Wireless Pers Commun (2017) 94:859–888
DOI 10.1007/s11277-016-3654-5

Benchmarking and Modeling of Routing Protocols for Delay Tolerant Networks

Osman Khalid[1] · Rao Naveed Bin Rais[2] · Sajjad A. Madani[1]

Published online: 31 August 2016

Abstract Delay Tolerant Networks (DTN) are deployed to establish communications in challenging environments with frequent disruptions and delays due to intermittently connecting nodes, such as sparsely distributed wireless sensor networks and mobile ad hoc networks. Routing in such networks is difficult as nodes have little information about the state of the network that has time evolving topology. Therefore, nodes must store, carry, and forward messages towards destinations during opportunistic contacts. In recent years, numerous simulation based studies have been conducted for DTN protocols under various platforms, parameters, and mobility scenarios. However, most of the evaluations were limited in terms of: (a) number of protocols compared, (b) simulation parameters, and (c) DTN scenarios. This paper performs a detailed comparative analysis of ten popular DTN routing protocols. The protocols are benchmarked for the performance metrics, such as: (a) delivery ratio, (b) latency, and (c) message overhead, under the variance of: (a) buffer capacity, (b) message size, (c) message rate, and (d) size of network. The simulation results provide a deeper insight into a protocol's strengths and weaknesses under diverse network conditions. As a further contribution, we proposed enhancements in the models of three routing schemes for DTNs. The proposed schemes autonomously adapt to the varying network conditions to reduce the messages' replication frequency by finding optimal routes for messages among sources and destinations nodes. Simulation results indicated significant improvement in performance of the proposed enhanced schemes.

✉ Osman Khalid
osman@ciit.net.pk

Rao Naveed Bin Rais
r.rais@ajman.ac.ae

Sajjad A. Madani
madani@comsats.edu.pk

1 COMSATS Institute of Information Technology, Islamabad, Pakistan

2 College of IT, Ajman University, Ajman, United Arab Emirates

Keywords Delay Tolerant Networks (DTNs) · Opportunistic networks · Routing protocols · Benchmarking

1 Introduction

Delay Tolerant Networks (DTNs) are characterized by their intermittent connectivity, frequent partitioning, long message delays, and lack of end-to-end communication paths [1, 2]. Initiated with emphasizing on the interplanetary communication, the focus of DTN research, later on, shifted towards establishing communications in the challenging terrestrial network environments, such as sparsely distributed mobile ad-hoc networks and wireless sensor networks. Figure 1 presents a few application areas of DTNs.

Wireless radio range variations, limited energy resources, sparsity of mobile nodes, and noise, to name a few, are the reasons due to which such networks suffer from frequent disruptions and delays in the process of message transfer [3]. Traditional mobile ad-hoc network (MANET) routing protocols, such as OLSR [4], AODV [5] and DSR [6] are not suitable for network environments with frequent disruptions, as such protocols require existence of an end-to-end communication path among a pair of source and destination [7]. The inherent uncertainty about network conditions makes DTN routing a daunting task. In the absence of end-to-end routing paths, nodes have to rely on opportunistic contacts to exchange messages [8, 9]. Yet, there is no guarantee that a message eventually reaches the destination, as the message may be dropped due to network congestion, lifetime expiry or simply because destination is never encountered, thereby yielding a best effort delivery service. Therefore, it is quite difficult for a DTN routing protocol to achieve a 100 % message delivery, and a lot of research has been focused on increasing the chances of message delivery by taking advantage of a number of network heuristics and nodes characteristics [10–13].

When evaluating the performance of a DTN routing protocol, typical performance metrics include: (a) message delivery ratio, (b) message latency, and (c) message overhead. Delivery ratio is the percentage of messages delivered successfully with respect to the total messages sent. A message's latency is the total time spent between message creation and eventual delivery to the destination. Overhead computation considers the number of extra transmissions for each delivered message. Primary objective of a DTN protocol is to

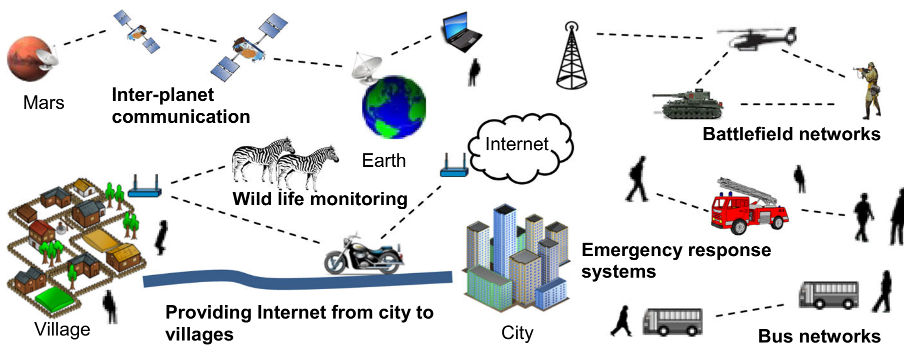

Fig. 1 Examples of DTNs where devices have intermittent connectivity

improve message delivery ratio with minimum latency and overhead, in a network with limited resources. Therefore, DTN protocols adopt various strategies and heuristics [14] to improve the routing performance, and are generally categorized as: (a) deterministic, (b) enforced, and (c) opportunistic. Among the aforementioned categories, opportunistic routing is the most challenging, as it is applied to networks where nodes' mobility information is not known beforehand, and nodes have to rely on random contacts for message transfer. Moreover, to transfer data to potential intermediate relay nodes, opportunistic routing utilizes various heuristics to find the suitability and fitness of a relay, such as (a) relay affiliation with community, (b) contact duration, (c) available bandwidth, (d) available storage at relay nodes, (e) message expiration time, and (f) message priority, etc. [11, 14, 15]. An ever increasing number of protocols addressing "opportunistic" DTN scenarios have been proposed in the past [10, 12, 13, 15–22]. However, still it is not much clear how existing solutions can be applied to a variety of DTN applications and environments, given their requirements and underlying network characteristics. Moreover, to this date, there is no specific study that performs large scale comparison of different routing schemes and gives a solid argument against or in favor of a specific protocol in a particular scenario and underlying network characteristics.

In recent years, numerous comparative studies have been conducted for DTN protocols, under various parameters, simulation platforms, and mobility scenarios [10, 14, 23–25]. All such studies, to some extent, present useful comparisons for DTN protocols. However, most of the evaluations were limited in terms of: (a) number of protocols compared, (b) simulation parameters, and (c) DTN scenarios. We address this very problem in this paper by presenting a thorough empirical comparison of ten carefully chosen "opportunistic" routing protocols. These ten protocols were selected because they seemed to be most appropriate in addressing the routing challenges for diverse DTN environments [26, 27], and cover almost all different approaches that are adopted for designing of DTN routing protocols. More specifically, these protocols cover a broad range of routing scenarios, such as, single-copy forwarding, replication/flooding, probabilistic routing, greedy, and nature inspired content dissemination techniques. The protocols are evaluated with Opportunistic Network Environment simulator (ONE) [28] by using both real-world traces [29] and synthetic mobility scenarios. Moreover, evaluations are done for performance metrics, such as (a) delivery ratio, (b) latency, and (c) overhead, so as to fully understand the strengths and weaknesses of these routing techniques. To the best of our knowledge, this is the first study that presents an extensive benchmarking of DTN protocols under diverse network conditions and parameters.

In addition to the aforementioned benchmarking, we go one step further by proposing three DTN routing protocols, namely, Enhanced Epidemic Scheme (EES), Adaptive Multi-Copy Spray (AMS), and Adaptive Source Token Multi-Copy Spray (ASTMS), by improving the models of the existing routing schemes. The presented techniques efficiently utilize the available network information to control message replication frequency, by autonomously adapting to the varying network conditions, and find optimal spatiotemporal routes for messages. The controlled distribution of message copies helps the efficient utilization of network resources, resulting in the improved delivery ratio and overhead performance.

In summary, our contributions in this paper are as follows:

1. We present basic building blocks of DTN routing and discuss various critical components that must be considered in designing of an efficient DTN protocol.

2. We perform empirical benchmarking of ten carefully-selected DTN routing schemes to evaluate their performance under similar platform and scenarios. This provides a good and useful comparative study of routing protocols in terms of their strengths and weaknesses.
3. The routing protocols are evaluated by varying different parameters, such as: (a) number of nodes, (b) message creation rate, (c) buffer size, (d) message size, and (e) bandwidth. The evaluation results are generated by using both the synthetic mobility models and the real mobility traces.
4. We propose three new DTN routing protocols by enhancing the models of the existing schemes. By introducing adaptability features in the proposed schemes, the protocols are able to control message replication frequency by autonomously adapting to the time varying network conditions. Our results indicate significant improvement in performance of the proposed schemes in terms of delivery ratio and overhead.

The rest of the paper is organized as follows. Section 2 demonstrates DTN routing problem and discusses the important factors that must be carefully considered in modeling of a DTN protocol. In Sect. 3, we present a comparative analysis of the ten selected DTN routing protocols with an emphasis on the strengths and weaknesses of each the protocol. The simulation and benchmarking of the protocols under real-world traces and synthetic mobility is performed in Sect. 4. In Sect. 5, we present the models of the three new routing schemes and analyze their performance in comparison with existing schemes. Section 6 concludes the paper.

2 DTN Routing Insights

The routing problem in DTNs can be expressed as: "Which messages to transfer during an opportunistic contact, such that, they contribute to overall improvement of network performance parameters, such as communication overhead, delivery ratio, and delay?" It is quite challenging to find a precise answer to this question as the routing performance is affected by many factors, such as: (a) message size, (b) message rate, (c) message life-time, (d) buffer size, (e) bandwidth, (f) transmission range, (g) interference, (h) node speed, (i) node energy, (j) mobility pattern, (k) node's sleep intervals, and (l) network size. The numerous combinations and values of the aforementioned factors constitute the different DTN scenarios. The applicability of a DTN protocol for various scenarios depends on the number of aforementioned factors considered while designing a protocol. In the following subsections, we define the basic building blocks of DTN routing and discuss various critical factors that affect the performance of a DTN protocol.

2.1 Forwarding Versus Replication

Message transfer in DTN routing is achieved through either *forwarding* or *replication* [30]. When a message is *forwarded* to a neighbor, the sending node deletes the local copy of message from the buffer. This way, only a single copy of message stays in the network [30]. Alternatively, when the message is *replicated,* both the sending and receiving nodes carry a separate copy of the same message. Forwarding yields minimum overhead and consumes less buffer space as there are lesser number of messages in the network. However, the decrease in message replicas also decreases the probability that a message will be delivered to the destination. Therefore, forwarding is mostly applicable for *deterministic* DTN environments, where nodes' mobility patterns are known beforehand or can be precisely predicted. Message replication frequency may be *controlled* [16] or *uncontrolled* [31]. The controlled replication scheme replicates a message only when a certain condition is satisfied. For example, the neighbor node is more likely to encounter with the destination node as compared to the node carrying the message. The uncontrolled replication is a flooding based technique in which maximum copies of the same message are floated unconditionally in the network. Increase in message copies also increases probability of a message reaching the destination, but at the cost of higher overhead and buffer consumption. Therefore, increased replication rate may also increase the message drop due to buffer overflows, which may reduce the overall delivery ratio.

2.2 Metadata Exchange

The routing protocols differ in the way they utilize the amount of information or *metadata* to perform message transfer decisions [14]. In many protocols, nodes maintain a record of their contacts with other nodes in the form of a list $L\{i, t_c, t_d, b\}$ having parameters: (a) node identification i, (b) contact time t_c, contact duration t_d, and (c) bytes transferred b, respectively [10, 17]. Based on the metadata, a node computes optimal routes for each message. The *uncontrolled replication-based* routing schemes do not utilize metadata [30, 31] as compared to the *controlled replication-based* routing schemes [10, 17], which utilize metadata to improve performance at the cost of increased complexity and computational requirements. When two nodes make a contact, the nodes share and update their respective metadata. This enables the information symmetry throughout the network.

2.3 Buffer Management

Routing schemes also differ in the way they employ buffer management policies. The buffer management typically includes: (a) message queue sorting and (b) message deletion. Message queue sorting is performed by assigning priorities to the messages, and a message of least priority may be deleted. One factor that influences the buffer management is available *contact time window*, which is the time duration between initialization and termination of connection between any two nodes. Suppose, a node i is at location L_i and has a transmission range R_i. The node $i$needs to transfer messages of size X bytes to the neighbor node$j$which is at the location L_j and have the transmission range R_j. Let the bandwidth available for this communication be Y bytes/s. Then, the total time required for the transfer of X bytes messages equals $T = X/Y$. When the two nodes communicate, they are in each other's effective range, which is mathematically represented as:

$$|L_i - L_j| \leq \min(R_i, R_j). \quad (1)$$

In the above equation, the parametersL_i and L_j represents the locations of nodes i and j, and R_i and R_j indicates the communication radius of the nodes i and j, respectively. Let $\overrightarrow{v_i}$ and $\overrightarrow{v_j}$ be velocities of both nodes, respectively. The contact duration window between the two nodes can be calculated as follows [24]:

$$T_{cw} = \frac{2 \times min(R_i, R_j) \times \cos\theta}{|\overrightarrow{v_i} - \overrightarrow{v_j}|}, \quad (2)$$

where θ is the angle between the relative velocity $\vec{V} = |\overrightarrow{v_i} - \overrightarrow{v_j}|$ and the straight line (distance vector ℓ) between the two nodes, as shown in Fig. 2.

Let m and n be the number of messages that both the nodes i and j need to transfer, respectively, then there must be sufficient time available for transfers, specified by:

$$\left(\sum_{k=1}^{m}(T_k)_i + \sum_{k=1}^{n}(T_k)_j\right) < T_{cw} - \beta. \tag{3}$$

In above equation, the parameter $(T_k)_i$ represents transfer time required by kth message of node i, whereas $(T_k)_j$ is the transfer time required by kth message of node j. The parameter β is scaling factor that depends on time spent in metadata exchange (control signaling) and link delays. In general, Eq. (3) produces the total message exchange time required by both the nodes in contact. An important question arises here, and can be stated as, "*Given a limited contact duration, which messages should be given priority over others?*" In [24], the messages are prioritized on the basis of their size, such that the messages that can fit in the real-time contact window are transferred first, whereas in [17] messages with low hop counts are given priority for quicker dissemination in the network. Lindgren et al. [16] gives priority to those messages whose destinations are most encountered by the neighboring relay node, whereas authors in [10] assign priorities to messages whose *utility* contributes to improvement of routing performance. In the next section, we briefly define the ten selected routing protocols and discuss the merits and demerits of each.

3 DTN Routing Protocols

Forevaluation, we selected the following ten routing protocols: (a) *Direct Transmission* [18], (b) *First Contact* [30], (c) *Epidemic* [31], (d) *Wave* [32], (e) *Life* [32], (f) *Spray and Wait* [12], (g) *Spray and Focus* [13], (h) *PRoPHET* [16], (i) *MaxProp* [17], and (j) *Rapid* [10]. The aforementioned protocols cover a broad range of scenarios and DTN applications. For instance, the *MaxProp* [17] and *Rapid* [10] protocols were designed specifically to target a bus-based DTN scenario, where nodes as buses follow mobility at fixed

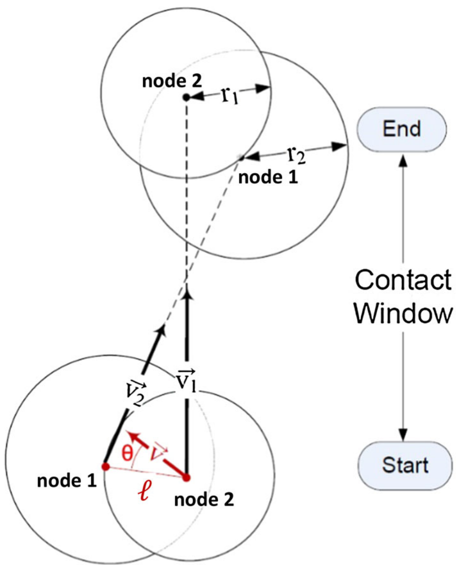

Fig. 2 Calculation of contact window [24]

schedules. The protocols, such as *Epidemic* [31], *Wave* [32], *Life* [32], *Spray and Wait* [12], and *Spray and Focus* [13] in general address the scenarios where nodes mostly follow random mobility patterns. Alternatively, the *Direct Transmission* [18], *First Contact* [30], and *PRoPHET* [16] protocols are designed to target the scenarios where the meeting schedules of nodes can be predicted. Because of diverse properties, the selected set of protocols provides a thorough investigation of the various insights about DTN routing schemes. We begin with the simplest routing protocol of the abovementioned list, and gradually proceed towards more complex models. The discussion gives us an insight into how an increase in complexity of protocols improves the performance for one metric at the cost of another.

3.1 Direct Transmission

Spyropoulos et al. [18] presented the simplest of DTN routing protocols. As the name suggests, the source node directly transmits the message to the destination node without any intermediate relaying. Therefore, at any time, only a single copy of message is present in the network. The benefit of *Direct Transmission* scheme is that it causes minimum overhead due to reduction in messages' copies. However, the source nodes may have to wait for longer periods or indefinitely to make direct contacts with the messages' destinations. Therefore, *Direct Transmission* may experience maximum latency as well as the minimum message delivery ratio. The *Direct Transmission* scheme may be useful for DTN scenarios where nodes' mobility pattern can be precisely predicted. One such application can be bus networks where buses followed fixed schedules for various routes.

3.2 First Contact

The *First Contact* scheme [30] allows a node to forward a message towards a randomly selected neighbor. After the message is successfully forwarded, the sender node deletes local copy of message. The message may be relayed through several hops before reaching the destination. This makes the *First Contact* a single-copy multi-hop forwarding scheme. A message maintains a list of hops traversed to avoid visiting the same hops again. The *First Contact* protocol does not exhibit optimal performance in terms of message delivery ratio. This is because the randomly selected neighbor may not appear to be a best candidate to forward message towards destination. Therefore, delivery ratio of *First Contact* does not significantly improve, when compared to *Direct Transmission.* Moreover, the *First Contact* scheme experiences higher overhead due to extra transmissions per message.

3.3 Epidemic

The *Epidemic* protocol [31] is an *uncontrolled replication-based* routing scheme that functions analogous to the way a disease spreads. A node having a message copy is said to be infected. When this node makes contact with another node, the infection is transferred to the other node such that at the end of communication both nodes are having same the infection (similar copies of a message). The *Epidemic* scheme spreads greater number of message copies in the network to improve message delivery probability. However, increasing message copies may cause greater overhead, higher utilization of buffers, and increased network congestion. Therefore, the *Epidemic* protocols are ideal for scenarios that have higher bandwidth and greater buffer storage available. The scenarios where nodes have limited buffer capacity, the *Epidemic* protocol may result in packet drop due to buffer overflows.

3.4 Life

The *Life* protocol [32] is based on the theory of *Conway's Game of Life* [33]. This theory simulates life of a *cell* depending on the number of live cells in the neighborhood. The *cell* represents a node in the network having the message replica. The *Game of Life* theory is applied in the *Life* protocol to control the message flooding. A node may replicate, delete, or keep a message copy depending on the minimum and maximum number of neighbors that have the copy of same message. As the protocol controls message replication by frequently deleting the extra message replicas in the neighborhood, the buffer utilization is reduced. This strategy also improves the message delivery ratio as fewer messages are dropped due to buffer overflows. The *Life* protocol is designed keeping in view the mobility pattern followed by people on sea beaches.

3.5 Wave

The *Wave* protocol [32] utilizes *tracking lists* to track messages that were recently relayed by a node. The idea is to prevent a node from receiving the same message replica again in short time duration. When a node receives a message, the message entry (such as, message identifier and receiving time) is maintained in the *tracking list.* During message exchange, the sender node transfers the message to the neighbor, but does not remove the message entry from the *tracking list.* This prevents the node from receiving the same message replica within a short time span. Such reductions in message replications minimize the overall overhead. However, decrease in the message replicas also decreases the message delivery probability of the *Wave* protocol, as compared to the *Life* protocol.

3.6 Spray and Wait

The *Spray and Wait* protocol *(Binary version)* [12] sets a limit on a message's maximum number of replicas in the network to reduce flooding. Every new message is assigned an L number of *replication tokens*, which represents the maximum number of replicas a message can have at any time in the network. During the *Spray* phase, a node replicates the message by setting $L/2$ tokens on the message copy that it has in buffer, and assigns $L/2$ to the message copy sent to the neighbor. A node having a message with token value equal to one enters the *Wait phase* for only that particular message. In the *Wait* phase, the node no longer relays the message, and waits to make a contact only with the message's destination. The *Spray and Wait* protocol experiences minimum overhead due to decrease in per message copies. It also exhibits minimum latency and indicates higher message delivery ratio. This is because, the controlled replication reduces message drop that results in quicker dissemination of messages. Moreover, the reduction in message copies prevents message drops due to shortage of buffer space.

3.7 Spray and Focus

The *Spray and Focus* protocol [13] utilizes the concept of *utility function*. The utility function quantifies the quality of a node to become a relay for a message. A node with greater and recent interactions with a message's destination is considered to have higher *utility* value for that particular message, and will be more suitable candidate to carry the message. The *Wait phase* in the *Spray and Wait* protocol is replaced by *Focus* phase in the *Spray and Focus* protocol. In the *Focus phase*, a node forwards the message to the neighbor, if and only if the neighbor has higher *utility* value for that message. As compared to the *Spray and Wait,* the *Spray and Focus* protocol experiences lesser latency as the messages do not have to wait in the buffers for indefinite periods to be transferred to the destinations. However, the increase in message transmissions also increases overhead.

3.8 PRoPHET

The *PRoPHET* protocol [16] calculates the *delivery predictability* for every node in the system. The *delivery predictability* value is quantified by the number of recent interactions of a node with other nodes in the network. Nodes perform the transitive updates of *delivery predictabilities* by sharing routing tables during contacts. A node replicates a message to the neighbor, if and only if the *delivery predictability* of the neighbor is greater than the sender node. This way the *PRoPHET* protocol attempts to reduce the overhead. However, when the network size is large, such as a city-wide DTN network, it may take significant time in building up of *delivery predictabilities*. Therefore, in such cases *PRoPHET* may experience increase in overhead due to higher number of replications. The overhead also results in the increased buffer shortages, which may lead to the reduced message delivery ratio for *PRoPHET*.

3.9 MaxProp

The *MaxProp* protocol [17] implements the message queue management by splitting the queue into two portions. In the first portion, those messages are prioritized that traversed least number of hops. In the second portion, the messages having the least cost paths towards destinations are given priority. The least cost paths are calculated by using a modified version of *Dijkstra's* algorithm. The *MaxProp* protocol prevents a message from repeatedly visiting the same hop by implementing hop-lists within a message. This significantly reduces the message overhead. The use of acknowledgements deletes the redundant message copies from the nodes' buffers, as a consequence, substantially reducing the message drops due to buffer overflows, and improving overall message delivery ratio.

3.10 Rapid

The *Rapid* protocol [10] employs the concept of message *utility*. Message *utility* is calculated on the basis of a node's past interactions with a message's destination and the amount of data exchanged during the interactions [10]. A message's *utility* is higher, if and only if, the message's replication causes improvement in any of the following routing metrics defined by the authors: (a) average delay, (b) worst-case delay, and (c) number of packets delivered before deadline [10]. During a transfer opportunity, the *Rapid* protocol calculates *marginal utility* of all the messages in the routing queue. Marginal utility quantifies the marginal increase in utility of the message, when the said message is replicated. The messages are sorted in the buffer such that the first message to be replicated has the highest *marginal utility*. If network size is large, such as a city-wide network, then the *Rapid* protocol may indicate lower performance. This is because of the time required for building-up of message utilities. Therefore, the large number of replications during such a period may lead to increased overhead.

4 Empirical Setups, Results, and Discussion

This section presents the comparisons of the ten selected protocols. We begin with providing an introduction to the simulator, the mobility scenarios, and the parameters considered for the simulations. Next, comparative results are discussed in detail for the selected performance metrics. Finally, a discussion on the results is provided along with ranking of protocols based on their performance.

4.1 The ONE Simulator

The simulations are performed with ONE simulator, which is a platform independent Java-based simulator, specifically designed for evaluating DTN routing protocols [28].Although, there exists some advanced network simulators, such as OMNET++ [34] and NS-3 [35] that offer generic platforms for packet-based communications. However, in wireless environments the aforementioned simulators provide only a limited and specific support for MANETs that assume end-to-end connectivity among nodes [36]. On the other hand, DTNs generally do not have end-to-end connectivity due to frequent topology changes, disruptions, and network partitions mainly caused by the node movement. Due to fairly limited support for DTN routing protocols, we did not use the above mentioned simulators for the simulations and instead preferred the ONE simulator. The ONE simulator has rich set of features for simulating DTNs and allows users to create scenarios based on various synthetic mobility models as well as real-world traces. Moreover, the ONE simulator offers a framework for implementing routing and application protocols with functionalities available for modeling nodes' movement, inter-node contacts, and routing and message handling. The MAC layer is partially implemented by allowing only two nodes to transfer messages at one time. Instead of fully modeling the lower layers, the ONE simulator makes simplifying assumptions about the data rates and the radio ranges [28], and rather focuses more on the routing aspects of network layer.

4.2 Mobility Scenarios

We utilized the mobility model classes provided by the ONE simulator [28]. Instead of using traditional Random walk or Random way-point mobility models, we utilized more realistic and configurable mobility models provided by the ONE simulator. The simplest built-in model is Random Map-Based Movement in which nodes randomly move on paths defined on the map. The next model is Shortest Path Map-Based Movement (SP-MBM) in which the nodes follow the shortest routes towards some target locations on the map. These target locations are either randomly chosen, or specified from a list of Points of Interest (POI). The POI may represent frequently visited real-world destinations, such as offices, homes, shopping malls, or restaurants. The Route Map-Based Movement (R-MBM) model forces the nodes to travel on pre-determined routes. Such routes are constructed to match bus routes, or train routes. Finally, the Working Day Movement Model (WDM) models human mobility patterns during a working week. We divided the nodes in different groups and assigned different locations and mobility models discussed above. For simulation map, we imported the map of City of Fargo, ND, USA from the *OpenStreetMap API* [37]. Using the GIS tool OPENJUMP [38], the map was post-processed and marked with various locations such as shops, homes, offices, meeting points, universities, and bus stations. In addition to the map based scenario, the protocols are also evaluated on real-world connectivity traces available at an online repository [29]. Figure 3 shows a portion of the selected map in OpenJUMP, whereas Fig. 4 shows the same portion of map opened in the ONE simulator. The simulations under real-traces were also performed in an attempt to get better insight into the suitability of a protocol for human mobility scenarios.

4.3 Simulation Parameters

The simulation parameters are indicated in Table 1(a). The parameter values are selected keeping in view the real-world scenarios where transfer opportunities and resources are often limited. A few of the DTN protocols utilize additional simulation parameters indicated in Table 1(b). The values of such parameters are selected as proposed by the authors in their respective literatures. For each point, a simulation of 20 runs is conducted and averaged.

4.4 Performance Metrics

The protocols are evaluated for three performance metrics: (a) delivery ratio, (b) latency, and (c) overhead. The following subsections illustrate these metrics.

4.4.1 Delivery Ratio

Delivery ratio is the percentage of messages delivered successfully. The unit of delivery ratio is "number of messages". The increased message delivery ratio is the major goal of any DTN routing protocol. Message delivery ratio is calculated as:

$$Delivery\,Ratio = \frac{1}{M}\sum_{k=1}^{M} R_k, \tag{4}$$

where M is total messages created and $R_k = 1$ if message m_k is delivered, otherwise $R_k = 0$.

4.4.2 Latency

Latency is the total time spent between message creation and delivery to the destination. The average latencies of messages contribute to the overall latency measure of protocol. A protocol must minimize latency but without compromising message delivery ratio. The latency average (in seconds) is given by:

$$Latency\,average(in\,seconds) = \frac{1}{N}\sum_{k=1}^{N} Receive\,Time_k - Creation\,Time_k, \tag{5}$$

where N is the total number of messages received, the parameters $Receive\,Time_k$ and $Creation\,Time_k$ represents the receiving time and creation time of message k.

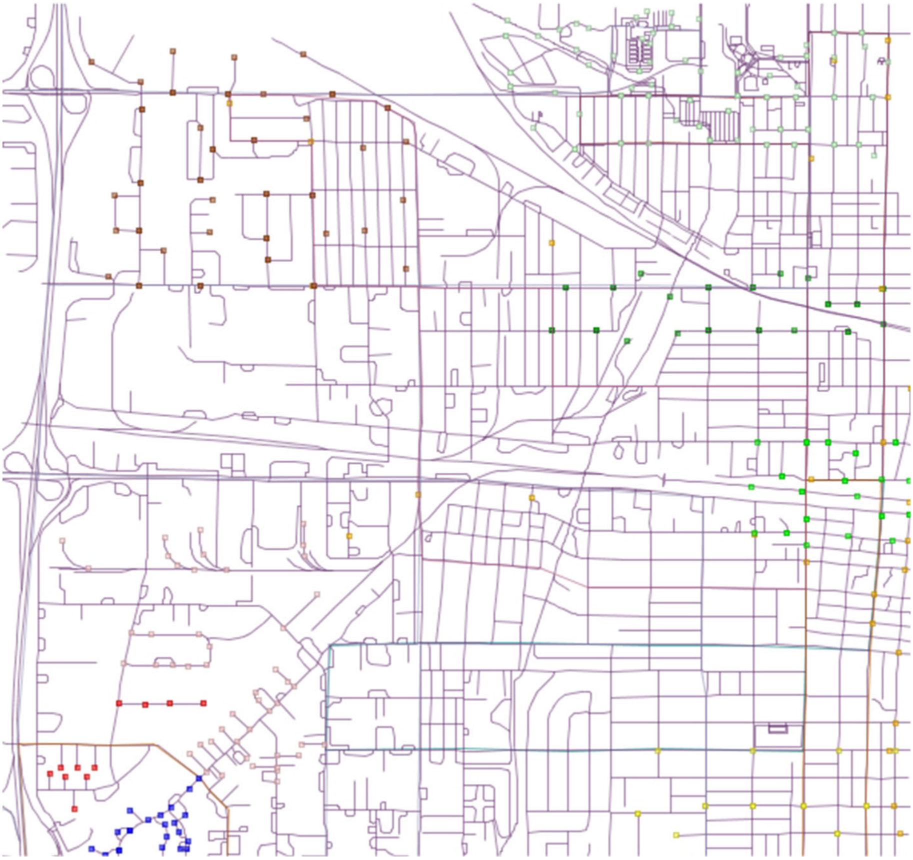

Fig. 3 A portion of map of City of Fargo. The *dots* on the map indicate the specific points of interest (POI)

4.4.3 Overhead

Overhead is the approximate measure of the consumption of bandwidth, energy, and storage by a protocol due to message transmissions. The overhead is calculated as the relative estimate of number of message transmissions:

$$Overhead = \frac{Total\,relayed - Total\,delivered}{Total\,delivered}. \tag{6}$$

In the above equation, *Total relayed* represents the total number of message relaying in the network and *Total delivered* indicates the total number of messages delivered to the destination, where $Total\,delivered \leq Total\,relayed$. The overhead ratio indicates extra transmissions for each delivered message. The overhead just represents a ratio with same metric (number of messages) as numerator and denominator; hence it does not have any units. For instance, in a single copy forwarding, if Total relayed is 5 and Total delivered is also 5, then overhead = (5–5)/5 = 0, which means, the messages were directly delivered to the destinations by the source nodes. On the other hand, if suppose Total relayed = 20 and total delivered = 5, then overhead = (20–5)/5 = 3. This implies that each of the

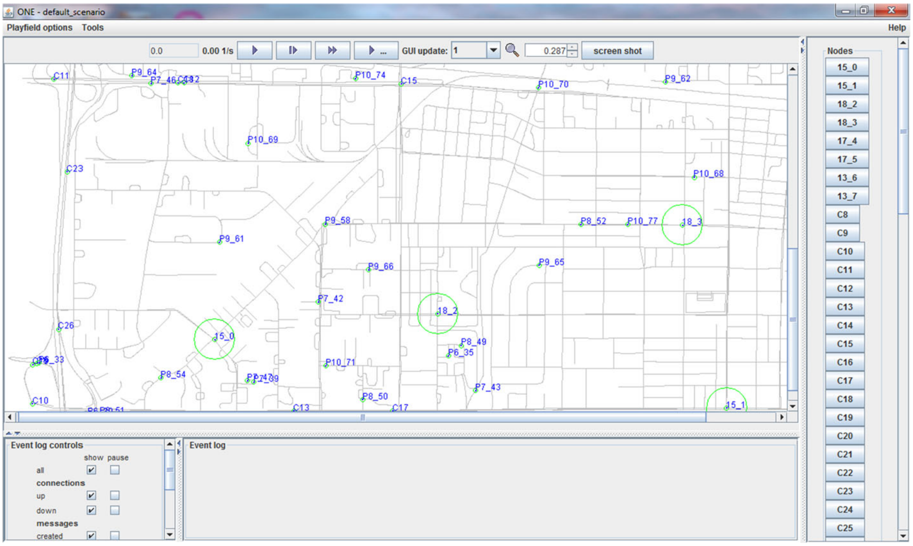

Fig. 4 A portion of map of City of Fargo, as shown in the ONE simulator. The *circle* surrounding nodes represent communication radius

delivered messages (out of 5) had on the average 3 extra transmissions before reaching the final destination node.

4.5 Comparative Analysis

In this subsection, we present the simulative comparisons of the selected DTN protocols by varying the following: (a) buffer capacity, (b) message creation rate, (c) number of nodes, and (d) message size. A few of the protocols exhibited similarities in average performance for the selected routing metrics. Therefore, based on the similarities in the empirical results, we divide the protocols into following groups: *Group1* (*a. MaxProp, b. Spray and Wait, c. Spray and Focus,* and *d. Rapid*), *Group2* (*a. PRoPHET, b. Life, c. Wave,* and *d. Epidemic*), and *Group3* (*a. Direct Transmission* and *b. First Contact*). It is also noteworthy to mention that the simulation results generated in this subsection depend on the selected parameter values for the evaluation. As DTN routing is greatly affected by the network conditions, we may observe different results in scenarios with dissimilar simulation settings.

4.5.1 Impact of Change in the Number of Nodes

Simulations are performed to observe the scalability of each protocol and results are obtained for average values of latency, delivery ratio, and overhead. As reflected in Fig. 5a, the *MaxProp* protocol shows the best performance among the other protocols of *Group1*. The *MaxProp* protocol's message acknowledgement mechanism removes the redundant packets from the buffers to allow enough space for new packets. This prevents message drop due to lack of buffer space. Moreover, the delivery ratio of *MaxProp* remains approximately constant. This is because the increase in network load due to the entry of new node is balanced with the establishment of new least-cost paths in the network.

Table 1 Simulation parameters

Parameter	Value(s)
(a) Commonly use simulation parameters	
World size	4250 × 3900 m
Simulation time per run	12 h
Radio Interface	Speed: 250 kbps (2 Mbps) Range: 20 m for cars, 100 m for busses
Message	Size: 500 KB–1 MB Interval: 1 per min TTL: 500 min
Buffer size	10–100 MB (maximum buffer size that a node is willing to allocate for message distribution)
Nodes	Buses: 8 Cars: 20 Pedestrians: 72 Total: 100 (any node can be source as well as destination)
Node average Speed	Buses: 10–35 km/h Cars: 10–50 km/h Pedestrians: 0–5 km/h
Mobility	City environment, real connectivity traces

Protocol	Ref	Parameters
(b) Protocol specific parameters		
Spray and wait	[24]	$L = 10–15\ \%$
Spray and focus	[26]	$U_{th} = 10–90$ $L = 5–10\ \%$ of nodes
PRoPHET	[16]	$P_{init} = 0.75, \beta = 0.25$ $\gamma = 0.98, T_u = 30$

Furthermore, *MaxProp* gives priority to the messages with low hop counts which allows the quicker propagation of newer messages, reducing the latency as depicted in Fig. 5b. The overhead for *MaxProp* as indicated in Fig. 5c is comparatively lower as the message replication is reduced with the increased network connectivity.

The message delivery ratio of the *Spray and Wait* and *Spray and Focus* protocols is roughly the same and constant, as both the protocols set a limit on the maximum number of message copies in the network to control message flooding. The latency of the *Spray and Focus* protocol is less than the *Spray and Wait* protocol as the former employs the utility based forwarding, whereas the latter waits with the last copy of message to make contact with the message's destination. However, the overhead of the *Spray and Focus* protocol is higher due to greater message transmissions as compared to the *Spray and Wait* protocol. Performance of the *Rapid* protocol decreases as the number of nodes increase. The *Rapid* protocol gives priority to messages with greater utilities. If the network area (and size) is large, as we considered in this simulation, nodes will take longer to generate accurate utility values for messages. This will cause uncontrolled message replication initially, which will also result in greater buffer usage. Therefore, the increase in message drop due lack of buffer space will result in lower delivery ratio for the *Rapid* protocol.

It can be further observed from Fig. 5 that the routing protocols of *Group2* experience low performance for almost all the three routing metrics. This is due to the flooding

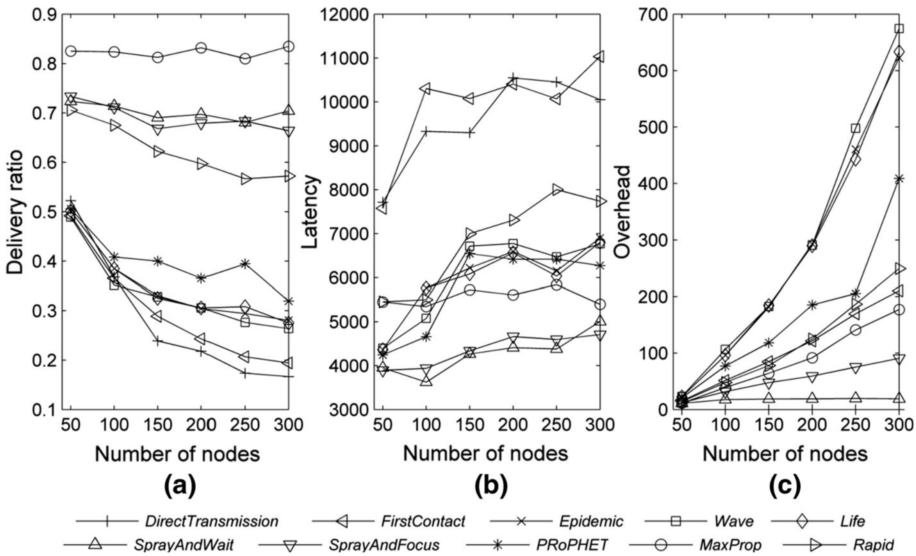

Fig. 5 Effect of increasing number of nodes on a protocol's **a** delivery ratio, **b** latency, and **c** overhead

strategies utilized by the *Group2* protocols. The increased flooding results in the shortage of buffer spaces in the network. Therefore, nodes drop messages more frequently to accommodate new messages. The *PRoPHET* protocol slightly performs better due to the controlled message replications. The delivery ratio performance of *Life* is better than *Wave* and *Epidemic* as the *Life* protocol deletes the redundant message copies, which are proportionate to the number of replicas in the neighborhood. The *Group3* protocols show the worst performance. Among the*Group3* protocols, the *First Contact* protocol forwards the single copy of message on a randomly selected connection. This may result in message loss if the selected neighbor fails to transfer message towards destination node. The same is the case with the *Direct Transmission* protocol that performs no forwarding unless the neighbor node is the message destination. Therefore, *Direct Transmission* exhibits the lowest message delivery ratio. From Fig. 5c, it is also evident that all protocols follow the similar behavior in terms of overhead, as the increase in number of nodes also increases the number of transmissions per message. However, this is not the case for *Direct Transmission* which has zero overhead because the *Direct Transmission* protocol performs no replications. The *Spray and Wait* protocol indicates best overhead performance due to the limited number of message transmissions.

4.5.2 Impact of Change in Message Creation Rate

The protocols are evaluated by increasing the network traffic. The message creation rate is varied from one message after ten seconds to one message after 180 s (3 min). As reflected in Fig. 6, the *Group1* outperformed the other groups in all the routing metrics. This is because, the *Group1* protocols perform least flooding. When the message creation rate is higher, then the nodes soon exhaust their buffer spaces resulting into the increased message drop rate. The decrease in the message creation rate favors each protocol in terms of message delivery ratio and latency. Among the *Group2* protocols, the *Wave* protocol exhibits better performance due to the utilization of the tracking lists to control flooding

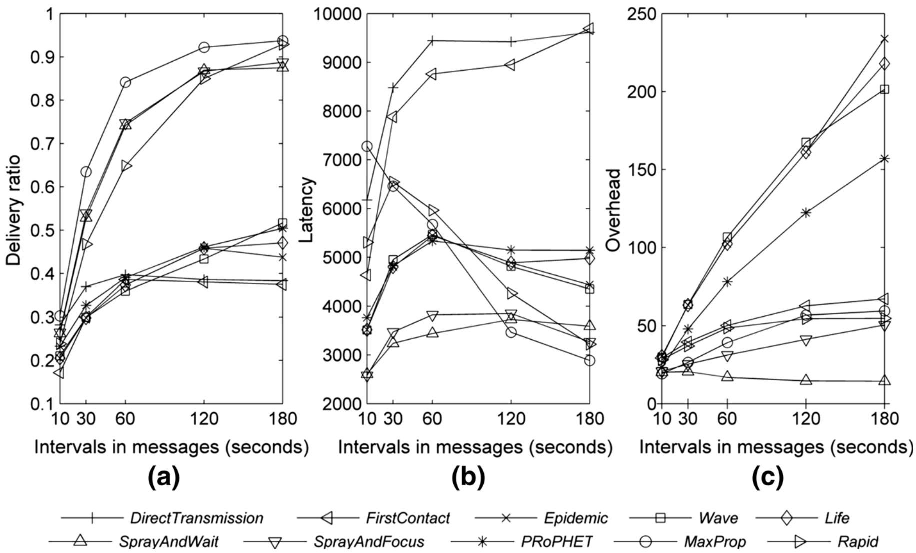

Fig. 6 Effect of decreasing the message creation rate on a protocol's **a** delivery ratio, **b** latency, and **c** overhead

[32]. The *Group3* protocols experience degradation in message delivery ratio and exhibit increased average packet delay. This is due to the fact that single copy routing strategy implemented by these protocols is less efficient for delivering messages. The overhead (Fig. 6c) appears to be increasing for the protocols with the decrease in message creation rate. Initially, the message creation rate is higher due to which large numbers of messages are dropped without being relayed, and as a result the overhead is smaller. As the message rate is decreased, there is also an increase in message relaying, which results in an increased overhead in accordance to (6). The *Spray and Wait* protocol indicates best performance in terms of overhead due to the fixed number of transmissions per message.

4.5.3 Impact of Change in Buffer Capacity

In this experiment, the simulations are performed to understand how protocols behave by varying buffer sizes. As Fig. 7a indicates, the delivery ratios of *Group1*, *Group2,* and *Group3* protocols initially increase and then achieve a nearly constant value as the buffer size is increased. The reason for such behavior is that initially there is an increased message drop rate due to low buffers, resulting in the low values of delivery ratio. As the buffer size is increased, delivery ratio also improves due to decrease in message drops. However, as reflected from Fig. 7a, raising buffer capacity beyond a certain level does not further contribute to the delivery ratio, as the other factors, such as mobility patterns, buffer management policies, and contact durations, also affect the communications in DTNs. Increasing buffer space favors *Rapid* and *MaxProp* in terms of reduced latency (Fig. 7b), as both of these protocols require memory space to store metadata information, which is utilized to calculate shortest paths to the destinations. The *Epidemic* protocol does not indicate significant improvement in delivery ratio and latency, despite increase in buffer capacity. The main reasons for such behavior are: (a) the buffer management policy of *Epidemic*, which is *First-In-First-Out* (FIFO) and (b) limited contact durations among

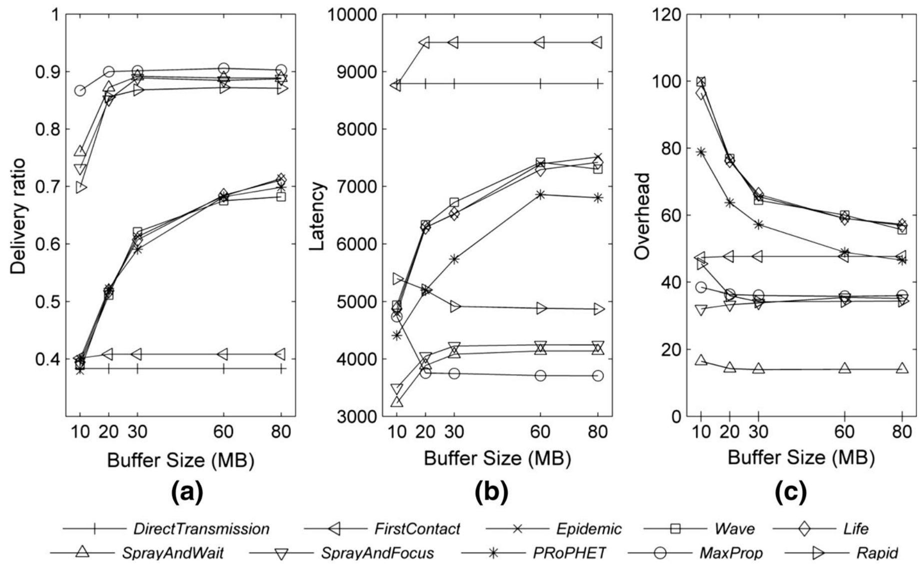

Fig. 7 Effect of increasing buffer size on **a** delivery ratio, **b** latency, and **c** overhead

mobile nodes. As long as the contact duration between two nodes is greater than the expected number of messages in transit, FIFO is a reasonable policy. However, if the available contact durations are limited relative to the number of messages, then only a subset of almost similar messages will be transferred on each contact. This will cause the messages at the end of queue to be flushed from buffer due to their life time expiry, resulting in overall decrease in delivery ratio.

The latency of the *PRoPHET* protocol is lesser than other group members as the *PRoPHET* protocol utilizes contacts' history information to relay messages to more appropriate neighbors that have higher delivery predictability for the messages' destinations. Increasing buffer space has interesting impact on overhead ratio (Fig. 7c). This behavior is in accordance with the (6).When the buffer size is small, fewer messages are delivered as more messages are dropped due to buffer overflows. Therefore, (6) will produce a higher overhead value. With increase in buffer capacity, the number of messages delivered also increases, which leads to a decrease in overhead. For *Group3* protocols, the improvement in buffer capacity does not have any significant effect on the delivery ratio. Therefore, the overhead ratio of these protocols remains unchanged.

4.5.4 Impact of Change in Message Size

Message size is an important factor for measuring routing performance in DTN environments. As reflected in Fig. 8, the increase in message size is almost inversely proportional to the message delivery ratio for all the groups. This is mainly due to the following reasons: (a) increase in message size decreases the number of messages exchanged during the limited contact opportunities of mobile nodes, and consequently, most of the messages are dropped as their life-time expires before reaching the destination, and (b) large messages occupy more buffer space resulting in the buffer shortage for new messages. Therefore, messages are frequently dropped to make room for new messages. The aforementioned

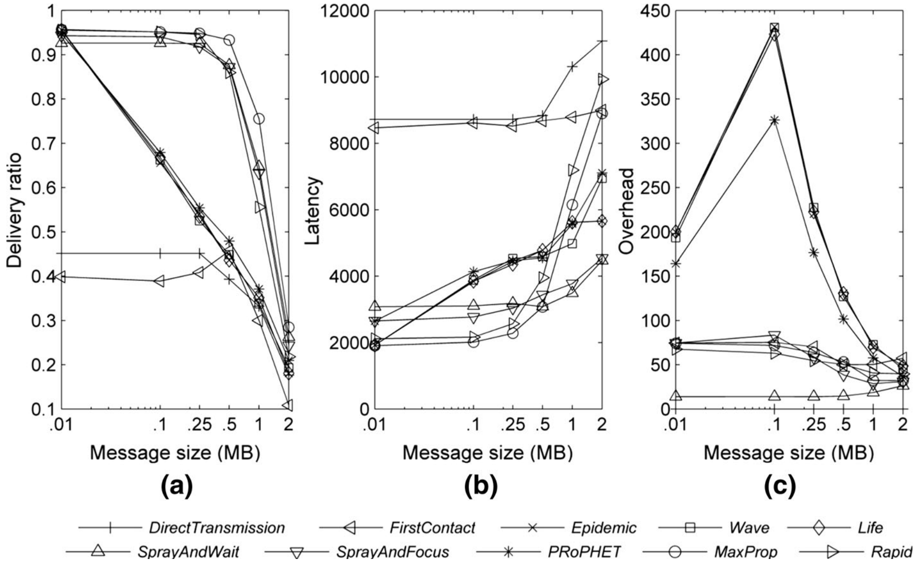

Fig. 8 Effect of changing message size on a protocol's **a** delivery ratio, **b** latency, and **c** overhead

point (a) also causes the latency to increase as depicted in Fig. 8b. The *Group1* protocols experience gradual degradation of routing performance for delivery ratio and latency. This is due to the fact that the *Group1* protocols adopt various measures to control flooding, resulting in lesser message drops due to buffers' overflow. In contrast, the *Group2* protocols exhibit a sudden decrease in performance due to their inherent flooding nature. Unlike all the other protocols, the delivery ratio of *Group3′sFirst Contact* protocol (Fig. 8a) appears to be increasing for message sizes between 100 and 500 KB. The reason for such an increase is that the *First Contact* protocol deletes the local copy of message after the message is forwarded to the neighbor. Therefore, space is conserved to accommodate more messages, which leads to the decrease in message drop rate. Alternatively, the *Direct Transmission* protocol experiences higher message drops as the nodes' less frequently encounter with the actual destinations of messages. Therefore, the messages are frequently dropped due to life-time expiry.

As depicted in Fig. 8c, the overhead initially increases, and then starts to decline. Initially, message size is smaller and nodes are able to perform more message transmissions, which results in an increase in the overhead. However, the overhead starts decreasing after the message size crosses approximately 100 KB. This is due to the decrease in message transfers, as the nodes cannot exchange enough messages in their limited contact opportunities, when the nodes are mobile and messages are of larger sizes.

4.5.5 Performance Comparisons with Real Trace Data

In this subsection, the simulations are performed with real-world connectivity traces. The trace datasets were collected under *Haggle project* during INFOCOM 2006 conference and are available at an online repository [29]. The parameters used for simulation are: bandwidth: 250 kbps (2 Mbps), message size: 500 KB–1 MB, packet-lifetime: 500 min, number of nodes: 98, and buffer size: 10–100 MB.

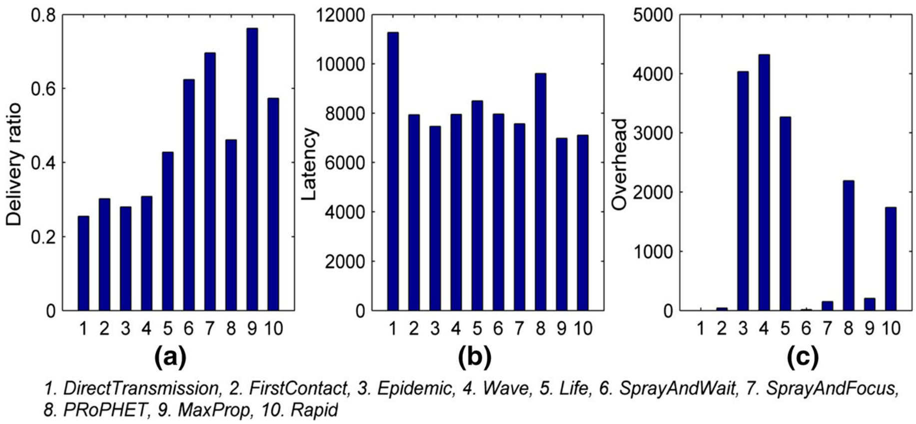

Fig. 9 Performance comparison with real connectivity traces for **a** delivery ratio, **b** latency, and **c** overhead

It can be observed from Fig. 9 that *Group1* outperforms the remaining groups in terms of message delivery ratio. The *MaxProp* protocol takes maximum advantage of repetitive mobility patterns of the nodes in calculating least cost paths. As opposed to the synthetic mobility scenario, the delivery ratio of the *Spray and Focus* protocol is greater than the *Spray and Wait* protocol for simulations performed under real-traces. A reason for such behavior is that the *Spray and Focus* protocol keeps on forwarding the packets to the high utility nodes. In this way, the packets are given chance to ultimately reach their destinations in scenarios where transfer opportunities are rare (due to greater inter contact times of nodes). The *Rapid* protocol does not present any significant change in performance for real-trace dataset. The *Rapid* protocol is based on link state routing and it might be challenging to precisely predict the optimal paths between source and destination nodes. Especially, in scenarios when the nodes do not encounter frequently, such as when participants are sitting in different conference sessions for longer durations. Therefore, the *Rapid* protocol experiences greater packet drops due to life time expiry. The latency of the *Spray and Focus* protocol is lesser than the *Spray and Wait* protocol. Moreover, the *MaxProp* protocol exhibits minimum latency among the*Group1* protocols as the *MaxProp* protocol more efficiently exploits the repeated mobility patterns in the calculation of least-cost paths.

Among the *Group2* protocols, the *PRoPHET* protocol outperforms the other schemes in terms of delivery ratio. This is primarily due to: (a) presence of stationary nodes that are most visited and aid the routing as their delivery predictability improves with time and (b) nodes with repetitive mobility act as message relays among various participant groups. However, as it may take longer to encounter a node of higher delivery predictability (such as a stationary node). Therefore, the *PRoPHET* protocol has highest latency among the*Group2* protocols. The *Life* protocol performs better than the *Wave* and *Epidemic* protocols in the conference scenario with low mobility. This is due to the very nature of the *Life* protocol that quickly deletes the message copies and re-gain messages, which helps low buffer occupancy and message circulation. However, good performance of the *Life* protocol comes at the expense of increased overhead (Fig. 9b). Not surprisingly, the *First Contact* protocol has better message delivery performance than the *Direct Transmission* protocol that has the highest latency among all protocols. The overhead of the *Wave* protocol is greatest among the *Group2* protocols. This is because the *Wave* protocol does not accept a message, if the message's entry is already present in the tracking list [32]. This may cause the nodes to occasionally miss the opportunities of becoming message relays. Consequently, there will be slight increase in message drop rate due to life time expiry as messages have to wait longer before being relayed. Therefore, the decrease in "total delivered" parameter of (6), will result in the increase of overhead ratio for the *Wave* protocol. The *Rapid* protocol transfers more messages before it can effectively build the metadata, resulting in increased overhead among the *Group1* protocols. The overhead of *Group3* protocols is least as they are single copy message relaying schemes, which perform minimum message transmissions.

4.5.6 Summary of Results and Discussions

Based on the simulation results, it can be concluded that performance of *Group1* protocols remained consistent in all the simulation scenarios, where the *MaxProp* protocol outperformed the rest of the protocols. The primary reason for good performance of the *Group1* protocols is the way these protocols control message flooding. Flooding causes buffer overflows that result in increased message drop rate. Moreover, the *MaxProp* and *Rapid* protocols utilize additional information in the form of metadata to find the shortest paths among sources and destinations, and to delete the messages that have reached the destinations. Alternatively, the *Spray and Wait,* and *Spray and Focus* protocols control flooding by setting a limit on maximum number of message copies in the network. Among the *Group2* protocols, the *PRoPHET* protocol manages to deliver more packets with lesser overhead as compared to the *Life*, *Wave*, and *Epidemic* protocols. This is because the *PRoPHET* protocol controls flooding by replicating a message to the peer, if and only if delivery predictability of peer is higher than the sender. Alternatively, the *Life*, *Wave*, and *Epidemic* protocols rely on maximized flooding that eventually results in higher message drop rate. The performance of *Group3* protocols is lowest as they implemented single copy forwarding, which also minimizes the message delivery chances when nodes' have longer inter-contact times. Therefore, based on the results, we can make following observations:

- Effect of number of nodes: Increase in number of nodes facilitates the message forwarding in DTN environments as more nodes are available to serve as relays and to carry message between source and destination. However, such facilitation is at the expense of increased overhead caused by the new nodes. Therefore, the new nodes joining the network do not play a vital role in the improvement of message delivery ratio, as reflected in Fig. 5.
- Effect of message creation rate: Fig. 6 depicts that decrease in the message creation rate improves the delivery ratio due to reduction in message drop rate. However, when the message rate is decreased enough (e.g., one message after every 120 s), the delivery ratio for most of the protocols attains a constant value. This constant behavior is due to the other factors involved in message drop, such as: (a) message queue sorting policies and (b) mobility patterns of nodes. Reduction in message creation rate will reduce latency if and only if the nodes' contact frequency is optimal enough not to cause the messages to wait longer into buffers.
- Effect of increasing the buffer space: Increase in buffer capacity of nodes improves message delivery ratio for all the protocols, especially for the*Group2* protocols that utilize flooding based techniques. However, as indicated in Fig. 7, even the large sized buffers may not produce 100 % message delivery ratio. This is due to the fact that

messages may be dropped with the expiry of messages' life time if there are long gaps in nodes' meetings. Another reason is the queue management policies, which may repeatedly favor only a subset of messages during message transfer. For example, the *PRoPHET* protocol gives priority to messages whose delivery predictability is higher and the *Epidemic* protocol sorts the queue on FIFO basis. This may cause other messages to wait long in buffers and subsequently be dropped due to life-time expiry. Therefore, in current scenario, Fig. 7 presents an estimate of upper bound in delivery ratio when the buffer capacity is assumed to be infinite.

- Effect of change in message size: When message size is smaller and buffer space is enough to accommodate new messages, the nodes are able to exchange more messages during a limited contact interval, resulting in an increased delivery ratio. As message size is increased, nodes have to wait longer for the occurrence of contacts of greater durations. This results in message drop due to life-time expiry and increases latency as indicated in Fig. 8.
- Effect of change in bandwidth: We also performed simulations by varying the bandwidth. Results indicated that if the bandwidth is higher, then there is an increase in message drop rate due to buffer overflows, but decrease in message drop due to message life time expiry. Alternatively, if bandwidth is less, then fewer messages are exchanged among nodes. Therefore, there is less message drop due to buffer overflow, but higher message drop due to message life time expiry. Therefore, in both the cases we achieved roughly the same results for message delivery ratio.

Based on our findings, we are now in a position to rank protocols depending upon their performance consistency under the parameters selected for simulation. Table 2 presents the numerical ranking of the protocols for message delivery ratio. The overview of results and our recommendations on the usage of protocol for a particular scenario is summarized in Table 3.

Table 2 Protocols ranking based on message delivery ratio

Protocol	Message delivery ratio				Overall score	Rankings
	Network size	Buffer impact	Message rate	Message size		
Direct transmission	10	10	10	9	39	10
First contact	9	9	10	10	38	9
Epidemic	7	6	8	8	29	8
Wave	8	8	6	6	28	7
Life	6	5	7	7	25	6
Spray and wait	2	2	3	2	9	2
Spray and focus	3	3	2	3	11	3
PRoPHET	5	7	5	5	22	5
MaxProp	1	1	1	1	4	1
Rapid	4	4	4	4	16	4

The protocol with lowest score is given top ranking

Table 3 Overview of results

Protocol	Delivery ratio	Buffer utilization	Energy efficiency	Complexity	Suggested scenario
Direct Transmission	Very low	Very low	Low-medium	Very low	Deterministic mobility scenarios
FirstContact	Very low	Very low	Low	Low	Random mobility/emergency scenario
Epidemic	Low	Very high	Very low	Low	Random mobility/emergency scenario
Wave	Medium	Medium	Medium	Medium	Partial mobility/limited area scenario
Life	Medium	Medium	Medium	Low-medium	Partial mobility/limited area scenario
Spray and wait	High	Low	High	Low-medium	Heterogeneous scenarios
Spray and focus	High	Low-medium	High	Medium–high	Heterogeneous scenarios
PRoPHET	Medium	Medium–high	Medium	High	Community based/ predictable mobility
MaxProp	Very high	High	Very high	Very high	Heterogeneous scenarios
Rapid	Medium–high	High	Medium–high	Very high	Limited vehicular mobility scenarios

5 Proposed Routing Models

In this section, we propose three routing models based on enhancements in the replication strategies of the following three protocols: (a) *Epidemic* [31], (b) *Spray and Wait* [12], and (c) *PRoPHET* [16].The reason for selection of only these three protocols is that these protocols are most cited in the DTN literature for evaluations, such as [10, 14, 17, 23, 24, 26, 27, 32], as the aforementioned protocols are known to exhibit consistent performances in many different type of DTN scenarios. Alternatively, the *MaxProp* and *Rapid* protocols were specifically designed for bus-based DTN scenarios, and the *Wave* and *Life* protocols were developed for scenarios where the nodes were mostly static. (We did not consider single copy routing schemes in our evaluations because of their low performance.) Therefore, to test the performance of our proposed enhanced schemes, we designed the scenarios that closely matched with the original scenarios in which the abovementioned protocols were evaluated. Moreover, to favor the *Epidemic* and *Spray and Wait* protocols, we introduced random mobility nodes in our defined scenario. Similarly to accommodate the *PRoPHET* protocol, our scenario also contains grid-based communities. The message replication in the proposed schemes is made adaptive to the varying network conditions. In the following subsections, we illustrate the proposed schemes.

5.1 Enhanced Epidemic Scheme

The *Epidemic* protocol performs large scale flooding to increase messages' delivery probability. However, flooding cause network congestion and buffer shortage, resulting in an increased message drop and overhead. In realistic scenarios, the nodes' densities at various network regions do not stay uniform as the network topology varies with time. Therefore, message replication should be adjusted depending upon a node's frequency of interactions with specific destinations. The same concept is applied here in the proposed enhancement of the *Epidemic* protocol.

The new technique *e-Epidemic* attempts to decrease message flooding in cases when specific set of destinations are repeatedly encountered. In *e-Epidemic*, when a node i comes into contact with a neighbor node j, both the nodes exchange metadata information about their recent interactions with other network nodes. This information is quantified as:

$$E_i^d = \frac{1}{T_c - T_i^d} \times \sum_k C_i^d(k). \quad \text{for} \quad k \in \mathbb{N} \tag{7}$$

In the above equation, the parameter E_i^d is the node i's estimated interaction with the destination node d. The parameter $\sum_k C_i^d(k)$ represents contact history of node i with the message's destination d, the parameter T_c is the current time, and T_i^d is the last interaction time between the node i and the node$d$. The last interaction times are maintained as local timers at the nodes, as was performed in [13]. Equation (7) states that a node i will "forward" a message to a neighbor node j, if and only if, the node j's estimated interaction rate (E_j^d) with destination d is higher and more recent than node i's interaction rate (E_i^d). Otherwise, node $i$will simply "replicate" the message.This behavior is different from the original *Epidemic* protocol where a message is always replicated, and as a result the *Epidemic* protocol exhibits low performance in scenarios with limited buffer size. Figure 10 reflects the performance improvement achieved by *e-Epidemic* for delivery ratio and overhead. The decreased number of copies per message reduces the number of transmissions per message, which results in the reduction of overhead. However, decreased number of copies per message also increases the overall latency (Fig. 10b). This is because of the increased hop count per message. The delivery ratio is also further improved by introducing passive message acknowledgements for already delivered messages. The acknowledgements help in clearing up buffers from redundant message copies. Therefore, there will be less message drop due to lack of buffer space.

5.2 Adaptive Multi-Copy Spray (AMS)

The *Spray and Wait* protocol controls flooding by setting a limit on the maximum number of message copies in the network. When a message is created, it is assigned a specific number of "forwarding tokens". When the message is replicated, the sender node splits the tokens into half, keeping half for itself and giving the other half to the neighbor node. Although splitting the message tokens into half reduces the overhead, a more optimal splitting based on varying network conditions may lead to further reduction in overhead and improved delivery ratio. Therefore, we propose an enhancement in the *Spray and Wait* protocol such that the token splitting is made adaptive, depending on a node's interaction frequency with other nodes in the network. The proposed technique, *Adaptive Mutli-Copy Spray (AMS)* utilizes the following equation to calculate a node's interaction history:

$$E_i^d = \left(\sum_k C_i^T(k) + \alpha \times \sum_l C_i^d(l) \right) \times \frac{T_c}{2T_c - T_i^d}, \tag{8}$$

where E_i^d is the estimated interaction rate. In (8), the parameter $\alpha = \sum_m C_i^d(m) / \sum_n C_i^T(n)$, where $\sum_m C_i^d(m)$ is the total contact duration of node i with the destination node d, and the

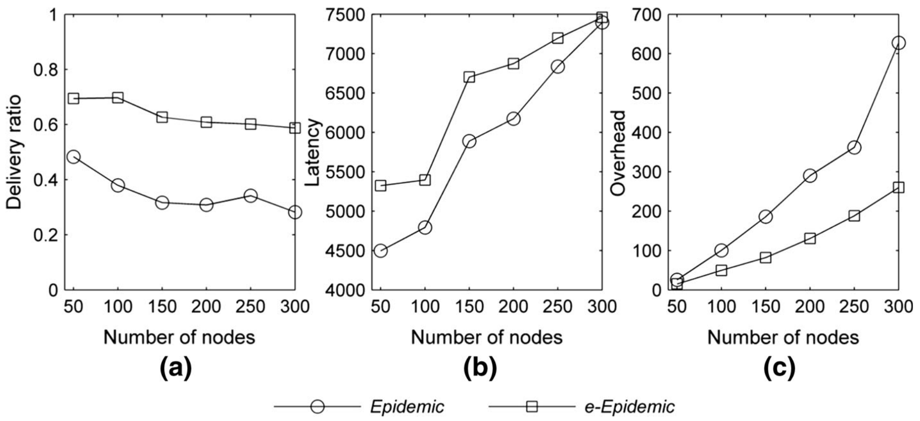

Fig. 10 Performance comparison between Epidemic and E-Epidemic protocol

parameter $\sum_n C_i^T(n)$ indicates the total contact duration of node i with all other nodes in the network. The parameter $\alpha$is the adaptive weight factor representing the fraction of a node's interactions with a destination d, and is added to the total interactions $\sum_k C_i^T(k)$ of the node i. The term $T_c/(2T_c - T_i^d)$ on the right hand side of (8) ensures that higher weightage should be given to more recent contacts. A newly created message m_k at node $i$is initially assigned L forwarding tokens, which is the same number of forwarding tokens used by the original versions of the *Spray and Wait* and *Spray and Focus* protocols ($L \approx$ 10–15 % of the total number of nodes). From Fig. 11, we observe that the *AMS* protocol achieves higher delivery ratio at roughly same value of L, with a network size of hundred nodes.

Let E_i^d and E_j^d be the estimated interaction histories of node i and j, respectively, with a message's destination d. Suppose, a message m_k hasL_k forwarding tokens. When node i transfersm_k to node j, the token splitting is performed, such that node i keeps L_k^i tokens and gives L_k^j tokens to node j according to the following equations:

$$\text{Node} \quad i \text{ tokens } L_k^i = \left\lfloor L_k \times \frac{E_j^d}{E_i^d + E_j^d} \right\rfloor, \tag{9}$$

$$\text{Node} \quad j \text{ tokens } L_k^j = \left\lceil L_k \times \frac{E_i^d}{E_i^d + E_j^d} \right\rceil. \tag{10}$$

It can be observed from (9) and (10) that node i keeps the floor whereas node j keeps the ceil value of tokens. The above two equations indicate that a node with lesser number of interactions with a message's destination will receive greater number of tokens and vice versa. The proposed approach is intended to improve the message delivery through those nodes which have less frequently interacted with the destination d, by increasing the per message replications performed by such nodes. The simulation results in Fig. 12 indicate that the proposed *AMS* scheme performs better than *Spray and Wait* and *Spray and Focus* in terms of delivery ratio and overhead because of the adaptability in the message replication. However, reduced overhead is at the expense of higher latency. This is because the greater numbers of messages are replicated through nodes that less frequently interacted with messages' destinations. The message drop rate due to buffer shortage is controlled by

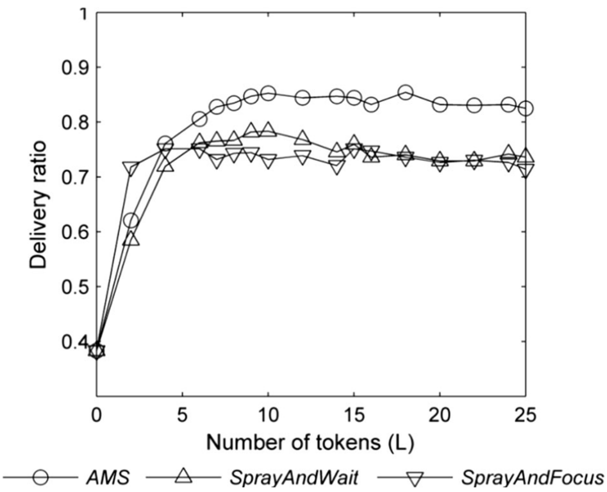

Fig. 11 The max delivery ratio in AMS, Spray and Wait, and Spray and Focus is achieved with L = 10–15

introducing passive message acknowledgements that further improve the delivery ratio performance.

5.3 Adaptive Source Token Multi-Copy Spray (ASTMS)

As illustrated in the *AMS* protocol, a message's initial token value is statically assigned (Sect. 5.2). In the proposed *ASTMS* scheme, the initial token assignment is made adaptive to a node's varying interaction history in the network. The idea presented here is mathematically formulated as follows:

$$L_k^i = \left\lceil \left(1 - E_i^d\right) \times L \right\rceil, \tag{11}$$

where the parameter L_k^i is the token value generated for the message m_k at the node i. The estimated interaction rate E_i^d is given by:

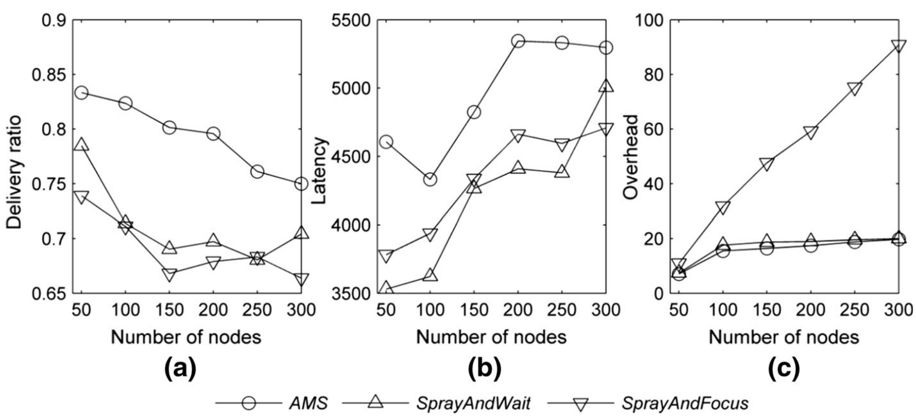

Fig. 12 Performance comparison of AMS with Spray and Wait, and Spray and Focus

$$E_i^d = \frac{C_i^d \times T_c}{C_i^T \times (2T_c - T_i^d)}. \tag{12}$$

In above equation, the parameter T_c represents the current time and T_i^d represents the last interaction time of node i with node d. The estimated interaction rate E_i^d depends on the fraction of recent contacts between node i and destination d, whereas C_i^T is the total interaction rate of i, in the whole network.The parameter L is similar to the one used in the previous subsection and is considered as an upper bound to the maximum value of forwarding token. Equation (11) indicates that if the node i more frequently interacts with the destination$d$, then smaller token value L_k^i will be generated for the message m_k at node i. Therefore, the initial token value is dependent on a node's interaction frequency with a specific destination set. Based on the proposed approach, we introduce adaptive token assignment mechanism in the *PRoPHET* protocol. The *PRoPHET* protocol performs message replication if and only if the delivery predictability of the current message is comparatively higher at the neighbor node. However, the *PRoPHET* protocol sets no limit to the maximum number of message replicas. This may result in an increased overhead and message drop rate in the resource constrained network environments. The aforementioned problem is addressed in the proposed *ASTMS* scheme in the following ways: (a) by introducing adaptability in initial token assignment, (b) by setting a limit on maximum number of message replicas, and (c) by introducing adaptability in token splitting during message replication.

Algorithm 1: *Pseudo-code for ASTMS scheme*

1: $m_k \leftarrow App;\ L_k^i = \lceil (1 - E_i^d) \times L \rceil;\ M_i = M_i \cup \{m_k\}$
2: **for each** $m_k \in M_i$**do**
3: **if** $L_k > 1$**then**
4:   **if** $E_j^d > E_i^d$ **then**
5:   $L_k^j = L_k \times E_j^d / (E_i^d + E_j^d)$
6: **end if**
7: *Replicate*m_k to ***j***
8: **else**
9: Find a neighbor k with highest E_k^d
10: *Forward*m_k to ***j***
11: **end if**
12: **end for**

All such enhancements depend on a source node's interaction history with the message's destination. Algorithm 1 presents the pseudo-code for the *ASTMS* scheme. A message created by an application (*App*) is assigned the initial token value and stored in the message queue at the node i (Line 1). A message with token value greater than one will be replicated, if and only if the interaction history of neighboring node is greater than the current node (Lines 3 and 4). The token splitting is performed in Line 5, such that, a node with more frequent interactions with the message destination will get higher number of tokens. A message with token value equal to one will be forwarded towards a neighbor with highest interaction value E_k^d (Lines 9 and 10). The same process will be repeated for other messages in the queue.

Figure 13 indicates the improved performance of proposed scheme in comparison to the *PRoPHET* protocol. The reduction in message replications by setting a limit on maximum number of message copies results in less utilization of buffer space as well as decreases the

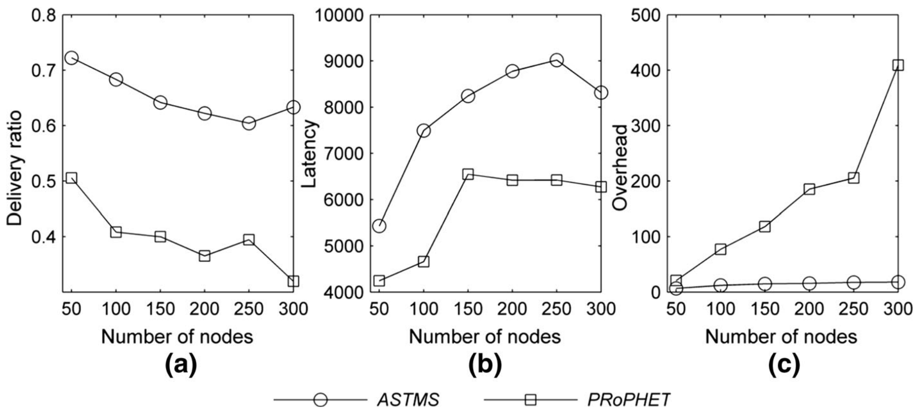

Fig. 13 Performance comparison of ASTMS with the PRoPHET protocol

message drop rate. Therefore, message delivery ratio is significantly improved. Moreover, the decrease in message copies improves the overhead, as per message transmissions are also minimized. However, the reduced overhead is at the expense of increased latency. This is because, with the decrease in message replications, the messages may have to wait longer to be delivered to the destinations.

6 Conclusions

This paper presented a detailed simulation and analysis of ten popular DTN routing protocols. Selecting the best protocol to be used in a given environment remains a difficult task, since comparisons are often clouded by different superfluous assumptions in the original design of a protocol. The simulation results indicated that the protocols that utilized additional network information to route messages indicate better performance than the protocols that mostly relied on flooding based techniques. However, increasing the size of metadata and protocol complexity also increases computational cost and buffer requirements. It is further observed that the performance of DTN routing protocols is greatly affected by mobility pattern of nodes and message queue management policies. Moreover, the simulation results provided a better understanding about which of the selected protocols fits best in a given network environment. As an additional contribution, we proposed three routing techniques by introducing adaptability in the replication strategies of the three most cited DTN routing protocols. When compared with existing protocols, the proposed schemes indicated significantly improved performance. In future, we intend to expand the functionality of proposed techniques to make them a workable solution for providing opportunistic message transfer in vehicular DTN environments, and testing the performance of protocols using famous mobility models, such as IDM and MOBIL.

References

1. Delay-Tolerant Networking Working Group. (DTN). https://datatracker.ietf.org/wg/dtnwg/. Accessed May 2015.

2. Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., et al. (2003). Delay-tolerant networking: An approach to interplanetary internet. *IEEE Communication Magazine, 41*(6), 128–136.
3. Cadger, F., Curran, K., Santos, J., & Moffet, S. (2016). Location and mobility-aware routing for improving multimedia streaming performance in MANETs. *Wireless Personal Communications, 86*(3), 1653–1672.
4. Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A. & Qayyum, A. (2003). Optimized link state routing protocol (OLSR), *RFC 3626*, October 2003.
5. Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad hoc on-demand distance vector (AODV) Routing. *RFC 3561*, July 2003.
6. Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. *Imielinski and Korth, Eds.,Mobile Computing, Vol. 353*, 1996.
7. Anjum, S. S., Noor, R. M., & Anisi, M. H. (2015). Review on MANET based communication for search and rescue operations. *Wireless Personal Communications* (pp. 1–22). First online: 22 December.
8. Cheng, R., Chen, N., Chou, Y., & Becvar, Z. (2015). Offloading multiple mobile data contents through opportunistic device-to-device communications. *Wireless Personal Communications, 84*(3), 1963–1979.
9. Horng, G. (2015). Opportunistic content sharing scheme for distributed network in city environments. *Wireless Personal Communications, 84*(4), 2327–2350.
10. Balasubramanian, A., Levine, B., & Venkataramani, A. (2010). Replication routing in DTNs: A resource allocation approach. *IEEE/ACM Transactions on Networking, 18*(2), 596–609.
11. Spyropoulos, T., Turletti, T., & Obrazcka, K. (2009). Routing in delay tolerant networks comprising heterogeneous populations of nodes. *IEEE Transaction on Mobile Computing, 8*(8), 1132–1147.
12. Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2005). Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM WDTN* (pp. 252–259).
13. Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2007). Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Proceedings of the fifth IEEE international conference on pervasive computing and communications workshop*, pp. 79–85.
14. Spyropoulos, T., Rais, R. N. B., Turletti, T., Obraczka, K., & Vasilakos, A. (2010). Routing for disruption tolerant networks: taxonomy and design. *ACM Journal Wireless Networks, 16*(8), 2349–2370.
15. Yasmin, S., Rais, R. N. B., Qayyum, A. (2015). A multi-attribute routing protocol for opportunistic network environments. In *Proceedings of 23rd international conference on computer communication and networks (ICCCN)* (pp. 1–6).
16. Lindgren, A., Doria, A., & Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review, 7*(3), 19–20.
17. Burgess, J., Gallagher, B., Jensen, D., & Neil Levine, B. (2006). MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proceedings of IEEE infocom* (pp. 1–11) April, 2006.
18. Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2004). Single-copy routing in intermittently connected mobile networks. In *Proceedings of sensor and ad hoc communications and networks* (pp. 235–244).
19. Hsu, Y.-F., & Chih-Lin, H. (2015). Erasure coding-based message forwarding to multiple destinations in intermittently connected networks. *International Journal of Ad-Hoc and Ubiquitous Computing, 9*(1), 75–85.
20. Kerdsri, J. & Wipusitwarakun, K. (2015). Dynamic rendezvous based routing algorithm on sparse opportunistic network environment. *International Journal of Distributed Sensor Networks, 11*(2). doi:10.1155/2015/819178.
21. Yasmin, S., Rais, R. N. B., & Qayyum, A. (2016). Resource aware routing in heterogeneous. *International Journal of Distributed Sensor Networks, 2016*, 1–18.
22. Ko, E., Kim, D., Park, H., Yeom, I., & Seo, E. (2015). An end-to-end rate control protocol for intermittently connected networks. *Wireless Personal Communications, 84*(1), 287–303.
23. Khalid, O., Khan, S. U., Kolodziej, J., Zhang, L., Li, J., Hayat, K., Madani, S. A., Wang, L., & Chen, D. (2012). A checkpoint based message forwarding approach for opportunistic communication. In *European conference of modeling and simulation*.
24. Sandulescu, G., & Nadjm-Tehrani, S. (2008). Opportunistic DTN routing with window-aware adaptive replication. In *Proceedings 4th Asian conference on internet engineering* (pp. 103–112).
25. Johari, S. R., Gupta, N., & Aneja, S. (2015). Experimental evaluation of routing schemes for intermittently connected wireless mobile networks. *Wireless Personal Communications*, 1–25. First online: 07 November 2015
26. Cao, Y., & Sun, Z. (2013). Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications Surveys and Tutorials, 15*(2), 654–677.

27. Voyiatzis, A. G. (2012). A survey of delay- and disruption-tolerant networking applications. *Journal of Internet Engineering, 5*(1), 1–331.
28. Keränen, A., Ott, J. & Kärkkäinen, T. (2009). The ONE simulator for DTN protocol evaluation. In *Proceedings of 2nd international conference on simulation tools and techniques*. doi.10.4108.
29. Infocom06 connectivity traces on CRAWDAD website (2016). http://crawdad.cs.dartmouth.edu/meta.php?name=cambridge/haggle#N100C4
30. Jain, S., Fall, K., & Patra, R. (2004). Routing in a delay tolerant network. In *Proceedings of conference on applications, technologies, architectures, and protocols for computer communications* (pp. 145–158).
31. Vahdat, A., & Becker, D. (2000). Epidemic routing for partially connected ad hoc networks. *Technical report CS-200006, Duke University*.
32. Ott, J., Keränen, A., & Hyytiä, E. (2011). BeachNet: Propagation-based information sharing in mostly static networks. In *Proceedings of ACM ExtremeCom*.
33. Gardner, M. (1970). The fantastic combinations of John Conway's new solitaire game life. *Scientific American* (pp. 120–123).
34. OMNET++Simulator. https://omnetpp.org/. Accessed February 2016.
35. NS-3 Simulator. https://www.nsnam.org/. Accessed February 2016.
36. Aung, C. Y., Seet, B. C., Zhang, M., Xie, L. F., & Chong, P. H. J. (2015). A review of group mobility models for mobile ad hoc networks. *Wireless Personal Communications, 85*(3), 1317–1331.
37. *OpenStreetMap (OSM) API website* (2016). http://www.openstreetmap.org/
38. *OPENJUMP GIS Software* http://www.openjump.org/. Accessed July 2015.

Dr. Osman Khalid completed his Ph.D. in 2014 at the North Dakota State University, Fargo, USA. Previously, he obtained degree of Masters in Computer Engineering from Center for Advanced Studies in Engineering (CASE), Islamabad—Pakistan. His area of research includes opportunistic networks, recommendation systems, and trust and reputation systems. He is Assistant Professor in COMSATS Institute of Information Technology, Abbottabad, Pakistan.

Dr. Rao Naveed Bin Rais received his M.S. and Ph.D. degrees in Computer Engineering from INRIA/University of Nice, Sophia Antipolis, France in 2007 and 2011 respectively. Before that, he received his B.E. in Computer Systems from National University of Sciences and Technology (NUST), Pakistan in 2002. He is Associate Professor, College of IT, Ajman University, UAE. He has 7+ years of teaching, 8+ years of research and 4+ years of industrial experience. His research interests include future internet architecture, network protocol design for heterogeneous networks and mobile ad-hoc networks (MANETs), information-centric and software-defined networking approaches.

Dr. Sajjad A. Madani is currently an Associate Professor and the Chairman of the Departmentof Computer Science, COMSATS Institute of InformationTechnology, Abbottabad, Pakistan. He has published more than 60 research articles in peerreviewedjournals and conferences. His research interestsinclude wireless sensor networks, applicationof IT to electrical energy networks, and large-scale systems.