Performance Analysis of Scheduling and Dropping Policies in Vehicular Delay-Tolerant Networks

Vasco N. G. J. Soares
Instituto de Telecomunicações /
U. Beira Interior, Covilhã, Portugal
Polytechnic Institute of C. Branco,
Castelo Branco, Portugal
vasco.g.soares@ieee.org

Farid Farahmand
Department of Engineering Science,
Sonoma State University
California, USA
farid.farahmand@sonoma.edu

Joel J. P. C. Rodrigues
Instituto de Telecomunicações /
Department of Informatics
University of Beira Interior,
Covilhã, Portugal
joeljr@ieee.org

Abstract—**Vehicular Delay-Tolerant Networking (VDTN) was proposed as a new variant of a delay/disruptive-tolerant network, designed for vehicular networks. These networks are subject to several limitations including short contact durations, connectivity disruptions, network partitions, intermittent connectivity, and long delays. To address these connectivity issues, an asynchronous, store-carry-and-forward paradigm is combined with opportunistic bundle replication, to achieve multi-hop data delivery. Since VDTN networks are resource-constrained, for example in terms of communication bandwidth and storage capacity, a key challenge is to provide scheduling and dropping policies that can improve the overall performance of the network. This paper investigates the efficiency and tradeoffs of several scheduling and dropping policies enforced in a Spray and Wait routing scheme. It has been observed that these policies should give preferential treatment to less replicated bundles for a better network performance in terms of delivery ratio and average delivery delay.**

Keywords- Vehicular Delay-Tolerant Networks; Delay-Tolerant Networks; Scheduling Policies; Dropping Policies; Performance Analysis

I. INTRODUCTION

The delay-tolerant network (DTN) architecture [1] was conceived to support data communication in highly challenged environments characterized by any combination of the following aspects: sparse connectivity, network partitioning, intermittent connectivity, long propagation delays, asymmetric data rates, high error rates, and even the potential non-existence of a contemporaneous end-to-end path. To handle these issues, the DTN architecture introduces a bundle layer, which builds a store-and-forward overlay network above the transport layers of underlying networks [2].

Although DTN architectural concepts were initially proposed to deal with interplanetary connectivity [3], over the last years they have been applied in terrestrial environments over a wide range of application scenarios including underwater networks [4], wildlife tracking networks [5], sparse wireless sensor networks [6], transient networks [7-9], disaster recovery networks [10], people networks [11], and military tactical networks [12].

Vehicular networks are another example of networks that can benefit from the application of the DTN paradigm [13-15]. It is important to note that these networks are characterized by a highly dynamic network topology, and short contact durations, which are caused by the high velocity of vehicles [16, 17]. In addition, limited transmission ranges, physical obstacles, and interferences, lead to connectivity disruption and intermittent connectivity issues [18]. Furthermore, these networks may be partitioned, because of the large distances usually involved and to low node density. Hence, a complete path from source to destination may not exist for most of the time.

The vehicular delay-tolerant network (VDTN) architecture has been proposed to deal with these challenging connectivity issues. VDTN architecture is based on the principle of asynchronous, bundle-oriented communication from the DTN architecture. However, the design of the VDTN network architecture, and its protocol layering, considers an Internet protocol (IP) over VDTN approach, and features an out-of-band signaling approach, with the separation between the control plane and the data plane [15].

The effective operation of a VDTN relies on the cooperation of network nodes to store-carry-and-forward data bundles. In addition, routing protocols may perform bundle replication to discover more possible paths, and thus increase the bundle delivery rate and decrease the delivery delay. However, the problem is that the combination of bundle storage during large periods of time and their replication leads to high storage and bandwidth overhead. As network nodes have limited resources, this may degrade the overall network performance. To tackle this problem, scheduling and dropping policies are used, determining bundle replication order at the contact opportunities, and taking bundle drop decisions when buffer space is exhausted.

This work studies the influence of scheduling and dropping policies on the performance of VDTN networks in terms of the delivery ratio and the delivery delay. This paper extends a preliminary contribution about the impact of scheduling and dropping policies for improving the bundles delivery time on VDTNs [19]. The paper has been extended with the introduction of new scheduling and dropping policies, based on distinct criteria, and the performance evaluation of the proposals through extensive simulation in different scenarios.

The remainder of this paper is organized as follows. Section II presents a brief overview of the VDTNs background, while Section III describes the problem statement, and related work. Section IV proposes the scheduling and dropping policies studied in this work. Section V focuses on the comparative analysis of the proposed approaches and Section VI concludes the paper and points further research directions.

II. VEHICULAR DELAY-TOLERANT NETWORKS

Vehicular delay-tolerant networking has been proposed to address challenged vehicular communications [15]. Although VDTN architecture is based on the DTN store-carry-and-forward model of routing, it presents unique characteristics such as *i)* IP over DTN approach; *ii*) control plane and data plane decoupling; and *iii*) out-of-band signaling.

Figure 1 shows a comparison between DTN and VDTN network architecture layers. As may be seen, DTN architecture introduces a bundle layer that creates a store-and-forward overlay network, allowing the interconnection of highly heterogeneous networks [20]. On the contrary, the VDTN architecture follows another approach based on IP over DTN, placing the DTN-based layer over the data link layer. Like on DTN architecture, bundles are also defined as the protocol data unit at the VDTN bundle layer. However, in a VDTN, a bundle aggregates several IP packets with common characteristics, such as the same destination node or generated with data from the same application.

The DTN store-carry-and-forward paradigm is used to recover from network partitions, and to cope with node sparsity. According to this paradigm, the network nodes keep the bundles on their buffers, while waiting for contact opportunities to forward them to intermediate nodes, or to the final destination. This long-term storage paradigm exploits opportunistic contacts between network nodes that arise with the vehicles mobility, to bring bundles closer and closer to destination.

Another distinctive feature of the VDTN architecture is the separation between control and data planes, as illustrated in Figure 1. The VDTN bundle layer is divided into the following sublayers: bundle signaling control (BSC) and bundle aggregation and de-aggregation (BAD). 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. BAD controls the data plane functions that deal with data bundles. These functions include, among others, buffer management and scheduling, traffic classification, data aggregation/de-aggregation, and forwarding.

VDTN uses out-of-band signaling, meaning that the control plane uses a separate, dedicated, low-power, low bandwidth, and long-range link to exchange signaling information. On the contrary, the data plane uses a high-power, high bandwidth, and short-range link to exchange data bundles. While the control plane link connection is always active to allow node discovery, the data plane link connection is active only during the estimated contact duration time, and if there are data bundles to be exchanged between the network nodes. Otherwise, it is not activated. The use of out-of-band signaling procedures is described in [15, 21], and offers considerable benefits because it not only ensures the optimization of the available data plane resources [21], but also allows saving power, which is very important for power-limited network nodes [15, 22].

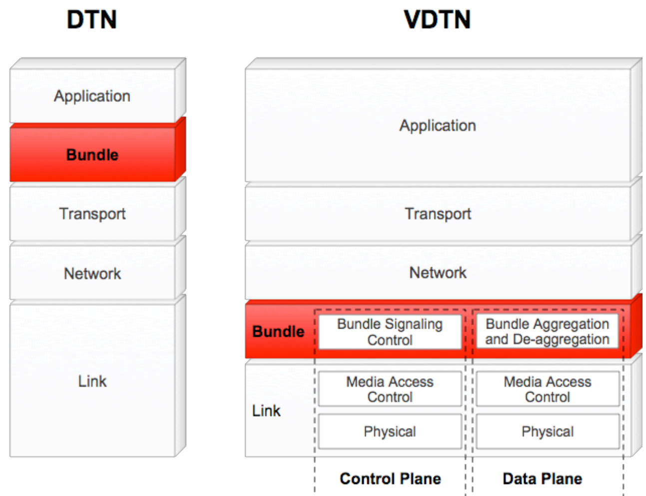

Figure 1. DTN and VDTN network architecture layers.

Figure 2 shows data exchange between two types of network nodes in a VDTN: mobile nodes (e.g., vehicles), and stationary relay nodes. Mobile nodes opportunistically collect and disseminate data bundles. Stationary relay nodes are power limited, fixed devices, which are located at road intersections. These nodes increase contact opportunities in scenarios with low node density. They allow passing by mobile nodes to pickup and deposit data on them. Thus, they contribute to increase the bundles delivery ratio and to decrease their delivery delay [23].

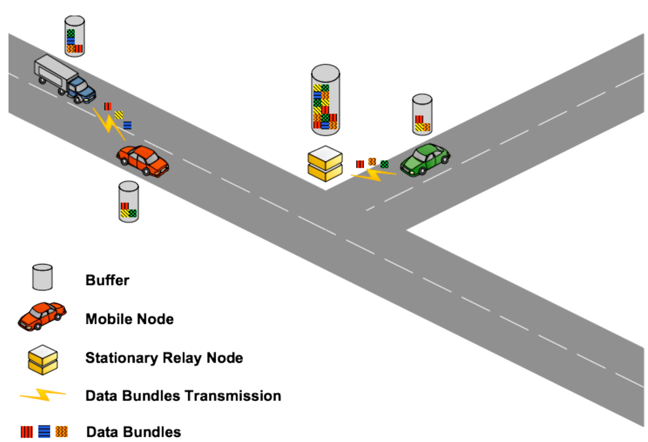

Figure 2. Illustration of VDTN network nodes exchanging data bundles.

III. PROBLEM STATEMENT

VDTNs are characterized by very high node mobility, which results in frequent topological changes and network partition. In these networks, the node density and the mobile nodes mobility pattern have a direct effect over the observed transmission opportunities, contact durations, and inter-contact times.

Figures 3 and 4 illustrate the effect of the mobile nodes (e.g., vehicles) density and mobile nodes velocity in a simulation scenario detailed in Section V. As expected, Figure 3 shows that the number of contact opportunities is directly related to the number of mobile nodes available on a network scenario. Moreover, it allows concluding that the number of contact opportunities grows as the mobile nodes velocity increases. As may be seen, this effect is more pronounced for the scenario with 50 mobile nodes.

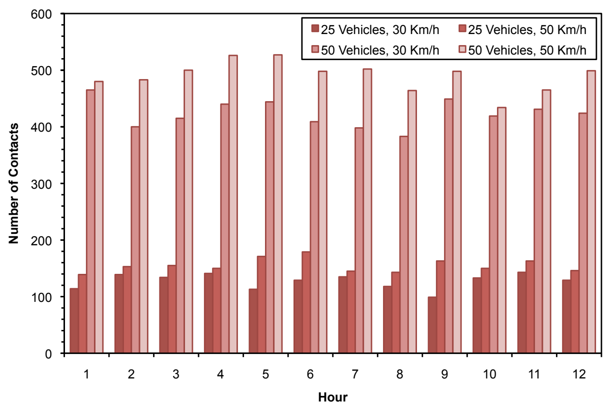

Figure 3. Number of contacts per hour between network nodes, with 25 or 50 vehicles moving at a speed of 30 Km/h or 50 Km/h.

Figure 4 shows that VDTN network nodes register short contact durations, due to the velocity of mobile nodes. As mobile nodes move at a faster speed, more contacts are registered, but the contact duration decreases even more. This impacts bundles transmission, since the available bandwidth is further restricted, which may turnout to be insufficient to transmit all intended bundles.

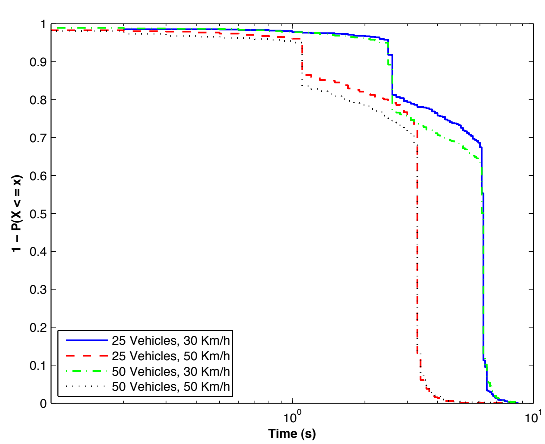

Figure 4. Contact durations with 25 or 50 vehicles moving at a speed of 30 Km/h or 50 Km/h.

Moreover, in such challenged scenarios, long-term storage is often combined with replication-based routing schemes [24]. Spreading multiple copies of bundles to several network nodes improves the delivery rate and/or reduces the delivery latency. However, 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 [25-27].

This emphasizes the need for efficient scheduling policies to decide the order by which bundles are transmitted at the brief contact opportunities, and efficient drop policies to decide which bundles are discarded when a node's buffer is full. Although scheduling policies and dropping policies play an important role in improving the overall performance of any DTN-based network, to the best our knowledge, little research has been done in this field.

In [28], Lindgren and Phanse compare the performance of Epidemic [29] and PRoPHET [30] routing protocols when different combinations of queuing and forwarding policies are used. They show that these policies can optimize the limited system resources utilization, leading to performance improvement of the routing protocols, in terms of message delivery, overhead, and end-to-end delay. They also conclude that when bandwidth is limited, it is not enough only to decide what messages should be forwarded, but also the order in which they must be forwarded.

In [31], Zhang *et al.* present an analysis of buffer-constrained Epidemic routing. Simple buffer management policies are evaluated. The authors conclude that with adequate buffer management schemes, smaller buffers can be used without negative impact on the delivery ratio observed with this routing protocol.

In [32], Krifa *et al.* also base their study on Epidemic routing. The authors consider the theory of encounter-based message dissemination to propose an optimal buffer management policy based on global knowledge about the network. This policy can either maximize the average delivery probability or minimize the average delivery delay.

In [33], Erramilli and Crovella observe that it is important to study forwarding and dropping policies independently of each other. The focus of their work is on comparing message prioritization schemes (for transmission or dropping) that do not take into account network information with schemes based on delegation forwarding algorithms [34]. The authors conclude that the latter schemes perform better in terms of delivery rate, delay and cost. Based on their results, the authors also state that forwarding policies have less impact in the network performance than dropping policies.

In [35], Li *et al.* study the impact of buffer management strategies under Epidemic routing. They propose a congestion control mechanism called N-Drop. This policy takes into account the number of times a message has been forwarded, and a threshold related to the size of the buffer, to decide which messages should be dropped when buffer overflow occurs.

IV. Scheduling and Dropping Policies

As part of the resource allocation mechanisms, each VDTN network node must implement some queuing discipline that governs how data bundles are buffered while being carried and waiting to be transmitted. This queuing discipline, illustrated in Figure 5, consists of both a scheduling and a dropping policy. The scheduling policy determines the order in which bundles are transmitted at a contact opportunity. The dropping policy selects bundles to be dropped upon buffer overflow.

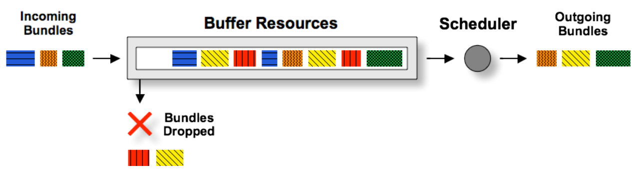

Figure 5. Illustration of a queueing discipline composed by a scheduling and a dropping policies.

This section describes several scheduling and dropping policies, and identifies some variations that can be applied to them. Their performance is evaluated and compared through a simulation study in Section V.

A. *Scheduling Policies*

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

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

Random: Due to short contact duration and finite bandwidth, FIFO approach may only serve bundles that arrived first to the node's buffer. To avoid this situation, random scheduling policy selects bundles randomly within a queue.

Remaining Lifetime (RL): Both FIFO and Random scheduling policies do not take into account the time-to-live (TTL) of bundles. TTL is a timeout value that expresses the amount of time that bundles should be stored before being discarded, since they are no longer meaningful. Remaining Lifetime scheduling policy orders bundles based on their remaining TTL. Two variations of this scheduling policy are considered, either (*i*) bundles with smaller remaining TTLs are scheduled to be sent first (RL Ascending Order) or (*ii*) bundles with longer remaining TTLs are scheduled to be sent first (RL Descending Order).

Replicated Copies (RC): This scheduling policy assumes that nodes keep track of the number of times each bundle has been replicated. Hence, two variations of RC policy may be considered. On the first, bundles that have been less replicated are scheduled to be sent first (RC Ascending Order) or, in the second, bundles that have been more replicated are scheduled to be sent first (RC Descending Order).

Both Remaining Lifetime and Replicated Copies scheduling policies need a tiebreaking rule. In the case of RL scheduling policy, a node may store in its buffer more than one bundle with the same remaining lifetime. The same happens with RC scheduling policy, where a node may store bundles that have been replicated the same number of times. FIFO or Random scheduling policies can be applied to tiebreak these cases.

B. *Dropping Policies*

The following dropping policies are considered in this work.

Drop Head: This dropping policy discards the bundle that has been stored for the longest period of time in the node's buffer, 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.

Remaining Lifetime (RL): Remaining Lifetime dropping policy selects bundles that get discarded based on their remaining TTL. Two variations of this policy are considered, either (*i*) the bundle with the smallest remaining TTL is discarded first (RL Ascending Order) or (*ii*) the bundle with the longest remaining TTL is discarded first (RL Descending Order).

Replicated Copies (RC): When buffer overflow occurs, the number of times each bundle has been replicated can be used to decide which bundle should be dropped. The following two variations of RC dropping policy may be used: (*i*) the bundle that has been less replicated is dropped first (RC Ascending Order) or (*ii*) the bundle that has been more replicated is dropped first (RC Descending Order).

For the same above-mentioned reasons, a tiebreaking rule must be used for both Remaining Lifetime and Replicated Copies dropping policies. FIFO or Random scheduling policies can be applied to tiebreak.

V. Performance Analysis

This section investigates the effect of the above described scheduling and dropping policies on the performance of a vehicular delay-tolerant network. The study was conducted by simulation using a modified version of the Opportunistic Network Environment (ONE) simulator [36]. 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 6. During a 12 hours period of time (e.g., from 8:00 to 20:00), mobile nodes (e.g. vehicles) move on the map roads between random locations, with random pause times between 5 and 15 minutes. To obtain scenarios with different numbers of contact opportunities we change the number of mobile nodes between 25 and 50 across the simulations. We also vary the mobile nodes average velocity between 30 Km/h and 50 Km/h, to obtain scenarios with different contact durations. Each of the mobile nodes has a 25 Megabytes buffer.

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.

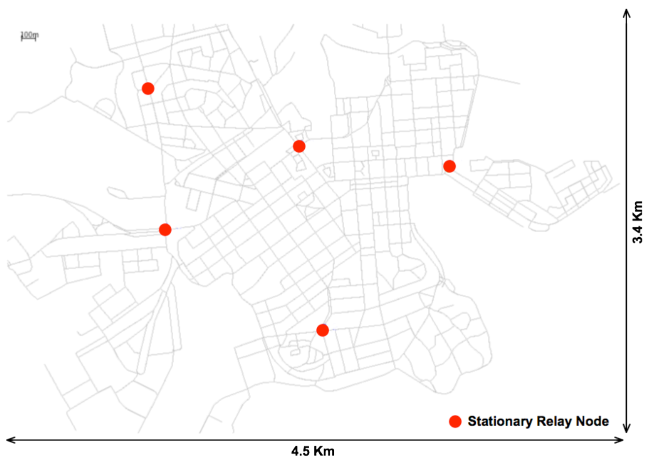

Figure 6. Helsinki simulation scenario (area of 4500×3400 meters), with the locations of the stationary relay nodes.

Data bundles are generated using an inter-bundle creation interval that is uniformly distributed in the range of [15, 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) that changes between 30, 60, 90, 120, 150, and 180 minutes, across the simulations, and are discarded when the TTL expires. 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 [37].

Spray and Wait [38] is used as the underlying DTN routing scheme. Spray and Wait routing limits the number of bundle replicas (copies), and it assumes two main phases. In the "spray phase", for each original bundle, L bundle copies are spread to L distinct relay nodes. At the "wait phase", using direct transmission, it waits until any of the L relays finds the destination node. This work considers a binary spraying method, where the source node starts with a number of copies N (assuming 12, in this study) to be transmitted ("sprayed") per bundle. Then, at any node A that has more than 1 bundle copies and encounters any other node B that does not have a copy, forwards to B $N/2$ bundle copies and keeps the rest of the copies. When a node carries only 1 copy left, it only forwards it to the final destination.

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, presented at Table I. A designation for each scheduling/dropping policy pair was created in order to improve chart readability in the next subsection. FIFO policy is used as a base-case of comparison with the other proposed policies. The results presented in the next subsection are averages from 12 simulation runs.

TABLE I. COMBINED SCHEDULING AND DROPPING POLICIES

Designation	Scheduling Policy	Dropping Policy	Tie-break
FIFO	FIFO	Head Drop	-
Random	Random	Random	-
RL ASC	Remaining Lifetime *Ascending Order*	Remaining Lifetime *Descending Order*	FIFO
RL DESC	Remaining Lifetime *Descending Order*	Remaining Lifetime *Ascending Order*	FIFO
RC ASC	Replicated Copies *Ascending Order*	Replicated Copies *Descending Order*	RL DESC
RC DESC	Replicated Copies *Descending Order*	Replicated Copies *Ascending Order*	RL DESC

B. *Performance Analysis for a Scenario with 25 Vehicles*

The evaluation study starts with a comparison of the delivery probability registered when 25 vehicles move with a speed of 30 Km/h. Figure 7 shows that when the initial bundles' TTL is lower than 120 minutes, FIFO, Random, RL ASC, and RC DESC policies register similar delivery probabilities. However, when the TTL is great than 120 minutes, RC DESC policy performs much worse than the other policies. This means that scheduling bundles that have been more replicated to be sent first, is not a good option.

Enforcing a RL DESC policy and, therefore, giving preferential treatment to bundles with larger remaining lifetimes, leads to increase the delivery ratio when compared to those policies. Since bundles exchanged between network nodes will have longer remaining lifetimes, this increases their probability to be relayed more times between network nodes, until eventually reaching the destination. This figure shows that when the initial bundles' TTL is equal or lower than 150 minutes, RL DESC increases the delivery probability about 3% for TTL=60min., 5% for TTL=90min., 5% for TTL=120min., 4% for TTL=150min., and 2% for TTL=180min., when compared to the traditional FIFO policy. For a TTL of 180 minutes, RC DESC and FIFO register a similar delivery probability.

RC ASC policy improves these results further. This policy gives preferential treatment to bundles that have been less replicated, using as a tiebreak criterion for bundles that have been replicated the same amount of times, a second scheduling policy - RL DESC. As may be observed, when RC ASC policy is compared to FIFO, it provides up to 3%, 6%, 7%, 6%, 6% and 5% of gain in delivery ratio, respectively, across all simulations.

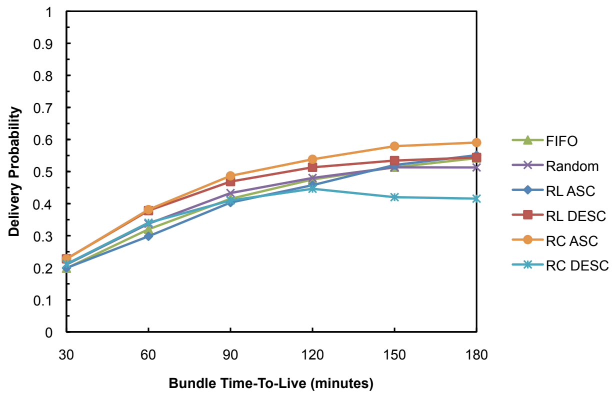

Figure 7. Bundle delivery probability as function of TTL in a scenario with 25 vehicles moving at a speed of 30 km/h.

Figure 8 shows a comparison between average delay and (initial) bundles TTL, for the combinations of scheduling and dropping policies considered. Average delay is an interesting metric, since minimizing the delivery delay reduces the time that bundles spend in the network and, thus, reduces the contention for resources.

As expected, the observed results show that deploying a RL DESC based policy and, therefore, giving preferential treatment to bundles with larger remaining TTLs, decreases the bundle average delay considerably. This policy performs better than the others in this performance metric. On the contrary, the other variant of RL policy - RL ASC - gives preferential treatment to bundles with lower remaining lifetimes, trying to deliver them before expiring. This results in the worst average delays across all simulations.

FIFO criterion based on the order of bundle arrival to the buffer, also leads to longer average delays. When RL DESC policy is compared to FIFO, bundles arrive at the destination nodes approximately 1, 3, 8, 14, 16, and 18 minutes sooner, in average. As a final note to Figures 7 and 8, it can be seen that RC ASC outperforms all the other policies in terms of delivery probability, presenting the second best results in terms of delivery delay, due to its tiebreak criterion.

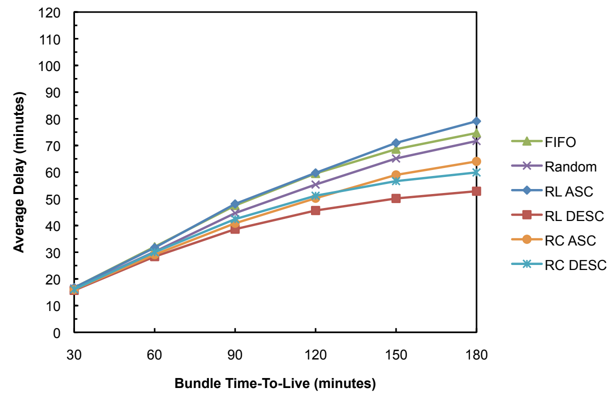

Figure 8. Bundle average delay as function of TTL in a scenario with 25 vehicles moving at a speed of 30 km/h.

Although the increase of vehicles average speed from 30 Km/h to 50 Km/h increases the number of contact opportunities (Figure 3), it decreases the contact duration (Figure 4). Hence, the number of bundles exchanged during a contact opportunity also decreases. As may be seen through the comparison between results shown in Figures 7 and 9, this resulted in lower delivery ratios for all combinations of scheduling and dropping policies, across all simulations. In this scenario, the difference in terms of performance between the policies decreases. The values of delivery ratio for RL DESC are close to the ones observed with RC ASC. Nevertheless, Figure 9 shows that RC ASC still performs better than the other policies, increasing about 3%, 4%, 8%, 8%, 7%, and 5% for each of the considered values of TTL, the bundle delivery probability, when compared to FIFO. Figure 10 shows that FIFO and RL ASC present the higher average delivery delay values. The difference, in terms of the average delay observed for RL DESC and RC ASC, increased for TTLs higher than 90 minutes, when compared to the results shown in Figure 7.

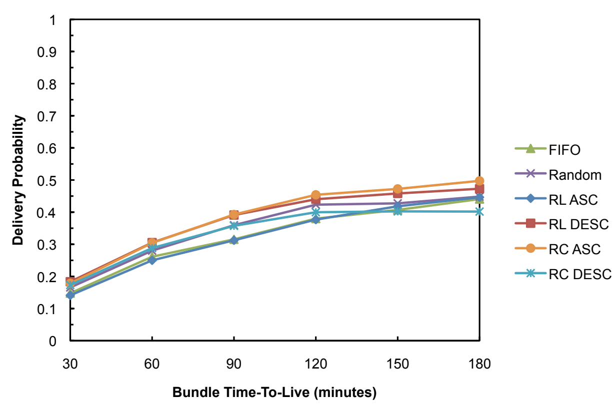

Figure 9. Bundle delivery probability as function of TTL in a scenario with 25 vehicles moving at a speed of 50 km/h.

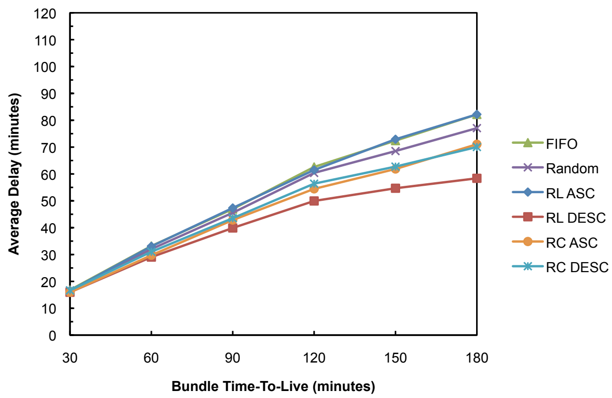

Figure 10. Bundle average delay as function of TTL in a scenario with 25 vehicles moving at a speed of 50 km/h.

C. *Performance Analysis for the Scenario with 50 Vehicles*

The second scenario considers 50 vehicles moving across the map (Figure 6). As shown in Figure 3, increasing node density the number of contact opportunities also increases. Then, it increases the number of relayed bundles and, potentially, causes more contention for network resources. Recall that the traffic generated is equal on both scenarios.

A comparison between results depicted in Figures 7 and 11 shows that policies have the same behavior and the delivery probability increases for all policies (Figure 11), when compared with the first scenario (Figure 7). Furthermore, this analysis also reveals that RC ASC registers the best results in terms of the delivery ratio, irrespective of the number of mobile nodes. In this second scenario, when vehicles move with an average speed of 30 Km/h, it presents gains of 4%, 9%, 6%, 6%, 4% and 5% for each of the considered values of TTL, respectively, when compared to the FIFO policy (Figure 11).

Figure 12 confirms the conclusions obtained in the first scenario. Although RC ASC policy requires slightly more time to deliver bundles than RL DESC, it achieves a higher delivery ratio (Figure 11).

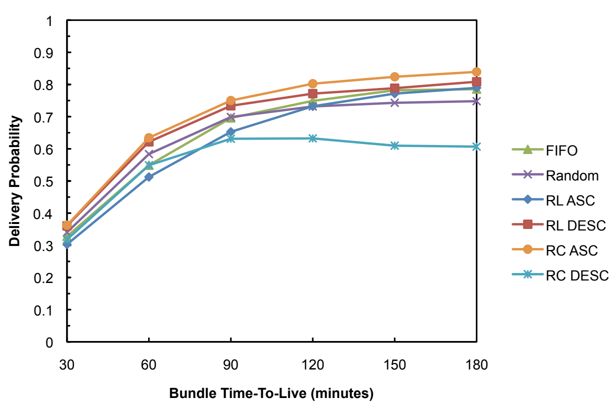

Figure 11. Bundle delivery probability as function of TTL in a scenario with 50 vehicles moving at a speed of 30 km/h.

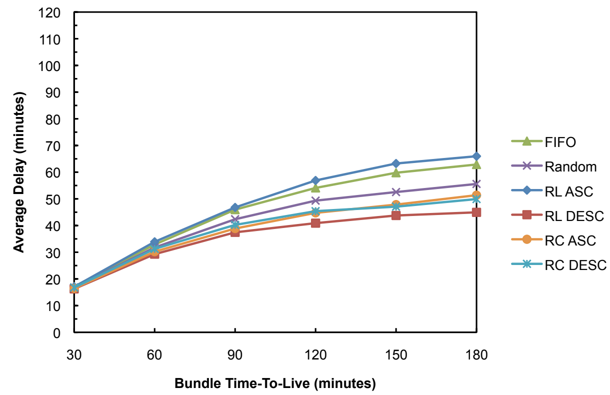

Figure 12. Bundle average delay as function of TTL in a scenario with 50 vehicles moving at a speed of 30 km/h.

Following the results shown in the previous scenario, increasing the vehicles average speed to 50 Km/h, decrease the number of successfully delivered bundles (Figures 11 and 13). However, it is interesting to observe that RC ASC policy is less affected by this change than the remaining policies. Due to this fact, RC ASC shows greatest improvements. Compared to FIFO, RC ASC increases the delivery probability in 6%, 11%, 11%, 10%, 6%, and 7% for each of the considered values of TTL, respectively.

As previously observed, the gains in the delivery ratio performance metric are attenuated when bundles have a large TTL. This is due to the fact that network nodes have large buffers and can carry and exchange these bundles during longer periods of time before expiring. However, increasing the TTL reinforces the improvement on average delay. When comparing with FIFO, bundles arrive at the destination nodes approximately 1, 4, 8, 12, 17, and 20 minutes sooner in average, if RC ASC policy is used.

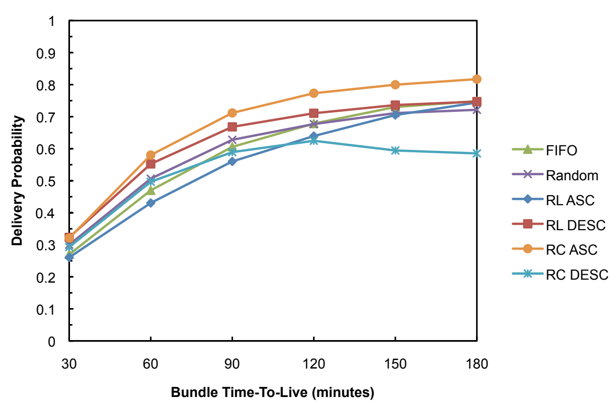

Figure 13. Bundle delivery probability as function of TTL in a scenario with 50 vehicles moving at a speed of 50 km/h.

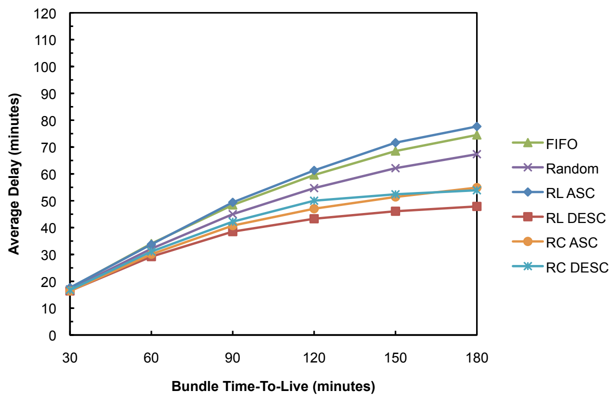

Figure 14. Bundle average delay as function of TTL in a scenario with 50 vehicles moving at a speed of 50 km/h.

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 a Spray and Wait routing scheme.

This study considered two urban scenarios with different node densities and contact durations. The simulation results reveal a good performance obtained by a combination of a scheduling policy and a dropping policy that gives preferential treatment to less replicated bundles. 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.

Acknowledgments

Part of this work has been supported by *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, in the framework of the Project VDTN@Lab, and by the Euro-NF Network of Excellence of the Seventh Framework Programme of EU, in the framework of the Project VDTN.

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] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 11, no. 3, pp. 327-339, June 2006.

[7] N4C and eINCLUSION, "Networking for Communications Challenged Communities: Architecture, Test Beds and Innovative Alliances," [Online]. Available: http://www.n4c.eu/ [Accessed: January, 2009].

[8] A. Doria, M. Uden, and D. P. Pandey, "Providing Connectivity to the Saami Nomadic Community," in *2nd International Conference on Open Collaborative Design for Sustainable Innovation*, Bangalore, India, December, 2002.

[9] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," in *IEEE Computer*, vol. 37, 2004, pp. 78-83.

[10] M. Asplund, S. Nadjm-Tehrani, and J. Sigholm, "Emerging Information Infrastructures: Cooperation in Disasters," in *Lecture Notes in Computer Science, Critical Information Infrastructure Security*, vol. 5508/2009: Springer Berlin / Heidelberg, 2009, pp. 258-270.

[11] 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.

[12] 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.

[13] Y. Shao, C. Liu, and J. Wu, "Delay-Tolerant Networks in VANETs," in *Vehicular Networks: From Theory to Practice*, S. Olariu and M. C. Weigle, Eds.: Chapman & Hall, 2009.

[14] L. Franck and F. Gil-Castineira, "Using Delay Tolerant Networks for Car2Car Communications," in *IEEE International Symposium on Industrial Electronics 2007 (ISIE 2007)*, Vigo, Spain, 4-7 June, 2007, pp. 2573-2578.

[15] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "A Layered Architecture for Vehicular Delay-Tolerant Networks," in *The Fourteenth IEEE Symposium on Computers and Communications (ISCC ’ 09)*, Sousse, Tunisia, July 5 - 8, 2009, pp. 122-127.

[16] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant

Networks," in *INFOCOM 2006 - The 25th IEEE International Conference on Computer Communications*, Barcelona, Catalunya, Spain, April 23-29, 2006, pp. 1-11.

[17] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, "The CarTel Mobile Sensor Computing System," in *The 4th ACM Conference on Embedded Networked Sensor Systems (ACM SenSys 2006)*, Boulder, Colorado, USA, October 31 - November 3, 2006, pp. 383-384.

[18] J. Morillo-Pozo, J. M. Barcelo-Ordinas, O. Trullos-Cruces, and J. Garcia-Vidal, "Applying Cooperation for Delay Tolerant Vehicular Networks," in *Fourth EuroFGI Workshop on Wireless and Mobility*, Barcelona, Spain, January 16-18, 2008.

[19] 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.

[20] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC 5050, November 2007, [Online]. Available: http://www.rfc-editor.org/rfc/rfc5050.txt.

[21] V. N. G. J. Soares, J. J. P. C. Rodrigues, F. Farahmand, and M. Denko, "Exploiting Node Localization for Performance Improvement of Vehicular Delay-Tolerant Networks," in *2010 IEEE International Conference on Communications (IEEE ICC 2010) - General Symposium on Selected Areas in Communications (ICC'10 SAS)*, Cape Town, South Africa, May 23-27, 2010.

[22] N. Banerjee, M. D. Corner, and B. N. Levine, "An Energy-Efficient Architecture for DTN Throwboxes," in *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, Anchorage, Alaska, USA, May 6-12, 2007, pp. 776-784.

[23] J. J. P. C. Rodrigues, V. N. G. J. Soares, and F. Farahmand, "Stationary Relay Nodes Deployment on Vehicular Opportunistic Networks," in *Mobile Opportunistic Networks: Architectures, Protocols and Applications*, M. K. Denko, Ed. USA: Auerbach Publications, CRC Press, May 2010.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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-dtnrg-prophet-02.

[31] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance Modeling of Epidemic Routing," *Computer Networks, Elsevier*, vol. 51, no. 10, pp. 2867-2891, July 2007.

[32] 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.

[33] V. Erramilli and M. Crovella, "Forwarding in Opportunistic Networks with Resource Constraints," in *ACM MobiCom Workshop on Challenged Networks (CHANTS 2008)*, San Francisco, California, USA, September 15, 2008, pp. 41-48.

[34] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation Forwarding," in *9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2008)*, Hong Kong, China, May 27-30, 2008, pp. 251-260.

[35] Y. Li, L. Zhao, Z. Liu, and Q. Liu, "N-Drop: Congestion Control Strategy under Epidemic Routing in DTN," in *5th International Wireless Communications and Mobile Computing Conference (IWCMC 2009)*, Leipzig, Germany, June 21-24, 2009, pp. 457-460.

[36] 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.

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

[38] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *ACM SIGCOMM 2005 - Workshop on Delay Tolerant Networking and Related Networks (WDTN-05)*, Philadelphia, PA, USA, August 22-26, 2005, pp. 252-259.