Chapter 5
Performance Evaluation of Fuzzy-Based Routing Protocols for Opportunistic Networks

Khuram Khalid, Isaac Woungang, Sanjay K. Dhurandher, and Jagdeep Singh

5.1 Introduction

Opportunistic networks (OppNets) are characterized by periodic connectivity and high node mobility [1]. In this context, the message in such networks is carried progressively from source to destination using the store-carry-and-forward mechanism [2] whereby a node that receives the data packets stores them in its local buffer (if space is available), awaiting for a forwarding opportunity to pass these packets to a suitable one-hop node qualified to carry them towards the destination. If no space is available in the node's buffer, the message is considered as lost in the network.

Various routing protocols for OppNets implementing the aforementioned store-carry-and-forward strategy [2] in conjunction with other techniques have been investigated in the literature. Representative ones are reported in [3–13]. This chapter focuses on fuzzy-based routing schemes for OppNets. Precisely, it reports on additional simulation studies of three recently proposed fuzzy-based routing protocols under both synthetic and real mobility traces, namely the RLFGRP [3], FCSG [4], and MARL-CC [5] schemes, in terms of delivery ratio, overhead ratio,

K. Khalid (✉) · I. Woungang
Department of Computer Science, Ryerson University, Toronto, ON, Canada
e-mail: khuram.khalid@ryerson.ca; iwoungan@ryerson.ca

S. K. Dhurandher
Department of Information Technology, Netaji Subhas University of Technology, New Delhi, Delhi, India

J. Singh
Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India
e-mail: Jagdeep@sliet.ac.in

I. Woungang, S. K. Dhurandher, *4th International Conference on Wireless, Intelligent and Distributed Environment for Communication*, Lecture Notes on Data Engineering and Communications Technologies 94, https://doi.org/10.1007/978-3-030-89776-5_5

and average latency, under varying number of nodes, time to live (TTL), and buffer size.

The remainder of the chapter is organized as follows. Section 5.2 presents an overview of the studied protocols. In Sect. 5.3, a performance comparison of the studied protocols is presented. Section 5.4 concludes the chapter.

5.2 Overview of Protocols

In this section, an overview of the studied routing protocols for OppNets is provided.

5.2.1 RLFGRP Protocol

The following notations are used in the FCSG algorithm:

- SN: Sender node
- RN: Receiver node
- MFNL: Message Forwarding Nodes List (Hashtable)
- m: Message
- MFS: Message Forwarding Set

The design of the Reinforcement Learning-based Fuzzy Geocast Routing Protocol (RLFGRP) [3] is illustrated in Algorithm 1. The scheme consists of two phases. In the first phase, a Q-learning mechanism is applied for the forwarding of the message towards the destination cast. More precisely, when a source node, say S, creates a message m to be sent to a destination node, say D, the destination geocast region is instantly defined, along with the associated cast definition, which represents an ensemble of two-dimensional points forming the geographic cast and a pair of epoch times defining the message lifetime.

Whenever a forwarding opportunity arises, i.e., the source node S encounters an intermediate node, say N, the likelihood of N to carry the message towards the destination cast is calculated by means of a fuzzy controller which takes the Q-value, reward value, and remaining buffer space of N as input parameters. Typically, the considered Q-learning mechanism is a continuous process out of which a reward (or penalty in the form of reduced reward) is assigned to node N (according to a reward function), along with its Q-value, based on the action it has taken. It should be noted that this reward function takes the Euclidean distance from node N to the source node S, the delivery probability of N, and the direction of N with respect to destination node D, as input parameters. Besides, during this Q-learning iterative process, the update of the Q-value of a node is done on the basis of its assigned reward value, its action, and the algorithm's discount and learning rates. Once the likelihood of any intermediate encountered node is calculated, it is verified whether that node belongs or not to a prescribed Message Forwarding Set (MFS), and if that

Algorithm 1 RLFGRP routing scheme [3]

```
Initialization during message generation:
m is created
Sender node and destination geocast region are defined

Phase 1: Moving the message towards its destination cast
for each SN that meets the next RN do
    Drop the expired messages from buffer
    for each m in the SN buffer do
        if m already exists in RN then
            skip and go to the next m in the for loop
        else if If m is not in the geocast region then
            for each RN in the neighbours of SN do
                Calculate the likelihood of RN
                if likelihood is from MFS then
                    Store RN in MFNL List
                end if
            end for
            if MNFL is empty then
                skip and go to the next m in the for loop
            end if
            for  each node J in MFNL do
                Update the Q-Value of RN
            end for
            if Q-value of RN is > than J's q-value then
                Forward message to the current RN
                Update the reward of the current RN
            end if
Phase 2: Distributing the message copy to all nodes within the geocast region
        else if m is in geocast region then
            if RN already exists in geocast region then
                if m does not exist in RN then
                    Forward a copy of m to RN
                end if
            end if
        end if
    end for
end for
```

is the case, that node's ID is referenced in a hash table and the Q-values of all nodes in this table are updated, then the node in this list which has the highest Q-value is selected as best forwarder of the message m towards its destination. In the second phase, a Check-and-Spray mechanism similar to that used in [13] is implemented to intelligently flood the message within the geocast region, hoping that it will reach its intended destination.

5.2.2 FCSG Protocol

The following notations are used in the FCSG algorithm:

- C: Number of remaining replicas of a message
- VH: Very high
- H: High
- M: Medium
- L: Low
- VL: Very low
- SN: Sender node
- RN: Receiver node
- m: Message
- L(): Likelihood function

The design of the Fuzzy-based Check-and-Spray Geocast Routing Protocol (FCSGRP) for opportunistic networks [4] is illustrated in Algorithm 2. The scheme consists of two phases. In the first phase, a multi-copying spray mechanism is applied for the forwarding of the message towards the destination cast. More precisely, similar to the previous RLFGRP scheme, when a source node, say S, creates a message m to be sent to a destination node, say D, it is initially allocated a payload data. The destination geocast region is also instantly defined, along with the cast definition, and *C* an integer value representing the maximum number of remaining message copies that can be generated by S and forwarded eventual encountered nodes during the whole message forwarding process is set.

Whenever a forwarding opportunity arises, i.e., the source node S opportunistically meets an intermediate node, the likelihood of this node to carry the message towards the destination cast is calculated by using the series of two fuzzy controllers. In this process, the first controller considers as inputs the speed and direction of that intermediate node to determine its movement, noting that the direction here is defined as the angle between the line joining node S to the centre of the geocast region and the line on which this intermediate node currently moves. The second controller uses the movement, remaining energy, and buffer space of that intermediate node as inputs to calculate its likelihood to carry the message towards the destination cast. Once the likelihood of any intermediate encountered node, say N, is calculated, it is verified whether this value is greater than the likelihood of the source node S. If that is the case, the message m is forwarded to node N and its *C* value gets decreased based on the likelihood fuzzy controller's output; otherwise, node N is deemed as not qualified to receive the message and the value of *C* is kept unchanged, and the search for a newly suitable intermediate node to carry the message towards the destination cast recommences. This process stands as long as the value of *C* is not equal to 0. When this value becomes 0 and the message

Algorithm 2 FCSG routing scheme [4]

```
Initialization during message generation:
m is created
Sender node and destination geocast region are defined
C is initialized

Phase 1: Moving a message towards its destination cast
for each SN that meets the next RN do
    drop expired messages from buffer
    for each m in the SN buffer do
        if m already exists in RN then
            skip this m and go to the next m in the for loop
        end if
        if C > 0 then
            if RN is located in geocast area then
                forward message to RN
                C = 0
                // In case RN is not located within the
                // geocast region, calculate the likelihood of
                // RN to receive the message.
            else if L(RN,m) > L(SN,m) then
                forward a copy of message to RN
                switch L(RN,m) do
                    case VH: C = C-5, break;
                    case H: C = C-4, break;
                    case M : C = C-3, break;
                    case L : C = C-2, break;
                    case VL : C = C-1, break;
            end if
Phase 2: Distributing message copy to all nodes inside the geocast
        else if C = 0 and m is in geocast region then
            if RN already exists in geocast region then
                if m does not already exist in RN then
                    forward a copy of m to RN
                end if
            end if
        end if
    end for
end for
```

has not yet reached the destination cast, no more message copies are generated and the message is considered as lost in the network. In the second phase, the same Check-and-Spray mechanism described in [13] is used as controlled flooding technique within the geocast region, hoping that the message will eventually get to its destination.

5.2.3 *MARL-CC Protocol*

The design of the Multi-Agent Reinforcement Learning Congestion Control (MARL-CC) [5] involves the use multiple agents present in the network environment to perform the routing of the message. As such, the messages in the network are considered as the agents and the nodes that participate in the message routing from source to destination are considered as the states. The protocol works as follows.

Whenever a message (agent) originated from a source node arrives at a given node (i.e., intermediate node), the Q-table's row for the address of the destination node is searched up and a global Q-table representing the set of all tables to be used in the routing process is built and learned using the so-called Train Routers algorithm. In this learning process, the Q-tables are meant to guide the agent to the next best forwarder nodes to carry it to its destination in the sense that the agent is most likely directed towards a routing path that achieves the maximum reward globally based on its action's effectiveness in controlling the network congestion level. Next, once the Q-tables for all nodes are populated, a Q-learning algorithm is invoked which considers the set of IDs of the neighbouring nodes of a node, the message to be forwarded, and the message destination, as input parameters to calculate (and update) the Q-value of nodes. Based on this, intelligent routing decisions regarding the selection of candidate forwarders for the message are determined by means of an implemented congestion controlled optimal policy that makes these routing decisions also contribute to the improvement of the message delivery probability in the sense that the network overhead is shown through simulations to be significantly reduced. The pseudocode of the MARL-CC algorithm is given in [5].

5.3 Performance Evaluation

In this section, the studied routing protocols for OppNets, namely RLFGRP [3], FCSG [4], and MARL-CC [5], are simulated and compared in terms of delivery ratio, average latency, and overhead ratio, under varying number of hosts, buffer size, and time to live (TTL). In this work, the delivery ratio is calculated as the ratio of the messages successfully delivered to the destination at the end of the simulations and the total number of messages generated in the network. The overhead ratio is a measure of the bandwidth efficiency, calculated as

$$\frac{\text{Number of relayed messages} - \text{Number of delivered messages}}{\text{Number of delivered messages}} \tag{5.1}$$

Algorithm 3 MARL-CC Routing

Input:
msgList[1..numNodes] = null,
neighbors[1..numNodes] = null,
neighborsPrev[1..numNodes] = null,
destNode[1..numMessages],
global$Q_{[nodeA]}$[destNode[message]][nodeB],global$Q_{[nodeB]}$[destNode[message]][nodeA], nodeA, nodeB
Procedure
for each message in msgList[nodeA] **do**
    if global$Q_{[nodeA]}$[destNode[message]][nodeB] is updated **then**
        Calculate reward using congestion factor CF_x
        select nextHop by Boltzmann Exploration scheme on
        the set {(x, global$Q_{[nodeA]}$[destNode[message]][x])
        textbar x ∈ neighbors[nodeA]}
        forwardMessage(message, nodeA, nextHop)
    else
        Update global$Q_{[nodeA]}$[destNode[message]][nodeB]
    end if
end for
for each message in msgList[nodeB] **do**
    if global$Q_{[nodeB]}$[destNode[message]][nodeA] is updated **then**
        Calculate reward using congestion factor CF_x
        select nextHop by Boltzmann Exploration scheme on the
        set {(x, global$Q_{[nodeB]}$[destNode[message]][x])
        |x ∈ neighbors[nodeB]}
        forwardMessage(message, nodeB, nextHop)
    else
        Update global$Q_{[nodeB]}$[destNode[message]][nodeA]
    end if
end for
update (msgList[nodeA], msgList[nodeB])
update (neighbors[nodeA], neighbors[nodeB]) and (neighborsPrev[nodeA],
neighborsPrev[nodeB])
return

Two mobility models are considered: (1) the Shortest Path Map-Based Movement (SPMBM) whose system model is a set of six groups of nodes, i.e., electric motor cars, pedestrian, bicycles, tram, vehicles, and office workers, that can leave or join the network at any time, and these nodes move on a shortest path determined by the Dijkstra algorithm, and (2) the INFOCOM 2006 dataset of real mobility traces [14]. The wireless interfaces used are Wi-Fi 802.11ac with a transmission speed of 433 Mbps and a range of 20 m and Bluetooth 802.16 v4.0, with a transmission speed of 2 Mbps and a range of 10 m. For the message scheduling, a sender and a destination cast are selected uniformly and randomly from a set of nodes and predefined casts.

Table 5.1 Simulation parameters

Terrain dimension	4500 × 3400 m, segmented into 16 casts
Simulation time	57,600 s
Warm-up and cool-down periods for every simulation	2 h each
Buffer sizes	10 (default), 5, 15, 20, 25, 30, 35, 40 (Mb)
Message lifetimes	120 (default), 30, 60, 90, 150, 180, 210, 240 (min)
Message payload	500 KB
Message generation	Every 25 to 35 s
Scheduling policy	Random
Number of nodes	195 (default), 126, 189, 252, 315, 378, 441, 504, 567
Node speed	0.5–1.5 m/s

Other simulation parameters are given in Table 5.1.

5.4 Simulation Results

5.4.1 SPMBM Model

Figure 5.1 shows that as the number of hosts is increased, the delivery ratio increases for all the studied protocols. This is due to the fact that the more the number of nodes in the network, the better the chance that the message be delivered. In terms of delivery ratio performance, RLFGRP is about 4.4% better than MARL-CC and 16.6% better than FCSG.

Figure 5.2 reveals that for all the studied protocols, the average latency decreases when the number of hosts is increased. This is the direct consequence of the increase in delivery ratio. It is also observed that in terms of average latency, RLFGRP outperforms the other protocols. Indeed, in terms of average latency performance, RLFGRP scheme is about 2.42% better than MARL-CC and 7.6% better than FCSG.

Figure 5.3 shows that when the buffer size of nodes is increased, the delivery ratio increases. This is due to the fact that as the buffer size increases, more messages are stored in a node, leading to a better delivery ratio of messages to their destinations. Indeed, in terms of delivery ratio performance, RLFGRP is about 3.6% better than MARL-CC and 7.5% better than FCSG.

Figure 5.4 shows that as the buffer size is increased, the average latency increases. This is due to the fact the messages tend to stay longer than expected in the node's buffer, causing some delay in its delivery to the destination. It is also observed that in terms of average latency performance, RLFGRP is about 5.9% better than MARL-CC and 17.9% better than FCSG.

Figure 5.5 shows that when the TTL is increased, the delivery ratio also increases, and this increase is more pronounced for RLFGRP. This is due to the fact that the

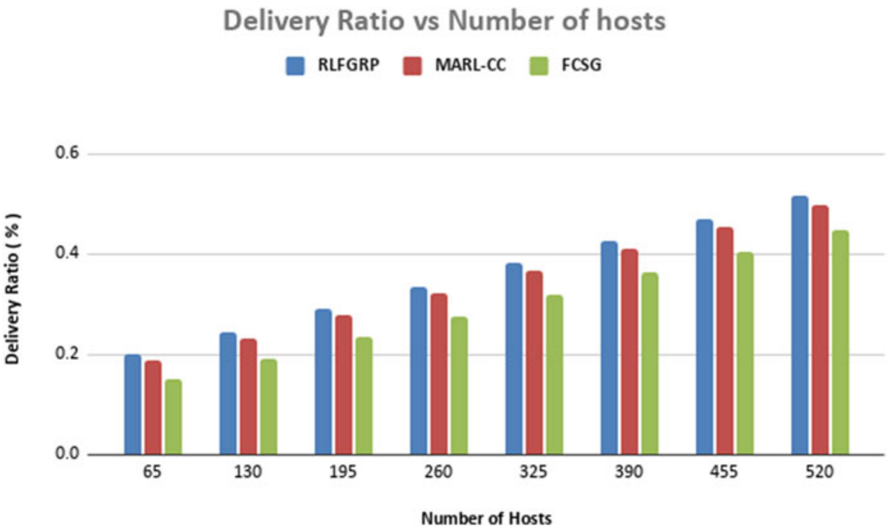

Fig. 5.1 Delivery ratio vs. number of hosts

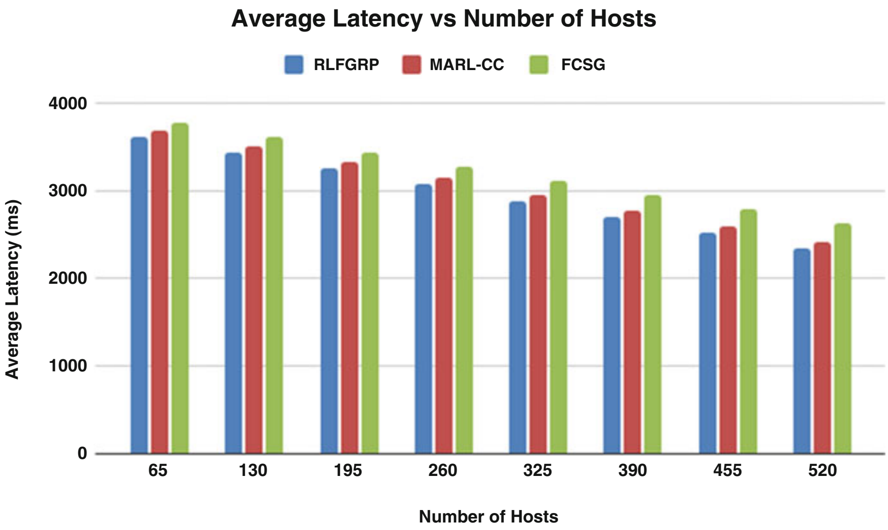

Fig. 5.2 Average latency ratio vs. number of hosts

messages get more time to find a suitable relay node to forward the message towards its destination. It is also observed that in terms of delivery ratio performance, RLFGRP scheme is about 2.4% better than MARL-CC and 13.6% better than FCSG.

In Fig. 5.6, it is observed that as the TTL increases, the average latency also increases for all the studied protocols, and this increase is less pronounced for

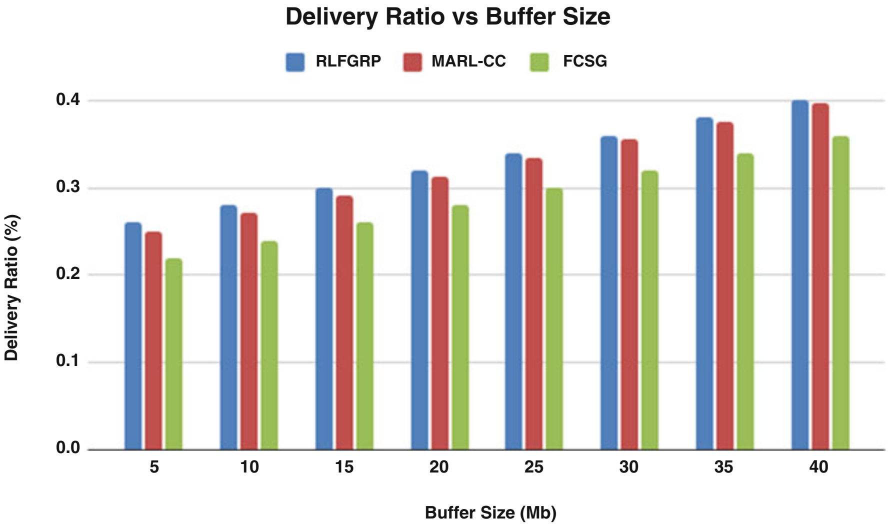

Fig. 5.3 Delivery ratio vs. buffer size

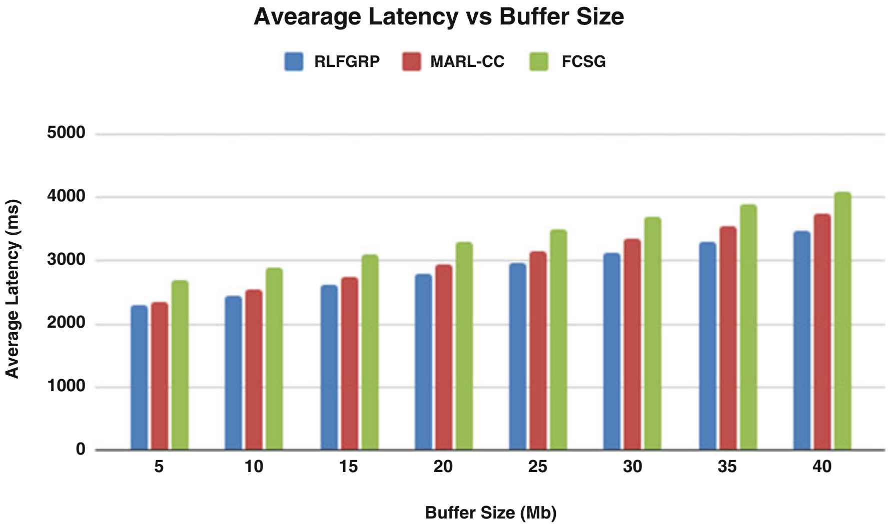

Fig. 5.4 Average Latency vs. buffer size

RLFGRP. This may be due to the fact that as the TTL of messages is increased, the number of forwarded messages also increases. In terms of average latency performance, it is found that RLFGRP is about 4.8% better than MARL-CC and 20.2% better than FCSG.

Figure 5.7 shows that for all studied protocols, as the number of hosts increases, the overhead ratio also increases. But this increase is less pronounced in the case

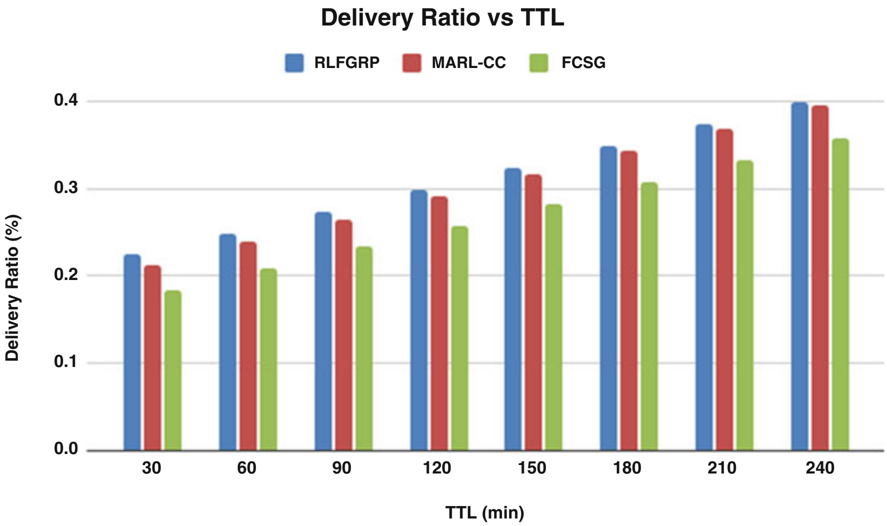

Fig. 5.5 Delivery Ratio vs. TTL

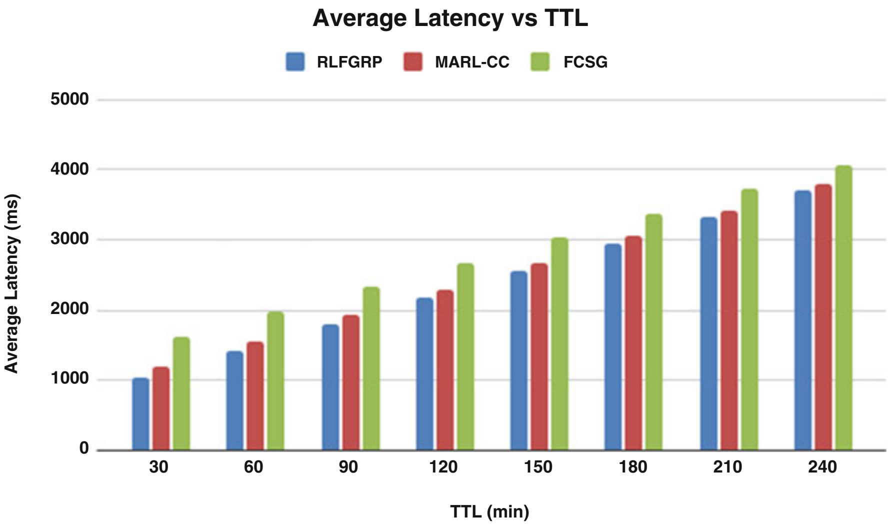

Fig. 5.6 Average latency vs. TTL

of RLFGRP. This might be attributed to the Check-and-Spray controlled flooding mechanism implemented in RLFGRP, which primarily imposes a threshold on the number of remaining message copies in the network to control the overhead reduction, by ensuring that the message once in the geocast region will not be spread outside that region. It is also observed that in terms of overhead ratio performance, RLFGRP is about 6.1% better than MARL-CC and 18.43% better than FCSG.

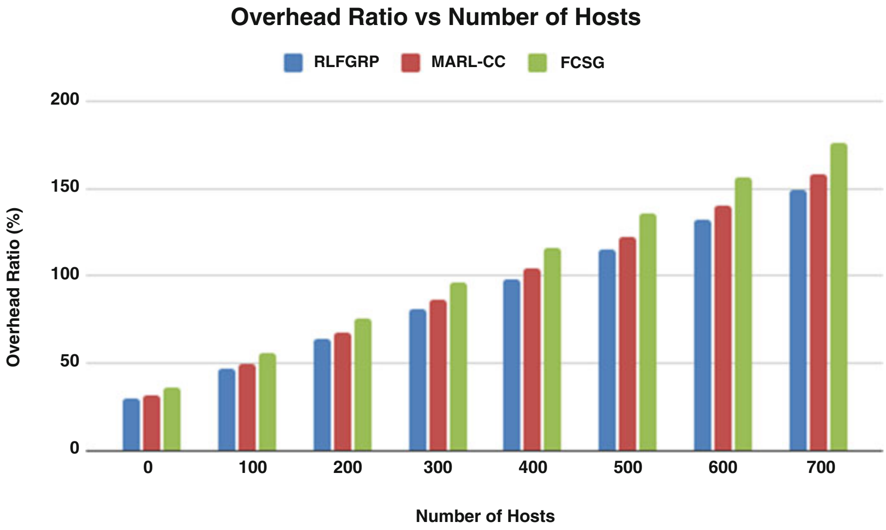

Fig. 5.7 Overhead ratio vs. number of hosts

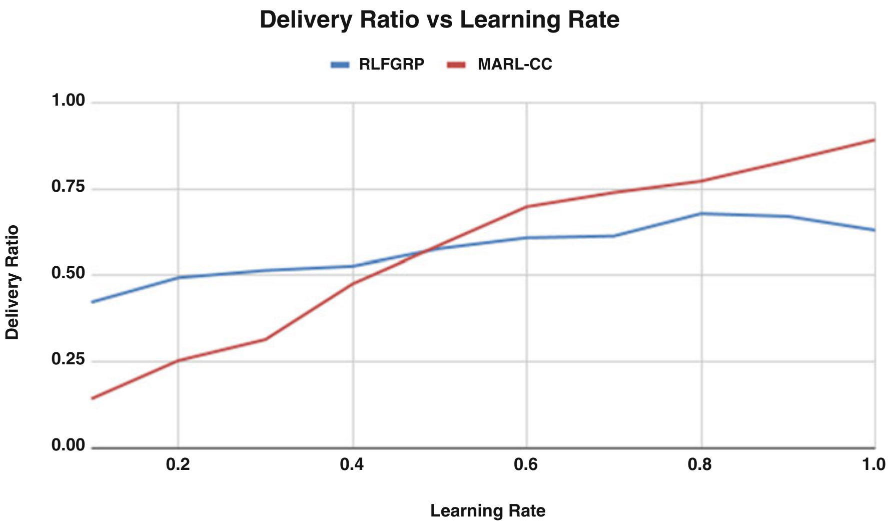

Fig. 5.8 Delivery ratio vs. learning rate

Figure 5.8 shows the effect of delivery ratio on the learning rate for the RLFGRP and MARL-CC learning algorithms. It is observed that RLFGRP performs well when the learning rate hits 0.8, whereas MARL-CC performs well when the learning rate hits 1.

5.4.2 *Real Mobility Traces Model*

In Fig. 5.9, it is found that all the studied protocols performed well when the buffer resources are sufficient and the delivery ratio increases as the buffer size is increased. It is also observed that in terms of delivery ratio performance, RLFGRP is about 1.4% better than MARL-CC and 6.85% better than FCSG.

In Fig. 5.10, it is observed that as the buffer size is increased, the average latency increases. It is also observed that in terms of average latency performance, RLFGRP is about 5.5% better than MARL-CC and 14.1% better than FCSG.

Figure 5.11 shows that when the TTL is increased, the delivery ratio also increases, and this increase is more pronounced for RLFGRP. This is due to the fact that the nodes get more time to find a suitable forwarder to carry the message towards its destination. It is also observed that in terms of delivery ratio performance, RLFGRP is about 2.0% better than MARL-CC and 15.7% better than FCSG.

In Fig. 5.12, it is observed that as the TTL increases, the average latency also increases for all studied protocols, and this increase is less pronounced for RLFGRP. It is also found that in terms of average latency performance, RLFGRP scheme is about 5.4% better than MARL-CC and 19.5% better than FCSG.

Figure 5.13 shows that for all the studied protocols, as the buffer size increases, the overhead ratio also increases. Moreover, in terms of overhead ratio performance, RLFGRP is about 5% better than MARL-CC and 11.5% better than FCSG.

Figure 5.14 shows that for all studied protocols, as the buffer size increases, the overhead ratio also increases. It is also observed that in terms of overhead ratio performance, RLFGRP is about 5.1% better than MARL-CC and 11.6% better than FCSG.

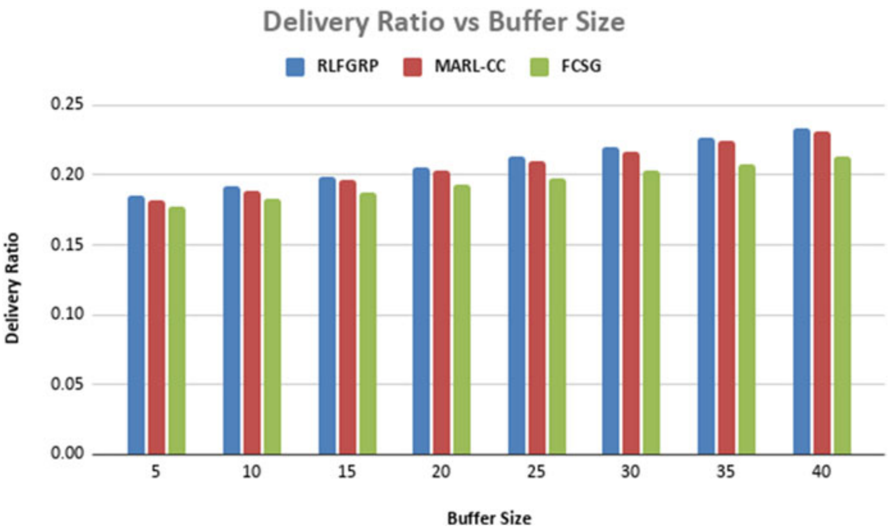

Fig. 5.9 Delivery ratio vs. buffer size

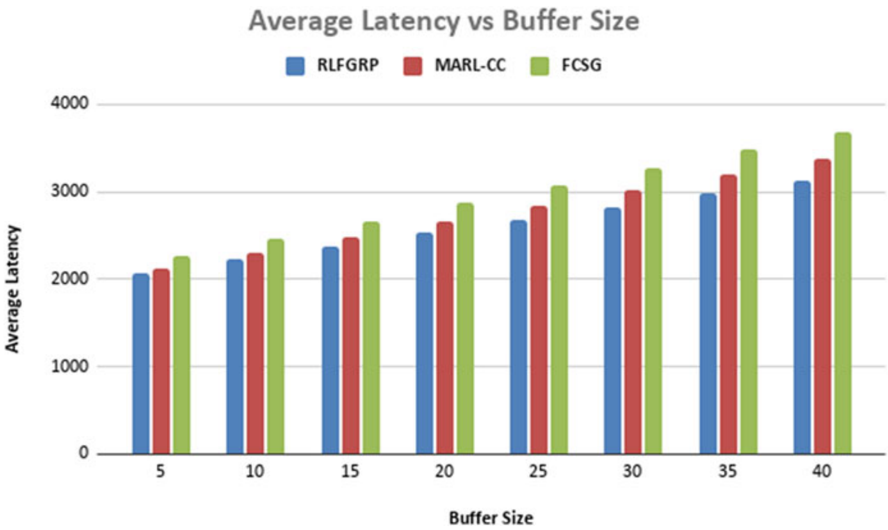

Fig. 5.10 Average latency vs. buffer size

Fig. 5.11 Delivery ratio vs. TTL

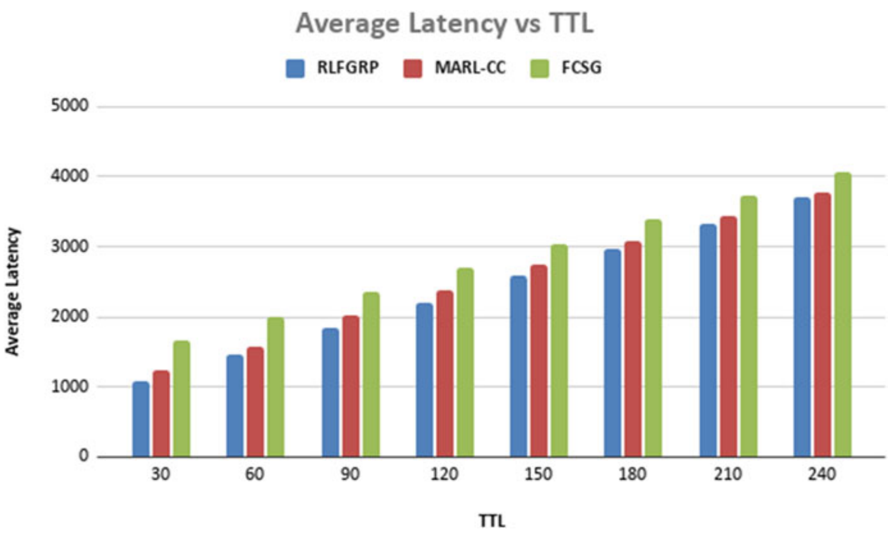

Fig. 5.12 Average latency vs. TTL

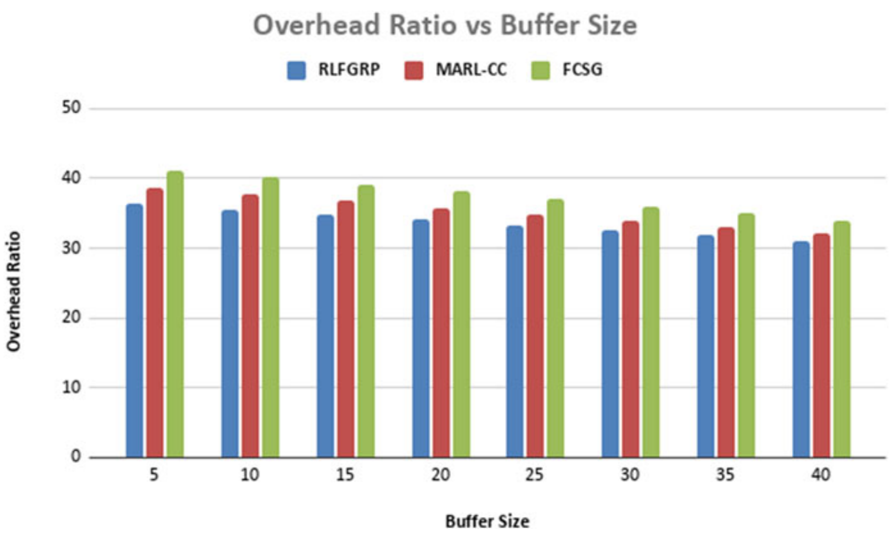

Fig. 5.13 Overhead ratio vs. buffer size

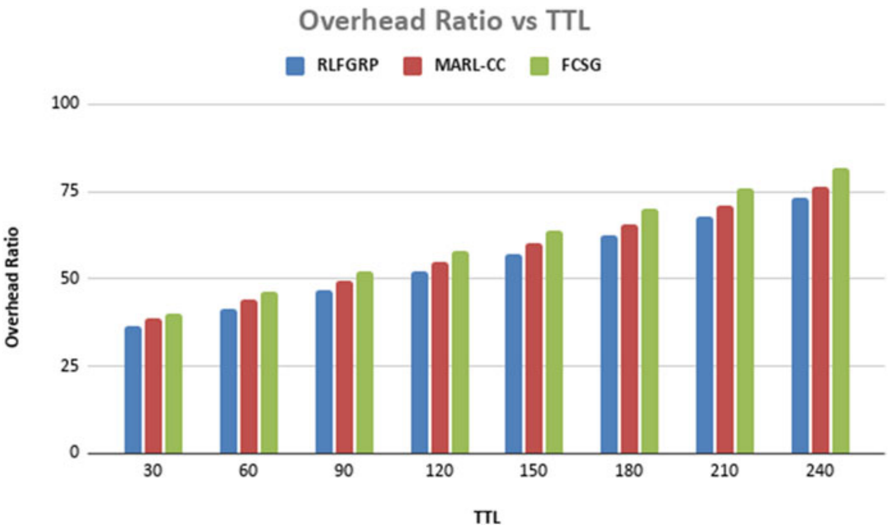

Fig. 5.14 Overhead ratio vs. TTL

5.5 Conclusion

In this chapter, we have compared three recently proposed fuzzy-based routing protocols for OppNets, namely RLFGRP, FCSG and MARL-CC, using the ONE simulator [15], under both the SPMBM model and the INFOCOM 2006 real mobility traces, considering the delivery ratio, average latency, and overhead ratio, as performance metrics. Simulation results have shown that RLFGRP outperforms FCSG and MARL-CC in terms of the above-mentioned metrics, under varying number of hosts, TTL, and buffer size. As future work, we plan to design the security-aware versions of the studied protocols and compare their resiliency against network attacks such as sybil attacks, wormhole attacks, and blackhole attacks.

Acknowledgments This work is partially sponsored by a grant held by the second author, from the National Science and Engineering Research Council of Canada (NSERC), reference number: RGPIN-2017-04423.

References

1. Lilien, L., Kamal, Z. H., Bhuse, V., & Gupta, A. (2007). The concept of opportunistic networks and their research challenges in privacy and security. In *Mobile and wireless network security and privacy* (pp. 85–117). Springer.
2. Huang, C. M., Lan, K. C., & Tsai, C. Z. (2008, March). A survey of opportunistic networks. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (AINA Workshops 2008)* (pp. 1672–1677). IEEE.

3. Khalid, K., Woungang, I., Dhurandher, S. K., & Singh, J. (2020). *Reinforcement learning-based fuzzy geocast routing protocol for opportunistic networks*. Accepted Feb. 25, 2021.
4. Khalid, K., Woungang, I., Dhurandher, S. K., Singh, J., & Barolli, L. (2021). A fuzzy-based check-and-spray geocast routing protocol for opportunistic networks. *Journal of High Speed Networks*, *27*(1), 1–12.
5. Singh, J., Dhurandher, S. K., & Woungang, I. (2020, December). Multi-agent reinforcement learning based efficient routing in opportunistic networks. In *2020 IEEE 17th India Council International Conference (INDICON)* (pp. 1–6). IEEE.
6. Kuppusamy, V., Thanthrige, U. M., Udugama, A., & Förster, A. (2019). Evaluating forwarding protocols in opportunistic networks: trends, advances, challenges and best practices. *Future Internet*, *11*(5), 113.
7. Rajaei, A., Chalmers, D., Wakeman, I., & Parisis, G. (2016, June). GSAF: Efficient and flexible geocasting for opportunistic networks. In *2016 IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (pp. 1–9). IEEE.
8. Dhurandher, S. K., Singh, J., Woungang, I., Takizawa, M., Gupta, G., & Kumar, R. (2019, November). Fuzzy geocasting in opportunistic networks. In *International Conference on Broadband and Wireless Computing, Communication and Applications* (pp. 279–292). Springer.
9. Rahimi, S., & Jamali, M. A. J. (2019). A hybrid geographic-DTN routing protocol based on fuzzy logic in vehicular ad hoc networks. *Peer-to-Peer Networking and Applications*, *12*(1), 88–101.
10. Jain, S., Chawla, M., Soares, V. N., & Rodrigues, J. J. (2016). Enhanced fuzzy logic-based spray and wait routing protocol for delay tolerant networks. *International Journal of Communication Systems*, *29*(12), 1820–1843.
11. Mottaghinia, Z., & Ghaffari, A. (2018). Fuzzy logic based distance and energy-aware routing protocol in delay-tolerant mobile sensor networks. *Wireless Personal Communications*, *100*(3), 957–976.
12. Banerjee, A., & Dutta, P. (2008). Fuzzy controlled adaptive geocasting in mobile ad hoc networks. *International Journal of Information Technology*, *14*(2), 109–131.
13. Khalid, K., Woungang, I., Dhurandher, S. K., Singh, J., & Rodrigues J. J. P. C. (2020). Energy-efficient check and spray geocast routing protocol for opportunistic networks. *Information*, *11*, 504, 1–18. https://doi.org/10.3390/info11110504
14. Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., & Chaintreau, A. (2009). CRAWDAD dataset Cambridge/haggle (v. 2009-05-29). CRAWDAD Wireless Network Data Archive.
15. Keränen, A., Ott, J., & Kärkkäinen, T. (2009, March). The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques* (pp. 1–10).