The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 12||Pages|| 01-10||2013||
ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805

Scheduling And Dropping Policies in Vehicular Delay Tolerant Network

[1,]Milind R. Penurkar , [2,]Aditi B. Gajipara
1,2,Dept. of Information Technology MIT College of Engineering
Pune,India

--ABSTRACT--
: Vehicular Delay-Tolerant Networking (VDTN) is projected as a new alternative of a delay tolerant network (DTN), designed for vehicular networks. The characteristic of VDTN are intermittent connectivity, high node mobility and shorted contact duration. Due to this messages are sent with more amount of delay and still so many of the messages are dropped. As VDTN networks are resource controlled, i.e. in terms of communication bandwidth and storage space, main challenge is to have scheduling and dropping policies that can get better overall performance of the network. This article investigates various scheduling and dropping policies in different routing scheme. We also have developed some new policy and compared its performance with other existing policies. It has been seen that our developed policies ought to give better results for an improved network performance in terms of delivery ratio and average delivery delay.

KEYWORDS *- Vehicular Delay Tolerant Networks; Delay-Tolerant Networks; Scheduling and Dropping Policies*

Date of Submission: 22 November 2013 Date of Acceptance: 20 December 2013

I. INTRODUCTION

The delay-tolerant network (DTN)[1] comes under the category of challenged network. These networks are presumed to experience sparse connectivity, no knowledge of end-to-end path, high latency, long transmission delay, and asymmetric data rates. Basically DTN was conceptualizing for the handling of interplanetary connectivity [3]. Now a day's DTN concepts are used at various scenarios like underwater networks [4], wildlife tracking networks [5], people network [6] and military tactical networks [7].Delay Tolerant Networking is based on Store-carry-and-forward routing principle. Figure 1 shows the comparison of TCP/IP model and DTN layered architecture. For this there is a new concept introduce that is of bundle layer. Over here data units distributed in this context, called *bundles*, are self-contained and application-level data units, which can often be large. The bundle layer binds the region specific lower layer so that application programs can communicate across multiple regions. The bundle layer stores and forwards entire bundle or bundle fragments among nodes.

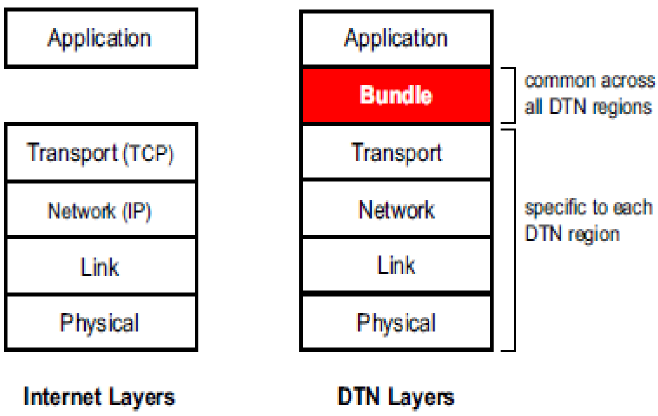

Fig.1 Generic network architecture layers: TCP/IP and DTN

In VDTN movement of vehicle & their message relaying service is used to enable network connectivity under unreliable condition. There are so many various routing algorithms have been developed for DTN's improved performance. Still there is not much focus on the scheduling of bundle and discarding policies. In VDTN there is constraint on buffer size means there is limited buffer and there is very short contact duration. So in available bandwidth it could be insufficient. Due to this we need efficient schedule policy to make a decision about which message should be selected when limited bandwidth is there and which message should be dropped when buffer is full, despite of any specific routing algorithm used. In this paper, we have studied the effect of scheduling and dropping policies on the performance of VDTN networks. This paper widens groundwork about the impact of scheduling and dropping policies with different routing algorithms for improving the bundles delivery time on VDTNs [8]. The paper has been extended with the introduction of some new scheduling and dropping policies, based on discrete criteria, and the performance assessment of the scheme through extensive simulation in various scenarios.

II. VEHICULAR DELAY-TOLERANT NETWORKS

The DTN architecture perception has also been extended to transit networks, called Vehicular Delay Tolerant Network (VDTN). In VDTN vehicles (e.g., cars, buses, and boats) are oppressed to offer a message relaying service by moving around the network and collecting messages from source nodes.

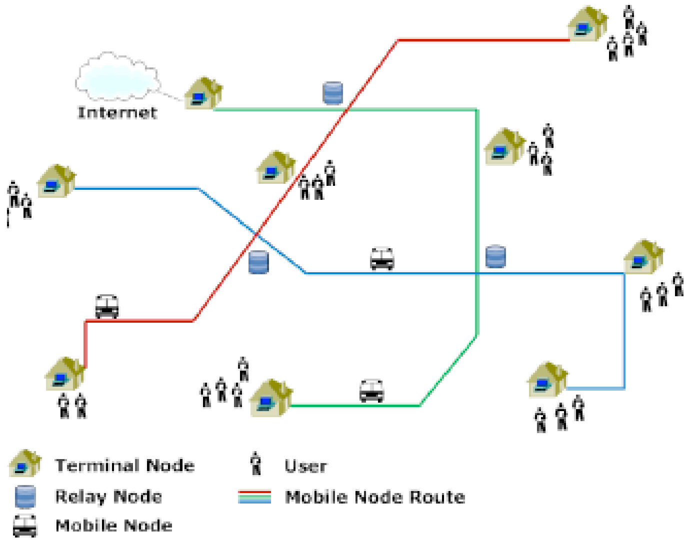

Fig.2 Example of a Vehicular Delay-Tolerant Network providing connection on rural and remote regions

The VDTN architecture is based on three different types of nodes as shown in Figure 2: *Terminal Nodes*, *Data Mules* and *Relay Nodes. Terminal nodes* are access points to VDTN. They provide the connection to end-users, and let them use non-real time applications. Among all the terminal nodes there is at least one node may have direct access to the Internet. *Data Mules (Mobile Ubiquitous LAN Extension)* are liable for physically carrying data between terminal nodes and they can exchange data with one another also. They extend the network coverage and increase communication opportunities. *Relay Nodes* are the fixed devices with low power needs and story-and-forward capabilities. Generally they are located at road intersections and allow the passing *Data Mules* to collect and leave data on them.Most of the problems in vehicular networks arise from the mobility and speed of vehicles that are responsible for a highly dynamic network topology and short contact durations. Limited transmission ranges, radio obstacles due to physical factors (e.g., buildings, tunnels, terrain and vegetation), and interferences (i.e., high congestion channels caused by high density of nodes), lead to disruption, intermittent connectivity, and significant loss rates. All these conditions make vehicular networks subject to frequent fragmentation/partition (i.e., end-to-end connectivity may not exist), resulting in small effective network diameter.As DTN, VDTN also uses the store-carry-and-forward paradigm to cop up with the problem of sparse connectivity. VDTN nodes store the message in their buffer while they are waiting for contact opportunities to forward message to other node or to the *terminal node (destination).*

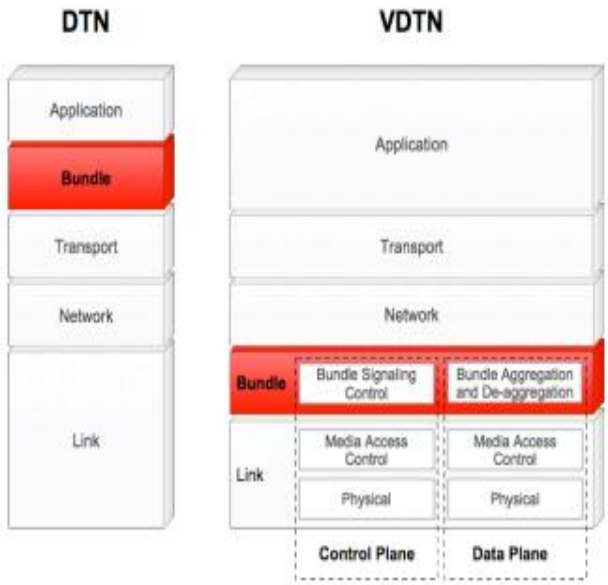

Fig.3 DTN and VDTN network layer architecture

VDTN also defines bundle as its protocol data unit. VDTN architecture is different from DTN architecture as shown in Figure 3. In VDTN architecture there is a clear separation between control plane and data plane. The bundle layer is further divided into Bundle Aggregation and De-aggregation (BAD) Layer and Bundle Signaling Control (BSC). BAD is located below the network layer in order to aggregate incoming IP data packets into bundle messages. BSC is responsible for executing the control plane functions, such as signaling messages exchange, node localization, resources reservation (at the data plane) and routing, among others. The signaling messages include information such as, but not limited to, node type, geographical location, route, velocity, data plane link range, power status, storage status, bundle format and size, delivery options, and security requirements, among others.

III. PROBLEM STATEMENT

VDTNs are exemplifying by very high node mobility, which results in frequent topological changes and network partition. Due to that, the node density and the mobile nodes mobility pattern have a direct effect over the transmission opportunities, contact durations, and inter contact times.Moreover, in such challenged scenarios, long-term storage is frequently combined with replication based routing schemes [10]. By creating several replicas of bundles and spread them across network improves delivery rate and decreases delivery latency. But, in a resource constrained network, these techniques can cause contention for network resources (e.g., bandwidth and storage), and can greatly influence the performance of routing protocols [11-13].

So we need more proficient scheduling policies to decide the order by which bundles can be transmitted at the immediate contact opportunities, and efficient drop policies to decide which bundles are dropped when a node's buffer is completely occupied. Though, scheduling policies and dropping policies play significant role in improving the overall performance of any DTN-based network.Lindgren and Phanse [14] have compared the performance of Epidemic [15] and PRoPHET [16] routing protocols by different combinations of queuing and forwarding policies. They show that different schedule and drop policies leads to performance improvement of routing algorithm.Krifa *et al.* [17] studied Epidemic routing. They propose an optimal buffer management policy. This policy can either maximize the average delivery probability or minimize the average delivery delay.

IV. SCHEDULING AND DROPPING POLICIES

Each VDTN node must apply various queuing mechanism which is a part of the resource allocation mechanism to manage in which way data bundle should be buffered, how they could accepted and how they could –assed on.

.

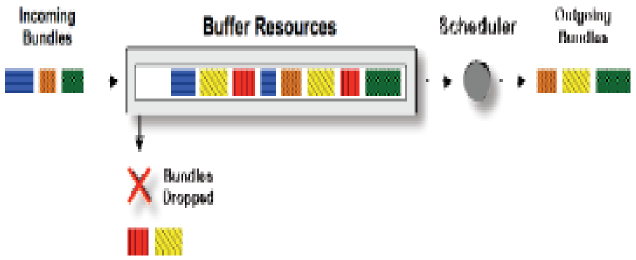

Fig.4 Illustration of a queuing discipline composed by a scheduling and a dropping policy.

As shown in Figure 4, the queuing discipline has both scheduled and drop policy. The scheduling policy decides the order by which bundles are passing on at a contact opportunity. The dropping policy chooses bundles to be discarded when buffer overflow occurs. This section describes various scheduling and dropping policies, and make out some variation which can be applied to them. Their performance is evaluated and compared by the simulation study in Section V.

A. Scheduling Policies

The following scheduling policies are considered and studied in this work.

FIFO: In this policy bundles are ordered to be transmitted at a contact opportunity, based on their arrival time at the node's buffer (based on a FCFS (first-come, first-served) approach).

Random: The limitation of Vehicular Delay Tolerant Network is finite bandwidth and nodes are having limited contact duration. Due to this issues FIFO approach may only serve bundles that arrived first to the node's buffer. To avoid this situation, random scheduling policy is used. This selects bundles randomly within a queue.

B. Dropping Policies

The following dropping policies are considered in this work.

Drop Head: In Drop Head policy whatever bundle has been stored for the longest period of time in the node's buffer has been discarded in this policy. Bundle/s have been dropped to create available apace for the next incoming bundle.

Random: When a receiving buffer is congested, this dropping policy randomly selects one of the bundles within a queue to be dropped.

We have introduced two new schedule and drop policy. This has been as follow.

```
Input: DTNHost
Output: Refreshed message list after drop
1:  NODE_ENERGY(DTNHost H)
2:  Define Threshold Energy TE
3:  IF H's Current Energy is less than TE Call DROP()
4:  ELSE Accept Message
5:  DROP()
6:  Define minttl // Threshold TTL
7:  SCAN all the message in the list
8:  CHECK If M starts with 'G'  then For All M starts with 'G' find M which is having lowest TTL &
    REMOVE it.
9:  UPDATE Buffer Content
```

Algorithm 1. NODE_ENERGY Algorithm

Here, in Algorithm.1 we are giving DTNHost as an input parameter. Every time message exchange occurs at that time we are checking the energy of the node. If the energy of node is higher threshold energy than only node accepts the message and schedule it for forwarding else we go for drop policy. In drop policy we are checking two parameters. First of all we are setting a value of minimum TTL.

Then we check that from the list if incoming message is starting from G (General message) then from all these messages whichever is having lowest TTL we will discard that message and at last we are updating the buffer content. In Algorithm 2, we are scheduling the incoming messages on the basis of two parameters that is buffer occupancy and priority. First we are checking the priority of the incoming message. Here, Accidental message are having highest priority then Traffic messages are there and at last General messages are there which are having the lowest priority. Then we check buffer occupancy. If there is free space then we accept the message and schedule it for forwarding but, if not then we are using dropping policy. We call drop policy to occupy the message with highest priority.In drop policy we are checking two parameters. First of all we are setting a value of minimum TTL. Then we check that from the list if incoming message is starting from A (Accidental message) then from all the messages which are general or traffic message we find whichever is having lowest TTL. Then we will discard those messages until we can occupy higher prioritized message. At last we are updating the buffer content.

```
Input : Message List
Output: Refreshed message list after drop
1:  SEND_QUEUE_BUFFER()
2:  SCAN all the message in the list
3:  CHECK incoming message M starts with 'A',
    'T', 'G'
4:  CHECK Free Buffer Size
5:  IF FreeBufferSize is greater than Size of M
    Accept message
6:  ELSE
7:  While Size of M is less than FreeBufferSize
8:  CALL DROP(M)
9:  DROP(M)
10: Define minttl // Threshold TTL
11: SCAN all the message in the list
12: CHECK If M starts with 'G' Then For All M
    starts with 'G' find M which is having lowest
    TTL & REMOVE it.
13: UPDATE Buffer Content
14: Else if M starts with 'T' Then For All M starts
    with 'T' find M which is having lowest TTL &
    REMOVE it.
15: UPDATE Buffer Content
```

Algorithm 2: SEND_QUEUE_BUFFER Algorithm

V. PERFORMANCE ANALYSIS

This section studies the effect of the above described scheduling and dropping policies on the performance of a vehicular delay-tolerant network. The study was carrying out by simulation using a modified version of the Opportunistic Network Environment (ONE) simulator [18]. ONE was modified to support the VDTN layered architecture model proposed in [15]. Additional modules were developed to implement the scheduling and dropping policies. Next subsections describe the simulation scenario and the corresponding performance analysis.

A. Simulation Scenario Parameters

The simulation scenario is based on a map-based model of a part of the city of Helsinki presented in Figure 4.). Mobile nodes (e.g. vehicles) move on the map roads between random locations, with random pause times between 5 and 15 minutes. The mobile nodes average velocity is between 30 Km/h and 50 Km/h. Each of the mobile nodes has buffer size of 25 Megabytes to 50 Megabytes. To increase the number of contact opportunities, five stationary relay nodes were placed at the road as may be seen in Figure 6. Each stationary relay node has a 500 Megabytes buffer.Data bundles are generated using an inter-bundle creation interval that is uniformly distributed in the range of [10,30] (seconds), and have random source and destination vehicles.
Data bundles size is uniformly distributed in the range of [250 KB, 2 MB] (bytes). Bundles have a time-to-live (TTL) 300 minutes, across the simulations, and are discarded when the TTL expires.

Fig.5 Helsinki simulation scenario (area of 4500×3400 meters) with the locations of the stationary relay nodes.

Increasing TTL leads to having more bundles stored at the network nodes' buffers, and during larger periods of time. Therefore, more bundles will be exchanged between network nodes, and this will also potentially increase buffer overflows. All network nodes use a data plane link connection with a transmission data rate of 4.5 Mbps and an omni-directional transmission range of 30 meters, as proposed in [19].Performance metrics considered in this study are the bundle delivery probability (measured as the relation of the number of unique delivered bundles to the number of bundles sent), as well as the bundle delivery delay (measured as the time between bundles creation and delivery). We measure the different performance results for the combination of the above-described scheduling and dropping policies.

B. Performance Analysis for various Scenario

Results are based on following Important Metrics- (i) Contact opportunities, (ii) Delivery probability, (iii) Delivery latency and (iv) Message TTL. Here in this graph, few transmission opportunities are registered when two relay nodes are deployed in the network (Figure 6)

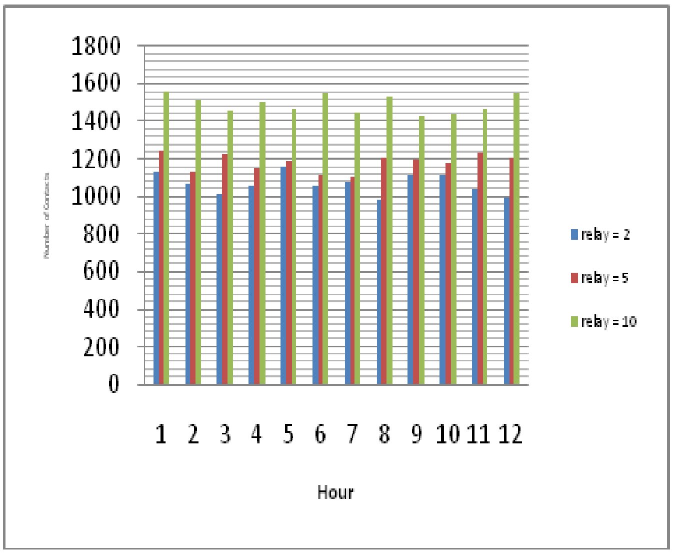

Fig.6 Number of contacts per hour between all network nodes.

Deploying more relay nodes augments the number of contact opportunities per hour among all network nodes. Introducing ten relay nodes increases the number of contacts. This effect suggests that relay nodes will contribute to increase in the number of messages exchanged between vehicles.

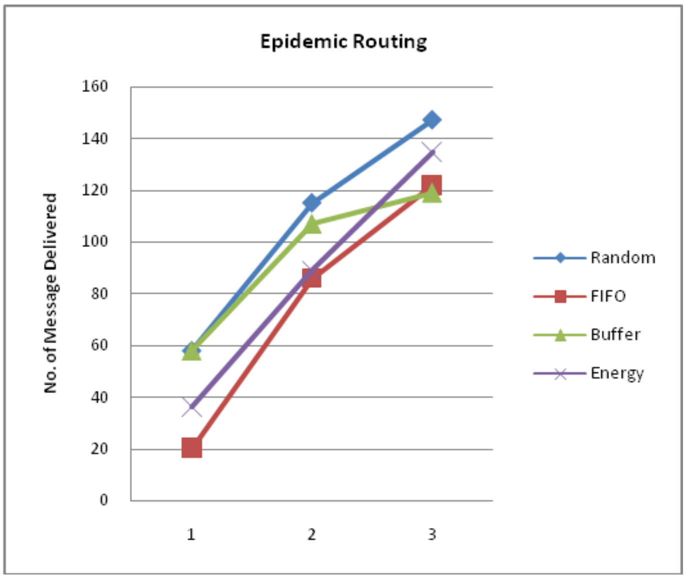

Fig.7 Number of messages delivered in Epidemic Routing

Figure 7 shows that as we increases the TTL number of delivered messages increases. In Epidemic routing we are using flooding mechanism. Over here random policy outperforms all other policy. After that SEND_QUEUE_BUFFER and then NODE_ENERGY policy is there.

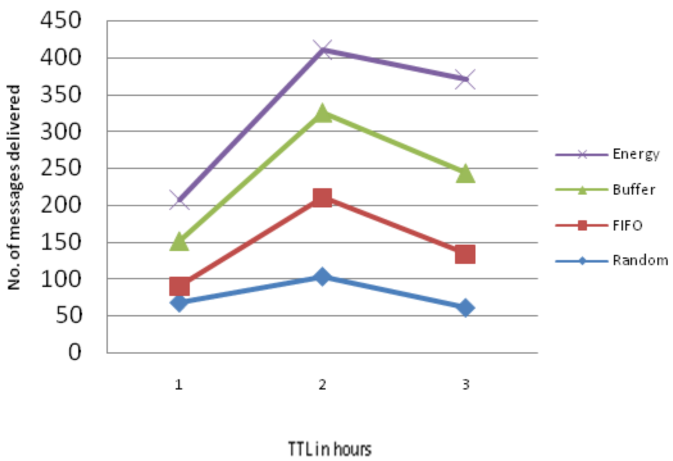

Fig.8 Number of messages delivered in FirstContact Routing

Figure 8 shows that in FirstContact routing NODE_ENERGY policy outstand all other policies. After that SEND_OUEUE_BUFFER then FIFO and then Random schedule policy is there. Here we are getting 20% better result compare to FIFO and Random scheduling policies.

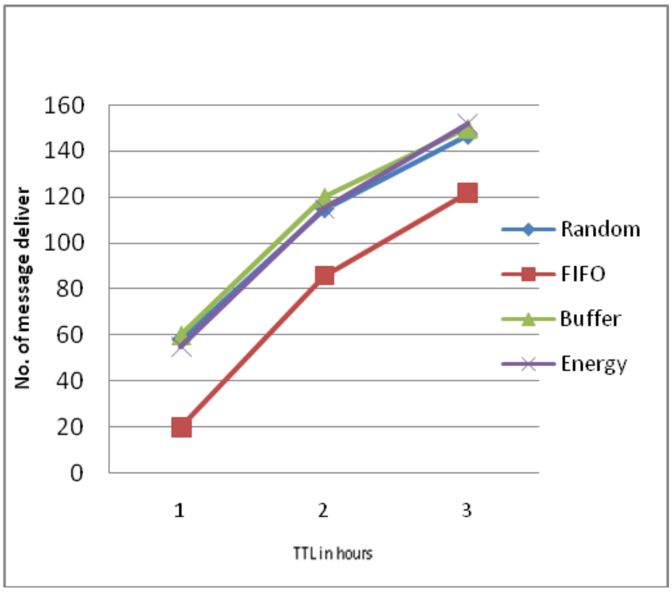

Fig.9 Number of messages delivered in Spray and Wait Routing

In Spray and Wait, as shown in Figure 9 the best results are of SEND_QUEUE_BUFFER. After that NODE_ENERGY policy is there. Here these two policies' results are almost equivalent. After that random policy is there and then FIFO is there.

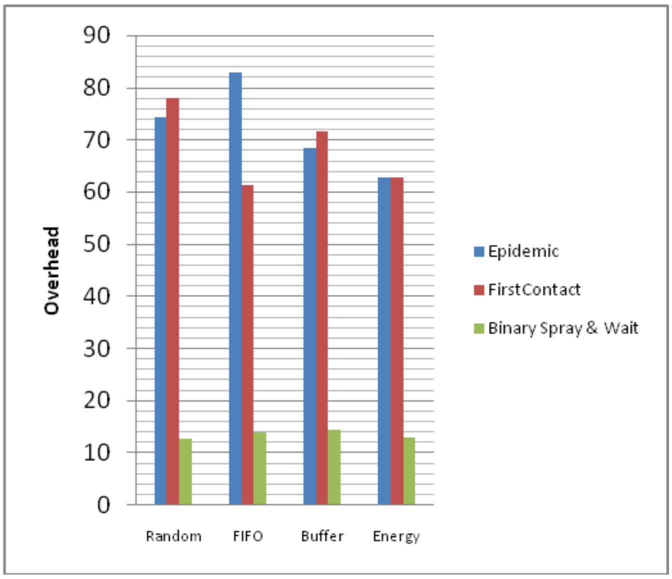

Fig.10 Overhead for various scheduling policy in different routing algorithm

Figure 10 shows Overhead for all the Scheduling policies with different routing algorithm. As we can see over here that there is highest amount of overhead in Epidemic routing as in Epidemic routing we are using flooding strategy. Due to that there are number of replicas are there in the network even for a single message. Random and FIFO are the in-build policies. Compare to that my implemented polices are showing less overhead. Overall SEND_QUEUE_BUFFER policy is having 10% less overhead compare to random and FIFO. While NODE_ENERGY is having overall 15% less overhead compare to all the policies.

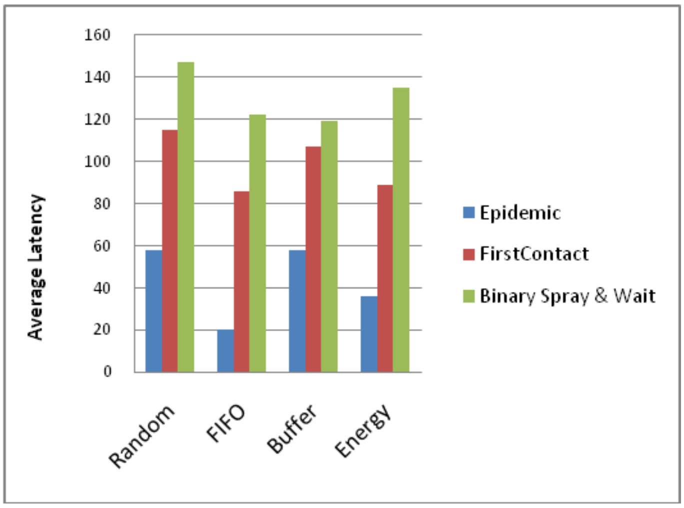

Fig.11 Average Latency for all scheduling policy with different routing algorithm.

The message average delay (Figure 11) is an interesting metric, since minimizing it reduces the time that messages spend in the network and reduces the contention for resources in the network (e.g. buffer). All routing protocols register similar values for the message average delay, and that relay nodes do not significantly affect this metric.

VI. CONCLUSIONS AND FUTURE WORK

This paper focused on the impact of scheduling and dropping policies on the performance of vehicular delay tolerant networks. This work tried to find a good alternative to the traditional FIFO scheduling with “drop head” dropping policy, which would improve the VDTN network performance. In this context, several combinations of scheduling and dropping policies were proposed, and their relative performance was analyzed in terms of bundle delivery probability and average delivery delay. These policies were enforced on various routing scheme. The simulation results reveal a good performance obtained by a combination of a scheduling policy and a dropping policy that gives preferential treatment to bundles with less TTL and energy. It has been shown that such an approach outperforms the commonly used FIFO scheduling and “drop head” buffer management, in both performance metrics. This result was obtained and confirmed for all simulation scenarios. For future work, we plan to investigate the use of scheduling and routing strategies based on geographical information for VDTNs.

REFERENCES

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC 4838, April 2007, [Online]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc4838.txt.

[2] K. Fall and S. Farrell, "DTN: An Architectural Retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, June 2008.

[3] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-Tolerant Networking: An Approach to Interplanetary Internet," in *IEEE Communications Magazine*, vol. 41, 2003, pp. 128-136.

[4] J. Partan, J. Kurose, and B. N. Levine, "A Survey of Practical Issues in Underwater Networks," in *1st ACM International Workshop on Underwater Networks, in conjunction with ACM MobiCom 2006*, Los Angeles, California, USA, Sep. 25, 2006, pp. 17 - 24.

[5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 5, pp. 96-107, 2002.

[6] N. Glance, D. Snowdon, and J.-L. Meunier, "Pollen: Using People as a Communication Medium," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 35, no. 4, pp. 429-442, March 2001.

[7] T. Olajide and A. N. Washington, "Epidemic Modeling of Military Networks using Group and Entity Mobility Models," in *5th International Conference on Information Technology :New Generations (ITNG 2008)*, Las Vegas, Nevada, USA, April 7-9, 2008, pp. 1303-1304.

[8] V. N. G. J. Soares, J. J. P. C. Rodrigues, P. S. Ferreira, and A. Nogueira, "Improvement of Messages Delivery Time on Vehicular Delay-Tolerant Networks," in *The 38th International Conference on Parallel Processing (ICPP- 2009) Workshops - The Second International Workshop on Next Generation of Wireless and Mobile Networks (NGWMN- 09)*, Vienna, Austria, September 22-25, 2009.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-copy Case," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 1, pp. 77-90, February 2008.

[10] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," in *ACM SIGCOMM 2007*, Kyoto, Japan, August 27-31, 2007, pp. 373-384.

[11] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24-37, 2006.

[12] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "Evaluating the Impact of Storage Capacity Constraints on Vehicular Delay-Tolerant Networks," in *The Second International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2009)*, Colmar, France, July 20-25, 2009, pp. 75-80.

[13] A. Lindgren and K. S. Phanse, "Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks," in *First International Conference on Communication System Software and Middleware (COMSWARE 2006)*, Delhi, India, January 8-12, 2006, pp. 1-10.

[14] A. Vahdat and D. Becker, "Epidemic Routing for Partially- Connected Ad Hoc Networks," Duke University, Technical Report CS-2000-06, April, 2000.

[15] A. Lindgren, A. Doria, E. Davies, and S. Grasic,"Probabilistic Routing Protocol for Intermittently Connected Networks," draft-irtf-dtnrg-prophet-02, March 9, 2009, [Online]. Available: http://tools.ietf.org/html/draft-irtf-dtnrgprophet- 02.

[16] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal Buffer Management Policies for Delay Tolerant Networks," in *Fifth Annual IEEE Communications Society Conference on Sensor,Mesh and Ad Hoc Communications and Networks – SECON 2008*, San Francisco, California, USA June 16-20, 2008, pp. 260-268.

[17] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Second International Conference on Simulation Tools and Techniques (SIMUTools 2009)*, Rome, March 2-6, 2009.

[18] A. Keränen and J. Ott, "Increasing Reality for DTN Protocol Simulations," Helsinki University of Technology, Networking Laboratory, Technical Report, July, 2007.