Enhanced fuzzy logic-based spray and wait routing protocol for delay tolerant networks

Sweta Jain¹,*†, Meenu Chawla¹, Vasco N. G. J. Soares² and Joel J. Rodrigues³

¹Maulana Azad National Institute of Technology, Mata Mandir, Department of CSE and IT, Bhopal, Madhya Pradesh India, 462051
²Instituto de Telecomunicações, Superior School of Technology - Polytechnic Institute of Castelo Branco, Av. do Empresário - Campus da Talagueira, Castelo Branco, Portugal, 6000-767
³Instituto de Telecomunicações, University of Beira Interior, Rua Marquês d’Ávila e Bolama, Covilhã, Portugal, 6201-001

SUMMARY

Delay tolerant networks are a class of ad hoc networks that enable data delivery even in the absence of end-to-end connectivity between nodes, which is the basic assumption for routing in ad hoc networks. Nodes in these networks work on store-carry and forward paradigm. In addition, such networks make use of message replication as a strategy to increase the possibility of messages reaching their destination. As contact opportunities are usually of short duration, it is important to prioritize scheduling of messages. Message replication may also lead to buffer congestion. Hence, buffer management is an important issue that greatly affects the performance of routing protocols in delay tolerant networks. In this paper, Spray and Wait routing protocol, which is a popular controlled replication-based protocol for delay tolerant networks, has been enhanced using a new fuzzy-based buffer management strategy Enhanced Fuzzy Spray and Wait Routing, with the aim to achieve increased delivery ratio and reduced overhead ratio. It aggregates three important message properties namely number of replicas of a message, its size, and remaining time-to-live, using fuzzy logic to determine the message priority, which denotes its importance with respect to other messages stored in a node’s buffer. It then intelligently selects messages to schedule when a contact opportunity occurs. Because determination of number of replicas of a message in the network is a difficult task, a new method for estimation of the same has been proposed. Simulation results show improved performance of enhanced fuzzy spray and wait routing in terms of delivery ratio and resource consumption. Copyright © 2014 John Wiley & Sons, Ltd.

Received 9 November 2013; Revised 19 February 2014; Accepted 23 March 2014

KEY WORDS: delay tolerant networks; spray and wait; buffer management; fuzzy logic

1. INTRODUCTION

The proliferation of mobile computing devices (like e.g., laptops, personal digital assistants, and mobile phones) together with the ubiquitous nature of the wireless communication and mobile technologies has started to transform the way of life of people ranging from every aspects of the society [1]. In the context of growing popularity for wireless technology, Mobile ad hoc Networks (MANETs) [2] have gained its own place in terms of its capability for providing improved communication in many fields like tactical networks, sensor networks, disaster recovery, and home networking.

Delay tolerant networks (DTNs), also known as opportunistic networks, are an emerging class of ad hoc networks where network density is sparse and messages can withstand long delays. They are mainly characterized by frequent link disruptions leading to long duration network partitions, sparse network density, limited device capability (such as battery power, buffer space, and processing power), asymmetric data rates, high error rates, and limited bandwidth [3–5]. DTNs find use in
many real life applications such as mobile sensor networks for wildlife tracking, interplanetary communications, underwater networks, pocket switched networks, and vehicular ad hoc networks (VANETs) [5]. These networks use a bundle layer, which is an overlay protocol lying between application layer and transport layer [5]. Bundles are arbitrary size messages consisting of multiple application packets, which are forwarded in store-carry and forward manner.

Conventional ad hoc routing protocols fail to work in such environments as they lack continuous end-to-end connectivity between nodes because of frequent link disruptions. Therefore, such networks make use of store-carry-forward strategy to transfer messages between the nodes, that is, the nodes buffer the messages and transfer it when a contact opportunity arises, and this process is repeated until the message eventually reaches its destination. Similarly, a carry and forward scheme is used in VANETs [6]. This is also somewhat similar to the problem of data dissemination in wireless networks, which has been recently studied in [7], where the main objective is to spread a particular content in a network while guaranteeing a low completion time and also minimizing resource consumption. However, the delays between node meetings in DTNs may be very long varying from hours, days to years, and hence nodes must be capable of storing messages in their buffers for long times. Thus, the problem of buffer management requires special attention in DTNs. Two major issues in buffer management include scheduling of messages when message transmission opportunity occurs and dropping of messages when node buffer overflows.

A lot of research has been performed in design of routing protocols for DTNs, to name a few, Epidemic [8], Prophet [9], Spray-and-Wait [10], MaxProp [11], Spray and Focus [12], and ORWAR[13]. They may be mainly classified as forwarding-based or replication-based. In forwarding-based routing protocols, only a single copy of the message exists in the network. In replication-based routing protocols, multiple copies of the message are created on contact opportunities to increase the delivery probability of the message to its destination. Although replication-based routing protocols may result in improvement in the delivery ratio, they also increase the resource consumption within the network [8, 10].

Replication-based routing protocols are further classified as flooding-based routing protocols, that allow any number of message copies to exist in the network, and quota-based routing protocols where only a limited number of copies of a message are allowed in the network. Quota-based routing protocols aim to achieve high delivery ratios with limited overhead as against flooding-based routing protocols where overheads are very high. Examples of flooding-based routing protocols are Epidemic, Prophet, MaxProp, and RAPID, whereas that of quota based routing protocols are spray and wait and spray and focus.

Delay tolerant networks consist of portable devices such as personal digital assistants, mobile phones, and sensors, which have limited buffer space. However, the requirement of long-term storage and message replication in DTN environments puts a burden on the limited node resources such as buffer space [14]. Moreover, contact duration between the nodes in DTNs is very short mainly because of node mobility and contention [15]. Hence, the assumption of infinite bandwidth is not valid, and it is necessary to prioritize messages while scheduling them. Even with controlled replication routing protocols, it is important to prioritize message scheduling because of short contact durations. Although the problem of scheduling has been studied recently in [16], it mainly deals with multimedia content and has been designed for multi-channel, multi-radio, and multi-hop networks. There ‘scheduling’ to refer to the combined operation of channel assignment, rate allocation, and routing.

In the current work, three important message parameters the number of replicas of a message in the network, message remaining lifetime, and size are used for taking scheduling decision. Importance of these parameters has been described in later sections. These parameters have values varying over different ranges; they have different units; their value is different for different messages. Hence, fuzzy logic (FL) is an effective technique to aggregate these parameters that might be otherwise difficult to formulate mathematically. FL is a powerful problem-solving tool with the ability to deal with conflicting situations and imprecise data using approximate human reasoning, without needing complex mathematical model [17]. It models what human beings have been doing or would do in a given situation. An FL controller is being proposed in this work, which
combines the number of replicas of a message, its remaining time-to-live (RTTL) and its size to decide the scheduling priority of a message. A new method for proper estimation of number of replicas of a message is being proposed, which overcomes the shortcomings of previous methods [18, 19].

The rest of the paper is organized as follows: Section 2 presents a brief survey of the major routing protocol categories of DTN and the different buffer management policies that have been proposed for DTNs in the past literature; Section 3 describes the proposed fuzzy-based buffer management scheme; Section 4 defines the simulation environment used for evaluation of the proposed work and presents the various simulation results; and finally, Section 5 concludes the paper and presents future work.

2. RELATED WORK

Routing in DTNs is a challenging task because of frequent network partitions and limited network resources such as buffer, bandwidth, and energy. Most of the routing protocols in DTN can be categorized as: single-copy, multiple-copy, flooding, utility based, history of encounter, and social routing.

In single-copy routing [20], the protocol only forwards one copy of the message to the node encountered using some forwarding strategy that can vary with each algorithm. In direct delivery, the source node delivers the message directly to the destination node. Although this approach has lower overhead, the delay can be extremely long especially if the nodes have zero-knowledge of the network’s topology. The worst case scenario may occur when the node carrying the data never gets to the range of the final destination and the message never gets transmitted. This scheme, however, would serve the purpose perfectly in case of full knowledge of mobility patterns, that is, if the path of the nodes is highly predictable or even predetermined [21]. The First Contact [20] routing forwards a message to the first encountered node only without any message replication.

In the case of blind flooding, each node transmits all the messages in its buffer to the encountered node except the common messages. The network is flooded with a lot of messages, which requires a high amount of buffer space leading to buffer congestion and high drop rate of messages. In Epidemic Routing [8], at each contact opportunity, nodes exchange summary vectors containing information about messages that are stored in their buffers to identify and request messages that are not present in their buffer. Recently in [22], a method based on adaptive compression of vectors has been proposed to reduce the overhead involved in exchange of summary vectors and request vectors and their storage. A strategy based on epidemic forwarding has been used in [23] for VANETs for data dissemination.

Multiple-copy routing fixes an ‘n’ number of copies of each message, which are then forwarded using a forwarding strategy specific to the protocol. Spray and wait routing algorithm [10], which is a popular multi-copy routing protocol, consists of two phases: the spray phase and the wait phase. There are two main versions of this protocol: source spray and wait and binary spray and wait. In source spray and wait, the source node first spreads n number of copies of the message to the first n encountered nodes. If the destination is not found in this phase, then these n nodes wait for meeting the destination and act as relays to directly deliver the message to the destination. In binary spray and wait, the source node keeps n/2 copies with it and transfers n/2 copies of the message to the first encountered node; this node again keeps n/4 copies with it and transfers remaining n/4 copies to the encountered node. When a node is left with only 1 copy, then it enters the wait phase. Thereby, spray and wait keeps a bound on the level flooding by limiting the number of relay nodes to n. This protocol exhibits a balanced combination of both flooding and direct delivery. Spray and Focus [12] routing algorithm is a variation of spray and wait routing protocol where forwarding in the wait phase is guided by utility functions.

Prophet [9] is a history of encounter-based routing algorithm, which takes information about the past encounters that each node has made with every other node in the network. Because in practical applications, nodes who have met frequently in the past have a larger probability to meet again, such nodes can be considered as good relays for forwarding the messages.

Social-based routing protocols [24, 25] find their significance in application areas like Pocket Switched Networks, also called as mobile social networks [26]. In Pocket Switched Networks, a
multitude of people carry mobile devices. Because the devices are carried by humans who work as a society, a study about their social characteristics may help in selection of better relays for message transmissions. Social-based routing algorithms use social characteristics like community structures, centrality, and similarity defined in sociology for choosing a good quality relay. In [27], authors have used DTN-like store-delegate-and-forward communication model and concept of social awareness among nodes to accomplish peer-to-peer file sharing application in a network composed of mobile smart phones.

Buffer management has a major impact on the performance of routing protocols in DTNs [11, 28–31], hence several buffer management schemes have been proposed. The existing buffer management policies can be broadly classified as policies that are based only on local information about messages, such as their arriving time, age, TTL, and hop-count and size; and policies which are based on the partial or complete network-wide information such as total number of copies of message in the network and meeting rate between two nodes. A number of buffer dropping and scheduling policies have been proposed for the former category and are described first.

Buffer management was first studied in [28] where authors presented four buffer management policies: drop-random, the packet to be dropped is chosen at random; drop-least-recently-received, the packet that has been in the node’s buffer for the longest time is dropped; drop-oldest, the packet that has been in the network for the longest time is dropped; and drop-least-encountered, the packet is dropped on the basis of its estimated likelihood of delivery. Simulation results showed that drop-least-encountered performs better than drop-oldest as it takes into account the meeting probability of nodes while taking packet dropping decision and packet that has least probability of reaching its destination is dropped first.

In [29], performance of epidemic routing has been examined using three traditional buffer management strategies drop-tail, which drops the message at the end of the buffer; drop-head, which drops the message at the front of the buffer; and source-prioritized drop-head, which is similar to drop head but the node gives priority to messages for which it is source and first drops relay packets, then its source packets. Simulation results show that it is beneficial to give priority to source packets.

Performance of Prophet has been assessed using different combinations of queuing policies and scheduling strategies in [30]. Queuing policies are based on different criterion namely, the forward count of the message, drop most forwarded first (MOFO); its lifetime, drop shortest lifetime first; its position in queue, drop message at the front of the queue (FIFO); and its delivery predictability, drop least probability message first (LEPR). MOFO gives best results, whereas LEPR gives worst performance. This is because in LEPR, messages with least delivery probability are dropped first. There may be messages whose destination may be far off and hence its delivery predictability at the source may be small. This may be lead to dropping of such messages at source without being forwarded to other nodes. The work in [30] also presents multiple scheduling strategies that decide the sequence of messages to be transmitted when two nodes meet. They are based on the delivery predictability metric of messages, which is defined in [9]. Let \( P(A, D) \) denote the delivery predictability of node A to meet node D and \( P(B, D) \) denote the delivery predictability of node B to meet node D. Four scheduling policies have been proposed, namely, GRTR, which says to forward the message only if \( P(B, D) > P(A, D) \); GRTRSort, which selects messages in descending order of the value of \( P(B, D) - P(A, D) \) and forwards the message only if \( P(B, D) > P(A, D) \); GRTRMax, which selects messages in descending order of \( P(B, D) \) and forwards the message only if \( P(B, D) > P(A, D) \); and COIN, which forwards the message only if \( X > 0.5 \), where \( X \in U(0, 1) \) is a random variable. From the simulation results, authors concluded that GRTRSort in conjunction with the MOFO queuing policy gives the best performance.

MaxProp [11] prioritizes both the list of packets to be transmitted to other peers and the list of packets to be dropped on the basis of hop counts and path likelihood to nodes, which is calculated using historical data. Prioritized epidemic routing [31] also uses hop count similar to MaxProp to assign priorities to messages. But it also uses path cost to assign drop priority of messages. Hop count only reflects the number of times a message has been forwarded on a particular path and does not give the distribution of message among other nodes, which might have a copy of the message but do not lie on this forwarding path.
Similar to [30], Vasco et al. have also evaluated different combinations of scheduling and dropping policies for vehicular delay tolerant networks in [32, 33]. Message ordering has been considered on the basis of FIFO, random order, ascending remaining lifetime, descending remaining lifetime, ascending replicated copies, and descending replicated copies. From the simulation results, it was concluded that the combination of a scheduling policy and a dropping policy based on descending replicated bundles gives better performance in comparison with other combinations. It outperforms the commonly used FIFO scheduling and ‘drop head’ buffer management. This is mainly because in FIFO scheduling, only the packets at the head of queue are forwarded repeatedly, whereas other packets at the end of queue do not get a chance to get forwarded. This study highlighted the various factors that affect dropping and scheduling decision in delay tolerant networks for vehicular scenarios.

$N$-drop policy [34] takes into account the number of times a message has been forwarded. If a message has been transferred for $N$ number of times, it is dropped when the buffer overflows; here, $N$ is a threshold related to the size of the buffer. The function defining relationship between value of $N$ and buffer size is not well defined.

In [19], enhanced buffer management policy (EBMP) has been proposed, which uses message utility functions based on three message properties: the estimated number of replicas (ETR), the age, and the RTTL of the message.

$$EBMP_{i}^{\text{delivery}} = \frac{1}{ETR_i} + \frac{1}{\log(AGE_i)} \quad (1a)$$

$$EBMP_{i}^{\text{delay}} = \frac{1}{ETR_i} + \log(RTTL_i) \quad (1b)$$

Nodes update their ETR value of messages present in their buffer, when they meet each other, by exchanging their my_forward values; here, my_forward value denotes the number of new replicas of a message created by a node. Although this scheme uses local message properties, simulation results show that its performance is comparable with buffer management policies, which use global message information. However, the major drawback of this scheme was that if two nodes meet frequently, then the method for estimating the number of replicas may result in unnecessarily high value of number of replica, although no new replicas have been created during the subsequent meetings. Suppose two nodes $A$ and $B$ have a common message in their buffer and have ETR value 2 and 3, respectively, of that message and my_forward value as 1 and 2, respectively, for the same message. Then, if $A$ and $B$ meet again and again, they are going to exchange my_forward value of this common message repeatedly and unnecessarily add it to the ETR value. This is explained in detail in Section 3.2.1 a).

A similar type of buffer management scheme MTSBS stands for Message Transmission Status Based Buffer Management Scheme (MTSBS) [35], which is based on estimated status of messages, considers the dissemination speed of a message in addition to the total number of copies of the message in the network to decide message priority. Here, dissemination speed of a message is calculated as the ratio of number of hops that a message has traveled to its RTTL.

In [36], a variety of data object dropping schemes for a data centric network architecture called Haggle have been proposed. They are least interested (LI), that is, drop the data object in which the least number of neighbors are interested in; most interested (MI)—drops the data objects in which most neighbors are interested in; max copies (MAX)—drops the data objects after they have been replicated max number of times; most forwarded (MF)—drop the data object with highest number of replications made at the local node; least forwarded (LF)—drop the data object with the lowest number of replications made at the local node; random—drop randomly chosen data objects from the node. Two extreme cases have also been compared: Infinite buffer, where each node has infinite buffer; and no object needs to be dropped and no buffer, where each node stores only those data objects in which it is interested in. The authors concluded that MF strategy performs the overall best with respect to delivery ratio, delay and overhead at a congested node.

The work in [37], that is, TTL-based routing (TBR) maintains two message lists in its buffer. The forward list, which stores the messages to be forwarded next when a contact opportunity arises, and the delete list, which stores the messages to be dropped when node buffer overflows. The forward
list stores the messages ranked by a forwarding priority \( (P_f) \), which is a function of message size \( (s_k) \), its TTL \( (TTL_k) \) and its hop count \( (h_k) \). The delete list ranks the messages according to dropping priority \( (P_d) \), which is a function of the number of copies \( (L_k) \) and size of the message \( (s_k) \). Moreover, hop count as discussed previously, only reflects the number of intermediate nodes in one possible forwarding path and does not indicate the whole message distribution over the entire network.

\[
P_f = \frac{1}{H_k*TTL_k*S_k} \\
P_d = \frac{L_k}{S_k}
\]

Buffer management policies presented in [38–42] solely use message size for taking drop decision when buffer overflows. Drop largest [38, 42] selects the largest size message to drop when its buffer overflows. Threshold drop [39] drops the message, which lies in the range of predefined thresholds. Equal drop [40] drops the message of equal size from the buffer when the node buffer overflows. Mean-drop [41] computes the mean of size of buffered messages at the node and drops only those message(s) that have the size greater than or equal to this mean value. All these policies only consider the size of message for taking dropping decision, whereas other important parameters, like the number of replicas, its RTTL have been neglected.

In a recent work [43], T. Senttawatcharawanit et al. have proposed a buffer management policy for social DTNs, which utilizes the social relation between sending and receiving node to take dropping decision. This policy makes use of community and centrality concepts to select messages that are least relevant for a node.

The aforementioned buffer management policies do not take into consideration network wide information about the messages and hence may result in suboptimal decisions [54]. Therefore, the success of any optimal buffer management policy relies on availability of sufficient global information about the messages. The two main optimal buffer management policies that consider network wide information are Resource Allocation Protocol for Intentional DTN (RAPID) [44] and optimal buffer management (OBM) [14, 45].

Resource Allocation Protocol for Intentional DTN [44] is the first utility-based buffer management policy. RAPID tries to address message replication problem in DTN while optimizing a specific routing metric like maximizing delivery ratio or minimizing average delivery delay, and based on this routing metric, it derives a per-message utility function. At any contact opportunity, a node first forwards those messages, which can locally provide highest increase in the utility function. However, to derive message utilities, RAPID requires the complete information about all the replicas of a given message. But due to long delays in DTNs, information propagated may become outdated and resulted in suboptimal buffer management decisions.

In RAPID, given limited bandwidth, how messages should be replicated in the system so as to optimize a specific routing metric is a key question. RAPID derives a per-message utility function from the routing metric. At a transfer opportunity, it replicates a message that locally can provide the highest increase in utility. Considering a routing metric such as minimizing average delay of messages, the corresponding utility \( U_i \) of a message \( i \) is the negative of the expected delay to deliver \( i \), i.e., the time has already been spent in the system before \( i \)'s copy reaches its destination. Let \( \delta U_i \) denote the increase in utility by replicating \( i \). Then, A RAPID node firstly replicates the message with the highest value of \( \delta U_i \) among messages in its buffer. In other words, the message with the highest marginal utility is firstly replicated. In general, \( U_i \) is defined as the expected contribution of message \( i \) to the given routing metric. For example, the metric minimizing average delay is measured by summing the delay of all messages. Accordingly, the utility of a message is its expected delay. Thus, RAPID is a heuristic approach based on locally optimizing marginal utility, i.e., the expected delay in utility per message used.

In order to derive message utilities, RAPID requires the flooding of information about all the replicas of a given message. Thus, information propagated may be stale because the characteristic of long delivery delay in opportunistic network. Then, RAPID is only a suboptimal policy in a number of respects and can not provide the optimal network performance.
In [14, 45], Krifa et al. have proposed an OBM policy, which is also based on global information of all the messages, to optimize the specific metrics such as maximizing delivery rate (DR) and minimizing average delivery delay. The OBM policy that aims to maximize average DR drops the message \( i \) that has smallest value of following utility function:

\[
\left(1 - \frac{m_i(T_i)}{L - 1}\right) \frac{1}{\lambda} R_i \exp(-\lambda n_i(T_i)R_i)
\]  

(S3)

Similarly, the OBM policy that minimizes average delivery delay drops the message \( i \) that has smallest value of following utility function:

\[
\frac{1}{n_i(T_i)} \left(1 - \frac{1 - m_i(T_i)}{\lambda} \right)
\]

Because obtaining global information about all messages in the network is impractical in DTNs, the authors have also proposed a history-based scheme which tries to estimate the number of copies of a message based on information gathered through node encounters. The performance of such scheme greatly depends on the accuracy of estimates used. Hence one has to rely on local estimates of the number of copies of a message, which is one of the main focuses of the current work.

Fuzzy logic has been effectively used in many areas of wireless networking like in references [46–48, 55] and even in the area of delay tolerant networks like in references [15, 18, 49, 50], which is the current area of interest.

In [49], FL has been used to assign a rank denoting delivery preference of each message stored in a node’s buffer, and based on this message delivery preference, the node ranks all potential messages and subsequently requests more preferred messages from the encountering node or drops less interested messages when the buffer is full. The message ranking is carried out on the basis of number of message copies and delivery probabilities of encountering nodes.

Adaptive Fuzzy Spray and Wait (AFSnW) [15] is a routing mechanism that integrates fuzzy-based buffer management scheme, inspired by [18], with spray and wait routing protocol. Similar to [18], it uses message size and forward transmission count (FTC) of a message to assign priority to messages. The boundaries of fuzzy membership functions of message size parameter have been adapted on the basis of locally calculated average message size in the node’s buffer. This message prioritization scheme is used while message scheduling. Random drop policy has been used to take drop decisions when buffer overflows.

Adaptive Fuzzy Routing in Opportunistic Network [51] combines three message parameters namely FTC, message size, and TTL using FL to determine the importance of a message among all the messages stored in the buffer. This message priority is used to decide the order of messages to be forwarded when a contact opportunity arises. Nothing has been mentioned about the dropping policy. Also, the simulation results are not very clear. Moreover, there is no comparison with a routing protocol that implements any buffer management policy. Similar to [18], their method for determining number of replicas of a message is not accurate. The FTC of a message is incremented by one whenever a message is transmitted from one node to another. However, if a node encounters another node, which has the same message but different values of FTC, nothing is carried out to correctly update the number of replicas of the message in the network.

From the above study, it is found that a number of factors may be used to decide message scheduling at any given transmission opportunity [50]: RTTL of message (the amount of time left before the message expires), hop count of message (number of hops a message has traveled from the source node to the current node), number of replicas of a message (number of nodes in the network having a message copy), size of a message, age of a message (time since the message has been created), delivery cost (e.g. delivery predictability of message), and distance to the destination (distance between the current buffer node and destination node of a buffer). Out of these parameters, delivery cost and distance metrics may be used for message prioritization, only if routing protocols use these metrics. In certain works [30, 32, 33, 36, 38–41], the aforementioned criterion are used singly while their combination
may result in better decision making. Some works have used multiple message parameters like in references [34, 35, 37] using simple functions, but a small increase or decrease in the value of one parameter may change the value of ranking function substantially, which is not desirable. A better approach

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Reference no./name of scheme</th>
<th>Dropping criterion used for evaluation</th>
<th>Scheduling criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>David et al. [28]</td>
<td>Drop-random; drop-least-recently-received; drop-oldest; drop-least-encountered</td>
<td>—</td>
</tr>
<tr>
<td>5.</td>
<td>Vasco et al. [32]</td>
<td>FIFO; random order; ascending remaining lifetime; descending remaining lifetime; ascending replicated copies and descending replicated copies.</td>
<td>Uses same criteria for scheduling as for dropping in different combinations.</td>
</tr>
<tr>
<td>7.</td>
<td>EBMP [19]</td>
<td>Two dropping functions: first based on estimated number of replicas and age; second based on estimated number of replicas and RTTL of message</td>
<td>—</td>
</tr>
<tr>
<td>8.</td>
<td>MTSBS [35]</td>
<td>Ranks messages according to the dissemination speed and total number of copies of the message; higher ranked messages are dropped.</td>
<td>—</td>
</tr>
<tr>
<td>9.</td>
<td>Bjurefors et al. [36]</td>
<td>Least interested; most interested; max copies; most forwarded least forwarded; random.</td>
<td>—</td>
</tr>
<tr>
<td>10.</td>
<td>TBR [37]</td>
<td>Ranks the messages according to a function of the number of replicas and size.</td>
<td>Ranks the messages using a function of message size, its TTL and its hop count.</td>
</tr>
<tr>
<td>11.</td>
<td>Drop largest [38]</td>
<td>Drops the largest size message.</td>
<td>—</td>
</tr>
<tr>
<td>12.</td>
<td>T-DROP [39]</td>
<td>Drops the message which lies in the threshold range of buffer.</td>
<td>—</td>
</tr>
<tr>
<td>13.</td>
<td>E-DROP [40]</td>
<td>Drops the message of size equal to the size of incoming message.</td>
<td>—</td>
</tr>
<tr>
<td>14.</td>
<td>Mean-drop [41]</td>
<td>Drops those message whose size is greater than the mean of size of buffered messages</td>
<td>—</td>
</tr>
<tr>
<td>15.</td>
<td>OBM [14, 45]</td>
<td>Drops the message on basis of 2 different utility functions based on global knowledge of all the messages: (1) maximizing average delivery ratio of the network and (2) minimizing the average delivery delay.</td>
<td>Uses same utility functions for scheduling as for dropping.</td>
</tr>
<tr>
<td>16.</td>
<td>FLDF [49]</td>
<td>Uses FL to combine number of copies of a message and delivery probability of message to determine message priority.</td>
<td>Uses same message ranking function for scheduling as for dropping.</td>
</tr>
<tr>
<td>17.</td>
<td>AFSnW [15]</td>
<td>Uses random dropping criterion</td>
<td>Uses FL to assign scheduling priority based on message size and forward transmission count.</td>
</tr>
<tr>
<td>18.</td>
<td>AFRON [51]</td>
<td>—</td>
<td>Uses FL to assign scheduling priority based on size, forward transmission count and RTTL of message.</td>
</tr>
</tbody>
</table>

PREP, prioritized epidemic routing; EBMP, enhanced buffer management policy; TTL, time-to-live; TBR, TTL based routing; T-DROP, threshold drop; E-DROP, equal drop; OBM, optimal buffer management; FLDF, Fuzzy Logic based Delivery Framework; AFSnW, Adaptive Fuzzy Spray and Wait; AFRON, adaptive fuzzy routing in opportunistic network; FL, Fuzzy logic; RTTL, remaining time-to-live.
for parameter aggregation is required. A summary of the buffer management schemes discussed in this section is presented in Table I.

In the current work, buffer management is achieved using a new message scheduling policy, which takes into account three most relevant message properties, namely the number of replicas of a message, the RTTL of a message, and its size. These are aggregated using FL controller having 19 output membership functions for message prioritization. The basic objective is to improve the delivery ratio through intelligent selection of messages at the time of scheduling.

A message that has a large number of replicas means it has been in the network for a long time and must have been spread widely; one of the copies will eventually reach the destination, hence, should be given smaller scheduling priority. A message which has short RTTL should be scheduled first as it will expire soon. A message with short RTTL also means that it has been in the network for a long time and may have been spread widely in the network. To take this fact into account forwarding priority is given to those messages that have a short remaining lifetime but whose number of replicas in the network is small. Message size also affects the scheduling decision and preference should be given to shorter messages. A number of short messages may be transmitted in comparison with one big message during a short contact opportunity resulting in the overall increase of the delivery ratio, as desired. Although a weighted approach may be used to combine these parameters, they have values over different ranges. For example, RTTL varies in tens of thousands, the message size is in few KBs, and the number of replicas is a small quantity in tens. Parameter aggregation may be carried out effectively using FL. Some of the previous works have applied FL to assign message priorities either for deciding their transmission priority as in references [15, 18, 49, 51]. However, their method for determining the number of replicas is inaccurate. The scheme proposed in this paper uses a more precise method for estimating the number of replicas of a message by exchanging a my_forward vector, as explained in detail in Section 3.2.1.

3. PROPOSED ALGORITHM

3.1. Design of Fuzzy Logic Controller

A new fuzzy-based buffer management policy for spray and wait routing protocol is proposed, which uses fuzzy decision mechanism to decide the message priority when a transmission opportunity arises. Spray and wait routing protocol is a controlled replication protocol and can gain higher delivery ratio with lower overhead than other replication-based routing protocols. The use of FL realizes a significant saving in complexity without trading off the system performance.

The two main phases in the design of FL controller are the following:

(1) Design of membership functions
(2) Design of fuzzy rule base

The three main steps of FL controller:

(1) Fuzzification: conversion of crisp inputs into fuzzy terms and determining the degree of membership.
(2) Rule evaluation: combining the fuzzy input variables using and, or, and not operators in the if (antecedent) part of the rule and then evaluating the then (consequent) part of the rule. Finally, the output of multiple rules is aggregated to get the final fuzzy output.
(3) Defuzzification: defuzzification is the process of conversion of fuzzy output into a crisp output.

![Figure 1. Fuzzy controller for generating message priority.](image-url)
(4) The three inputs to the proposed fuzzy controller are the following: ETR, RTTL, and message size (size). These are aggregated using the designed fuzzy rule base to produce message_priority as the output. The block diagram of proposed fuzzy controller is given in Figure 1. The parameters and their description are discussed in the following subsections.

3.2.1 Description of input and output fuzzy variables.

(a) Estimated Number of Replicas: ETR is the first parameter used for deciding the message priority. In previous works [11, 31], hop count is used as an estimate of the number of replicas. But the hop count of a message only reflects the number of intermediate nodes in one possible forwarding path and does not indicate the complete message distribution over the entire network. In [15, 18], FTC is used as an indicator of the number of duplicate copies of a message in the network. The source initializes FTC of the message it generates as 1. This FTC value is incremented on both the sender and the receiver sides upon successful transmission. But in this method, if two nodes having a common message meet again and have different values of FTC, nodes do not update their FTC value, thus giving inconsistent FTC value.

In the current work, the method for estimating number of replicas is an enhancement of the previous work [19]. In [19], each node A maintains two variables for each message i it has in its buffer: the estimated total number of replicas (ERA), which denotes the total number of replicas as estimated by node A; and my forward (MFA), which means the number of message replicas created by node A itself. Initially, node A initializes ERA to 1. In the first case, when node A meets another node B and forwards the message i, it increments its MFA by one and also ERA by one; node B derives it ERB from ERA and sets its MFB as zero.

\[ \begin{align*}
    ERA_i &= ERA_i + 1 \\
    MFA_i &= MFA_i + 1 \\
    ERB_i &= ERA_i \\
    MFB_i &= 0
\end{align*} \] (4) (5)

Figure 2. Method of ETR updation in enhanced buffer management policy (EBMP).
When node A meets node B that already has a copy of message \( i \) they exchange their MF values and update their ER values, respectively.

\[ ER_i^A = ER_i^A + MF_i^B \]
\[ ER_i^B = ER_i^B + MF_i^A \]  \hspace{1cm} (6)

However, if the same nodes meet frequently, then there is a high likelihood that the ER value is falsely incremented because of exchange of MF values as shown in Figure 2. To overcome this problem, a new scheme is being proposed, where nodes maintain the replica list of each message. The replica list stores the id of all nodes to which node itself or other encountered node has forwarded a copy of the message. Because the number of replications of a message is controlled by the L-value of spray and waits routing protocol, the size of ETR list is limited to L. Whenever two meeting nodes have a common message, they exchange the replica list of this message and take its set difference and update their respective replica lists. The size of the set difference is added to the ETR value of the corresponding nodes. This avoids the false updation of ETR as depicted in Figure 3.

In [14, 45], the authors have derived a message utility function with respect to the total DR, which is a function of the global state of the message in the network. In case of buffer overflow, a node should drop that message \( i \), which has the smallest value of the utility function defined in Equation (7):

\[ \left( 1 - \frac{m_i(T_i)}{L - 1} \right) \lambda R_i \exp(-\lambda n_i(T_i)R_i) \leq \left( 1 - \frac{m_i(T_i)}{L - 1} \right) \lambda R_i \frac{\exp(R_i)}{\exp(\lambda n_i(T_i))} \]  \hspace{1cm} (7)

Here, \( n_i(T_i) \) is the number of copies of message \( i \) after elapsed time \( T_i \), \( m_i(T_i) \) is the number of nodes who have seen a copy of message \( i \) after elapsed time \( T_i \), \( R_i \) is RTTL of message \( i \), \( L \) is total number of nodes in the network, \( K(t) \) is the total number of messages in the network at time \( t \), and \( \lambda \) is the meeting rate between two nodes.
From the formula (4), it may be deduced that the message which has the highest number of replicas or the one which has largest value of \( n_i(T_i) \), if dropped, will result in largest change in delivery ratio as the utility function varies in inverse proportion to exponential of \( n_i(T_i) \). This suggests the reason for high scheduling priority to messages having small ETR value.

This is also in concurrence with the law of diminishing marginal utility, which has been mentioned in [35]. This law states that as a user increases consumption of a product, there is a decline in the marginal utility that user derives from consuming each additional unit of that product. Hence, in [35], the authors have used a buffer management policy that drops messages that have larger estimated number of copies and faster dissemination speed than other messages. From the perspective of the whole network, if a message with fewer copies is dropped, the decrease in utility (in terms of delivery ratio) is larger than the increase in utility if a message with larger number of copies is kept. On the contrary, when a message having a large number of copies in the network is dropped, the decrease in utility is lesser than the increase in utility when a node accepts a message with few copies. Messages that have a large number of copies in the network have seen higher transfer opportunities and will have relatively higher probability to reach their destination. When the number of such messages in the network is controlled, the loss of utility is limited. However, it brings more transfer opportunities for messages that have fewer copies in the network.

Estimated number of replicas has been defined using three membership functions: small, medium, and large, as shown in Figure 4. Because spray and wait routing protocol has been used, the maximum number of copies of a message is defined by L-value of the protocol. Hence, the range of ETR varies from 0 to L. As L has been taken to be 6 in simulation, ETR varies from 0 to 6. The membership function low takes value 1 when the number of copies of a message is 1 or 2, that is, when the message has been initially generated; the membership function medium takes value 1 when the number of copies of a message is 3, that is, when half the copies of a message have been distributed; and the membership function high takes value 1 when the number of copies of a message is 5 or 6, that is, when almost all copies of a message have been spread.

(b) Remaining time-to-live: each message in DTN is assigned a TTL, an application-based parameter, after which the message expires. It denotes the time for which the message will remain valid in the network. All the nodes including source node and relay nodes will drop the message or its copies once its TTL expires. The RTTL of a message is equal to (initial TTL value—current time) and it denotes the time remaining before the message expires. The RTTL of replicated messages is the same as the original message. A message that has a smaller RTTL will expire faster and should be given preference while scheduling. A message with higher value of RTTL is relatively new in the network and is liable to be spread further within the network in subsequent contacts. The reason for including RTTL as one of the input parameters is to increase the conversion rate of relayed messages to delivered messages; hereby resulting in decrease in overhead ratio. Rather than simply taking absolute value of RTTL, the percentage of TTL value left of the message is considered, which is calculated as follows:

\[
RTTL = \frac{\text{initialTTL} - \text{currentTime}}{\text{initialTTL}} \times 100
\]
For example, consider a message $M$ having TTL value 1000 time units, and if current time is 500 time units, then RTTL of $M$ is

$$RTTL = \frac{1000 - 500}{1000} \times 100 = 50\%$$

Fuzzy membership functions for RTTL have been defined over the range 0-100. Three membership functions are used for RTTL: low, medium, and high as shown in Figure 5. The range of RTTL is taken to be (0-100). The membership function low takes value 1 when percentage of TTL left is 25% of initial TTL; the membership function medium takes value 1 when percentage of TTL left is 40-60% of initial TTL; and finally the membership function high takes value 1 when percentage of TTL left is 75% of initial TTL.

(c) Message size (size): message size also affects the message scheduling policy. As the transmission opportunity between nodes is limited, the probability of successful transmission of smaller messages is higher. Hence, transmitting smaller message maximizes the number of successful transmissions in each short contact opportunity. According to [18–22,24–26], this approach reduces partial transmissions and increases the number of messages delivered. Three membership functions have been
defined for message size (size) small, average, big as shown in Figure 6. The range of size is taken to be [0-Max_size], where Max_size is the maximum allowable size of a message in the network.

(d) Message priority (message_priority): It is a 19-membership function output variable as shown in Figure 7, which assigns the scheduling priorities to different messages varying from MP1-MP19 with MP1 being the highest priority. This priority is assigned on the basis of three input variables: ETR, RTTL, and size of a message. Nineteen output membership functions are used to achieve higher accuracy of message_priority.

3.2.2 Description of Fuzzy Rule Base. The heart of any fuzzy system is its rule base which defines the input output mapping. Table II shows the rule base of the fuzzy controller used to decide the message priority.

<table>
<thead>
<tr>
<th>Fuzzy rule</th>
<th>ETR</th>
<th>RTTL</th>
<th>Size</th>
<th>Message_priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1</td>
<td>Low</td>
<td>Low</td>
<td>Small</td>
<td>MP1</td>
</tr>
<tr>
<td>Rule 2</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>MP1</td>
</tr>
<tr>
<td>Rule 3</td>
<td>Low</td>
<td>Low</td>
<td>Large</td>
<td>MP2</td>
</tr>
<tr>
<td>Rule 4</td>
<td>Low</td>
<td>Medium</td>
<td>Small</td>
<td>MP2</td>
</tr>
<tr>
<td>Rule 5</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>MP2</td>
</tr>
<tr>
<td>Rule 6</td>
<td>Low</td>
<td>Medium</td>
<td>Large</td>
<td>MP3</td>
</tr>
<tr>
<td>Rule 7</td>
<td>Low</td>
<td>High</td>
<td>Small</td>
<td>MP3</td>
</tr>
<tr>
<td>Rule 8</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
<td>MP4</td>
</tr>
<tr>
<td>Rule 9</td>
<td>Low</td>
<td>High</td>
<td>Large</td>
<td>MP4</td>
</tr>
<tr>
<td>Rule 10</td>
<td>Medium</td>
<td>Low</td>
<td>Small</td>
<td>MP2</td>
</tr>
<tr>
<td>Rule 11</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>MP2</td>
</tr>
<tr>
<td>Rule 12</td>
<td>Medium</td>
<td>Low</td>
<td>Large</td>
<td>MP4</td>
</tr>
<tr>
<td>Rule 13</td>
<td>Medium</td>
<td>Medium</td>
<td>Small</td>
<td>MP5</td>
</tr>
<tr>
<td>Rule 14</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>MP6</td>
</tr>
<tr>
<td>Rule 15</td>
<td>Medium</td>
<td>Medium</td>
<td>Large</td>
<td>MP7</td>
</tr>
<tr>
<td>Rule 16</td>
<td>Medium</td>
<td>High</td>
<td>Small</td>
<td>MP8</td>
</tr>
<tr>
<td>Rule 17</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>MP9</td>
</tr>
<tr>
<td>Rule 18</td>
<td>Medium</td>
<td>High</td>
<td>Large</td>
<td>MP10</td>
</tr>
<tr>
<td>Rule 19</td>
<td>High</td>
<td>Low</td>
<td>Small</td>
<td>MP11</td>
</tr>
<tr>
<td>Rule 20</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>MP12</td>
</tr>
<tr>
<td>Rule 21</td>
<td>High</td>
<td>Low</td>
<td>Large</td>
<td>MP13</td>
</tr>
<tr>
<td>Rule 22</td>
<td>High</td>
<td>Medium</td>
<td>Small</td>
<td>MP14</td>
</tr>
<tr>
<td>Rule 23</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>MP15</td>
</tr>
<tr>
<td>Rule 24</td>
<td>High</td>
<td>Medium</td>
<td>Large</td>
<td>MP16</td>
</tr>
<tr>
<td>Rule 25</td>
<td>High</td>
<td>High</td>
<td>Small</td>
<td>MP17</td>
</tr>
<tr>
<td>Rule 26</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>MP18</td>
</tr>
<tr>
<td>Rule 27</td>
<td>High</td>
<td>High</td>
<td>Large</td>
<td>MP19</td>
</tr>
</tbody>
</table>

ETR, estimated number of replicas; RTTL, remaining time-to-live.

When a node A meets another node B, the messages are exchanged in the following order as in [11]:

Step 1: All messages in node A’s buffer destined to node B are sent first and deleted from node A’s buffer. Also, both nodes A and B add ID of these messages in their acknowledgement list.
Step 2: Routing information between nodes is exchanged and message status is updated.
Step 3: Acknowledgement of messages that have been delivered to node A and acknowledgement of other messages that node A has received from other nodes are propagated to node B.
Step 4: Messages are transferred from message buffer considering based on message_priority value.

Algorithm 1: Procedure EFSnWR: encounter (A, B)
1: for each message m ∈ BufferA do
2:   if m ∈ deliveredMessagesACKS OR TTL(m) expires then
3:     // if m belongs to list of delivered message or its TTL has expired
4:       A.removeFromBuffer(m); // remove message m from node A’s buffer
5:   end if
6: for each message m ∈ BufferA do
7:   D=m.Destination;
8:   if B = D then // if destination of message m is encountered node B
9:     B.addInBuffer(m); // then transfer message m to node B
10:    Add m to deliveredMessagesACKS;  //
11:   else
12:   compute_fuzzy(m) // compute priority of message m
13:   sortMsgByFuzzyPriority(BufferA) // sort messages in node A’s buffer based
14:      // on fuzzy priority
15: end if-else
16: for each message m ∈ BufferA do // transfer messages in node A’s buffer to node B
17:   B.addInBuffer(m)
18: end for.

Whenever a node receives a message from another encountered node, if the message buffer in the node has free space, it accepts the new message. Otherwise, it randomly selects a message from the buffer and drops it. This is carried out to make the algorithm fairer. The concept of acknowledgements has also been used as presented in [11, 52] in order to help to explicitly clear previously delivered messages in the network. When nodes send acknowledgements of already delivered messages to other nodes, these messages, which are buffered at intermediate nodes, are removed and storage capacity for upcoming messages is improved. This is a very important feature because network nodes have limited storage capabilities. Moreover, it also helps to stop replicating/forwarding already delivered messages, thus also saving bandwidth resources.

4. SIMULATION ENVIRONMENT

The performance study of proposed and existing protocols has been carried out by simulating various DTN scenarios on the Opportunistic Network Environment (ONE) simulator [53], which is a very popular and powerful DTN simulation tool.

Message structure has been defined in ONE simulator, which includes fields such as message ID, source node ID, destination node ID, message size, message path, creation time, receive time, initial TTL, and response size. It also allows for appending new fields for specifying additional message properties.

The performance of Enhanced Fuzzy Based-SnW Routing Protocol has been compared with the performance of three other spray-based routing protocols: simple spray and wait (SnW) [10] and two other variations of spray and wait routing protocol, which apply buffer management techniques involving multiple message parameters namely, AFSnW protocol [15] and TBR [37]. The spray and wait routing protocol in ONE simulator has been modified to implement the current work.

Three metrics have been used to analyze the new and existing protocols’ performance.

Delivery ratio: this is defined as ratio of total number of messages delivered (N_{delv}) to the total number of messages created (N_{creat}).

\[
DR = \frac{N_{delv}}{N_{creat}}
\]
Latency average (ALat): this is defined as the median of the time that was required by a message to get delivered.

\[ ALat = \text{median}(T_{\text{delv}} - T_{\text{creat}}) \]  

(10)

where, \( T_{\text{delv}} \)—time of message delivery, \( T_{\text{creat}} \)—time of message creation, delivered—list of messages that are delivered.

Overhead ratio: it is defined as the ratio of the total number of messages relayed (\( N_{\text{rel}} \)) to the total number of messages delivered (\( N_{\text{delv}} \)).

\[ OR = \frac{N_{\text{rel}} - N_{\text{delv}}}{N_{\text{delv}}} \]  

(11)

For purpose of comparison of proposed buffer management scheme with existing schemes, the simulation scenario has been taken from [37]. Performance of different protocols has been studied under two mobility models namely map-based movement model and random waypoint model. To evaluate the efficiency of protocols in a city environment, map-based movement model has been used, whereas random waypoint model has been used to examine the effectiveness of protocols in a random scenario.

4.1.1. Map-based vehicular model. In map-based vehicular model the nodes of the network move according to actual streets in an imported map. A simulation environment corresponding to a map of 4500 \( \times \) 3400 m section of Helsinki city has been considered. Three types of nodes have been considered for the simulation—cars, trams, and pedestrians. The transmission ranges for pedestrians, cars, and trams are taken to be 10, 20, and 20 m, respectively. Data transmission rate of all nodes in the network is taken to be 250 KBps (2 Mbps). The buffer sizes for pedestrians, cars, and trams are taken to be 5, 10, and 50 MB, respectively. Speed of pedestrians, cars, and trams vary within [0.5, 1.5] m/s, [2.7, 13.9] m/s and [8, 11] m/s, respectively. In each of the iteration, the node distribution is kept as: 12\% trams, 28\% cars, and 60\% pedestrians. Table III summarizes the parameters used for simulating the first scenario.

4.1.2. Random waypoint mobility model. In Random waypoint mobility model, nodes move in random directions with random speeds. In this scenario, a simulation area of 4500 \( \times \) 3400 m has been considered. Nodes have a transmission range of 30 m and buffer space of 25 MB. Nodes move within a speed range of [0.5, 5] m/s and pause time between 0 and 120 s. Table IV summarizes the parameters used for simulating the second scenario.

4.2. Simulation results and analysis

4.2.1. Results for map-based movement model. The impact of the network size on the performance metrics has been studied to compare the performance of all the protocols. The number of nodes varies from 50 to 200 in step size of 25.

Figure 8 shows the variation of delivery ratio with varying number of nodes in the network for map-based vehicular movement model. It can be observed from the figure that with the increase in number of nodes, the delivery ratio increases for all the protocols. This is due to increase in number of contact opportunities with the increase in number of nodes in the network. Enhanced fuzzy SnW routing (EFSnWR) achieves—6.7-12.2\% higher delivery ratio than TBR, 3.5-20\% higher delivery ratio than AFSnW, and 22.5-30.4\% higher delivery ratio than SnW routing protocol. Table V presents the result of T-test between the delivery ratios of EFSnWR and other quota-based protocols, such as TBR, AFSnW, and SnW. The T-test compares the actual difference between the means of two data in relation to the variation in the data. The differences in mean delivery ratios between EFSnWR and TBR by varying the number of nodes are 0.058257; that between EFSnWR and AFSnW are 0.0537714 and that between EFSnWR and SnW are 0.128314. The positive value in the differences of the means demonstrates the better performance of EFSnWR.
Table III. Summary of simulation parameters for first scenario.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility model</td>
<td>Map-based movement model</td>
</tr>
<tr>
<td>Simulation time</td>
<td>12 hours</td>
</tr>
<tr>
<td>Simulation area</td>
<td>$4500 \times 3400$ m</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>50-200</td>
</tr>
<tr>
<td>Number of groups of nodes</td>
<td>3 (pedestrians, cars, trams)</td>
</tr>
<tr>
<td>Number of pedestrians</td>
<td>60% (total number of nodes)</td>
</tr>
<tr>
<td>Number of cars</td>
<td>28% (total number of nodes)</td>
</tr>
<tr>
<td>Number of trams</td>
<td>12% (total number of nodes)</td>
</tr>
<tr>
<td>Transmission speed</td>
<td>250 KBps (for all groups)</td>
</tr>
<tr>
<td>Transmission range</td>
<td>10 m (pedestrians), 20 m (cars, trams)</td>
</tr>
<tr>
<td>Buffer size</td>
<td>5 MB (pedestrians), 10 MB (cars), 100 MB (trams)</td>
</tr>
<tr>
<td>Node speed</td>
<td>Pedestrians = [0.5, 1.5] m/s, Cars = [2.7, 13.9] m/s, Trams = [7, 10] m/s</td>
</tr>
<tr>
<td>Message generation interval</td>
<td>0.6 s</td>
</tr>
<tr>
<td>Message TTL</td>
<td>60-240 min</td>
</tr>
<tr>
<td>Value of $L$</td>
<td>6</td>
</tr>
</tbody>
</table>

TTL, time-to-live.

Table IV. Summary of simulation parameters for second scenario.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility model</td>
<td>Random waypoint model</td>
</tr>
<tr>
<td>Simulation time</td>
<td>12 hours</td>
</tr>
<tr>
<td>Simulation area</td>
<td>$4500 \times 3400$ m</td>
</tr>
<tr>
<td>Number of groups of nodes</td>
<td>1</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>50-200</td>
</tr>
<tr>
<td>Transmission speed</td>
<td>250 KBps</td>
</tr>
<tr>
<td>Transmission range</td>
<td>30 m</td>
</tr>
<tr>
<td>Node speed</td>
<td>0.5-5 m/s</td>
</tr>
<tr>
<td>Message generation interval</td>
<td>0.6 s</td>
</tr>
<tr>
<td>Message TTL</td>
<td>60-240 min</td>
</tr>
<tr>
<td>Value of $L$</td>
<td>6</td>
</tr>
</tbody>
</table>

TTL, time-to-live.

Figure 8. Variation of delivery ratio with varying number of nodes in Map Based Mobility Movement Model.

in comparison with other protocols. The $p$-values of the $T$-tests are less than 0.001 for TBR and SnW routing protocols, which denotes that our simulation results are extremely statistically significant; while it is less than 0.01 for AFSnW, which denotes that our simulation results are statistically significant.
Enhanced fuzzy spray and wait routing uses a more accurate method for determining the number of replicas of a message as compared with TBR and AFSnW, which helps in proper ranking of messages. The buffer management strategy used in EFSnWR gives messages with small number of replicas and which are near their deadline in the network, an opportunity to reach the destination; whereas AFSnW does not use RTTL as a parameter for message prioritization. Hence, EFSnWR achieves such high delivery ratios because of the intelligent message scheduling policy incorporated in the routing protocol.

Figure 9 compares the change in overhead ratio of different protocols with changing number of nodes in the network for Map-based movement model. The results show that EFSnWR is the most resource friendly protocol as its overhead is minimum among all the routing protocols. The main reason behind such low overhead in EFSnWR is its effective message forwarding. Also as due consideration is given to RTTL message parameter, majority of relayed messages are able to reach their destination without being dropped in transit; thereby reducing the overhead ratio. As the number of delivered messages by using EFSnWR is higher than other protocols in all the scenarios, hence its corresponding overhead ratio is smaller in all the scenarios. This also shows that in order to achieve high delivery ratio, it does not use additional relay messages. The overhead ratio increases with the increase in number of nodes in the network for all the protocols as more transmissions take place because of more contact opportunities between nodes. As simple SnW uses FIFO scheduling without any acknowledgements, its overhead ratio is highest in all the cases. The use of acknowledgements removes already delivered messages from node buffers preventing their further dissemination. EFSnWR achieves 12.5-20.2% lesser overhead ratio when compared with TBR; 4.8-15.5% lesser overhead ratio than AFSnW; and 20.5-28% lesser overhead ratio than SnW routing protocol. The $T$-test results presented in Table VI prove the statistical significance of our results.

<table>
<thead>
<tr>
<th>Difference in mean</th>
<th>p-value</th>
<th>Statistical significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBR</td>
<td>0.052857143</td>
<td>1.4796E-05</td>
</tr>
<tr>
<td>AFSnW</td>
<td>0.053771429</td>
<td>0.002167274</td>
</tr>
<tr>
<td>SnW</td>
<td>0.12831423</td>
<td>1.54715E-05</td>
</tr>
</tbody>
</table>

TBR, time-to-live-based routing; AFSnW, Adaptive Fuzzy Spray and Wait; SnW, Spray and Wait.

Figure 10 presents the variation of latency median with the varying number of nodes in the network. EFSnWR protocol’s latency is higher as compared with that of AFSnW and TBR. The latency is computed considering the messages that have been delivered to the destination. As EFSnWR delivers more messages in comparison with other routing protocols, its latency median is higher than other quota-based protocols. Also, as low RTTL is kept as basis for message forwarding, preventing them from dropping; this may be dropped otherwise in other algorithms. But the main objective of increased delivery ratio is achieved.
4.2.2 Results for random waypoint model: As in random waypoint movement model, the node movement is random; the average number of node contacts is lesser as compared with map-based movement model. Hence, the delivery ratio of all the protocols is smaller than that in map-based movement model, which is apparent from Figure 8 and Figure 11. But in case of random waypoint model also, the delivery ratio of EFSnWR is higher than that of other protocols. EFSnWR achieves 7-10% higher delivery ratio than TBR, 3.5-21.4% higher delivery ratio than AFSnW, and 13-25% higher delivery ratio than SnW routing protocol. These results are analogous to the results observed in map-based movement model. Table VII presents the result of T-test between the delivery ratios of EFSnWR and other quota-based protocols, such as TBR, AFSnW, and SnW. The differences in mean delivery ratios between EFSnWR and TBR for varying the number of nodes are 0.037714; that between EFSnWR and AFSnW are 0.061143 and that between EFSnWR and SnW are 0.081385. The positive value in the aforementioned differences demonstrates the better performance of EFSnWR in comparison with other protocols. The p-values of the T-tests are less than 0.001 for TBR and SnW routing protocols, which denotes that our simulation results are extremely statistically significant; whereas it is less than 0.01 for AFSnW, which denotes that our simulation results are statistically significant. Figures 12 and 13 present the variation of overhead ratio and latency median for varying number of nodes, respectively, random waypoint model. Table VIII

![Figure 10. Variation of latency median with varying number of nodes for Map based vehicular movement model.](image1)

![Figure 11. Variation of delivery ratio with varying number of nodes for Random Waypoint movement model.](image2)
presents the results of paired T-test between overhead ratio of EFSnWR and other quota based routing protocol.

At higher contact opportunities, that is, in the case where number of nodes in the network is more, EFSnWR shows prominent improvement in both map based movement scenario and Random Waypoint scenario. This is because at every contact opportunity, the algorithm makes better scheduling decision than other algorithms.

Figure 14 shows the effect of increasing buffer size on delivery ratio of various protocols. As the buffer size of nodes increases, the delivery ratio also increases as more number of messages may be

Table VII. Results of paired T-test between the delivery ratios of EFSnWR and other quota based protocol: random waypoint movement model for varying number of nodes.

<table>
<thead>
<tr>
<th></th>
<th>Difference in mean</th>
<th>p-value</th>
<th>Statistical significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBR</td>
<td>0.037714286</td>
<td>0.00086855</td>
<td>Extremely statistically significant</td>
</tr>
<tr>
<td>AFSnW</td>
<td>0.061142857</td>
<td>0.00617678</td>
<td>Statistically significant</td>
</tr>
<tr>
<td>SnW</td>
<td>0.081385714</td>
<td>1.4441E-05</td>
<td>Extremely statistically significant</td>
</tr>
</tbody>
</table>

TBR, time-to-live-based routing; AFSnW, Adaptive Fuzzy Spray and Wait; SnW, Spray and Wait.

Table VIII. Results of paired T-test between the overhead ratios of EFSnWR and other quota based protocol: random waypoint movement model for varying number of nodes.

<table>
<thead>
<tr>
<th></th>
<th>Difference in mean</th>
<th>p-value</th>
<th>Statistical significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBR</td>
<td>−1.1202</td>
<td>7.0244E-06</td>
<td>Extremely statistically significant</td>
</tr>
<tr>
<td>AFSnW</td>
<td>−1.737971429</td>
<td>5.0553E-06</td>
<td>Extremely statistically significant</td>
</tr>
<tr>
<td>SnW</td>
<td>−1.930285714</td>
<td>5.9475E-06</td>
<td>Extremely statistically significant</td>
</tr>
</tbody>
</table>

TBR, time-to-live-based routing; AFSnW, Adaptive Fuzzy Spray and Wait; SnW, Spray and Wait.
accommodated in a node’s buffer and hence reducing message drops. Even for small buffer sizes, our protocol performs well in terms of delivery ratio and overhead ratio because of better buffer management. For small buffer sizes, all protocols show low delivery ratio as messages get dropped at earlier stages only and sometimes without being forwarded. As buffer size increases, all protocols uniformly show improvement in delivery ratio, but improvement is more significant in EFSnWR because of proper scheduling of messages. Figure 15 and 16 show variation of overhead ratio and latency with variation in buffer size at nodes.
5. CONCLUSION

In this paper, spray and wait routing protocol has been enhanced using a new fuzzy-based buffer management policy called EFSnWR, with the objective of achieving high delivery ratio using proper aggregation of multiple message parameters. EFSnWR uses a fuzzy prioritization-based message scheduling policy along with random drop policy. Using global information is difficult in delay tolerant environments where nodes move randomly and delays are long. Therefore, this scheme uses local message properties: ETR of a message, its size, and RTTL to decide the message priority. Since determining exact number of replicas of a message is difficult in DTN environment, a method for local estimation of number of replicas has also been proposed in this work, which is an improvement over previous works. Simulation results prove the success of proposed buffer management scheme in accomplishing the desired objective by improving the delivery ratio and overhead ratio as compared with other variations of spray and wait routing protocol like TBR and AFSnW. In future, the proposed work will be tested on real mobility traces to study its performance in real environments. Further work will be carried out to improve the delivery ratio by incorporating the proposed scheme using traffic differentiation.

ACKNOWLEDGEMENTS

Dr Joel Rodrigues and Dr Vasco N. G. J. Soares would like to thank the Instituto de Telecomunicações, Next Generation Networks and Applications Group, Covilhã Delegation, and to the National Funding from the Fundação para a Ciência e Tecnologia through the Pest-OE/EEI/LA0008/2013 Project for their support.

REFERENCES


17. Siddiqui N, Adeli H. Introduction to Computational Intelligence. John Wiley & Sons, 2013; DOI: 10.1002/9781118534823.ch1


