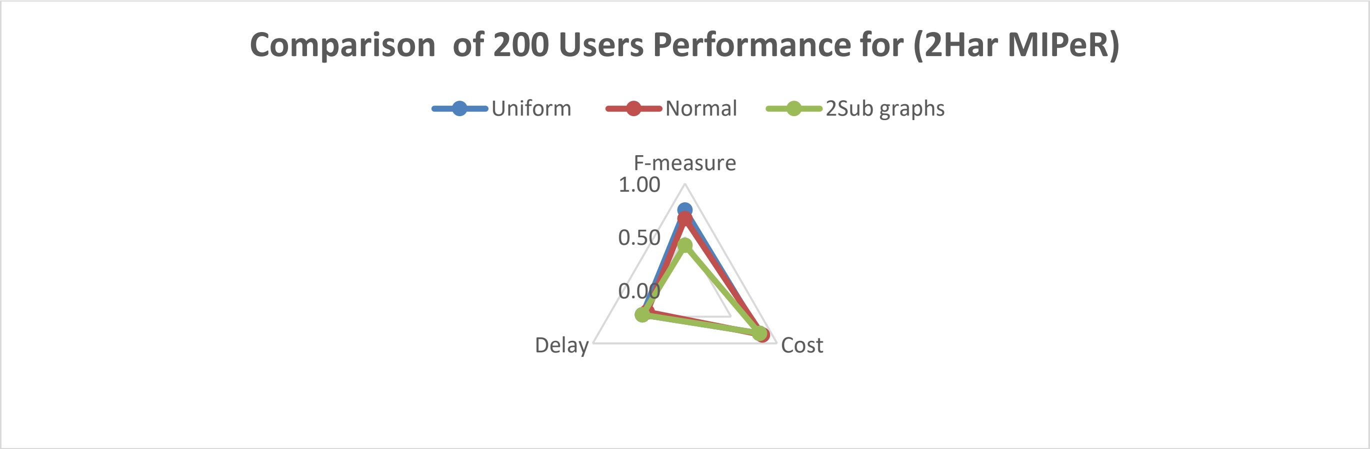


FIGURE 25: 2Har-MIPeR 200 users among all setting

3.4 Multiple Hops IPeR (3MIPeR)

For 3 hops MIPeR many experiments with various settings are implemented as illustrated in FIGURE 26

3Hops	Average	District Uniform	50 users
			100 users
			200 users
		Standard Normal	50 users
			100 users
			200 users
		Two Disjoint Sub Graphs	50 users
			100 users
			200 users
	Maximum	District Uniform	50 users
			100 users
			200 users
		Standard Normal	50 users
			100 users
			200 users
		Two Disjoint Sub Graphs	50 users
			100 users
			200 users
	Harmonic Mean	District Uniform	50 users
			100 users
			200 users
		Standard Normal	50 users
			100 users
			200 users
		Two Disjoint Sub Graphs	50 users
			100 users
			200 users

FIGURE 26: 3Hops MIPeR Experiments

3.4.1 Steps of the algorithms (3-MIPeR)

We experimented with 3 hops by using the contacts and the contacts of the contacts of the encountered nodes. For example, in 3Har-MIPeR, the IPeR of a node at a specific time is the harmonic mean of its own IPeR and IPeR values of all its contacted nodes in addition to their direct contacts now. The aim is to rank the nodes based on 3 hops, to determine the best node to forward the message to. As illustrated in FIGURE 27, the message holder node has an IPeR value of 0.3. MH sends a copy of the message to two nodes which appear in FIGURE 27 as rectangles with a double border.

The node that has an initial IPeR value of 0.2, when it meets the message holder node, it calculates a new IPeR value based on the harmonic mean of its IPeR value and the IPeR values of its four contacted nodes at this time slot, illustrated as diamonds. Before that, each node, which has the diamond shape, calculates its new IPeR value based on its own current contacts. So, the node which has an initial IPeR value of 0.7, updated its IPeR to be 0.4 based on the three contacted nodes illustrated as pentagons. Finally, the node whose initial value of 0.2 updates its IPeR to be 0.4. Consequently, it gets a copy of the message.

The privacy of each node is maintained as each node calculates its value individually and they share a final value that is not reversible. Thus, other nodes cannot extract the private info of the node’s profile.

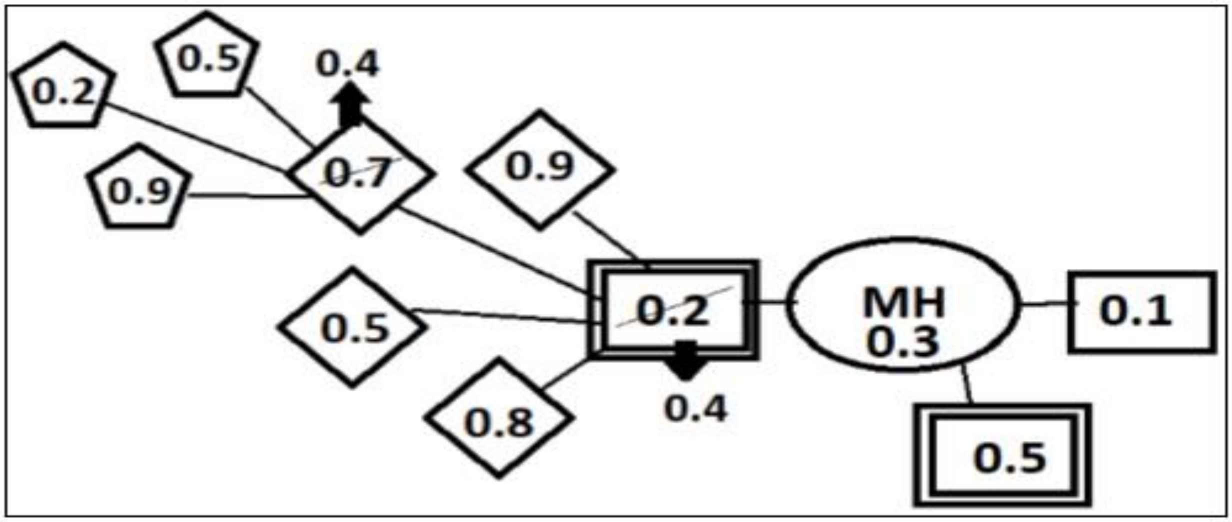
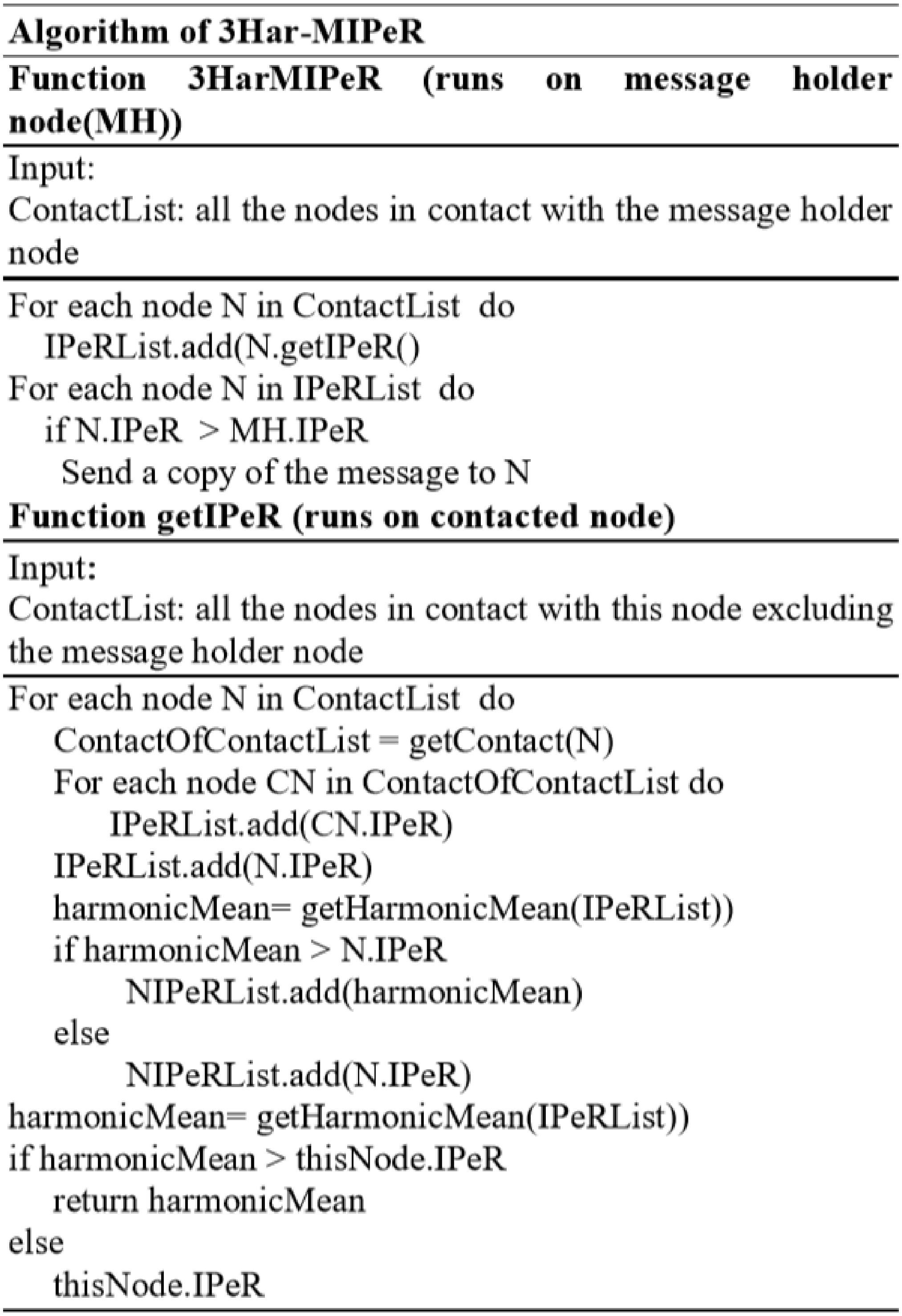


FIGURE 27: 3 Har-MIPeR

3.4.2 3MIPeR Algorithm

TABLE 17: Pseudo code of 3Har-MIPeR



Algorithm of 3Har-MIPeR

Function 3HarMIPeR (runs on message holder node(MH))

Input:
ContactList: all the nodes in contact with the message holder node

```
For each node N in ContactList  do
   IPeRList.add(N.getIPeR()
For each node N in IPeRList  do
   if N.IPeR  > MH.IPeR
     Send a copy of the message to N
```

Function getIPeR (runs on contacted node)

Input:
ContactList: all the nodes in contact with this node excluding the message holder node

```
For each node N in ContactList  do
    ContactOfContactList = getContact(N)
    For each node CN in ContactOfContactList do
         IPeRList.add(CN.IPeR)
    IPeRList.add(N.IPeR)
    harmonicMean= getHarmonicMean(IPeRList))
    if harmonicMean > N.IPeR
          NIPeRList.add(harmonicMean)
    else
          NIPeRList.add(N.IPeR)
harmonicMean= getHarmonicMean(IPeRList))
if harmonicMean > thisNode.IPeR
    return harmonicMean
else
    thisNode.IPeR
```

3.4.3 The experiments

3.4.3.1 Three Hops IPeR based on Harmonic Mean (3Har-MIPeR)

This experiment is identical to 2Har-MIPeR. However, this experiment is founded on the contacts and the contacts of the contacts of the encountered nodes.

3.4.3.2 Three Hops IPeR based on Average (3Avg-MIPeR)

This experiment is identical to 2Avg-MIPeR. However, this experiment is founded on the contacts and the contacts of the contacts of the encountered nodes.

3.4.3.3 Three Hops IPeR based on Maximum (3Max-MIPeR)

This experiment is identical to 2Max-MIPeR. However, this experiment is founded on the contacts and their contacts of the encountered nodes.

3.4.4 Results

Setting 1

<u>50 Users</u>

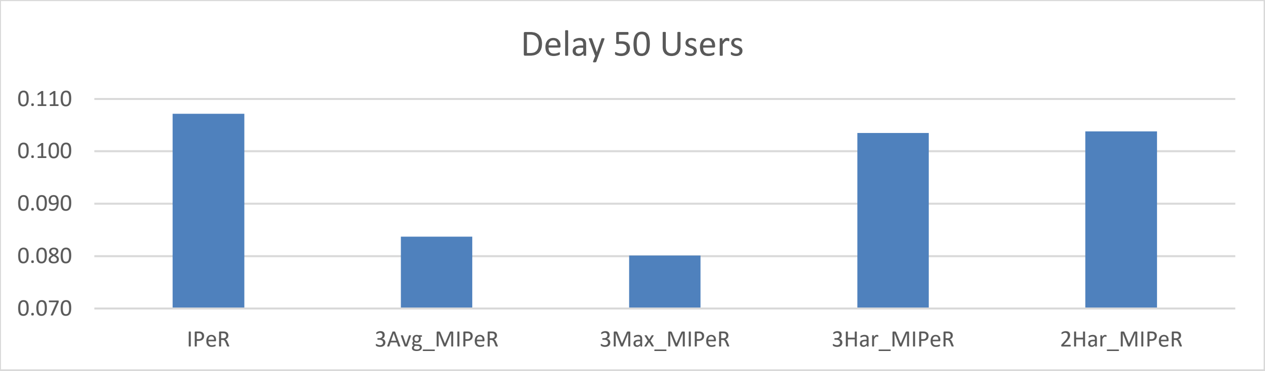


FIGURE 28: Delay of Setting 1, 3Har-MIPeR, 50 Users

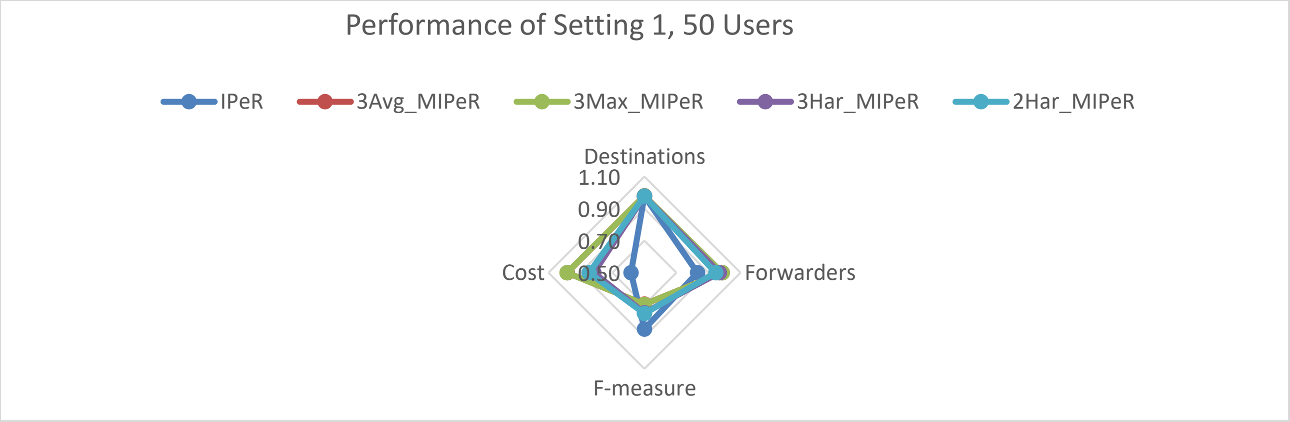


FIGURE 29: Delivery, Cost & F-measure for setting1, 3Har-MIPeR, 50 Users

TABLE 18: Performance of Setting 1, 3Har-MIPeR, 50Users

App	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.60	29.90	0.107	0.85	29
3Avg_MIPeR	17.65	35.45	0.084	0.70	49
3Max_MIPeR	17.65	35.50	0.080	0.70	49
3Har_MIPeR	17.60	34.85	0.104	0.75	41
2Har_MIPeR	17.60	34.00	0.104	0.76	42

100 Users

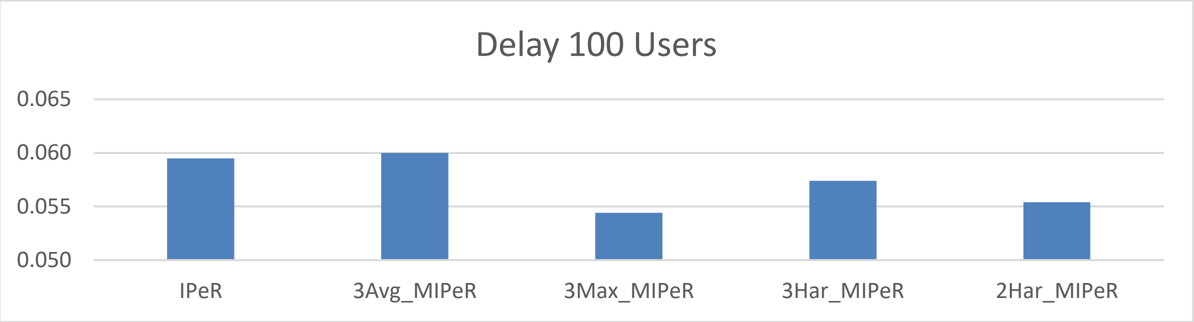


FIGURE 30: Delay of Setting 1, 3Har-MIPeR, 100 Users

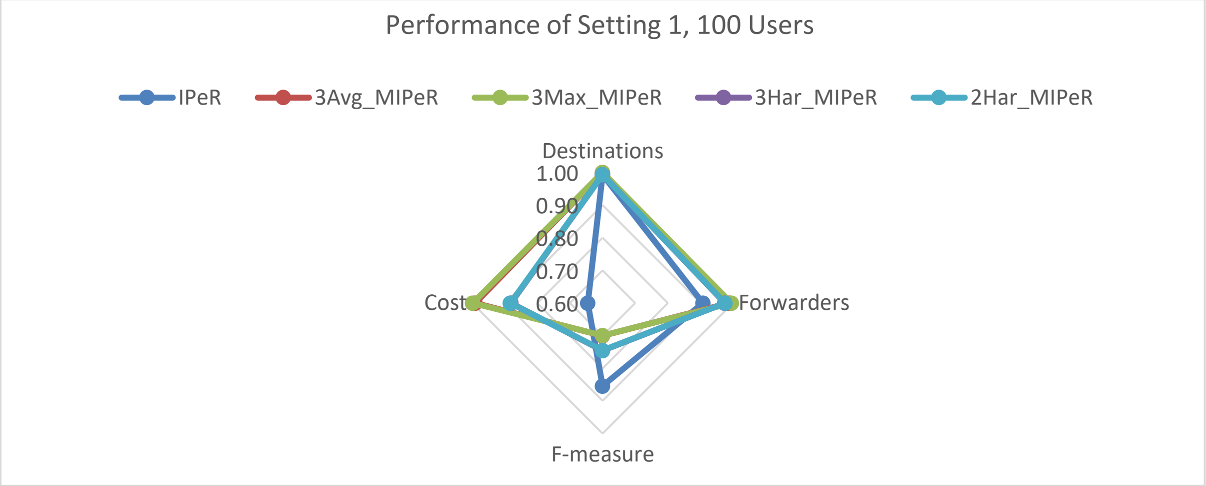


FIGURE 31: Delivery, Cost & F-measure of Setting 1, 3Har-MIPeR, 100 Users

TABLE 19: Performance of Setting 1, 3Har-MIPeR, 100Users

App	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	**17.88**	**32.68**	**0.060**	**0.85**	**65**
3Avg_MIPeR	**18.00**	**35.58**	**0.060**	**0.70**	**99**
3Max_MIPeR	**18.00**	**35.83**	**0.054**	**0.70**	**100**
3Har_MIPeR	**17.88**	**35.10**	**0.057**	**0.75**	**88**
2Har_MIPeR	**17.88**	**35.13**	**0.055**	**0.75**	**88**

200 Users

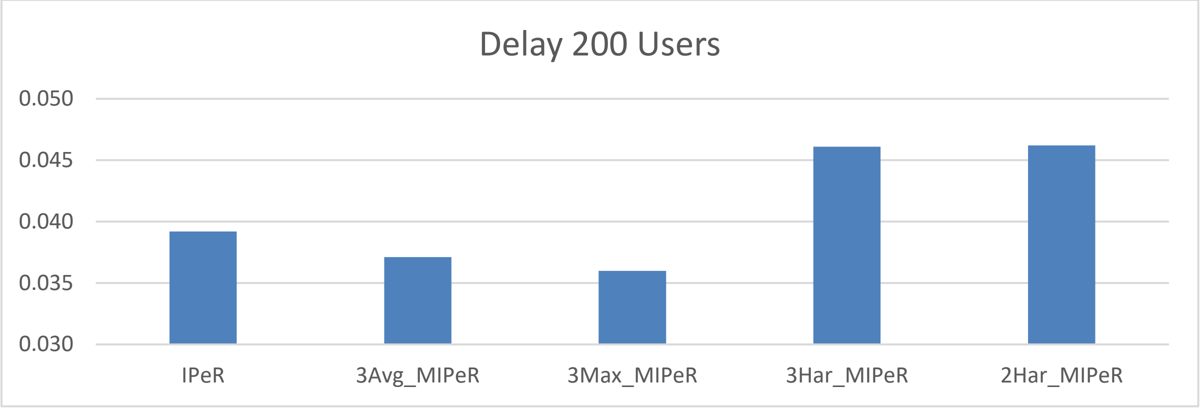


FIGURE 32: Delay of Setting 1, 3Har-MIPeR, 200 Users

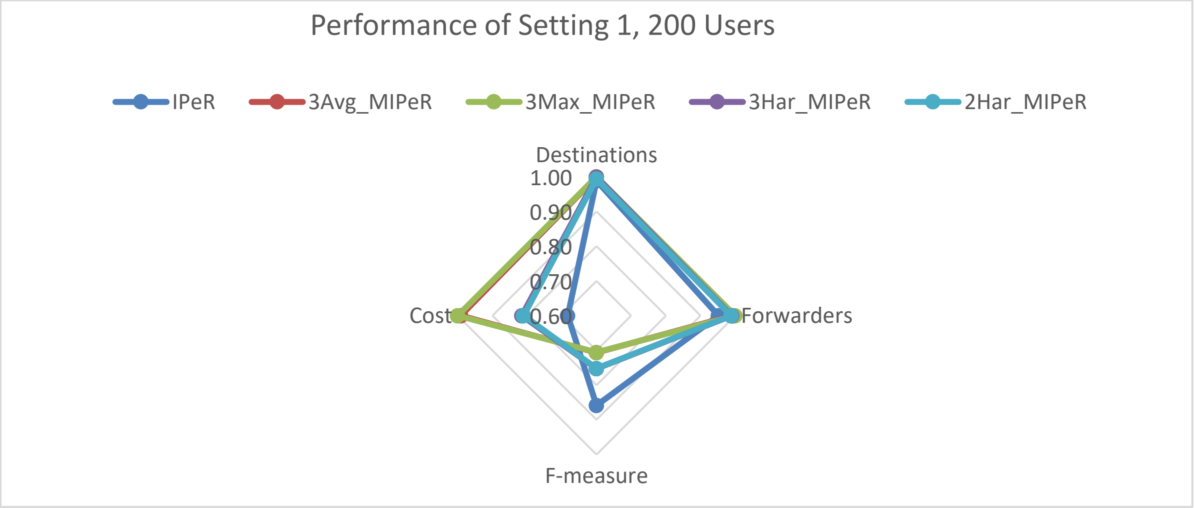


FIGURE 33: Delivery, Cost & F-measure of Setting 1, 3Har-MIPeR, 200 Users

TABLE 20: Performance of Setting 1, 3Har-MIPeR, 200Users

App	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	**17.80**	**34.74**	**0.039**	**0.86**	**137**
3Avg_MIPeR	**17.99**	**36.20**	**0.037**	**0.71**	**198**
3Max_MIPeR	**18.00**	**36.49**	**0.036**	**0.71**	**200**
3Har_MIPeR	**17.99**	**36.14**	**0.046**	**0.75**	**163**
2Har_MIPeR	**17.90**	**36.08**	**0.046**	**0.75**	**162**

Discussion of Results

Based on the previous experiments 2Har-MIPeR was the best. So, we compared it to the 3 hops-MIPeR to define the best algorithm of MIPeR.

For 50 users, 3Avg-MIPeR and **3Max-MIPeR** reached **0.3% more destination nodes**, while **3Har-MIPeR and 2Har-MIPeR** reached the same number of destination nodes compared to IPeR. This is the same result as 2-MIPeR.

For the number of reached **interested forwarders all the proposed algorithms performed better than IPeR. 3Avg-MIPeR** and **3Max-MIPeR reached 19%, 3Har-MIPeR, and 2Har-MIPeR reached** 17% and 14% more interested forwarders in comparison to IPeR, respectively.

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **3Max-MIPeR** reduced delay by **25%. 3Avg-MIPeR reduced it by 22% and 3Har-MIPeR diminished it by 4%. 2Har reduced it by 3%.**

All the proposed algorithms perform worse in terms of cost and F-measure. The reason is that to perform better in delay they need to contact a larger number of uninterested forwarders which decreases the F-measure and increases cost. The best of them is **2Har-MIPeR, which diminished F-measure by 11%. Then, 3Har-MIPeR, 3Max-MIPeR and 3Avg-MIPeR decreased it by 12%, 18% and 18%, respectively.**

For **the cost, the best algorithm among the proposed algorithms is 3Har-MIPeR, which increased it by 40%. Then, 2Har-MIPeR, 3Max-MIPeR and 3Avg-MIPeR increased by 45%, 68% and 68%, respectively.**

For 100 users, **3Avg-MIPeR** and **3Max-MIPeR** reached **0.7% more destination nodes**, while **3Har-MIPeR and 2Har-MIPeR** reached the same number of destination nodes compared to IPeR.

For the number of reached **interested forwarders all the proposed algorithms performed better than IPeR. 3Max-MIPeR, 3Avg-MIPeR, 2Har-MIPeR and 3Har-MIPeR** reached **10%, 9%, 7% and 7%** more interested forwarders in comparison to IPeR, respectively.

For **delay, all the proposed algorithms slightly enhanced** it in comparison to IPeR except 3Avg-MIPeR performed equally compared to IPeR. **The best algorithm** was **3Max-MIPeR** which reduced delay by **9%. Then, 2Har-MIPeR and 3Har-MIPeR enhanced it by 7% and 4%, respectively.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them is **2Har-MIPeR** and **3Har-MIPeR,** which diminished **F-measure by 13%.** Then**, 3Max-MIPeR** and **3Avg-MIPeR** decreased it by **18%.**

For **the cost, the best algorithm among the proposed algorithms is 3Har-MIPeR and 2Har-MIPeR** which increased **it by 37%. 3Avg-MIPeR and 3Max-MIPeR** increased it by **53%, and 55%** compared to IPeR, respectively.

For 200 users, all the proposed algorithms reached **1% more destination nodes** compared to IPeR except **2Har**-MIPeR reached **0.5%** more destinations. For the number of reached **interested forwarders, all the proposed algorithms performed better than IPeR. 3Max-MIPeR** reached **5%,** while **2Har-MIPeR, 3Har-MIPeR and 3Avg-MIPeR** reached **4%** more interested forwarders in comparison to IPeR.

For **delay, 3Max-MIPeR and 3Avg-MIPeR enhanced** it in comparison to IPeR by 8% and 5%, respectively. However**, 2Har-MIPeR and 3Har-MIPeR** increased delay by **18%.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them are **2Har-MIPeR** and **3Har-MIPeR,** which diminished **F-measure by 12%.** Then**, 3Max-MIPeR** and **3Avg-MIPeR** decreased it by **18%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR and 3Har-MIPeR,** which increased **it by 19%.** Then, **3Avg-MIPeR** increased it by **45%** and **3Max-MIPeR** increased cost by **46%** compared to IPeR.

For more details see appendix B.

Conclusion of Setting 1

In conclusion, the results are almost identical to 2MIPeR results. 3Har-MIPeR performs slightly better than 2Har-MIPeR in terms of delivery ratio, reached interested forwarders. They perform equally in delay and F-measure. While 2Har-MIPeR performs trivially better compared to 3Har-MIPeR in terms of cost.

Setting 2

50 Users

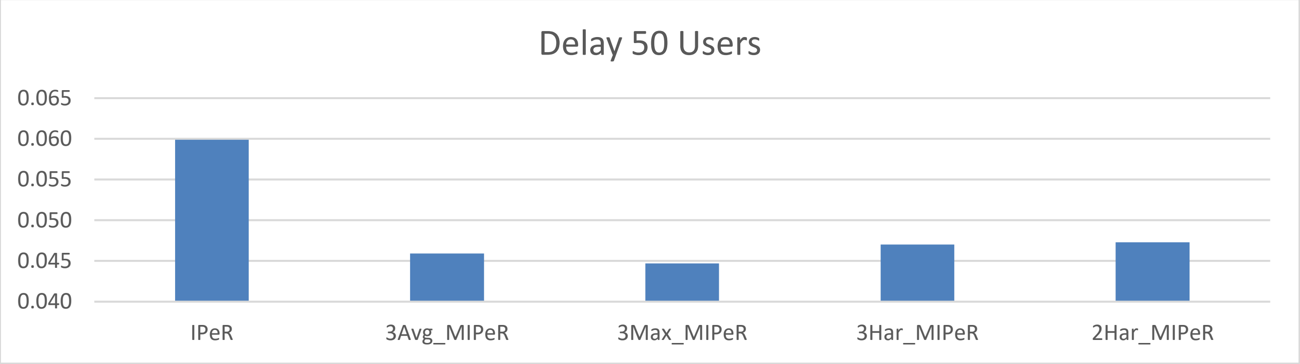


FIGURE 34: Delay of Setting 2, 3Har-MIPeR, 50 Users

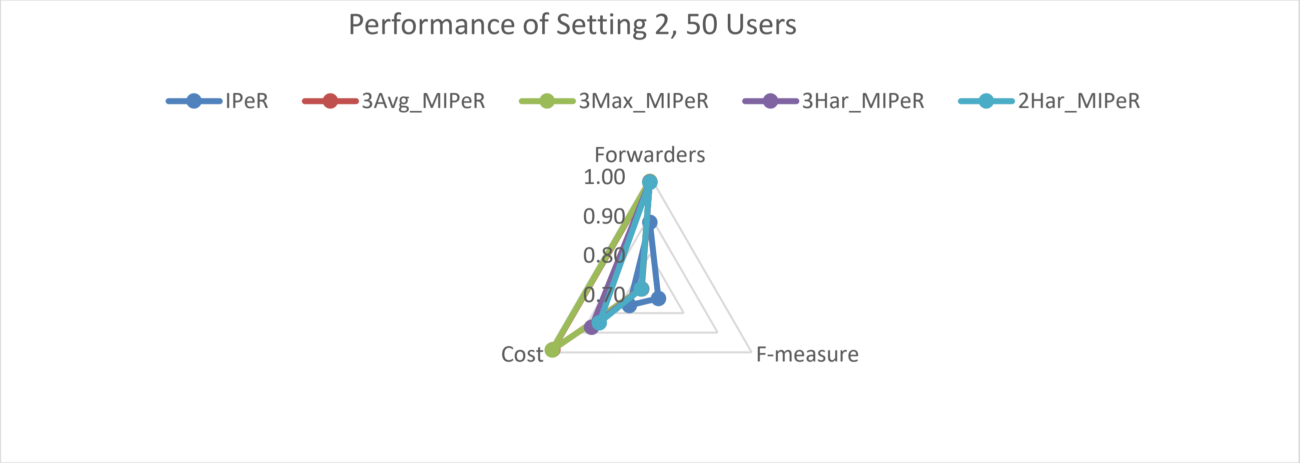


FIGURE 35: Delivery, Cost & F-measure of Setting 2, 3Har-MIPeR, 50 Users

TABLE 21: Performance of Setting 2, 3Har-MIPeR, 50Users

App	Forwarders	Delay	F-measure	Cost
IPeR	**43.20**	**0.060**	**0.73**	**38**
3Avg_MIPeR	**48.25**	**0.046**	**0.68**	**49**
3Max_MIPeR	**48.35**	**0.045**	**0.68**	**49**
3Har_MIPeR	**48.25**	**0.047**	**0.68**	**44**
2Har_MIPeR	**48.20**	**0.047**	**0.68**	**42**

100 Users

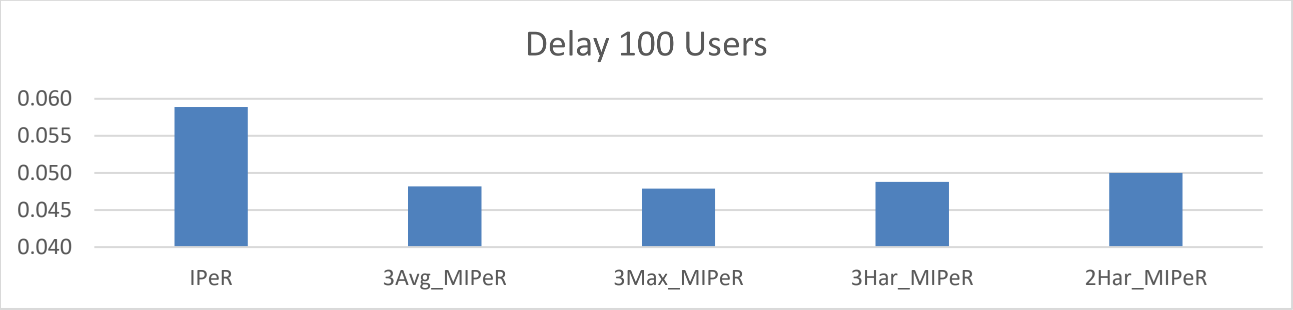


FIGURE 36: Delay of Setting 2, 3Har-MIPeR, 100 Users

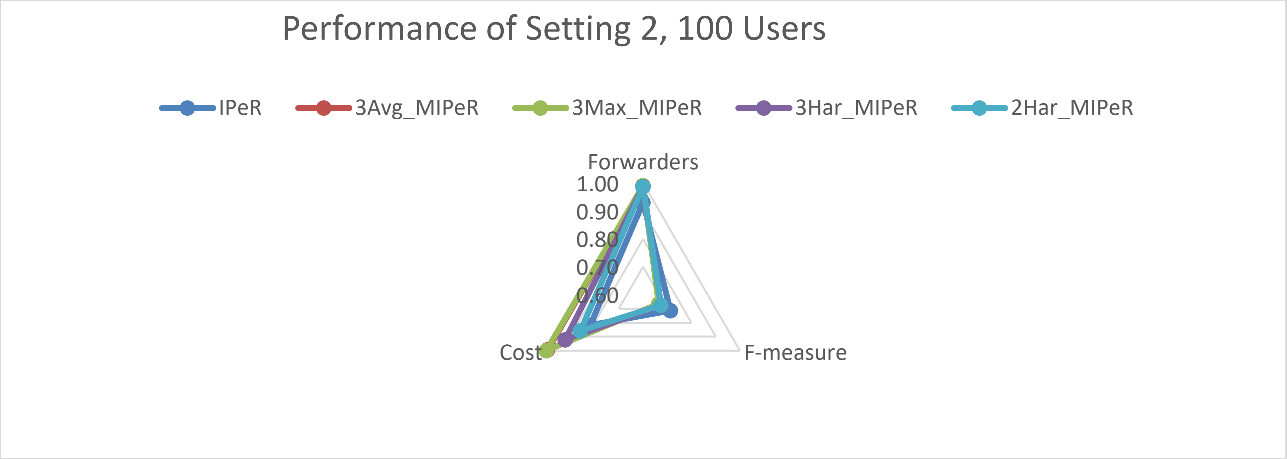


FIGURE 37: Delivery, Cost & F-measure of Setting 2, 3Har-MIPeR, 100 Users

TABLE 22: Performance of Setting 2, 3Har-MIPeR, 100Users

App	Forwarders	Delay	F-measure	Cost
IPeR	**45.58**	**0.059**	**0.71**	**82**
3Avg_MIPeR	**48.40**	**0.048**	**0.67**	**99**
3Max_MIPeR	**48.53**	**0.048**	**0.67**	**100**
3Har_MIPeR	**48.35**	**0.049**	**0.67**	**92**
2Har_MIPeR	**48.33**	**0.050**	**0.67**	**86**

200 Users

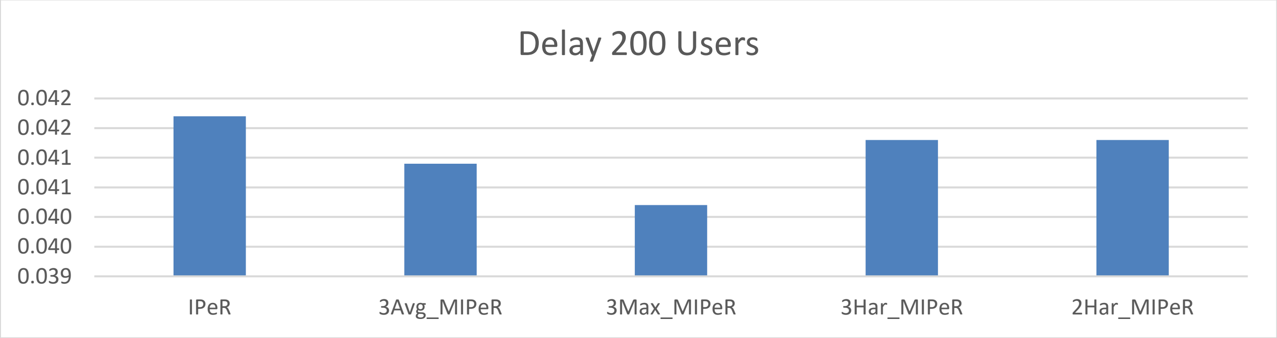


FIGURE 38: Delay of Setting 2, 3Har-MIPeR, 200 Users

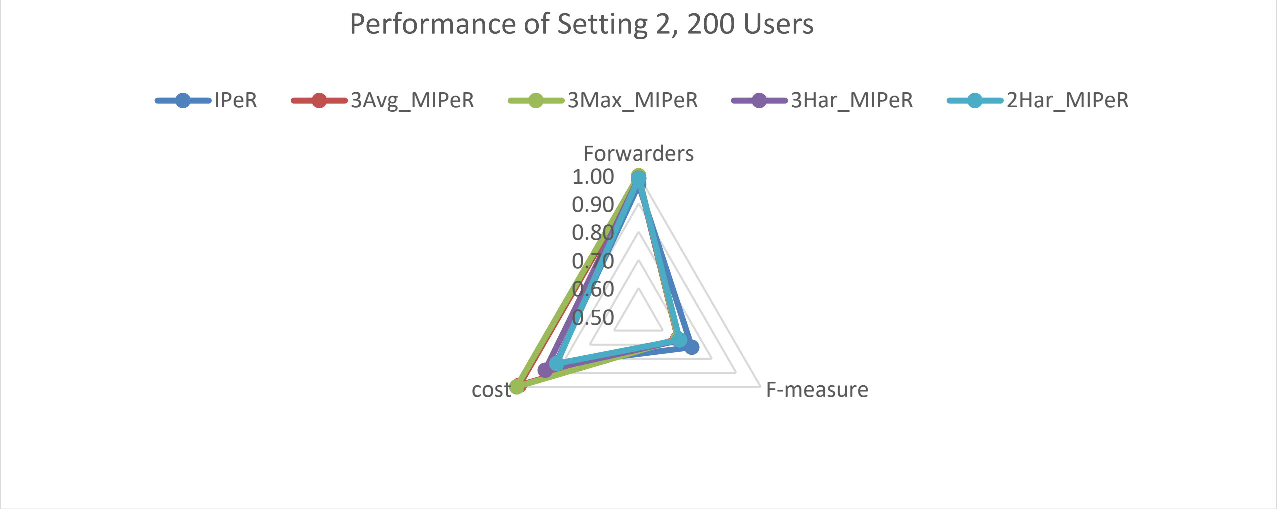


FIGURE 39: Delivery, Cost & F-measure of Setting 2, 3Har-MIPeR, 200 Users

TABLE 23: Performance of Setting 2, 3Har-MIPeR, 200Users

App	Forwarders	Delay	F-measure	cost
IPeR	**47.36**	**0.042**	**0.72**	**168**
3Avg_MIPeR	**48.49**	**0.041**	**0.66**	**198**
3Max_MIPeR	**48.96**	**0.040**	**0.66**	**200**
3Har_MIPeR	**48.49**	**0.041**	**0.67**	**177**
2Har_MIPeR	**48.44**	**0.041**	**0.67**	**167**

Discussion of Results

Based on the previous experiments 2Har-MIPeR was the best. So, we compared it to the 3 hops-MIPeR to define the best algorithm of MIPeR.

For 50 users, for the number of reached **interested forwarders, all the proposed algorithms performed better than IPeR with slight differences. The best is 3Max-MIPeR which reached 12%** more interested forwarders in comparison to IPeR.

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **3Max-MIPeR is the best which reduced** delay by **25%. 3Avg-MIPeR, 3Har-MIPeR and 2Har-MIPeR reduced it by 23%, 22% and 21%, respectively.**

All the proposed algorithms perform worse in terms of cost and F-measure. All of them increased F-measure by 7%. For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR, which increased it by 13%. Then, 3Har-MIPeR, 3Avg-MIPeR and 3Max-MIPeR increased by 16%, 31% and 31%, respectively.**

For 100 users, for the number of reached **interested forwarders, all the proposed algorithms performed almost equally among each other and better than IPeR. 3Max-MIPeR is the best and enhanced it by 7%.**

For **delay, all the proposed algorithms slightly enhanced** it in comparison to IPeR. **The best algorithms were 3Max-MIPeR and 3Avg-MIPeR** which reduced delay by **18%. Then, 3Har-MIPeR and 2Har-MIPeR enhanced it by 17% and 14%, respectively.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them is **2Har-MIPeR** and **3Har-MIPeR,** which diminished **F-measure by 6%.** Then**, 3Max-MIPeR** and **3Avg-MIPeR** decreased it by **7%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR** which increased **it by 5%. 3Har-MIPeR, 3Avg-MIPeR and 3Max-MIPeR** increased it by **13%, 22%, and 22%** compared to IPeR, respectively.

For 200 users, for the number of reached **interested forwarders, all the proposed algorithms performed better than IPeR. 3Max-MIPeR** reached **3%,** in comparison to IPeR.

For **delay, all the proposed algorithms performed better in comparison to IPeR. 3Max-MIPeR is the best, which reduced delay by 4%.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them are **2Har-MIPeR** and **3Har-MIPeR,** which diminished **F-measure by 7%.** Then**, 3Max-MIPeR** and **3Avg-MIPeR** decreased it by **8%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased **it by 0.6%.**

For more details see appendix B.

Conclusion of Setting 2

In conclusion, the results are almost identical to 2MIPeR results. 3Har-MIPeR performs slightly better than 2Har-MIPeR in terms of reached interested forwarders and cost. They perform equally in delay and F-measure.

Setting 3

50 Users

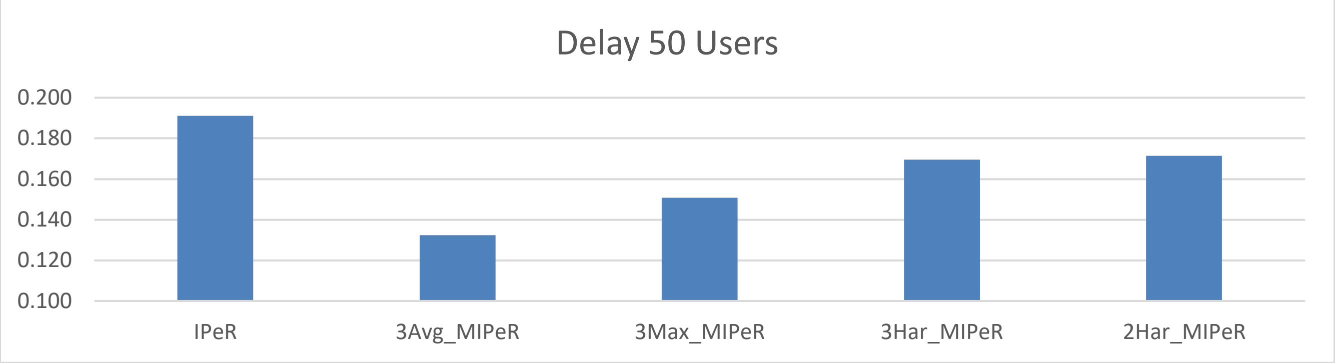


FIGURE 40: Delay of Setting 3, 3Har-MIPeR, 50 Users

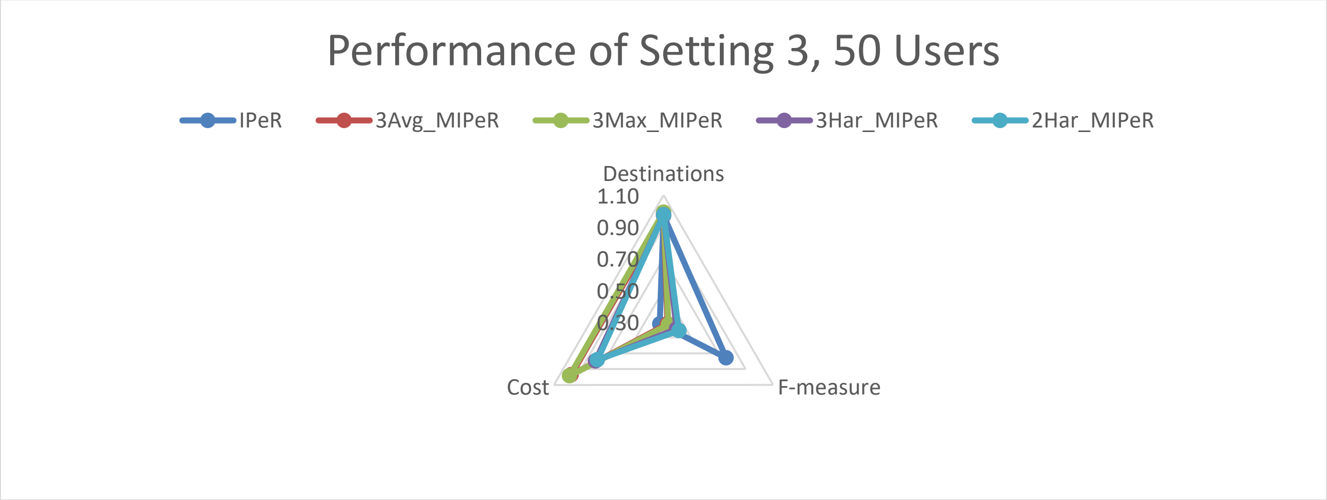


FIGURE 41: Delivery, Cost & F-measure of Setting 3, 3Har-MIPeR, 50 Users

TABLE 24: Performance of Setting 3, 3Har-MIPeR, 50Users

App	Destinations	Delay	F-measure	Cost
IPeR	**19.40**	**0.191**	**0.76**	**16**
3Avg_MIPeR	**19.50**	**0.132**	**0.33**	**49**
3Max_MIPeR	**19.85**	**0.151**	**0.34**	**49**
3Har_MIPeR	**19.50**	**0.170**	**0.39**	**40**
2Har_MIPeR	**19.50**	**0.171**	**0.41**	**39**

100 Users

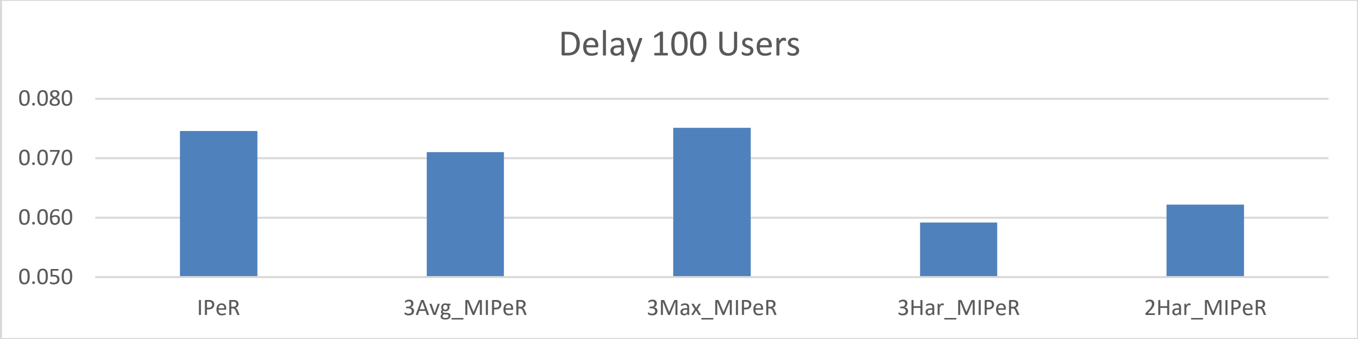


FIGURE 42: Delay of Setting 3, 3Har-MIPeR, 100 Users

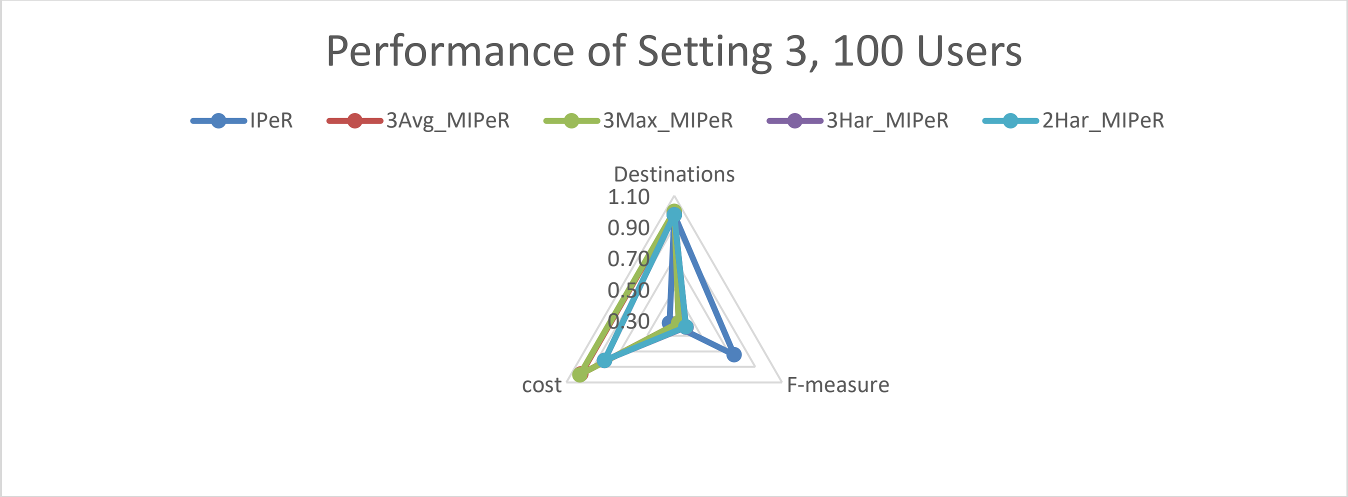


FIGURE 43: Delivery, Cost & F-measure of Setting 3, 3Har-MIPeR, 100 Users

TABLE 25: Performance of Setting 3, 3Har-MIPeR, 100Users

App	Destinations	Delay	F-measure	cost
IPeR	**19.55**	**0.075**	**0.74**	**33**
3Avg_MIPeR	**19.88**	**0.071**	**0.33**	**99**
3Max_MIPeR	**20.00**	**0.075**	**0.33**	**100**
3Har_MIPeR	**19.55**	**0.059**	**0.39**	**82**
2Har_MIPeR	**19.55**	**0.062**	**0.39**	**82**

200 Users

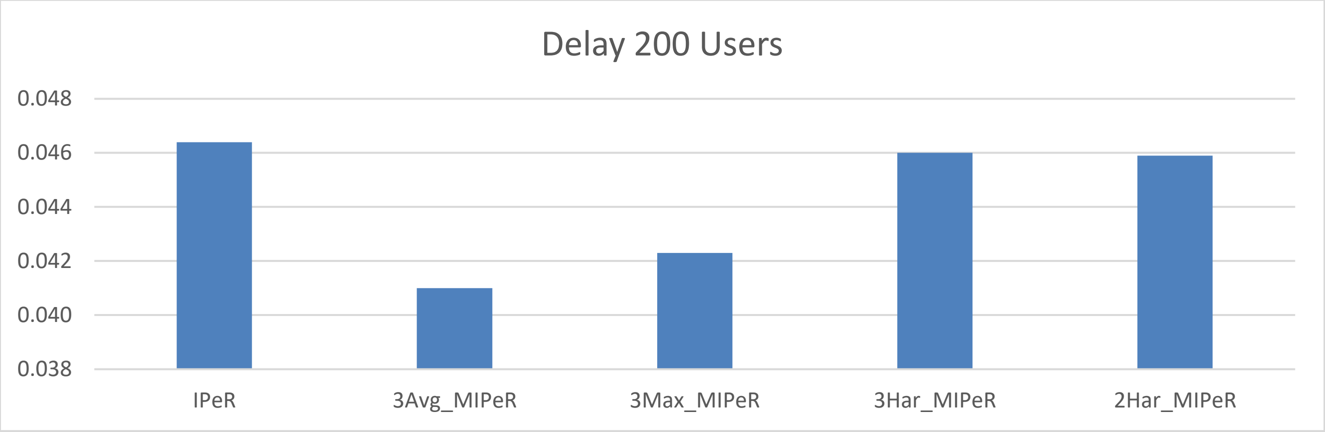


FIGURE 44: Delay of Setting 3, 3Har-MIPeR, 200 Users

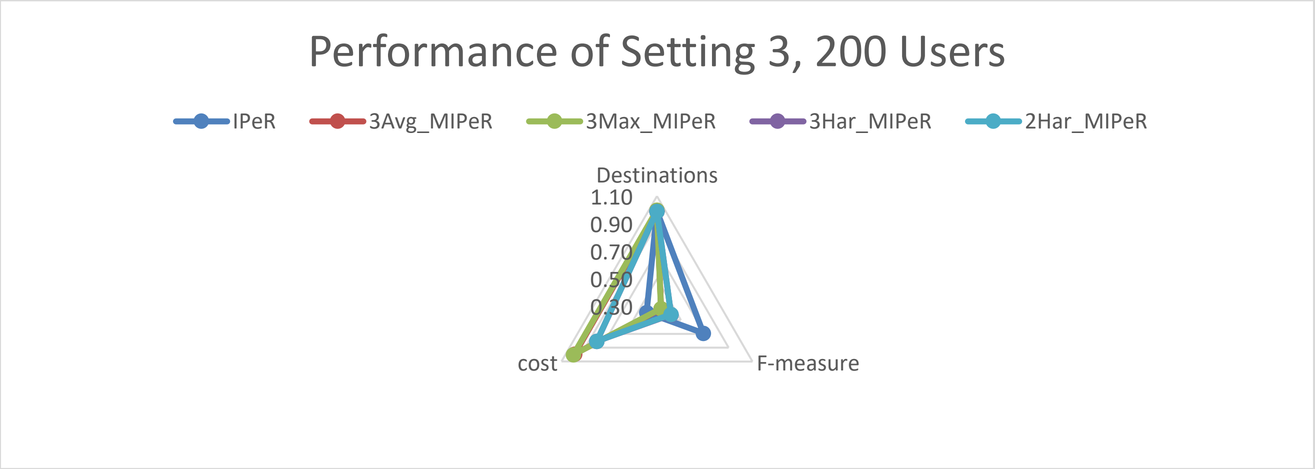


FIGURE 45: Delivery, Cost & F-measure of Setting 3, 3Har-MIPeR, 200 Users

TABLE 26: Performance of Setting 3, 3Har-MIPeR, 200Users

App	Destinations	Delay	F-measure	cost
IPeR	**19.80**	**0.046**	**0.69**	**77**
3Avg_MIPeR	**19.84**	**0.041**	**0.33**	**198**
3Max_MIPeR	**19.99**	**0.042**	**0.33**	**200**
3Har_MIPeR	**19.80**	**0.046**	**0.42**	**161**
2Har_MIPeR	**19.80**	**0.046**	**0.42**	**161**

Discussion of Results

Based on the previous experiments 2Har-MIPeR was the best. So, we compared it to the 3 hops-MIPeR to define the best algorithm of MIPeR.

For 50 users, for **the delivery ratio, all the proposed algorithms performed better than IPeR with slight differences. The best is 3Max-MIPeR which reached 2.3%** more destination nodes in comparison to IPeR.

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **3Avg-MIPeR is the best which reduced** delay by **31%. 3Max-MIPeR, 3Har-MIPeR and 2Har-MIPeR reduced it by 21%, 11% and 10%, respectively.**

All the proposed algorithms perform worse in terms of cost and F-measure. All of them increased F-measure by 7%. For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR, which increased it by 46%. Then, 3Har-MIPeR, 3Avg-MIPeR and 3Max-MIPeR increased by 48%, 56% and 56%, respectively.**

For 100 users, for **delivery ratio, all the proposed algorithms performed almost equally among each other and better than IPeR. 3Max-MIPeR is the best and enhanced it by 2%.**

For **delay, all the proposed algorithms enhanced** it in comparison to IPeR. **The best algorithm was 3Har-MIPeR, which enhanced it by 21%. Then, 2Har-MIPeR** reduced delay by **17%.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them is **2Har-MIPeR** and **3Har-MIPeR,** which diminished **F-measure by 48%.** Then**, 3Max-MIPeR** and **3Avg-MIPeR** decreased it by **55%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR and 3Har-MIPeR** which increased **it by 144%. 3Avg-MIPeR and 3Max-MIPeR** increased it by **169% and 199%** compared to IPeR, respectively.

For 200 users, for the delivery ratio, **all the proposed algorithms performed better than IPeR. 3Max-MIPeR** reduced it by **0.9%,** in comparison to IPeR.

For **delay, all the proposed algorithms performed better in comparison to IPeR. 3Avg-MIPeR is the best, which reduced delay by 12%.**

All the proposed algorithms perform worse in terms of cost and **F-measure**. The best of them are **2Har-MIPeR** and **3Har-MIPeR,** which diminished **F-measure by 40%.** Then**, 3Max-MIPeR** and **3Avg-MIPeR** decreased it by **52%.**

For **the cost, the best algorithm among the proposed algorithms is 2Har-MIPeR,** which increased **it by 108%.**

For more details see appendix B.

<u>Conclusion of Setting 3</u>

In conclusion, the results are almost identical to 2MIPeR results. 3Har-MIPeR performed almost identical to 2Har-MIPeR with respect to all metrics.

Conclusion on 3MIPeR

All the 3-hop MIPeR versions performed almost identical to their counter 2-hop versions. 3Har-MIPeR is the best among the other 3-hop proposed algorithms in terms of F-measure. 2HarMIPeR performed equal or better than 3Har-MIPeR in F-measure and cost. The reason is that they reach the same number of destinations without significant enhancement in delay. The reason is that 3-hop algorithms reached most of the nodes earlier than their 2-hops corresponds. Then, they stop reaching more destinations or interested forwarders. Finally, they catch the rest of the destinations and the interested forwarders near the end of the simulation time which increases the average of the delay.

Conclusion on MIPeR

In this CHAPTER, we took the first steps towards showing the impact of incorporating multiple hop awareness into interest-based socially aware opportunistic forwarding algorithms, namely, MIPeR. Our simulation-based evaluation demonstrates the promising gain in delivery ratio, contacted interested forwarders ratio, and delay. Our contribution is defined as (1) adding multiple hops to IPeR enhances its delivery ratio, contacted interested forwarders ratio, and delay slightly. However, it increased the cost and significantly decreased F-measure. (2) using a 3-hop variation of the algorithm does not perform any significant gain. (3) 2Har-MIPeR is the best compared to the proposed algorithms in terms of cost and F-measure. (4) 2Max-MIPeR is the best in terms of delay, delivery ratio, and the ratio of contacted interested forwarders. (5) using 2-hop awareness can be useful for environments in which we can sacrifice cost to contact more destinations and interested forwarders such as disaster rescue.

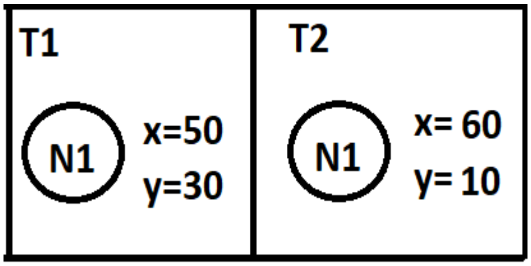
CHAPTER 4: Embrace Direction as a Guiding Criterion (Directional-IPeR)

In this CHAPTER, we introduce direction awareness to interest-aware social-based forwarding algorithms that recognize destination nodes by their interest profile.

Overview

Directional-IPeR introduces direction awareness into the forwarding decision of the IPeR algorithm. The aim is to increase the chance that a copy of the message is sent to diverse directions with certain ratios to increase the probability of reaching the destinations. With the inspiration of the Direction Entropy-Based Forwarding Scheme (DEFS) [66] each node has its transmission range divided into 4 quarters namely R1, R2, R3, R4 with an angle of 90 degrees as illustrated in FIGURE 47. Each node stores its locations at the latest two-time slots as illustrated in TABLE 27. Then, it compares them to find the difference, which will map its direction of motion to one of the 5 states (S0, S1, …S4). Note that state S0 indicates that the node did not move as illustrated in RequestState algorithm in TABLE 28.

TABLE 27: Storing a node locations at the latest two-time slots (T1 & T@)



T1	T2
N1 x=50 y=30	N1 x= 60 y= 10

TABLE 28: RequestState Algorithm

```
RequestState
    xDif ← X[T1] - X[T2]
    yDif ← Y[T1] - Y[T2]
    if ((xDif >= 0) && (yDif <= 0))
        state = 1
    else if ((xDif >= 0) && (yDif >= 0))
        state = 2
    else if ((xDif <= 0) && (yDif <= 0))
        state = 4
    else if ((xDif <= 0) && (yDif >= 0))
        state = 3
    else
        state=0
Return state
```

Several experiments and settings are used to examine the performance of such an algorithm. The Directional-IPeR experiments are illustrated in FIGURE 46.

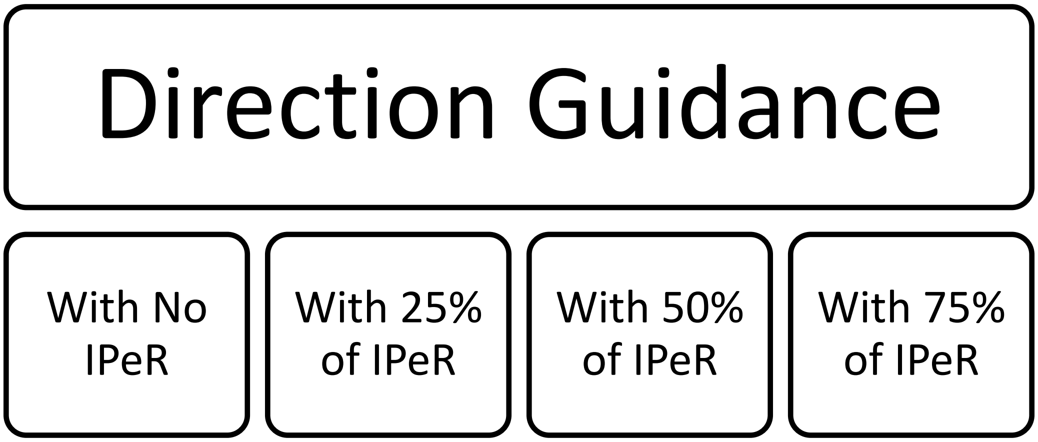


FIGURE 46: Directional-IPeR Experiments

4.1 Directional-IPeR

4.1.1 Steps of the Algorithm

In IPeR, when a group of nodes has IPeR values higher than or equal to that of the message holder node, they receive a copy of the message. The Directional-IPeR algorithm examines the four main directions of the nodes in this group. If MH finds a direction to which no nodes are heading, it sends a copy of the message to other nodes that are heading in this direction but has an IPeR value greater than the tolerance factor. Such a case is illustrated in FIGURE 47 as the circle in R3. It sacrifices a percentage of The IPeR value threshold for selecting the forwarder nodes to be sure that all the four main directions are covered. The goal is to increase the delivery ratio but with the consideration of the cost. The privacy of each node is maintained as each node calculates its value individually and they share a final value that is not reversible. Thus, other nodes cannot extract the private info of the node's profile.

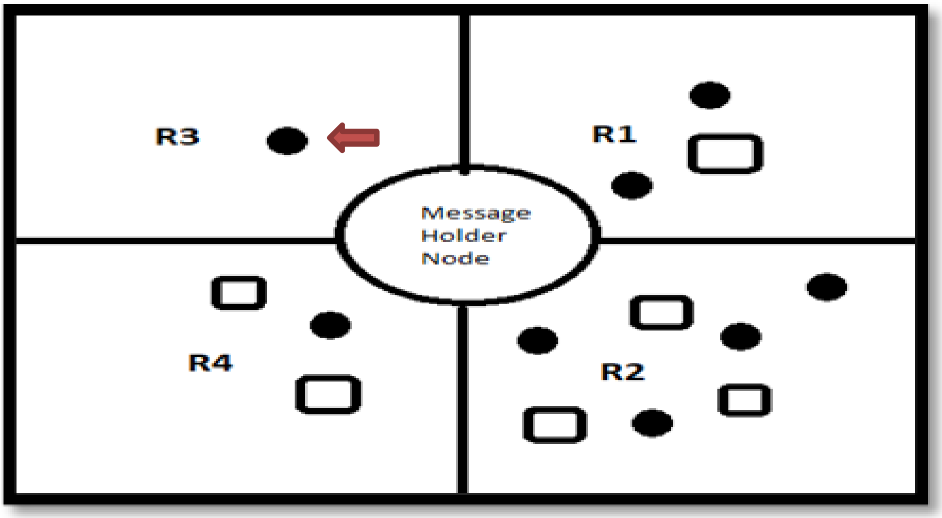


FIGURE 47: Directional IPeR

The different values of the threshold of Directional-IPeR are:

- Zero → No consideration of IPeR. It includes all the nodes with no consideration of their IPeR values
- 25% of IPeR. It includes all the nodes which have IPeR values of more than 25% of the IPeR value of the message holder node.

- 50% of IPeR. It includes all the nodes which have IPeR values of more than 50% of the IPeR value of the message holder node.

- 75% of IPeR. It includes all the nodes which have IPeR values of more than 75% of the IPeR value of the message holder node.

4.1.2 Directional-IPeR Algorithm

```
Function Directional-IPeR
--------------------------------------------------------------------------------
Input:
IPeRList ← all the nodes that in contact with the current node and their IPeR value is greater than or equal to the current node’s IPeR value
ContactList ← all the nodes that in contact with the current node and their IPeR value is greater than or equal to 75% of the current node’s IPeR value
--------------------------------------------------------------------------------
Output:
Directional_IPeRList
--------------------------------------------------------------------------------
Declare lists s1, s2, s3, s4, Directional_IPeRList as lists of type node
Declare ExtraControlMessages as a number
Directional_IPeRList ← IPeRList
For each node in IPeRList do
get the difference in location between the last timeslot and current timeslot to know the direction of the node
increment ExtraControlMessages by 1
        add each node to the corresponding state (s1, s2, s3, s4)
if any state is empty
        For each node in ContactList do
        get the difference in location between the last timeslot and current timeslot to know the direction of the node
                increment ExtraControlMessages by 1
                if the corresponding state is empty and this node does not have a copy of the message
                   add the node to the corresponding state
                  add the node to the Directional_IPeRList
```

4.1.3 Results

Setting 1

50 Users

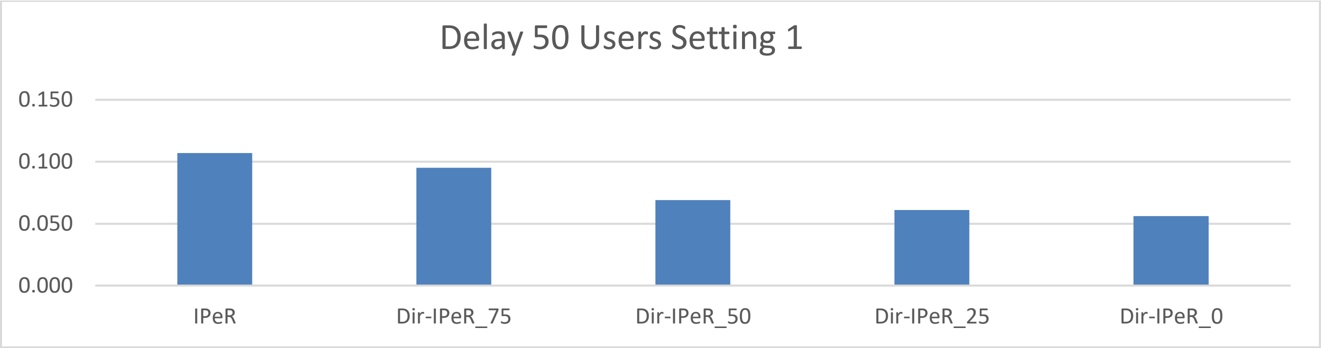


FIGURE 48: Delay of Setting 1, Directional-IPeR, 50 Users

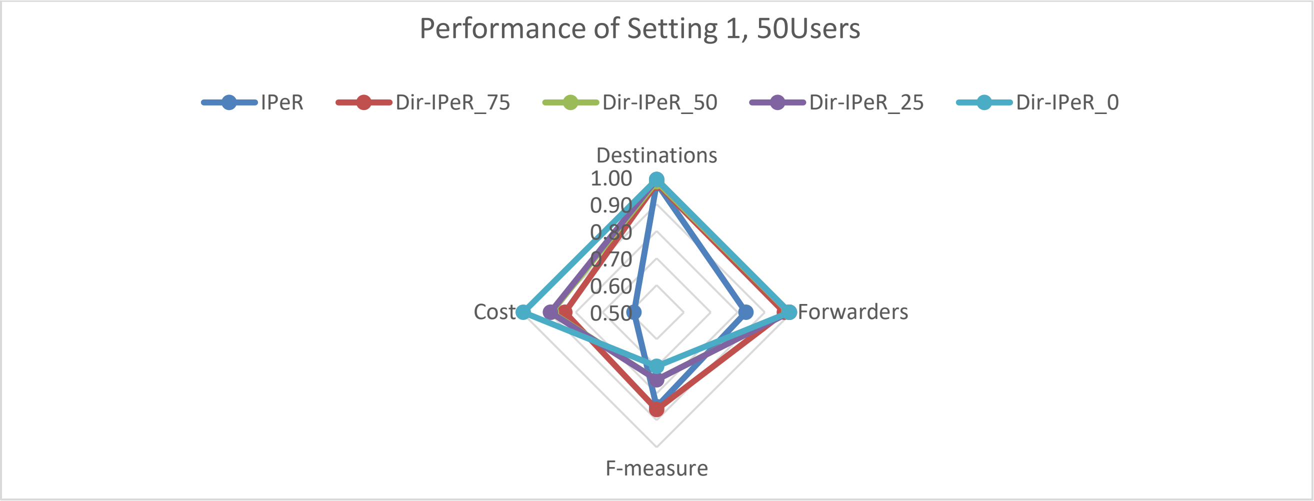


FIGURE 49: Performance of Setting 1, Directional-IPeR, 50Users

TABLE 29: Performance of Setting 1, Directional-IPeR, 50Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.60	29.90	0.107	0.85	29	0
Dir-IPeR_75	17.60	35.00	0.095	0.86	42	36
Dir-IPeR_50	17.70	35.60	0.069	0.75	45	51
Dir-IPeR_25	17.85	35.70	0.061	0.75	45	54
Dir-IPeR_0	17.85	35.70	0.056	0.70	50	67

100 Users

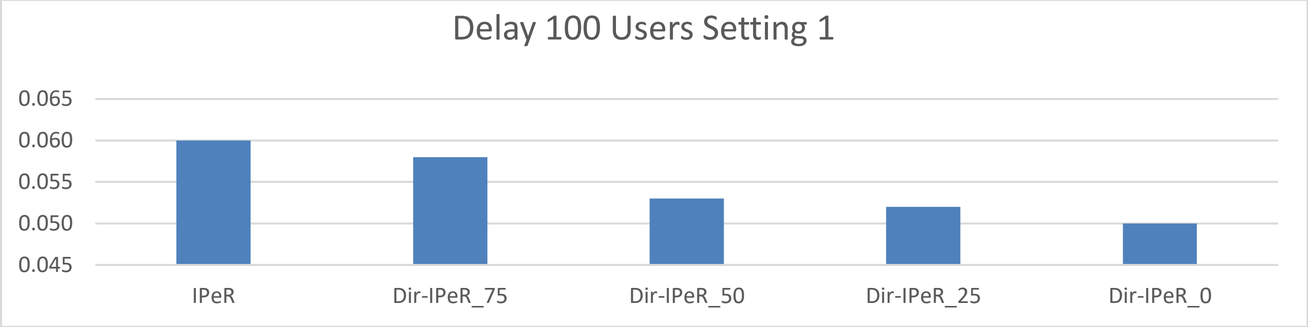


FIGURE 50: Delay of Setting 1, Directional-IPeR, 100 Users

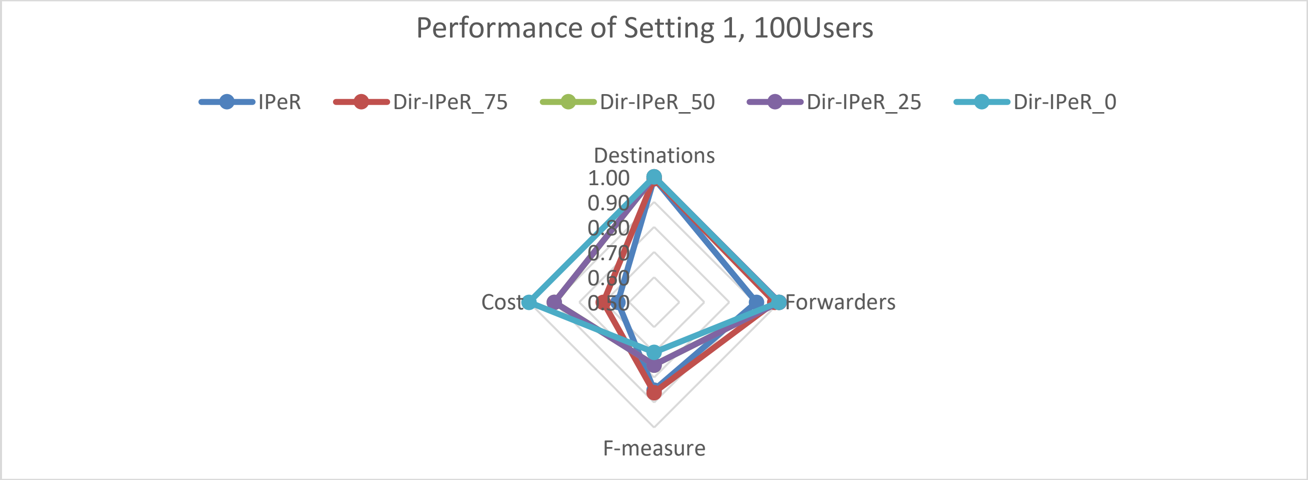


FIGURE 51: Performance of Setting 1, Directional-IPeR, 100Users

TABLE 30: Performance of Setting 1, Directional-IPeR, 100Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.88	32.68	0.060	0.85	65	0
Dir-IPeR_75	17.88	35.33	0.058	0.86	70	81
Dir-IPeR_50	18.00	35.93	0.053	0.75	90	122
Dir-IPeR_25	18.00	35.98	0.052	0.75	90	138
Dir-IPeR_0	18.00	35.90	0.050	0.70	100	182

200 Users

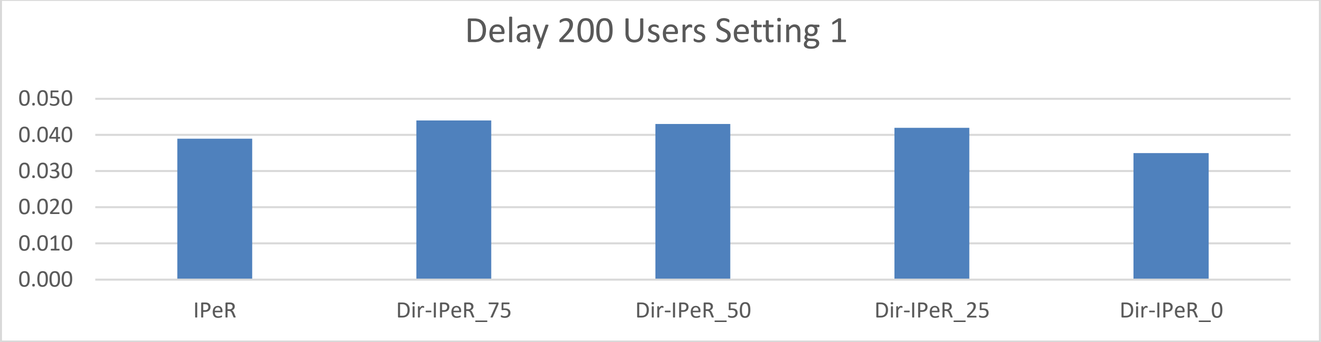


FIGURE 52: Delay of Setting 1, Directional-IPeR, 200 Users

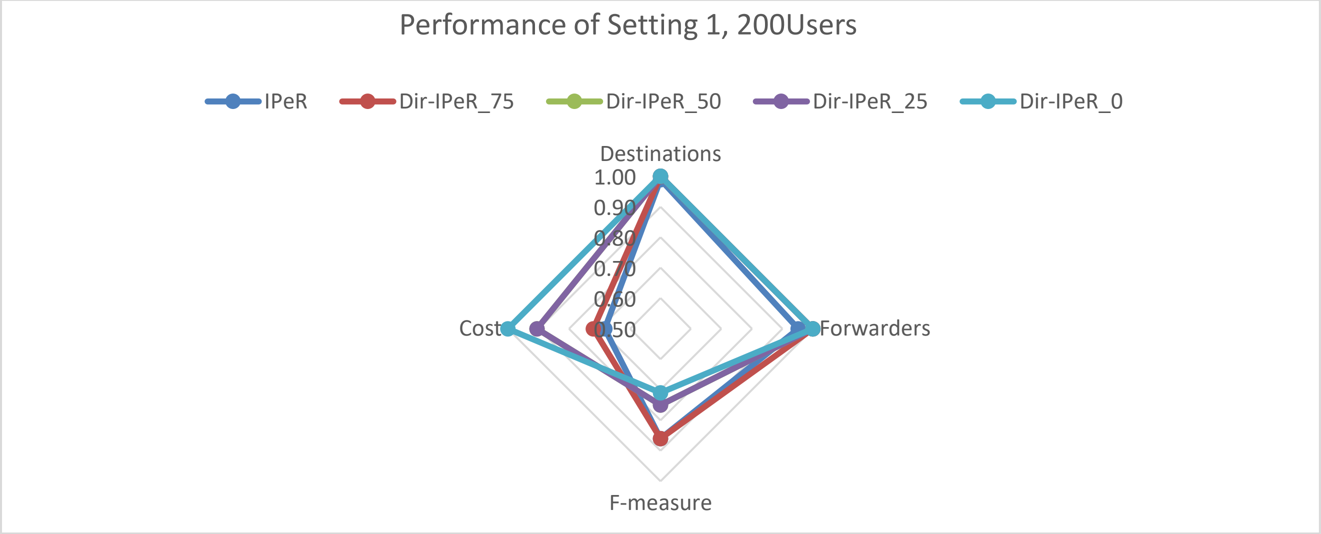


FIGURE 53: Performance of Setting 1, Directional-IPeR, 200Users

TABLE 31: Performance of Setting 1, Directional-IPeR, 200Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.80	34.74	0.039	0.86	137	0
Dir-IPeR_75	17.99	36.49	0.044	0.86	144	189
Dir-IPeR_50	18.00	36.49	0.043	0.75	181	325
Dir-IPeR_25	18.00	36.49	0.042	0.75	181	389
Dir-IPeR_0	18.00	36.49	0.035	0.71	200	529

Discussion of Results

*For discrete uniform distribution of interest, for 50 users***,** Directional-IPeR-50, 25, and zero equally enhanced **delivery ratio** by 1% and the **number of reached interested forwarders** by 19% in comparison to IPeR. However, Directional-IPeR-75 achieved equal delivery ratio compared to IPeR and enhanced the number of reached interested forwarders by 17%. Directional-IPeR-75, 50, 25, and zero enhanced **delay** by 11%, 36%, 43%, and 48%, respectively. Directional-IPeR-75 is the only algorithm that enhanced **F-measure**, which is by 1%. Yet, Directional-IPeR-50, 25, and zero decreased it by 12%, 12% and 18%, respectively. For **the cost**, the best algorithm among the proposed algorithms is Directional-IPeR-75 which increased it by 44%. Nevertheless, Directional-IPeR-50, 25, and zero increased it by 52%, 53%, and 70%, respectively.

For 100 users, all the proposed algorithms performed the same as in 50 users for **delivery ratio** and **F-measure**. Directional-IPeR-50, 25, and zero equally reached more **interested forwarders** by 10%. While Directional-IPeR-75 enhanced the number of reached interested forwarders by 8%. Directional-IPeR-75, 50, 25, and zero enhanced **delay** by 3%, 12%, 13% and 17%, respectively. For **the cost**, the best algorithm among the proposed algorithm is Directional-IPeR-75 which increased it by 9%. However, Directional-IPeR-50, 25, and zero increased it by 39%, 39%, and 55%, respectively.

For 200 users, all the proposed algorithms equally enhanced **delivery ratio** by 1%, and the number of **reached interested forwarders** by 5%. For **delay**, Directional-IPeR-zero is the only algorithm that enhanced delay by 10%. Though, Directional-IPeR- 75, 50, 25, increased delay by 13%, 10% and 8%, respectively. The reason is that these algorithms reached more destination nodes, compared to IPeR, near the end of the simulation time. Using the average to calculate delay, reflected as an increase of delay. Directional-IPeR-75 performed equally to IPeR in terms of **F-measure.** Nevertheless, Directional-IPeR-50, 25, and zero decreased F-measure by 13%, 13% and 17%, respectively. For **the cost**, the best algorithm among the proposed algorithms is Directional-IPeR-75 which increased it by 5%. Directional-IPeR 50, 25, and zero increased it by 33%, 33% and 47%, respectively. The increase of delay is due to reaching more destinations almost towards the end of the simulation time, which increased the average of the delay.

For more details see appendix C.

Conclusion of Setting 1

In conclusion, Directional-IPeR-75 is the best algorithm proposed for direction guidance in terms of F-measure and cost. At the same time, it performs better than IPeR in terms of delivery ratio, reached interested forwards, and F-measure. Besides, it enhances delay in low users' density environment. For the different densities of users, 200 users' density has the best results in terms of delivery ratio, reached interested forwarders, and delay. Furthermore, Directional-IPeR algorithms produce control messages to calculate nodes' directions. The least control messages are exerted by Directional-IPeR-75 for all user densities. It is illustrated in the TABLE 32.

TABLE 32: Performance of Directional-IPeR-75 for Setting 1

Algorithm	**Destinations**	**Forwarders**	**Delay**	**F-measure**
50 Users	**17.60**	**35.00**	**0.095**	**0.86**
100 Users	**17.88**	**35.33**	**0.058**	**0.86**
200 Users	**17.99**	**36.49**	**0.044**	**0.86**

This reveals that using Directional-IPeR-75 at the discrete uniform distribution of interest, the denser the environment is, the more delivery ratio, the more reached interested forwarders, and the less delay exerted by the algorithm.

Setting 2

50 Users

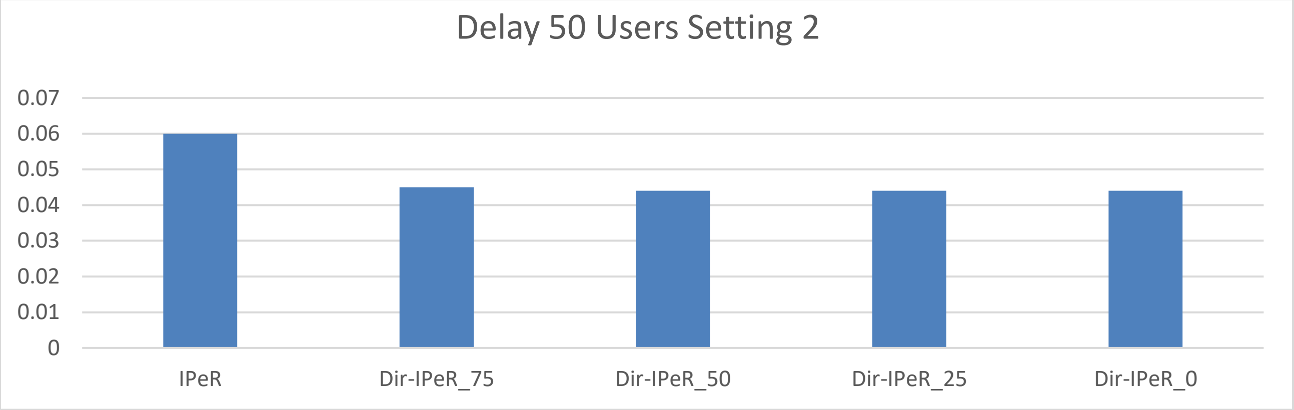


FIGURE 54: Delay of Setting 2, Directional-IPeR, 50 Users

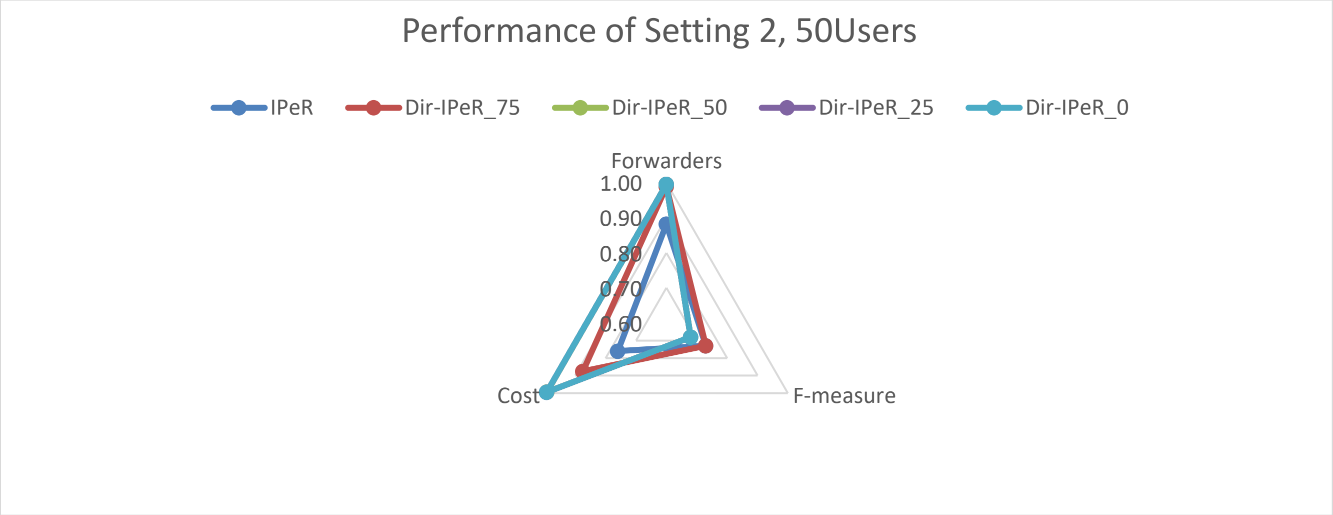


FIGURE 55: Performance of Setting 2, Directional-IPeR, 50Users

TABLE 33: Performance of Setting 2, Directional-IPeR, 50Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	43.20	0.060	0.73	38	0
Dir-IPeR_75	48.45	0.045	0.73	44	50
Dir-IPeR_50	48.75	0.044	0.68	50	68
Dir-IPeR_25	48.75	0.044	0.68	50	72
Dir-IPeR_0	48.75	0.044	0.68	50	73

100 Users

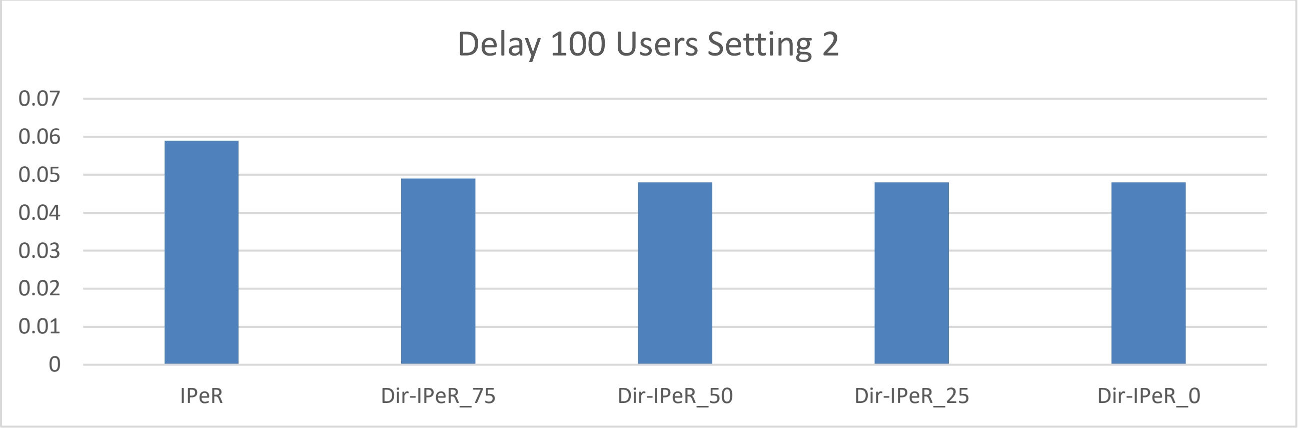


FIGURE 56: Delay of Setting 2, Directional-IPeR, 100 Users

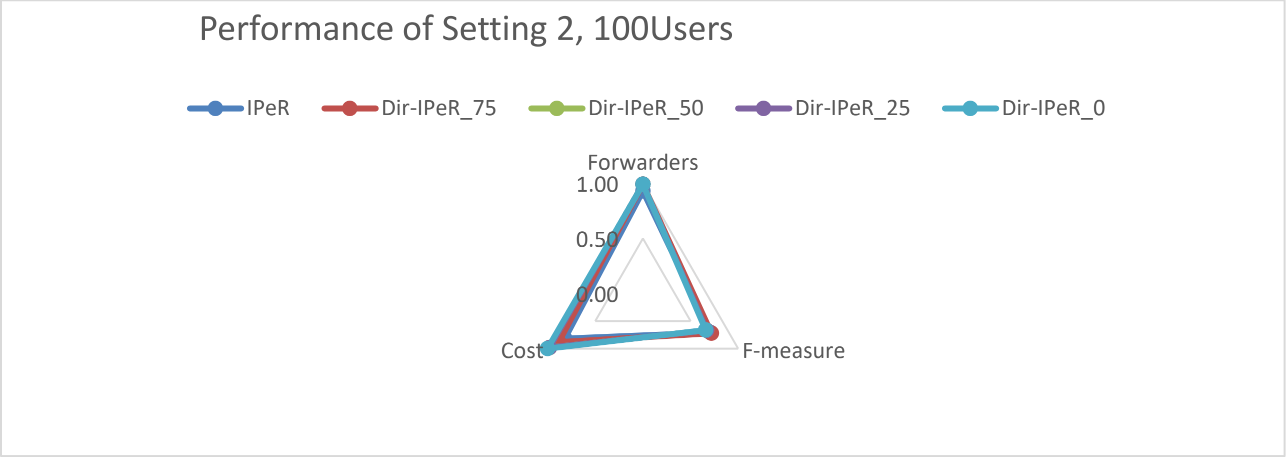


FIGURE 57: Performance of Setting 2, Directional-IPeR, 100Users

TABLE 34: Performance of Setting 2, Directional-IPeR, 100Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	45.58	0.059	0.71	82	0
Dir-IPeR_75	48.55	0.049	0.72	88	133
Dir-IPeR_50	48.55	0.048	0.67	98	194
Dir-IPeR_25	48.55	0.048	0.67	98	211
Dir-IPeR_0	48.55	0.048	0.66	100	222

200 Users

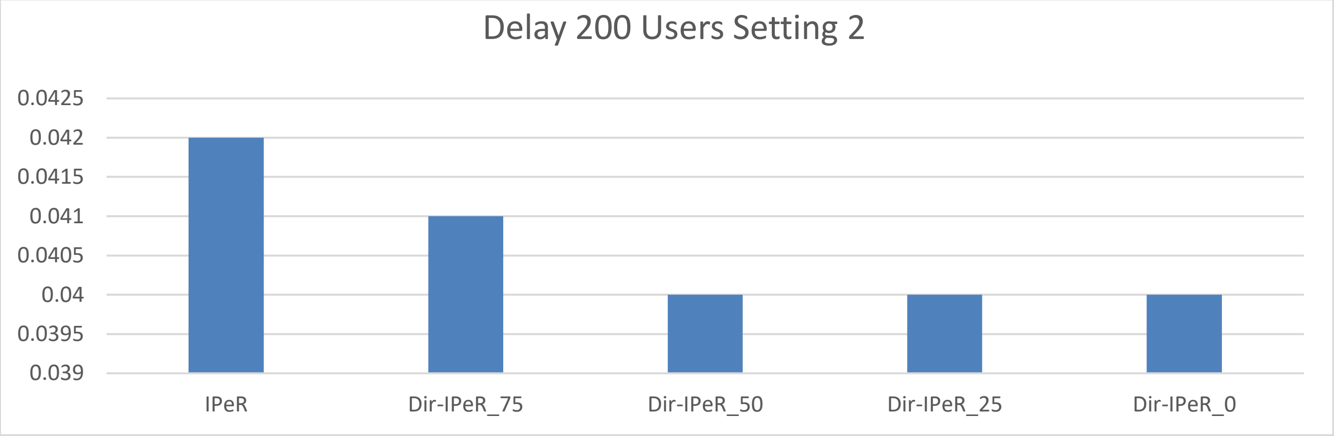


FIGURE 58: Delay of Setting 2, Directional-IPeR, 200 Users

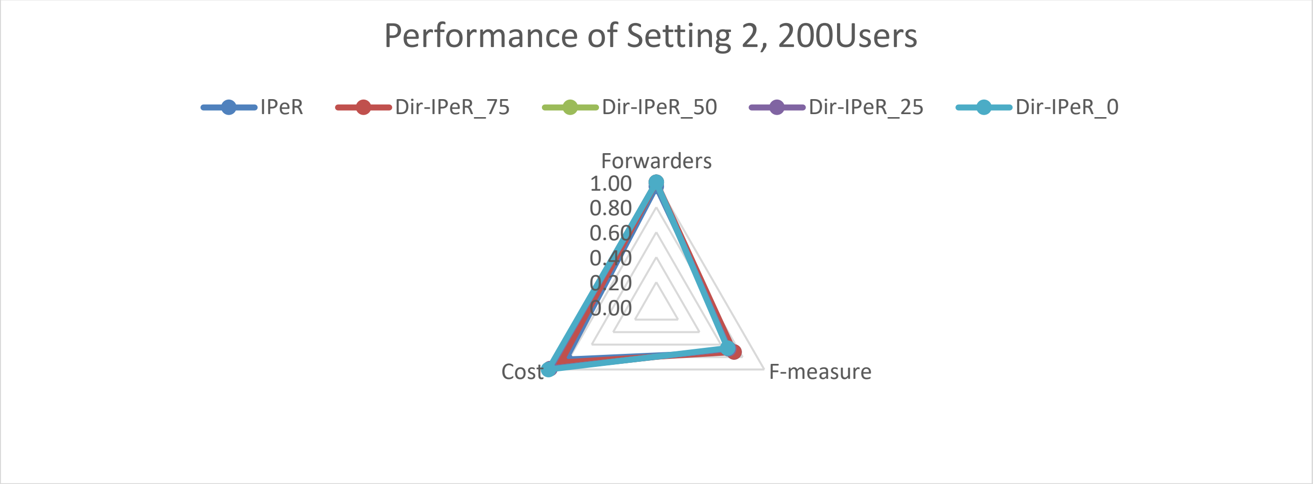


FIGURE 59: Performance of Setting 2, Directional-IPeR, 200Users

TABLE 35: Performance of Setting 2, Directional-IPeR, 200Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	47.36	0.042	0.72	168	0
Dir-IPeR_75	48.98	0.041	0.72	177	336
Dir-IPeR_50	48.98	0.040	0.67	197	515
Dir-IPeR_25	48.98	0.040	0.67	197	564
Dir-IPeR_0	48.98	0.040	0.66	200	595

Discussion of Results

For discrete normal distribution of interest, the delivery ratio is equal in all algorithms in all users' densities. For 50 users, Directional-IPeR-50, 25, and zero equally enhanced the number of reached interested forwarders by 13% and delay by 27% in comparison to IPeR. However, they increased the cost by 31%. On the other hand, Directional-IPeR-75 enhanced the number of reached interested forwarders by 12% and performed equally in delay in comparison to IPeR. It is the best algorithm in terms of cost as it increased it by 15% only. For F-measure, Directional-IPeR-75 and IPeR performed equally.

However, Directional-IPeR-50, 25, and zero equally decreased F-measure by 7%.

For 100 users, all variations of Directional-IPeR performed the same in respect to reached interested forwarders. They enhanced it by 7% compared to IPeR. For delay, Directional-IPeR-50, 25, and zero enhanced it by 19%. However, Directional-IPeR-75 decreased delay of 17%. For F-measure, Directional-IPeR 75 is the only algorithm that enhanced it by 1%. Directional-IPeR 50, 25 and zero decreased it by 6%, 6% and 7%. For the cost, the best algorithm among the proposed algorithms is Directional-IPeR-75 which increased it by 8%. Yet, Directional-IPeR-25, 50, and zero increased it by 20%, 20%, and 22%, respectively.

For 200 users, for the number of reached interested forwarders, all the proposed algorithms performed better than IPeR by 3%. In addition, they enhanced delay. Directional-IPeR-zero, 25, and 50 decreased delay by 5%. While Directional-IPeR-75 enhanced it by 2%. Directional-IPeR-75 performed equally to IPeR with respect to F-measure. However, the other algorithms decreased it by 7% and 8%. For the cost, the best algorithm among the proposed algorithms is Directional-IPeR-75 which increased it by 6%. Conversely, Directional-IPeR 50, 25, and zero increased it by 17%, 17%, and 19%, respectively.

For more details see appendix C.

Conclusion of Setting 2

For discrete normal distribution of interest, Directional-IPeR-75 is the best algorithm for direction guidance in terms of F-measure and cost in comparison to the proposed algorithms. Added, it performs better than IPeR in terms of reached interested forwards, and delay. For F-measure, it performs equally or better compared to IPeR. For the different densities of users, 200 users' density has the best results in terms of reached interested forwarders and delay. It is illustrated in the TABLE 36.

TABLE 36: Performance of Directional-IPeR-75 for Setting 2

Algorithm	Forwarders	Delay	F-measure
50 Users	48.45	0.045	0.73
100 Users	48.55	0.049	0.72
200 Users	48.98	0.041	0.72

This reveals that the denser the environment is, the more reached interested forwarders, and the less delay exerted by the algorithm.

Setting 3

<u>50 Users</u>

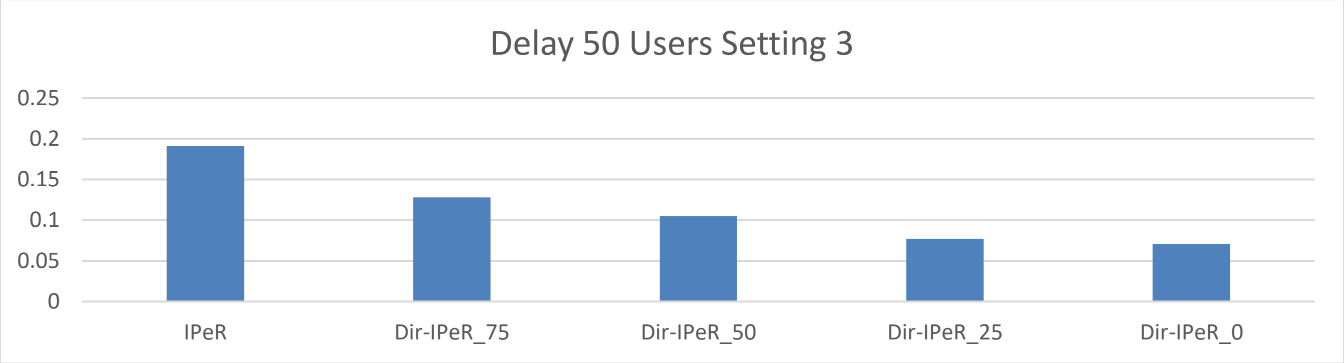


FIGURE 60: Delay of Setting 3, Directional-IPeR, 50 Users

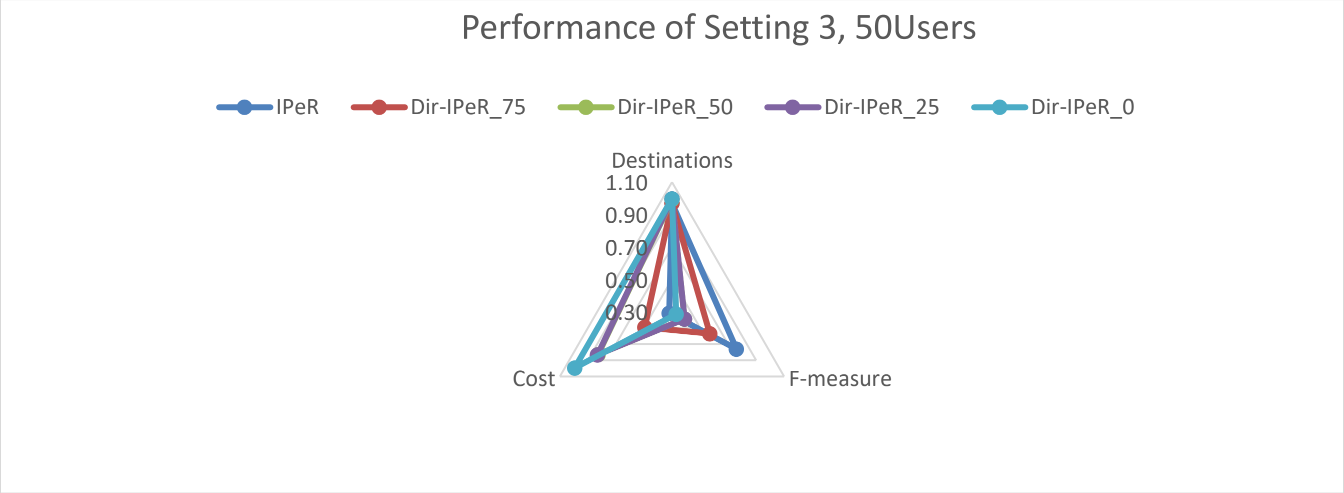


FIGURE 61: Performance of Setting 3, Directional-IPeR, 50Users

TABLE 37: Performance of Setting 3, Directional-IPeR, 50Users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.40	0.191	0.76	16	0
Dir-IPeR_75	19.40	0.128	0.57	25	25
Dir-IPeR_50	19.85	0.105	0.39	41	48
Dir-IPeR_25	19.95	0.077	0.39	42	51
Dir-IPeR_0	19.95	0.071	0.33	50	66

100 Users

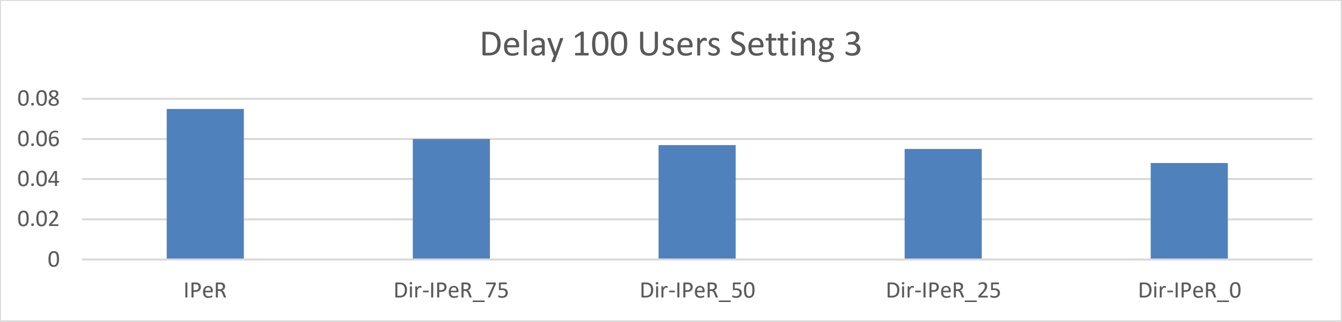


FIGURE 62: Delay of Setting 3, Directional-IPeR, 100 Users

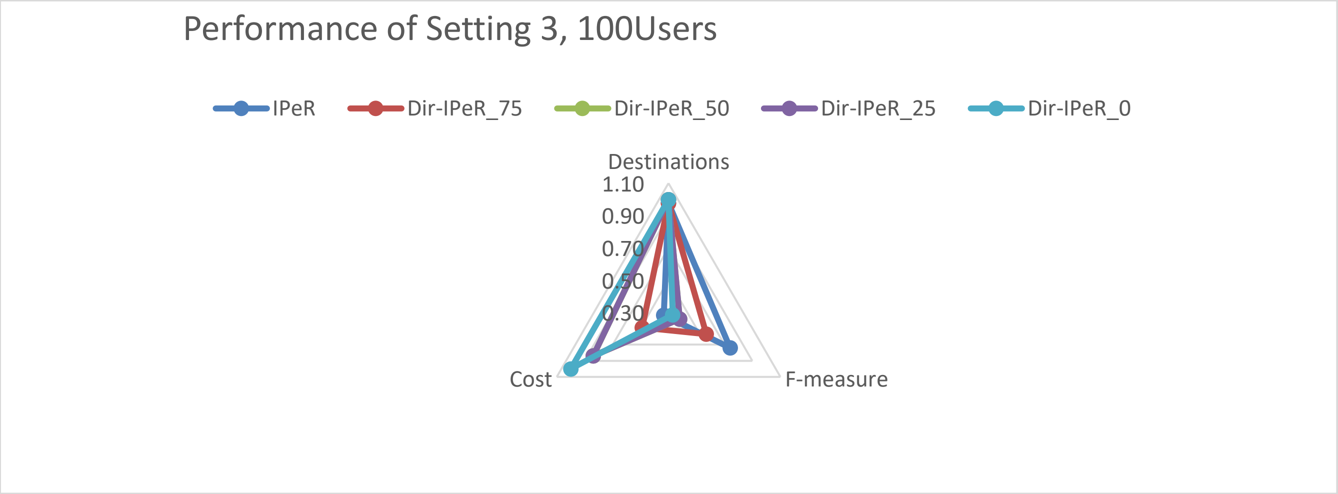


FIGURE 63: Performance of Setting 3, Directional-IPeR, 100Users

TABLE 38: Performance of Setting 3, Directional-IPeR, 100Users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.55	0.075	0.74	33	0
Dir-IPeR_75	19.55	0.060	0.57	49	54
Dir-IPeR_50	19.98	0.057	0.38	84	121
Dir-IPeR_25	19.98	0.055	0.38	84	129
Dir-IPeR_0	19.98	0.048	0.33	100	190

200 Users

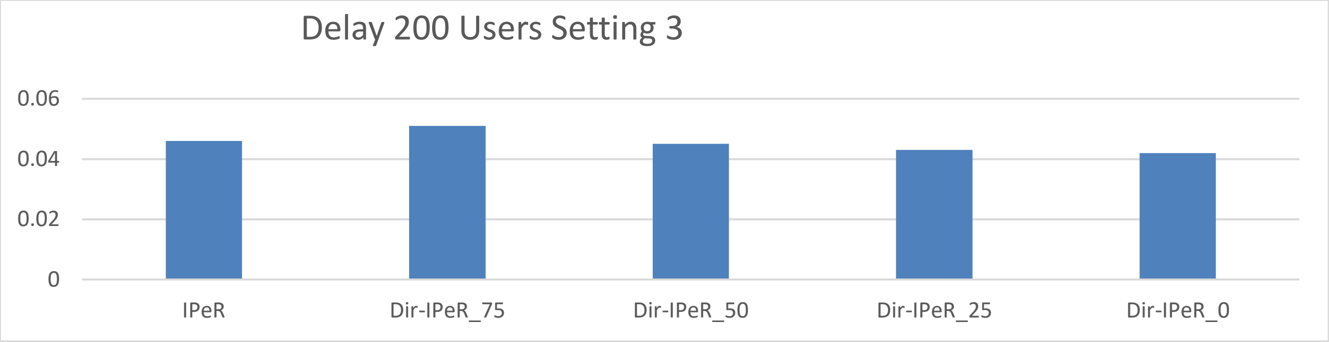


FIGURE 64: Delay of Setting 3, Directional-IPeR, 200 Users

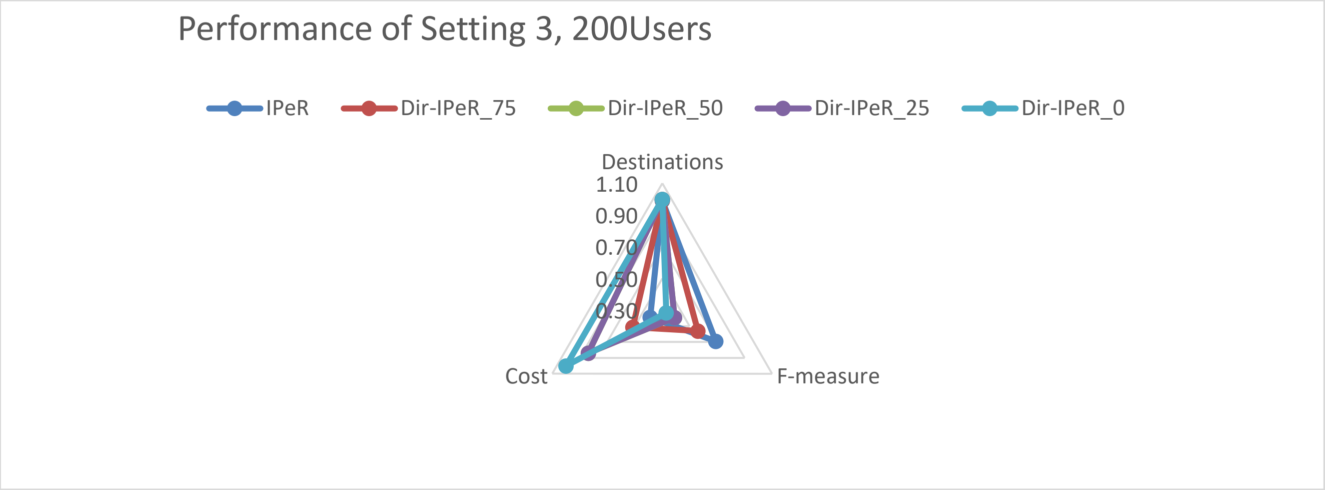


FIGURE 65: Performance of Setting 3, Directional-IPeR, 200Users

TABLE 39: Performance of Setting 3, Directional-IPeR, 200Users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.80	0.046	0.69	77	0
Dir-IPeR_75	19.90	0.051	0.56	103	122
Dir-IPeR_50	19.99	0.045	0.39	168	315
Dir-IPeR_25	19.99	0.043	0.39	168	333
Dir-IPeR_0	19.99	0.042	0.33	200	512

Discussion of Results

In this setting, there is no interested forwarder. *For 50 users*, Directional-IPeR-50, 25, and zero enhanced the **delivery ratio** by 2%, 3%, and 3% respectively. While Directional-IPeR-75 performed equally to IPeR. All the proposed algorithms enhanced **delay** in comparison to IPeR. Directional-IPeR-75, 50, 25, zero improved delay by 33%, 45%, 60% and 63%, respectively. For **F-measure** and **cost**, Directional-IPeR-75 performed the best compared to the proposed algorithms. Directional-IPeR-75, 50, 25, and zero decreased F-measure by 25%, 49%, 49% and 57%, respectively. In addition, they increased cost by 55%, 158% 160%, and 211% respectively.

For 100 users, all variations of Directional-IPeR performed the same in respect to **delivery ratio**, except Directional-IPeR-75 performed equally to IPeR, they enhanced it by 2% compared to IPeR. For **delay**, Directional-IPeR-75, 50, 25 and zero enhanced it by 20%, 24%, 27%, and 36%. For **F-measure** and **cost**, Directional-IPeR-75 performed the best compared to the proposed algorithms. For F-measure, Directional-IPeR-75, 50, 25 and zero decreased it by 23%, 49%, 49%, and 55%. For cost, Directional-IPeR-75, 50, 25 and zero increased it by 46%, 151%, 151% and 199%, respectively.

For 200 users, all the proposed algorithms performed better than IPeR by 1% in terms of **delivery ratio**. Directional-IPeR-50, 25 and zero decreased **delay** by 2%, 7%, and 9%. However, Directional-IPeR 75 increased it by 11%. For **F-measure** and **cost**, Directional-IPeR-75 performed the best compared to the proposed algorithms. Directional-IPeR-75, 50, 25 and zero decreased F-measure by 19%, 43%, 43% and 52%. For the cost, they increased it by 33%, 116%, 116%, and 158%, respectively. The increase of delay is due to reaching more destinations almost the end of the simulation time which increased the average of the delay. This behavior is consistent among the different settings.

For more details see appendix C.

Conclusion of Setting 3

For two disjoint subgraphs distribution of interest, Directional-IPeR-75 is the best algorithm proposed for direction guidance in terms of F-measure and cost. It performs better than IPeR in terms of delay in low users' density and delivery ratio in high users' density. For the different densities of users, 200 users' density has the best results in terms of delivery ratio, and delay. It is illustrated in the TABLE 40

TABLE 40: Performance of Directional-IPeR-75 Setting 3

Algorithm	Destinations	Delay	F-measure
50 Users	19.40	0.128	0.57
100 Users	19.55	0.060	0.57
200 Users	19.90	0.051	0.56

In this such challenging environment, where there are no interested forwarders, Directional-IPeR-75 performed better than IPeR as it can approach more delivery ratio with a slight increase of delay in dense environments and perform equally in terms of delivery ratio with decreased delay in sparse environments. For this reason, this setting is to be eliminated in the following experiments

4.2 Directional-IPeR with Random Discard

This algorithm is identical to Directional-IPeR. However, to simulate real life scenarios, we should expect that some uninterested nodes may refuse to participate in the message delivery process as they have no interest in this message content. We tried this attitude in Directional-IPeR. Consequently, there is no message delivery that is taking place from those nodes.

4.2.1 Results (Directional-IPeR with Random Discard)

Setting 1

<u>50 Users</u>

TABLE 41: Performance of Directional-IPeR 75 Random Setting 1, 50 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.60	29.90	0.107	0.85	29	0
Dir-IPeR_75	17.60	35.00	0.095	0.86	42	36
Dir-IPeR_75Int_Random	17.60	35.00	0.095	0.92	31	36

<u>100 Users</u>

TABLE 42: Performance of Directional-IPeR 75 Random Setting 1, 100 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.88	32.68	0.060	0.85	65	0
Dir-IPeR_75	17.88	35.33	0.058	0.86	70	81
Dir-IPeR_75Int_Random	17.88	35.33	0.058	0.90	64	81

<u>200 Users</u>

TABLE 43 : Performance of Directional-IPeR 75 Random Setting 1, 200 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.80	34.74	0.039	0.86	137	0
Dir-IPeR_75	17.99	36.49	0.044	0.86	144	189
Dir-IPeR_75Int_Random	17.99	36.49	0.044	0.92	129	189

Discussion of Results

For 50 users, Directional-IPeR-75-Random enhanced the number of reached interested forwarders by 17%, delay by 11%, and F-measure by 8%. While it performed equally to IPeR in terms of delivery ratio. However, it increased cost by 7% in comparison to IPeR. Directional-IPeR-75-Random enhanced F-measure by 7% and cost by 26% in comparison to Directional-IPeR-75. While they performed equally in terms of delivery ratio, the number of reached interested forwarders, and delay.

For 100 users, Directional-IPeR-75-Random enhanced the number of reached interested forwarders by 8%, delay by 3%, F-measure by 6%, and cost by 1% in comparison to IPeR. While it performed equally to IPeR in terms of delivery ratio. Directional-IPeR-75-Random enhanced F-measure by 5% and cost by 9% in comparison to Directional-IPeR-75. While they performed equally in terms of delivery ratio, the number of reached interested forwarders, and delay.

For 200 users, Directional-IPeR-75-Random enhanced delivery ratio by 1%, the number of reached interested forwarders by 5%, F-measure by 7%, and cost by 6%. However, the delay is increased by 13% in comparison to IPeR. Directional-IPeR-75-Random enhanced F-measure by 7% and cost by 10% in comparison to Directional-IPeR-75. While they performed equally in terms of delivery ratio, the number of reached interested forwarders, and delay. The delay is increased in this number of users as there is an increase in the number of destinations. These reached destinations are captured late in the simulation time. Taking the average is affected by those high numbers of seconds.

Conclusion of Setting 1

In conclusion, for discrete uniform distribution of interest, Directional-IPeR-75-random performed better in all metrics compared to IPeR except in the delivery ratio. It performed equally in sparse environments. It enhanced F-measure and cost compared to Directional-IPeR-75. The denser the environment is the more delivery ratio, the more number of reached interested forwarders, and the less delay.

Setting 2

50 Users

TABLE 44: Performance of Directional-IPeR 75 Random Setting 2, 50 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	43.20	0.060	0.73	38	0
Dir-IPeR_75	48.45	0.045	0.73	44	50
Dir-IPeR_75Int_Random	48.45	0.045	0.83	36	50

100 Users

TABLE 45 : Performance of Directional-IPeR 75 Random Setting 2, 100 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	45.58	0.059	0.71	82	0
Dir-IPeR_75	48.55	0.049	0.72	88	133
Dir-IPeR_75Int_Random	48.55	0.049	0.84	70	133

200 Users

TABLE 46: Performance of Directional-IPeR 75 Random Setting 2, 200 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	47.36	0.042	0.72	168	0
Dir-IPeR_75	48.98	0.041	0.72	177	336
Dir-IPeR_75Int_Random	48.98	0.041	0.83	139	336

Discussion of Results

For 50 users, Directional-IPeR-75-Random enhanced the number of reached interested forwarders by 12%, delay by 25%, F-measure by 14%, and cost by 5% in comparison to IPeR. On the other hand, Directional-IPeR-75-Random enhanced F-measure by 14% and cost by 18% in comparison to Directional-IPeR-75. While they performed equally in terms of the number of reached interested forwarders and delay.

For 100 users, Directional-IPeR-75-Random enhanced the number of reached interested forwarders by 7%, delay by 17%, F-measure by 18%, and cost by 15% in comparison to IPeR. It enhanced F-measure by 16% and cost by 120% in comparison to Directional-IPeR-75. While they performed equally in terms of the number of reached interested forwarders and delay.

For 200 users, Directional-IPeR-75-Random enhanced the number of reached interested forwarders and delay by 3%, F-measure by 15%, and cost by 17% in comparison to IPeR. Directional-IPeR-75-Random enhanced F-measure by 15%, cost by 21%, and delay by 1% in comparison to Directional-IPeR-75. While they performed equally in terms of the number of reached interested forwarders.

Conclusion of Setting 2

In conclusion, for discrete normal distribution of interest, Directional-IPeR-75-random performed better in all metrics compared to IPeR. It enhanced F-measure and cost compared to Directional-IPeR-75. It performs better in terms of the number of reached interested forwarders and delay in high-density environments

Conclusion on Directional-IPeR

For all distributions of interest and users' densities, Directional-IPeR-75, which is based on including nodes with IPeR value not less than 75% of the message holder's IPeR value, is the best algorithm proposed for direction guidance in terms of F-measure and cost.

For uniform distribution, it performs better than IPeR in terms of delivery ratio (up to 1% for 200-user experiments), reached interested forwarder nodes (up to 19% for 50-user experiments), and F-measure (up to 1% for 50 and 100 user experiments). In addition, it reduces delay (up to 11% for 50 users). For the different densities of users, the denser the environment is the more delivery ratio, the more reached interested forwarders, and the less delay exerted by the algorithm. For example, if we have two environments. one environment encompasses 100 users while the other environment has 1000 users. Within the latter environment, more destination nodes, more interested forwarders, and less delay are achieved.

For the normal distribution, it performs better than IPeR in terms of reached interested forwarder nodes (up to 13% for 50 users), and delay (up to 27% for 50 users). For F-measure, it performs equally or better compared to IPeR. For the different user densities, the denser the environment is, the more reached interested forwarders, and the less delay exerted by the algorithm .

For the two disjoint subgraphs distribution, it performs better than IPeR in terms of delay by 33% in low users' density and in terms of the delivery ratio by 1% in high users' density. The performance in dense environments. In this such challenging environment, where there are no interested forwarders, Directional-IPeR-75 performed better than IPeR as it can approach more delivery ratio with a slight increase of delay in dense environments and perform equally in terms of delivery ratio with decreased delay in sparse environments.

Directional-IPeR-75-random performed better in all metrics compared to IPeR. For uniform distribution, it enhanced F-measure (up to 8% for 50 users) and cost (up to 7% for 50 users). For the normal distribution, it enhanced F-measure (up to 18% for 100 users) and cost (up to 21% for 200 users). The denser the environment is, the more delivery ratio, the greater number of reached interested forwarders, and the less delay

CHAPTER 5: Hybrid of MIPeR and Directional-IPeR (DMIPeR)

Based on the best results of Directional-IPeR, MIPeR is merged with the best variation of Directional-IPeR. Therefore, two extra pieces of information are added to each node, namely, its contacts and its direction of motion. When two nodes meet, then specify that its value will be the MIPeR value, not the IPeR value. Then, a group of nodes is formulated. Each node in this group has their MIPeR value greater than the IPeR value of the message holder node. Based on that, if any direction is not covered by this group, other nodes which are expected to head to this direction get a copy of the message based on a predefined IPeR tolerance factor.

5.1 Steps of the Algorithm (DMIPeR)

1. Two extra pieces of information are added to each node. They are its contacts and direction.

2. When two nodes meet, the IPeR rank of the encountered node is calculated based on the harmonic mean of its IPeR and IPeR values of its contacts. If the harmonic mean of IPeR values of the nodes currently in contact is higher than the IPeR of the node itself, the harmonic mean is considered as the new IPeR value of the node. Otherwise, the IPeR value of the node remains unchanged. Finally, the message holder node sends a copy of the message only if the harmonic IPeR value of the encountered node is higher than its value.

3. The direction of the nodes is calculated as in Directional-IPeR. If any direction is not covered by the nodes which are selected based on harmonic mean IPeR, other nodes which are expected to head to this direction gets a copy of the message. However, these nodes must have an IPeR value that is equal to or greater than 75% of the IPeR value of the message holder node.

5.2 DMIPeR Algorithm

Function DMIPeR (runs on message holder node)

Input:

IPeRList ← all the nodes in contact with the message holder node and their IPeR value is >= to the message holder node's IPeR value

ContactList ← all the nodes in contact with the current node and their IPeR >= X% of message holder IPeR value, and their IPeR value< the message holder node's IPeR value

Output:

DMIPeRList

```
 2HarIPeRList ← 2HopHarmonicMean(IPeRList) [111]
 DMIPeRList ← Directional-IPeR(2HarIPeRList, ContactList)
```

5.3 Results

Settings 1

50 Users

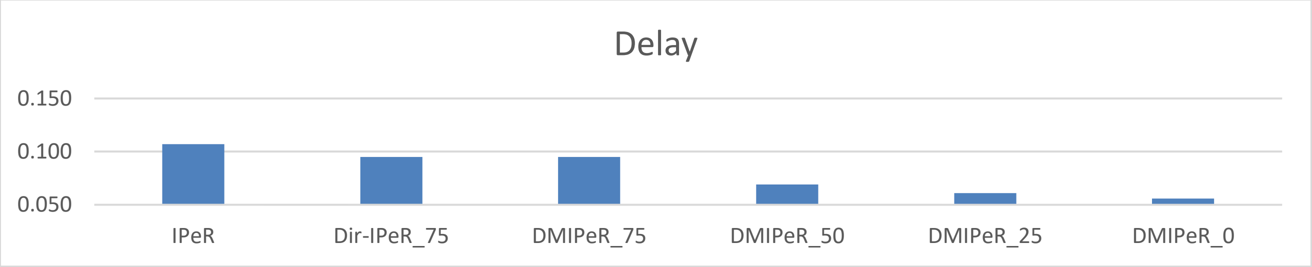


FIGURE 66: Delay of Setting 1, DMIPeR, 50 Users

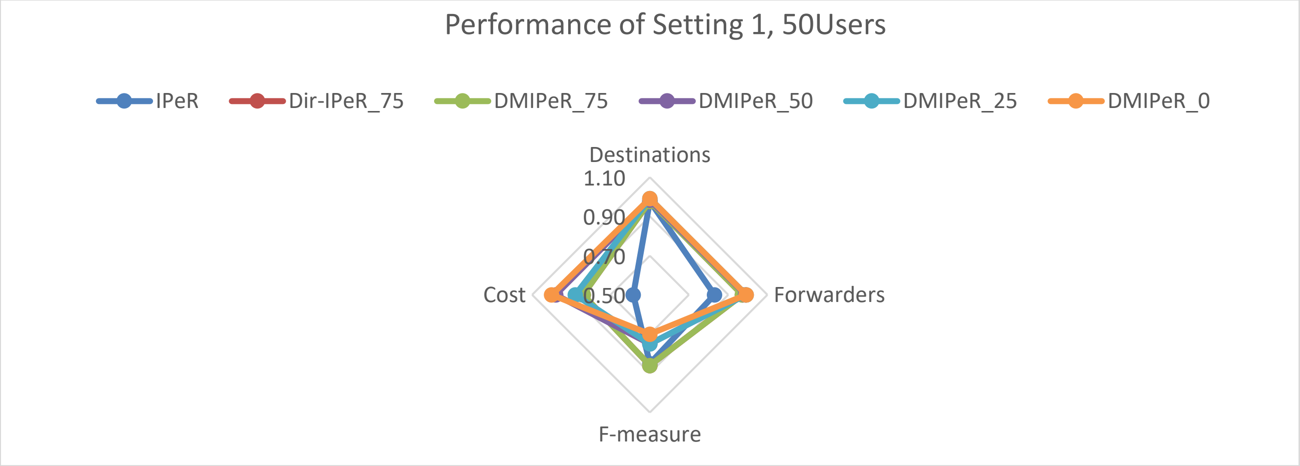


FIGURE 67: Delivery & F-measure of Setting 1, DMIPeR, 50 Users

TABLE 47: Performance of Setting 1, DMIPeR, 50 Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.60	29.90	0.107	0.85	29	0
Dir-IPeR_75	17.60	35.00	0.095	0.86	42	36
DMIPeR_75	17.60	35.00	0.095	0.86	42	36
DMIPeR_50	17.70	35.60	0.069	0.75	49	51
DMIPeR_25	17.85	35.70	0.061	0.75	44	54
DMIPeR_0	17.85	35.70	0.056	0.70	50	67

100 Users

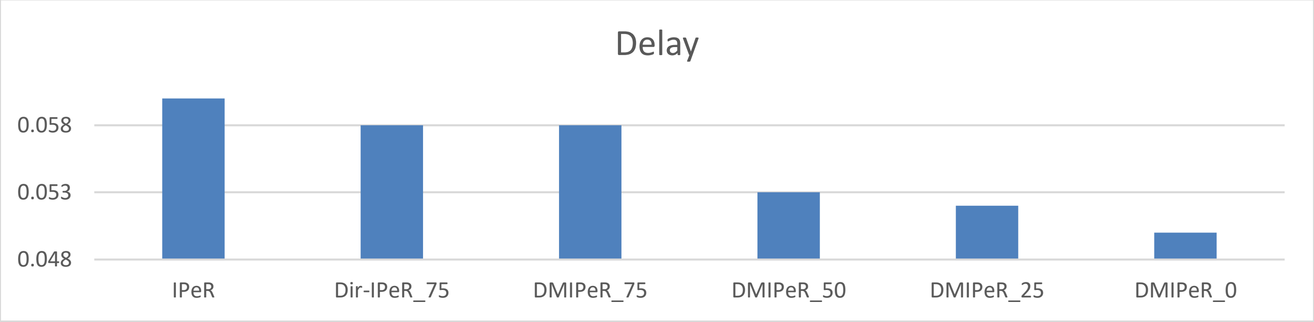


FIGURE 68: Delay & Cost of Setting 1, DMIPeR, 100 Users

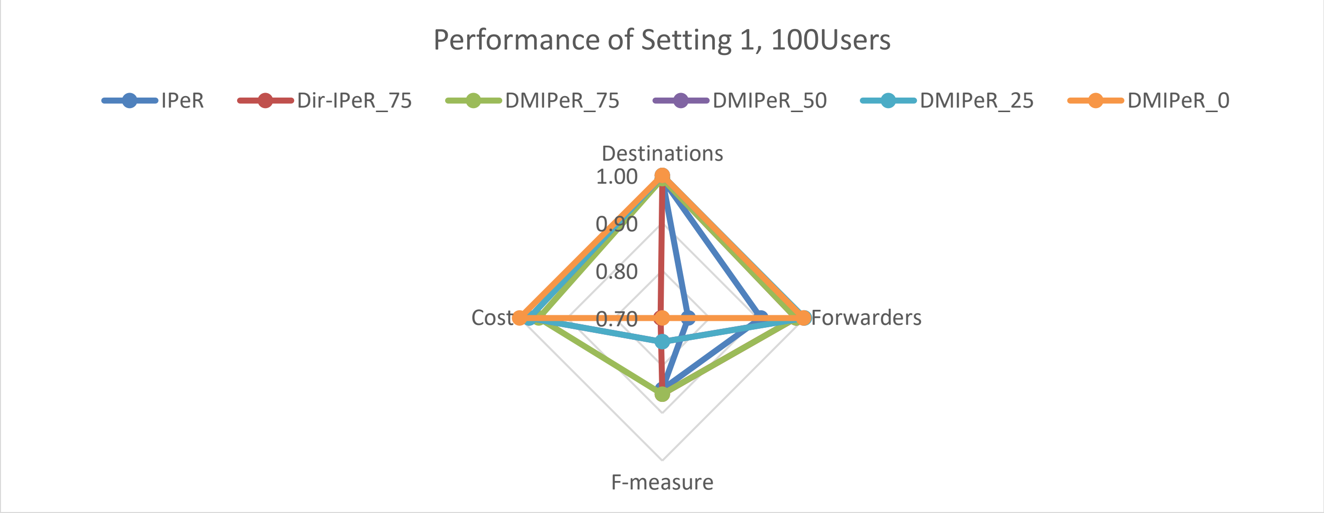


FIGURE 69: Delivery & F-measure of Setting 1, DMIPeR, 100 Users

TABLE 48: Performance of Setting 1, DMIPeR, 100 Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.88	32.68	0.060	0.85	65	0
Dir-IPeR_75	17.88	35.33	0.058	0.86	70	81
DMIPeR_75	17.88	35.33	0.058	0.86	96	81
DMIPeR_50	18.00	35.93	0.053	0.75	98	122
DMIPeR_25	18.00	35.98	0.052	0.75	98	138
DMIPeR_0	18.00	35.90	0.050	0.70	100	182

200 Users

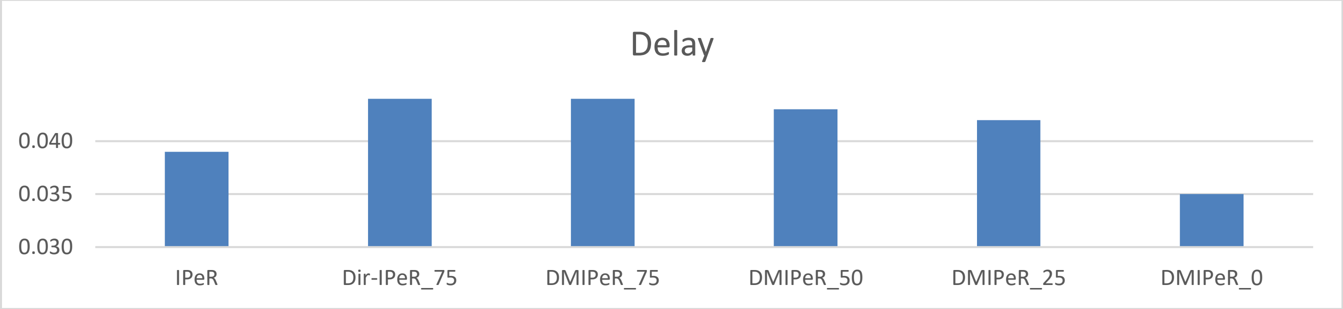


FIGURE 70: Delay & Cost of Setting 1, DMIPeR, 200 Users

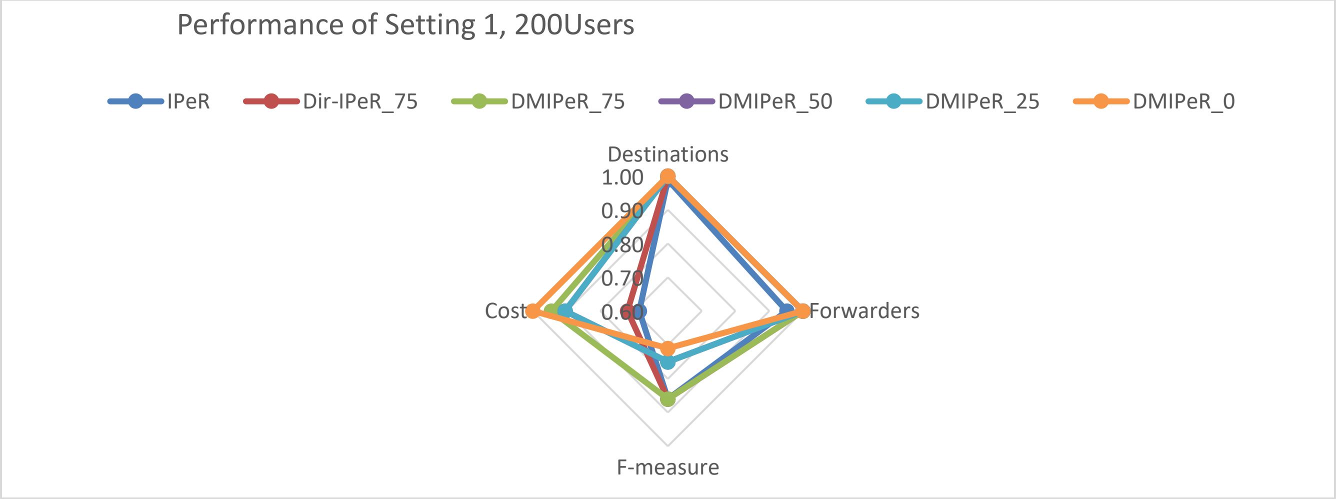


FIGURE 71: Delivery & F-measure of Setting 1, DMIPeR, 200 Users

TABLE 49: Performance of Setting 1, DMIPeR, 200 Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.80	34.74	0.039	0.86	137	0
Dir-IPeR_75	17.99	36.49	0.044	0.86	144	189
DMIPeR_75	17.99	36.49	0.044	0.86	189	229
DMIPeR_50	18.00	36.49	0.043	0.75	181	325
DMIPeR_25	18.00	36.49	0.042	0.75	181	389
DMIPeR_0	18.00	36.49	0.035	0.71	200	529

Discussion of Results

For discrete uniform distribution of interest, DMIPeR-75 was the best performance in terms of cost and F-measure compared to all the variations of DMIPeR in different users' densities.

For 50 users, DMIPeR-75 performed the same as Directional-IPeR-75 in all metrics.

For 100 users, DMIPeR-75 performed the same as Directional-IPeR-75 in all metrics except it increased cost by 36%.

For 200 users, DMIPeR-75 performed the same as Directional-IPeR-75 in all metrics except it increased cost by 31%. Here the trend of increasing the delay in this setting did not occur as it reached the last destination node early.

For more details see appendix D.

Conclusion of Setting 1

For all user densities for discrete uniform distribution of interest, **Directional-IPeR 75 performs the best in all metrics.**

Settings 2

50 Users

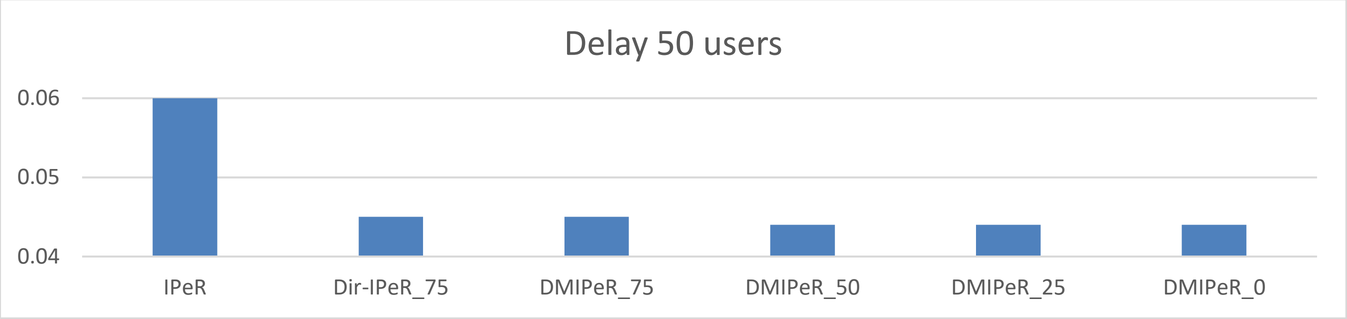


FIGURE 72: Delay & Cost of Setting 2, DMIPeR, 50 Users

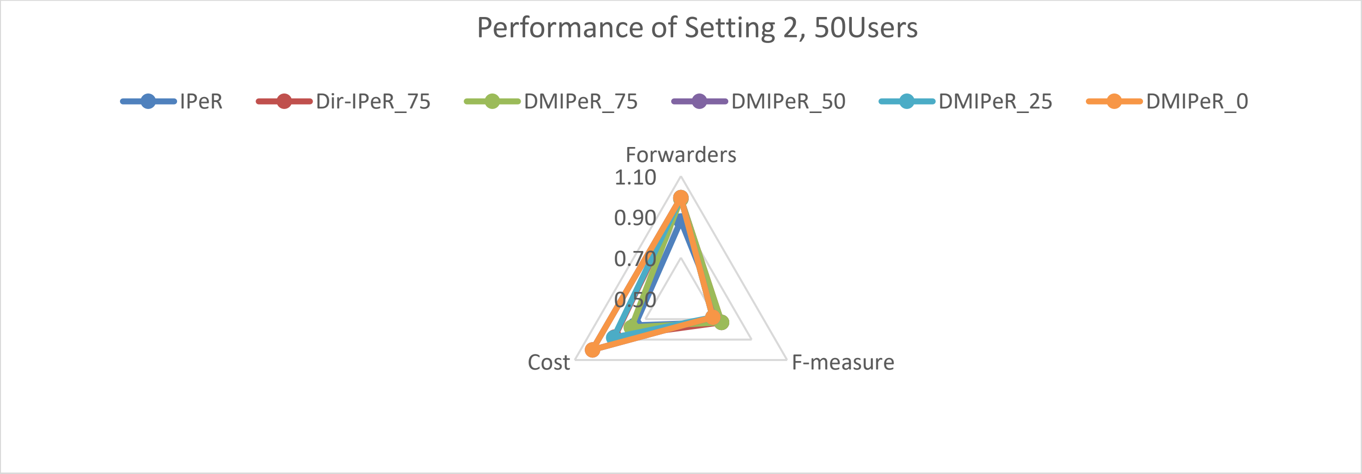


FIGURE 73: Delivery & F-measure of Setting 2, DMIPeR, 50 Users

TABLE 50: Performance of Setting 2, DMIPeR, 50 Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	43.20	0.060	0.73	38	0
Dir-IPeR_75	48.45	0.045	0.73	44	50
DMIPeR_75	48.45	0.045	0.73	49	50
DMIPeR_50	48.75	0.044	0.68	50	68
DMIPeR_25	48.75	0.044	0.68	50	72
DMIPeR_0	48.75	0.044	0.68	50	73

100 Users

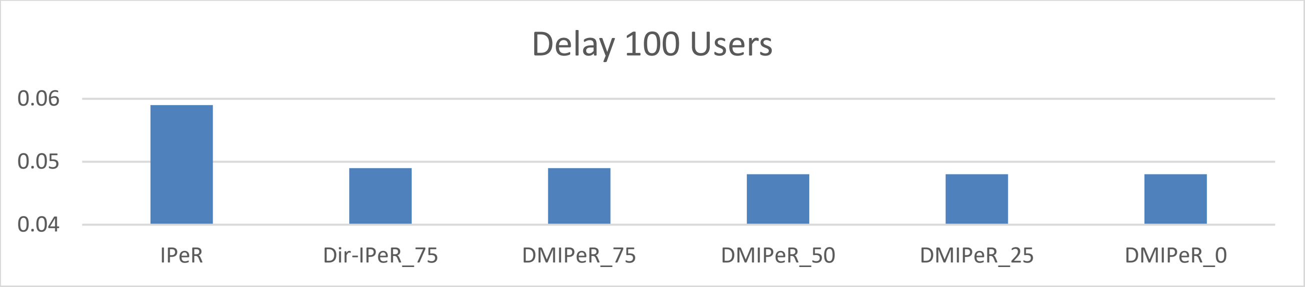


FIGURE 74: Delay & Cost of Setting 2, DMIPeR, 100 Users

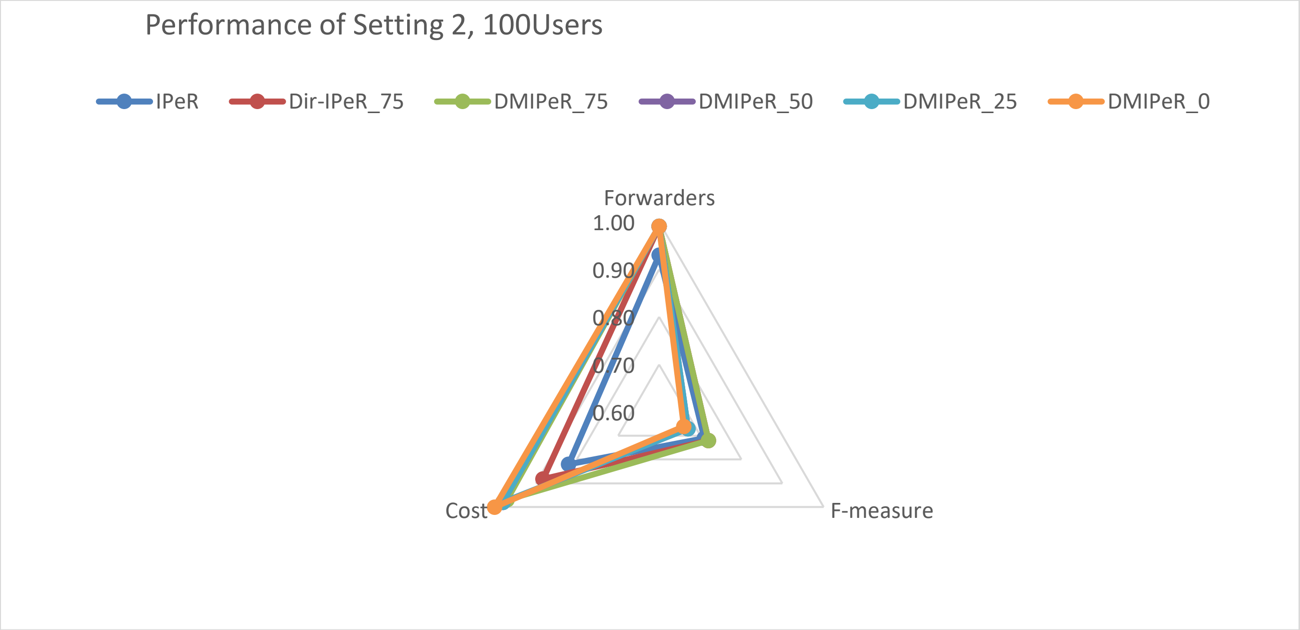


FIGURE 75: Delivery & F-measure of Setting 2, DMIPeR, 100 Users

TABLE 51: Performance of Setting 2, DMIPeR, 100 Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	45.58	0.059	0.71	82	0
Dir-IPeR_75	48.55	0.049	0.72	88	133
DMIPeR_75	48.55	0.049	0.72	98	133
DMIPeR_50	48.55	0.048	0.67	98	194
DMIPeR_25	48.55	0.048	0.67	98	211
DMIPeR_0	48.55	0.048	0.66	100	222

200 Users

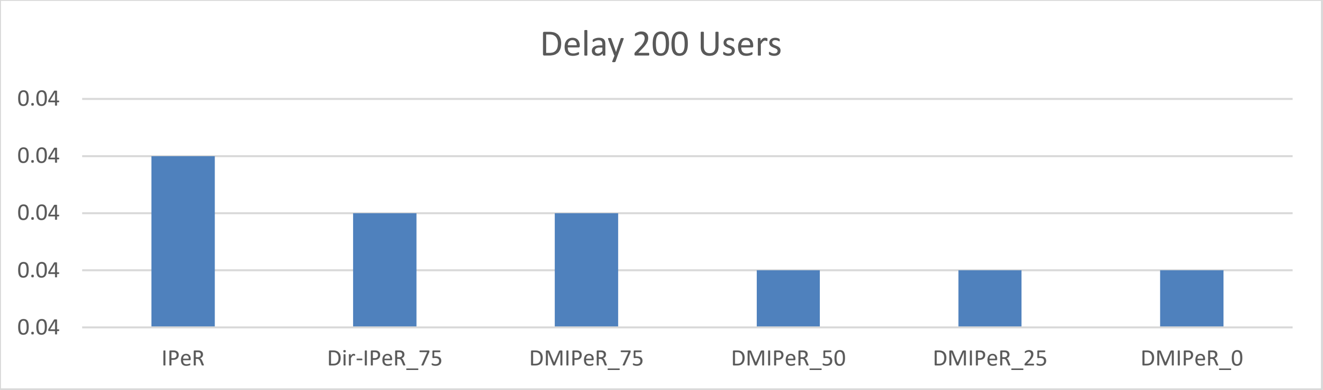


FIGURE 76: Delay & Cost of Setting 2, DMIPeR, 200 Users

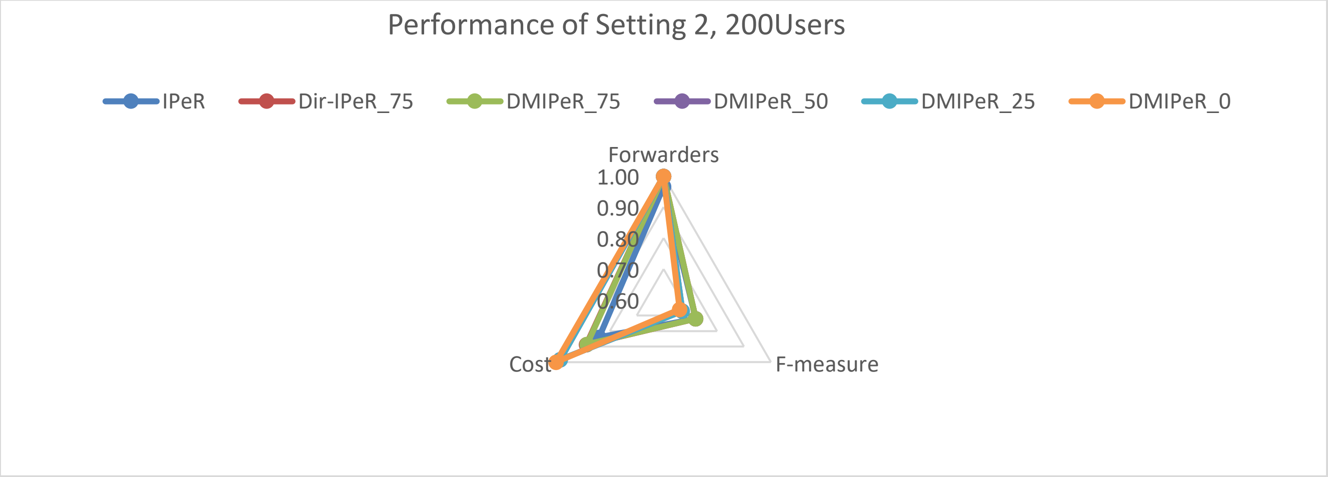


FIGURE 77: Delivery & F-measure of Setting 2, DMIPeR, 200 Users

TABLE 52: Performance of Setting 2, DMIPeR, 200 Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	47.36	0.042	0.72	168	0
Dir-IPeR_75	48.98	0.041	0.72	177	336
DMIPeR_75	48.98	0.041	0.72	197	304
DMIPeR_50	48.98	0.040	0.67	197	515
DMIPeR_25	48.98	0.040	0.67	197	564
DMIPeR_0	48.98	0.040	0.66	200	595

Discussion of Results

For discrete normal distribution of interest, DMIPeR-75 was the best performance in terms of cost and F-measure compared to all the variations of DMIPeR in different users' densities.

For 50 users, it performed equally to Directional-IPeR-75 in all metrics. However, it increased cost by 13%.

For 100 and 200 users, it performed equally to Directional-IPeR-75 in all metrics. However, it increased cost by 11%.

For more details see appendix D.

Conclusion of Setting 2

For all user densities for discrete normal distribution of interest, **Directional-IPeR-75 performs the best in all metrics.**

Settings 3

50 Users

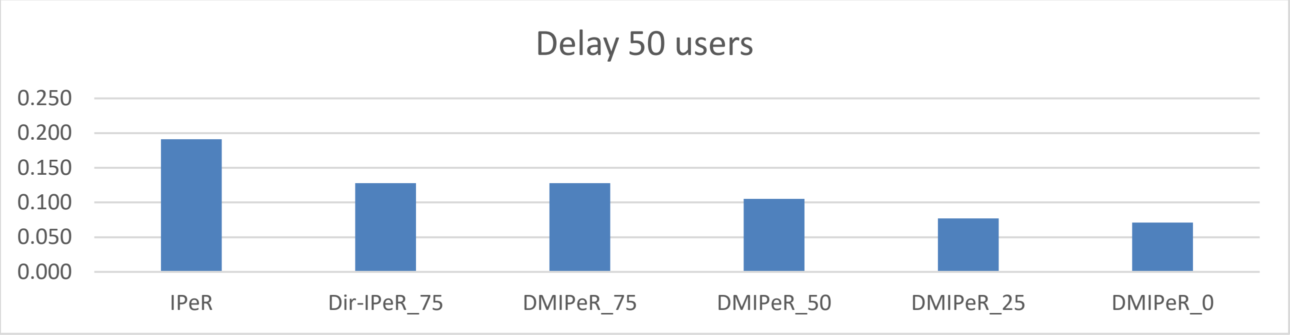


FIGURE 78: Delay of Setting3, DMIPeR, 50 Users

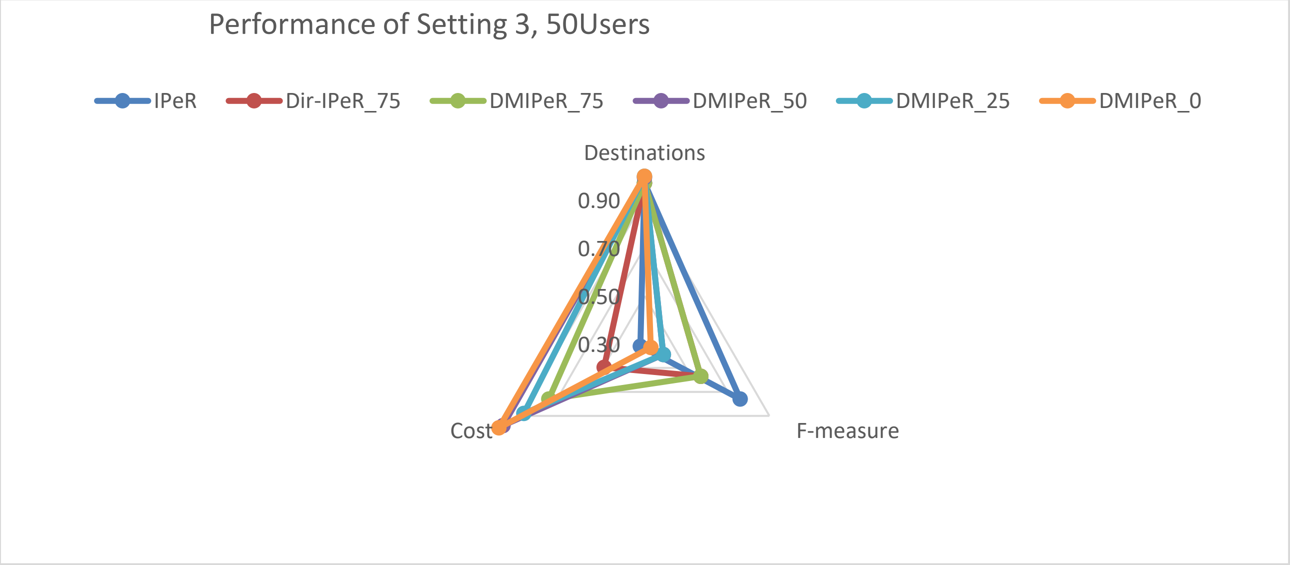


FIGURE 79: Delivery, Cost & F-measure of Setting 3, DMIPeR, 50 Users

TABLE 53: Performance of Setting 3, DMIPeR, 50 Users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.40	0.191	0.76	16.00	0
Dir-IPeR_75	19.40	0.128	0.57	24.73	25
DMIPeR_75	19.40	0.128	0.57	42.00	25
DMIPeR_50	19.85	0.105	0.39	49.00	48
DMIPeR_25	19.95	0.077	0.39	44.00	51
DMIPeR_0	19.95	0.071	0.33	50.00	66

<u>100 Users</u>

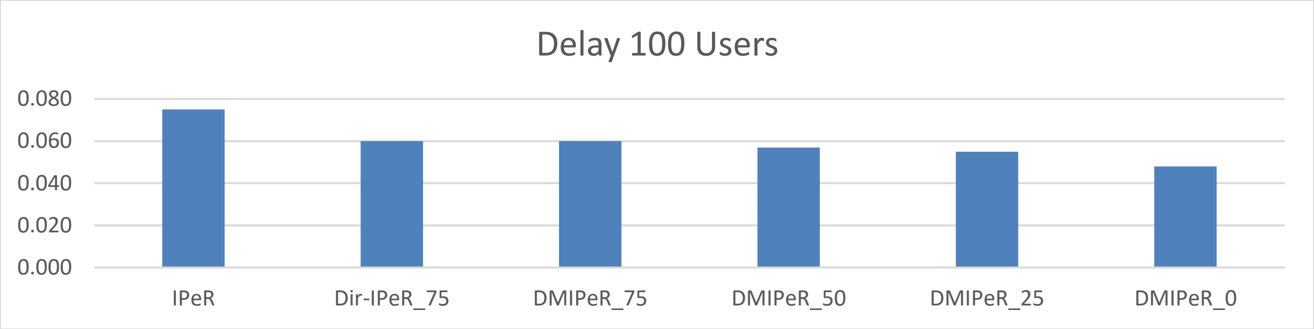


FIGURE 80: Delay & Cost of Setting 3, DMIPeR, 100 Users

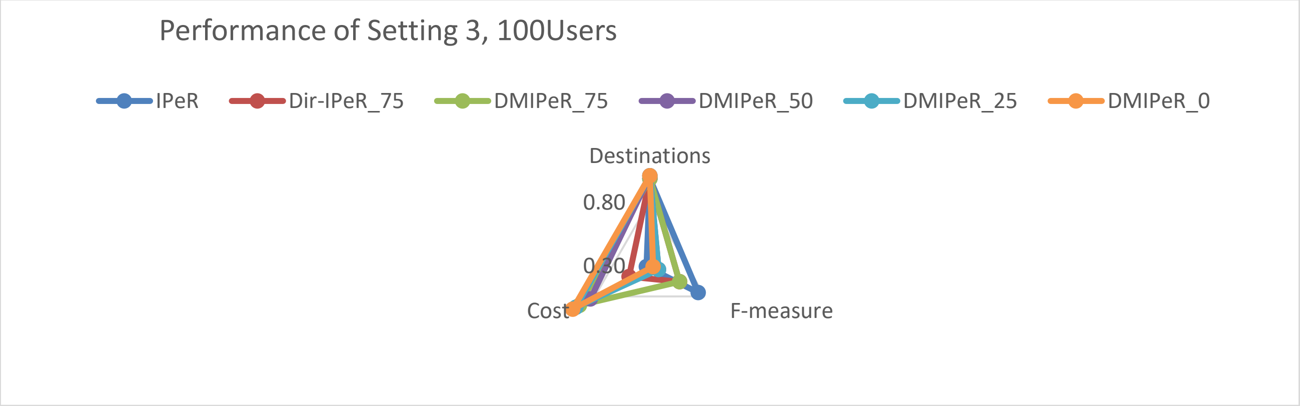


FIGURE 81: Delivery & F-measure of Setting 3, DMIPeR, 100 Users

TABLE 54: Performance of Setting 3, DMIPeR, 100 Users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.55	0.075	0.74	33	0
Dir-IPeR_75	19.55	0.060	0.57	49	54
DMIPeR_75	19.55	0.060	0.57	83	90
DMIPeR_50	19.98	0.057	0.38	84	121
DMIPeR_25	19.98	0.055	0.38	97	129
DMIPeR_0	19.98	0.048	0.33	100	190

200 Users

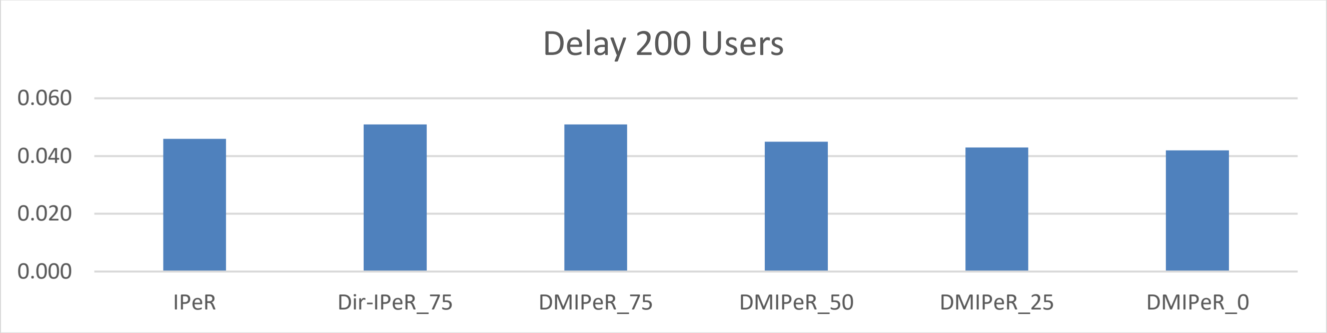


FIGURE 82: Delay & Cost of Setting 3, DMIPeR, 200 Users

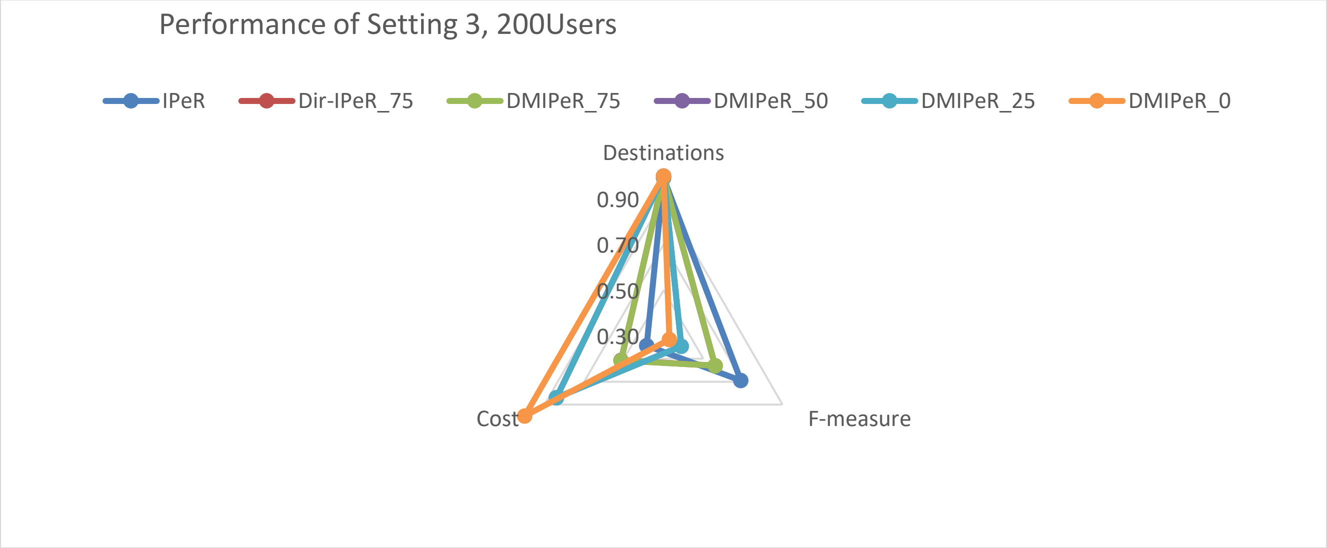


FIGURE 83: Delivery & F-measure of Setting 3, DMIPeR, 200 Users

TABLE 55: Performance of Setting 3, DMIPeR, 200 Users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.80	0.046	0.69	77	0
Dir-IPeR_75	19.90	0.051	0.56	103	122
DMIPeR_75	19.90	0.051	0.388	166	191
DMIPeR_50	19.99	0.045	0.39	168	315
DMIPeR_25	19.99	0.043	0.39	168	333
DMIPeR_0	19.99	0.042	0.33	200	512

Discussion of Results

For two disjoint subgraphs distribution of interest, DMIPeR-75 was the best performance in terms of cost and F-measure compared to all the variations of DMIPeR in different users' densities.

For 50 and 100 users, it performed equally to Directional-IPeR-75 in all metrics. However, it increased cost by 70%.

For 200 users, it performed equally to Directional-IPeR-75 in all metrics. However, it increased cost of 61%.

For more details see appendix D.

Conclusion of Setting 3

For all user densities for discrete normal distribution of interest, **Directional-IPeR-75 performs the best in all metrics.**

Conclusion on Hybrid of MIPeR and Directional-IPeR (DMIPeR)

Directional-IPeR-75, which is the best algorithm for Directional-IPeR, performs equal to its corresponding DMIPeR in all metrics. However, it increases cost (up to 36% for 100 users' density in Uniform Interest Distribution, up to 13% for 50 users' density in Normal Interest Distribution, up to 61% for 200 users' density in 2 Disjoint Subgraphs Interest Distribution). Adding 2 hops did not gain any enhancement in all metrics.

CHAPTER 6: Adding Directional Guidance to Other Related Algorithms

To authenticate the performance of Directional-IPeR, direction awareness is integrated to other algorithms that IPeR proved to be their superior. They include Interest aware algorithm, PeopleRank and PeopleRank integrated with various percentages of interest.

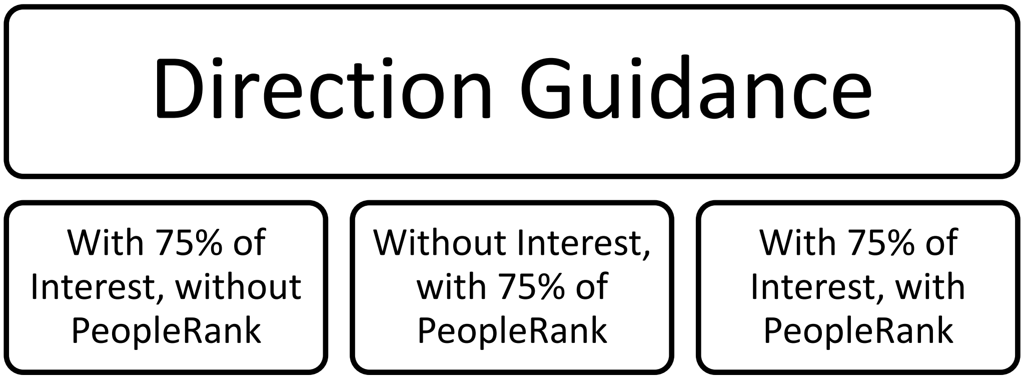

Direction Guidance
With 75% of Interest, without PeopleRank
Without Interest, with 75% of PeopleRank
With 75% of Interest, with PeopleRank

FIGURE 84: Adding Direction guidance to other algorithms

6.1 Directional-Interest-75

1. PeopleRank of the node is not calculated or considered
2. Interest is more than 75% of Interest of the message holder node.

6.2 Directional-PeopleRank-75

1. PeopleRank is more than 75% of Interest of the message holder node.
2. Interest of the node is not calculated or considered

6.3 Directional-PeopleRank-Interest-75

1. PeopleRank of the node is calculated or considered
2. Interest is more than 75% of Interest of the message holder node.

Results

Setting 1

50 Users

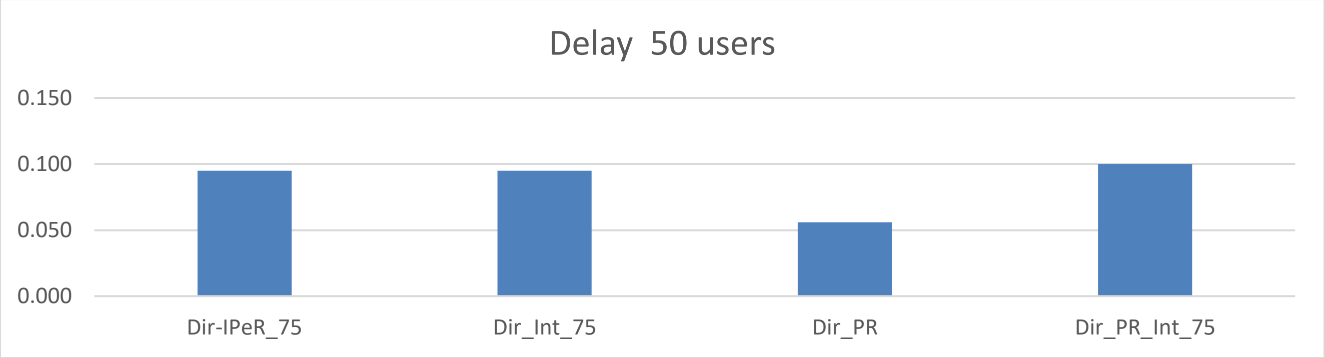


FIGURE 85: Delay & Cost of Setting 1, Directional-Interest, 50 Users

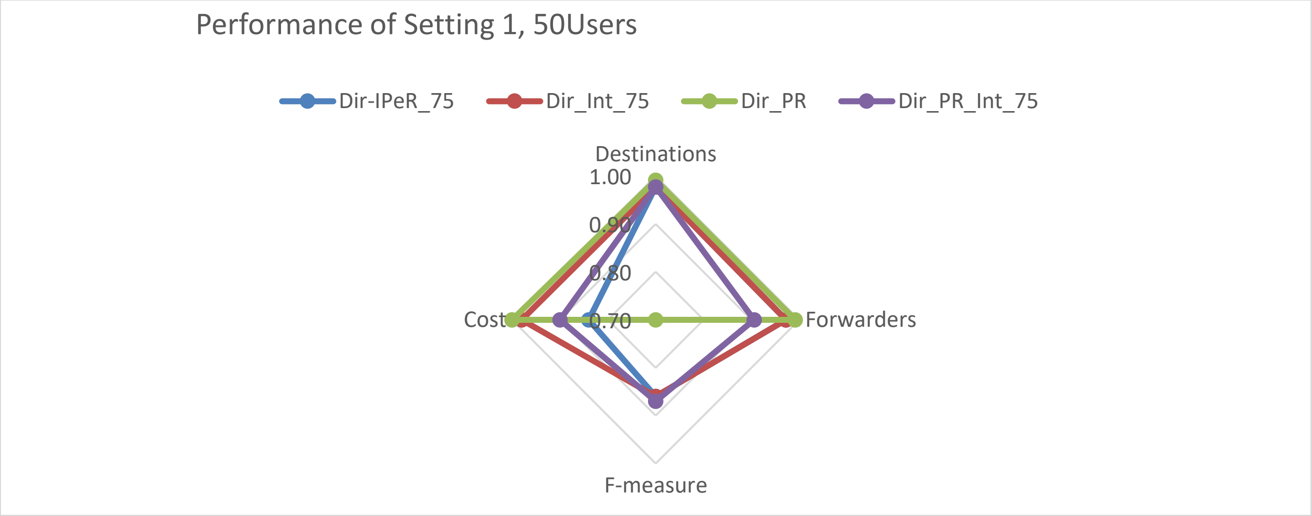


FIGURE 86: Delivery & F-measure of Setting 1, Directional-Interest, 50 Users

TABLE 56: Performance of Setting 1, Directional-Interest, 50Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.60	29.90	0.107	0.85	29	0
Dir-IPeR_75	17.60	35.00	0.095	0.86	42	36
Dir-Int-75	17.60	35.00	0.095	0.86	49	34
Dir-PR	17.85	35.70	0.056	0.70	50	49
Dir-PR-Int-75	17.60	32.60	0.100	0.87	45	31

100 Users

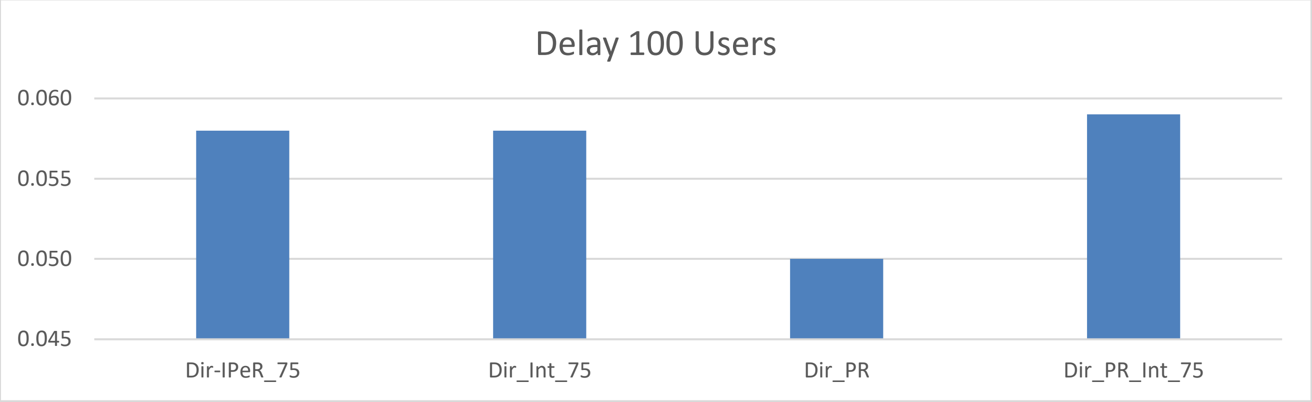


FIGURE 87: Delay of Setting 1, Directional-Interest, 100 Users

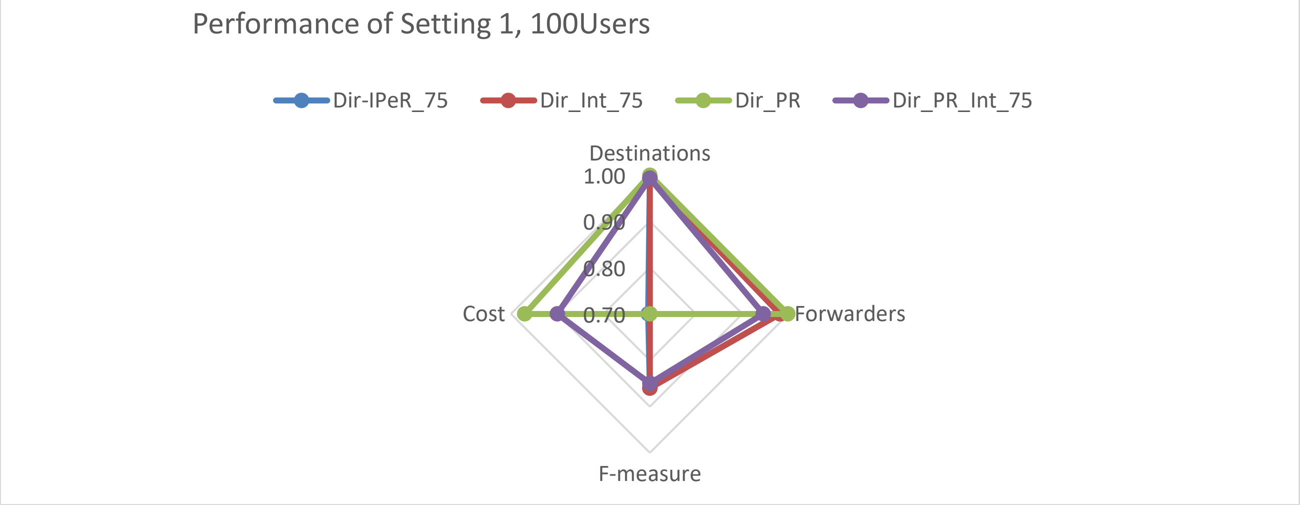


FIGURE 88: Delivery, Cost & F-measure of Setting 1, Directional-Interest, 100 Users

TABLE 57: Performance of Setting 1, Directional-Interest, 100Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.88	32.68	0.060	0.85	65	0
Dir-IPeR_75	17.88	35.33	0.058	0.86	70	81
Dir-Int-75	17.88	35.33	0.058	0.86	70	70
Dir-PR	18.00	35.93	0.050	0.70	97	99
Dir-PR-Int-75	17.88	34.03	0.059	0.85	90	66

200 Users

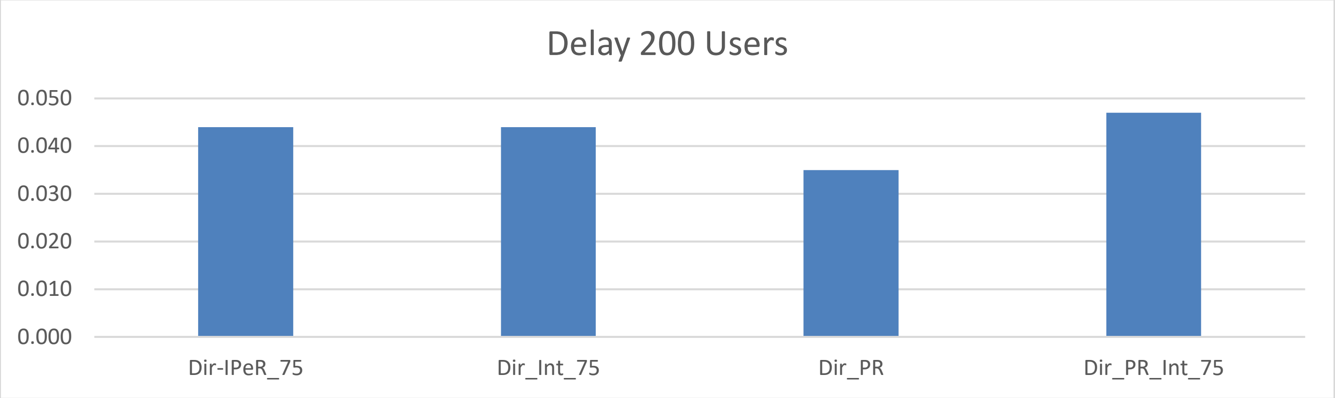


FIGURE 89: Delay of Setting 1, Directional-Interest, 200 Users

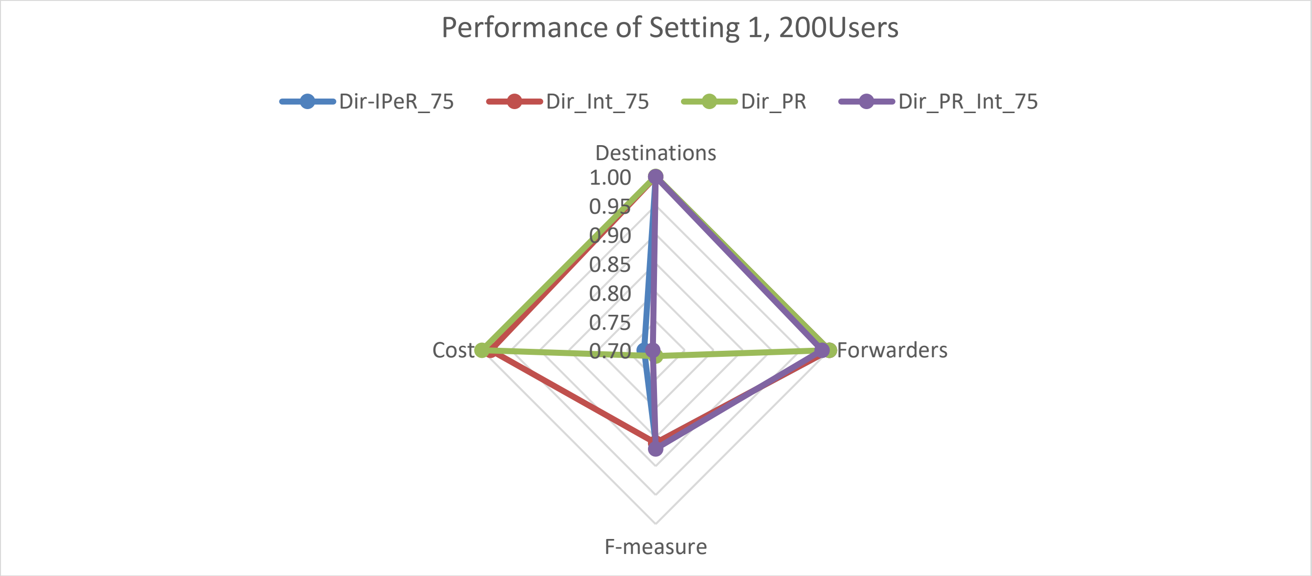


FIGURE 90: Delivery, Cost & F-measure of Setting 1, Directional-Interest, 200 Users

TABLE 58: Performance of Setting 1, Directional-Interest, 200Users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.80	34.74	0.039	0.86	137	0
Dir-IPeR_75	17.99	36.49	0.044	0.86	144	189
Dir-Int-75	17.99	36.49	0.044	0.86	144	197
Dir-PR	18.00	36.49	0.035	0.71	200	199
Dir-PR-Int-75	17.99	36.04	0.047	0.87	141	140

Discussion of Results

For 50 users, *Directional-PeopleRank* flooded the network by 100% data messages and 98% control messages. Consequently, it performed better compared to Directional-IPeR-75 in terms of delivery ratio, the number of reached interested forwarders, and delay by 1%, 2%, and 41%, respectively. However, it decreased F-measure by 19%.

Directional-Interest-75 performed equally to Directional-IPeR-75 in all metrics except that it increased cost by 17%.

Directional-PeopleRank-Interest-75 performed better in terms of F-measure by 1%. It performed equally in the delivery ratio. However, it performed worse in terms of the number of reached interested forwarders, delay, and cost by 7%, 5%, and 7%, respectively.

For 100 users, *Directional-PeopleRank* flooded the network by having a cost of 97% and control messages by 99%. Consequently, F-measure is decreased by 19%, and other metrics are enhanced by 1% for delivery ratio, 2% for the number of reached interested forwarders, and 14% for the delay.

Directional-Interest-75 performed almost exactly to Directional-IPeR-75 in all metrics.

Directional-PeopleRank-Interest-75 performed equally in the delivery ratio. However, it performed worse in terms of the number of reached interested forwarders, delay, cost, and F-measure by 4%, 2%, 28%, and 1%, respectively.

For 200 users, *Directional-PeopleRank* flooded the network by having the cost and control messages of 100%. Consequently, F-measure is decreased by 1%. It performed better compared to Directional-IPeR-75 in terms of delivery ratio, the number of reached interested forwarders, and delay by 1%, 1%, and 7%, respectively.

Directional-Interest-75 performed equally to Directional-IPeR-75 in all metrics except that it increased cost by 24%.

Directional-PeopleRank-Interest-75 performed equally in the delivery ratio. However, it performed worse in terms of the number of reached interested forwarder, delay, cost, and F-measure by 1%, 7%, 3%, and 2%, respectively.

Setting 2

50 Users

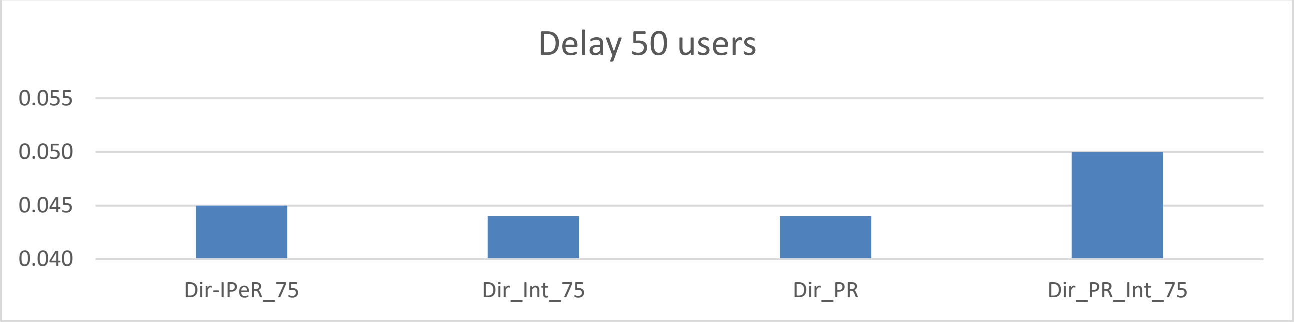


FIGURE 91: Delay of Setting 2, Directional-Interest, 50 Users

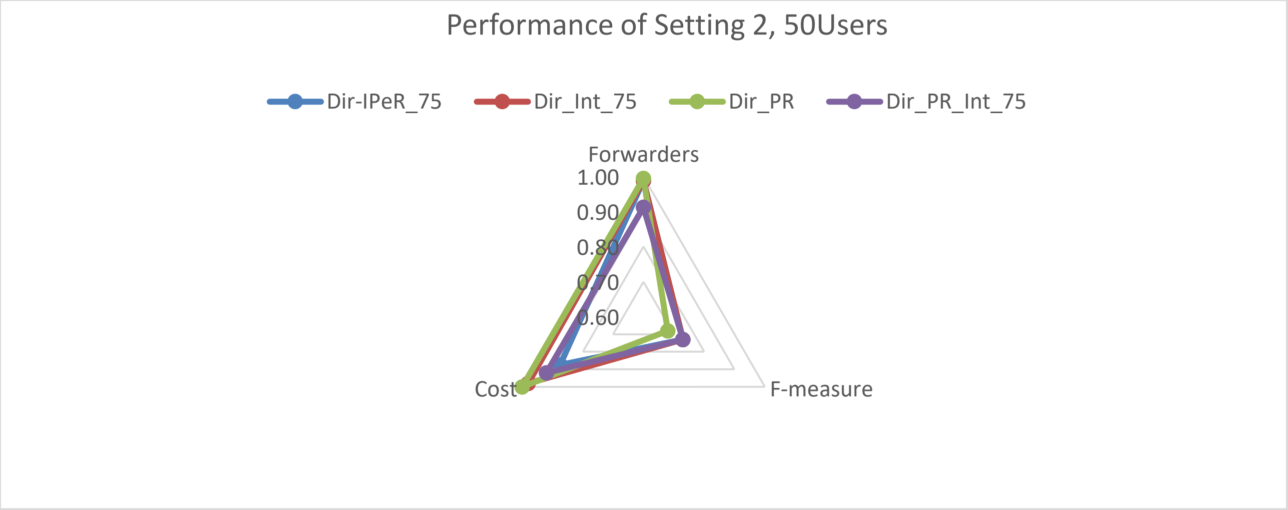


FIGURE 92: Delivery, Cost & F-measure of Setting 2, Directional-Interest, 50 Users

TABLE 59: Performance of Setting 2, Directional-Interest, 50Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	43.20	0.060	0.73	38	0
Dir-IPeR_75	48.45	0.045	0.73	44	50
Dir_Int_75	48.45	0.044	0.73	49	43
Dir_PR	48.75	0.044	0.68	50	49
Dir_PR_Int_75	44.70	0.050	0.73	46	39

100 Users

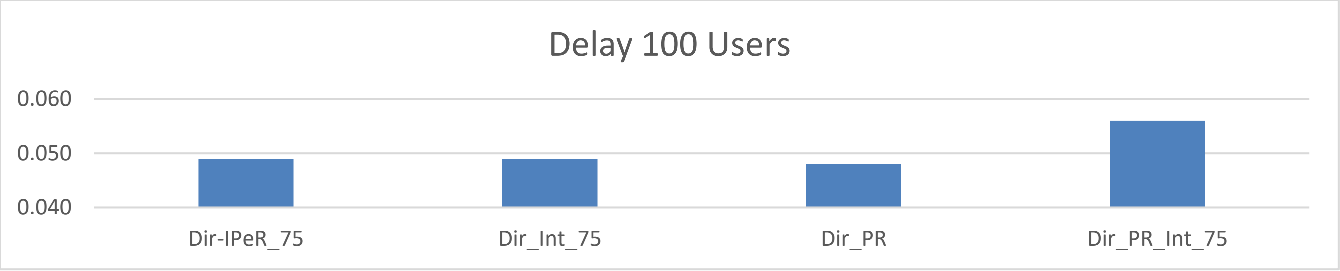


FIGURE 93: Delay of Setting 2, Directional-Interest, 100 Users

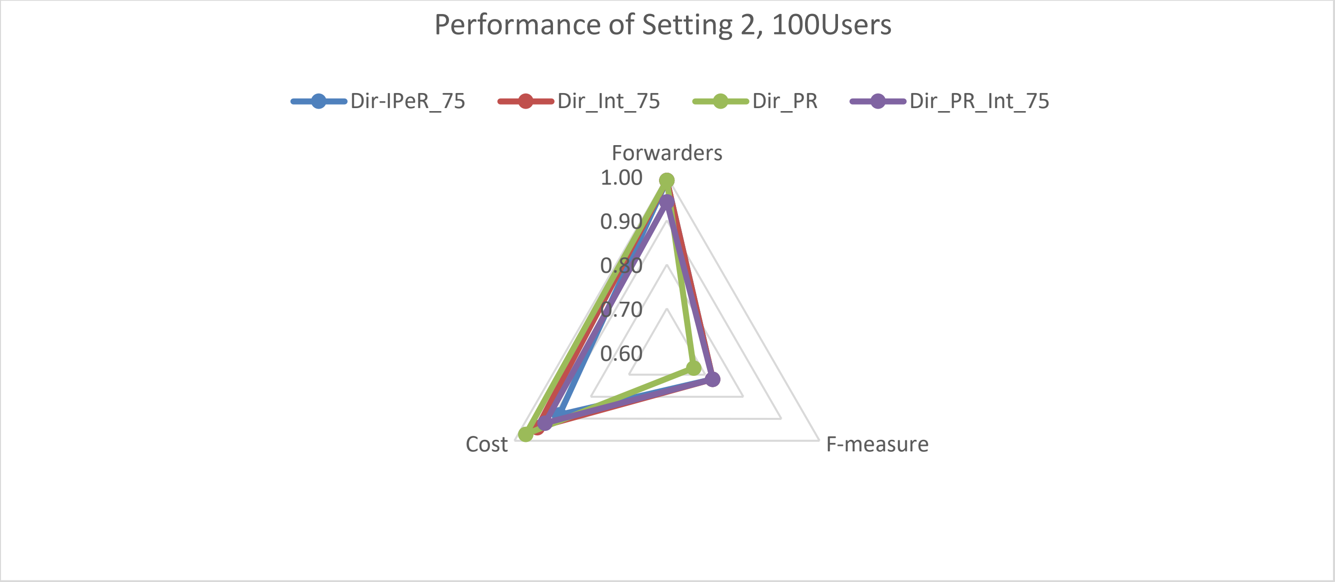


FIGURE 94: Delivery, Cost & F-measure of Setting 2, Directional-Interest, 100 Users

TABLE 60: Performance of Setting 2, Directional-Interest, 100Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	45.58	0.059	0.71	82	0
Dir-IPeR_75	48.55	0.049	0.72	88	133
Dir_Int_75	48.55	0.049	0.72	94	88
Dir_PR	48.55	0.048	0.67	97	99
Dir_PR_Int_75	46.15	0.056	0.72	92	83

200 Users

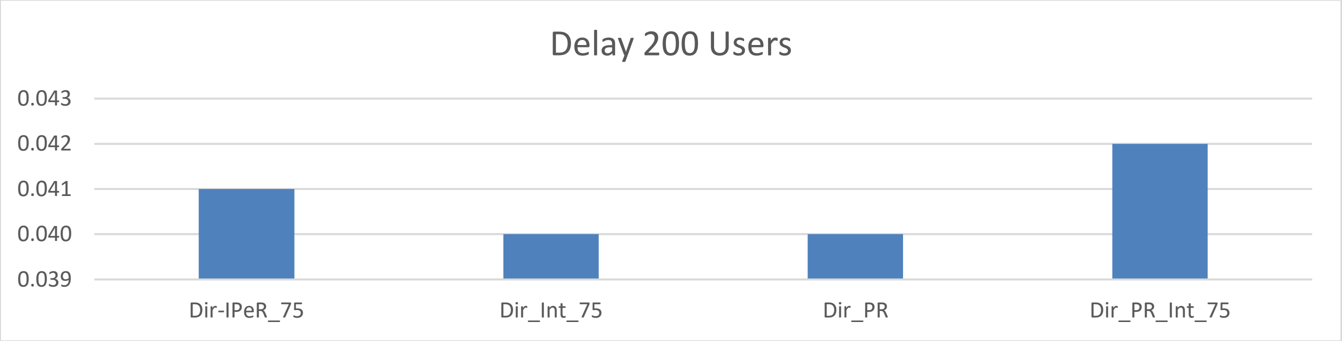


FIGURE 95: Delay of Setting 2, Directional-Interest, 200 Users

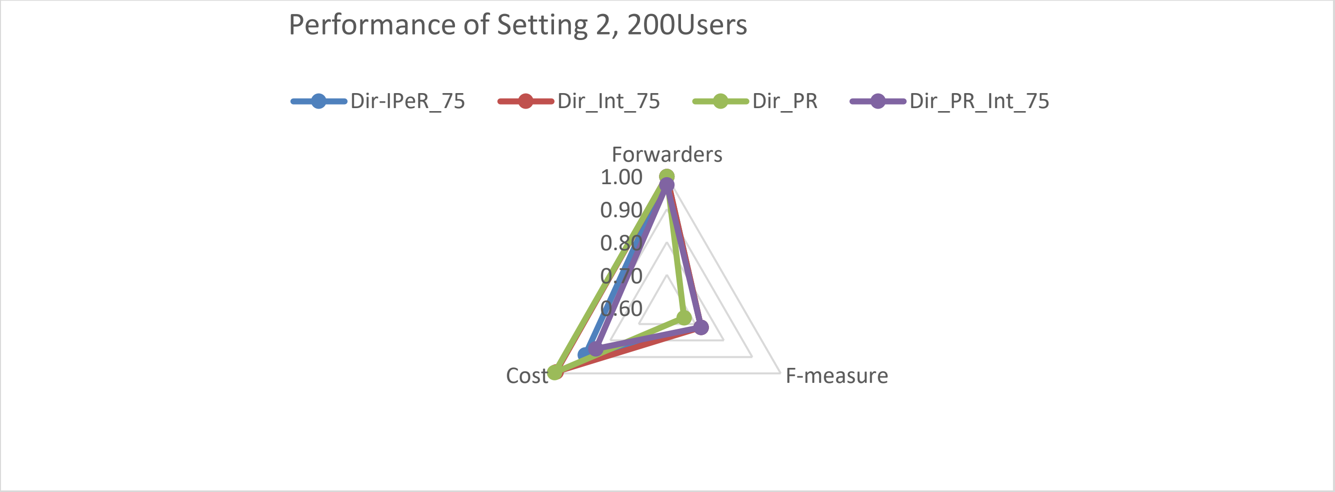


FIGURE 96: Delivery, Cost & F-measure of Setting 2, Directional-Interest, 200 Users

TABLE 61: Performance of Setting 2, Directional-Interest, 200Users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	47.36	0.042	0.72	168	0
Dir-IPeR_75	48.98	0.041	0.72	177	336
Dir_Int_75	48.98	0.040	0.72	198	177
Dir_PR	48.98	0.040	0.66	199	199
Dir_PR_Int_75	47.75	0.042	0.72	170	170

Discussion of Results

For 50 users, *Directional-PeopleRank* flooded the network by 98% data messages and 100% control messages. Consequently, it performed better compared to Directional-IPeR-75 in terms of the number of reached interested forwarders, and delay by 1% and 2%, respectively. However, it decreased F-measure by 7%.

Directional-Interest-75 performed equally to Directional-IPeR-75 in terms of the number of reached interested forwarders and F-measure. It enhanced the delay by 2%. However, it increased cost by 12%.

Directional-PeopleRank-Interest-75 performed equally in terms of F-measure. However, it performed worse in terms of the number of reached interested forwarders, delay, and cost by 8%, 11%, and 5%, respectively.

For 100 users, Directional-PeopleRank flooded the network by having a cost of 97% and control messages by 99%. Consequently, F-measure is decreased by 7%. The number of reached interested forwarders remains equal. Delay is enhanced by 2%.

Directional-Interest-75 performed almost exactly to Directional-IPeR-75 in all metrics, except it increased cost by 7%.

Directional-PeopleRank-Interest-75 performed equally in F-measure. However, it performed worse in terms of the number of reached interested forwarders, delay, and cost 4%, 5%, 14%, and 4%, respectively.

For 200 users, *Directional-PeopleRank* flooded the network by having the cost and control messages of 100%. Consequently, F-measure is decreased by 8%. It performed equally in terms of the number of reached interested forwarders. Delay is enhanced by 2%.

Directional-Interest-75 performed equally to Directional-IPeR-75 in all metrics except that it increased cost by 12% and enhanced delay by 2%.

Directional-PeopleRank-Interest-75 performed equally in F-measure. However, it performed worse in terms of the number of reached interested forwarder, and delay, by 3% and 2%. It enhanced cost by 4% compared to Directional-IPeR-75.

Conclusion on Adding Direction Guidance to Other Related Algorithms

Comparing Directional-IPeR-75 to Directional-Interest-75, they performed equally in all metrics, however, Directional-IPeR-75 performed better in terms of cost up to 17% in sparse density for uniform distribution. For normal distribution, the enhancement in cost is up to 11%. In high users' density, they performed equally for uniform distribution but Directional-IPeR-75 enhanced cost by 12%.

Comparing Directional-IPeR-75 to Directional-PeopleRank, Directional-IPeR-75 performed better in terms of F-measure and cost up to 19% in low users' density for uniform distribution. For normal distribution, the enhancement in F-measure is up to 7% and cost is up to 14%. In high users' density, Directional-IPeR-75 enhanced F-measure by 17% and cost is up to 39% for uniform distribution. For normal distribution, Directional-IPeR-75 enhanced F-measure up to 8% and cost up to 13%.

CHAPTER 7: Comparing Directional-IPeR to Social Cast

Social Cast is a publish-subscribe routing framework that utilizes metrics of social interaction such as, patterns of movements to predict the best nodes to forward a message. It used a mobility model based on a social network [112]. To benchmark our work, Social Cast algorithm [112] is compared to Directional-IPeR-75 in a uniform distribution setting.

Results

50 users

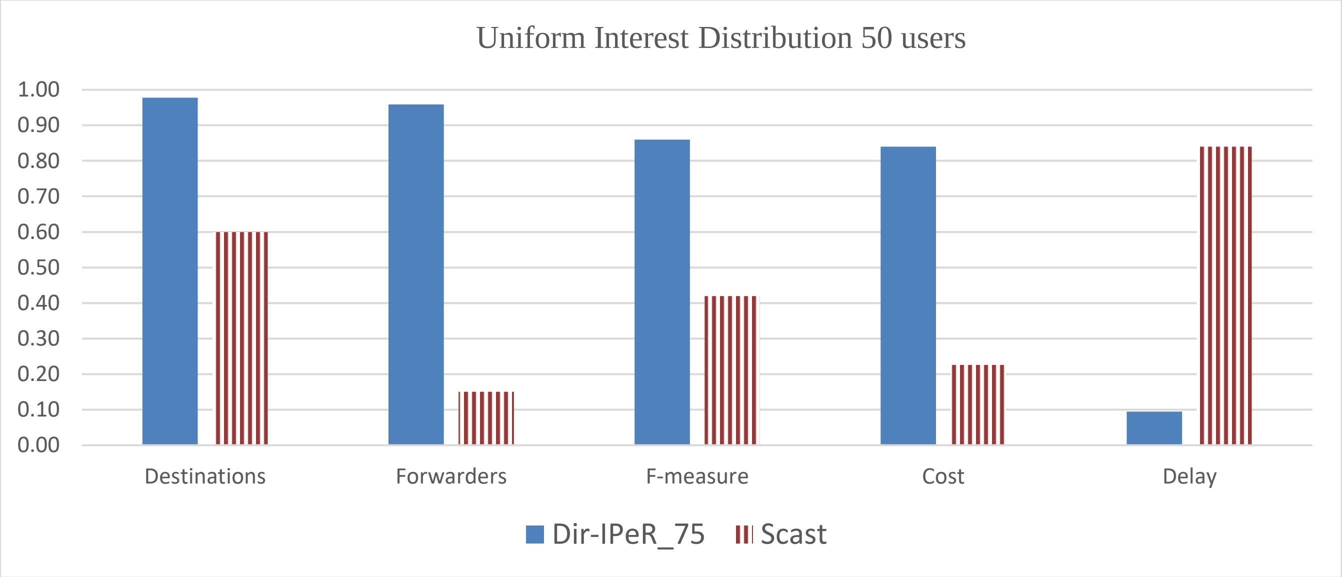


FIGURE 97: Comparing Directional-IPeR-75 to Social Cast, 50 Users

200 users

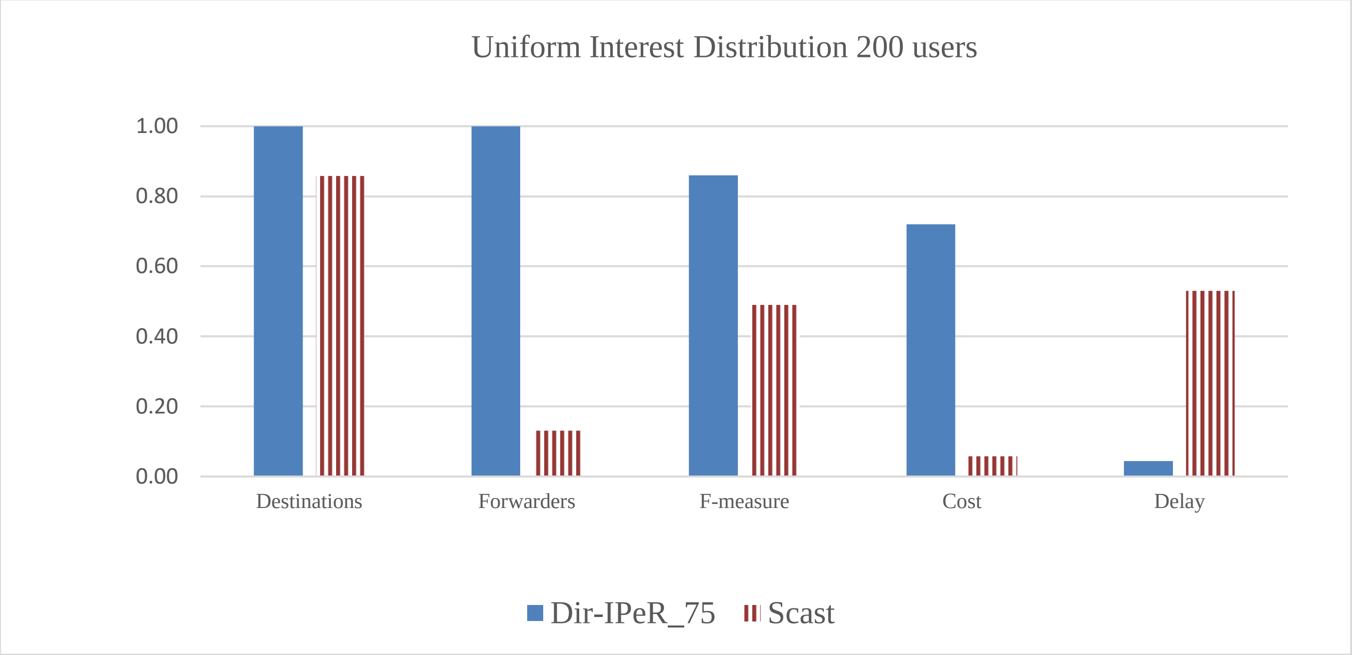


FIGURE 98: Comparing Directional-IPeR-75 to Social Cast, 200 Users

Conclusion on Comparing Directional-IPeR to Social Cast

Directional-IPeR-75 is compared to the Social Cast algorithm [112] in a uniform distribution setting. Directional-IPeR-75 performed better in all metrics except cost in high and low density of users. Directional-IPeR-75 performs better than Social Cast in terms of delivery ratio (up to 39% for 50-user experiments), reached interested forwarder nodes (up to 87% for 200-user experiments), and F-measure (up to 51% for 50 user experiments). Also, it reduces delays (up to 1200% for 200 users).

CHAPTER 8: Conclusion and Future Work

For MIPeR, 2Har-MIPeR performed the best in terms of cost and F-measure compared to all the proposed algorithms. In addition, it performed better in terms of the number of reached interested forwarders and delay compared to IPeR. The denser the environment is, the more delivery ratio, the more reached interested forwarders, the less cost and less delay exerted by the algorithm. It performed the best in discrete uniform distribution of interest in terms of F-measure, the best in the discrete normal distribution in terms of delay and performed equally in terms of cost for discrete uniform distribution and two disjoint subgraphs distribution of interest.

For discrete uniform distribution of interest, Directional-IPeR-75 is the best algorithm proposed for direction guidance in terms of F-measure and cost. At the same time, it performs better than IPeR in terms of delivery ratio, reached interested forwards and F-measure. In addition, it enhances delay in low users' density environment. For the different densities of users, 200 users' density has the best results in terms of delivery ratio, reached interested forwarders, and delay. This reveals that using Directional-IPeR-75 at discrete uniform distribution of interest, the denser the environment is, the more delivery ratio, the more reached interested forwarders, and the less delay exerted by the algorithm. Furthermore, Directional-IPeR algorithms produce control messages to calculate nodes directions. The least control messages are exerted by Directional-IPeR-75 for all user densities.

For discrete normal distribution of interest, Directional-IPeR-75 is the best algorithm for direction guidance in terms of F-measure and cost in comparison to the proposed algorithms. Added, it performs better than IPeR in terms of reached interested forwards, and delay. For F-measure, it performs equally or better compared to IPeR. For the different densities of users, 200 users' density has the best results in terms of reached interested forwarders and delay. This reveals that the denser the environment is, the more reached interested forwarders, and the less delay exerted by the algorithm.

For two disjoint subgraphs distribution of interest, Directional-IPeR-75 is the best algorithm proposed for direction guidance in terms of F-measure and cost. It performs better than IPeR in terms of delay in low users' density and delivery ratio in high users' density. For the different densities of users, 200 users' density has the best results in terms of delivery ratio, and delay. In this such challenging environment, where there are no interested forwarders, Directional-IPeR-75 performed better than IPeR as it can approach more delivery ratio with slight increase of delay in dense environments and perform equally in terms of delivery ratio with decreased delay in sparse environments.

For discrete uniform distribution of interest, Directional-IPeR-75-random performed better in all metrics compared to IPeR except in the delivery ratio. It performed equally in sparse environments. It enhanced F-measure and cost compared to Directional-IPeR-75. The denser the environment is, the more delivery ratio, the more number of reached interested forwarders, and the less delay. For discrete normal distribution of interest, Directional-IPeR-75-random performed better in all metrics compared to IPeR. It enhanced F-measure and cost compared to Directional-IPeR-75. It performs better in terms of the number of reached interested forwarders and delay in high density environments.

Directional-IPeR-75 performs equal to DMIPeR-75 in all metrics. However, increased cost. Adding 2 hops or three hops to it did not gain any enhancement in all metrics.

Comparing Directional-IPeR-75 to Directional-Interest-75, Directional-PeopleRank-Interest-75, it performed better in terms of cost in sparse density at discrete uniform distribution of interest. At discrete normal distribution, it performed better in terms of F-measure. At sparse density it performed better in terms of cost as well. The best F-measure is gained by Directional-PeopleRank-Interest-75. The other metrics are performed the best by Directional-PeopleRank. However, F-measure and cost are decreased drastically.

For the future work, other guiding criteria are to be tested and compared to Directional-IPeR 75. Other datasets other than SLAW are to be used to represent different mobility patterns. For MIPeR, weight can be added when evaluating the value of the node and its contacts.

References

[1] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical report CS-200006, Duke University, 2000.

[2] X. Hu, T. Chu, V. C. Leung, E. C. H. Ngai, P. Kruchten and H. C.Chan, "A survey on mobile social networks: Applications, platforms, system architectures, and future research directions," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 3, pp. 1557-1558, 2015.

[3] S. Pathak, N. Gondaliya and N. Raja, "A survey on PRoPHET based routing protocol in delay tolerant network," in *Emerging Trends & Innovation in ICT (ICEI), 2017 International Conference*, 2017.

[4] S. R. Bharamagoudar and S. V. Saboji, "Routing in Opportunistic Networks: Taxonomy, Survey.," in *International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEEECCOT)* , 2017.

[5] E. Chung, J. Joy and M. Gerla, "DiscoverFriends: Secure social network communication in mobile ad hoc networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International 2015 Aug 24 (pp. 7-12). IEEE.*, 2015.

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

[7] J. Wu and Y. Wang, Eds., Opportunistic Mobile Social Networks, CRC Press, 2014.

[8] M. K. Denko, Ed., Mobile Opportunistic Networks: Architectures, Protocols and Applications, CRC Press, 2016.

[9] Z. Sui, "The Research of the Route Protocols in Opportunistic Network," *Computational Intelligence and Communication Networks (CICN),* pp. 192-196, 2015.

[10] S. Pal, B. K. Saha and S.Misra, "Game Theoretic Analysis of Cooperative Message Forwarding in Opportunistic Mobile Networks.," *IEEE transactions on cybernetics,* vol. 47, no. 12, pp. 4463-4474, 2017.

[11] N. Rajpoot and S. K. Rajendra, "An improved PRoPHET routing protocol for underwater communication," *In Communication Networks (ICCN),* pp. 27-32, 2015.

[12] C. Sobin, V. Raychoudhury, G. Marfia and A. Singla, "A survey of routing and data dissemination in delay tolerant networks," *Journal of Network and Computer Applications,* no. 67, pp. 128-146, 2016.

[13] J. Hom, L. Good and S. Yang, "A survey of social-based routing protocols in delay tolerant networks," in *Computing, Networking and Communications (ICNC)*, 2017.

[14] L. Liu and Y. Jing, "A Survey on Social-Based Routing and Forwarding Protocol in Opportunistic Networks," *Computer and Information Technology (CIT), IEEE 12th International Conference*, 2012.

[15] 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 & Tutorials,* vol. 15, no. 1, pp. 387-401, 2013.

[16] M. Kawecki and R. O. Schoeneich, "Mobility-based routing algorithm in delay tolerant networks.," *EURASIP Journal on Wireless Communications and Networking,* pp. 81-90, 2016.

[17] J. Burgess, B. Gallagher, D. Jensen and N. Levine, "Max Prop: Routing for Vehicle-Based Disruption-tolerant Networks," in *Proceedings IEEE INFOCOM, 25TH IEEE International Conference on Computer Communications*, 2006.

[18] N. Gondaliya, D. Kathiriya and M. Shah, "Contact frequency and contact duration based relay selection approach inside the local community in social delay tolerant network," *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics (pp.). Springer, .,* pp. 609-617, 2016.

[19] K. Bylykbashi, E. Spaho, L. Barolli and M. Takizawa, "Comparison of Spray and Wait and Epidemic Protocols in Different DTN Scenarios," in *International Conference on Broadband and Wireless Computing, Communication and Applications*, 2017.

[20] T. -K. Huang, C. -K. Lee and L. -J. Chen, "PRoPHET+: An adaptive PRoPHET-based routing protocol for opportunistic network," *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 112-119, 2010.

[21] N. Vastardis and K. Yang, "Mobile social networks: Architectures, social properties, and key research challenges," *IEEE Communications Surveys & Tutorials 15,* no. 3, pp. 1355-1371, 2013.

[22] X. Deng, L. Chang, J. Tao, J. Pan and J. Wang, "Social profile-based multicast routing scheme for delay-tolerant networks," *IEEE International Conference on Communications (ICC),* pp. 1857-1861, 2013.

[23] H. Takasuka, K. Hirai and K. Takami, "Development of a Social DTN for Message Communication between SNS Group Members," *Future Internet,* vol. 10, no. 4, p. 32, 2018.

[24] B. Guo, D. Zhang, Z. Yu, X. Zhou and Z. Zhou, "Enhancing spontaneous interaction in opportunistic mobile social networks," *Communications in Mobile Computing,* vol. 1, no. 1, pp. 1-6, 2012.

[25] S. Bhattacharjee, S. Basu, S. Roy and S. Bit, "Best-effort delivery of emergency messages in post-disaster scenario with content-based filtering and Priority-enhanced PRoPHET over DTN," *8th International Conference on Communication Systems and Networks (COMSNETS). IEEE.,* pp. 1-7, 2016.

[26] Z. Yu, Y. Liang, B. Xu, Y. Yang and B. Guo, "Towards a smart campus with mobile social networking," in *International Conferenceon Cyber, Physical and Social Computing*, 2011.

[27] P. Tarasewich, R. C. Nickerson and M. Warkenti, "An introduction to wireless mobile social networking in opportunistic communication," *Information Systems,* vol. 4, no. 3, pp. 435-438, 2013.

[28] C. H. Lee, "Exploiting heterogeneity in mobile opportunistic networks: An analytic approach," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference*, 2010.

[29] C. Boldrini and A. Passarella, "Hcmm: Modelling spatial and temporal properties of human mobility driven by users social relationships," *Computer Communications,* vol. 33, no. 9, p. 1056 – 1074, 2010.

[30] S. Agoujil, M. Hajar and Y. Qaraai, "Improving the data delivery using DTN Routing Hierarchical Topology (DRHT)," *Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference. IEEE.,* pp. 1-5.

[31] S. Jain, M. Chawla, V. N. Soares and J. J. Rodrigues, "Enhanced fuzzy logic-based spray and wait routing protocol for delay tolerant networks.," *International Journal of Communication Systems,* vol. 29, no. 12, pp. 1820-1843, 2016.

[32] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and Wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of ACM SIGCOMM 2005 Workshop on Delay Tolerant Networking and Related Networks (WDTN-2005)*, Philadelphia, 2005.

[33] N. Derakhshanfard, M. Sabaei and A. M. Rahmani, "Sharing spray and wait routing algorithm in opportunistic networks," *Wireless Networks,* vol. 22, no. 7, pp. 2403-2414, 2016.

[34] K. Chaubey and P. Mistri, "An Encounter Based Routing in Delay Tolerant Network (DTN): A Hybrid Approach," *International Journal Of Innovative Research In Computer And Communication Engineering,* vol. 4, no. 5, pp. 8657-8662, 2016.

[35] P. Maitreyi and M. S. Rao, "Design of binary spray and wait protocol for intermittently connected mobile networks," in *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*, 2017.

[36] D. Maywad, C. Agrawal and K. Sanjay, "Improved Spray and Wait Protocol for DTN Networks. context," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 7, no. 16, pp. 441-444, 2017.

[37] H. Chen and W. Lou, "Contact expectation based routing for delay tolerant networks.," *Ad Hoc Networks,* vol. 36, no. 1, pp. 244-257, 2015.

[38] K. Wang and H. Guo, "An improved routing algorithm based on social link awareness in delay tolerant networks," *Wireless Personal Communications,* vol. 75, no. 1, p. 397–414, 2013.

[39] S. Batyabal and P. Bhaumik, "Analysing social behaviour and message dissemination in human based delay tolerant network.," *Wireless Networks,* vol. 21, no. 3, pp. 513-529, 2015.

[40] Y. Xu and X. Chen, "Social-similarity-based multicast algorithm in impromptu mobile social networks," in *Global Communications Conference (GLOBECOM)*, 2014.

[41] M. Kadadha, H. Al-Ali, M. A. Mufti, A. Al-Aamri and R. Mizouni, "Opportunistic mobile social networks: Challenges survey and application in smart campus," in *Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016.

[42] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *INFOCOM*, 2012.

[43] A. Mtibaa, M. May and M. Ammar, "Social forwarding in mobile opportunistic networks: A case of peoplerank," in *Handbook of Optimization in Complex Networks*, New York, NY, Springer, 2012, pp. 387-425.

[44] S. A. Al Ayyat, S. A. Gamal and K. A. Harras, "Interest aware peoplerank: Towards effective social-based opportunistic advertising," pp. 4428-4433, 2013.

[45] W. Li, A. Joshi and T. Finin, "Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach," *in 2010 Eleventh International Conference on Mobile Data Management,* p. 85–94, 2010.

[46] N. Moati, H. Otrok, A. Mourad and J.-M. Robert, "Reputation-based cooperative detection model of selfish nodes in cluster-based qos-olsr protocol," *Wireless Personal Communications,* vol. 75, no. 3, p. 1747– 1768, 2013.

[47] T. Abdelkader, K. Naik, A. Nayak, N. Goel and V. Srivastava, "A performance comparison of delay-tolerant network routing protocols," *IEEE Network,* vol. 30, no. 2, pp. 46-53, 2016.

[48] E. Spaho, K. Bylykbashi, L. Barolli, V. Kolici and A. Lala, "Evaluation of different DTN routing protocols in an opportunistic network considering many-to-one communication scenario," in *19th International Conference on Network-Based Information Systems (NBiS)*, 2016.

[49] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 32-40, 2007.

[50] W. Cherif, M. Khan, F. Filali, S. Sharafeddine and Z. Dawy, "P2p group formation enhancement for opportunistic networks with wi-fi direct.," *IEEE Wireless Communications and Networking Conference (WCNC),* pp. 1-6, 2017.

[51] G. Palla, I. Derényi, I. Farkas and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature,* vol. 435, no. 7043, pp. 814-824.

[52] P. Hui, J. Crowcroft and E. Yoneki, "BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks," *IEEE Transactions on Mobile Computing,* vol. 10, no. 11, pp. 1576-1589., 2011.

[53] X. Meng, G. Xu, T. Guo, Y. Yang, W. Shen and K. Zhao, "A novel routing method for social delay-tolerant networks," *Tsinghua Science and Technology,* vol. 24, no. 1, pp. 44-51, 2019.

[54] J. Chang and C. Chen, "CROP: Community relevance-based opportunistic routing in delay tolerant networks," *EICE TRANSACTIONS on Communications,* Vols. E97-B, no. 9, pp. 1875-1888, 2014.

[55] J. Wu and Z. Chen, "Data decision and transmission based on mobile data health records on sensor devices in wireless networks," *Wireless Personal Communications,* vol. 90, no. 4, pp. 2073-2087, 2016.

[56] Z. Chen and J. Wu, "Applying a sensor energy supply communication scheme to big data opportunistic networks," *TIIS,* vol. 10, no. 5, pp. 2029-2046, 2016.

[57] A. Perrig, J. Stankovic and D. Wagner, "Security in wireless sensor networks.," pp. 53-57, 2004.

[58] H. Dubois-Ferriere, M. Grossglauser and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing* , pp. 257-266, 2003.

[59] E. Bulut and B. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 23, no. 12, pp. 2254-2265, 2012.

[60] S. Hossen and M. Rahim, "Impact of mobile nodes for few mobility models on delay-tolerant network routing protocols," *International Conference on Networking Systems and Security (NSysS),* pp. 1-6, 2016.

[61] D. Hrabcak, M. Matis, L. Dobos and J. Papaj, "Social Based Mobility Model with Metrics for Evaluation of Social Behaviour in Mobility Models for MANET-DTN Networks," *Advances in Electrical and Electronic Engineering,* vol. 15, no. 4, pp. 606-612, 2017.

[62] L. Barkhuus and A. Dey, "Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns," *Interact,* pp. 702-712, 2003.

[63] J. LeBrun, C. Chuah, l. D. Ghosa and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," *IEEE 61st Vehicular Technology Conference,* pp. 2289-2293, 2005.

[64] S. -K. Kim, J. -H. Choi and S. B. Yang, "Hotspot: Location-based Forwarding Scheme in an Opportunistic Network," *Adhoc & Sensor Wireless Networks,* no. 26, 2015.

[65] S. Dhurandher, S. Borah, I. Woungang, A. Bansal and A. Gupta, "A location Prediction-based routing scheme for opportunistic networks in an IoT scenario," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 369-378, 2018.

[66] M. Jeon, S. K. Kim, J. H. Yoon, J. Lee and S. B. Yang, "A direction entropy-based forwarding scheme in an opportunistic network," *Journal of Computing Science and Engineering*, vol. 8, no. 9, pp. 173-179, 2014.

[67] Y. Wang and J. Wu, "A dynamic multicast tree based routing scheme without replication in delay tolerant networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 3, p. 424–436, 2012.

[68] S. Kim and S. Han, "Distinguishing social contacts and pass-by contacts in DTN routing," *Telecommunication Systems*, vol. 68, no. 4, pp. 669-685, 2018.

[69] H. Haddadi, P. Hui, T. Henderson and I. Brown, "Targeted Advertising on the Handset : Privacy and Security Challenges," *Pervasive Advertising*, p. 1–19, 2011.

[70] L. Song and D. F. Kotz, " Routing in Mobile Opportunistic Networks.," *Mobile Opportunistic Networks: Architectures, Protocols and Applications*, no. 1, 2016.

[71] T. Choksatid, W. Narongkhachavana and S. Prabhavat, "An efficient spreading epidemic routing for delay-tolerant network," *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*, pp. 473-476, 2016.

[72] D. Sharma, S. Dhurandher, M. Obaidat, S. Pruthi and B. Sadoun, "A priority based message forwarding scheme for Opportunistic Networks," *Computer, Information and Telecommunication Systems (CITS), 2016 International Conference. IEEE.*, 2016.

[73] A. Gupta, I. Bhattacharya, P. Banerjee, J. Mandal and A. Mukherjee, "DirMove: direction of movement based routing in DTN architecture for post-disaster scenario," *Wireless Networks*, vol. 22, no. 3, pp. 723-740, 2016.

[74] H. Al-Ghanimi, M. Radenkovic and A. Hassan, "Opportunistic Communication in Highly Congested and Dynamic Urban Areas.," *IOSR Journal of Engineering (IOSRJEN)*, vol. 3, no. 10, pp. 46-52, 2013.

[75] V. Samyal and Y. Sharma, "Performance Evaluation of Delay Tolerant Networks Routing Protocols under varying Time to Live," vol. 8, no. 1.

[76] H. Kaur and H. Kaur, "An enhanced spray-copy-wait DTN routing using optimized delivery predictability," *Computer Communication, Networking and Internet Security*, pp. 603-610, 2017.

[77] T. Spyropoulos, K. Psounis and C. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pp. 79-85, 2007.

[78] S. Azzuhri, H. Ahmad, M. Portmann, I. Ahmedy and R. Pathak, "An efficient hybrid MANET-DTN routing scheme for OLSR," *Wireless Personal Communications,* vol. 89, no. 4, pp. 1335-1354, 2016.

[79] Y. Cai, Y. Fan and D. Wen, "An incentive-compatible routing protocol for two-hop delay-tolerant networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 1, pp. 266-277, 2016.

[80] P. Yuan and C. Wang, " OPPO: An optimal copy allocation scheme in mobile opportunistic networks," *Peer-to-Peer Networking and Applications,* vol. 11, no. 1, pp. 102-109, 2018.

[81] K. Avhad, S. Limkar and A. Kulkarni, "Probability and Priority Based Routing Approach for Opportunistic Networks," in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, 2014.

[82] J. Lee, S. K. Kim, J. H. Yoon and S. B. Yang, "Snapshot: a forwarding strategy based on analyzing network topology," *opportunistic networks. Wireless networks,* vol. 21, no. 6, pp. 2055-2068, 2015.

[83] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan and Z. Cai, "A novel contact prediction-based routing scheme for DTNs," *Transactions on Emerging Telecommunications Technologies,* vol. 28, no. 1, p. 2889, 2017.

[84] J. Fraire and J. Finochietto, "Routing-aware fair contact plan design for predictable delay tolerant networks," *Ad Hoc Networks,* vol. 1, pp. 303-313, 2015.

[85] Z. Huijuan and L. Kai, "A Routing Mechanism Based on Social Networks and Betweenness Centrality in Delay Tolerant Networks," *International Journal of Computer Science & Information Technology,* vol. 7, no. 6, pp. 107-116.

[86] Z. Huijuan and L. Kai, "A Routing Mechanism Based on Social Networks and Betweenness Centrality in Delay Tolerant Networks," *International Journal of Computer Science & Information Technology,* vol. 7, no. 6, pp. 107-116, 2015.

[87] Y. Kim, C. Kim, Y. Han, Y. Jeong and D. Park, "Betweenness of expanded Ego networks in sociality-aware delay tolerant networks.," *InUbiquitous Information Technologies and Applications,* pp. 499-505, 2013.

[88] E. Bulut and B. Szymanski, "Friendship based routing in delay tolerant mobile social networks," *IEEE Global Telecommunications Conference GLOBECOM,* pp. 1-5, 2010.

[89] C. -M. Kim, Y. -H. Han, J. -S. Youn and Y. -S. Jeong, "A Socially Aware Routing Based on Local Contact Information in Delay-Tolerant Networks," *The Scientific World Journal,* vol. 7, 2014.

[90] M. Newman, "Analysis of Weighted Networks," *Physical Review E,* vol. 70, no. 15, p. 056131, 2004.

[91] S. Nelson, M. Bakht and R. Kravets, "Encounter-based routing in DTNs," *IEEE INFOCOM* , pp. 846-854, 2009.

[92] C. Casetti, C. Chiasserini, Y. Duan, P. Giaccone and A. Manriquez, " Data connectivity and smart group formation in Wi-Fi Direct multi-group networks," *IEEE Transactions on Network and Service Management,* vol. 15, no. 1, pp. 245-259, 2018.

[93] J. Guan, Q. Chu and I. You, "The social relationship based adaptive multi-spray-and-wait routing algorithm for disruption tolerant network," *Mobile Information Systems,* 2017.

[94] J. WU, Z. Chen and M. Zhao, "Effective information transmission based on socialization nodes in opportunistic networks," *Computer Networks,* vol. 129, pp. 297-305, 2017.

[95] Y. Li, Y. Cao, S. Li, D. Jin and L. Zeng, "Integrating forwarding and replication in dtn routing: A social network perspective," *IEEE 73rd Vehicular Technology Conference (VTC Spring),* pp. 1-5, 2011.

[96] N. Gondaliya and D. Kathiriya, "Community detection using inter contact time and social characteristics based single copy routing in delay tolerant networks," *Int. J Ad hoc Sens Ubiquitous Comput,* vol. 7, no. 1, pp. 21-35, 2016.

[97] T. Wang, Y. Zhou, Y. Wang and M. Tang, "Novel Opportunistic Network Routing Based on Social Rank for Device-to-Device Communication," *Journal of Computer Networks and Communications,* 2017.

[98] Y. Cao, N. Wang, Z. Sun and H. Cruickshank, "A reliable and efficient encounter-based routing framework for delay/disruption tolerant networks," *IEEE Sensors Journal,* vol. 15, no. 7, pp. 4004-4018, 2015.

[99] M. Orlinski and N. Filer, "The rise and fall of spatio-temporal clusters in mobile ad hoc networks," *Ad Hoc Networks.,* vol. 11, no. 5, pp. 1641-54, 2013.

[100] F. Zeng, N.Zhao and W. Li, "Effective Social Relationship Measurement and Cluster Based Routing in Mobile Opportunistic Networks," *Sensors,* vol. 17, no. 5, pp. 1109-1117, 2017.

[101] S. K. Dhurandher, D. K. Sharma, I. Woungang and A. Saini, "Efficient routing based on past information to predict the future location for message passing in infrastructure-less opportunistic networks," *The Journal of Supercomputing,* vol. 71, no. 5, pp. 1695-1711, 2015.

[102] M. Xia, Q. Wang, Q. Wang, C. Cao, L. Wang and H. Guo, "Two-level community-based routing in delay tolerant networks," *Wireless Personal Communications,* vol. 96, no. 4, pp. 5687-5704, 2017.

[103] T. Abdelkader, K. Naik, A. Nayak, N. Goel and V. Srivastava, "SGBR: A routing protocol for delay tolerant networks using social grouping.," *IEEE Transactions on Parallel and Distributed Systems,* vol. 24, no. 12, pp. 2472-2481, 2013.

[104] V. Erramilli, A. Chaintreau, M. Crovella and C. Diot, "Diversity of forwarding paths in pocket switched networks," *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement,* pp. 161-174, 2017.

[105] S. Dhurandher, D. Sharma, I. Woungang and S. Bhati, "HBPR: history based prediction for routing in infrastructure-less opportunistic networks," *IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)* , pp. 931-936, 2013.

[106] S. Dhurandher, S. Borah, I. Woungang, D. Sharma, K. Arora and D. Agarwal, "An encounter and distance based routing protocol for opportunistic networks," *The 30th IEEE International Conference on Advanced Information Networking and Applica,* 2016.

[107] S. Dhurandher, S. Borah, M. Obaidat, D. Sharma, S. Gupta and B. Baruah, "Probability-based controlled flooding in opportunistic networks," *12th International Joint Conference on e-Business and Telecommunications (ICETE),* vol. 6, pp. 3-8, 2015.

[108] A. Balasubramanian, B. Levine and A. Venkataramani, " DTN routing as a resource allocation problem," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 373-384, 2007.

[109] S. A. Al Ayyat, S. A. Gamal and K. A. Harras, "SAROS: A social-aware opportunistic forwarding simulator," *IEEE Wireless Communications and Networking Conference,* pp. 1-7, 2016.

[110] K. Lee, S. Hong, I. Rhee and S. Chong, "SLAW: SelfSimilar Least-Action Human Walk," *IEEE / ACM Transactions on Networking*, vol. 20, pp. 515-529, 2012.

[111] Y. S. Shahin, S. A. Al Ayyat and S. G. Aly, "MIPeR: Enhanced Multiple Hops Interest Aware PeopleRank for Opportunistic Mobile Social Networks," Cairo, 2020.

[112] P. Costa, C. Mascolo, M. Musolesi and G. (. Picco, " Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on selected areas in communications*, vol. 26, no. 5, pp. 748-760, 2008.

[113] C. M. Huang, K. C. Lan and C. Z. Tsai, "A survey of opportunistic networks," *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference,* pp. 1672-1677, 2008.

[114] V. Arnaboldi, M. Conti and F. Delmastro, "Cameo: A novel contextaware middleware for opportunistic mobile social networks," *Pervasive and Mobile Computing*, vol. 11, pp. 148-167, 2014.

[115] A. C. Champion, Z. Yang, B. Zhang, J. Dai, D. Xuan and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," *IEEE Transactions on Parallel and Distributed Systems,* vol. 4, no. 8, pp. 1535-1545, 2013.

[116] N. Aion, L. Helmandollar, M. Wang and J. W. P. Ng, "Intelligent campus (icampus) impact study," *Web Intelligence and Intelligent Agent Technology (WI-IAT). IEEE/WIC/ACM International Conferences,* vol. 3, p. 291–295, 2012.

[117] V. Arnaboldi, M. Conti, F. Delmastro, G. Minutiello and a. L. Ricci, "Droidopppathfinder: A context and social-aware path recommender system based on opportunistic sensing," *World of Wireless, Mobileand Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops*, pp. 1-3, 2013.

[118] E. Bulut and B. K. E. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2254-2265, 2012.

[119] T. C. Tsai and H. H. Chan, "Novel Scheme for the Distribution of Flyers Using a Real Movement Model for DTNs," *Frontier Computing*, pp. 937-947, 2016.

[120] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan and Z. Cai, "A novel contact prediction-based routing scheme for DTNs," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, p. 2889, 2017.

[121] W. Jia, Z. Chen and M. Zhao, "Effective information transmission based on socialization nodes in opportunistic networks," *Computer Networks*, vol. 129, pp. 297-305, 2017.

[122] V. Kostakos, "Temporal graphs," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 6, pp. 1007-1023, 2009.

[123] S. El Alaoui and B. Ramamurthy, "EAODR: A novel routing algorithm based on the Modified Temporal Graph network model for DTN-based Interplanetary Networks," *Computer Networks*, vol. 129, pp. 129-141, 2017.

[124] A. Mtibaa, M. May and C. Diot, "Method to manage an opportunistic communication network," *inventors; Thomson Licensing SA, assignee. United States patent application*, vol. 13, no. 517, p. 459, 2012.

[125] M. Matis and J. Papaj, "An enhanced hybrid social based routing algorithm for MANET-DTN.," *Mobile Information Systems. .*

[126] A. Mei, N. Piroso and J. Stefa, "Count on me: Reliable broadcast and efficient routing in DTNs through social skeletons," *Journal of Parallel and Distributed Computing.*, vol. 96, pp. 95-105, 2016.

[127] A. Dziekonski and R. Schoeneich, "DTN Routing Algorithm for Networks with Nodes Social Behavior," *International Journal of Computers, Communications & Control*, vol. 11, no. 4, pp. 457-471, 2016.

[128] A. K. Pietiläinen, E. Oliver, J. LeBrun, G.Varghese and C. Diot, "MobiClique: middleware for mobile social networking," pp. 49-54.

[129] A.-K. Pietiläinen, "Opportunistic Mobile Social Networks at Work," Th`ese de Doctorat de l'Universit´e Paris VI, Paris, 2010.

[130] S. A. Al Ayyat, K. A. Harras and S. G. Aly, "Interest aware peoplerank: Towards effective social-based opportunistic advertising," in *Wireless Communications and Networking Conference (WCNC)*, 2013.

Appendix A: 2-MIPeR results

1. Setting 1

50 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.60	29.90	0.107	0.85	29
2Har_MIPeR	17.60	34.00	0.104	0.76	42
2Avg_MIPeR	17.65	35.45	0.085	0.70	49
2Max_MIPeR	17.65	35.50	0.084	0.70	49
25%_2Max_MIPeR	17.60	35.10	0.086	0.75	44
50%_2Max_MIPeR	17.60	35.10	0.087	0.75	44
75%_2Max_MIPeR	17.60	35.10	0.087	0.75	44
IPeR/Avg	0.00	0.19	0.206	0.18	1
IPeR/Max	0.00	0.19	0.215	0.18	1
IPeR/X%Max	0.00	0.17	0.187	0.12	1
IPeR/Har	0.00	0.14	0.028	0.11	0

100 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.88	32.68	0.060	0.85	65
2Har_MIPeR	17.88	35.13	0.055	0.75	88
2Avg_MIPeR	18.00	35.53	0.059	0.70	99
2Max_MIPeR	18.00	35.83	0.057	0.70	100
25%_2Max_MIPeR	17.88	35.43	0.049	0.75	89
50%_2Max_MIPeR	17.88	35.43	0.049	0.75	89
75%_2Max_MIPeR	17.88	35.43	0.049	0.74	89
IPeR/Avg	0.01	0.09	0.017	0.18	1
IPeR/Max	0.01	0.10	0.050	0.18	1
IPeR/X%Max	0.00	0.08	0.183	0.12	0
IPeR/Har	0.00	0.07	0.083	0.12	0

200 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.80	34.74	0.039	0.86	137
2Har_MIPeR	17.90	36.08	0.046	0.75	162
2Avg_MIPeR	17.99	36.14	0.043	0.71	198
2Max_MIPeR	18.00	36.49	0.036	0.71	200
25%_2Max_MIPeR	18.00	36.49	0.043	0.75	181
50%_2Max_MIPeR	18.00	36.49	0.043	0.75	181
75%_2Max_MIPeR	18.00	36.49	0.043	0.75	181
IPeR/Avg	0.01	0.04	0.103	0.17	0
IPeR/Max	0.01	0.05	0.077	0.17	0
IPeR/X%Max	0.01	0.05	0.103	0.13	0
IPeR/Har	0.01	0.04	0.179	0.13	0

2. Setting 2

50 users

Algorithm	Forwarders	Delay	F-measure	Cost
IPeR	43.20	0.060	0.73	38
2Har_MIPeR	48.20	0.047	0.68	42
2Avg_MIPeR	48.20	0.045	0.67	49
2Max_MIPeR	48.25	0.047	0.67	49
25%_2Max_MIPeR	48.25	0.047	0.67	43
50%_2Max_MIPeR	48.25	0.047	0.67	43
75%_2Max_MIPeR	48.25	0.047	0.67	44
IPeR/Har	0.12	0.217	0.07	0
IPeR/Avg	0.12	0.250	0.08	0
IPeR/Max	0.12	0.217	0.08	0
IPeR/X%Max	0.12	0.217	0.08	0

100 users

Algorithm	Forwarders	Delay	F-measure	Cost
IPeR	45.58	0.059	0.71	82
2Har_MIPeR	48.33	0.050	0.67	86
2Avg_MIPeR	48.38	0.049	0.67	99
2Max_MIPeR	48.48	0.048	0.66	100
25%_2Max_MIPeR	48.48	0.048	0.67	98
50%_2Max_MIPeR	48.48	0.048	0.67	98
75%_2Max_MIPeR	48.48	0.048	0.67	98
IPeR/Har	0.06	0.153	0.06	0
IPeR/Avg	0.06	0.169	0.06	0
IPeR/Max	0.06	0.186	0.07	0
IPeR/X%Max	0.06	0.186	0.06	0

200 users

Algorithm	Forwarders	Delay	F-measure	Cost
IPeR	47.36	0.042	0.72	168
2Har_MIPeR	48.44	0.041	0.67	167
2Avg_MIPeR	48.00	0.041	0.66	198
2Max_MIPeR	48.95	0.040	0.66	200
25%_2Max_MIPeR	48.95	0.040	0.67	197
50%_2Max_MIPeR	48.95	0.040	0.67	197
75%_2Max_MIPeR	48.95	0.041	0.67	197
IPeR/Har	0.02	0.024	0.07	0
IPeR/Avg	0.01	0.024	0.08	0
IPeR/Max	0.03	0.048	0.08	0
IPeR/X%Max	0.03	0.048	0.07	0

3. Setting 3

50 users

Algorithm	Destinations	Delay	F-measure	Cost
IPeR	19.40	0.191	0.76	16
2Har_MIPeR	19.50	0.171	0.41	39
2Avg_MIPeR	19.50	0.159	0.33	49
2Max_MIPeR	19.50	0.144	0.33	49
25%_2Max_MIPeR	19.50	0.153	0.38	41
50%_2Max_MIPeR	19.50	0.153	0.38	41
75%_2Max_MIPeR	19.50	0.153	0.38	41
IPeR/Har	0.01	0.105	0.46	1
IPeR/Avg	0.01	0.168	0.57	2
IPeR/Max	0.01	0.246	0.57	2
IPeR/X%Max	0.01	0.199	0.50	2

100 users

Algorithm	Destinations	Delay	F-measure	Cost
IPeR	19.55	0.075	0.74	33
2Har_MIPeR	19.55	0.062	0.39	82
2Avg_MIPeR	19.88	0.075	0.33	99
2Max_MIPeR	19.98	0.079	0.33	100
25%_2Max_MIPeR	19.98	0.082	0.39	84
50%_2Max_MIPeR	19.98	0.082	0.39	84
75%_2Max_MIPeR	19.98	0.082	0.39	84
IPeR/Har	0.00	0.173	0.47	1
IPeR/Avg	0.02	0.000	0.55	2
IPeR/Max	0.02	0.053	0.55	2
IPeR/X%Max	0.02	0.093	0.47	2

200 users

Algorithm	Destinations	Delay	F-measure	Cost
IPeR	19.80	0.046	0.69	77
2Har_MIPeR	19.80	0.046	0.42	161
2Avg_MIPeR	19.99	0.045	0.36	196
2Max_MIPeR	19.99	0.044	0.33	200
25%_2Max_MIPeR	19.99	0.045	0.39	168
50%_2Max_MIPeR	19.99	0.045	0.39	168
75%_2Max_MIPeR	19.99	0.045	0.39	168
IPeR/Har	0.00	0.000	0.39	1
IPeR/Avg	0.01	0.022	0.48	2
IPeR/Max	0.01	0.043	0.52	2
IPeR/X%Max	0.01	0.022	0.43	1

Appendix B: 3-MIPeR results

1. Setting 1

50 users

App	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	17.60	29.90	0.107	0.85	29
3Avg_MIPeR	17.65	35.45	0.084	0.70	49
3Max_MIPeR	17.65	35.50	0.080	0.70	49
3Har_MIPeR	17.60	34.85	0.104	0.75	41
2Har_MIPeR	17.60	34.00	0.104	0.76	42
IPeR/Avg	**0.00**	**0.19**	**0.219**	**0.18**	**1**
IPeR/Max	**0.00**	**0.19**	**0.253**	**0.18**	**1**
IPeR/3Har	**0.00**	**0.17**	**0.035**	**0.12**	**0**
IPeR/2Har	**0.00**	**0.14**	**0.032**	**0.11**	**0**

100 users

App	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	**17.88**	**32.68**	**0.060**	**0.85**	**65**
3Avg_MIPeR	**18.00**	**35.58**	**0.060**	**0.70**	**99**
3Max_MIPeR	**18.00**	**35.83**	**0.054**	**0.70**	**100**
3Har_MIPeR	**17.88**	**35.10**	**0.057**	**0.75**	**88**
2Har_MIPeR	**17.88**	**35.13**	**0.055**	**0.75**	**88**
IPeR/Avg	**0.01**	**0.09**	**0.008**	**0.18**	**1**
IPeR/Max	**0.01**	**0.10**	**0.086**	**0.18**	**1**
IPeR/3Har	**0.00**	**0.07**	**0.035**	**0.13**	**0**
IPeR/2Har	**0.00**	**0.07**	**0.069**	**0.13**	**0**

200 users

App	Destinations	Forwarders	Delay	F-measure	Cost
IPeR	**17.80**	**34.74**	**0.039**	**0.86**	**137**
3Avg_MIPeR	**17.99**	**36.20**	**0.037**	**0.71**	**198**
3Max_MIPeR	**18.00**	**36.49**	**0.036**	**0.71**	**200**
3Har_MIPeR	**17.99**	**36.14**	**0.046**	**0.75**	**163**
2Har_MIPeR	**17.90**	**36.08**	**0.046**	**0.75**	**162**
IPeR/Avg	**0.01**	**0.04**	**0.054**	**0.18**	**0**
IPeR/Max	**0.01**	**0.05**	**0.082**	**0.18**	**0**
IPeR/3Har	**0.01**	**0.04**	**0.176**	**0.12**	**0**
IPeR/2Har	**0.01**	**0.04**	**0.179**	**0.12**	**0**

2. Setting 2

50 users

App	Forwarders	Delay	F-measure	Cost
IPeR	**43.20**	**0.060**	**0.73**	**38**
3Avg_MIPeR	**48.25**	**0.046**	**0.68**	**49**
3Max_MIPeR	**48.35**	**0.045**	**0.68**	**49**
3Har_MIPeR	**48.25**	**0.047**	**0.68**	**44**
2Har_MIPeR	**48.20**	**0.047**	**0.68**	**42**
IPeR/Avg	**0.12**	**0.234**	**0.07**	**0**
IPeR/Max	**0.12**	**0.254**	**0.07**	**0**
IPeR/3Har	**0.12**	**0.215**	**0.07**	**0**
IPeR/2Har	**0.12**	**0.210**	**0.07**	**0**

100 users

App	Forwarders	Delay	F-measure	Cost
IPeR	**45.58**	**0.059**	**0.71**	**82**
3Avg_MIPeR	**48.40**	**0.048**	**0.67**	**99**
3Max_MIPeR	**48.53**	**0.048**	**0.67**	**100**
3Har_MIPeR	**48.35**	**0.049**	**0.67**	**92**
2Har_MIPeR	**48.33**	**0.050**	**0.67**	**86**
IPeR/Avg	0.06	0.182	0.07	0
IPeR/Max	0.06	0.187	0.07	0
IPeR/3Har	0.06	0.171	0.06	0
IPeR/2Har	0.06	0.151	0.06	0

200 users

App	Forwarders	Delay	F-measure	cost
IPeR	**47.36**	**0.042**	**0.72**	**168**
3Avg_MIPeR	**48.49**	**0.041**	**0.66**	**198**
3Max_MIPeR	**48.96**	**0.040**	**0.66**	**200**
3Har_MIPeR	**48.49**	**0.041**	**0.67**	**177**
2Har_MIPeR	**48.44**	**0.041**	**0.67**	**167**
IPeR/Avg	0.02	0.019	0.08	0
IPeR/Max	0.03	0.036	0.08	0
IPeR/3Har	0.02	0.010	0.07	0
IPeR/2Har	0.02	0.010	0.07	0

3. Setting 3

50 users

App	Destinations	Delay	F-measure	Cost
IPeR	**19.40**	**0.191**	**0.76**	**16**
3Avg_MIPeR	**19.50**	**0.132**	**0.33**	**49**
3Max_MIPeR	**19.85**	**0.151**	**0.34**	**49**
3Har_MIPeR	**19.50**	**0.170**	**0.39**	**40**
2Har_MIPeR	**19.50**	**0.171**	**0.41**	**39**
IPeR/Avg	0.01	0.307	0.56	2
IPeR/Max	0.02	0.211	0.56	2
IPeR/3Har	0.01	0.113	0.48	1
IPeR/2Har	0.01	0.103	0.46	1

100 users

App	Destinations	Delay	F-measure	cost
IPeR	**19.55**	**0.075**	**0.74**	**33**
3Avg_MIPeR	**19.88**	**0.071**	**0.33**	**99**
3Max_MIPeR	**20.00**	**0.075**	**0.33**	**100**
3Har_MIPeR	**19.55**	**0.059**	**0.39**	**82**
2Har_MIPeR	**19.55**	**0.062**	**0.39**	**82**
IPeR/Avg	0.02	0.048	0.55	2
IPeR/Max	0.02	0.007	0.55	2
IPeR/3Har	0.00	0.206	0.48	1
IPeR/2Har	0.00	0.166	0.48	1

200 users

App	Destinations	Delay	F-measure	cost
IPeR	**19.80**	**0.046**	**0.69**	**77**
3Avg_MIPeR	**19.84**	**0.041**	**0.33**	**198**
3Max_MIPeR	**19.99**	**0.042**	**0.33**	**200**
3Har_MIPeR	**19.80**	**0.046**	**0.42**	**161**
2Har_MIPeR	**19.80**	**0.046**	**0.42**	**161**
IPeR/Avg	0.00	0.116	0.51	2
IPeR/Max	0.01	0.088	0.52	2
IPeR/3Har	0.00	0.009	0.39	1
IPeR/2Har	0.00	0.011	0.39	1

Appendix C: Directional-IPeR results

Setting 1

50 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.60	29.90	0.107	0.85	29.20	0
Dir-IPeR_75	17.60	35.00	0.095	0.86	42.00	36
Dir-IPeR_50	17.70	35.60	0.069	0.75	44.50	51
Dir-IPeR_25	17.85	35.70	0.061	0.75	44.70	54
Dir-IPeR_0	17.85	35.70	0.056	0.70	49.70	67
IPeR/Dir-IPeR_75	0.00	0.17	0.112	0.01	0.44	
IPeR/Dir-IPeR_50	0.01	0.19	0.355	0.12	0.52	
IPeR/Dir-IPeR_25	0.01	0.19	0.430	0.12	0.53	
IPeR/Dir-IPeR_0	0.01	0.19	0.477	0.18	0.70	

100 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.88	32.68	0.060	0.85	64.55	0
Dir-IPeR_75	17.88	35.33	0.058	0.86	70.35	81
Dir-IPeR_50	18.00	35.93	0.053	0.75	89.85	122
Dir-IPeR_25	18.00	35.98	0.052	0.75	89.90	138
Dir-IPeR_0	18.00	35.90	0.050	0.70	100.00	182
IPeR/Dir-IPeR_75	0.00	0.08	0.033	0.01	0.09	
IPeR/Dir-IPeR_50	0.01	0.10	0.117	0.12	0.39	
IPeR/Dir-IPeR_25	0.01	0.10	0.133	0.12	0.39	
IPeR/Dir-IPeR_0	0.01	0.10	0.167	0.18	0.55	

200 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.80	34.74	0.039	0.86	0.68	0
Dir-IPeR_75	17.99	36.49	0.044	0.86	0.72	189
Dir-IPeR_50	18.00	36.49	0.043	0.75	0.90	325
Dir-IPeR_25	18.00	36.49	0.042	0.75	0.90	389
Dir-IPeR_0	18.00	36.49	0.035	0.71	1.00	529
IPeR/Dir-IPeR_75	0.01	0.05	0.128	0.00	0.05	
IPeR/Dir-IPeR_50	0.01	0.05	0.103	0.13	0.33	
IPeR/Dir-IPeR_25	0.01	0.05	0.077	0.13	0.33	
IPeR/Dir-IPeR_0	0.01	0.05	0.103	0.17	0.47	

Setting 2

50 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	43.20	0.060	0.73	38.00	0
Dir-IPeR_75	48.45	0.045	0.73	43.75	50
Dir-IPeR_50	48.75	0.044	0.68	49.65	68
Dir-IPeR_25	48.75	0.044	0.68	49.70	72
Dir-IPeR_0	48.75	0.044	0.68	49.70	73
IPeR/Dir-IPeR_75	0.12	0.250	0.00	0.15	
IPeR/Dir-IPeR_50	0.13	0.267	0.07	0.31	
IPeR/Dir-IPeR_25	0.13	0.267	0.07	0.31	
IPeR/Dir-IPeR_0	0.13	0.267	0.07	0.31	

100 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	45.58	0.059	0.71	81.78	0
Dir-IPeR_75	48.55	0.049	0.72	88.23	133
Dir-IPeR_50	48.55	0.048	0.67	97.98	194
Dir-IPeR_25	48.55	0.048	0.67	97.98	211
Dir-IPeR_0	48.55	0.048	0.66	99.95	222
IPeR/Dir-IPeR_75	0.07	0.169	0.01	0.08	
IPeR/Dir-IPeR_50	0.07	0.186	0.06	0.20	
IPeR/Dir-IPeR_25	0.07	0.186	0.06	0.20	
IPeR/Dir-IPeR_0	0.07	0.186	0.07	0.22	

200 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	47.36	0.042	0.72	168.18	0
Dir-IPeR_75	48.98	0.041	0.72	177.48	336
Dir-IPeR_50	48.98	0.040	0.67	196.95	515
Dir-IPeR_25	48.98	0.040	0.67	196.95	564
Dir-IPeR_0	48.98	0.040	0.66	199.95	595
IPeR/Dir-IPeR_75	0.03	0.024	0.00	0.06	
IPeR/Dir-IPeR_50	0.03	0.048	0.07	0.17	
IPeR/Dir-IPeR_25	0.03	0.048	0.07	0.17	
IPeR/Dir-IPeR_0	0.03	0.048	0.08	0.19	

Setting 3

50 users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.40	0.191	0.76	16.00	0
Dir-IPeR_75	19.40	0.128	0.57	24.73	25
Dir-IPeR_50	19.85	0.105	0.39	41.35	48
Dir-IPeR_25	19.95	0.077	0.39	41.55	51
Dir-IPeR_0	19.95	0.071	0.33	49.70	66
IPeR/Dir-IPeR_75	0.00	0.330	0.25	0.55	
IPeR/Dir-IPeR_50	0.02	0.450	0.49	1.58	
IPeR/Dir-IPeR_25	0.03	0.597	0.49	1.60	
IPeR/Dir-IPeR_0	0.03	0.628	0.57	2.11	

100 users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.55	0.075	0.74	33.45	0
Dir-IPeR_75	19.55	0.060	0.57	48.90	54
Dir-IPeR_50	19.98	0.057	0.38	83.90	121
Dir-IPeR_25	19.98	0.055	0.38	83.95	129
Dir-IPeR_0	19.98	0.048	0.33	99.95	190
IPeR/Dir-IPeR_75	0.00	0.200	0.23	0.46	
IPeR/Dir-IPeR_50	0.02	0.240	0.49	1.51	
IPeR/Dir-IPeR_25	0.02	0.267	0.49	1.51	
IPeR/Dir-IPeR_0	0.02	0.360	0.55	1.99	

200 users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.80	0.046	0.69	77.43	0
Dir-IPeR_75	19.90	0.051	0.56	102.83	122
Dir-IPeR_50	19.99	0.045	0.39	167.63	315
Dir-IPeR_25	19.99	0.043	0.39	167.63	333
Dir-IPeR_0	19.99	0.042	0.33	199.95	512
IPeR/Dir-IPeR_75	0.01	0.109	0.19	0.33	
IPeR/Dir-IPeR_50	0.01	0.022	0.43	1.16	
IPeR/Dir-IPeR_25	0.01	0.065	0.43	1.16	
IPeR/Dir-IPeR_0	0.01	0.087	0.52	1.58	

Appendix D: Hybrid of MIPeR and Directional-IPeR (DMIPeR)

Setting 1

50 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.60	29.90	0.107	0.85	29.20	0
Dir-IPeR_75	17.60	35.00	0.095	0.86	42.00	36
DMIPeR_75	17.60	35.00	0.095	0.86	42.00	36
DMIPeR_50	17.70	35.60	0.069	0.75	49.00	51
DMIPeR_25	17.85	35.70	0.061	0.75	44.00	54
DMIPeR_0	17.85	35.70	0.056	0.70	50.00	67
DMIPeR_75/IPeR	0.00	0.17	0.112	0.01	0.00	
DMIPeR_50/IPeR	0.01	0.19	0.355	0.12	0.17	
DMIPeR_25/IPeR	0.01	0.19	0.430	0.12	0.05	
DMIPeR_0/IPeR	0.01	0.19	0.477	0.18	0.19	

100 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.88	32.68	0.060	0.85	64.55	0
Dir-IPeR_75	17.88	35.33	0.058	0.86	70.35	81
DMIPeR_75	17.88	35.33	0.058	0.86	96.00	81
DMIPeR_50	18.00	35.93	0.053	0.75	98.00	122
DMIPeR_25	18.00	35.98	0.052	0.75	98.00	138
DMIPeR_0	18.00	35.90	0.050	0.70	100.00	182
DMIPeR_75/IPeR	0.00	0.08	0.033	0.01	0.36	
DMIPeR_50/IPeR	0.01	0.10	0.117	0.12	0.39	
DMIPeR_25/IPeR	0.01	0.10	0.133	0.12	0.39	
DMIPeR_0/IPeR	0.01	0.10	0.167	0.18	0.42	

200 users

Algorithm	Destinations	Forwarders	Delay	F-measure	Cost	Control
IPeR	17.80	34.74	0.039	0.86	137.00	0
Dir-IPeR_75	17.99	36.49	0.044	0.86	144.00	189
DMIPeR_75	17.99	36.49	0.044	0.86	189.00	229
DMIPeR_50	18.00	36.49	0.043	0.75	181.00	325
DMIPeR_25	18.00	36.49	0.042	0.75	181.00	389
DMIPeR_0	18.00	36.49	0.035	0.71	200.00	529
DMIPeR_75/IPeR	0.01	0.05	0.128	0.00	0.31	
DMIPeR_50/IPeR	0.01	0.05	0.103	0.13	0.26	
DMIPeR_25/IPeR	0.01	0.05	0.077	0.13	0.26	
DMIPeR_0/IPeR	0.01	0.05	0.103	0.17	0.39	

Setting 2

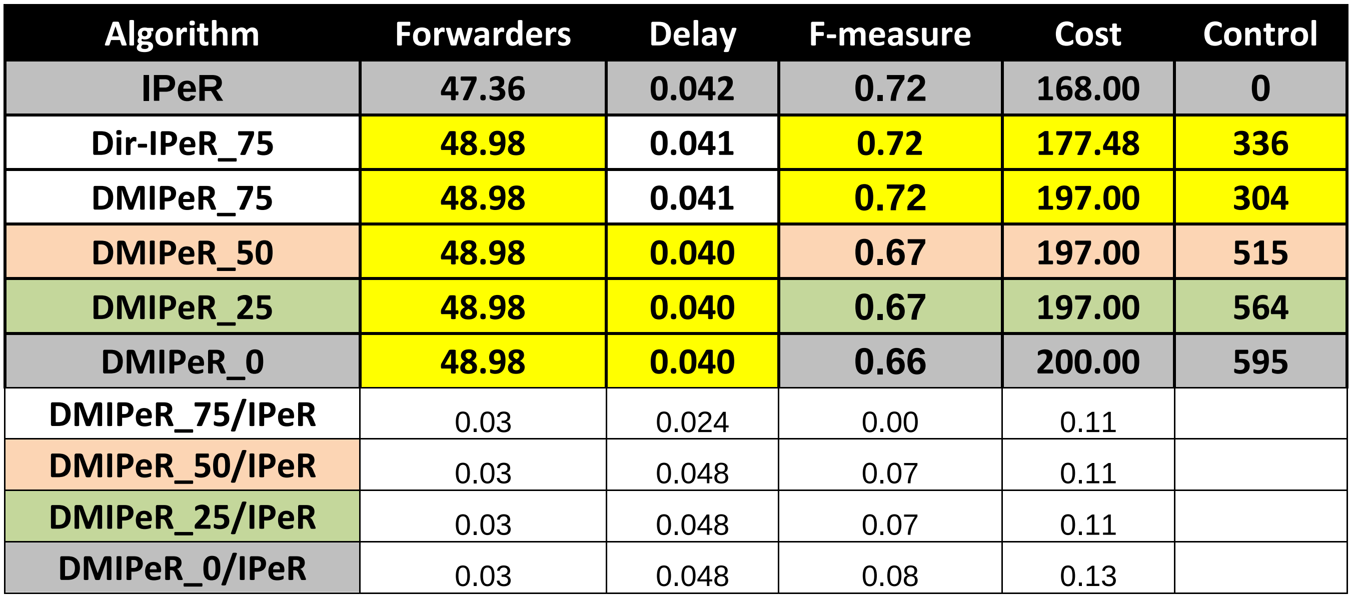
50 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	43.20	0.060	0.73	38.00	0
Dir-IPeR_75	48.45	0.045	0.73	43.75	50
DMIPeR_75	48.45	0.045	0.73	49.35	50
DMIPeR_50	48.75	0.044	0.68	50.00	68
DMIPeR_25	48.75	0.044	0.68	50.00	72
DMIPeR_0	48.75	0.044	0.68	50.00	73
DMIPeR_75/IPeR	0.12	0.250	0.00	0.13	
DMIPeR_50/IPeR	0.13	0.267	0.07	0.14	
DMIPeR_25/IPeR	0.13	0.267	0.07	0.14	
DMIPeR_0/IPeR	0.13	0.267	0.07	0.14	

100 users

Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	45.58	0.059	0.71	82.00	0
Dir-IPeR_75	48.55	0.049	0.72	88.23	133
DMIPeR_75	48.55	0.049	0.72	98.00	133
DMIPeR_50	48.55	0.048	0.67	98.00	194
DMIPeR_25	48.55	0.048	0.67	98.00	211
DMIPeR_0	48.55	0.048	0.66	100.00	222
DMIPeR_75/IPeR	0.07	0.169	0.01	0.11	
DMIPeR_50/IPeR	0.07	0.186	0.06	0.11	
DMIPeR_25/IPeR	0.07	0.186	0.06	0.11	
DMIPeR_0/IPeR	0.07	0.186	0.07	0.13	

200 users



Algorithm	Forwarders	Delay	F-measure	Cost	Control
IPeR	47.36	0.042	0.72	168.00	0
Dir-IPeR_75	48.98	0.041	0.72	177.48	336
DMIPeR_75	48.98	0.041	0.72	197.00	304
DMIPeR_50	48.98	0.040	0.67	197.00	515
DMIPeR_25	48.98	0.040	0.67	197.00	564
DMIPeR_0	48.98	0.040	0.66	200.00	595
DMIPeR_75/IPeR	0.03	0.024	0.00	0.11	
DMIPeR_50/IPeR	0.03	0.048	0.07	0.11	
DMIPeR_25/IPeR	0.03	0.048	0.07	0.11	
DMIPeR_0/IPeR	0.03	0.048	0.08	0.13	

Setting 3

50 users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.40	0.191	0.76	16.00	0
Dir-IPeR_75	19.40	0.128	0.57	24.73	25
DMIPeR_75	19.40	0.128	0.57	42.00	25
DMIPeR_50	19.85	0.105	0.39	49.00	48
DMIPeR_25	19.95	0.077	0.39	44.00	51
DMIPeR_0	19.95	0.071	0.33	50.00	66
DMIPeR_75/IPeR	0.00	0.330	0.25	0.70	
DMIPeR_50/IPeR	0.02	0.450	0.49	0.98	
DMIPeR_25/IPeR	0.03	0.597	0.49	0.78	
DMIPeR_0/IPeR	0.03	0.628	0.57	1.02	

100 users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.55	0.075	0.74	33.00	0
Dir-IPeR_75	19.55	0.060	0.57	48.90	54
DMIPeR_75	19.55	0.060	0.57	83.00	90
DMIPeR_50	19.98	0.057	0.38	84.00	121
DMIPeR_25	19.98	0.055	0.38	97.00	129
DMIPeR_0	19.98	0.048	0.33	100.00	190
DMIPeR_75/IPeR	0.00	0.200	0.23	0.70	
DMIPeR_50/IPeR	0.02	0.240	0.49	0.72	
DMIPeR_25/IPeR	0.02	0.267	0.49	0.98	
DMIPeR_0/IPeR	0.02	0.360	0.55	1.04	

200 users

Algorithm	Destinations	Delay	F-measure	Cost	Control
IPeR	19.80	0.05	0.69	77.00	0.00
Dir-IPeR_75	19.90	0.05	0.56	102.83	122.00
DMIPeR_75	19.90	0.05	0.56	166.00	191.00
DMIPeR_50	19.99	0.05	0.39	168.00	315.00
DMIPeR_25	19.99	0.04	0.39	168.00	333.00
DMIPeR_0	19.99	0.04	0.33	200.00	512.00
DMIPeR_75/IPeR	0.01	0.109	0.19	0.61	
DMIPeR_50/IPeR	0.01	0.022	0.43	0.63	
DMIPeR_25/IPeR	0.01	0.065	0.43	0.63	
DMIPeR_0/IPeR	0.01	0.087	0.52	0.94	