Dynamic Weight Dropping Policy For Improve PROPHET Routing Performance

Abubaker Alhutaba
Department of Computer Engineering
The Collage of Industrial Technology
Misurata , Libya
alhutaba_a_abu@yahoo.com

Salem Sati
Department of Computer Science
University of HHU
Düsseldorf, Germany
sati@cs.uni-duesseldorf.de

Mohamed Eshtawie
Department of Computer Engineering
The Collage of Industrial Technology
Misurata , Libya
eshtawie@yahoo.com

Abstract –Delay **tolerant Network is designed to operate over long distances where latency measured in hours or even days is un avoidable. Therefore focusing on routing protocols performance based on store-carry-forward ignoring physical buffer limitation is highly interested. In this paper, dropping policy termed Dynamic Wight Dropping Policy to improve the PROPHET performance is implemented. The implemented dropping policy is applied on PROPHET version 2 routing protocols that recommend FIFO by default. Different parameters such as message life/buffer time as delay metric and replication/hop count number as overhead metric will be analyzed. The implemented dropping policy is evaluated and compared with MOFO and SHLI policies and the simulation results show an improvement in PROPHET performance.**

Index Terms— **Opportunistic Network, PROPHET Routing, Dropping Policies, Message Information, Routing Performance.**

I. Introduction

The Delay Tolerant Network routing protocols have designed to operate in an environment where there is no stable end to end path. The DTN research has focused primarily on the performance of routing protocols that built on Store-Carry-Forward manner without considering of Physical buffer limitation. Therefore, the need of messages sorting and dropping them according to the priority is very critical in such environment. The DTN routing divided into two main types i.e. flooding-based and utility-based routing. In utility-based routing, messages are forwarded to other nodes based on calculated cost and finally delivered to the destination via multiple hops. Utility-based of protocols highly concerned with the reduction of the number of message replication so that node resources consumption in terms of storage, bandwidth and power is reduced.

There are many proposed routing approaches of utility-based which consider the node buffer limitation by controlling the number of the message copies spread on the network. These protocols still using multi-copies spraying in the network to achieve required message delivery. Unfortunately, this causes serious congestion and consumes nodes' storage space. On the other hand, the flooding-based routing replicate the message copy to every encountered node. This assumption of unlimited of node resources is paid against the redundancy of unlimited message copies. One representative protocol for flooding-based routing is Epidemic routing protocol. In order to solve the message overhead and node resources consumption problem of the epidemic protocol, several utility-based routing schemes have been proposed. Spray and Wait protocol and PROPHET protocol are among these protocols. In PROPHET (probability routing protocol using history of encounters and transitivity) protocol, the delivery predictability cost between two encountered nodes is calculated. the calculations done are based on contact information history between connected nodes. This allows the prediction of the higher delivery which leads to higher probability in future contacts between encountered nodes. In PROPHET protocol, a message is copied to a contact node only when the delivery predictability cost to the destination node of encountered node is larger than that cost of the node carrying the message. By doing this, PROPHET protocol achieves good delivery probability as well as satisfying low overhead. Clearly, PROPHET forwarding strategy reduces the number of message copies for minimizing the node re-sources consumption in terms of Storage, Bandwidth, and power. Based on the idea of PROPHET utility function to improve the performance of the protocol, it is still assumed that the nodes have infinite buffer size. Therefore, the drop policies play a very important role in the opportunistic network performance. Here the physically limited buffer in each node should use reasonably. Furthermore, the question of how to design an effective drop policy in opportunistic networks becomes a challenge. Obviously, most of the buffer management research focused on one parameter of message mobility. This refers to information according to buffer policies which are an index of scheduling or dropping priority. Currently, most dropping policies of buffer management, such as drop tail, drop front and drop largest policies are performing insufficiently in opportunistic networks. This is because each one of these drop policies consider one of performance metrics since it takes one parameter as decision criteria.

II. RELATED WORK

Increasing network performance without effecting the delivery delay is the aim of many drop policies for opportunistic networks. Current drop policies concentrate on the message information or node mobility in case of buffer congested. However, these policies do not care about the dynamic behavior of the environment when decide to drop the messages from the node's buffer. Clearly, these policies do not reduce drop rate as needed by the DTN application. One of the traditional buffer management policies is Drop Front, which index the queue based on FIFO order (sort messages oldest first). Ruan [1] has evaluated the performance of MOFO drop policy with different routing protocols. He proved that MOFO improves the performance of the opportunistic networks by increasing the delivery rate, and minimizing the over overhead. Lindgren [2] studied few collections of available buffer management policies for PROPHET. Their work show that by selecting suitable buffer management policies the performance of routing can improve. They concluded that MOFO policy gives better performance metrics when comparing to existing FIFO and SHLI. Krifa [3], proposed Global Knowledge-based Scheduling Drop (GBSD) that requires global network state. Therefore, GBSD is complicated to be practical in actual environments. Ruan [1] proposed in his paper a comprehensive integrated buffer management, which takes all information relevant to message delivery and network resources into consideration. Moreover, it focused on a subdivision of the field, such as queuing strategy, cache replacement, or redundancy removal. Shin [4] proposed an approached for buffer management, this work used the estimated total number of replicas of a message and provided improved performance. This study proposes an enhanced buffer management policy that utilizes message properties. It considers two utility functions by message properties i.e. number of replicas, the age, and the remaining TTL. Rashid [5] proposed a buffer management, which takes message sizes into account. This study focuses on the local knowledge-based policies to minimize the drop of the messages dynamically and maximize the delivery ratio. The weight criteria of the message calculated based on message's properties, which are message size, remaining TTL, message buffer time, hop count, and replication count. The limitation facing the dynamic topology problems is the complexity in getting global knowledge information in DTN environment. This issue makes local knowledge-based policies take more effect than the globally based information. Therefore, more attention is paid to local knowledge as a part of global information. This information considered as the base of drop policy for controlling dynamic drop policy.

III. PROPHET ROUTING PROTOCOL

PROPHET is one of DTN routing protocols which use the cost to forwarded the message. It is a Probabilistic Routing Protocol [6],[7] and is updated from version1 that based on the event of the contact to version2 which consider contact information such as duration, frequency and history. This section considers the updated version two. In general, PROPHET uses the node encounter history and the transitivity as metrics to calculate predictability. The delivery predictability computed at each node for a known destination to select the best next-hop of the buffered message. Hence, the node with frequent encounters with stable connectivity holds high probability to encounter again in near future. Equation 1 calculates the nodes delivery probability.

$$P_{(a,b)} = (1 - P_{(a,b)_{old}})P_{enc} \qquad (1)$$

Where:

$P_{(a,b)}$: Probability node a meet node b

$P_{(a,b)_{old}}$: Old probability of node a to meet node b

P_{enc} : Encountered probability (meeting probability)

In addition PROPHET use aging that describes the node with large elapsed encountering interval so that not to be encountered in the future. Hereby, the delivery probability between peer wise of nodes is aged, which decrees the delivery probability based on the time interval and aging constant gamma (γ , this factor changed based on elapsed time between encountered nodes). The aging constant determines the reduction rate and is represented as in equation 2.

$$P_{(a,b)} = P_{(a,b)_{old}} \gamma^{k} \qquad (2)$$

Where K: Number of time slots between encountered nodes. The third part of PROPHET routing calculations is the transitivity, which describes how to calculate the transitivity between three nodes A, B and C.

$$P_{(a,c)} = (1 - P_{(a,c)_{old}})P_{(a,c)_{old}} P_{(a,b)} P_{(b,c)} \beta \qquad (3)$$

$$P_{(a,b)} = (1 - P_{(a,b)_{old}})P_{enc}$$

Where β : Transitivity factor used to bound the probability between three nodes a, b and c. In general, PROPHET routing considering tradeoffs between its performance in terms of delivery ratio, delay, and node resource limitations in terms of energy and storage.

IV. PROPOSED APPROACH

The main aim of buffer management policies is to maximize the global delivery ratio of the network. Therefore, the proposed drop policy which determines its drop decisions based on the message information and mobility pattern of nodes contacts. Clearly the proposed Dynamic Wight Drop Policy (DWDP) designed based on the utility function of the message. This utility based on per-message and per-node information. The proposed dynamic Wight drop policy use the utility value of the messages by the node to select a message when the node buffer is full. The utility value of a message woes calculated based on local information. Since all variables in the utility function assumed to be local information of the node. In the context of DTN, it is difficult to find the global information of message copies in the network. One possibility is to use the local information of nodes that approximates the number of message copies disseminated at an instant time in the network. This information help to decide when and which message woes dropped and selected to drop. Because the limited buffer size of the node, message copies eventually should be discarded so that new coming messages can be stored. This drop message decision made by the intermediate node based on its local information. On the other hand, there is the other drop decision which determined by the message originator, where source determines the value of TTLr (remaining Time To Live) when message created, Normally, copies of the message will be dropped when its TTL_r is expired. If the TTL expires before the message carrier encounters any nodes, then the message copy will drop, this assumption gives an idea of that may (buffering time) $T_{BUF} = TTL_r$, because in some scenario the message with the longer buffering time at the intermediate node is not always to be the message that has replicated many times. Therefore, we would like to give the message higher chance of delivery if there is a free buffer space at the node. This is the main idea of normalizing by the ratio of message TTL only as shown in the utility function. In addition, to increase the delivery the system should minimize the delay and overhead of the message, the drop policy will discard the message with maximum U(m) value.

$$U(m)_{DWDP} = \frac{TTL_e}{TTL}(1 + H_c \frac{T_{BUF}}{TTL_r} + R_c T_{BUF})$$

Where:

$U(m)_{DWDP}$: Utility function of the proposed DWDP.

TTL_e : Elapsed Time of TTL.

H_c : Hop count.

R_c : Replication count.

The function consider the time and delay of the message, which caused by the mobility pattern of the node movement, this part of $\frac{TTL_e}{TTL}$ consider the delay from the source as TTLe and the message life at the intermediate node as message carrier R, the two delay values divided by the message life TTL to detect any shortest life message in the buffer because those the main delay component of the end to end delay, therefore it related to the message life which is the maximum delay allowed by the message originator TTL.

The other part $(1 + H_c \frac{T_{BUF}}{TTL_r} + R_c T_{BUF})$ is related to the overhead of the message, because we believe that the hop count of the message is the overhead from the message source to the current message carrier R, on the other hand, the number of transmissions (replication) will impact of over the entire network overhead.

V. EXPERIMENT AND RESULTS

The proposed DWDP approach is compared with the commonly used drop by DTN approaches. Among these approaches, MOFO, FIFO, and SHLI are considered and their performance is evaluated and compared with the proposed approach. These drop policies are popular and frequently used in routing protocols for opportunistic networks. The following metrics are used in the comparison.

- ***Delivery Ratio*** is the number of delivered messages to the total number of generated messages.
- ***Overhead Ratio*** is the average number of intermediate nodes used for one delivered message.
- ***Average Delay*** is the average delay of all messages delivered successfully.

A. Data and Experimental Settings

The performance evaluation is performed using the Opportunistic Network Environment (ONE) Simulator [8],[9]. The ONE Simulator is applied to compare the proposed DWDP with the three representative dropping policies: FIFO, SHLI, and MOFO. In addition, a simple drop policies which either maximize or minimize a set of message-related parameters to decide which message to drop is considered. The following message related parameters are considered:

- **Arrival time**: The time at which the message arrived at the current node. FIFO drops the message which was received first.
- **TTL**: The remaining time to live of the message Equals SHLI if minimized i.e. dropping the message with smallest TTL.
- **Hops**: Number of hops the message passed from the source to the current node.
- **Replication Count**: Number of copies of this message that were spread previously by the current node.

As the TTL and the workload has a very large influence on the quality and costs, we varied the TTL between 5 values, which are 100, 200, 300, 400 and 500, as well as we varied the workload to match different traffic patterns. The simulation setup is presented in Table 1. The different drop approaches performance evaluated under different message TTLs shows the performance of the selected drop policies under increasing congestion and increasing expected delay.

TABLE 1: SIMULATION SETTINGS

No	Settings	Map of downtown Helsinki, Finland
1	Simulation time	12h
2	Number of devices (n)	126
3	Group Type / speed	80 Pedestrians (0.5 to 1.5km/h) 40 Cars (10 to 80km/h) 6 Trains (10 to 80 km/h)
4	Simulation area	Helsinki, Finland Map
5	Routing protocols	PRoPHET V2 with GTRTMax
6	Interface type	Simple Broadcast
7	Transmission range	250 m
8	Bandwidth	250 KBps
9	Drop policies used	FIFO, MOFO, MinTTL(SHLI), DL, DDP
10	Message sizes ranges	Scenario 1: 500 KB - 1 MB Scenario 2: 64 KB - 512 KB & 512 B - 2 KB Scenario 3: 1 MB - 5 MB & 64 KB - 512 KB & 512 B - 2 KB
11	Message creation interval	Scenario 1: 25s - 35s Scenario 2: 25s - 35s & 1s - 5 s Scenario 3: 60s - 120s & 25s - 35s & 1s - 5s
12	Time-to-live (TTL)	100, 200, 300, 400, 500 min
13	Default buffer size	Pedestrians = 5 MB, Cars = 50 MB, Trains = 50 MB

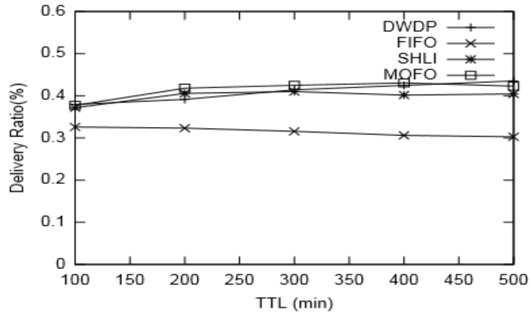
Furthermore, the comparison with the increase of message TTL means that the replication probability of all messages will increase. Therefore, the congestion and drop rate (overhead) in addition to delay will rise gradually. To achieve the largest delivery rate for all different TTL values, attention needs to be paid to delay and overhead. This is reasonable as a large TTL of a message gives more time for message copies to be buffered in intermediate nodes with free space without being discarded. This may help in positive side to increase the delivery rate. In negative side while the long-time buffering in forwarding nodes will decrease the available buffer spaces and increase the average overhead and delay. Normally when buffer spaces runs out, the message copies will be dropped based on the policy criteria. This will lead to a degrading in delivery ratio.

B. Numerical Results

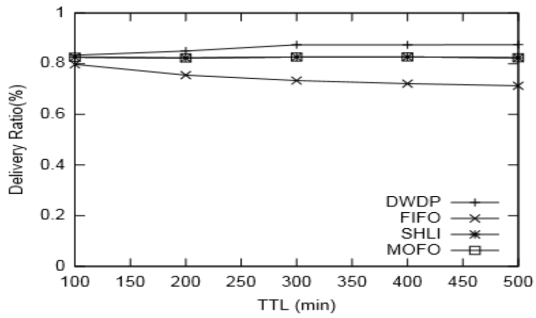
In this paper, 60 simulations were carried out within the ONE simulator. Four dropping policies, 3 scenarios and 5 TTL values, for comparing the proposed Dynamic Weight Drop Policy with other dropping policies used with PROPHET v2. The results for the metrics Delivery Ratio, Delay and Overhead Ratio are depicted over time in Figures 1, 2 and 3 for the three scenarios. The Dynamic Dropping Weight Policy is compared with MOFO (MOst FOrwarded) and SHLI (SHort LIfe)and FIFO (First In First Out) with detail in this section. The performance of the proposed drop policy is depicted and analyzed for different values of traffic rate and message life time TTL.

The delivery ratio is the main indicator of the network performance. Corresponding figures for the three Scenarios are depicted in Figure 1. When considering the delivery ratio, the results show that only the DWDP drop policy achieves good results compared with SHLI and MOFO. The FIFO policy yields also acceptable results but do not achieve consistently good results compared with DWDP, SHLI, FIFO and MOFO. Figure 1 shows the results of DWDP, SHLI and MOFO. For Scenario 2 and 3, it is observed that only a maximum relative difference in the delivery ratio of 5.1766%. For Scenario 2, TTL=500s, where DWDP is able to deliver 5.1766% more messages than MOFO.

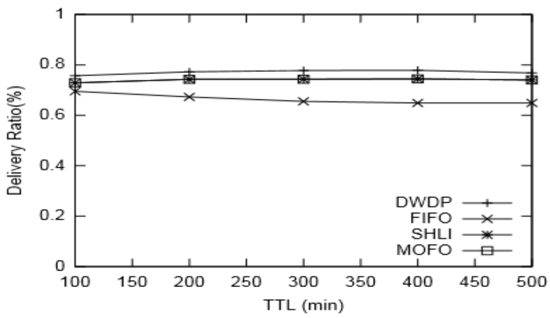
In general, the differences in the delivery ratio variant based on the compared policy criteria. In Scenario 1 of Figure 1 (a), MOFO and SHLI result in up to 12.45% more delivered messages compared with FIFO. The delivery ratio of proposed DWDP has a maximum of 2.81% delivered messages with MOFO in comparison to 4.18% delivered messages for SHLI in Scenario 1. The delivery ratios are rather low here for FIFO dropping policy.



a- Scenario 1



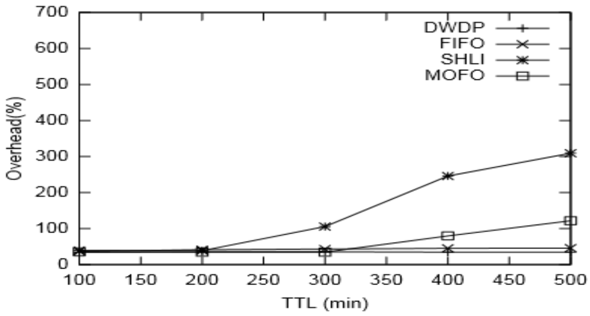
b- Scenario 2



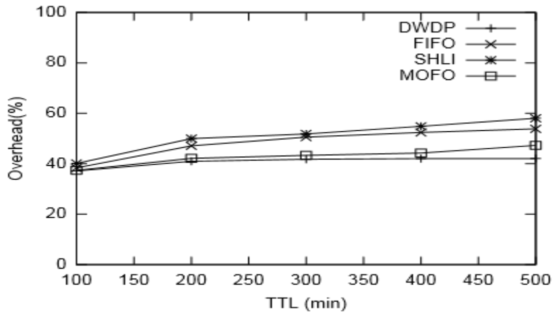
c- Scenario 3

Figure 1: Delivery Ratio

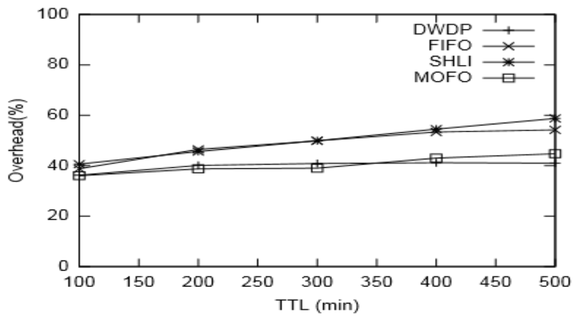
This is due to the low mobility of the nodes and potentially large distances to travel for a message for a successful delivery. Figures 1(a, b, c) reflecting the delivery ratio for the various TTLs and scenarios. SHLI and MOFO are very close to DWDP in having a maximum 2.2251% more messages delivered in Scenario 2 and 3 than MOFO and 10% better than SHLI drop policy. It may not give clear indication, where with some cases we got quite improving in delivery ratio. At the same time we pay a lot of network resources (overhead and delay) for this small improvement in the delivery ratio. Therefore, as Figure 1 show the delivery ratio of the scenario 1 with variant message TTL for PRoPHET routing protocol using FIFO, SHLI and MOFO in comparison with the proposed DWDP policy. For Scenario 1, a larger TTL parameter, e.g. TTL=500s, results in a better performance of the proposed DWDP policy. Also the performances of SHLI and MOFO are still lower in that case. The difference in the delivery ratio between proposed DWDP policy regarding SHLI and MOFO are around 3, 4 % respectively. This is coming from the dynamic behavior of proposed DWDP policy, because as TTL of the message increases or traffic pattern rate increases the congestion or drop rate is increased. This higher drop rate shows that the delivery rate of the proposed DWDP will better than the SHLI and MOFO policies. This is clear in other scenarios 2 and 3 Figures 1(b) and 1(c). While the performances in scenario 1 is close similar on a high level, the costs are very different These two figures 1(b) and 1(c) show that the difference in delivery ratio between proposed DWDP policy regarding SHLI and MOFO are 10% instead of 4% in Figure 1(a) of the scenario 1.



a- Scenario 1



b- Scenario 2

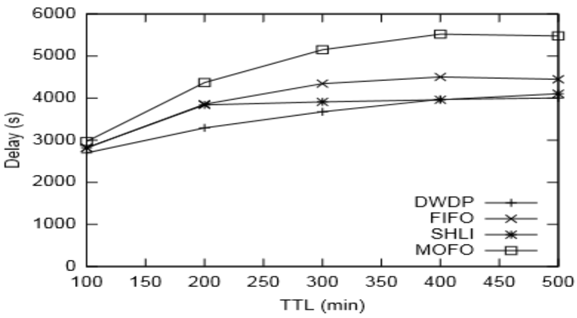


c- Scenario 3

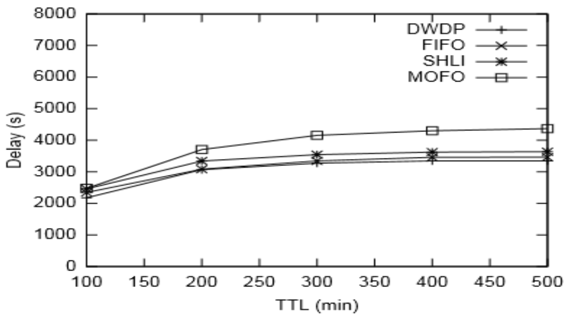
Figure 2: Overhead Ratio

The overhead ratio for varying the TTL of the messages and the traffic rate through the scenarios is shown in Figure 2 for the different Scenarios 1, 2 and 3, respectively. Here it is worth while to mention that DWDP has the lowest traffic overhead together with MOFO in Figure 2(b) and 2(c), while the other policies including SHLI are far from the optimum. Figures 2(a), 2(b) and 2(c) show that the overhead of SHLI grows greatly in Scenario 1. Figure 2(a) shows that in Scenario 1 with increasing TTL both policies SHLI and MOFO are affected.

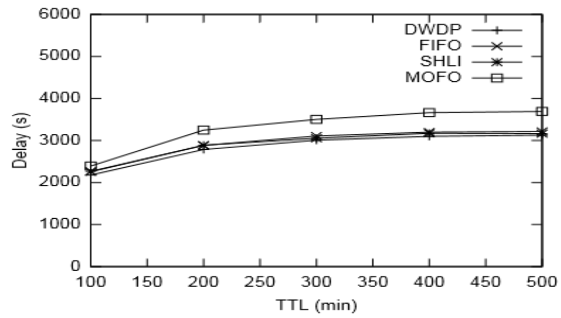
In Scenario 1 with TTL=500, SHLI produces enormous traffic overhead of up to 309.3. MOFO produces an overhead of 121.68%, while DWDP remains throughout the scenarios and TTL variations at 34.046% to 42.073%. MOFO requires in best case 4.5% less traffic overhead than DWDP but in the worst case 2.5 times more. The overhead ratio of the proposed drop policy DWDP stays with few exceptions always less than all compared policies FIFO, SHLI and MOFO at a very constant level of around 38-40%.



a- Scenario 1



b- Scenario 2



c- Scenario 3

Figure 3: Delay

This is as DWDP considers the overhead, in terms of Hop Count and Replication Count, for the history of the message from the message originator to the message's final destination. SHLI does not consider the overhead at all as it uses the message TTL as decision making condition for the message to be dropped. On the other hand, the MOFO policy creates less overhead than SHLI as the MOFO decision is considering the Hop Count and Replication Count only indirectly. While SHLI is to be considered as not practicable for this reason. MOFO is still on par with proposed DWDP. A decision on the quality of the remaining two drop policies is given through the observation of the corresponding delay of the delivered messages.

The delay for the different Scenarios 1 to 3 with varying values for TTL and traffic rate is shown in

Figures 3(a), 3(b) and 3(c). The Figures show that the delay increases as message TTL increases in all three scenarios. While for Scenario 1, DWDP yields the best average delay for the delivered messages, with SHLI being close by, for Scenario 2 DWDP produces the best delay values. For Scenario 3, both drop policies SHLI and DWDP are on par with 7.8429% delay difference in the worst case.

MOFO has in average much higher delays, ranging from 4% to 40% more than DWDP, for the delivered messages. the Figure 3 shows that the delay of proposed drop policy DWDP close to the values of both policies SHLI and MOFO, this is because FIFO considers on parameter of delay which is buffer time at the intermediate nodes. where SHLI consider the delay from the message originator to the current message carrier. The proposed drop policy DWDP considers both the TTL as well as the buffer time TBUF and thus gains an additional advantage.

The MOFO policy does not consider the delay at all, as it uses the message overhead, Hop Count and Replication Count, as decision making condition for the message to drop.

VI. CONCLUSION AND FUTURE WORK

In this work, DWDP is proposed to improve the performance of the PROPHET routing protocol version 2. With the ONE simulator, an evaluation of the drop policy with 3 other drop policies i.e. MOFO, FIFO, and SHLI is performed. The performance of these policies is analyzed with different TTLs for the messages and with different traffic scenarios. The evaluation shows that DWDP is quite better with respect to MOFO and SHLI in terms of message delivery ratio. The simulation experiments indicate that the proposed drop policy DWDP is better that the existing drop policies approaches as it is able minimize the overhead ratio and the delay while being quit better with the delivery ratio. SHLI produces extensive traffic overhead, while the MOFO drop policy results in a systematically increased delay. Thus, we improved the performance of PROPHET protocol version 2 by employing the proposed Dynamic Weight Drop Policy DWDP.

As future work, the application of the proposed DWDP on different routing protocols such as the Spray & Wait routing protocol is considered. In addition, the option to consider more information for scheduling and dropping messages than a single message can carry.

REFERENCES

[1] D. Pan, Z. Ruan, N. Zhou, X. Liu, and Z. Song. A comprehensive-integrated buffer management strategy for opportunistic networks. EURASIP J. Wireless Comm. and Networking, 2013:103, 2013.

[2] F. Legendre and A. Helmy, editors. Proceedings of the 6th ACM workshop on Challenged networks, CHANTS@MOBICOM 2011, Las Vegas, NV, USA, September 19-23, 2011. ACM, 2011.

[3] A. Krifa, C. Barakat, and T. Spyropoulos. Message drop and scheduling in dtns: Theory and practice. IEEE Trans. Mob. Comput., 11(9):1470{1483, 2012.

[4] K. Shin and S. Kim. Enhanced buffer management policy that utilises message properties for delay-tolerant networks. IET Communications, 5(6):753{759, 2011.

[5] S. Rashid, Q. Ayub, and A. H. Abdullah. Reactive weight based buffer management policy for DTN routing protocols. Wireless Personal Communications, 80(3):993{1010, 2015.

[6] A. Lindgren, A. Doria, and S. G. E. Davies and. Rfc 6693: Probabilistic routing protocol for intermittently connected networks. IETF, 2012.

[7] S. Grasic, E. Davies, A. Lindgren, and A. Doria. The evolution of a DTN routing protocol - prophetv2. In Proceedings of the 6th ACM workshop on Challenged networks, CHANTS@MOBICOM 2011, Las Vegas, NV, USA, September 19-23, 2011, pages 27{30, 2011.

[8] A. Keranen, T. Karkkainen, and J. Ott. Simulating mobility and dtns with the ONE (invited paper). JCM, 5(2):92{105, 2010.

[9] A. Keranen, J. Ott, and T. Karkkainen. The ONE simulator for DTN protocol evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, SimuTools 2009,March 2-6, 2009, page 55, 2009.