

## Research Article

# Resource Aware Routing in Heterogeneous Opportunistic Networks

**Sadaf Yasmin, Rao Naveed Bin Rais, and Amir Qayyum**

*Center of Research in Networks and Telecom (CoReNeT), Capital University of Science and Technology, Islamabad 44000, Pakistan*

Correspondence should be addressed to Sadaf Yasmin; [syasmin@corenet.org.pk](mailto:syasmin@corenet.org.pk)

Received 19 September 2015; Revised 15 December 2015; Accepted 20 December 2015

Academic Editor: Bernabe Dorronsoro

Copyright © 2016 Sadaf Yasmin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In heterogeneous scenarios, nodes greatly differ with respect to their capabilities and mobility patterns. Moreover, episodic connectivity in opportunistic networks further aggravates the problem of finding a suitable next-hop to obviate unnecessary utilization of network resources. In this paper, we present a Multiattribute Routing Scheme (MARS) based on “Simple Multiattribute Rating Technique” (SMART) that collects samples of vital information about a node’s different characteristics. This stochastic picture of a node behavior in multiple dimensions is then effectively employed in calculating its next-hop fitness. We also devise a method based on learning rules of neural networks which dynamically determines relative importance of each dimension to maximize next-hop utility of a node. With simulations, using synthetic and real mobility traces against well-known utility-based schemes, we show that MARS can achieve better delivery ratios despite introducing limited redundancy within the network.

## 1. Introduction

In sparse mobile ad hoc networks, there may exist no contemporaneous end-to-end path between a pair of nodes. This violates the assumption on which many ad hoc routing protocols [1, 2] are based, thereby providing room for conducting research in identifying ways to provide connectivity in an environment that does not rely on the Internet’s basic assumptions, such as presence of bidirectional end-to-end paths. Wildlife monitoring, disaster recovery networks, and military applications are some of the examples of such environments. Opportunistic networking help to cope with this dichotomy by making use of transmission opportunities emerging from coarse-grained mobility within the network. Sporadic links appearing among nodes can be eventually construed over a period of time as presence of a complete path between a source and destination pair. Therefore, routing in these networks is performed in per-hop manner, in which every node opportunistically selects a next-hop relay to achieve eventual delivery, with store-carry-and-forward paradigm [3].

However, due to intrinsic nature of opportunistic networks, based on sporadic connectivity, high packet loss

ratios are inevitable. Intuitively, many routing protocols [4–7], for opportunistic environments, are accustomed to spread multiple redundant copies of a message in order to achieve throughput and efficiency in end-to-end latency. These replication based schemes can be categorized into greedy-based [4, 5] and utility-based approaches. Greedy-based replication assumes homogeneous set of nodes [8] and spread replicas on every proximity encounter to increase the probability of successful delivery. However, this assumption reveals itself unrealistic in many scenarios where nodes exhibit heterogeneous characteristics, in terms of their mobility patterns and available resources, that might be deterrent for their participation in the delivery process. There has been a succession of utility-based routing schemes [6, 7]—choosing intermediate relays based on next-hop fitness of nodes—to avoid unnecessary overhead incurred by greedy replication in heterogeneous environments. In this case, number of extra copies of a message is directly proportional to the number of nodes presenting next-hop fitness above certain threshold value. Hence, limitations in determining next-hop utility of a node can ironically result in very high number of message replicas compared to those actually needed to achieve successful delivery at eventual destination. Each copy

consumes energy to transmit along with extra computational resources, thereby leading to suboptimal performance in resource-stringent networks. Number of forwarding tokens, however, can also be assigned to limit message replication [8–10]. In this case, first encountering nodes meeting the given utility criteria are selected for distribution of tokens at each hop. Hence, correct determination of next-hop fitness of a node becomes more apparent to achieve eventual delivery.

Moreover, the intrinsic nature of utility-based schemes, based on pruning the epidemic distribution tree, makes it inevitable that the majority of the network traffic is carried by only the most suitable nodes, resulting in an unfair load distribution [11]. Intuitively, the utility heuristics based on any single attribute are not sustainable as the inefficiency of the algorithm to get itself cognizant of changing network characteristics can quickly deplete constraint resources in few better nodes.

These problems illustrate the need of a routing protocol that can achieve high delivery rates in resource-stringent environments, along with better load-balancing, despite introducing limited redundancy within the network. With this in mind, in this paper, we present a hybrid utility-based routing protocol called Multiattribute Routing Scheme (MARS) for opportunistic environments that determines a node's next-hop fitness through an optimized combination of a set of multiple parameters based on "Simple Multiattribute Rating Technique" (SMART) [12]. These parameters can be selected based on destination independent and dependent characteristics of a node to better reflect its suitability as next-hop relay. Moreover, MARS is not based on rigorous forecasting. We devise a method to assign scores to different attributes, like it is done in "Simple Multiattribute Rating Technique" (SMART) [12], based on past history of encounters of a node with other nodes. Further, with MARS, we can assign different number of forwarding tokens to limit number of replicas of a message along with uncontrolled replication.

Routing decisions based on multiple attributes are explored in many routing protocols [13–15] to provide QoS in MANETs. Rank Order [13] and Threshold [14–16] methods are generally used to select a best route when multiple metrics are involved. These techniques generally fail to provide a good compromise on available metrics because of not considering all objectives simultaneously. Moreover, these protocols only work in environments where all potential candidates that is routes are available at the time of making a decision. Analogously, many existing routing protocols for opportunistic networks determine next-hop utility or fitness of a node based on multiple metrics that can be well placed in social-context based or prediction-based utility schemes. Social-context based protocols take into account social ties between nodes while determining next-hop fitness of a node. These schemes are applicable in environments where user profile including name, address, and occupation is attached to every node [17, 18] or they can be well divided into different communities [19–21]. The utility of a node is determined taking into account an increasing match between the context of the node and of destination, as an important parameter. These schemes are beyond the scope of this paper. Instead, we are interested in prediction based utility schemes where next-hop

fitness of a node is calculated based on destination-dependent or destination-independent utility or fitness parameters [8]. Analogous to these schemes, MARS determines destination independent/dependent characteristics of nodes and employs them in calculating their next-hop fitness or utility.

With limited evaluations, an earlier version of the protocol can be found in [22]. We extend that paper with more comprehensive evaluation of MARS against well-known routing schemes using a variety of performance metrics and parameters, and in heterogeneous synthetic and real network environments. Moreover, we propose an adaptive weighting mechanism based on error correction learning which is a widely used technique in neural networks.

Our contributions in this paper and differences from [22] can be summarized as follows:

- (i) Instead of offline normalization by defining an upper limit for each individual attribute, we define a common time window, and with this we describe a new method for dynamic calculation of individual scores within range  $[0 \cdots 1]$ . With this approach, scalability of MARS to add new parameters is improved.
- (ii) An autoweight adaptation mechanism is proposed to dynamically define relative importance of different attributes for maximizing next-hop utility of a node.
- (iii) MARS is thoroughly evaluated against variety of parameters in heterogeneous synthetic and real mobility traces. Moreover, comparison against well-known destination-dependent, destination-independent, and hybrid utility-based schemes is included to depict a more realistic picture on performance gain achieved through MARS.
- (iv) With offline adjustment of weights and normalization of individual attribute, as done in [22], optimal values can be identified for configuring the protocol to give better performance based on the given scenario. However, realistic results are presented in this paper with dynamic adaptations, as the protocol itself gets cognizant of networking conditions to maximize performance in different scenarios.

Rest of the paper is organized as follows. In Section 2 we present the literature review of some existing routing protocols. Section 3 contains an overview and elaborated description on design of MARS with details on adaptive weighting mechanism. Load-balancing and choosing number of forwarding tokens with MARS are explained in Section 4. Considerations regarding privacy and MARS's computational overheads are discussed in Section 5. Simulation results are presented in Section 6 with conclusion at the end in Section 7.

## 2. Related Work

In heterogeneous scenarios nodes may differ in terms of their capabilities and moving patterns. In this case, forwarding without considering node's candidacy to contribute in the delivery process may waste network resources with unnecessary overhead [23]. This arises the need to discover better

relays, in terms of their ability or “utility” to help in delivering a message to its intended destination. In the following, we present a review of well-known existing schemes.

Social-context based utility schemes are applicable in environments where user profile including name, address, and occupation is attached to every node [17–19] or they can be well divided into different communities [19–21]. In this case, utility of a node is determined taking into account an increasing match between the context of the node and of destination. On the other hand, in history/prediction based schemes, every node try to build a partial picture of network topology based on its history of encounters with other nodes. For example, EBR [10] suggests to handover tokens to social nodes having higher rates of encounters with other nodes in the network. Spyropoulos et al. [8], has also exploited sociability of a node as its utility to forward packets.

Most utility-based schemes [6, 7, 9] are generally aimed to characterize future probability of contact with potential destinations by identifying mobility behavior of nodes within the network. Work from Balasubramanian et al. [24] called RAPID routing replicates messages depending on per packet utility functions to minimize specific metrics such as average delay, maximum delay, and number of packets missing the deadline. In RAPID, the utility value assigned to packets is mostly based on intermeeting time between peer nodes. A packet is replicated only if it has high utility value compared to other packets within node’s buffer while the fitness of relay node being able to contribute in the delivery process is ignored. A number of schemes use age of last encounter timers along with location information to determine utility of a node [8, 25]. Another utility parameter that is used in a recent study, named interest-tree based routing [26], is defined as users’ interests. It is based on the reasoning that people with similar interests often meet frequently. In [27] a node maintains a history table of its visited locations through a global positioning system (GPS). The exchanged history table is then utilized by a node to predict future location of peer. Along with this location, speed stability and future direction are used to calculate a utility metric. If the utility of peer is greater than a defined threshold, it is selected as next-hop for further forwarding the message.

MARS is intrinsically different from these techniques as along with characterizing the mobility, it also considers other stochastic information related to network resources, such as, available free buffer space at different nodes.

A utility model, with buffer and remaining energy at a node along with its social value, is proposed in [28]. This utility value is then used to select ferries among other network participants for message forwarding. However, in this work [28] a global information about present resources within the given network is required to describe the ranking of each node. Calculation of delivery probability based on multiple utility functions, applied on different context information of a node is also explored in CAR [29] and SCAR [30]. The composed utility function is calculated using multiattribute utility problem [31]. These protocols require accurate prediction of future value of context information [32]. However, accurate forecasting may not be possible in many mobile scenarios and becomes more appealing in highly heterogeneous network

environments where nodes differ greatly in terms of their capabilities, and mobility patterns. In another study [33] a node’s delivery capability (NDC) from available energy, speed, and distance from destination is calculated. However, the proposed linear combination [33] lacks a generic solution to involve more than one mutually inclusive attribute pair.

MARS is not based on rigorous forecasting techniques. We devise a method to assign scores to different attributes, like it is done in “Simple Multiattribute Rating Technique” (SMART) [12]. Secondly, the ability to dynamically learn the environment intrinsic to MARS provides the flexibility to make decisions based on available local information, which can certainly improve scalability of the protocol in real world deployments. Moreover, transitive fitness calculation in MARS allows communication between two far apart nodes in highly heterogeneous scenarios. Further, we consider DTN routing as multiresource optimization problem in highly heterogeneous mobile opportunistic networks.

### 3. Multiattribute Routing Scheme (MARS)

*3.1. Overview.* Multiattribute Routing Scheme (MARS) is a hybrid utility-based replication and forwarding scheme for opportunistic networks. With MARS next-hop selection is based on fitness value of a node to message destination. Every MARS node maintains a contact table which contains its next-hop fitness to so far known destinations. On sporadically emerging links within the network, every node shares its fitness table with its peer after aging its values. The aging mechanism used in MARS is similar to PROPHET routing protocol [6], which helps in removing stale information about node’s next-hop fitness to known destinations in order to avoid misled routing decisions. Message transfer is initiated if node has a message  $m$  in its buffer with pending delivery and decides to forward it based on next-hop fitness value of peer node to  $m$ ’s destination (forwarding mechanism is discussed in Section 3.6). We calculate fitness of a node based on multiple parameters such as encounter rate, contact duration, and free buffer space. A simplistic exemplary scenario for calculating next-hop fitness based on multiple node attributes is shown in Figure 1. The key points we want to highlight with the help of this example are the following. First, analogously, in MARS all parameters (represented as  $P_1, P_2, P_3$  in Figure 1) are considered simultaneously for fitness calculation, and only the node which represent a better trade-off of all considered parameters is selected as next-hop relay. For example, both  $n_1$  and  $n_2$  can be used as next-hop relays for destination  $D_2$ . Secondly, this helps in better load distribution over network resources as it helps in avoiding situations, where only few nodes may bear most of the forwarding overhead due to representing high values in any single attribute (such as number of encounters). Load-balancing with MARS is explained in detail in Section 4. In MARS, parameter combination consisted of destination-independent (such as free memory space) and destination-dependent characteristics (such as encounter rate and contact duration) of nodes; it is therefore stated as hybrid utility-based scheme. We also propose an adaptive mechanism for assigning weights to determine relative importance of

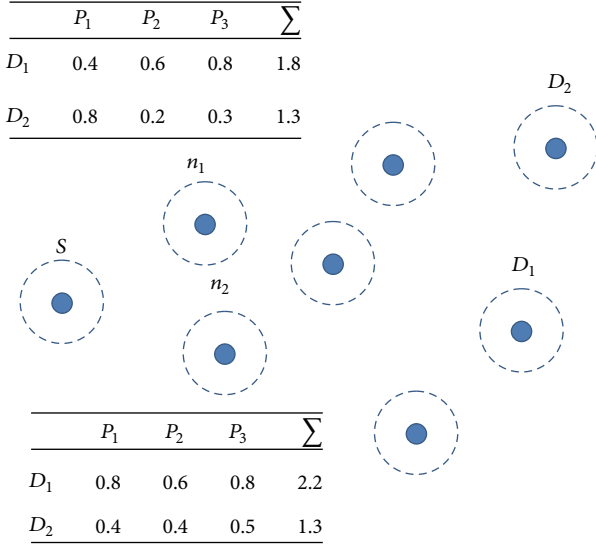


FIGURE 1: Next-hop fitness based on multiple parameters.

each parameter to form an optimized combination reflecting application and network requirements.

Moreover, in many scenarios it is often useful to limit the number of forwarding tokens to some required value as can be seen in work from Spyropoulos et al. [5]. Therefore, MARS exhibits this flexibility to reduce overhead by assigning number of forwarding tokens with messages. Even when no limit is defined on replication MARS can efficiently steward network resources by taking wise decisions while selecting next-hop nodes. Different symbols used in this paper are defined in List of Frequently Used Symbols.

**3.2. Next-Hop Fitness Calculation.** In MARS, every node calculates its fitness to serve as next-hop for other nodes within the network, as a function of different metrics based on “Simple Multiattribute Rating Technique” (SMART) [12]. These metrics are carefully selected to represent destination-dependent and destination-independent characteristics of a node. Moreover, every metric is assigned an associated weight to represent its relative importance within an optimized combination. Opportunistic networks are characterized to have uncertain topologies with frequent partitions due to high mobility and heterogeneous capabilities of nodes. Therefore, in order to achieve stability and correctness of opinion about utility of a node, it is important to consider variations in its behavior over a period of time. For this purpose, we introduce a *Time Window* ( $\Delta T$ ) to maintain a limited history of past information, that is, employed in calculating individual score of each attribute. A sample of new information is added on every encounter between two nodes, and interactions older than  $\Delta T$  are removed.

The generalized formula to calculate next-hop fitness at a node  $p$  for a destination node  $q$  is then represented as

$$F_{(p,q)} = \sum_{i=1}^n (w_i \times \widehat{S}(\mu_{(p,q)}; \Delta T)_i). \quad (1)$$

Weight  $w_i$  is defined as associated weight of  $i$ th parameter and its value can vary within range  $[0, 1]$ . Moreover, sum of all weights used for different parameters is always equal to 1. This tweaking of weights help us fine tune different network parameters to better calculate next-hop fitness values helping in forwarding a message towards its destination.  $\mu[1 \dots n]$  are some given independent characteristics of a node  $p$ , such as available free space or average residual energy, and attributes that are correlated with a destination, such as encounter rate or contact duration with node  $q$  that is considered to calculate its fitness as next-hop relay for the latter node. We have shown  $\widehat{\cdot}$  on  $S$  to represent change in score of each attribute with respect to time.

In this paper, the metrics which we consider for our calculations are (1) encounter rate ( $\widehat{S}(\lambda_{p,q}; \Delta T)$ ), (2) contact duration ( $\widehat{S}(\Omega_{p,q}; \Delta T)$ ), (3) free buffer space ( $\widehat{S}(\Theta_p; \Delta T)$ ), and (4) meeting likeliness ( $\widehat{S}(\Upsilon_{p,q}; \Delta T)$ ). Hence, (1) can be rewritten as follows:

$$F_{(p,q)} = w_1 \times \widehat{S}(\lambda_{p,q}; \Delta T) + w_2 \times \widehat{S}(\Omega_{p,q}; \Delta T) + w_3 \times \widehat{S}(\Theta_p; \Delta T) + w_4 \times \widehat{S}(\Upsilon_{p,q}; \Delta T). \quad (2)$$

Here, the *Time Window*  $\Delta T$  is an important parameter to be defined, so as its width should involve a reasonable set of samples to help in making an educated guess about the dynamic behavior present within the network. Certainly, this can be defined keeping in mind a rough idea about interencounter times within the given network. In this paper, we have used empirically defined values of  $\Delta T$  in different scenarios. However, many experimental studies [34, 35] intend to analyze properties of real and synthetic mobility models, whose findings can provide a better guideline to configure opportunistic forwarding algorithms.

Moreover, attribute scores are real numbers within range  $[0, 1]$ ; their higher values represent a high probability that node  $p$  can be used as relay for destination  $q$  by other nodes in the network. In the following, we explain in detail calculation of each value.

**3.2.1. Estimate of Contact Duration.** We define  $\widehat{S}(\Omega_{p,q}; \Delta T)$  as the estimate of time duration; the two nodes stay in each other’s vicinity. Intuitively, its high value results in high next-hop fitness of  $p$  to  $q$ , as it allows large data exchanges between two nodes.

Mobile nodes in an opportunistic network may occasionally come close in each other’s vicinity, remain connected for a period of time, and then move apart in different directions. This connected period between the two corresponding nodes is called a contact duration. If we take  $t_1$  as the time of contact between  $p$  and  $q$ , and  $t_2$  as the time when the two nodes depart from each other’s transmission range, then *Contact Duration* ( $c$ ) is defined as follows:

$$c = \min \{ \Delta T, t_2 - t_1 \}. \quad (3)$$

This maximum threshold on contact duration is defined to avoid long-term neighborhood problem [9]. In case a contact remains for an indefinitely large time, an update on contact

duration information is made on every  $\Delta T$  time, and the two nodes recalculate their next-hop fitness to each other.

Next, we define  $\Omega_{p,q}$  as a set, which contains samples of contact durations within time window  $\Delta T$ , between nodes  $p$  and  $q$  paired with time-stamps at which the contact occurs, defined as follows:

$$\Omega_{p,q} = \{t \mapsto c \mid \begin{array}{l} t \text{ is time of contact and } c \text{ is duration of the contact} \end{array}\}. \quad (4)$$

Then, on every contact between  $p$  and  $q$ ,

$$\Omega_{p,q} = \Omega_{p,q} \cup \{t \mapsto c\}. \quad (5)$$

Next, we calculate an exponential moving weighted average ( $\hat{Y}(\Omega_{p,q})$ ) of all contact duration samples within interval  $\Delta T$  as follows:

$$\hat{Y}(\Omega_{p,q}) = \frac{\sum_{i=1}^{|\Omega_{p,q}|} \text{dom}((t \mapsto c)_i)^* \times \text{ran}((t \mapsto c)_i)^\dagger}{\sum_{i=1}^{|\Omega_{p,q}|} \text{dom}((t \mapsto c)_i)}, \quad (6)$$

where  $t \mapsto c \in \Omega_{p,q}$  (\* domain of the  $i$ th pair  $t \mapsto c$  of set  $\Omega_{p,q}$ ) ( $\dagger$  range of the  $i$ th pair  $t \mapsto c$  of set  $\Omega_{p,q}$ ).

Taking weighted average this way helps recent behavior of nodes to have a higher influence while calculating estimate of contact duration, as high weights in terms of time are assigned to the latest activities.

$$T = \begin{cases} 0 & \text{if } \Theta \geq \max \text{ for } \Delta T \text{ time} \\ \Delta T & \text{if } \Theta < \max \text{ for } \Delta T \text{ time} \\ \text{time}(\Theta \geq \max) - \text{time}(\Theta < \max) & \text{Otherwise.} \end{cases} \quad (8)$$

Here,  $\text{time}(\Theta < \max)$  is the initial time; occupied-buffer was observed less than the maximum buffer value. Then the average time it takes to fill the buffer at node  $p$  is calculated as

$$AT_\Theta = \begin{cases} \frac{1}{N} \sum_{i=1}^N T_i & \text{if } i \geq 1 \\ \frac{\Delta T}{2} & \text{Otherwise.} \end{cases} \quad (9)$$

Here,  $N$  is the total number of calculated samples. A node might get congested due to sudden data flows for some time duration, while its buffer remains free most of the time. Moreover, filling of the buffer sometimes indicates node's willingness and a high feasibility to serve as a next-hop relay. Therefore, the time it takes to fill the buffer should be averaged over a large duration or by considering a reasonable amount of samples. For results in this paper we do not put any limit on number of samples; however, for real time networks that are supposed to be running for long duration a limit on sample size could be defined to better reflect recent

Now, an estimate of contact duration between two nodes  $p$  and  $q$  can be calculated as follows:

$$\hat{S}(\Omega_{p,q}; \Delta T) = \begin{cases} \frac{\hat{Y}(\Omega_{p,q}) \times |\Omega_{p,q}|}{\Delta T} & \text{if } (\hat{Y}(\Omega_{p,q}) \times |\Omega_{p,q}|) \geq \Delta T \\ 1 & \text{Otherwise} \end{cases} \quad (7)$$

if we consider a scenario, where two nodes have small interencounter time and often meet each other for short duration, within time window  $\Delta T$ . Then it is possible to have a small weighted average due to short individual contact periods, despite remaining connected with each other most of the time. In order to avoid such behavior, we multiply the calculated weighted average with the size of the set  $\Omega_{p,q}$  before normalizing with the time window. Contact duration plays very important role while determining next-hop fitness of a node. A large estimate on contact durations between two nodes represents that they stay in each other's vicinity most of the time. Therefore, they can better be used as relays for each other.

**3.2.2. Estimate of Free Buffer Space.**  $\hat{S}(\Theta_p; \Delta T)$  is defined as the estimate of time; free buffer space is available at relay node  $p$ . High value represents that node  $p$  has enough free space in its buffer to relay messages most of the time. Node  $p$  reads its occupied-buffer defined as  $\Theta$  at regular intervals to calculate the time  $T$  it takes to get its buffer full:

activity at a node. Then estimate of time, free buffer space available at a node is normalized as follows:

$$\hat{S}_{\text{buff}}(\Theta_p; \Delta T) = \frac{AT_\Theta}{\Delta T}. \quad (10)$$

**3.2.3. Estimate of Meeting Likelihood.**  $\hat{S}(\mathcal{U}_{p,q}; \Delta T)$  is defined as the probability of time duration in which a contact between two nodes  $p$  and  $q$  can occur. High value indicates larger time duration within which the two nodes can meet, thereby resulting in high next-hop fitness of node  $p$  to  $q$ . A rigorous definition of meeting times between two nodes can be found in [36]. For calculating an estimate of meeting likelihood between  $p$  and  $q$  we first define a set  $\mathcal{U}_{p,q}$ , similar to  $\Omega_{p,q}$  described earlier in Section 3.2.1, which contains samples of interencounter times between two nodes paired with contact time within time window  $\Delta T$  as follows:

$$\mathcal{U}_{p,q} = \{t \mapsto e \mid \begin{array}{l} t \text{ is time of contact and } e \text{ is inter-encounter time} \end{array}\}. \quad (11)$$

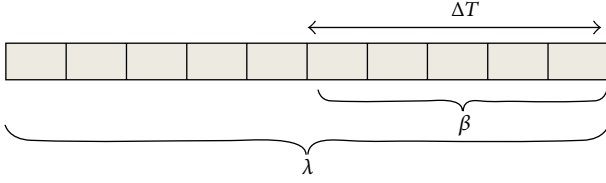


FIGURE 2: Representation of encounter window and encounter count.

Similarly, on every contact between  $p$  and  $q$ ,  $\mathcal{U}_{p,q}$  is updated as follows:

$$\mathcal{U}_{p,q} = \mathcal{U}_{p,q} \cup \{t \mapsto e\}. \quad (12)$$

Here,  $e$  is calculated as

$$e = \begin{cases} \min \{\Delta T, t\} & \text{if no. of encounters} = 1 \\ t - t_1 & \text{if no. of encounters} > 1, \end{cases} \quad (13)$$

where  $t_1$  is the last connection down time of nodes  $p$  and  $q$  and  $t$  is current connection time.

Next, we define an exponential moving weighted average  $\hat{Y}(\mathcal{U}_{p,q})$  of all interencounter time samples present within the set  $\mathcal{U}_{p,q}$  as follows:

$$\begin{aligned} \hat{Y}(\mathcal{U}_{p,q}) &= \frac{\sum_{i=1}^{|\mathcal{U}_{p,q}|} \text{dom}((t \mapsto e)_i)^\ddagger \times \text{ran}((t \mapsto e)_i)^\S}{\sum_{i=1}^{|\mathcal{U}_{p,q}|} \text{dom}((t \mapsto e)_i)}, \end{aligned} \quad (14)$$

where  $(t \mapsto e)_i \in \mathcal{U}_{p,q}$ . ( $\ddagger$  domain of the  $i$ th pair  $t \mapsto e$  of set  $\mathcal{U}_{p,q}$ ) ( $\S$  range of the  $i$ th pair  $t \mapsto e$  of set  $\mathcal{U}_{p,q}$ ). Now, an estimate of interencounter time between two nodes  $p$  and  $q$  can be defined as

$$\hat{\Sigma}_{\text{int-enc}}(\mathcal{U}_{p,q}; \Delta T) = \begin{cases} \frac{\hat{Y}(\mathcal{U}_{p,q})}{\Delta T} & \text{if } \hat{Y}(\mathcal{U}_{p,q}) \leq \Delta T \\ 1 & \text{Otherwise.} \end{cases} \quad (15)$$

Average interencounter time is usually calculated by dividing the sum of intercontact times by total number of encounters [37]. In this way, all interencounter times are assigned equal weights. However, we say that, by taking a running weighted average, recent behavior of node movements can better be depicted. Next, we can calculate meeting likeliness between two nodes  $p$  and  $q$  as

$$\hat{S}(\mathcal{U}_{p,q}; \Delta T) = 1 - \hat{\Sigma}(\mathcal{U}_{p,q}; \Delta T). \quad (16)$$

**3.2.4. Estimate of Rate of Encounter.** Every node  $p$  maintains an encounter count  $\beta_{p,q}$  of its encounters with node  $q$  within time interval  $\Delta T$ . Along with this it also maintains an encounter counter  $\lambda_{p,q}$  of all its encounters with node  $q$ . The difference between these two parameters is shown in Figure 2.

Next, if we take  $\delta$  as size of the time window  $\Delta T$ , we can calculate an estimate on rate of encounter between two nodes  $p$  and  $q$  as follows:

$$\hat{S}(\lambda_{p,q}; \Delta T) = 1 - \left( 1 - \left( \frac{\lambda_{p,q} + \beta_{p,q}}{\lambda_{p,q} + \delta} \right) \right)^{\lambda_{p,q}}. \quad (17)$$

This way, an estimate on encounter rate of two nodes gradually increases over time. We can see that size of the time window should be reasonably large to represent moving behavior of two nodes. On the other hand, by including the encounter counter of previous contacts, we can have a smooth impact of rate of encounter on determining the overall utility of a node.

**3.3. Weight Assignment.** When we calculate next-hop fitness of a node based on multiple attributes, determining relative importance of each dimension becomes very important. By efficiently adjusting different weights assigned to different dimensions, an optimized combination can be formed to calculate next-hop fitness reflecting underlying network characteristics. For example, if there are large interencounter times among nodes within the network, high weight can be assigned to contact duration and available resources at nodes in order to have next-hop fitness of nodes be consistent with network conditions. Adaptive weight assignment to different attributes can result in more educated calculation of next-hop fitness of nodes, which help to improve performance in terms of delivery ratio and routing overhead.

The next-hop fitness equation of MARS can be roughly mapped onto a neural network architecture. If we consider that the desired output that is next-hop fitness of a node is 1. Then, we can use error correction learning [38] to adjust weights at a node in order to maximize its next-hop fitness for known destinations. Let  $F_p$  be the set of fitness values of a node  $p$  for known destinations with whom  $p$  had a direct encounter. We then calculate our error function  $\xi(F_p)$  defined as average deviation of fitness values of node  $p$  from desired output, that is, 1, as follows:

$$\xi(F_p) = \frac{\sum_{i=1}^{|F_p|} (1 - f_i)}{|F_p|}, \quad \text{where, } f_i \in F_p. \quad (18)$$

This average deviation from desired fitness value of a node is back-propagated to adjust individual weights. Next, we define our input parameters at any given node  $p$  as follows: For example, if one of the parameters to calculate fitness of node  $p$  is *Estimate Of Contact Duration*, then,  $A$  is taken as the sorted list of all *Estimates Of Contact Durations* of  $p$  with nodes to whom  $p$  has a direct fitness value (directly encountered nodes). We can then define our  $n$  input parameters as follows:

$$X_i = \text{Median}(A_i) \quad \forall i \in \{1 \dots n\}. \quad (19)$$

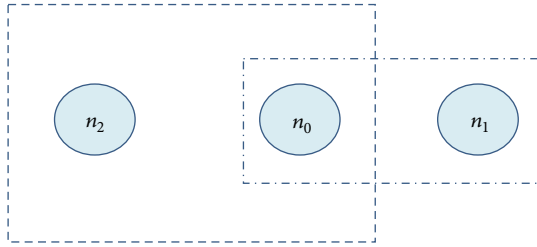
Before taking median, we first, age all values within set  $A_i$  with (20). Now we can define weight adaptation at node  $p$  in Algorithm 1.

$c$  (line (3)) is the learning rate and  $\eta$  (line (4)) is the momentum parameter [38]. The momentum parameter helps

```

(1) AdjustWeights( $\xi(F_p)$ ) {
(2)    $\forall i \in \{1 \dots n\}$ 
(3)      $\Delta w_i = c \times \xi(F_p) \times X_i$ 
(4)      $w_{i(\text{new})} = \eta \times w_{i(\text{old})} + \Delta w_i$ 
    We normalize weights for their sum equal to 1
    as following [45]:
(5)    $\forall i \in \{1 \dots n\}$ 
(6)      $w_i = \frac{w_{i(\text{new})}}{\sum_{j=1}^n w_{j(\text{new})}}$ 
(7) }

```

ALGORITHM 1: Weight adaptation at node  $p$ .FIGURE 3: Mobility scenario for weight adaptation at node  $n_0$ .

to improve convergence time of the algorithm and it is usually set to 0.9, as described in [39], by default. The delta change in weight of the  $i$ th attribute is first derived by multiplying the corresponding value with the *error* and *learning rate* (line (3)). Then the new weight is calculated by adding this delta change into old weight value at line (4) as defined in [38]. We have first described in section that in our case the sum of all the corresponding weights must be equal to 1. Therefore, after every update, weights are again normalized to limit the values within the 0-1 range (line (6)).

In order to show how learning rate affects weight adaptation at a node, let us take the example of a scenario shown in Figure 3. The two boxes show that  $n_0$  has different kind of mobility patterns with  $n_1$  and  $n_2$ . In Figure 4 weight adaptation for different attributes at  $n_0$  is shown against taking different learning rate values. Here,  $W_D$ ,  $W_B$ ,  $W_E$ , and  $W_M$  are corresponding weights of Estimate of Contact Duration, Estimate of Free Buffer Space, Estimate of Rate of Encounter, and Estimate of Meeting Likelihood, respectively,

at node  $n_0$ . We can see that when learning rate is small, there is little variation from the original value at each iteration, and the weights are slowly adapted. Here, it is pertinent to mention that by considering the median value among the available information a sudden significant change in weights can be avoided. However, the weights will be slowly adapted to reflect towards new environment around the node. A node obsolesces information older than the given time window, thereby, we can safely opine that weight adaptation can take no longer than  $\Delta T$  to reflect changing network dynamics.

**3.4. Aging Mechanism.** Opportunistic networks are known to have dynamic typologies and unpredictable moving patterns. Therefore, information regarding mobility behavior of a node recorded at some time instance in past may no longer be valid to guide forwarding decisions. Consequently, it is required to age next-hop utilities on regular intervals in order to reflect onto changing behaviors present in the network and avoid inconsistent decisions. For this reason, we use the same aging mechanism as used in [6] for eliminating stale information from the network. Therefore,

$$F_{(p,q)\text{new}} = F_{(p,q)\text{old}} \times \gamma^k, \quad (20)$$

where  $\gamma$  is the aging constant having value between  $[0, 1]$  and  $k$  is the time units elapsed since last aging of the fitness value is done.

**3.5. Transitive Next-Hop Fitness Calculation.** Grasic et al. [6] suggest the effectiveness of transitive property of a routing scheme for opportunistic networks. In many scenarios it is possible that some nodes may never had a direct encounter; one of the examples of such a network can be presence of multiple disjoint communities with only few nodes visiting different regions [6]. In such scenarios, transitive calculation of next-hop fitness helps in forwarding packets towards destination through multiple intermediate hops. However, we believe that transitive next-hop fitness of a node should be comparatively low as compared to direct next-hop fitness values. Next, we define transitive next-hop fitness calculation in MARS with the help of following example scenario:

$$S \longrightarrow B \longrightarrow M \longrightarrow D. \quad (21)$$

Here  $B$  calculates its fitness to  $D$  through  $M$  as follows:

$$F_{(B,D)\text{new}} = \begin{cases} \left[ \frac{\min \{F_{(B,M)}, F_{(M,D)}\}}{2} \right] & \text{If previously calculated through } M \\ \max \left\{ F_{(B,D)\text{old}}, \left[ \frac{\min \{F_{(B,M)}, F_{(M,D)}\}}{2} \right] \right\} & \text{Otherwise.} \end{cases} \quad (22)$$

The value is divided by 2 at each hop in order to incorporate the effect of uncertainty in opportunistic networks and presence of multiple intermediate forwarding

nodes. The node that is multiple hops away from destination will have small fitness value due to division at each hop.

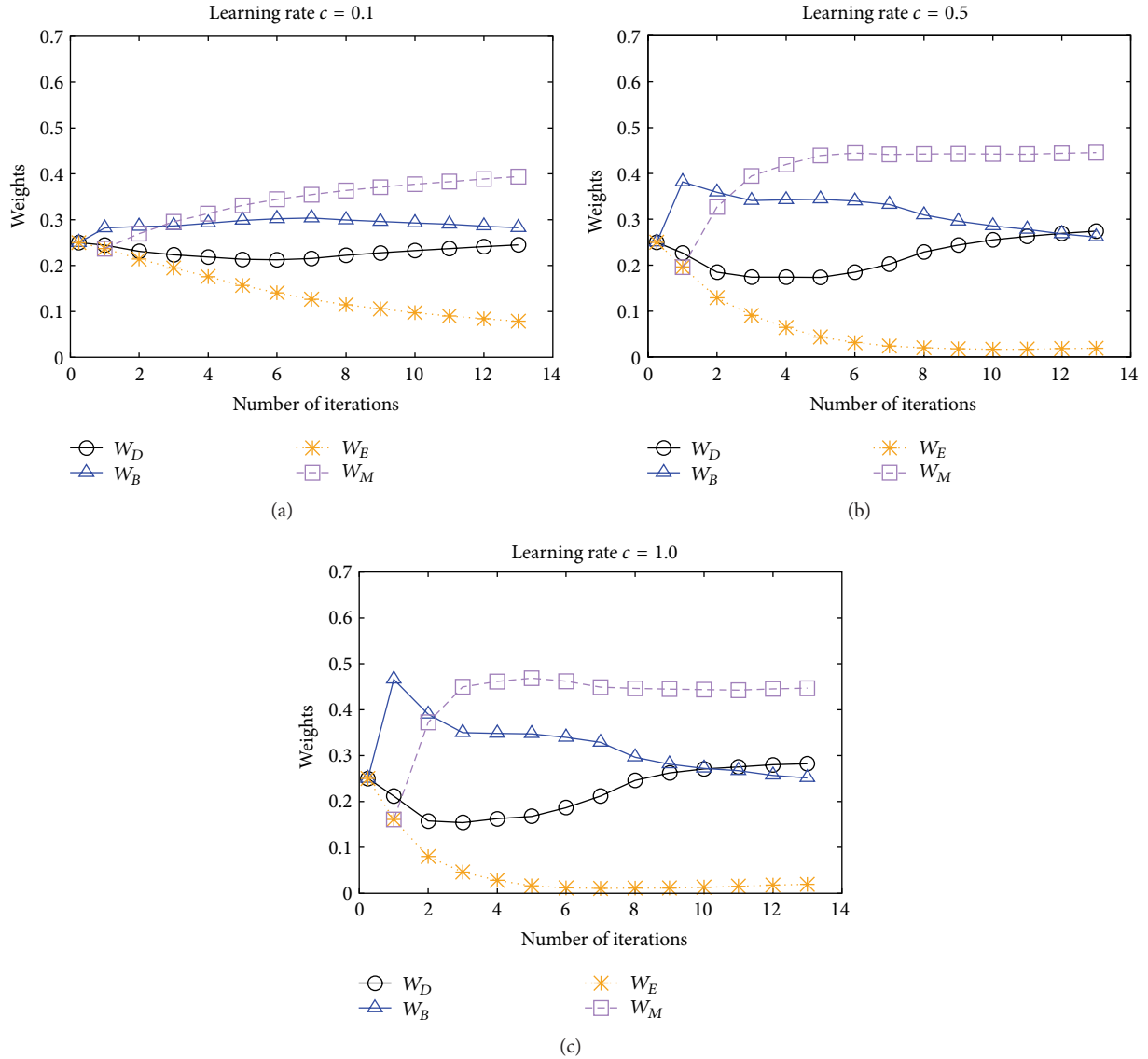


FIGURE 4: Effect of changing learning rate “ $c$ ” on weight adaptation at node  $n_0$ .

**3.6. Forwarding Policy.** We associate a forwarding threshold with every message on a relay node including source, which is initially set to next-hop fitness value of the current custodian to message’s destination. Whenever a copy of the message is forwarded, its associated threshold is updated to next-hop fitness of peer. The forwarding threshold of a message ensures that redundant copies should only be sent to nodes having greater next-hop fitness to destination than the nodes to whom it is previously sent. In this way, we can reduce overhead over the network by restricting message replication only to nodes, presenting themselves as better candidates than the previously selected relays. Moreover, a message threshold is aged using the same mechanism as defined in (20) before comparing with peer’s utility value. Further, if a node is connected to two or more nodes at a time, a message is only sent to a node which presents the highest next-hop fitness to destination among currently connected peers.

Moreover, with MARS, we can assign different number of forwarding tokens ( $L$ ) in order to limit number of replicas of a message. When no limit is defined, it keeps on replicating the packet to nodes presenting next-hop fitness to destination until (1) message lifetime is expired, (2) it is dropped due to buffer overflow, or (3) it is offloaded to destination node. In following we define behavior of MARS with different number of forwarding tokens.

**3.6.1. Single Copy Forwarding.** MARS turns into forwarding scheme when only single token is assigned. In this way, every MARS node relinquishes the custody of the message and removes it from its buffer on forwarding it to first encountering node, presenting next-hop fitness to destination higher than the node itself. In this way, only a single copy of the message exists within the network. Although this approach affects performance, simulations show that MARS



is able to achieve comparable delivery rates with single copy forwarding as compared to existing routing scheme.

**3.6.2.  $L$ -Copy Forwarding.** When  $L > 1$ , every node sends  $\lceil L/2 \rceil$  copies of a message to peer whose next-hop fitness is greater than the node itself and corresponding forwarding threshold of the message; this is analogous to how  $L$  copies are forwarded in Spray and Wait binary mode [5]. However, here, message transferring is based on utility of a node. When a node is left with only one copy of a message it is forwarded according to single copy forwarding.

#### 4. Load-Balancing and Choosing Number of Copies ( $L$ ) with MARS

Destination-dependent utility-based schemes imply that messages might be forwarded towards destination through few “better” nodes presenting high next-hop fitness to it based on given utility criteria. Intuitively, it can lead to significant load imbalance in opportunistic networks which can degrade performance in resource-stringent environments. The uneven load distribution results in poorer utilization of total network capacity; secondly, it can quickly deplete resources (e.g., battery drainage) in heavily utilized nodes, which are, ironically, the most vital for long term functioning of the network.

We can extrapolate the effectiveness of using multiple parameters with MARS through the simplistic scenario shown in Figure 5. Consider a network of five nodes with two sources and one destination node. Both sources and destination nodes are stationary while  $n_1$  and  $n_2$  move between source and destination locations. If we consider the case where source decides next-hop nodes based only on number of encounters with destination, then both  $S_1$  and  $S_2$  will send all their packets to  $D$  through  $n_1$ , shown with solid lines in Figure 5, because of high encounter rate between  $n_1$  and  $D$ . In this way,  $n_1$  can get congested over some time resulting in message drop due to buffer overflows, while resources available at  $n_2$  are never used. Considering the given scenario, if fitness of a node is based on two parameters such as encounter rate and free space instead of considering only a single attribute, then  $n_2$  node can also be used as intermediate relay by  $S_1$ , due to available free buffer space when  $n_1$  get congested, shown with dotted line in Figure 5. Some studies such as [8, 11] address the issue of load-balancing in opportunistic networks. However, the clear distinction we intend to make here is that the intrinsic nature of MARS, based on using multiple parameters for calculating next-hop fitness of nodes, secondly, dynamic determination of relative importance of each parameter with adaptive weights, helps in effective utilization of network capacity with better load-balancing.

Moreover, the flexibility of allocating number of copies ( $L$ ) for a message to be used with MARS can substantially reduce unnecessary transmissions within the network in many scenarios. From this, we can say that selecting a high value of  $L$  to achieve acceptable delivery ratios (network performance) inevitably impact resource utilization. It is reminiscent that minimizing the required copies ( $L$ ) may

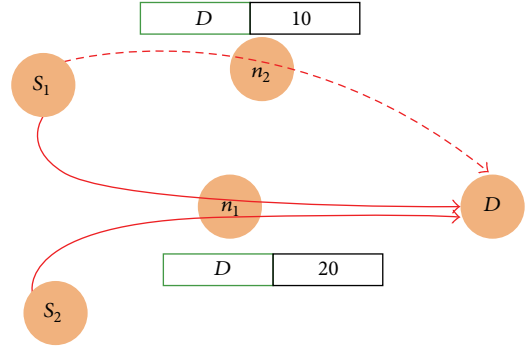


FIGURE 5: Example scenario showing effectiveness of using multiple parameters in path selection.

serve to induce traffic loads; a given network is capable of handling without impacting its performance.

MARS is a destination-dependent utility-based scheme that spread  $L$  copies analogous to binary Spray and Wait routing [5]. Nevertheless, when a node is left with only one copy of a message it is forwarded according to single copy forwarding as discussed in Section 3.6. In destination-dependent utility schemes, every custodian of a copy is leveraged to make an educated decision, based on next-hop fitness of a node, to bring forward the packet to destination. Inevitably, every extra copy serves to cope with high packet loss ratios and possibility of wrongly selecting a next-hop relay due to limitations in the given utility function.

Assume  $R$  is the delivery ratio that can be achieved through single copy forwarding with MARS, and  $R_{\min}$  is the minimum delivery ratio required by an application. Let  $\eta$  is the ratio of error in calculating next-hop fitness of nodes. Let  $\ell$  be the packet loss ratio of the network due to congestion at nodes or long term partitions within the network. Then optimization of  $L$  with MARS is expressed in

$$1 - (1 - R)^L \geq R_{\min}. \quad (23)$$

We adapt the model presented in (23) from [23]. Here, in our case  $R$  can effectively represent magnitude of  $\eta$  and  $\ell$  within the network, as only single copies are sent in the network to survive through error and loss ratios and are finally delivered to eventual destinations.  $R$  might be close to  $R_{\min}$  if  $\eta$  and  $\ell$  are significantly low, and vice versa. From here, we can imply that in any utility-based controlled replication scheme,

$$L \propto (\eta \times \ell). \quad (24)$$

Here, we argue that  $\eta$  and  $\ell$  significantly depend on the utility protocol that is in use to route messages. Effective load-balancing to prevent message drops due to overloading available resources and, secondly, intelligent calculation of next-hop utility of nodes considering multiple aspects at a time can substantially reduce these ratios. Hence, delivery cost ( $L$ ) to achieve certain delivery ratio is also reduced. We defer the question of selecting optimal copies ( $L$ ) with MARS for future work; nevertheless, copies at intermediate hops can be dynamically adapted base on a recent study [37]

where it is selected based on remaining life-time of a message. However, here we assume that the number of copies is set by the application. We have done simulations taking a very low value of  $L$  and simulation results greatly help to elucidate our point that determining next-hop utility of a node in opportunistic networks is a multiobjective function. Moreover, intrinsic benefits of defining a multiattribute utility function along with dynamic adjustment of corresponding weights are twofold; first, it results in educated decisions regarding message carriers; second, it results in effective load-balancing by preventing overloading of precious network resource.

## 5. Other Considerations

As the opportunistic routing schemes tend to make decisions based on an increased knowledge base about the surrounding network, many considerations are needed to be made regarding the involved security and privacy issues along with computational overheads. We intend to address these concerns related to our proposed utility scheme (MARS) in this section.

In this work, we assume that the nodes behave in an altruistic manner and do not intend to maliciously deviate from defined protocol rules. In MARS, calculation of next-hop fitness is based on the principle of mutual interest, and a node is only required to share its utilities with peer in order to help with forwarding messages, while information regarding resources and mobility remains encapsulated. Moreover, to calculate next-hop fitness for peer a node calculates estimates of its own resources and mobility related information, that is, common to both nodes, such as, their encounter rates and interencounter times. Secondly, to calculate transitive fitness through peer, nodes only share their next-hop utility tables, which are also used in taking forwarding decisions, as discussed earlier. Thereby, a node's mobility details with other network participants remain hidden from peer. This helps in ensuring node privacy to facilitate cooperation among nodes. However, it is possible that a node disseminates maliciously defined high utility values in order to attract network data and later drops messages to launch black hole attack. This kind of attack may severely affect single copy variant of MARS, but multicopy MARS routing can potentially overcome this issue, as one of the copies may still be able to reach the destination through benign relays. Handling such attacks independent of the given routing protocol in opportunistic networks is recently discussed in many studies [40, 41]. However, it is pertinent to mention here that in MARS taking the minimum of two values while calculating transitive next-hop fitness as in (22) can avoid polluting the utility tables of neighboring nodes.

Time and space complexity are other two important considerations to be made for deployment in real world situations. Time complexity of calculating an individual score when there are  $n$  given samples in MARS is not more than  $\mathcal{O}(n)$ . Secondly, taking the median of an  $n$  sized list according to Blum et al. [42] renders the worst case time complexity for weight adaptation defined in Algorithm 1 to be  $\mathcal{O}(n)$ . Hence, we can safely say that the time complexity of MARS is  $\mathcal{O}(n)$ . Coming to space complexity of MARS,

let us say the sample size, for making an estimate on score of any given attribute, such as contact-duration, is  $n$  within  $\Delta T$  time. As we know, MARS involves multiple parameters that can be categorized into destination-independent and destination-dependent attributes. Certainly, for any  $l$  destination-independent parameters, time complexity is defined as  $\mathcal{O}(l \times n)$ . However, for destination-dependent attributes it also depends on the number of unique interacting nodes, that is,  $(m)$ . Thereby, for any  $k$  destination-dependent parameters space complexity can be defined as  $\mathcal{O}(m \times k \times n)$ , where  $m$  can grow up to the size of the network. Hence, the worst case space complexity of MARS is  $\mathcal{O}(m \times k \times n)$ .

## 6. Evaluation and Results

For the evaluation of the protocol we use ONE simulator [43]. We compare MARS with well-known routing protocols such as PRoPHET [6], RAPID [24], EBR [10], SCAR [30], and direct-delivery (DD) routing. For PRoPHET and EBR we use suggested settings in [6] and [10], respectively. In PRoPHET and MARS time unit for aging fitness values is set to 30 seconds that is default value suggested in [6]. We assume full battery at all nodes. Therefore, we set battery level to 1 in SCAR for all the scenarios. In SCAR exchange threshold is set to 0.1 as used in default settings of the simulator.

We set  $\Delta T$  equal to 1 hr in scenario-1 and scenario-3, while it is set to 6 hrs in scenario-2 to cope with high expected delays in real environments. Learning rate  $c$  is set to 1 considering heterogeneous mobility patterns in all scenarios. All parameters are initially assigned equal weights by default, that is, 0.25, as there are four attributes which we consider in this paper.

Performance metrics we use to compare MARS with other schemes are defined below. Here  $M$  is total number of delivered messages and  $M'$  is total number of undelivered messages:

$$\text{Delivery Ratio} = \frac{\text{msgs\_delivered}}{\text{msgs\_created}}, \quad (25)$$

$$\text{Overhead Ratio} = \frac{\text{msgs\_relayed} - \text{msgs\_delivered}}{\text{msgs\_delivered}}, \quad (26)$$

$$\text{Average Latency} = \frac{M\_latency + M'\_latency}{\text{msgs\_created}}, \quad (27)$$

where

$$M\_latency = \sum_{i=1}^M \text{receive\_time}_i - \text{creation\_time}_i, \quad (28)$$

$$M'\_latency = M' \times \text{msgTTL}.$$

*6.1. Scenario-1: Simulations with Community-Based Mobility Model.* In this we take 3 communities within 800 m  $\times$  400 m area. In each community 10 nodes are placed which move with RandomWaypoint mobility model, around 5 attraction points. The mobility model is shown in Figure 6. We assume Bluetooth devices with 10 m range and 250 Kbps

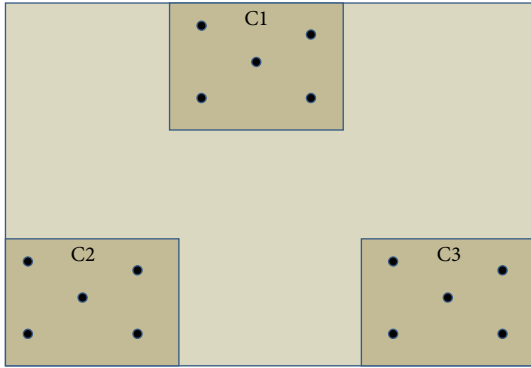


FIGURE 6: Scenario-1: community based mobility model.

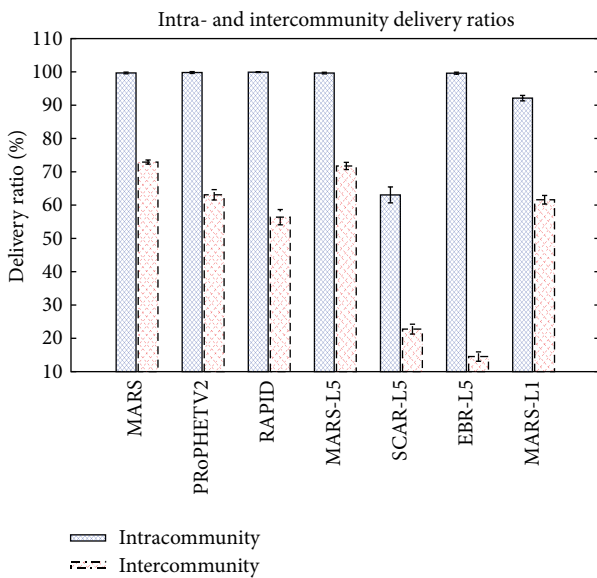


FIGURE 7: Scenario-1b: comparison of intercommunity and intra-community delivery ratios.

bandwidth. Buffer size of nodes within the communities is set to 50 MB while buffer size of traveling nodes is set to 100 MB. Simulation time is set to one day with message TTL of two hours. Message size is set to 2 MB, and we run simulations 10 times, each time using different seeds for nodes in each community and traveling nodes. Traffic load is set to 40 msgs/hr with random source and destination pairs unless it is stated. Similarly, six traveling nodes move across different communities unless where it is stated.

**6.1.1. Effect of Number of Traveling Nodes.** In Figure 8, we investigate the effect of number of traveling nodes across different communities. It is somehow correlated with the level of connectivity within the network. By increasing number of nodes that can travel across communities, the possibility of successful communication between two nodes residing in two different communities also increases. We can see that MARS clearly outperforms other protocols in terms of delivery ratio (Figure 8(a)) and average latency (Figure 8(c)).

Careful selection of next-hop nodes in MARS also helps in maintaining reasonable overhead, which is significantly smaller than PROPHET and RAPID routing as can be seen in Figure 8(b). Overhead of MARS-L1 and MARS-L5 is higher than SCAR-L5 and EBR-L5 due to increased number of intermediate hops, between a source destination pair, used by the protocol. However, it keeps on decreasing with increased connectivity within the network. Analogously, as expected, average latency of protocols (Figure 8(c)) also decreases with increased connectivity. SCAR-L5 introduces very small routing overhead and also has almost asymptotic behavior of delivery ratio curve on increasing traveling nodes. From here, we can deduce the effectiveness of calculating transitive next-hop fitness of nodes, by MARS and PROPHET, in highly heterogeneous mobility scenarios. MARS-L1 is able to achieve comparable delivery rates than PROPHET routing. Hence, the gap between delivery ratios of MARS and its variants MARS-L5 and MARS-L1 represents that replication introduced by the protocol effectively contributes in increasing performance of the network. Hence, MARS, with the ability to calculate utility based on multiple attributes along with adaptive weighting mechanism, is better able to adapt itself to changing mobility patterns to improve network performance. In some scenarios it is possible that the source forwards all the copies of a message before directly encountering the destination node, which may drop due to congestion at intermediate hops. Therefore, we can see high delivery rates with direct-delivery routing as compared to SCAR-L5 at some points on Figure 8(a).

Figure 7 shows comparison of intercommunity and intra-community delivery ratios achieved by different protocols when there are with 10 nodes that can travel across different communities. Here, it is clear that MARS and MARS-L5 have highest intercommunity delivery rates despite smaller overhead as can be seen in Figure 8(b). Hence, we can deduce that MARS is better able to identify heterogeneous mobility patterns within the network.

**6.1.2. Effect of Traffic Load.** It is evident in Figure 9 that MARS is able to sustain high delivery rates (Figure 9(a)) with increasing traffic loads despite small overhead than PROPHET and RAPID routing (Figure 9(b)). RAPID has slightly high delivery ratio than MARS at low data rates but the protocol is unable to cope with high traffic demands due to large overhead. Introducing limited redundancy in case of MARS and MARS-L5 helps in avoiding large message drops resulting in increased performance in terms of delivery ratio and end-to-end latency (Figure 9(c)) in resource-stringent environments. EBR-L5 and SCAR-L5 also introduce small overhead as shown in Figure 9(b) but at the cost of high end-to-end latency (Figure 9(c)) and smaller delivery rates as compared to MARS-L5. Here, we can see that even if the number of forwarding tokens is not associated with messages, MARS does not overload network resources due to careful selection of next-hop relays. The effectiveness of next-hop fitness calculation of MARS is more evident at high data rates as MARS-L1 has similar delivery ratio as compared to PROPHET, while its delivery rate is significantly higher

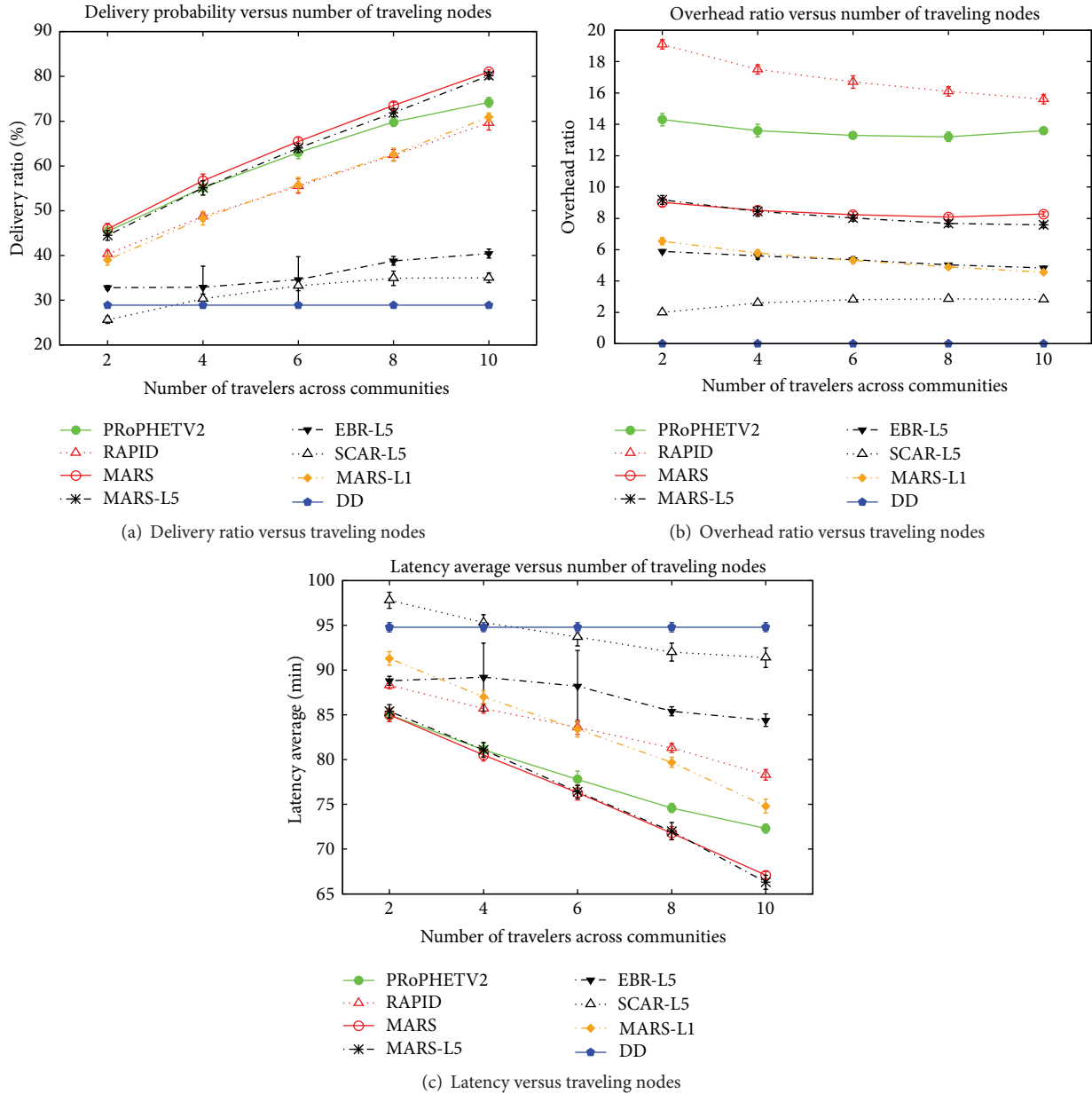


FIGURE 8: Scenario-1a: comparison of MARS with different number of forwarding tokens, PRoPHET, RAPID, EBR, and SCAR routing against varying traveling nodes across communities in community based mobility scenario.

than RAPID routing. In Figure 9(b) high overhead of MARS-L1 compared to SCAR-L5 and EBR-L5 is due to messages travelling through an increased number of intermediate hops.

6.1.3. *Effect of Node Density.* In this scenario, there are six traveling nodes, while we increase number of nodes within each community. When we increase number of nodes, the amount of connectivity or chances of frequent encounters between two nodes also increases within the community. With increased density overhead of unlimited replication schemes, that is, PRoPHET and MARS, also increases proportional to number of nodes presenting next-hop fitness to potential destinations. It also increases in case of RAPID

routing due to increased number of contact opportunities within the network as can be seen in Figure 10(b). However, we can see that MARS does not greedily occupy increasing network resources, while it still maintains high delivery ratios shown in Figure 10(a). Overhead also increases due to messages traveling through increased number of intermediate hops before they are finally delivered at eventual destination as can be seen in case of MARS-L5 and MARS-L1. Decrease in end-to-end latency can also be observed in Figure 10(c) due to increasing delivery rates and because of chances of somewhat stable paths in dense environments.

Results with RAPID routing at 25 nodes/community are not included due to very large computational time required

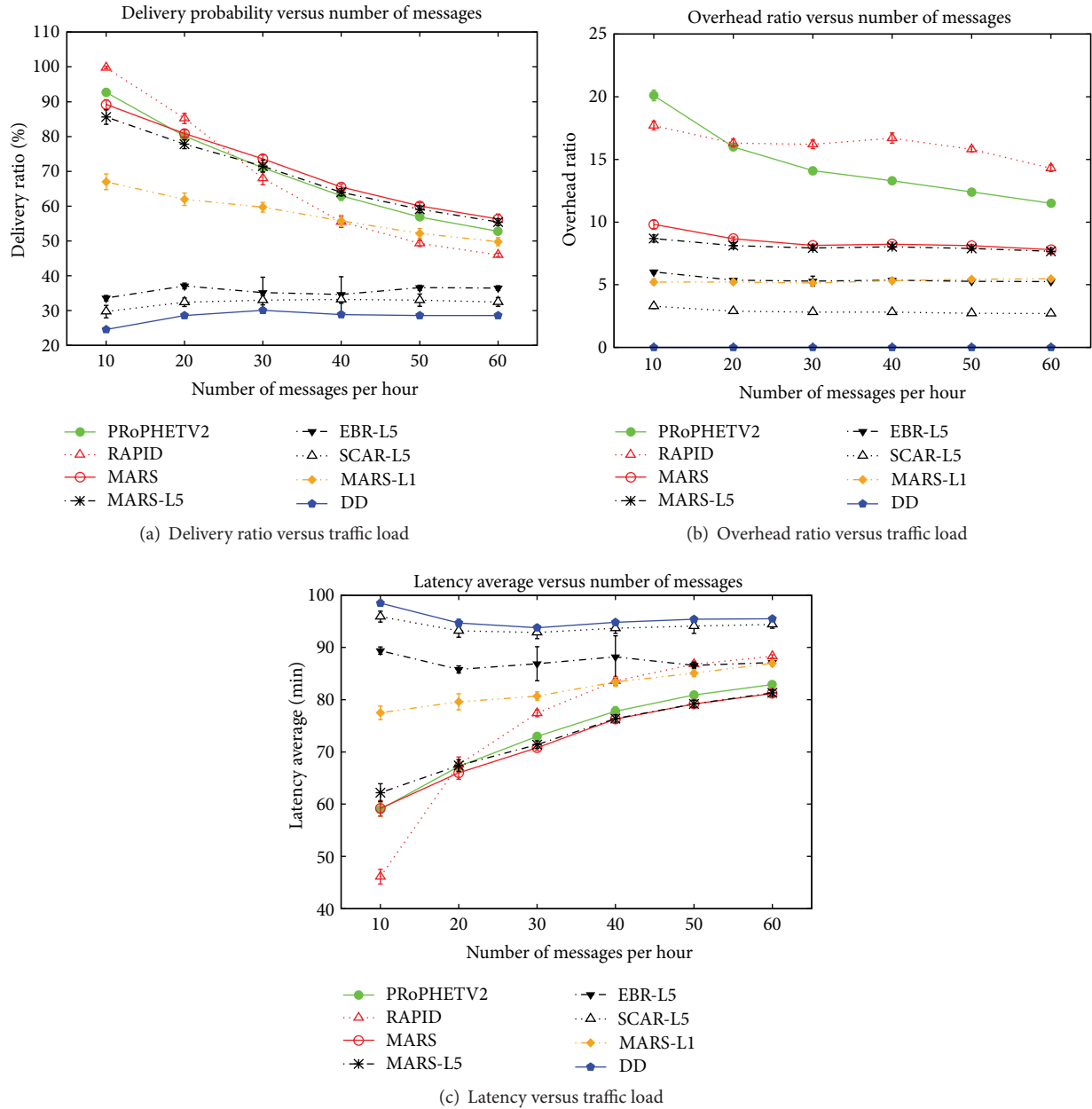


FIGURE 9: Scenario-1c: comparison of MARS with different number of forwarding tokens, PRoPHET, RAPID, EBR, and SCAR routing against varying traffic loads in community based mobility scenario.

by the protocol. As we can see that MARS-L5 shows consistent delivery rates as compared to MARS; hence, a trade-off can be achieved between acceptable delivery ratios and introduced routing overhead with our scheme, when there are limited resources within the network.

6.2. Scenario-2: Simulations with Real Traces. To evaluate the performance of our scheme on real test-bed traces we used connectivity and traffic traces of N4C deployments in 2010. We thank Grasic et al. [6] for sharing these traces and message creation patterns with us. The N4C project was aimed to deploy DTN systems for providing Internet services

to remote areas in Swedish; details about the project can be seen at [44]. These traces are collected from 18 DTN nodes and time duration of these traces is 56 days. One helicopter flight with data mules is scheduled every day and message lifetime is set to 3 days. After analyzing individual interencounter times among node pairs within the given N4C mobility traces, we observe that most of the contacts occur within less than 6 hrs time. In this regard, the interencounter CCDF graph shows a very few percentages more than 12 to 13 hours, where the largest intercontact time is observed to be of 23 days. Therefore, to safely have enough observations, we have taken  $\Delta T$  equal to 6 hrs in this scenario.

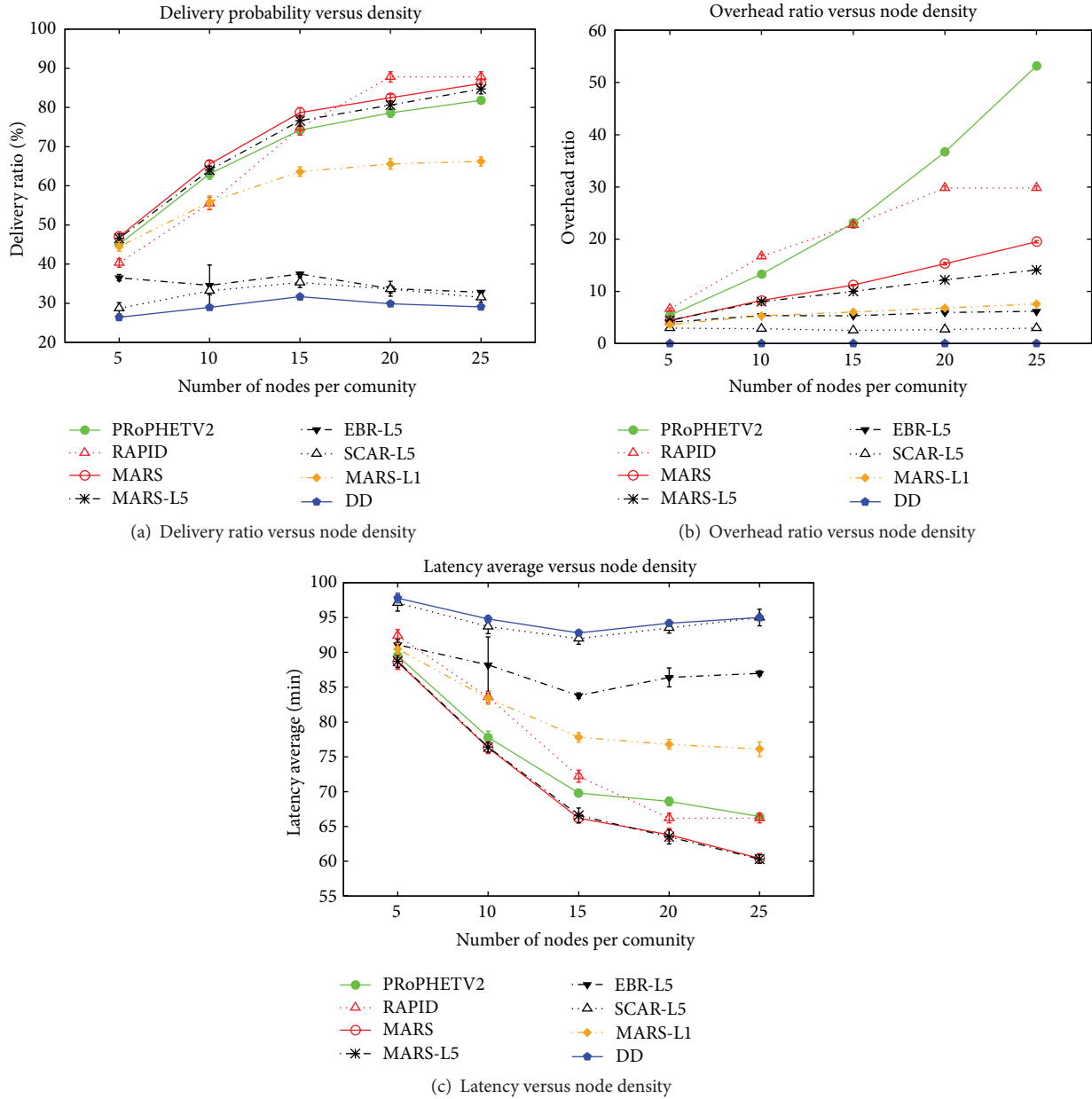


FIGURE 10: Scenario-1d: comparison of MARS with different number of forwarding tokens, PRoPHET, RAPID, EBR, and SCAR routing against varying node density in different communities in community based mobility scenario.

6.2.1. *Effect of Buffer Capacity.* In Figure 11 our scheme is compared with other routing schemes against varying buffer sizes with message size fixed to 2 MB. By avoiding congestion due to small routing overhead and intelligent selection of next-hop nodes, delivery ratio of our scheme (MARS and MARS-L5) is higher than all other schemes at small buffer sizes (Figures 11(a) and 11(b)). With increase in buffer size at nodes, capacity of the network to store extra redundant copies is also increased. Therefore, we can observe an increase in delivery ratio, in Figure 11(a), in case of PRoPHET routing, but at the cost of significantly large routing overhead as can be seen in Figure 11(b). Overhead of PRoPHET and RAPID routing in Figure 11(b) keeps on decreasing with

increase in buffer size due to less number of message drops but it is still higher than MARS routing. Delivery rate of EBR and SCAR with 5 tokens almost remains constant at large buffer sizes while, in MARS-L5, due to the ability of scheme to learn changing network behavior, it keeps on increasing with increase in buffer size of network nodes. Despite introducing limited redundancy within the network MARS and MARS-L5 have smaller end-to-end latency as compared to other protocols in Figure 11(c). Chances of congestion at intermediate nodes can be avoided by keeping the overhead at a constant rate. From here, we can infer that better performance can be achieved by introducing limited redundancy into the network, which can survive message

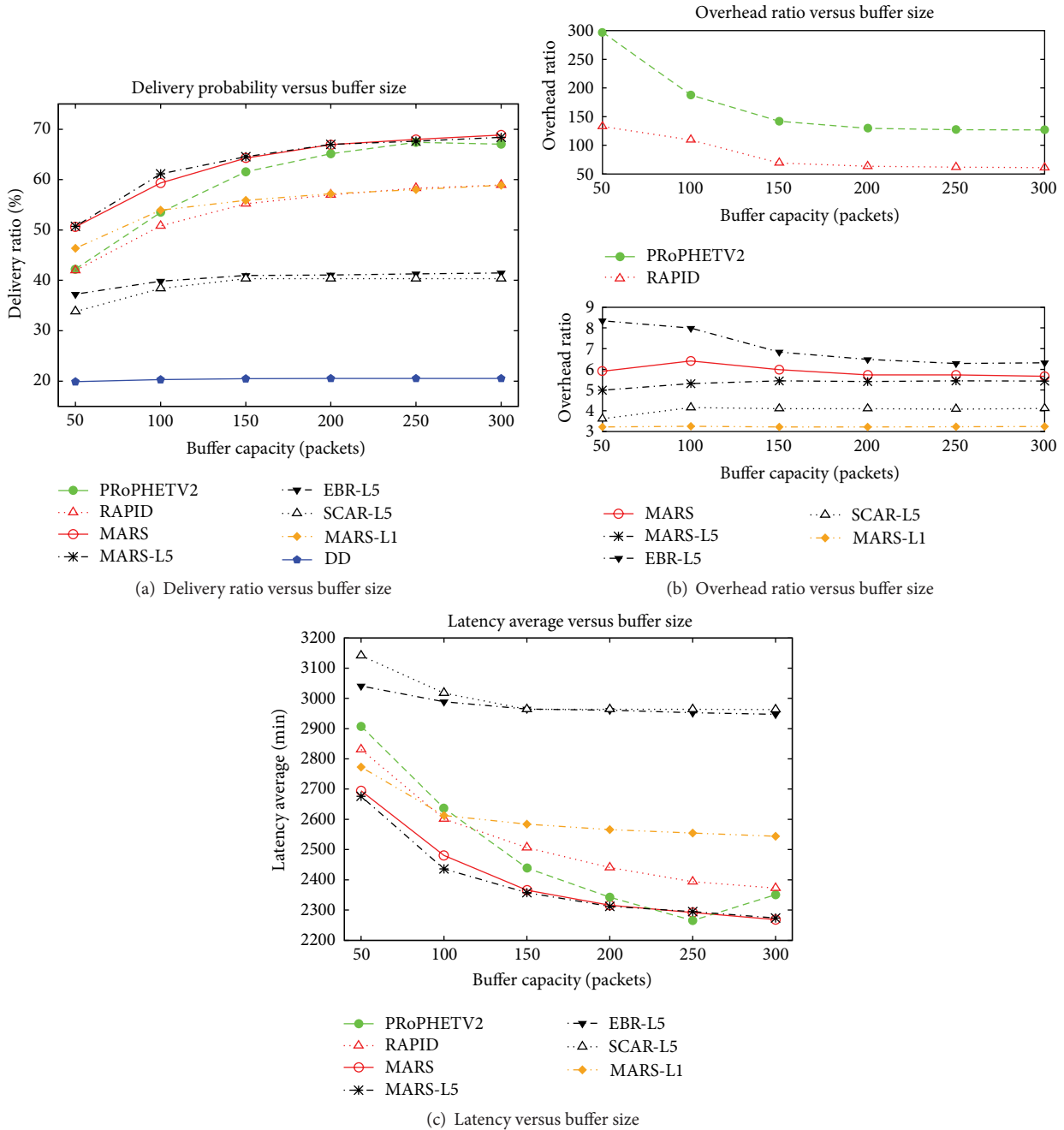


FIGURE 11: Scenario-2: comparison of MARS with different number of forwarding tokens, PRoPHET, RAPID, EBR, and SCAR routing against varying buffer sizes in N4C real traces.

drops due to buffer overflows. This is the reason why we can observe higher delivery ratio in case of MARS-L5 as compared to MARS at low buffer sizes.

6.3. Scenario-3: Simulations with Cluster-Based Movement. In scenario-3 we use ClusterMovemnet Model [43] with default settings in ONE simulator for evaluating MARS with other schemes. Simulation time is set to 12 hrs and there are 160 nodes divided into 4 different groups within the network. There are three clusters of nodes and 40 ferries that move

around these clusters to carry data traffic. We run simulations against five different seed values. Messages of size within range 50 k and 150 k are generated after every 25 to 35 seconds randomly choosing source destination pairs. Message lifetime is set to 5 hours. Results against this movement model are shown in Figure 12. Results with RAPID routing are not included in this scenario due to indefinite computational time required by ONE implementation of the protocol.

SCAR-L5 has negligible overhead but with high end-to-end latency and delivery ratio slightly less than MARS and

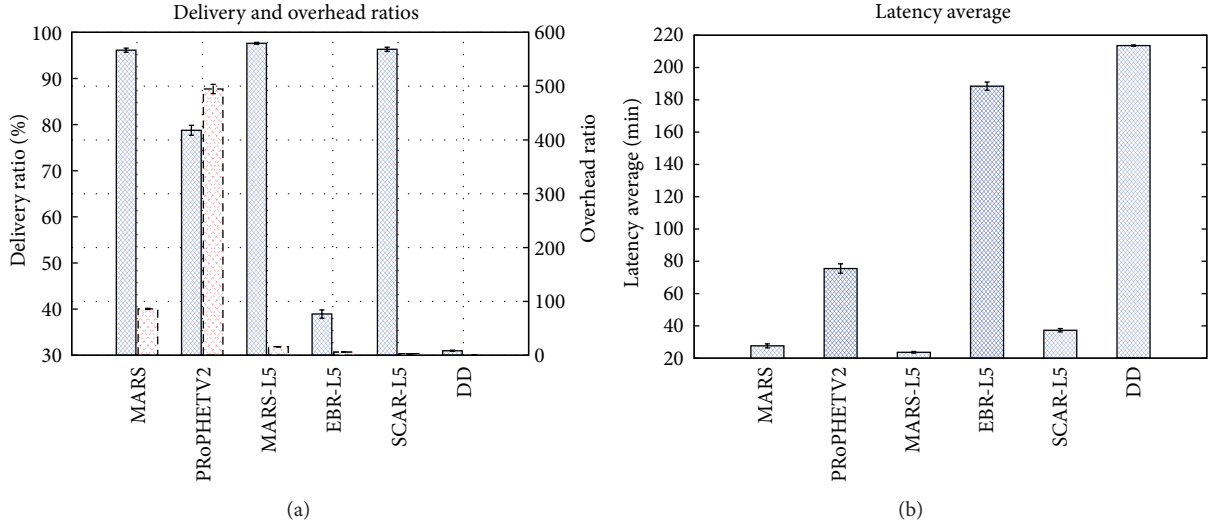


FIGURE 12: Scenario-3: comparison of MARS with different number of forwarding tokens, PRoPHET, EBR, and SCAR routing in cluster-movement based mobility model.

MARS-L5 in this case as shown in Figures 12(a) and 12(b), respectively. Overhead of unlimited replication schemes that is PRoPHET and MARS is high due to high density within the network as a large number of nodes present higher next-hop fitness to potential destinations. However, MARS is still able to limit its overhead to a significantly lower value than PRoPHET routing.

MARS-L5 outperforms MARS in this case. It has slightly higher delivery ratio than MARS despite introducing limited redundancy, and it still has end-to-end latency comparable to MARS. Hence, it further strengthens our point, and introducing limited redundancy into the network by taking wiser decisions helps in achieving better performance than trying to occupy network resources through greedy replication.

Although only 5 replications of each message are allowed with MARS-L5, but as we have stated that when only single copy is left with a node, MARS turns into forwarding scheme. Hence, some copies may keep on moving within the network until their lifetime is expired. Each transmission contributes in increasing routing overhead. Therefore, overhead of MARS-L5 is higher than 5 in Figure 12(a), although there are only 5 replicas for each message present within the network.

## 7. Conclusion

We present a hybrid utility protocol MARS, in this paper, to support communication in heterogeneous opportunistic networks. MARS considers both destination-dependent and destination-independent attributes of a node and employs them in calculating its next-hop fitness. We show, with simulations, using synthetic and real mobility trace, that fitness value calculated in this way can help in taking wiser decisions regarding forwarding of a message towards its eventual destination. MARS is based on SMART as in sparse heterogeneous environments accurate forecasting of future behavior of a node might not be possible. Likewise, we devise a general mechanism to calculate relative score of

each dimension considering samples collected over a finite time interval  $\Delta T$ . Here, we have used empirically defined values of  $\Delta T$ . However, as future work this value can also be dynamically adjusted to configure MARS. In this way, based on learning the contact patterns present within any given network any initially defined value of  $\Delta T$  can be updated accordingly.

We also demonstrate that error correction techniques from neural networks can be effectively applied to determine relative importance of each dimension when we simultaneously consider multiple attributes of a node to calculate delivery probabilities within different 24 networking environments. Moreover, in order to avoid arbitrarily large overhead, number of forwarding tokens can also be assigned with MARS. Due to considering multiple dimensions and real time determination of each dimension's score, MARS can better distribute its overhead which can effectively utilize total network capacity.

Evaluations in variety of heterogeneous environment, such as synthetic and real word mobility traces, show that MARS exhibits very small overhead and still achieves better delivery ratios and end-to-end latency as compared to other protocols taken into consideration. Hence, it can achieve better performance in resource-stringent environments categorized with sparse connectivity.

## List of Frequently Used Symbols

$F_{(p,q)}$ :	$p$ 's next-hop fitness to $q$
$\Delta T$ :	Time window
$\Omega_{p,q}$ :	Set of contact duration samples of node $p$ with node $q$ within time interval $\Delta T$
$\hat{Y}(\Omega_{p,q})$ :	Exponential moving weighted average of contact durations of node $p$ with node $q$
$\hat{S}(\Omega_{p,q}; \Delta T)$ :	Estimate of contact duration at $p$ with node $q$
$\Theta_p$ :	Occupied-buffer space at node $p$



$\widehat{S}(\Theta_p; \Delta T)$ :	Estimate of free buffer space
$\lambda_{p,q}$ :	Encounter count of node $p$ 's encounters with node $q$
$\beta_{p,q}$ :	Encounter count of node $p$ 's encounters with node $q$ within time interval $\Delta T$
$\widehat{S}(\lambda_{p,q}; \Delta T)$ :	Estimate of rate of encounter
$\mathcal{U}_{p,q}$ :	Set of samples of interencounter times between nodes $p$ and $q$ within time interval $\Delta T$
$\widehat{Y}(\mathcal{U}_{p,q})$ :	Exponential moving weighted average of interencounter times between nodes $p$ and $q$
$\widehat{S}(\mathcal{U}_{p,q}; \Delta T)$ :	Estimate of meeting likeliness
$L$ :	Number of tokens.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] T. Clausen and P. Jacquet, *Optimized Link State Routing (OLSR)* RFC 3626, IETF Networking Group, 2003.
- [2] C. Perkins, E. Belding-Royer, S. Das et al., "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, Internet RFCs, 2003.
- [3] V. Cerf, S. Burleigh, A. Hooke et al., *RFC 4838, Delay-Tolerant Networking Architecture*, vol. 2, IRTF DTN Research Group, 2007.
- [4] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep. CS-200006, Duke University, 2000.
- [5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.
- [6] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a DTN routing protocol—PROPHETv2," in *Proceedings of the 6th ACM Workshop on Challenged Networks*, pp. 27–30, ACM, Las Vegas, Nev, USA, September 2011.
- [7] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Max-Prop: routing for vehicle-based disruption-tolerant networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, vol. 6, pp. 1–11, IEEE, Barcelona, Spain, April 2006.
- [8] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in delay-tolerant networks comprising heterogeneous node populations," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1032–1047, 2009.
- [9] A. Elwhishi and P.-H. Ho, "SARP—a novel multi-copy routing protocol for intermittently connected mobile networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–7, IEEE, Honolulu, Hawaii, USA, December 2009.
- [10] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 846–854, Rio de Janeiro, Brazil, April 2009.
- [11] J. M. Pujol, A. L. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 837–845, Rio de Janeiro, Brazil, April 2009.
- [12] W. Edwards, "How to use multiattribute utility measurement for social decisionmaking," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 5, pp. 326–340, 1977.
- [13] H. Badis and K. Al Agha, "QOLSR, QoS routing for ad hoc wireless networks using OLSR," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 427–442, 2005.
- [14] Q. Xue and A. Ganz, "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 154–165, 2003.
- [15] C. Zhu and M. S. Corson, "Qos routing for mobile ad hoc networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 2, pp. 958–967, IEEE, New York, NY, USA, June 2002.
- [16] I. Gerasimov and R. Simon, "Performance analysis for ad hoc QoS routing protocols," in *Proceedings International Mobility and Wireless Access Workshop (MobiWac '02)*, pp. 87–94, IEEE, Fort Worth, Tex, USA, October 2002.
- [17] C. Boldrini, M. Conti, and A. Passarella, "Exploiting users' social relations to forward data in opportunistic networks: the HiBOP solution," *Pervasive and Mobile Computing*, vol. 4, no. 5, pp. 633–657, 2008.
- [18] H. A. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic routing protocol for intermittently connected mobile ad hoc network (PROPICMAN)," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '07)*, pp. 1–6, IEEE, Espoo, Finland, June 2007.
- [19] H. A. Nguyen and S. Giordano, "Context information prediction for social-based routing in opportunistic networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1557–1569, 2012.
- [20] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.
- [21] L. Gao, M. Li, A. Bonti, W. Zhou, and S. Yu, "Multidimensional routing protocol in human-associated delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2132–2144, 2013.
- [22] S. Yasmin, R. N. B. Rais, and A. Qayyum, "A multi-attribute routing protocol for opportunistic network environments," in *Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN '14)*, pp. 1–6, Shanghai, China, August 2014.
- [23] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, 2010.
- [24] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in DTNs: a resource allocation approach," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 596–609, 2010.
- [25] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, 2008.
- [26] F. Xia, Q. Yang, J. Li, J. Cao, L. Liu, and A. M. Ahmed, "Data dissemination using interest-tree in socially aware networking," *Computer Networks*, vol. 91, pp. 495–507, 2015.
- [27] S. K. Dhurandher, D. K. Sharma, I. Woungang, and A. Saini, "Efficient routing based on past information to predict

- the future location for message passing in infrastructure-less opportunistic networks,” *The Journal of Supercomputing*, vol. 71, no. 5, pp. 1694–1711, 2015.
- [28] F. Xiao, X. Xie, Z. Jiang, L. Sun, and R. Wang, “Utility-aware data transmission scheme for delay tolerant networks,” *Peer-to-Peer Networking and Applications*, pp. 1–9, 2015.
- [29] M. Musolesi and C. Mascolo, “Car: context-aware adaptive routing for delay-tolerant mobile networks,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2009.
- [30] C. Mascolo and M. Musolesi, “SCAR: context-aware adaptive routing in delay tolerant mobile sensor networks,” in *Proceedings of the International Conference on Wireless Communications and Mobile Computing (IWCMC '06)*, pp. 533–538, ACM, Vancouver, Canada, July 2006.
- [31] R. L. Keeney, *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*, Cambridge University Press, 1993.
- [32] M. Musolesi and C. Mascolo, “Evaluating context information predictability for autonomic communication,” in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pp. 495–499, IEEE, Buffalo, NY, USA, June 2006.
- [33] R. Cabacas and I.-H. Ra, “A novel energy-aware priority transmission scheme based on context-metric queuing for delay tolerant networks,” in *Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering (ISEEE '14)*, vol. 2, pp. 1095–1099, IEEE, Sapporo, Japan, April 2014.
- [34] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, “Power law and exponential decay of intercontact times between mobile devices,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1377–1390, 2010.
- [35] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, “On the levy-walk nature of human mobility,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 630–643, 2011.
- [36] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*, University of California, Berkeley, Berkeley, Calif, USA, 2002.
- [37] J. Miao, O. Hasan, S. B. Mokhtar, L. Brunie, and G. Gianini, “A delay and cost balancing protocol for message routing in mobile delay tolerant networks,” *Ad Hoc Networks*, vol. 25, pp. 430–443, 2015.
- [38] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan Publishers, Hamilton, NJ, USA, 1994.
- [39] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, 2007.
- [40] D. Pham and C. K. Yeo, “Detecting colluding blackhole and greyhole attack in delay tolerant networks,” *IEEE Transactions on Mobile Computing*, pp. 1–15, 2015.
- [41] N. Li and S. K. Das, “A trust-based framework for data forwarding in opportunistic networks,” *Ad Hoc Networks*, vol. 11, no. 4, pp. 1497–1509, 2013.
- [42] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, “Time bounds for selection,” *Journal of Computer and System Sciences*, vol. 7, no. 4, pp. 448–461, 1973.
- [43] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE simulator for DTN protocol evaluation,” in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools '09)*, ICST, Rome, Italy, 2009.
- [44] Networking for Communications Challenged Communities, 2013, <http://www.n4c.eu/>.
- [45] O. Macchi and E. Eweda, “Second-order convergence analysis of stochastic adaptive linear filtering,” *IEEE Transactions on Automatic Control*, vol. 28, no. 1, pp. 76–85, 1983.