MaxHopCount: DTN congestion control algorithm under MaxProp routing

Youssef HARRATI† and Abdelmounaim ABDALI†,

Cadi Ayyad University, faculty of Sciences & Techniques / Laboratory of Applied Mathematics and Informatique (LAMAI), Marrakech, Morocco

Summary

Internet and communication networks have almost covered every area on the globe. Today the big challenge for classic internet, is maintaining connectivity in hard conditions like intermittent connectivity or power outages, and in difficult topographies such as under-water or Interplanetary Networks.

In these challenging environments, a new networking model has been proposed; it is called Delay Tolerant networks which follows the Store-Carry-and-Forward mechanism. Hence, a node may keep a message in its buffer for long time. And when a delivery or forward opportunity arises, it transmits it to other node(s). One of the big issues that confront this mechanism is the congestion of the buffer due to the big number of messages and the limited memory size. Here, in order to deal with this buffer overload, researchers have proposed buffer management algorithms, also called: Drop Policies.

In the current work, we propose a new Drop policy which we have compared to other existing policies in different conditions and environments, and we've noticed that it gives better result in term of delivered messages, network overhead and also latency average.

Key words:

Buffer management; Delay Tolerant Networks; Drop policy; Mobility models; Network Simulator (ONE); Scheduling.

1. Introduction

The classic networking and internet model [1] provides an end-to-end communication, which assumes that the path between source and destination is safe and that the bandwidth is very large. This traditional architecture cannot be applied in some special environment such as lack of path between source and destination, limited bandwidth interplanetary or underwater networks. In these challenging circumstances, researchers have proposed a new networking architecture called DTN (Delay Tolerant Networks).

The new concept DTN have been proposed by Kevin Fall et .al [2] in 2003. It became recently one of the most studied topics, and researchers consider it to be one of the future mobile networks aspects.

DTN is based on Store and Forward mechanism, hence, every node has its own storage where it stores received message until the appearance of other node which can drive it to its destination. The problem with this process is the limited node storage size, hence, when a node receives many messages, the buffer become full and the node is congested. In order to deal with this problem, researchers have proposed a set of drop policies.

In this paper, we propose a new drop policy called "MaxHopCount" [3] and we compare it to some existing buffer management policies with different routing protocols. Then we decide which routing protocol is better for our algorithm and what are its suitable conditions and environments.

The rest of this paper is organized as follows: Section 2 is about the state of the art where we give brief definition of some existing buffer management policies and other characteristic of DTNs. Section 3 introduces our new algorithm "MaxHopCount". Section 4 summarizes our simulation results and discussion. And finally, we reserved Section 6 for conclusion and future work.

2. State of the art

2.1 Drop policies in DTN

To avoid the node congestion and buffer overload problem, researchers have developed a set of buffer management policies. The quality of communication may differ from a policy to another depending on the environment and conditions (traffic density, area size, buffer size, TTL …). TABLE 1 gives brief definitions of some existing DTN drop policies:

Table 1: description of some buffer management policies

Policy	***Description***
Drop Front (FIFO)	The message which arrived first in the buffer will be selected first to be dropped. [4]

Manuscript received May 5, 2017
Manuscript revised May 20, 2017

Drop Tail (LIFO)	The buffer in this strategy is ordered in a LIFO manner. Hence, the message which arrives last to the queue will be the first dropped message.[4]
Drop Oldest (SHLI)	The message with the lowest time to live value (TTL) is the oldest one, and is the first selected message to be dropped.[5]
E-Drop (Equal Drop)	This policy deletes the message with the exact or the nearest size to new coming message. This strategy minimizes the number of dropped messages. [6]
Drop Youngest	The youngest message is the message with the longest remaining time to live (TTL), and it is the first dropped message when applying this policy. [5]
Drop Largest	Each message has a specific size; this buffer management strategy drops the message with the biggest size in the queue to free more space by dropping few messages. [7]
Evict Most Forwarded First (MOFO)	The message which has been forwarded to most number of nodes will be dropped first.[8]
Evict Most Favorably Forwarded First (MOPR)	Each message in a node has a forwarding predictability FP, initially set to 0. When the message is forwarded, the FP value increases. The message with highest FP value will be deleted first.[9]

2.2 Routing in DTN

Routing protocols in Delay Tolerant Networks are classified according to many criteria. Some can classify routing algorithms as replication-based algorithms (i.e:the protocol creates message replicas) or forwarding-based algorithms (i.e: the protocol doesn't replicate messages). Another taxonomy describes DTN routing protocols as random (Deliver messages randomly) or network topology based protocols (use network history data to deliver messages).

Epidemic routing, ProphetV2, Spray and wait, MaxProp, Rapid, First Contact and Direct Delivery are only few among other routing protocols which are included in all these classifications.

NB: All routing protocols in Delay Tolerant Networks are based on Store-carry and Forward mechanism.

2.2.1 Store and forward

During the movement of node from source to destination, which can last for a long delay, every node should carry messages in its buffer until their delivery.
Through the Store and forward mechanism **Erreur ! Source du renvoi introuvable.**, every node has a local storage buffer in which messages are saved and carried along the movement process. Depending on the size of this buffer, the number of carried messages is limited.
When the storage capacity is exceeded, new messages will be automatically rejected due to congestion which can negatively impact the delivery rate of the network.

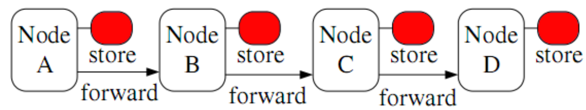

Fig. 1 Store and forward mechanism

2.2.2 Epidemic

Epidemic is the rapid spread of infectious disease to a large number of people in a given population within a short period of time. In DTN the same principle is used for Epidemic routing protocol. Here, every node transmits replicas of the message to newly discovered nodes. Theoretically, this algorithm needs to have unlimited buffer size and unlimited energy to give high delivery rate, but practically these conditions are impossible to implement.[10]

2.2.3 Prophet V2

The Prophet protocol "Probabilistic Routing Protocol using History of Encounters and Transitivity" Assumes that two nodes which meet very frequently, have a high probability to get in touch again.
The Prophet Algotithm uses this meeting probability to decide which node will carry the message at the next hop.[11]

2.2.4 MaxProp

In this protocol, every node contains a node-meeting probability vector of size n-1 (where n is the number of nodes in the network) when the current node meets another node k, the kth element of the probability vector is incremented. Then the vectors are exchanged between the two nodes. This stored information about the entire network will help compute the shortest path using a depth-first search.[12]

2.2.5 Spray & Wait

Spray and Wait combines the speed of epidemic routing with the simplicity of direct transmission. So, for the spray phase: for each message M at the source node, L copies of M are forwarded to L different relays (intermediate nodes). Then in the wait phase: if the destination is not found among the relays, every relay will perform direct transmission to the message destination only, and so on until each message in the source node buffer reaches its destination. [13]

2.2.6 FirstContact

A very simple routing algorithm for DTN, and a very quick one, this protocol forwards just a single copy of the message or the fragment of message to the first available contact.[14]

2.3 Nodes mobility in DTN

Nodes mobility in DTN networks is arbitrary and random. By geographically tracking real nodes movement and gathering sets of parameters such as speed, direction, acceleration…etc. Many mobility patterns (mobility models) have been proposed.
Some patterns are random, such as random walk or random waypoint while others are map based like Shortest Path movement.

2.3.1 Random walk

For this model, the node mobility is random, so a node chooses random direction and random speed before it starts walking. First, the node selects a random direction angle between 0 and 2π, then it chooses an arbitrary speed between [Vmin, Vmax] and finally after reaching t time it chooses other random values and starts walking again.[15]

2.3.2 Random waypoint

In opposite of Random walk, in random waypoint the node stops for a random pause period when it reaches a destination point before it starts walking again with other speed and direction angle.[15]

3. The new approach "MaxHopCount"

3.1 Description

During the message lifetime from the creation by the source node till it reaches the destination node, it traverses many other relay nodes. At every node the message stays for a period of time in the buffer, before jumping to other node's buffer. Every message has a set of information, containing its size, source node, destination… etc. and also it contains information about the "hop count", which refers to the number of nodes the message has passed through during its path from source to the current node. A high hop count means that the message has crossed lot of nodes, and then there may be lot of replicas at the network, so deleting this message from the current node's buffer will not impact the delivery. Otherwise, a low hop count means low number of replicas at the network so removing this message may mitigate the probability of delivery.

```
If the new message size is greater than the buffer size
        The message too big for the buffer
        Don't accept the new message
Endif
While the freeBufferSize is lower than the message size
        Initialize m to the message with the max hop count
        If m is null
                The buffer is empty
                We can't remove any more messages
                Don't accept the new message
        endif
        Delete message m
End while
Accept the message
```

3.2 The Flow-sheet

First, the function verifies if the buffer size is less than the new message size, if so; the function rejects the message because the buffer size is not enough and can't hold it. Otherwise,
the algorithm chooses the message with the highest Hop Count value and removes it from the buffer, and then the algorithm compares the free buffer space with the new (waiting) message size if there is enough space it accepts the message, if not it repeats the last action until there is enough space or there are no more messages in the buffer. Figure 2 below summarizes this algorithm.

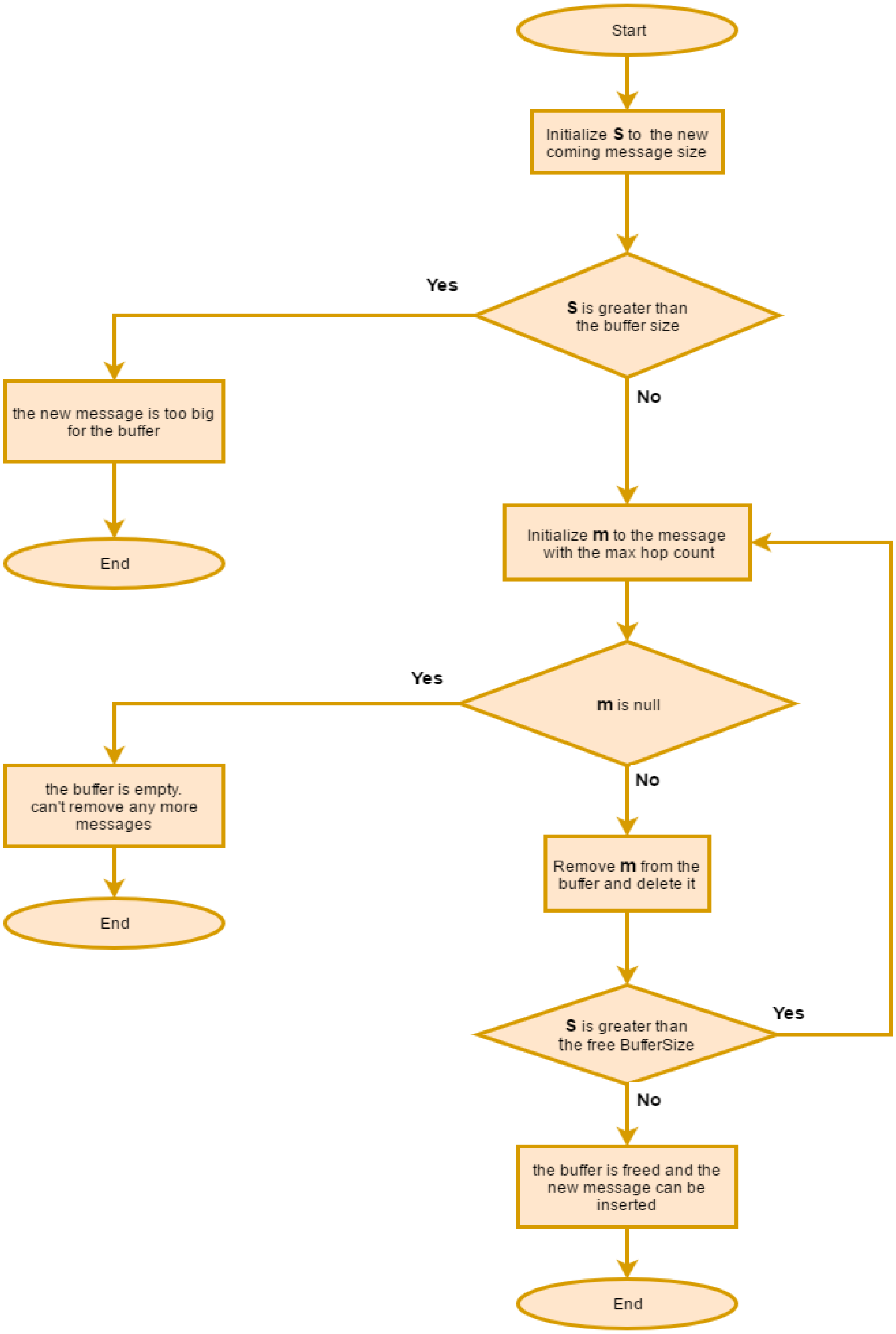

Fig. 2 our new algorithm MaxHopCount flowsheet

4. Results and discussion

4.1 Simulation tools

The ONE Simulator is a discrete event simulator written in Java. The major aspire of the simulator is to relate DTN (store-carry-forward) of message for long time, where the probability of disconnections and failures enlarged. [16]

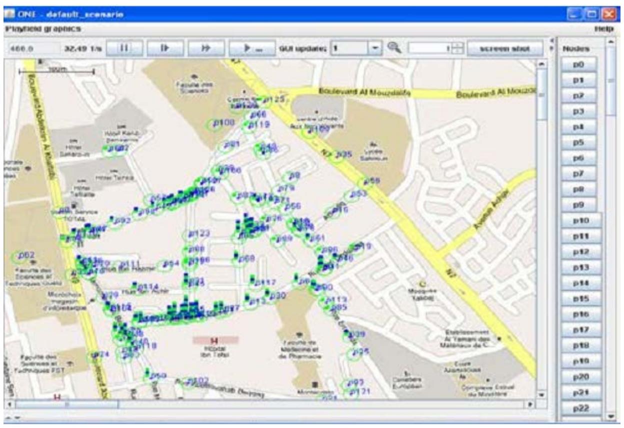

Fig. 3 ONE GUI mode of the ONE Simulator area: Cadi Ayyad University

4.2 Common metrics for performance evaluation

The following metrics are commonly used when evaluating scenarios related to DTN protocols. [17]

4.2.1 Delivery ratio.

Suppose that **M** be the set of all messages created in the network and $\mathbf{M_d}$ be the set of all messages delivered. Then, the delivery ratio is computed as:

$$M_d / M$$

4.2.2 Average latency of message delivery.

Now let the **i^th^** delivered message was created at time $\mathbf{c_i}$ and delivered at time $\mathbf{d_i}$. Then the average message delivery latency is computed as:

$$\left(\sum_{i=1}^{M_d}(d_i - c_i)\right) \Big/ M_d$$

4.2.3 Overhead ratio

Let $\mathbf{r_i}$ be the number of replications of any message $\mathbf{m_i}$ $\mathbf{\in M}$. Then the overhead ratio is determined as:

$$\left(\sum_{i=1}^{M}(r_i - M_d)\right) \Big/ M_d$$

4.3 Simulation parameters

In our simulations, we experience different environments by changing some parameters like routing protocols where we compare MaxProp, Epidemic and ProphetV2 routers. Table 2 contains the important parameters of our simulations.

Table 2 : SIMULATION PARAMETERS

Variable	***Value***
movement Model	RandomWalk - RandomWayPoint
Router	Epidemic – ProphetV2 – MaxProp
buffer Size	5M
drop Policy	FIFO – DL – DY – SHLI – MaxHopCount - MOFO
Message TTL (in minutes)	60 -120 -180 -240 -300
number of Hosts	200
Message creation interval (in seconds)	25,35
Messages size	500k,1M
World Size (width, height; meters)	500, 500

4.4 MaxHopCount With RandomWalk

In the first simulation we compare routing protocols to different TTL values, and then we analyze the results in term of delivery rate and overhead ratio. The drop policy this simulation is our algorithm MaxHopCount

4.4.1 Delivery rate

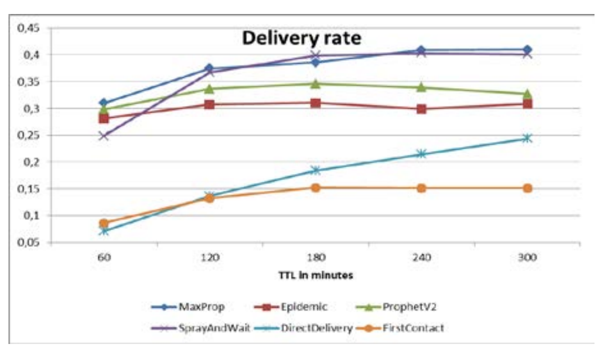

Fig. 4 Delivery rate as a function of TTL for different routers. The drop policy is MaxHopCount

As shown in the graph above Figure 4, our policy MaxHopCount, gives better delivery rate on MaxProp Routing protocol. Also we can notice that the rate of delivered messages keeps increasing for Maxprop protocol while it have a stable value for other routers.

4.4.2 Overhead Ratio

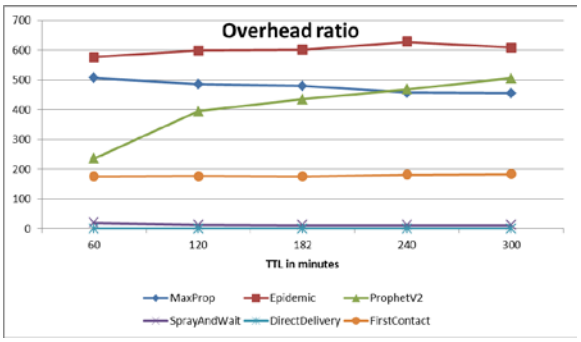

Fig. 5 Overhead ratio by TTL for different routers.

The Overhead ratio is the number of message replicas in the network, so a good DTN algorithm must have low Overhead ratio.

From Figure 4, we notice that when the TTL value is higher the overhead ratio of MaxHopCount is decreasing while this ratio keeps increasing for other policies. So our policy gives better results when the TTL is high.

4.4.3 Average latency

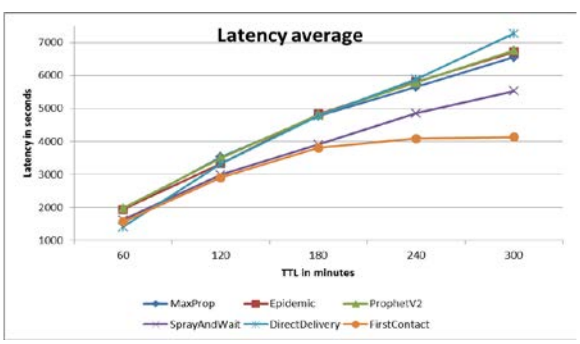

Fig. 6 Latency by TTL for different routers

The latency is the average time between message creation and its delivery to the destination. The goal of all DTN protocols is to have a low latency average.

From Figure 6 above, we observe that there is not a big difference between routing protocols in term of latency, the only factor that impacts this metric is the TTL value, hence, when the TTL is short, the latency is also short. But when the TTL increases, the Latency average goes bigger.

4.5 Comparing MaxHopCount to other drop policies - RandomWalk

For this second simulation, we have chosen MaxProp routing protocol as it gives the best delivery rate. Then we compared our proposed policy MaxHopCount with other different drop policies, and then we analyze the results in term of delivery rate and overhead ratio.

The mobility model we have chosen for this simulation is RandomWalk.

4.5.1 Delivery rate

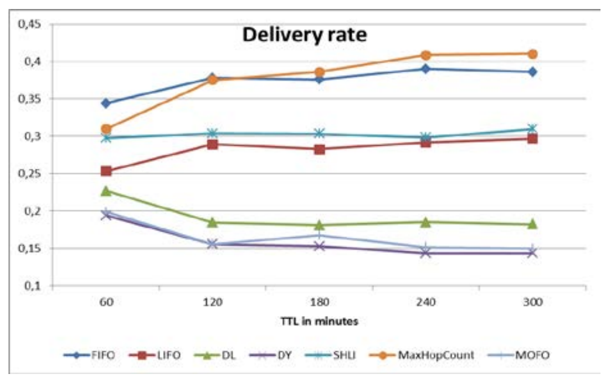

Fig. 7 Delivery rate as a function of TTL for different drop policies. The router here is MaxProp.

At this stage, we have fixed the router to MaxProp routing protocol then we compared our policy MaxHopCount to other drop policies by changing the TTL value.

As a result of this simulation we can note that our policy gives better results than other drop policies and as the TTL value grows up the delivery rate of MaxHopCount keeps increasing.

4.5.2 Overhead ratio

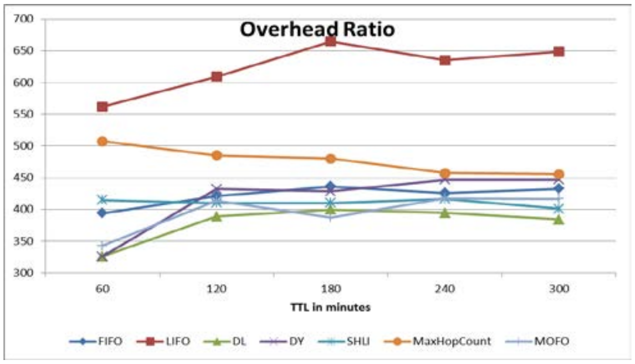

Fig. 8 Overhead ratio as a function of TTL for different drop policies. The router here is MaxProp.

From figure 8 we can notice that the overhead ratio of our policy is not the lowest one, but, in opposite of other policies, it keeps decreasing while the TTL value is high.

4.5.3 Average latency

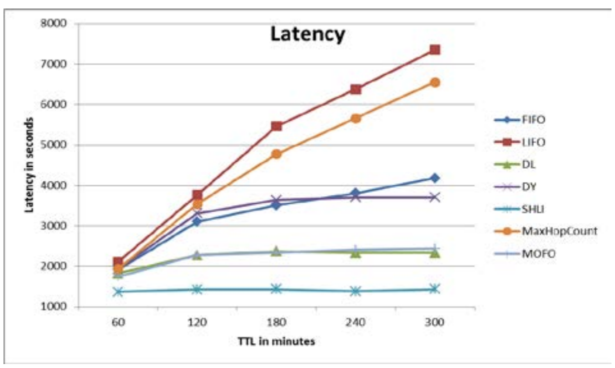

Fig. 9 Average Latency as a function of TTL for different drop policies.

The Average latency of our policy MaxHopCount is a bit higher than most other policies when the TTL is long, but for short TTL all buffer management policies have almost the same latency average.

4.6 Comparing MaxHopCount to other drop policies - RandomWayPoint

For this third simulation, we kept MaxProp as a routing protocol. Then we compared MaxHopCount algorithm to other different drop policies using RandomWayPoint movement model, and finally we analyzed the results in term of delivery rate, overhead ratio and average latency.

4.6.1 Delivery rate

Fig. 10 Delivery rate as a function of TTL for different drop policies.

The first chart gives an idea on the delivery rate of different policies by TTL. It is clear that MaxHopCount and FIFO have almost the same delivery probability and it is the highest one whatever is the TTL value. with another look on the chart we can notice that the TTL has actually no impact on the delivery in RandomWayPoint model while all line in the above graph are nearly stable.

4.6.2 Overhead ratio

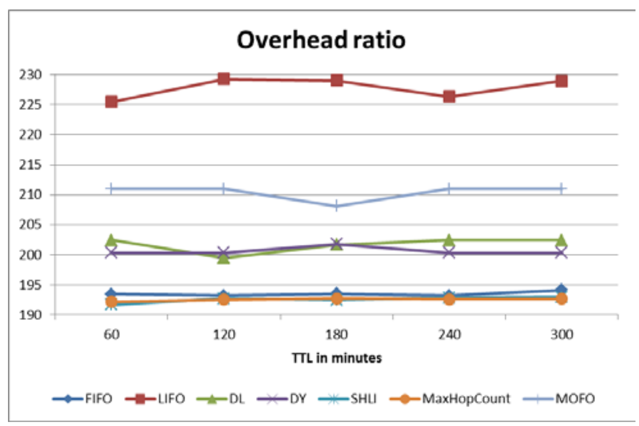

Fig. 11 Overhead ratio as a function of TTL for different drop policies.

The lowest overhead ratio for this simulation is MaxHopCount and Shortest TTL (SHLI) and they are a bit lower than FIFO policy so our policy does not overload the network by many messages for this simulation conditions and also here the TTL value has a very low impact on the overhead ratio.

4.6.3 Average latency

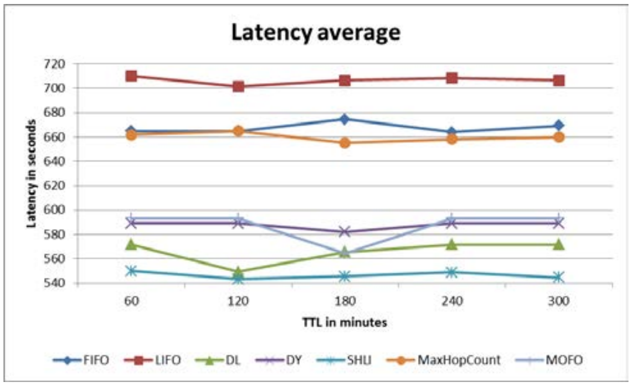

Fig. 12 Latency Average as a function of TTL for different drop policies.

In term of latency, all policies have a delivery time between 540 and 720 seconds (the difference is 180 seconds) so even if a policy has a better latency than the other, the difference stay very small and insignificant.

Here also we can easily perceive that the average of latency is nearly stable regardless of the TTL value.

5. Conclusion & future works

In this work, we compared different DTN drop policies to our proposed algorithm, and we discussed the results in term of delivery rate, overhead ratio, and Average Latency. The result we deduct is that our new policy MaxHopCount is optimal for high MaxProp routing protocol. Also, by changing the movement model has a big impact on this result. Because while the delivery rate increases proportionally to the TTL value for RandomWalk movement model, this rate keeps the nearly stable the same value for RandomWayPoint.

Our new policy has better delivery rate, latency and overhead than other policy and its best conditions are those we have experienced above, but it still a situation that needs improvement it is where two messages have the same number of hops, here we have to consider other criteria and apply other drop policy to deal with this ambiguity, that will be our future work.

References

[1] A. Arvidsson and P. Karlsson, “On traffic models for TCP/IP,” Teletraffic Sci. Eng., pp. 457–466, 1999.

[2] K. Fall, “A delay-tolerant network architecture for challenged internets,” in Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, 2003, pp. 27–34.

[3] 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, p. 37, 2016.

[4] S. Rashid, Q. Ayub, M. Soperi Mohd Zahid, and A. H. Abdullah, “E DROP An Effective Drop Buffer Management Policy for DTN Routing Protocols,” Int. J. Comput. Appl., vol. 13, no. 7, pp. 8–13, 2011.

[5] A. Krifa and B. Chadi, “An Optimal Joint Scheduling and Drop Policy for Delay Tolerant Networks Sous,” 2008.

[6] Y. Li, L. Zhao, Z. Liu, and Q. Liu, “N-Drop: congestion control strategy under epidemic routing in DTN,” in Proceedings of the 2009 international conference on wireless communications and mobile computing: connecting the world wirelessly, 2009, pp. 457–460.

[7] S. Rashid, Q. Ayub, M. Soperi Mohd Zahid, and A. H. Abdullah, “Impact of Mobility Models on DLA (Drop Largest) Optimized DTN Epidemic Routing Protocol,” Int. J. Comput. Appl., vol. 18, no. 5, pp. 35–39, 2011.

[8] a. Krifa, C. Baraka, and T. Spyropoulos, “Optimal Buffer Management Policies for Delay Tolerant Networks,” 2008 5th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Networks, 2008.

[9] S. Mansuri, H. Shah, and Y. Kosta, “Performance Analysis of Drop Policies for Different Mobility Models in DTN,” Int. J. Comput. Appl., vol. 59, no. 14, 2012.

[10] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Tech. Rep. number CS-200006, Duke Univ., pp. 1–14, 2000.

[11] A. El Ouadrhiri, I. Rahmouni, M. El Kamili, and I. Berrada, “Controlling messages for probabilistic routing protocols in Delay-Tolerant Networks,” in 2014 IEEE Symposium on Computers and Communications (ISCC), 2014, pp. 1–6.

[12] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks.,” in INFOCOM, 2006, vol. 6, pp. 1–11.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, 2005, pp. 252–259.

[14] K. Massri, A. Vernata, and A. Vitaletti, “Routing protocols for delay tolerant networks: a quantitative evaluation,” in Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, 2012, pp. 107–114.

[15] A. Keranen, “Opportunistic network environment simulator,” Spec. Assign. report, Helsinki Univ. Technol. Dep. Commun. Netw., 2008.

[16] A. Keränen, “The ONE Simulator for DTN Protocol Evaluation,” Proc. Second Int. ICST Conf. Simul. Tools Tech., p. 55, 2009.

[17] B. Saha, “Commonly Used Metrics for Performance Evaluation,” 2014. [Online]. Available: http://delay-tolerant-networks.blogspot.com/2014/03/commonly-used-metrics.html. [Accessed: 01-Sep-2016].

Youssef HARRATI ICT Engineer graduated in 2013 from the Cadi Ayyad University of Marrakech / Morocco. PhD student at the Laboratory of Applied Mathematics and Computer Science (LAMAI) of the doctoral studies center of Cadi ayyad University Marrakech / Morocco. His research interests are computer science, Delay Tolerant Networks, sensor networks and interplanetary networks.

Abdelmounaïm Abdali PhD in Solid Mechanics and Structures in University of Amiens in 1996, France. He is a Professor in computer science at the University Cadi Ayyad, Faculty of Sciences and Technics, Marrakech, Morocco. Member at the Laboratory of Applied Mathematics and Computer Science (LAMAI) Marrakech,Morocco. His research interests are computer science, DTN Network, business intelligence, Semantic web, Big Data , ubiquitous computing, biomechanics, bone remodeling, numerical simulation in solid mechanics and structures.