Drop Policies for DTN Routing Protocols with Delivery Probability Estimation

Miguel Pinheiro Rodrigues
Instituto de Engenharia de Sistemas e
Computadores (INESC-ID),
Universidade de Lisboa, Portugal
miguel.pinr@gmail.com

Naércio Magaia
Instituto de Engenharia de Sistemas e
Computadores (INESC-ID),
Universidade de Lisboa, Portugal
naercio.magaia@tecnico.ulisboa.pt

Paulo Rogério Pereira
IInstituto de Engenharia de Sistemas e
Computadores (INESC-ID),
Universidade de Lisboa, Portuga
prbp@inesc.pt

***Abstract* — Delay Tolerant Networks (DTN) are characterized by a lack of end-to-end connectivity. As such, messages (called bundles) can be stored in buffers for a long time. Network congestion can result in poor delivery rates, as bundles are dropped before having a chance of reaching their destination. Some routing protocols, such as MaxProp and Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET), maintain estimations of delivery probabilities for each destination. In this paper, a new drop policy called Largest Bundle's Hosts Deliverability (LBHD) is proposed that considers all the hosts that received a replica of the same bundle, and their respective delivery probability as estimated by a routing protocol. LBHD uses this additional information to better manage congestion. Simulation results show that LBHD consistently achieves the best delivery probability when paired with PRoPHET and compared with other drop policies proposed in the literature. Also, when paired with MaxProp, LBHD shows the most efficient performance among all the other state of the art policies considering performance metrics such as average delay, overhead ratio and bundle delivery rate. In addition, another drop policy called One Hop Delivery Estimation Drop (OHDED) is proposed. OHDED takes advantage of the encounter predictions of every node in the network stored in every node when using MaxProp. By accurately predicting the bundles that have the highest probability of being delivered directly or in two hops, the results show the best performance in delivery rate and overhead ratio in high congestion scenarios.**

Keywords: Delay Tolerant Networks, DTN, Drop Policies, LBHD, OHDED, Routing Protocols.

I. INTRODUCTION

As several articles state ([1]–[3]), Delay Tolerant Networks (DTN) were historically originated as a proposal for Inter Planetary Networks (IPN) to provide communications between satellites, planets and base stations. This was a scenario in which high delays and frequent disconnections were very common, so the network should be able to take all of this into account, and use proper mechanisms to optimally manage this type of links.

With the broader utilization of the DTN architecture in mobile networks with intermittent connectivity, the need of passing information through store, carry and forward methodologies arises. All data pertaining to a source- destination interaction is aggregated in a message called bundle, to be routed taking advantage of the communication opportunities. Since the major routing protocols and congestion mechanisms available were developed for continuous connectivity, the necessity of creating compatible protocols for DTNs became a reality.

The development of vehicular networks, opportunistic networks, networks for emergency response, military operations, tracking and monitoring applications raised the question of how to optimize the already proposed strategies for DTNs in networks with unpredictable frequent disconnections and high mobility. The development of mechanisms that would properly manage the buffer space and schedule the bundles to be forwarded and dropped became a challenge.

The simulation and proper evaluation of the algorithms performance is still a challenge in the research community. Since the natural human mobility is very hard to simulate, more realistic simulation scenarios are being developed by the scientific community to better evaluate the performance and overall quality of these algorithms.

In this work, an analysis of the state of the art of the available drop policies used with the Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [4] and MaxProp [5] routing protocols is made, comparing the results of such policies with two proposed policies.

The protocols were chosen for their delivery probability prediction mechanisms, as it was thought that by using such information, new drop policies with good performance could be proposed, and a good performance in terms of delivery rate, overhead ratio and latency could be expected.

The remainder of this paper is structured as follows. Section II provides an analysis of the state of the art and an explanation of all the existing drop policies used. Later in section III, one of the proposed drop policies is explained and simulated paired with the PRoPHET routing protocol. In section IV, this policy is adapted to be used with MaxProp routing protocol and another drop policy is proposed and explained. Additionally, sections III and IV provide an analysis of the simulation results. In the last section, some conclusions are drawn and future research topics are proposed.

II. STATE OF THE ART

In the present section, the routing protocols and drop policies used in the latter sections are presented and explained.

A. Routing Protocols

In the following topics, the PRoPHET and MaxProp routing protocols, which were used to test the proposed drop policies performance, will be described.

1) PRoPHET

By not eliminating message replications that would not improve the delivery ratio in the network, the Epidemic protocol is very resource hungry, since that in a real scenario the nodes encounters are not purely random. By taking advantage of this predictability, it is possible to improve the delivery ratio.

As explained in [6], one of the main routing protocols of this type is the Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) protocol [4]. In this protocol, each of the nodes contains a vector with the delivery predictability of the other nodes of the network. The delivery predictability is calculated using the information provided from past encounters [7].

In the PRoPHET protocol, a node carrying a bundle only transfers it to nodes with higher delivery probability than itself, managing to improve the redundancy and the delivery ratio, since as explained above, this is done using information of past encounters [6].

Sukhbir in [8] explains that since the PRoPHET mainly uses past information, in the initial state where there is an uniform probability distribution, the performance is at its minimum. PRoPHET++ was created with the intention of improving the main PRoPHET disadvantage.

Patel in [7] introduces the PRoPHET++ protocol. As a way to solve the problem listed above, the PRoPHET++ protocol, in the early stages behaves as the Epidemic protocol, as a way to initially disseminate the messages, preventing dropping of the initial messages.

2) *MaxProp*

One of the proposed features of the MaxProp protocol is that once a bundle is delivered to its destination, an acknowledgment is used to delete the existing messages copies in the network [6]. By using this particular buffer management approach, MaxProp manages to lower the packet drop ratio. Such feature may be implemented in other routing protocols.

In the MaxProp routing protocol [5], every node in the network has a table of the probabilities of each node delivering a bundle to another. This table is updated at every encounter to maintain the most current values. This delivery probability table is used to estimate the cost of delivery of a certain bundle by any node in the network. This cost is estimated through Dijkstra's algorithm [9], in order to optimize the probability calculation since the algorithm has to approximate the cost of delivery, with small error and without iterating through all the delivery probability table.

This cost is computed calculating the delivery probability for a certain location. It is also able to send messages in specific order that considers message hop counts and delivery probabilities based on previous encounters.

This protocol does not require the knowledge of the network connectivity (present nor future), the node's location (present nor future) nor the stationary relay nodes [6].

The main disadvantages of the protocol are the decreased time for message exchange (by exchanging the tables, the contact time used to transfer the requested messages is shorter), and not being suited for sparse networks, i.e. it does not provide a connected graph.

B. *Drop Policies*

Drop policies act when the buffer of a node is full, due to congestion, to free enough space to store a new bundle that the node generated or that was received from a neighbour. The focus of this work was in developing new drop policies for routing protocols with delivery prediction. In this section, a brief explanation of the existing drop policies is provided. First, some simple traditional drop policies that require no information are described. Then, some more specialized policies that use the delivery probability value estimated by PRoPHET to choose the bundle to be dropped are described.

1) *Random*

For an incoming bundle that has to be stored in a full node's buffer, in the Random policy, as the name would suggest, the dropped bundle is chosen randomly among every stored bundle [3].

2) *Drop Head*

In the Drop Head policy, as mentioned by Soares in [10], when the node's buffer is full, the first dropped bundle from the Node is the first incoming one, and so the priority in the bundle queue is such that the first bundle to arrive is the first to be dropped. Of course, the information of the time the bundle was received has to be kept.

3) *Drop Oldest*

In Drop Oldest (DO) [11], the analysis is based on the Remaining Time To Live (RTTL) of the bundle. The bundle with the shortest RTTL is the dropped one.

4) *Drop Youngest*

The Drop Youngest (DY) algorithm functions as an inverse of the DO algorithm. In this case the chosen bundle to be dropped is the one with the largest RTTL.

5) *Drop Most Hops*

The number of hops corresponds to the number of nodes a bundle was passed through until it reached the current node. In the Drop Most Hops policy, the chosen bundle to be dropped is the one with the most hops [12].

6) *MOFO*

Evict Most Forwarded First (MOFO) [4] used by Rani in [13] chooses the bundle which has been forwarded the largest number of times to be dropped. Every bundle has a corresponding forward counter variable – stored separately in each node – named Number of Forwards (NF), which starts at 0 and is incremented every time the bundle is forwarded.

7) *MOPR*

Evict Most Favourably Forwarded First (MOPR), is a weighted version of MOFO. Instead of incrementing the Forwarding Predictability (FP) by one each time the bundle is forwarded, this algorithm increments FP with the delivery predictability of the receiving node. Of course, as in MOFO, the FP is initialized at 0. As illustrated by Lindgren in [14], MOPR policy follows the following equation:

$$FP = FPo1d + P. \qquad (1)$$

Being FPo1d the value of the bundle before it was forwarded, FP the updated value and P the deliverability predictability the receiving node has for the message.

8) *LEPR*

Evict least probable first (LEPR), as described in [14] and [15], is a policy in which the least probable bundle to be delivered by a node is dropped. This policy can only function when paired with routing protocols in which the delivery probability is estimated e.g. PRoPHET and MaxProp [5].

For each bundle, considering the bundles' destination, a delivery probability is estimated by the routing protocol. Afterwards, the bundle for which the delivery probability is the lowest is chosen to be dropped.

III. LBHD PAIRED WITH PROPHET

After the analysis of the MOPR and LEPR drop policies, some points of improvement were identified. In this section, the proposed drop policy is described and simulated using PRoPHET. As a start, the major points of improvement of the best performing policies in this category will be listed. Secondly, the proposed policy will be described in detail, listing the main features of the policy and how the proposed algorithm aims to surpass the points of improvement described in part A.

A. Major Points of Improvement

1) Only the nodes which receive a bundle should count

In the LEPR policy, the estimation of the delivery probability of the bundle's hosting node, calculated by PRoPHET, does not consider the number of times a bundle has been forwarded nor the probability of delivery of the new hosts.

Since the information may not be very recent, and the network changes with time, a node may be prioritizing bundles which have been replicated to other nodes with high delivery probability instead of dropping them. Seeing that, tendentiously, a much-replicated bundle has a large possibility to be delivered, and one which was not as much replicated has a lower delivery probability, it would make sense to account only the delivery probability of the nodes which have received the bundle so far.

2) No Transitivity in MOPR's FP value

In the MOPR drop policy there is no transitivity of the FP value. For instance, when a node receives a new bundle in its buffer, the FP for that bundle is 0, being increased every time this node relays the bundle for other nodes, independently of how many copies the previous nodes have created before.

This effect may be minimized by transferring the FP value with the bundle when it is forwarded to another node.

3) No update between hosts of the same bundle

In the MOPR drop policy, every node has a FP value for every bundle stored in the buffer. The FP is not updated with information from other nodes that carry the same bundle.

To have the most updated value of a bundle's FP, which may have been more replicated by a node than another, when an encounter occurs, the nodes should exchange the FP values for the bundles each one stores, and in the case that both the nodes carry the same bundle, the bundle's FP value of both nodes should be updated to the largest value between the two.

B. LBHD – Algorithm Description

Largest Bundle's Hosts Deliverability (LBHD) is the proposed drop policy, which is intended to have better performance than all the drop policies in section II.B. It chooses the bundle to be dropped through the delivery probability of all the hosts of a certain bundle replica, updating the metric at every encounter opportunity.

Through some key ideas of LEPR and MOPR, described in [14], such as the use of the estimated delivery probabilities and the sum of the delivery probabilities of all the nodes, respectively, a new metric was developed.

The new metric is called Hosts Deliverability (HD) and is stored in a new field in every bundle. The HD value is calculated using the equation:

$$HD(t) = HD(t - 1) + Pi \rightarrow dest, \qquad (2)$$

In which the HD(t) corresponds to the HD value in the present moment, HD(t – 1) to the value previously stored in the buffer, and Pi→dest to the probability of delivery – estimated by the routing protocol – of node i to the destination of the bundle. Node i corresponds to a node that carries the bundle, i.e. every time a bundle is replicated, its HD value is incremented with the delivery probability of the new carrier.

Since it is not possible to infer the delivery probability of a node i, which is going to carry the bundle to its destination, this information must be sent by node i – the new carrier – to the sender node, as soon as node i receives the bundle.

It is also important to refer that through (2), as soon as a node receives a new bundle, it updates the HD metric of the bundle with its own delivery probability to the bundle's destination. In this way, when a bundle is replicated, the information of the delivery probability of the sender node is already included in the HD metric.

When there is congestion and there is no buffer space left, the first bundle to be dropped is the one with the highest HD value. Since a bundle with a large HD value has been already replicated to other nodes with good delivery probabilities for that bundle, the impact of dropping it is reduced. The new bundles in the network and the ones created by the carrying node, which tendentiously have lower HD values, should be prioritized so that these types of bundles have a fair chance of being delivered.

1) Only the nodes which receive a bundle should count

Using the HD metric, the first point of improvement of section III.A is addressed, since that only the probabilities of the nodes which actually receive the bundle are considered in the HD metric.

2) No Transitivity in MOPR's FP value

To address the second point listed in section III.A, it was decided that the HD value should be included in the bundle itself instead of storing it in the node, which would cause the next host to lose information about the previous replication events of the bundle in other hosts.

3) No update between hosts of the same bundle

Addressing the third and final point of improvement, the nodes must exchange their bundles' HD values to get the most updated HD value for each bundle, and in case there is a match

– both nodes have the same bundle buffered – the HD value of both bundles is updated to the largest value between the two, as illustrated in Fig. 1, where only bundle B3 is held by both nodes and the corresponding metric exchanged.

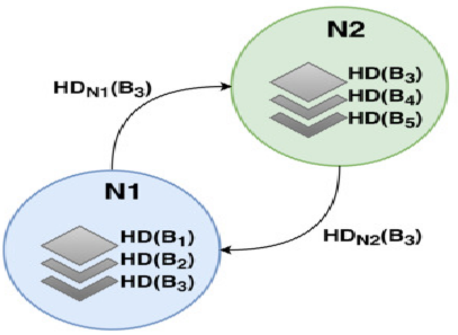

Figure 1. HD values exchange between nodes.

C. Simulation

In this section, the main performance metrics and the simulated scenario are described. The simulations were run using the Opportunistic Network Environment (ONE) simulator [16], [17] which is capable of simulating a delay- tolerant network with a specified mobility scenario and measure its performance through the generated reports and data that may be post processed.

1) Performance Metrics

The performance metrics used to evaluate the performance of different drop policies in the ONE simulator were:

- Bundle delivery rate – ratio of the successfully delivered bundles over the total number of bundles generated.
- Average delay – Average time the bundles take to be delivered.
- Overhead ratio – Ratio of the difference between the relayed and delivered messages (RM and DM respectively), over the number of delivered messages, as illustrated by the equation:

$$Overhead = \frac{R_M - D_M}{D_M}. \qquad (3)$$

2) Simulation scenario

The main goal of the scenario was that the DTN would be between some load and congestion. This could be achieved by using a small buffer space in the nodes, large messages and short message creation interval. The chosen scenario of the simulation is illustrated in Table I and II, for the general parameters in Table I and the more specific node parameters in Table II.

Influenced by Soares in [10], the simulations were run varying the Time To Live (TTL) of the bundle, which was varied from 30 to 300 minutes in increments of 30 minutes. This allows varying the network load, as a larger TTL results in messages circulating for longer time and a higher network load.

TABLE I. PROPHET SCENARIO – GENERAL CONFIGURATIONS

General Configurations		
Routing Protocol	PRoPHET	
Movement Model	Shortest Path Map Based Movement	
Number of Seeds	12	
Total Time	24 hours	
Warmup Time	10 000 sec.	
Message Size interval	250 kB to 2 MB	
Message Creation Interval	15 to 30 sec.	
Long range radio	Data Speed	4.5 Mbps
	Range	30 m
Highspeed radio	Data Speed	40 Mbps
	Range	10 m

Several simulations with different seeds were performed and the results averaged for statistical confidence. Other values such as the number of nodes, speeds, etc. were dimensioned to create a realistic scenario in terms of movement. There are a total of 156 nodes classified in 4 different groups. The chosen movement model was the Shortest Path Map Based Movement (SPMBM).

As explained in [18] by Keranen, SPMBM is based on the simple random map-based model, in which the nodes move to randomly determined positions on the map following the roads defined by the map data. Although, in SPMBM model the nodes do not wander randomly around the map, instead the Dijkstra's shortest path algorithm is used to calculate the shortest paths from the current location to a randomly selected destination.

TABLE II. PROPHET SCENARIO – NODE CONFIGURATIONS

Node Configurations		
Car 1	**Number of hosts**	**100**
	Wait Time	5 to 15 min.
	Node Speed	30 to 50 km/h
	Buffer Size	8 MB
	Radio Interfaces	1 Long range radio
Car 2	**Number of hosts**	**50**
	Wait Time	5 to 15 min.
	Node Speed	10 to 50 km/h
	Buffer Size	5 MB
	Radio Interfaces	1 Long range radio
Tram 1	**Number of hosts**	**4**
	Wait Time	10 to 30 sec.
	Node Speed	25 to 35 km/h
	Buffer Size	50 MB
	Radio Interfaces	1 Long range radio
Tram 2	**Number of hosts**	**2**
	Wait Time	10 to 30 sec.
	Node Speed	25 to 35 km/h
	Buffer Size	50 MB
	Radio Interfaces	1 Long range radio 1 Highspeed radio

After the simulation, the results were processed and organized in the tables and figures below. In the next section, the results of the proposed performance metrics described in section III.C.1 will be displayed and analysed.

D. Results

1) Bundle Delivery Rate

The results in terms of the Bundle Delivery Rate are presented in the Table III and also displayed in Fig. 2. Fig. 2 also shows 95% confidence bars for every point. In most cases, these intervals are very small, meaning there is a high confidence in the values presented.

TABLE III. PROPHET SIMULATION: BUNDLE DELIVERY RATE

TTL [min.]	Bundle Delivery Rate						
	Random	Drop Head	DO	DY	MOPR	LEPR	LBHD
30	40.84%	41.44%	45.67%	36.93%	43.08%	42.25%	44.66%
60	46.18%	47.70%	59.64%	36.90%	54.78%	54.69%	61.54%
90	41.25%	43.38%	60.86%	29.82%	57.11%	57.48%	65.64%
120	36.48%	38.92%	60.85%	24.66%	56.90%	58.84%	65.97%
150	32.94%	35.96%	60.83%	21.11%	56.78%	59.47%	65.77%
180	30.67%	33.83%	60.80%	19.13%	55.64%	58.62%	64.25%
210	28.79%	32.42%	60.80%	17.27%	53.67%	56.64%	63.43%
240	27.60%	31.48%	60.80%	15.79%	52.10%	53.73%	62.12%
270	26.51%	30.78%	60.80%	14.35%	49.63%	50.34%	60.93%
300	25.81%	29.76%	60.80%	13.46%	47.18%	46.40%	60.34%

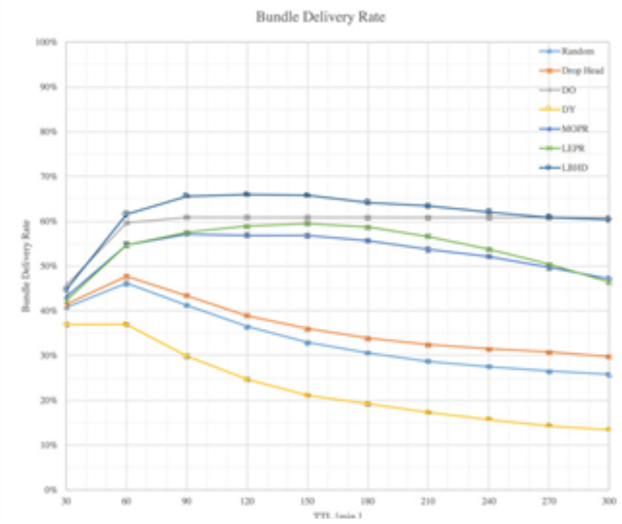

Figure 2. Bundle Delivery Rate varying the TTL of the message using the PRoPHET routing protocol.

For low TTL values, most bundles are dropped before they have a chance of reaching the destination. When the TTL increases, the chances of having enough time to reach the destination increase, improving the delivery rate. However, as TTL increases, bundles also occupy buffer space for longer time, resulting in network congestion and smaller improvements on the delivery rate, or even a drop in the delivery rate.

The DO policy has a good delivery rate performance. By dropping the messages which have the lowest RTTL in the buffer, DO is dropping bundles that had the smallest time to reach their destinations and probability.

It is important to refer that the Drop Head, DY and Random policies have poor results for TTLs larger than 60 minutes. In the case of Random for dropping bundles without any criteria; DY for prioritizing older bundles in the network instead of the ones which have recently been created; and Drop Head for dropping only based on the receive time of the bundle, which accomplishes superior results to Random and DY, but is not ideal since other variables may have more impact for a congested network, as is the case of the RTTL, used by the DO policy.

The LBHD policy is the one that achieves the highest delivery rates, surpassing Drop Head, DO, LEPR and MOPR, by an average of 37%, 3.9%, 14% and 17%, respectively. This is due to the added features and functionalities that none of the other studied policies have.

By trading the information of the probability of delivery of the relayed replicas of the bundles, and updating its value whenever two hosts of the same bundle have a contact opportunity, this policy can better manage a full buffer in congestion scenarios.

As expected, when congestion increases, LBHD delivery rate decreases since it becomes harder to manage the stored bundles.

2) *Average Delay*

The Average Delay is shown in Table IV and also in Fig. 3, with 95% confidence bars for every point. The proposed LBHD policy performs better than LEPR and MOPR, by an average of 26% and 34%, respectively. The default Drop Head policy performs an average of 21% better than LBHD.

The best policy, in terms of latency, is DO, achieving an average delay reduction of 45% in comparison with LBHD. This is due to the fact that the oldest bundles of each buffer are discarded and so, the probability of a bundle being delivered in the end of its life is small, which lowers the average delay in the network.

On the other hand, LBHD has a higher delivery rate, since it manages to hold older bundles which may be delivered in the end of their life increasing the average delay. This is the price to pay for a better delivery rate.

TABLE IV. PROPHET SIMULATION: AVERAGE DELAY

TTL [min.]	Average Delay [s]						
	Random	Drop Head	DO	DY	MOPR	LEPR	LBHD
30	1083.53	1074.75	1044.50	1114.85	1102.50	1080.36	1075.79
60	1862.78	1830.54	1654.98	1949.20	2039.83	1946.26	1916.40
90	2349.99	2198.55	1773.76	2410.85	2867.00	2650.99	2528.67
120	2601.59	2346.72	1783.84	2699.73	3647.22	3297.35	2981.31
150	2660.37	2377.27	1781.23	2921.82	4424.47	3926.74	3337.76
180	2709.89	2366.55	1782.03	2921.04	5170.77	4547.00	3578.15
210	2687.94	2379.16	1782.03	2923.17	5959.85	5180.69	3783.23
240	2695.13	2339.63	1782.03	2833.82	6664.20	5803.71	3905.71
270	2678.41	2289.54	1782.03	2737.90	7425.15	6439.50	3993.96
300	2611.50	2290.11	1782.03	2642.21	8213.77	7061.09	4096.80

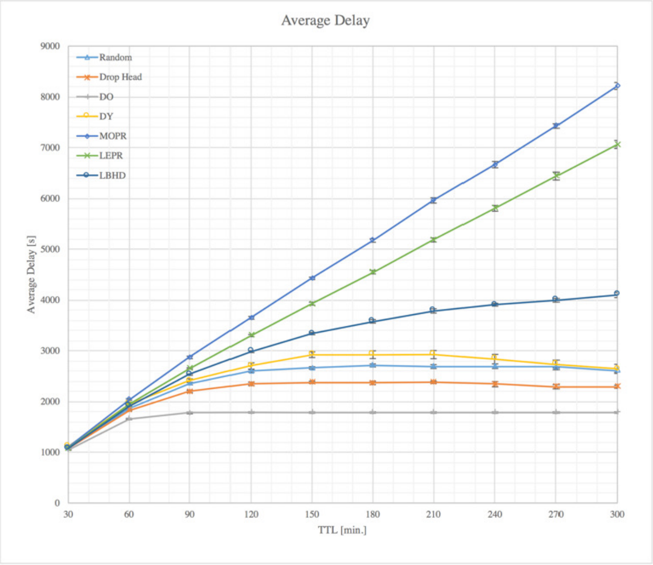

Figure 3. Average Delay in seconds, varying the TTL of the message using the PRoPHET routing protocol.

3) *Overhead Ratio*

The results for the overhead ratio are presented in Table V and in Fig. 4, with 95% confidence bars for every point.

LBHD behaves well for large TTL values, although the results of MOPR throughout all the simulated TTLs are better

TABLE V. PROPHET SIMULATION: OVERHEAD RATIO

TTL [min.]	Overhead Ratio						
	Random	Drop Head	DO	DY	MOPR	LEPR	LBHD
30	63.20	60.38	97.75	72.28	55.39	87.03	81.02
60	86.55	79.43	162.79	122.17	64.36	148.56	123.43
90	112.81	99.33	171.04	182.03	65.76	174.44	136.92
120	136.28	118.27	171.22	236.30	60.47	179.48	144.00
150	157.32	132.24	171.32	281.43	53.67	176.53	146.77
180	174.69	145.21	171.34	311.24	48.31	173.21	151.33
210	190.91	155.83	171.34	343.71	42.38	169.67	153.49
240	202.48	161.29	171.34	365.66	37.76	165.53	156.60
270	211.02	165.28	171.34	379.76	33.86	161.58	159.18
300	217.79	171.26	171.34	390.26	30.17	159.57	160.89

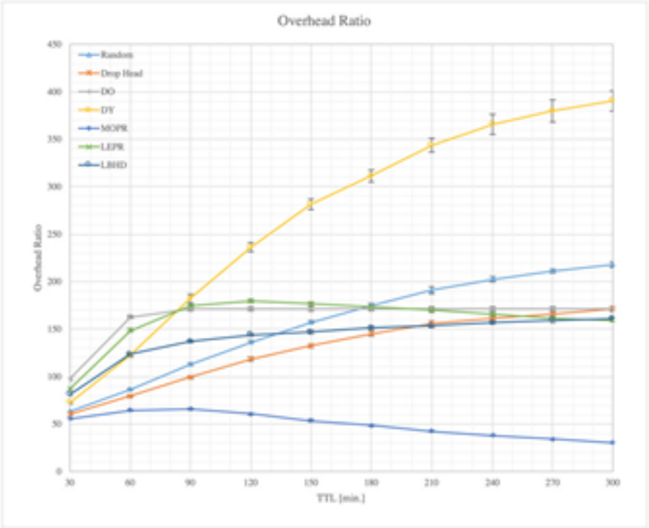

Figure 4. Overhead ratio, varying the TTL of the message using the PRoPHET routing protocol.

For smaller TTL values, since the congestion is not significant, LBHD allows more replicas of the bundles to be spread throughout the network.

For larger TTL values, since the network is highly congested, LBHD constrains the created number of replicas of the bundles, since it tendentiously prioritizes bundles with lower number of replicas. Thus, the overhead does not increase significantly for larger TTL values.

IV. LBHD AND OHDED PAIRED WITH MAXPROP

It was decided to simulate LBHD policy using the MaxProp routing protocol, to test its applicability for other routing protocols with delivery probability prediction.

An alternative drop policy called One Hop Delivery Estimation Drop (OHDED) policy was also developed to be used specifically with the MaxProp routing protocol, since it exploits its table of encounter probabilities. The OHDED policy was developed as an improvement to the default Drop Max Hops drop policy used in MaxProp.

A. LBHD algorithm adaptation to MaxProp

For the MaxProp algorithm, an adaptation of the used metric was made. Since that LBHD's HD metric depends on the probability of delivery of a node calculated by the PRoPHET routing protocol, an adapted approach for the MaxProp routing protocol was used.

$$HD(t) = HD(t-1) + Cost_{i \rightarrow dest} \qquad (4)$$

In the MaxProp protocol, the cost of a bundle being delivered by any node in the network is computed, as previously mentioned. Since that every node can compute the cost of a node delivering a bundle to its destination, the HD metric in (2) was updated to in which the HD(t) corresponds to the HD value in the present moment, HD(t − 1) to the value previously stored in the buffer, and Costi→dest to the cost of delivery – estimated by the MaxProp routing protocol with the delivery probabilities for the best path – of node i to the destination of the bundle. Node i corresponds to a node that carries the bundle, i.e. every time a bundle is replicated, its HD value is incremented with the cost of delivery of the new carrier.

In this way, by dropping the bundle with the largest HD value, the dropped bundle is one that has several replicas already in the network, which has low impact in the delivery probability.

The rest of the LBHD drop policy features were maintained.

B. OHDED – Algorithm Description

Since that in the PRoPHET protocol, each node possesses a table with the encounter probability between every node in the network, it was thought that this information could be used to predict which bundles should be dropped, and which ones should be prioritized.

Once a buffer is full, the proposed OHDED algorithm computes the Bundle's Delivery Probability (BDP) metric which takes into consideration the scenarios represented in Fig. 5, in which it is considered the probability of direct delivery of the bundle to its destination node and also the probability of its delivery to one certain bundle in the network which finally delivers it to its destination node.

$$BDP = P_{i \rightarrow d} + \sum_{\substack{j=1 \\ j \neq i,d}}^{N} P_{i \rightarrow j} P_{j \rightarrow d} \, . \qquad (5)$$

As represented in the equation in which BDP represents the BDP metric, the present node with its buffer full, d the bundle's destination node, j one node in the network different from i and d, and N the total number of nodes in the network. The variables: Pi→d, Pi→j and Pj→d , correspond, respectively, to the encounter probability, estimated by MaxProp: of the present node i with the destination node d; of the present node i with another node in the network j; and of the node j with the bundle destination node d.

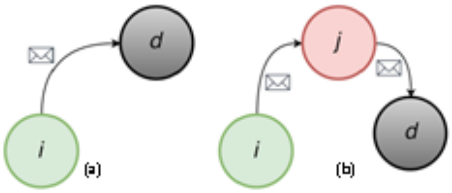

Figure 5. Situations considered for bundle delivery in the OHDED algorithm: (a) direct delivery, 1 hop; (b) one intermediate node, 2 hops.

After BDP is computed, the first chosen bundle to be dropped is the one whose BDP is the lowest. In this way, the node will only maintain in the buffer bundles for which it has a substantial probability to deliver directly; or bundles to replicate to another node, which the present node has a large encounter probability with, and the other node has also a high encounter probability with the bundle's destination.

C. Simulation scenario

The chosen simulation scenario is similar to the one described in section III. The general chosen parameters and settings of the simulation scenario are described in Table VI, and the more specific node configurations set are described in Table VII.

TABLE VI. MAXPROP SCENARIO – GENERAL CONFIGURATIONS

General Configurations		
Routing Protocol	MaxProp	
Movement Model	Shortest Path Map Based Movement	
Number of Seeds	12	
Total Time	24 hours	
Warmup Time	10 000 sec.	
Message Size interval	250 kB to 2 MB	
Message Creation Interval	15 to 30 sec.	
Long range radio	Data Speed	4.5 Mbps
	Range	30 m
Highspeed radio	Data Speed	40 Mbps
	Range	10 m

As previously mentioned, congestion was forced through the buffer size, message size interval and message creation interval parameters, to create a scenario where the drop policies have an intensive use and consequently affect the performance metrics' results.

The chosen drop policies to be compared with the LBHD drop policy, in this scenario, were Drop Head, Drop Oldest, Drop Most Hops and OHDED. Since the Drop Most Hops policy is the default drop policy in the MaxProp routing protocol, it was logical to compare the proposed drop policy with the standard one to understand if LBHD can improve the performance of MaxProp as it can with PRoPHET.

D. Results

Having the settings and simulation parameters set, the results of the performance metrics proposed in section III.C.1) will be displayed and analysed.

TABLE VII. MAXPROP SCENARIO – GENERAL CONFIGURATIONS

Node Configurations		
Car 1	**Number of hosts**	**100**
	Wait Time	5 to 15 min.
	Node Speed	30 to 50 km/h
	Buffer Size	8 MB
	Radio Interfaces	1 Long range radio
Car 2	**Number of hosts**	**50**
	Wait Time	5 to 15 min.
	Node Speed	10 to 50 km/h
	Buffer Size	5 MB
	Radio Interfaces	1 Long range radio
Tram 1	**Number of hosts**	**4**
	Wait Time	10 to 30 sec.
	Node Speed	25 to 35 km/h
	Buffer Size	50 MB
	Radio Interfaces	1 Long range radio
Tram 2	**Number of hosts**	**2**
	Wait Time	10 to 30 sec.
	Node Speed	25 to 35 km/h
	Buffer Size	50 MB
	Radio Interfaces	1 Long range radio 1 Highspeed radio

1) *Bundle Delivery Rate*

The bundle delivery rate performance metric results are shown in Table VIII and illustrated in Fig. 6. Fig. 6 also shows 95% confidence bars for every point. In most cases, these intervals are very small, meaning there is a high confidence in the values presented.

The LBHD policy had the best performance in the lower TTL values, being surpassed by OHDED and Drop Most Hops policy for larger TTL values. The OHDED policy performed better than Drop Head, DO, Drop Most Hops and LBHD by an average of 17%, 11%, 1% and 2%, respectively. Although the difference between the Drop Most Hops, OHDED and LBHD in this performance metric is not considerable for TTLs larger than 90 minutes, the best performing policy, for larger TTLs is OHDED.

The LBHD policy performs better than the Drop Head policy (default drop policy in most of the routing protocols), by an average of 15%. Also, LBHD has a gain of 9%, 0% and – 2% when compared with DO, Drop Most Hops and OHDED, respectively. The results show that LBHD has a similar performance to the standard drop policy in MaxProp routing protocol – Drop Most Hops, while having superior results when compared with the other state of the art policies. Also, the Drop Most Hops policy good performance is related to the fact that by eliminating bundles – which have been replicated many times already – beyond tendentiously having a smaller RTTL, have already had multiple opportunities of being replicated, thus will not have a large impact in the bundle delivery rate.

The remaining two drop policies – Drop Head (FIFO) and DO – had the worst results since both policies do not use any of the delivery probabilities estimated by MaxProp. DO had a better performance than Drop Head since that, as already stated previously, the RTTL of the bundle is a very important parameter, which is not considered in the Drop Head drop policy.

2) *Average Delay*

The Average Delay is shown in Table IX and also in Fig. 7, with 95% confidence bars for every point.

From these results, it was inferred that the best performing policy was DO. The good performance in terms of average delay of the DO policy is related to its algorithm, in which the smaller RTTL bundles are the first dropped ones, which decreases the probability of delivering a bundle which has been in the network for a long time, hence the average delay decreases, but the delivery rate also decreases.

TABLE VIII. MAXPROP SIMULATION: BUNDLE DELIVERYRATE

TTL [min.]	Bundle delivery rate				
	Drop Head	DO	Drop Most Hops	OHDED	LBHD
30	55.24%	59.36%	60.12%	59.55%	62.21%
60	65.81%	69.92%	77.53%	77.80%	79.47%
90	68.83%	74.65%	84.51%	85.83%	85.07%
120	70.16%	76.78%	88.01%	89.83%	87.73%
150	70.91%	77.97%	89.99%	92.32%	88.27%
180	71.50%	78.62%	91.05%	93.29%	89.04%
210	71.70%	78.72%	92.05%	94.18%	89.52%

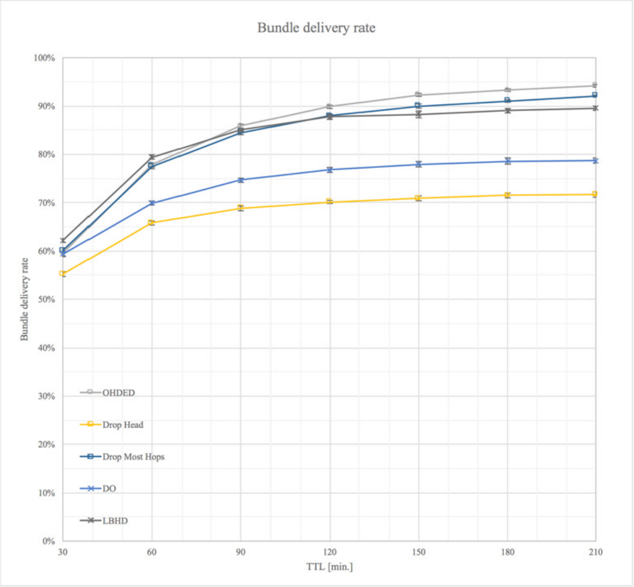

Figure 6. Bundle delivery rate, varying the TTL of the message, between LBHD, OHDED and other drop policies paired with MaxProp routing protocol.

The Drop Head and LBHD drop policies also had a good performance in terms of average delay. By dropping the firstly stored bundle, tendentiously, the dropped bundles will also have a small RTTL.

In terms of average delay, the LBHD drop policy performed better than Drop Head, DO, Drop Most Hops and OHDED policies by an average of –7%, –22%, 25% and 27%, respectively. Since that, in LBHD every time a bundle is replicated, the HD metric is incremented, being dropped the bundle which has the largest value, implicitly the bundles which have been in the network longest, having lower RTTLs, will be dropped.

The Drop Most Hops and OHDED drop policies show the worst performance in the simulated scenario, this is related to the fact that neither of the policies considers nor measures the RTTL of the bundle, which increases the probability of delivering a bundle in the end of its lifecycle.

3) *Overhead ratio*

The results for the overhead ratio are presented in Table X and in Fig. 8, with 95% confidence bars for every point.

For the overhead ratio, the OHDED, Drop Most Hops and LBHD policies had the best performance in the simulation.

As regards the average gains of LBHD policy when compared to Drop Head, DO, Drop Most Hops and OHDED these were of 20%, 11%, 0% and –2%, respectively. Hence, we may conclude that the performance of LBHD was equivalent to the Drop Most Hops policy. Since the network is highly congested, LBHD constrains the created number of replicas of the bundles, since it tendentiously prioritizes bundles with lower number of replicas which explains the good performance in terms of overhead.

TABLE IX. MAXPROP SIMULATION: AVERAGE DELAY

TTL [min.]	Average Delay [s]				
	Drop Head	DO	Drop Most Hops	OHDED	LBHD
30	972.96	869.49	1027.38	1029.52	982.37
60	1416.94	1148.73	1724.62	1732.48	1472.95
90	1606.95	1365.63	2184.65	2217.35	1709.37
120	1727.86	1521.46	2505.05	2597.04	1854.63
150	1812.05	1620.11	2741.93	2862.58	1968.77
180	1853.41	1674.86	2914.42	3068.51	2047.56
210	1894.44	1712.51	3003.05	3169.84	2085.88

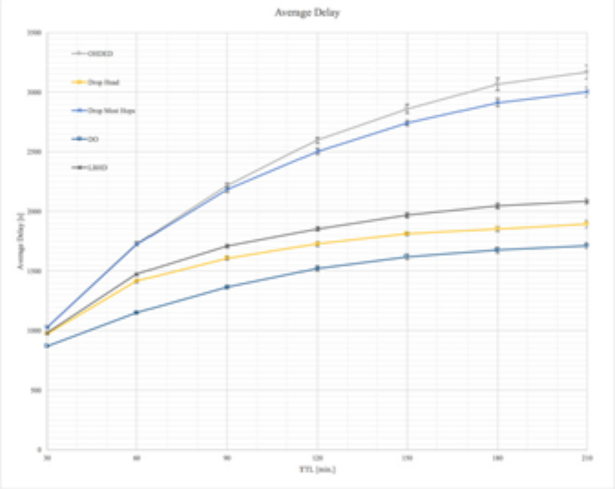

Figure 7. Average delay in seconds, varying the TTL of the message, between LBHD, OHDED and other drop policies paired with MaxProp routing protocol.

Furthermore, since that for TTLs larger than 90 minutes, both OHDED and Drop Most Hops achieve better performance results in terms of bundle delivery rate when compared to LBHD, it was expected that – bearing in mind equation (3) – they both would have better performance in terms of the overhead ratio. The OHDED drop policy achieves an average gain of 21%, 13%, 2% and 2% in performance when compared to Drop Head, DO, Drop Most Hops and LBHD, respectively, which implies that OHDED achieved the best performance in terms of overhead ratio.

Since the OHDED drop policy tends to drop the least probable bundle to be delivered by its carrying node either in direct delivery or doing two hops, it had the most successful results in terms of bundle delivery rate and overhead ratio, for larger TTLs. Considering that the number of delivered bundles is accounted in equation (3), by achieving the largest delivery rates, OHDED is able to decrease the overhead ratio.

Nevertheless, LBHD obtained the best results in terms of bundle delivery rate and overhead ratio in lower congestion scenarios – lower TTLs – also, in terms of average delay, its performance was better than the performance of the two other best policies (Drop Most Hops and OHDED) in all the simulated scenarios. Bearing in mind its results in terms of bundle delivery rate and overhead ratio for larger TTLs, and taking in consideration the excellent performance in terms of average delay, when compared to Drop Most Hops and OHDED, we may classify LBHD as the most efficient policy in all the performance metrics. Since the average gain in performance in average delay, when compared to Drop Most Hops, was of 25% and in the other performance metrics the gain was of 0%, it is concluded that the LBHD policy improves the Drop Most Hops policy – the standard drop policy in MaxProp routing protocol.

TABLE X. MAXPROP SIMULATION: OVERHEAD RATIO

TTL [min.]	Overhead Ratio				
	Drop Head	DO	Drop Most Hops	OHDED	LBHD
30	77.15	71.29	70.91	71.22	67.75
60	66.18	61.28	55.64	55.31	53.62
90	63.15	57.27	50.40	49.39	49.87
120	61.62	55.59	48.01	46.72	48.30
150	60.98	54.78	46.67	45.28	47.99
180	60.44	54.23	45.94	44.42	47.58
210	60.34	54.23	45.36	43.97	47.28

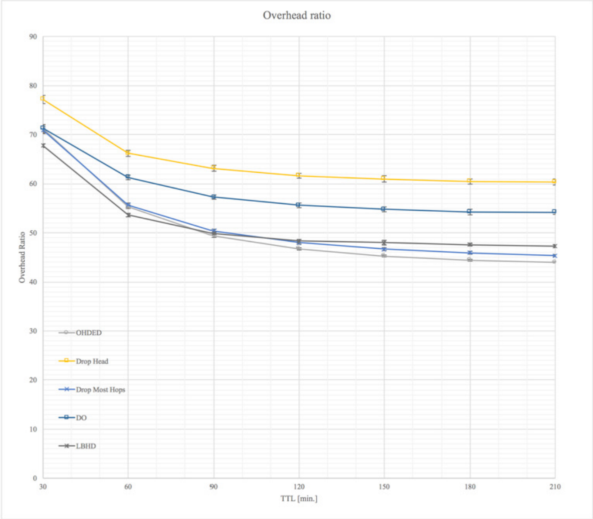

Figure 8. Overhead ratio, varying the TTL of the message, between LBHD, OHDED and other drop policies when paired with MaxProp routing protocol.

V. CONCLUSION AND FUTURE WORK

In brief, this paper proposes a new LBHD drop policy, which drops the bundles with the highest estimated delivery probability in congestion situations, as the probability of another node carrying the same bundle delivering it is high. An alternative drop policy call OHDED is also proposed to be paired with the MaxProp routing protocol, achieving not only the lowest overhead ratios but also the highest bundle delivery rates in high congestion scenarios.

When paired with PRoPHET, the LBHD policy achieved the best delivery rates, as compared with other existing drop policies, for a wide range of network loads, ranging from low congestion to high congestion. For low congestion, the overhead slightly increases, as more bundles are allowed to exist in the network. But as congestion increases, the overhead increase is minimal, as LBHD effectively controls congestion.

The LBHD policy achieved an average delivery rate 37% better than the default Drop Head (FIFO) policy, and an average of 3.9%, 14% and 17% better than DO, LEPR and MOPR, respectively. As regards the average delay, the improvement of the delivery rate under congestion comes with the price of an increased average delay, as the additional bundles that are delivered take the longest time to reach their destinations due to congestion and would be dropped with other dropping policies.

When paired with MaxProp, both proposed drop policies – LBHD and OHDED – had an excellent performance. It is important to refer that LBHD was the best performing policy especially in low congestion scenarios, having the best results in all the performance metrics. For larger TTLs, in terms of bundle delivery rate and overhead ratio, OHDED achieved the best results among all the simulated drop policies. However, taking also into account the average delay performance metric, the most consistent policy in all the performance metrics was LBHD. Bearing in mind that LBHD had a similar performance to the standard MaxProp drop policy (Drop Most Hops) in terms of bundle delivery rate and overhead ratio, and since that LBHD has an average gain of 25% in terms of delay when compared to Drop Most Hops, the proposed LBHD policy may be considered the most efficient one.

As future work, additional simulations of the policies in other delivery probability estimation routing protocols and with different movement models are suggested. Also, it is proposed to add a new feature that would allow the bundle holder's information to be spread throughout the network, i.e. by transmitting at each encounter, the information of to which nodes each bundle has been replicated to – as it is done in the MaxProp routing protocol with the bundle delivery acknowledgement messages. Moreover, it is proposed to relate the RTTL of each bundle to LBHD's HD metric to further improve the average delay. For congestion situations, incoming bundles could be rejected in some cases, instead of dropping existing bundles from the node's buffer, managing to decrease the overhead ratio while maintaining or increasing the delivery rate.

REFERENCES

[1] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," ACMSIGCOMMComput.Commun.Rev.,vol.34,no.4,p.145,2004.

[2] S. M. Tornell, C. T. Calafate, J. C. Cano, and P. Manzoni, "DTN protocols for vehicular networks: An application oriented overview," IEEE Commun. Surv. Tutorials, vol. 17, no. 2, pp. 868–887, 2015.

[3] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "A survey on congestion control for delay and disruption tolerant networks," Ad Hoc Networks, vol. 25, no. PB, pp. 480–494, 2015.

[4] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks," RFC 6693. IETF, 2012.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," Proc. IEEE Infocom, 2006.

[6] N. Benamar, K. D. Singh, M. Benamar, D. El Ouadghiri, and J. M. Bonnin, "Routing protocols in Vehicular Delay Tolerant Networks: A comprehensive survey," Comput. Commun., vol. 48, pp. 141–158, 2014.

[7] D. Patel and R. Shah, "A Survey on Improved PROPHET Routing Protocol in DTN," Int. Res. J. Eng. Technol., vol. 2, no. 8, pp. 1237–1240, 2015.

[8] Sukhbir and R. Singh, "Effective routing protocols for delay tolerant network," Int. J. Mod. Eng. Res., vol. 2, no. 4, pp. 1732–1735, 2012.

[9] J. Chen, "Dijkstra's shortest path algorithm," J. Formaliz. Math., vol. 15, no. 2, pp. 1–9, 2003.

[10] V. N. D. G. D. J. Soares and J. J. P. C. Rodrigues, "Performance Analysis of Scheduling and Dropping Policies in Vehicular Delay-Tolerant Networks," Int. J. Internet Technol., vol. 3, no. 1/2, pp. 1–9, 2010.

[11] R. Sulma, A. Qaisar, S. M. Z. M., and A. A.Hanan, "E- DROP An Effective Drop Buffer Management Policy for DTN Routing Protocols," Int. J. Comput. Appl., vol. 13, no. 7, pp. 8–13, 2011.

[12] Y. Harrati and A. Abdali, "MaxHopCount: A New Drop Policy to Optimize Messages Delivery Rate in Delay Tolerant Networks," Int. J. Interact. Multimed. Artif. Intell., vol. 4, no. 1, pp. 37–41, 2016.

[13] A. Rani, S. Rani, and H. S. Bindra, "Performance Evaluation of Mofo Buffer Management Technique With Different Routing Protocols in Dtn Under Variable Message Buffer Size," vol. 3, no. 2. International Journal of Engineering Research & Technology (IJERT), Sirsa, Haryana, India, pp. 82–86, 2014.