Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9773402, 12 pages
https://doi.org/10.1155/2021/9773402

Research Article

SS-Drop: A Novel Message Drop Policy to Enhance Buffer Management in Delay Tolerant Networks

Obaid ur Rehman,[1] **Irshad Ahmed Abbasi**,[2] **Hythem Hashem**,[2] **Khalid Saeed**,[3] **Muhammad Faran Majeed**,[3] **and Sikandar Ali**[4,5]

[1] *Department of Computer Science, National University of Modern Languages, Islamabad, Pakistan*
[2] *Department of Computer Science, Faculty of Science and Arts at Belgarn, University of Bisha, Sabt Al-Alaya 61985, Saudi Arabia*
[3] *Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal, Dir (U), Khyber Pakhtunkhwa, Pakistan*
[4] *Department of Computer Science & Technology, China University of Petroleum-Beijing, Beijing 102249, China*
[5] *Beijing Key Lab of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing 102249, China*

Correspondence should be addressed to Irshad Ahmed Abbasi; aabasy@ub.edu.sa and Sikandar Ali; sikandar@cup.edu.cn

Received 15 April 2021; Accepted 21 May 2021; Published 9 June 2021

Academic Editor: Suleman Khan

A challenged network is one where traditional hypotheses such as reduced data transfer error rates, end-to-end connectivity, or short transmissions have not gained much significance. A wide range of application scenarios are associated with such networks. Delay tolerant networking (DTN) is an approach that pursues to report the problems which reduce communication in disrupted networks. DTN works on store-carry and forward mechanism in such a way that a message may be stored by a node for a comparatively large amount of time and carry it until a proper forwarding opportunity appears. To store a message for long delays, a proper buffer management scheme is required to select a message for dropping upon buffer overflow. Every time dropping messages lead towards the wastage of valuable resources which the message has already consumed. The proposed solution is a size-based policy which determines an inception size for the selection of message for deletion as buffer becomes overflow. The basic theme behind this scheme is that by determining the exact buffer space requirement, one can easily select a message of an appropriate size to be discarded. By doing so, it can overcome unnecessary message drop and ignores biasness just before selection of specific sized message. The proposed scheme Spontaneous Size Drop (SS-Drop) implies a simple but intelligent mechanism to determine the inception size to drop a message upon overflow of the buffer. After simulation in ONE (Opportunistic Network Environment) simulator, the SS-Drop outperforms the opponent drop policies in terms of high delivery ratio by giving 66.3% delivery probability value and minimizes the overhead ratio up to 41.25%. SS-Drop also showed a prominent reduction in dropping of messages and buffer time average.

1. Introduction

Wireless technology has a great impact over our lives due to pervasive communication. With the passage of time, a paradigm shift has occurred in networking and its applications. Challenged network, i.e., Remote Area Networks [1], Military Battlefield Networks [2], Postdisaster Response Networks (PDRNs) [3], Interplanetary Networks (IPNs) [4], and Energy Constrained/Sparse.

Wireless Sensor Networks (WSNs) [5] is a network where traditional hypotheses such as reduced data transfer error rates, end-to-end connectivity, or short transmissions have not much significance [6]. A wide range of applications scenarios are associated with such networks [7]. Delay tolerant networking is an approach that pursues to report the problems that reduce communication in disrupted networks [6]. The Delay tolerant network is referred to that network in which end-to-end connectivity rarely exists [8]. Delay tolerant networking (DTN) architecture was presented in 2003 in which the problem of credible message delivery is tried to address by adopting the notion of store-carry and forward [6]. In such networks, store-carry and forward paradigm is used to achieve a successful sending of messages to destinations [9]. Providing the prearrangement of buffer policies is problematic. Especially in resource-oblige delay tolerant networks (DTNs) where a high ratio for delivery of data and least overhead is supposed to be achieved in dense congested circumstances [10].

Recent research suggests that proper buffer management policies can enhance effectiveness of DTN routing protocols. Routing in DTN is also a challenging problem [8, 11]. In order to minimize the impact of buffer overflow in such congested and constrained environments, buffer management policies should carefully select the messages to be discarded instead of a blind dropping. Several buffer management schemes have been proposed which consider various message attributes in order to select the appropriate messages to be dropped [9, 12, 13, 14].

The latest essential issue mostly disregarded by the DTN researchers was the influence of dropping messages in terms of buffer overflow. This is the reason why traditional buffer management policies like drop front, drop last, and drop random do not have any efficient mechanism to select a message to be dropped. As an alternative such schemes depends upon the order which the messages adopt for arriving and residing in the buffer. Many schemes are fully dependent on the priority assigned to each message. Such schemes are not suitable for DTNs because their performance is better in frequently connected networks rather than frequently disconnected ones. To improve the delivery ratio, to reduce communication overhead, to avoid redundant message drop, and to enhance buffer management are the key points of this research.

The paper is summarized as Section 2 consists of details about buffer management in DTN. Section 3 contains the significant work done. Section 4 comprises a proposed model of the system. Section 5 consists of simulation settings. Section 6 is all about results and discussion. And Section 7 includes brief conclusion and discusses future work.

2. Buffer Management in DTN

Buffer management is an important mechanism to control the buffer resources. Buffer management is responsible for scheduling of messages. An effective buffer management is necessary to make decisions for dropping messages. Whenever the buffer becomes overflow, it needs to drop some messages in order to make room for new ones [15].

2.1. Buffer Management Architecture. Figure 1 describes basic buffer management architecture at a DTN router. Upon arrival, an incoming message is classified according to the criterion implemented by that router and then stored in its buffer. At a contact opportunity, the scheduler decides the order by which messages should be replicated or transmitted as contact durations are limited. In case of buffer overflow, messages which should be discarded are decided by the buffer manager. In other words, scheduling strategies govern the order in which the messages are passed in case of bandwidth and contact constraints.

2.2. Buffer Management Policies in DTN. One of the major issues in DTN is that it faces the problem of buffer overflow due to which dropping of messages is the intense need. For the said reason, several message drop policies were used, and these techniques got different updates and versions time to time. A comprehensive and efficient buffer management technique is the soul requirement of DTN. Hence, for a short review, the existing message drop policies of the DTN have been divided into two different categories: policies having/-keeping network knowledge and policies do not having/keeping network knowledge.

2.2.1. Policies Having No Network and Its Component Knowledge. DTN possesses many buffer management policies in which some requires or use the local information available for a message in network like when the message enter in to the buffer or node, how much is time-to-live (TTL) for the message, etc. These network policies which does not keeps the knowledge of remaining network are quite useful as it is independently working, and it does not require any specific routing scheme to be implemented. Some examples of such policies are as follows:

(1) Drop Front (DF). This policy has the capability or mechanism of message dropping from the front of memory as the buffer becomes overflow. Drop front works on the first-in first-out mechanism. The main problem with this policy is that maximum end-to-end connectivity is mandatory for it, which is quite impossible in DTNs [16].

(2) Drop Last (DL). This scheme works in contrary to drop front. It uses the LIFO algorithm where the last incoming message is selected for dropping upon buffer overflow. This policy increases the probability of message dropping which directly increases the availability of free buffer space for new messages. On the other hand, it leads towards biasness by selecting every time the last message to drop [16].

(3) Drop Random (DR). This policy works on random selection of messages for dropping as buffer becomes overflow. The random deletion property ensures unbiasedness towards message selection. In this technique, each and every message has an equal priority of deletion. In contrast, it may lead towards the wastage of valuable information by dropping such messages which has to travel a lot in future in the network [17].

(4) Drop Youngest (DY). In DTN, every message has its life time called tme-to-live (TTL). TTL of a message depends upon how much time it spends and for how much time it will travel in a network. Drop youngest buffer management policy drops a very recent and freshly arrived message, whose TTL is the highest amongst all the arrived messages. The reason behind this policy is to minimize the overhead ratio as it

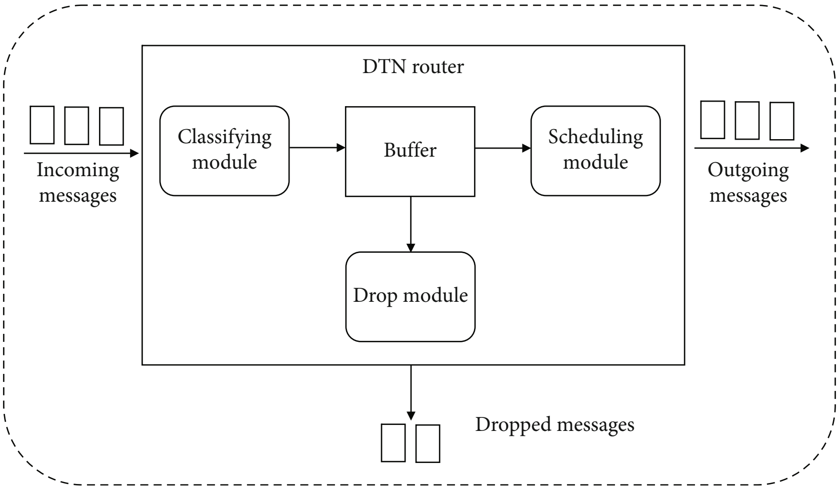


Figure 1: DTN buffer management architecture.

is very much clear that the message whose TTL is long has consumed very little resources [18].

(5) Drop Oldest (DOA). In disrupted networks, a message which traveled for more time in a network and left with very less TTL is considered to be the oldest message. Drop oldest is that type of policy, who's specific and targeted victims are the oldest messages. It always selects those messages to drop who are left with lowest TTL [16].

(6) Drop Largest (DLA). DLA selects size as a criterion for dropping of message as buffer becomes saturated [19]. It selects largest size messages in the buffer for dropping to free up space for the incoming messages. This scheme also ensures that messages having small size will be left for more time in the network, and they can live better with. Hence, they got more forwarding predictability as compared to large size messages [20].

2.2.2. Policies Having Network Knowledge. Other group of buffer management schemes keeps the information about network components such that the number of nodes in the network, number of copies of each message, gathering or meeting time, and ratio of the nodes. These policies take its decisions to drop messages on the basis of available information. Some examples of such schemes are as follows.

(1) Evict Most Forwarded First (MOFO). In DTN, messages move from sender to receiver through hop to hop count. MOFO drop policy considers the quantity of hop counts as a parameter to drop a message from buffer as it gets overflow. MOFO first sets a sending counter value for a message and then a message with the highest counter value that is selected to drop from buffer. This property of MOFO ensures that the messages with a lower counter value may have good chance to travel more within the network [21].

(2) RAPID's Utility-Based Drop. In 2007 [22], developed a new message drop policy which considers the information of network knowledge named as RAPID or resource allocation policy. This scheme intends to keep the information, based on locally optimizing variable which is the estimated delay in function per message used. In general, this can be stated as "the expected contribution of message to the given routing metric." This means that the variable or utility it considers is the expected delay of the message.

(3) Global Knowledge-Based Drop (GBD). Every message in the buffer will have some statistical attributes like message size, time-to-live (TTL), sending node information, and arrival time. In 2008, [23] presented a buffer management policy, that considers all such statistical information to originate every message value on the basis of contact history of a node in the network. The maximization of average delivery ratio and minimization of average delivery delay makes the GBD a very special dropping policy.

(4) History-Based Drop (HBD). Sustaining full and wide network knowledge is much difficult and impossible in real scenarios. That was the reason that global knowledge-based drop policy (GBD) failed in a real sense as it cannot be implemented easily. [22] introduced another message drop policy called history-based drop (HBD). In contrast to (GBD), (HBD) uses local information of the network to calculate its parameters and other useful information. HBD is similar to GBD in few aspects like it must have the information of deterministic nodemobility, and it also have the mandatory requirement of unlimited bandwidth and as well as the requirement of identical message size.

(5) Evict Most Probable First (MOPR). In DTN, a node may contain one or more than one messages at a time. Every message has its own local information like its (TTL) and size. (MOPR) is adding another variable to the forwarding message naming it the forwarding predictability (FP) value. This (FP) value starts from zero (0) and getting increments as far as the message travels in the network. This means that when buffer overflow occurs (MOPR), it will select this (FP) value as a parameter and will target a message for deletion having greater (FP) value amongst all. This mechanism produced

quite reasonable results but this much statistical calculations and values were much hard to sustain [23].

(6) Evict Least Probable First (LEPR). As discussed, MOPR used the high FP value for selection of message to drop. As opposed to MOPR, [24] introduced a new policy known as "Evict Least Probable First" (LEPR). It also first calculates the FP value like MOPR, but in contrast to MOPR, it drops the message with the least FP value [25].

3. Related Work

In DTN, congestion is occurred due to nonavailability of end-to-end connectivity, node mobility, energy constraints, and limited buffer space. Due to these reasons, the message is not delivered to destination on time. To avoid congestion, various drop policies have been adopted which drops the message from buffer to vacate a space for the incoming messages.

According to the scheme proposed in [26], the lengthiest message should be dropped. The purpose behind is the less possibility of messages to be forwarded which stay for a longer time inside the buffer, or it has already been delivered. Another advantage of this scheme is its ability to accommodate a large number of incoming messages with only fewer drops.

Evict Most Forwarded First (MOFO) is suggested by [24], as the name suggests attempts to increase the dissemination of messages over the network by dropping the message which has been sent the most number of times. This enables messages with a lower hop count to travel further within the network. MOFO which keeps track of numbers of times a message is forwarded and message of the highest number will drop and then it provides a good chance for the least forward message to be forwarded.

Drop largest (DLA) proposed in [19] is a popular drop policy which utilizes the size of message as an attribute for selection of messages to drop upon a full buffer. Another advantage of this scheme is its ability to accommodate a large number of incoming messages with only fewer drops. This gives small-sized messages more chance to be forwarded.

Another group of buffer management policies discussed by [27] requires some or complete knowledge about the network (e.g., amount of nodes in the network, quantity of replicas of message, and meeting ratio among two nodes), to make their message discarding decision.

RAPID or resource allocation problem in DTNs was proposed by [22], and it is used to implement a buffer management scheme that considers network wide information. It is an empirical approach which focuses on locally optimizing minimal utility, i.e., the expected delay in utility per message used.

Research conducted in [23], a decent buffer management strategy named global knowledge-based drop (GBD), proposed an idea which focuses on messages in the buffer of a node via overall information about the network. It uses some statistical knowledge to develop per message utility based on node's contact history. Furthermore, GBD works upon global knowledge about the network state. As global knowledge is problematic to acquire, hence GBD is difficult to implement. A deployable variant of GBD called the history-based drop (HBD) uses local information to evaluate global values. Being a variant of GBD, it too has unrealistic norms about the state of the network like deterministic node mobility, unlimited bandwidth, and same message size. This limits the practical implementation of this approach severely.

The authors in [28] have explained message scheduling policies, message dropping policies, and buffer management methods. In any node/router, if a channel is not available, then to ensure the right mobility of the message router/node is required to maintain some queue. Their strategies are FIFO which prefers to deliver messages on a first come first serve basis.

According to the research conducted in [18], an effective mechanism has been discussed to drop a message upon buffer overflow. This paper focuses on sized-based mechanisms. The idea given by the authors of this paper is named as weight-based drop (WBD) policy. Whenever the message arrives at a destination node and it has not found a free space in the buffer, so this mechanism will drop the one with the largest size among all in order to vacate a space for incoming messages.

Researchers in [29] elaborate the importance of dropping messages and buffer management in DTNs. Drop policies, iterative replication, and storing of messages produce congestion which is over hidden by dropping messages. The author has emphasized upon the fact that effective buffer management must be there to make decisions for dropping messages. Whenever the buffer becomes overflow, it needs to drop some messages in order to make room for new ones.

In [30], researchers have suggested a message drop policy with average latency, message delivery, and network overhead called max hop count (MHC). In this drop policy, a message stays for some time on each node in the network moving from source towards destination. Each message has a specific hop count which shows the number of nodes from which the message has been passed moving toward the destination. Maximum hop count means that a message has been passed from many nodes and has leaved replicas on each node, and so dropping such messages may not affect its delivery. Low hop count means that a message has passed from fewer amounts of nodes during transmission from source to destination and has leaved less number of message copies in the network, and dropping of such messages may lead poor delivery ratio. The movement model used for this purpose is random walk. The routing protocols for max hop count are Epidemic and PROPHER V2. The experimental results show that it works better than other message drop polices such as LIFO, FIFO, and MOFO in terms of delivery rate and overhead ratio with high TTL which is above 180 minutes and is much realistic than other polices, and the TTL of other polices in realistic environment may reach to days.

[31] proposed an effective buffer management policy which is called size-aware drop (SAD). SAD tries to improve the buffer consumption and reduce unnecessary message drop. It improves the data delivery ratio by determining the requirements based on discrepancy of recently arrived

message and existing free space. It picks the appropriate message for drop in order to minimize overhead. Due to appropriate message selection, the delivery probability of SAD is better than that of other polices. The authors have used two movement models, i.e., Random Way Point and Disaster. The performance of the SAD is better than DOA, DLA, and MOFO in terms of delivery rate, overhead ratio, and buffer time average. Its performance is better in the case if the messages available in the buffer have greater or equal size of an incoming message and in case of small messages through which process it selects more than one message to accommodate an incoming large message.

[32] suggested a new message drop policy for buffer management called novel buffer management scheme. It works on two parameter,s i.e., hop count and TTL. In this mechanism, a buffer is logically partitioned on hop count and threshold value. The messages with low hop count than threshold value are stored in the buffer according to their hop count, and messages with high hop count than threshold value are stored in the buffer according to their TTL values. The messages with low hop count in the buffer are to be transmitted first because these messages have been passed from fewer amounts of nodes and still need long movement to reach to the destination. Dropping of such messages may cause poor delivery ratio. The messages stored in the buffer with TTL valued are to be dropped first because it has traveled for long time in the network and have leaved many copied in the network. So, dropping of such messages enhances delivery rate. It takes intelligent decision about message delivery and message drop to manage the buffer and select the smart relay for routing. Experimental results show that it works better than other polices like drop least, drop most forwarded, and drop most recently received in terms of high delivery rate and low overhead ratio.

[33] proposed a buffer management policy called space time drop (ST Drop). The routing algorithms used for this work are epidemic-based, probabilistic, and social aware. In this policy, a message in the network with great space and time has high probability to be delivered, and that message from the buffer would be dropped that can accommodate the new arrived message. The space and time coverage is calculated by the formula $\mathrm{STc} = \mathrm{Sc} \times \mathrm{Tc}$, where "Sc" is the coefficient of space and "Tc" is the coefficient of time. The routing strategies for this research work were Prophet and Bubble Rap. The results of the ST drop in terms of reducing overhead and increasing message delivery are high as compared to other message drop polices. It works better for improving the cost effectiveness of the opportunistic network.

[34] proposed a message drop policy under city-based environments for DTN which is called Priority Queue Based Reactive Buffer Management Policy (PQB-R). It classifies the buffer messages into source, relay, and destination lists. An individual drop metric has also been used to each list. The Drop Expire Message Module is activated by the buffer manager to drop a message from the buffer to create space for incoming message if it meets the algorithmic criteria. The new message will be accommodated if the available buffer space is larger than the message size and if not so, then the buffer will activate Reactive Message Drop Module to drop the messages from the buffer until the free buffer space becomes greater than message size, and the new message is accommodated. To drop a relayed message, it uses the metrics TTL, hop count (HC), message size (MS), and created time (CT), while to drop a source message, it uses the metrics TTL, CT, and MS. The shortest path map-based movement model has been used as the movement model, and the routing protocol is epidemic. It works better to reduce latency, message drop, and message transmission and to increase message delivery as compared to drop largest (DLA), LIFO, and MOFO.

[21] presented a message drop policy called Proposed Buffer Management Policy (PBMP) with number of forwards (NoF) and TTL. This policy drops those messages which have high number of forwarding and minimum value of TTL. The main theme of this proposed policy is to enhance the message delivery ratio and minimize the message drop. The proposed policy has been compared with MOFO under Epidemic Routing Protocol, and it has been observed that the proposed policy works better than MOFO. The delivery probability of proposed policy is 13.18% higher than MOFO. This delivery probability is high due to selection of two metrics, i.e., NoF and lowest TTL which delay the message in the network, and chances of delivery is increased, while in MOFO, only the number of forwards (NoF) is checked and drop those messages which has the highest NoF which yields low delivery probability.

In a summary, DTN is referred to such network in which end-to-end connectivity is rarely exists. In such networks, store-carry and forward paradigm is used to achieve a successful sending of messages to destinations. To maximize the delivery prospect in a DTN, it is critical to drop such messages upon a full buffer which are the least likely to be delivered to the final destination. Several buffer management schemes have been proposed and discussed in this section which considers various message attributes in order to select the appropriate messages to drop. A summary of common message drop policies is shown in Table 1. The proposed scheme SS-Drop implies a simple but intelligent mechanism to determine the inception size in order to select a message to be dropped as the buffer becomes overflow.

4. Proposed System Model

Recent studies state that whenever a new message arrives and its size is greater than the available free space in the receiving node, some messages need to be dropped in order to make room for the upcoming message. Every time dropping messages leads towards the wastage of valuable resources which the message has already consumed.

To overcome this problem, it is important to control excessive dropping of messages. For achieving this goal, different mechanisms have been proposed to determine which message has to be dropped. Like some researchers have agreed upon to drop the message with a short time-to-live (TTL). Other groups of researchers have suggested a scheme to drop such messages which traveled for a long time in the network. Later on studies have selected size as criteria for

Table 1: Summary of common message drop policies.

Message drop policy	Policy type	Major contributions
Drop front (DF) [16]	Policy without network knowledge	Drop front (DF) uses the first-in first-out (FIFO) strategy to drop messages in case of buffer overflow.
Drop last (DL) [16]	Policy without network knowledge	Unlike drop front (DF), drop last (DL) uses the last in first-out (LIFO) technique to drop.
Drop random (DR) [17]	Policy without network knowledge	Randomly select messages for drop to show unbiasedness.
Drop youngest (DY) [18]	Policy without network knowledge	Dropping of messages whose TTL is long.
Drop oldest (DOA) [16]	Policy without network knowledge	DOA selects the message with maximum residential time in the buffer for dropping.
Drop largest (DLA) [19] [20],	Policy without network knowledge	DLA policy drops the message with the largest size from the buffer as the buffer gets overflow.
Evict most forwarded first (MOFO) [21]	Policy with network knowledge	MOFO keeps the message sent counter for the sent messages, and the message with the highest sent counter value is dropped first.
RAPID's utility-based drop [22]	Policy with network knowledge	RAPID or resource allocation problem is a policy that considers network wide information. The schemes emphasize on a locally optimizing variable which is the expected delay in utility per message used.
Global knowledge-based drop (GBD) [23]	Policy with network knowledge	This policy focuses on the overall statistical information of messages. Moreover, GBD requires global knowledge of the network state.
History-based drop (HBD) [22]	Policy with network knowledge	The history-based message drop policy uses local information to calculate a comprehensive value for the selection of message to be dropped.
Evict most probable first (MOPR) [23]	Policy with network knowledge	A forwarding predictability (FP) value is assigned to every message. MOPR selects those messages for dropping that possesses the highest FP value.
Evict least probable first (LEPR) [24, 25]	Policy with network knowledge	LEPR selects a message to drop which has the minimum FP value.

dropping messages which can favor in avoiding excessive drops and also decreases undue retransmissions which contribute to small delivery probability and greater overhead due to increased resource consumption [35].

The proposed solution is a size-based policy which determines an inception size for the selection of messages to be deleted as buffer becomes overflow. The basic theme behind this scheme is that by determining the exact buffer space requirement, one can easily select a message of appropriate size to be discarded. By doing so, unnecessary message drop can be overcome and ignore biasness just before selection of specific sized messages.

4.1. Spontaneous Size Drop (SS-Drop). The proposed scheme Spontaneous Size Drop (SS-Drop) implies a simple but intelligent mechanism to determine this inception size. Figure 2 shows the model for estimation of inception size for message selection. Whenever the buffer receives new message, it starts analyzing the size of new message. If the arriving message's size is greater than the existing free space in the buffer, this scheme then follows a few simple steps. Firstly, it calculates the difference among the size of incoming messages and the available buffer space. By doing so, it estimates an inception value which it considers as the inception size. Now, any message residing in the buffer of the receiving node, which is equal to or greater than this inception size, is selected to drop.

$$\text{UBS} = \text{Total Buffer Size} - \text{Occupied Buffer}, \tag{1}$$

$$\text{Inception Size (IS)} = \text{Message Size} - \text{Unoccupied Buffer Size}. \tag{2}$$

Equation (1) calculated the value of unoccupied buffer size (UBS) by subtracting the value of the occupied buffer from that of total buffer size. In Equation (2), the value of an inception size will be calculated by subtracting the already calculated unoccupied buffer size from the size of incoming message. This calculation will help the DTN router to drop such messages from the saturated buffer whose size is equal to the inception size.

4.2. Working of SS-Drop. The flow chart in Figure 3 explains the working of a proposed policy that how it will calculate the inception size of a message. And then on the basis of this inception value, how it will drop the message upon buffer's overflow.

First of all, the proposed model will check the size of an incoming message and then calculate the unoccupied size of the buffer. On the basis of these calculations, if the incoming message size is less than that of unoccupied size, it will allow the message to buffer, but if this expression becomes false, the proposed model then calculates the inception size. After that, the inception message size will be checked with

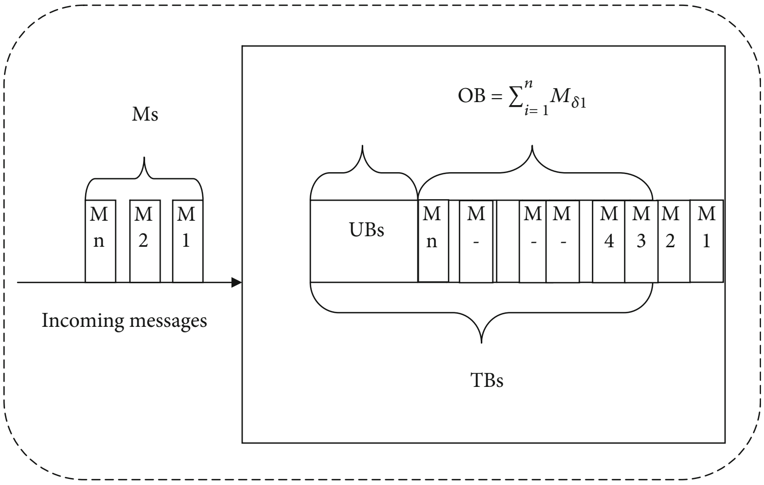
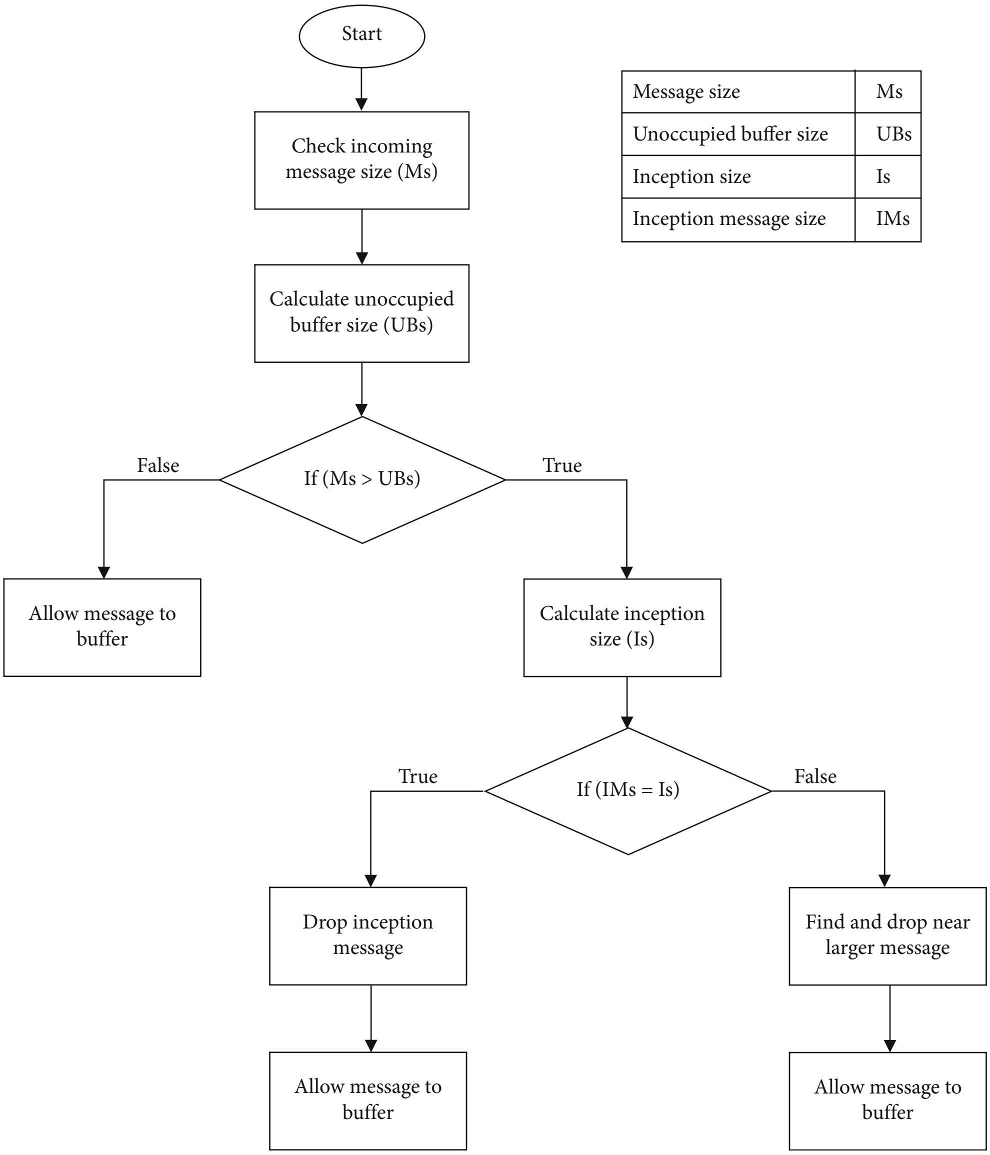


FIGURE 2: Proposed model for estimation of the inception size for message drop.



Message size	Ms
Unoccupied buffer size	UBs
Inception size	Is
Inception message size	IMs

FIGURE 3: Flow chart of proposed policy.

```
Input = $Xi$
Output = $Xj$, >= $j$
$Xi$ (Incoming message)
for i =1 to N do
        Compute $S_i.X_i$ (check size of incoming message)
end for
for i =1 to k do
        Compute $S_i.X_i$ (Check size of incoming message)
        $OB_i = S_i.X_i + OB_i$ (Occupied buffer size)
        $UB_i = TB_s - OB_i$ (Unoccupied Buffer Size)
end for
for j =1 to k do
        if$X_i < UB_i$then
            Enqueue $X_i$
        else
            Compute $I_s = S_i.X_i - UB_i$
            for I to n do
                if$IS_i == S_i.X_i$then
                    Drop $X_i$
                    Enqueue $IS_i$
                else
                    $Sub_{p}roced(IS_i,S_i.X_i)$
                end if
            end for
        end if
end for
```

Algorithm 1: Algorithm of SS-Drop policy.

```
Input = $IS_i,.X_i$
$S((S_i.X_i)$
if $((S_i +1 X_i +1) > IS_i)$ then
        Drop $X_i$ +1
        Enqueue $X_i$
end if
```

Algorithm 2: Algorithm of sub-NS.

the calculated inception size. If the message size and the inception size become equal, the mechanism will drop the inception message; otherwise, it will drop that message which is of near larger size than that of the inception size.

4.3. Pseudocode of SS-Drop. Algorithms 1 and 2 represent the algorithms of Spontaneous Size Drop.

5. Simulation Setting

In order to compare the efficiency of proposed novel message drop policy, i.e., Spontaneous Size Drop (SS-Drop) with other well-known traditional message dropping strategies (drop front, drop oldest, and drop largest), the simulation setup has been done by selecting Epidemic as a routing protocol due to its flooding nature of forwarding messages. The mobility model used in this research work is the shortest path map-based movement mobility model (SPMBM). FIFO has been used as a scheduling scheme throughout the simulation. Table 2 is summarizing the parameter list.

Table 2: Simulation parameters.

Parameters	Values		
	Set 1	Set 2	Set 3
No. of nodes	15	30	60
Size of buffer (MB)	3 MB	4 MB	5 MB
Simulator	Opportunistic network environment (ONE)		
Movement model	Shortest path map-based movement model		
Routing protocols	Epidemic routing protocol		
Group interface	Bluetooth		
Transmission speed	300 kbps		
Transmission range(m)	15 m		
Node speed (m/s)	1-3 (m/s)		
Simulation time	3600 sec		
Message creation interval	15 s-25 s		
Message sizes	100kB-1 MB		
Area of simulation	1600 m × 1600 m		

The parameters used in this simulation is the combination of three groups of nodes. First group consists of 15 nodes with 3 MB of buffer size, second group contains 30 nodes along with 4 MB buffer size, and the third group consists of 60 nodes with 5 MB buffer size. All the nodes have intragroup communication plus intergroup communication as well. Details of other parameters are as follows.

5.1. Epidemic Protocol. Epidemic is considered to be the pioneer protocol designed for communication in a dispersed network where end-to-end connectivity does not exist [36]. Epidemic is a flooding nature protocol. In this protocol, a unique ID is assigned to each and every message, and this ID is associated with the message and all its generated copies until the message has been dropped or successfully delivered to a particular destination. Epidemic sends and forwards the messages in the form of flood to all the nodes within the contact of the sender node, which increases the ratio of successful delivery, but on the other hand, it maximizes the overhead by consuming large amounts of valuable resources.

5.2. Shortest Path Map-Based Movement Mobility Model. Movement models dictate the nodes how to move in the simulation. Each model facilitates the nodes in terms of coordination, speed, and pause time. In this research work, the shortest path map-based movement mobility model is used as the mobility model. It is considered to be one of the mature shortest path models because it works on Dijkstra's shortest path algorithm in order to find out its way through the map [37].

6. Results and Analysis

To evaluate the effectiveness and performance of said messages, drop policies following metrics were used.

(1) Delivery probability

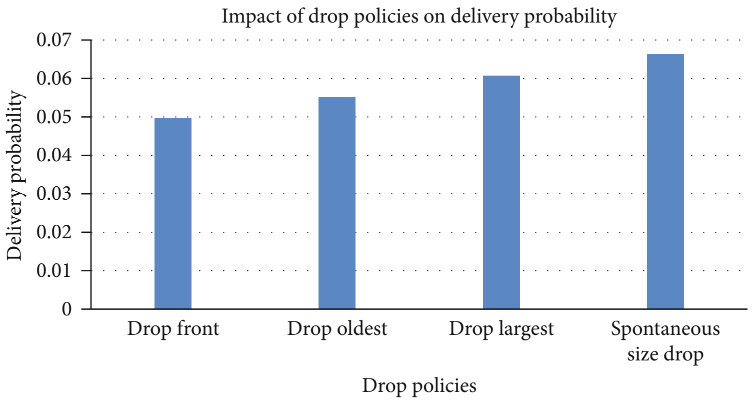


FIGURE 4: Comparison of message dropping srategies w.r.t delivery probability.

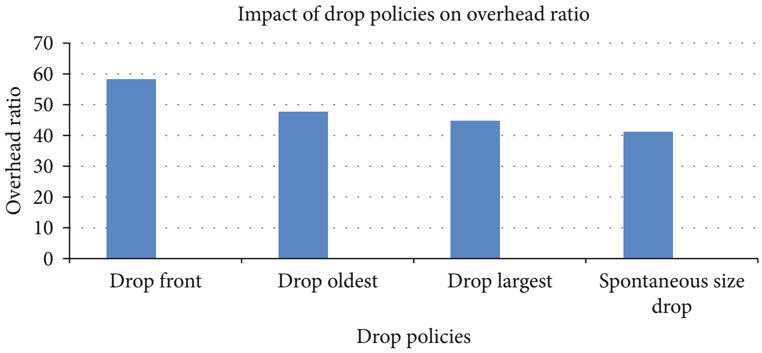


FIGURE 5: Comparison of message dropping strategies w.r.t overhead ratio.

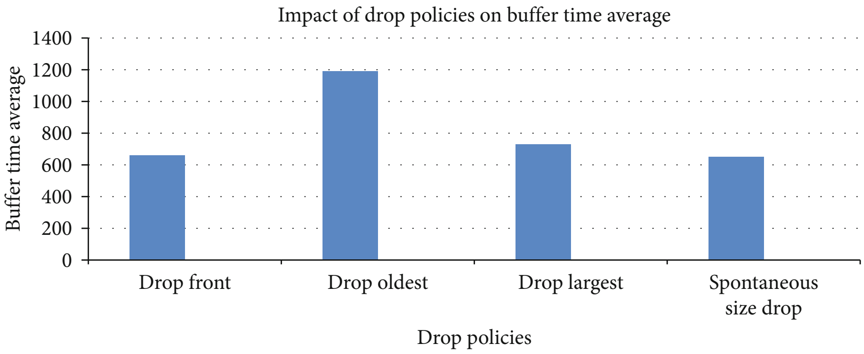


FIGURE 6: Comparison of message dropping strategies w.r.t buffer time average.

(2) Overhead ratio

(3) Buffer time average

(4) Dropped messages

(5) Messages delayed

6.1. Delivery Probability. Delivery probability can be explained as the percentage of the fully delivered messages within a given time period. Basically, it is the ratio of the number of messages delivered over the number of messages created. Delivery probability can be shown in Equation (3).

$$\text{Delivery Probability} = \frac{\text{Total Messages Delivered}}{\text{Total Messages Generated}}. \quad (3)$$

Figure 4 shows the effect of dropping strategies on delivery probability. It is seen that by applying a flooding nature Epidemic protocol, drop front policy gives a decent probability value of 0.496, while drop oldest and drop largest is slightly better and, respectively, gives the results 0.552 and 0.608, respectively. However, epidemics with spontaneous size drop due to its intelligent message dropping mechanism outperform all other dropping strategies by achieving the improved 0.663 delivery probability value.

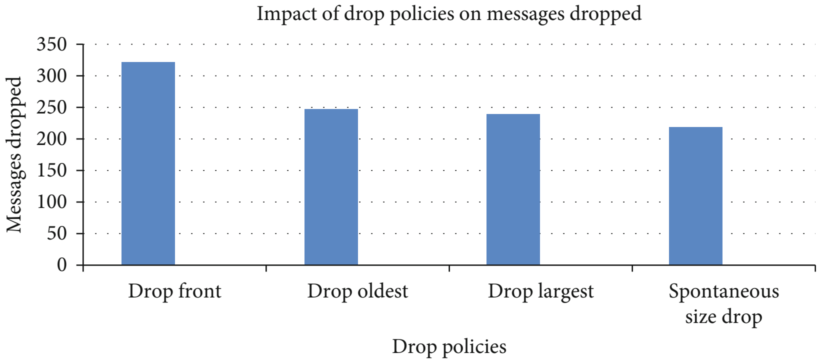


Figure 7: Comparison of message dropping strategies w.r.t messages dropped.

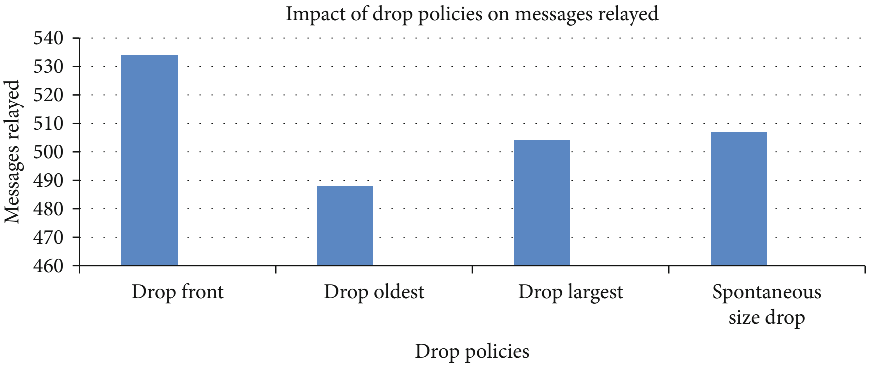


Figure 8: Comparison of message dropping strategies w.r.t messages relayed.

6.2. *Overhead Ratio (OR).*

The word overhead ratio determines the percentage of total messages communicated (TMC) to the total messages delivered (TMD). Mathematically, it can be represented as in Equation (4).

$$\mathrm{OR} = \frac{\mathrm{TMC} - \mathrm{TMD}}{\mathrm{TMD}}. \tag{4}$$

To minimize the overhead ratio is also a vital metric of this research work. The key objective is to maximize the delivery probability opposite to minimize the overhead ratio. Figure 5 shows that by applying intelligent drop policy, overhead ratio can be reduced. The graph depicts drop front that has a high overhead ratio of 58.33 as compared to values of drop oldest 47.80 and drop largest 44.82. Message redundancy can be overcome by intelligently selecting messages, which results in minimizing overhead. The spontaneous size drop shows by giving optimized 41.25 overhead ratio.

6.3. *Buffer Time Average.*

Normally, time taken by the message in the buffer at every node is referred to as average buffer time. It can be calculated by using the formula shown in Equation (5).

$$\text{Buffer Time Average} = \frac{\sum ni = 1 \text{ Buffer Stay Time of Message}}{\text{Total Messages in the Buffer}}. \tag{5}$$

Figure 6 analyzed the effect of existing policies and proposed Spontaneous Size Drop (SS-Drop) with respect to buffer time average. Epidemic with drop oldest policy allows messages to stay for a longer period of time in the buffer in order to achieve greater buffer time average. Epidemic with drop front, drop largest, and proposed scheme Spontaneous Size Drop achieves relatively smaller buffer stay time average. The reason is that drop front iteratively drops messages while both size-based schemes drop messages based on the requirements of buffer space. None of these schemes take time into consideration due to which relatively newer messages can be dropped as well. Thus, as a result, the overall buffer time average of messages is relatively smaller as compared to drop oldest. However, in this scenario, against popular belief, buffer time average does not seem to have much effect on the delivery probability.

6.4. *Messages Dropped.*

Redundant message drop is another cause of increased overhead while the prime objective of any buffer management scheme is to control the number of messages dropped. Figure 7 shows that drop front causes the maximum number of messages to be dropped thus increasing overhead. The rest of the schemes (i.e., drop oldest, drop largest, and Spontaneous Size Drop) reduces this message drop significantly leading to reduced overhead ratio. This is due to selective behavior rather than blind drop of messages. SS-Drop has the least message drop ratio among the other buffer management schemes. This could be due to the fact that it has the ability to select intelligently a message to be dropped.

6.5. Messages Relayed. Figure 8 shows the effect of each scheme on the number of messages relayed in order to achieve successful delivery. Drop front has the lowest delivery probability with the highest overhead ratio which may have resulted from its maximum number of message relaying. Drop oldest has the least amount of messages relayed due to longer buffer time average. Drop largest and Spontaneous Size Drop closely follow behind.

7. Conclusion and Future Work

This research proposed an efficient buffer management policy Spontaneous Size Drop (SS-Drop) which estimates the difference of the available buffer size and the size of the incoming message and then determines an inception size for message discard.

To evaluate its performance, an Epidemic routing protocol is used along with the shortest path map-based movement model. In the shortest path map-based movement model scenario, the proposed policy SS-Drop showed complete superiority over the opponent drop policies drop front (DF), drop oldest (DOA), and drop largest (DLA) in terms of maximum probability and minimum overhead ratio. Spontaneous Size Drop also outperforms other drop policies in terms of dropping messages. Due to encouraging results obtained in this research, it is clear that the great potential lies in the inspection and exploration of this area. An interesting future direction is to evaluate the performance of the proposed scheme with other scheduling/forwarding policies. Designing a new scheduling scheme based on size, for further optimization, is an interesting extension to this work.

Data Availability

Data are available upon request to the corresponding authors.

Conflicts of Interest

The authors declare no potential conflict of interests.

Acknowledgments

This study was supported by the China University of Petroleum-Beijing and Fundamental Research Funds for Central Universities under grant no. 2462020YJRC001.

References

[1] R. Fulvio, *Application-gateway in a DTN environment*, 2021.

[2] K. M. Wook and C. Y. Won, "An efficient routing protocol with overload control for group mobility in delay tolerant networking," *Electronics*, vol. 10, no. 4, p. 521, 2021.

[3] Z. Jianguo, Z. Changjia, K. Yuqin, and T. Shenghui, "Integrated satellite-ground post-disaster emergency communication networking technology," *Natural Hazards Research*, vol. 1, pp. 4–10, 2021.

[4] S. Sanhua, "An effective congestion control scheme based on early offload for space delay/disruption tolerant network," *International Journal of Security and Networks*, vol. 16, no. 1, pp. 28–36, 2021.

[5] R. David, J. Arshad, and M. Luca, "Protocol transformation for transiently powered wireless sensor networks," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pp. 1112–1121, Virtual Event Republic of Korea, 2021.

[6] F. Kevin, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27–34, Karlsruhe, Germany, 2003.

[7] R. D. Satya, S. Koushik, M. Nandini, and P. Sinha Bhabani, "Delay and disruption tolerant networks: a brief survey," in *Intelligent and Cloud Computing*, M. Debahuti, B. Rajkumar, M. Prasant, and P. Srikanta, Eds., pp. 297–305, Springer Singapore, Singapore, 2021.

[8] D. K. Lobiyal, "Location based contact time energy efficient routing (LCTEE) approach for delay tolerant networks," *Wireless Personal Communications*, vol. 108, no. 4, pp. 2639–2662, 2019.

[9] M. G. Douglas and M. S. Kumar, "Catora: congestion avoidance through transmission ordering and resource awareness in delay tolerant networks," *Wireless Networks*, vol. 26, no. 8, pp. 5919–5937, 2020.

[10] Q. Ayub, S. Rashid, and M. S. Zahid, "Buffer scheduling policy for opportunistic networks," *International Journal of Scientific & Engineering Research*, vol. 2, no. 7, 2011.

[11] S. Nidhi, P. Sudhakar, and K. Sanjay, "Friendship and location-based routing in delay tolerant networks," in *Evolutionary Computing and Mobile Sustainable Networks*, V. Suma, B. Noureddine, and W. Haoxiang, Eds., pp. 781–789, Springer Singapore, Singapore, 2021.

[12] J. Sweta and C. Meenu, "A fuzzy logic based buffer management scheme with traffic differentiation support for delay tolerant networks," *Telecommunication Systems*, vol. 68, no. 2, pp. 319–335, 2018.

[13] R. Sulma and A. Qaisar, "Integrated sized-based buffer management policy for resource-constrained delay tolerant network," *Wireless Personal Communications*, vol. 103, no. 2, pp. 1421–1441, 2018.

[14] R. Animesh, A. Tamaghna, and D. B. Sipra, "Fairness in message delivery in delay tolerant networks," *Wireless Networks*, vol. 25, no. 4, pp. 2129–2142, 2019.

[15] L. Yong, Q. Mengjiong, J. Depeng, S. Li, and Z. Lieguang, "Adaptive optimal buffer management policies for realistic DTN," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pp. 1–5, Honolulu, HI, USA, 2009.

[16] R. Sulma, A. Qaisar, Z. M. Soperi Mohd, and A. A. Hanan, "E-drop: An effective drop buffer management policy for DTN routing protocols," *International Journal of Computer Applications*, vol. 13, pp. 118–121, 2011.

[17] J. Sweta and C. Meenu, "Survey of buffer management policies for delay tolerant networks," *The Journal of Engineering*, vol. 2014, no. 3, pp. 117–123, 2014.

[18] R. Sulma, A. Qaisar, and A. A. Hanan, "Reactive weight based buffer management policy for DTN routing protocols," *Wireless Personal Communications*, vol. 80, no. 3, pp. 993–1010, 2015.

[19] R. Sulma and A. Qaisar, "Efficient buffer management policy DLA for DTN routing protocols under congestion," *International Journal of computer and Network Security*, vol. 2, no. 9, pp. 118–121, 2010.

[20] S. Rashid, Q. Ayub, M. S. Zahid, and A. H. Abdullah, “Impact of mobility models on DLA (drop largest) optimized DTN epidemic routing protocol,” *International Journal of Computer Applications*, vol. 18, no. 5, pp. 1–7, 2011.

[21] S. V. Kumar and G. Nisha, “Comparision of MOFO drop policy with new efficient buffer management policy,” *International Journal of Innovative Research & Studies*, vol. 8, pp. 456–460, 2018.

[22] B. Aruna, L. Brian, and V. Arun, “DTN routing as a resource allocation problem,” in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 373–384, New York, United States, 2007.

[23] K. Amir, B. Chadi, and S. Thrasyvoulos, “Optimal buffer management policies for delay tolerant networks,” in *2008 5th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*, pp. 260–268, San Francisco, CA, USA, 2008.

[24] L. Anders and S. Phanse Kaustubh, “Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks,” in *2006 1st International Conference on Communication Systems Software & Middleware*, pp. 1–10, New Delhi, India, 2006.

[25] S. A. Pereira, B. Scott, H. C. Massaki, and O. Katia, “DTN congestion control unplugged: A comprehensive performance study,” in *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks*, pp. 43–48, New York, United States, 2015.

[26] J. A. Davis, A. H. Fagg, and L. B. Neil, “Wearable computers as packet transport mechanisms in highlypartitioned ad-hoc networks,” in *Proceedings Fifth International Symposium on Wearable Computers*, pp. 141–148, Zurich, Switzerland, 2001.

[27] K. Sathita and E. Hiroshi, “Independent DTNs message deletion mechanism for multi-copy routing scheme,” in *Proceedings of the Sixth Asian Internet Engineering Conference*, pp. 48–55, New York, United States, 2010.

[28] I. Saeid, “A novel queue management policy for delay-tolerant networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.

[29] G. Animesha, J. Shweta, and B. D. Singh, “An analysis optimal buffer management policy to improve QOS in DTN routing protocol,” *International Journal of Computer Applications*, vol. 975, p. 888, 2012.

[30] H. Youssef and A. Abdelmounaim, “MaxHopCount: a new drop policy to optimize messages delivery rate in delay tolerant networks,” *IJIMAI*, vol. 4, no. 1, pp. 37–41, 2016.

[31] M. Momina, H. Fazle, I. Muhammad, M. A. Ali, and V. Vasilakos Athanasios, “An adaptive and efficient buffer management scheme for resource-constrained delay tolerant networks,” *Wireless networks*, vol. 22, no. 7, pp. 2189–2201, 2016.

[32] C. C. Sobin, “An efficient buffer management policy for DTN,” *Procedia Computer Science*, vol. 93, pp. 309–314, 2016.

[33] M. D. Silva, I. O. Nunes, R. A. Mini, and A. A. Loureiro, “ST-Drop: a novel buffer management strategy for D2D opportunistic networks,” in *2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1300–1305, Heraklion, Greece, 2017.

[34] A. Qaisar, N. M. Asri, R. Sulma, and H. H. Adnan, “Correction: priority queue based reactive buffer management policy for delay tolerant network under city based environments,” *PloS one*, vol. 14, no. 10, article e0224826, 2019.

[35] W. John, C. Vania, and A. M. Dias, “Performance of opportunistic epidemic routing on edge-markovian dynamic graphs,” *IEEE Transactions on Communications*, vol. 59, no. 5, pp. 1259–1263, 2011.

[36] A. Vahdat and D. Becker, *Epidemic routing for partially connected ad hoc networks*, 2000.

[37] E. Frans, K. Ari, K. Jouni, and O. Jörg, “Working day movement model,” in *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models*, pp. 33–40, New York, United States, 2008.