EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Performance assessment of IP over vehicular delay-tolerant networks through the VDTN@Lab testbed

João AFF Dias[1], João NG Isento[1], Bruno MC Silva[1], Vasco NGJ Soares[1,2] and Joel JPC Rodrigues[1*]

Abstract

Vehicular delay-tolerant network (VDTN) is a network architecture based on the delay-tolerant network paradigm, which was designed to provide low-cost asynchronous vehicular communications in environments with disruptions, intermittency, variable delays, and network partition. This article proposes a laboratory testbed for VDTNs, called VDTN@Lab. It aims to support research studies related with the design, emulation, performance evaluation, and diagnose of new VDTN protocols, services, and applications. It intends to demonstrate the applicability of VDTNs over multiple application environments. VDTN@Lab features an emulation capability, allowing live experiments with prototyped hardware and software embedded into robotic cards, desktop, and netbooks computers. The proposed prototype is demonstrated and evaluated with Epidemic, and Spray, and Wait routing protocols, using different combinations of scheduling and dropping policies, in scenarios with different vehicular mobility models (bus movement and random movement across roads).

Keywords: vehicular delay-tolerant networks, layered architecture, IP over VDTN, bundle layer, performance evaluation, testbed, prototype

1. Introduction

Vehicular networking has attracted growing research attention in the last years and it has shown a great potential for application to a wide range of real-world scenarios. Examples include networks to disseminate information advertisements or safety related information [1-3], networks to distribute multimedia content [4,5], and monitoring networks for data collection [6]. Vehicular networks can also be employed to provide connectivity to remote rural communities and regions [7-12], or to assist communication between rescue teams and other emergency services in catastrophe hit areas lacking a conventional communication infrastructure [13]. However, the establishment of network connectivity among vehicles and between vehicles and roadside infrastructure faces challenging connectivity issues. Most of them arise from the high mobility of vehicles, which is responsible for a highly dynamic network topology, and to short contact durations [14,15]. Limited transmission ranges, physical obstacles, and interferences lead to disruption and intermittent connectivity [16]. Furthermore, the large distances usually involved and low node densities contribute to network partition. Therefore, a contemporaneous end-to-end path from source-to-destination often does not exist.

The related literature offers several available approaches to solve the problem of providing communication in vehicular networks. Vehicle ad hoc networks (VANETs) [17,18] were proposed as a special type of ad hoc networks for inter-vehicular communications. However, conventional routing schemes for VANETs assume end-to-end connectivity. Thus, they are not able to deal with network disconnection, partitions, or long-time delays [19-21]. These limitations were overcome by applying the store-carry-and-forward paradigm of the delay-tolerant network (DTN) architecture [22], creating the concept of "DTN-enabled VANETs" [6,23]. In a DTN-based network, data delivery is increased by allowing nodes to store data when there is no contact with other nodes, to carry it until meeting other nodes, and forwarding it based on a routing scheme.

* Correspondence: joeljr@ieee.org
[1]Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
Full list of author information is available at the end of the article

More recently, vehicular delay-tolerant networks (VDTNs) were proposed as an alternative network architecture for sparse vehicular networks [24]. VDTN architecture also adopts a store-carry-and-forward paradigm from DTNs. Nonetheless, it distinguishes itself from the DTN architecture by positioning the bundle layer between the network and data link layers, introducing a clear separation between control and data planes using out-of-band signaling.

Designing protocols for VDTNs poses a number of technical challenges due to the nature of vehicular environments and to a variety of factors including node heterogeneity, node interactions, node cooperation, and limited network resources. Usually, researchers propose and evaluate new services and protocols using simulation and theoretical analysis techniques. However, these techniques typically abstract many details of the real-world, and these simplifications tend to impair performance in real-world environments. Thus, although simulation and theoretical analysis are helpful in a preliminary evaluation of new protocols and algorithms, it is essential to implement, test, and evaluate them in a testbed (prototype) network for performance assessing under real-world conditions. In this sense, this article focuses on the performance evaluation of IP over VDTNs through a prototype, presenting the design, and construction of a laboratory testbed for VDTNs, called VDTN@Lab. The motivation for this prototype is to provide a framework for demonstration and validation of the VDTN architecture, allowing the development, performance evaluation, and validation of new services and protocols, as well as VDTN applications. The proposed testbed features (i) an emulation capability allowing live experiments with prototyped hardware and software embedded into robotic cars, computers/laptops, and netbooks; (ii) an integrated environment capable to emulate VDTN protocol stacks, services, and applications; and (iii) operation under emulated realistic operating conditions.

The rest of the article is organized as follows. Section 2 presents the VDTN architecture while Section 3 describes available testbeds used on research works related to vehicular networks. The design of the proposed testbed for VDTN networks is presented in Section 4. Section 5 focuses on the performance evaluation and validation of the proposed testbed. Section 6 concludes the article and pinpoints directions for future studies.

2. Vehicular delay-tolerant networks

VDTNs are complex systems where a variety of mobile and fixed nodes can freely interact with each other. Terminal nodes represent the access points to the VDTN network and may be both fixed and mobile (e.g., vehicles that also act as end points of a communication). Mobile nodes are opportunistically exploited to collect and disseminate data bundles through the VDTN network. They move along roads and carry data that must be delivered to the terminal nodes. Stationary relay nodes are fixed devices with store-and-forward capabilities that are located at road intersections. Mobile nodes interact with them to deposit and pickup data. Relay nodes increase contact opportunities in scenarios with low node density. Hence, they contribute to increase the data bundles delivery ratio, and decrease their delivery delay [25].

In order to support communication in sparse and disconnected vehicular network scenarios, VDTN presents a network architecture based on the following based design principles [24]: (i) Internet protocol (IP) over VDTN approach; (ii) end-to-end, asynchronous, and variable-length bundle oriented communication; and (iii) separation between control and data planes using out-of-band signaling. Different to DTN architecture proposal that introduces a bundle layer between the transport and application layer to allow the interconnection of highly heterogeneous networks [26], VDTN architecture places the bundle layer over the data link layer introducing an IP over VDTN approach [24]. The protocol data unit at the VDTN bundle layer is called a *bundle*, wish aggregates several IP packets with several common properties, like the same destination node.

VDTN uses the principle of store-carry-and-forward routing proposed for DTNs [22]. This paradigm solves the problems caused by intermittency, disconnection and long delays, and can be described as follows. A network node stores a bundle using some form of persistent storage while waiting for a future opportunistic connection. When a communication opportunity arises, the bundle is forwarded to an intermediate node, according to a hop-by-hop routing scheme. This process is repeated and the bundle will be relayed hop-by-hop until eventually reaches its destination.

VDTN architecture identifies two logical planes (using out-of-band signaling), i.e., the control plane and the data plane [24]. These planes are logically divided into two layers, the bundle signaling control (BSC) layer and the bundle aggregation and de-aggregation (BAD) layer. BSC is responsible for executing the control plane functions such as signaling messages exchange, resources reservation (at the data plane), and routing. The data plane functions that deal with data bundles are executed in BAD. These functions include data bundles aggregation/de-aggregation, queuing and scheduling, and traffic classification.

Control plane uses a low-power, low bandwidth, and long-range link, and it is always active to allow node discovery. The data plane uses high-power, high bandwidth, and short-range link, and it is only active during the estimated contact duration time, and if there are data bundles to be exchanged between the network nodes according to the routing protocol [24,27]. Otherwise, the data plane connection is not activated. This approach is considered important because it not only ensures the optimization of the available data plane resources (e.g., storage and bandwidth), but also allows to save power, which is very important for energy-constrained network nodes such as stationary relay nodes [24,28]. These nodes are usually power-limited since they may run on solar panels or batteries. Figure 1 illustrates this concept. At the time $t + t_0$, a mobile node and a relay node detect each other and start exchanging signaling messages through the control plane connection. Both nodes use routing information to determine which bundles should be forwarded. Then, the data plane connection is configured and activated on both nodes at the time $t + t_1$. The bundles are exchanged until the time $t + t_2$. The data plane connection is deactivated after that time instant, since the nodes are no longer in the data plane link range of each other.

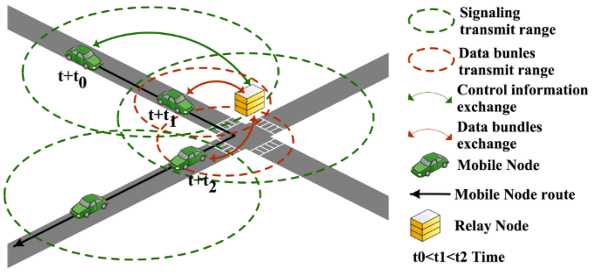


Figure 1 Illustration of the VDTN control information and data bundles exchange.

3. Related study

Over the last years, several testbeds have been developed to support the development and evaluation of architectures and protocols for vehicular networks. The most part of them are developed for a particular topic of research, ranging from physical layer aspects to applications demonstration and validation. This section surveys the most relevant related literature, describing relevant available vehicular network testbeds and highlights important aspects considered on the design and construction of the proposed VDTN@Lab.

VanLAN [29,30] is a testbed composed of 11 basestations and 2 vehicles, which was developed to investigate the characteristics of WiFi-based connectivity in urban settings. It was used to evaluate how the raw connectivity varies as the vehicle moves and whether it is stable across traversals of the same location.

In [31], the authors were interested in assessing the possibility of exploring open Wi-Fi networks in urban and suburban areas to allow data uploads from cars to Internet servers. A measurement study was conducted over a vehicular testbed. Nine distinct cars collected data about open APs deployed in and around the Boston metropolitan area, during a period of 290 h of driving.

A large-scale VANET testbed running over 4000 taxis in Shanghai is presented in [32]. The information about GPS data collected from the taxis was used to construct a realistic, large-scale mobility model, which was named Shanghai urban vehicular network.

Cabernet [33] is a system developed for improving open WiFi data delivery to moving vehicles based on two components: QuickWiFi for improving connection establishment time, and Cabernet Transport Protocol for improving throughput over opportunistic WiFi networks. This system was evaluated under real-world conditions on a 10-taxi testbed in the Boston area.

In [34], a real vehicular ad hoc testbed composed of two vehicles and an access point was used to test the feasibility of a peer-to-peer file sharing application named CarTorrent. Another example of a VANET testbed composed of two cars is presented in [35]. The main objective of this testbed was to conduct experimental measurements of vehicle-to-vehicle communication, in order to study the critical factors that affect the quality of a video transmission over a VANET in different scenarios.

Demonstrator for Intelligent Vehicular Environments [36] is a modular, flexible (i.e., easily adapted and updated), reconfigurable testbed demonstrator that allows studying network-layer aspects of vehicular communications (e.g., intra-vehicular, inter-vehicular, and vehicle to infrastructure communication), and advanced services for vehicular users.

UMass DieselNet [37,38], CarTel [15,39], and Drive-Thru [40,41]are examples of real-world testbeds that were developed for supporting research and development of delay-tolerant networking techniques in vehicular communications.

The UMass DieselNet [37,38] is a bus-based DTN testbed running on 40 buses operated by the UMass Amherst (USA). This testbed has been used to study routing protocols for DTN networks, mobility models of bus-to-bus connectivity, and to investigate the use of throwboxes (i.e., stationary relay nodes) to increase contact opportunities and throughput.

Fleet testbed [42] is composed of 27 cars, each of them running a CarTel [15,39] embedded platform which interfaces with a variety of sensors in a car, processes the collected data, and delivers it to an Internet server, providing services to users. CarTel uses wireless networks opportunistically, and shields applications from the underlying details and network disruptions.

The Drive-thru Internet project [40,41] investigated the problematic of providing Internet access to mobile users in moving vehicles (cars, trains, etc.), based on WLAN hot spots deployed along the roads, in rest areas, or at gas stations. The project proposed an architecture that allows applications to deal with intermittent connectivity, and evaluated the communication characteristics when UDP or TCP standard transport protocols were used.

Deploying and operating a real-world testbed to increase knowledge about vehicular communications and to evaluate the behavior/performance of protocols, services, and applications under a large-scale network supposes a great effort and has a high associated cost. Moreover, such a testbed has limited flexibility and its use is limited to those who have access to it. These insights motivated the proposal, design, and creation of a versatile laboratory testbed for VDTN networks, the VDTN@Lab.

This testbed gathered contributions and insights from the above-described projects. The communication between nodes is based on Bluetooth and IEEE 802.11 technologies, for example, considered in [15,36]. One of the developed vehicular mobility models available in VDTN@Lab was inspired by DieselNet [37,38]. The proposal and construction of the different node types also collects contributions from all the above-described projects. The proposed testbed will be used for performance evaluation and analysis of disconnected vehicular networks. It will implement the VDTN architecture and demonstrate the applicability of VDTNs in a variety of application scenarios.

4. Overview of the VDTN testbed design

This section describes VDTN@Lab, a testbed created for demonstrating the VDTN architecture and its protocols, services, and applications in a laboratory environment. VDTN@Lab aims to provide a framework for the validation of the VDTN architecture. The next two sections present the VDTN@Lab requirements analysis and discuss hardware and software technologies used to create the prototype.

4.1. Testbed requirements

Unified modeling language (UML) [43] was used for the requirement analysis and high-level design of the VDTN testbed. Due to space constraints, it is not possible to present all UML diagrams in this article. Figures 2 and 3 present two diagrams that were chosen to illustrate some important concepts of the VDTN network architecture. Figure 2 shows a use case diagram for a VDTN terminal node. All network nodes execute

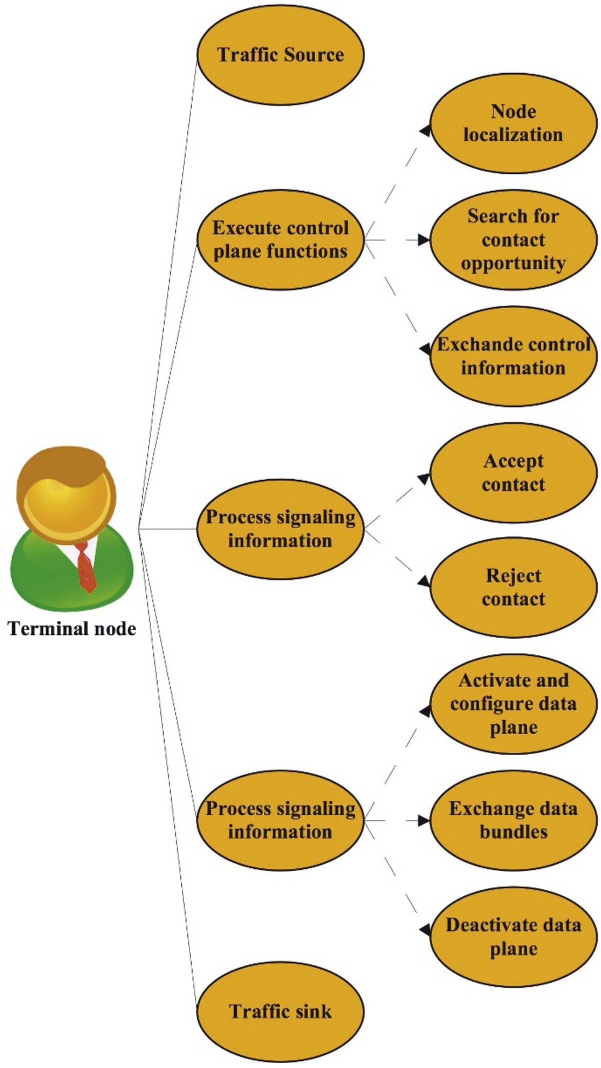

Traffic Source
Node localization
Execute control plane functions
Search for contact opportunity
Exchande control information
Accept contact
Process signaling information
Reject contact
Terminal node
Activate and configure data plane
Process signaling information
Exchange data bundles
Deactivate data plane
Traffic sink

Figure 2 UML use cases diagram of a terminal node.

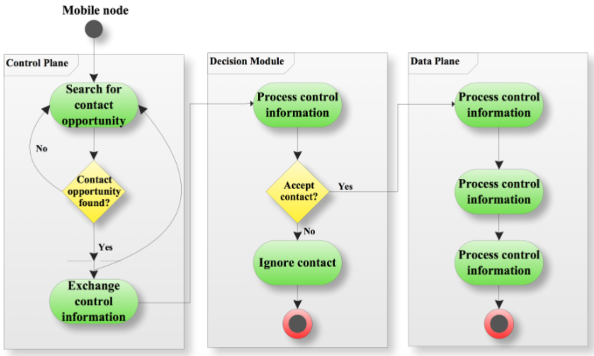

Mobile node
Control Plane
Search for contact opportunity
No
Contact opportunity found?
Yes
Exchange control information
Decision Module
Process control information
Accept contact?
Yes
No
Ignore contact
Data Plane
Process control information
Process control information
Process control information

Figure 3 UML activity diagram of a mobile node, describing the control plane and data plane interaction, coordinated by a decision module.

the same actions in the control plane and in the data plane. However, terminal nodes perform additional functions, since they represent the traffic sources and the traffic sinks in a VDTN network. Network nodes use their control plane link connection to detect contact opportunities. When two nodes are within the range of each other, both nodes exchange the control information. Then, this information is processed and it is used for deciding if the contact should be accepted or rejected. A contact is accepted if at least one of the nodes stores a bundle that should forwarded to the other node (according to a routing protocol). Additional criteria can also be employed in this decision process, like buffer congestion or energy constraints, which are announced in signaling (i.e., control) messages. If the contact is accepted, both nodes reserve resources at the data plane. Hence, they activate and configure their data plane link connection, where operations related to data bundles are performed.

Figure 3 illustrates an activity diagram of a mobile node, which represents a workflow of stepwise activities and actions describing control plane and data plane interaction, coordinated by the decision module. This activity diagram is equal for all network nodes, and represents the concept of control plane and data plane separation with out-of-band signaling. Each network node autonomously manages its control plane and data plane link connections. Nodes are always searching for new contact opportunities, using their control plane link connection (low-power, low bandwidth, long-range), which is always active. A decision module is responsible for processing the control information exchanged at a new contact opportunity to decide whether to accept the contact, and for determining the amount of time that lasts the contact [27]. Then, the data plane link connection (high-power, high bandwidth, short-range) is activated, and remains in this state only during the estimated period of time that lasts the contact. Nodes use their data plane link to exchange data bundles. The BSC layer executes the control plane functions, such as signaling messages exchange, resources reservation (at the data plane), and routing. The BAD layer executes the data plane functions that deal with data bundles. These functions include storage management, queuing and scheduling, and traffic classification, among others.

4.2. Testbed specifications and design

The presented testbed was created for a laboratory environment. Its design is modular with well-defined interfaces between the hardware and software components. This enables updating different hardware/software components with minimal impacts on the others. Another important aspect is that interested researchers can easily reproduce this testbed, as the needed hardware to perform it is not expensive, it is easily available, and easy to set up. Furthermore, the software is hardware device independent as much as possible and it has been developed in such a way as to adapt itself to a future deployment in a real-world testbed with minimum efforts.

The testbed considers three types of network nodes previously described. Desktop and laptop computers are used to emulate terminal nodes and relay nodes. Mobile nodes (e.g., vehicles) are emulated through LEGO MINDSTORMS NXT robotic cars [44] and a netbook computer. A mobile node is shown in Figure 4. As may be seen, each robotic car carries a netbook for having processing, networking, and storage capabilities. LEGO NXT robots are programmed with several mobility models (e.g., random movement across roads or bus movement), allowing performance evaluation studies under different movement patterns. All network nodes support Bluetooth [45] and IEEE 802.11 b/g [46] technologies. These technologies are used to support the VDTN out-of-band signaling with the separation between control and data planes. Bluetooth connection is used to exchange signaling information (control plane), whereas IEEE 802.11 b/g is used for data bundles exchange (data plane). Figure 4 shows some interactions between mobile nodes, terminal nodes, and relay nodes.

Several software modules were created in C# programming language and deployed in the network nodes. They were developed using the. NET Framework for running in the desktops, laptops, and netbooks with Windows 7 operating system. The software modules implement the above-described VDTN architecture principles, as well as several routing protocols (e.g., First Contact [47], Epidemic [48], Source and Binary Spray and Wait [49]), and scheduling and dropping policies (e.g., FIFO, LIFO, random, lifetime-based, replicated-copies). They also

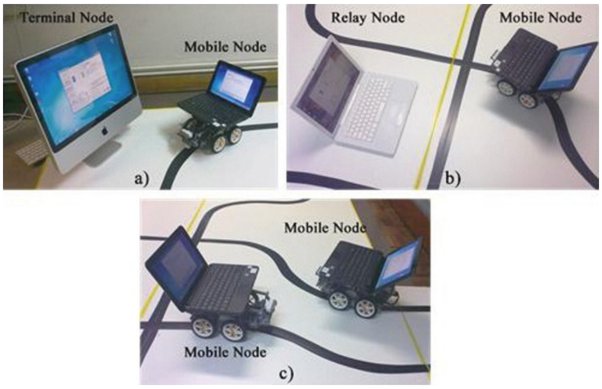


Figure 4 Illustration of the interactions among VDTN nodes: **(a)** a mobile node and a terminal node establishing contact; **(b)** a mobile node and a relay node establishing contact; and **(c)** two mobile nodes establishing contact.

provide functionalities to emulate network resource constraints (e.g., energy, storage, bandwidth, range), to emulate different operation conditions, and to emulate network applications with different traffic characteristics and different "quality of service" requirements. In addition, the software modules provide management tools and advanced statistics reports. For example, it is possible to collect statistical data about the delivery ratio and average delay, the bundles drop ratio, the number of contacts per hour, the average contact time, and the historic of nodes that have stored-carried-and-forwarded each delivered bundle. Figure 5 presents an illustration of the software modules for terminal and mobile nodes, respectively.

The class diagram shown in Figure 6 details the main classes of the software developed for the testbed. This comprehensive diagram provides an overview of the virtualization of the VDTN network model. Class attributes and methods were omitted to improve the diagram readability. The main class application, called *VDTNHost*, interacts with the classes responsible for data exchange, which is the *ControlPlaneLink* that is used for signaling messages exchange, and the *DataPlaneLink* that is used for data bundles exchange. *VDTNHost* class also interacts with the classes that implement control plane (*BSC*) and data plane (*BAD*) separation. As expected, both *Signaling* and *Routing* classes are connected to *BSC* class. The *Signaling* class is responsible for generating and processing signaling messages. The *Routing* class generates and processes routing protocols information, and selects which bundles should be exchanged, based on the routing protocol under use. The *BAD* class interacts with the *BufferManagement* class that is responsible for applying a drop policy when buffer congestion occurs, and with the *Scheduling* class that applies a scheduling policy to determine the order by which bundles should be sent at a contact opportunity. The *BAD* class also is connected with the *Classification* class that implements the functions for traffic differentiation [50], and with the *De/Aggregation* class that is responsible for forming data bundles by aggregating IP packets, according to a bundle assembly algorithm.

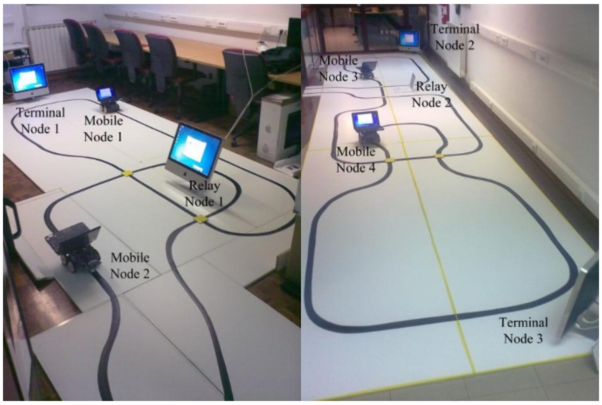


Figure 6 UML class diagram illustrating the main classes and important relationships in VDTN architecture.

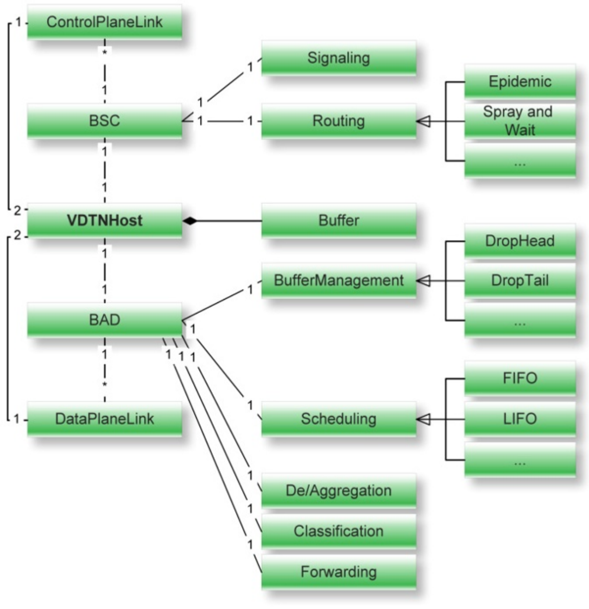


Figure 5 Illustration of the created software modules for:(a) terminal nodes and **(b)** mobile nodes emulation.

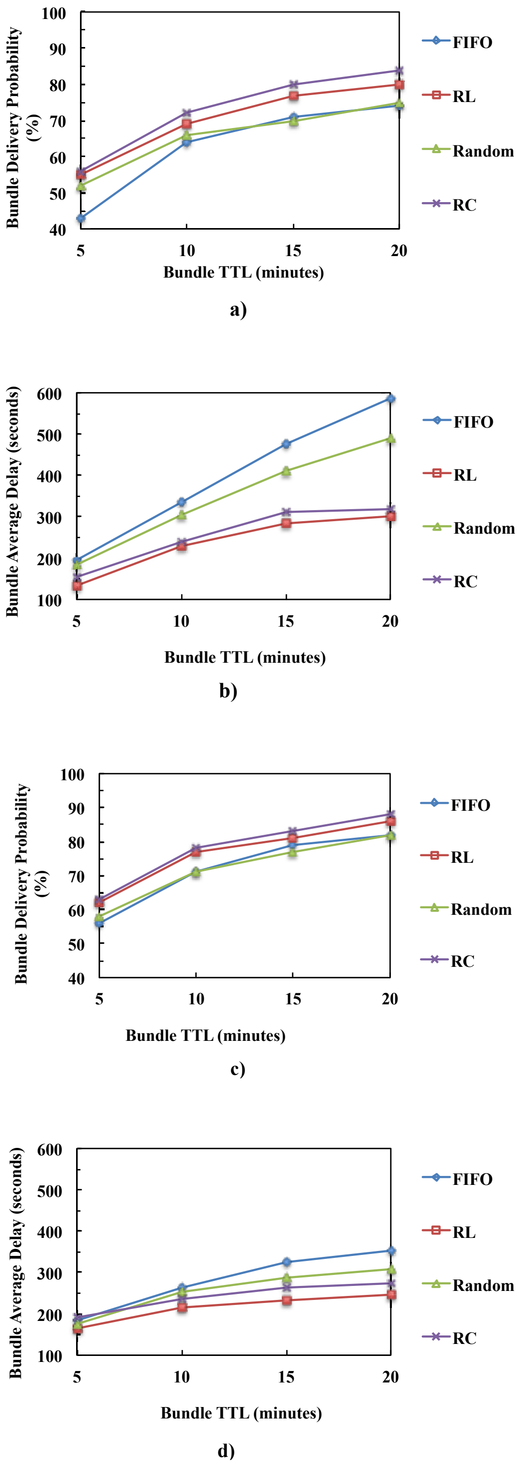
5. Performance evaluation and testbed demonstration

This section focuses on the testbed demonstration and performance evaluation of IP over VDTNs and considers two sections. The first section presents the laboratory testbed network scenarios considered in this study. The results analysis is discussed in the last subsection.

5.1. Network scenarios

Two scenarios were set up to demonstrate the VDTN testbed. Both scenarios consider three fixed terminal nodes, four mobile nodes, and two relay nodes. Terminal nodes are placed at different points (edges) of the laboratory. In the first scenario, the mobile nodes follow predefined paths to emulate bus routes. In the second scenario, these mobile nodes have a random movement along the roads. In both scenarios, mobile nodes move with different velocities. Parallel with a study based on a testbed composed by real vehicles [51], and assuming a scale of 1:50 (1 m in the laboratory testbed represents 50 m in a real scenario), *Mobile node 1* moves at 48 km/h, *mobile node 2* at 40 km/h, *mobile 3* at 36 km/h, and *mobile node 4* at 24 km/h. Relay nodes are placed on roads intersections. Figure 7 shows photos of the VDTN laboratory testbed and some of the above-presented nodes.

At the very beginning, all nodes have their buffers empty and are ready to receive and transfer bundles. Each type of network node has different buffer sizes. Terminal nodes have a buffer size with 50 Mbytes, relay nodes have 75 Mbytes, and finally mobile nodes have 25 Mbytes for storage space. The nodes' buffer space is confined to these values in order to show more clearly the impact of different combinations of dropping and scheduling policies.



a)

b)

c)

d)

Figure 7 Photos of the VDTN@Lab testbed.

Different combinations of scheduling and dropping policies are enforced at the network nodes, namely *first-in first-out* (FIFO), *Remaining Lifetime* (RL), *Random*, and *Replicated Copies* (RC). In a *FIFO* combination, bundles are scheduled by the order they were received. When buffer overflow occurs, bundles stored for the longest period of time are dropped first. Using a *Remaining lifetime* combination, bundles are scheduled according to their remaining time-to-live (TTL). Bundles with longer remaining TTL are scheduled to be sent first. To perform the dropping operation, this combination drops bundles with smaller remaining TTL first. In a *Random* combination, bundles are scheduled and dropped by a random order. The *Replicated copies*combination assumes that each node keeps track of the number of times each bundle has been replicated. Bundles that have been less replicated are scheduled first, while bundles that have been more replicated are dropped first.

Bundles have random source and destination terminal nodes and are generated at each 20 s. Data bundles' size is uniformly distributed in the range of [250 KB, 2 MB] (bytes). When a bundle reaches its final destination it is marked as delivered. For testbed experiments, the bundles' TTL changes between 5, 10, 15, and 20 min. It is assumed a fully cooperative opportunistic environment and each experiment has a duration of 1 h.

Epidemic and Binary Spray and Wait are used as underlying routing schemes. Epidemic is a flooding-based routing protocol where nodes exchange the bundles they do not have in common. In an environment with infinite buffer space and bandwidth, it provides the optimal solution. The Binary Spray and Wait protocol creates a number of copies (N) to be transmitted ("sprayed") per bundle. Any node A that has more than one bundle copy 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 bundles. When a node only has one copy left of a bundle, it only forwards it to the final destination. For this routing protocol it is assumed $N = 3$.

The 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 generated), the bundle average delivery delay (measured as the time between bundle creation and delivery), and the number of dropped bundles. The results presented in the next section include the average of 30 testbed experiments per parameter setting.

5.2. Performance analysis for a scenario with a mobility model based on bus movement model

The performance analysis starts with the observed bundle delivery probability, when a mobility model based on bus movement is considered. Different combinations of scheduling and dropping policies are enforced on Epidemic and Binary Spray and Wait routing protocols. As may be seen in Figure 8a, for Epidemic routing RC combination presents the best results. It improves the delivery probability in 13, 12, 9, and 10% when compared with FIFO (for each value of bundles' TTL); 4, 6, 10, and 9% when compared with Random combination; and 1, 3, 3, and 4% when compared with RL combination. This is caused by storage and bandwidth constraints, that limit the number of bundles being carried, and the number of bundles exchanged at a contact opportunity. The RC combination gives preferential treatment to bundles that have been less replicated, balancing the number of copies of each bundle in the network and improving the bundle delivery delay.

Figure 8b shows the bundle average delivery delay as function of bundle TTL for the same network conditions and routing protocol. As may be seen, the RL contributes to decrease the bundle average delivery delay. When compared to FIFO, bundles arrive to their final destination approximately 62, 106, 192, and 284 s sooner in average. When compared to Random, bundles arrive to their final destination approximately 13, 76, 128, and 189 s sooner. When compared to RC bundles arrived to their final destination approximately 21, 13, 28, and 17 s sooner. This happens because RL combination of scheduling and dropping policies forwards first bundles with longer remaining TTLs that will have more time left to reach their destination, and drop first bundles with smaller remaining TTLs.

Figure 8c shows the observed bundle delivery probability when Binary Spray and Wait routing protocol is considered. As may be seen, the RC also presents the best results. It presents gains of 7, 7, 4, and 6% when compared with FIFO, 5, 7, 6, and 6% when compared with Random and 1, 1, 2, and 2% when compared with RL. This happens because of the same reasons stated above for Epidemic routing protocol. With Spray and Wait the bundle delivery probability is higher than Epidemic because Spray and Wait limits the number of copies of a bundle. This will cause less bandwidth utilization and less congestion at the network nodes buffers.

Figure 8d shows the bundle average delivery delay as function of bundle TTL for Binary Spray and Wait. As

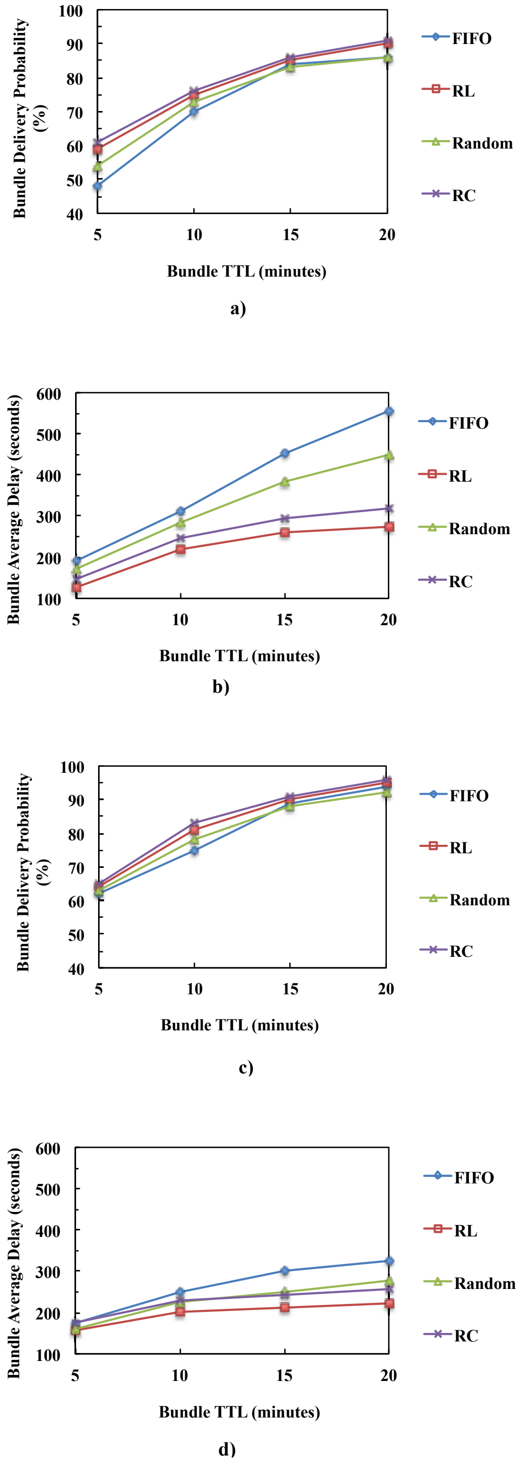


Figure 8 Bundle delivery probability for (a) Epidemic and **(c)** Spray and Wait; and bundle average delay for **(b)** Epidemic and **(d)** Spray and Wait, as function of bundle TTL using different combinations of scheduling and dropping policies, when a mobility model based on bus movement is considered.

may be seen, RL contributes to decrease the bundle average delivery delay. When compared to FIFO, bundles arrive to their final destination approximately 18, 48, 93, and 105 s sooner in average, 11, 36, 52, and 60 s sooner when compared to Random and 25, 21, 31, and 25 s sooner when compared to RC. It is interesting to observe that, comparing with Epidemic, Spray and Wait presents a significant decrease of the bundle average delivery delay for all combinations of scheduling and dropping policies.

5.3. Performance analysis for a scenario with a mobility model based on random movement along roads

The study focuses on the performance evaluation of VDTNs when a mobility model based on random vehicular movement along roads is assumed. As may be seen in Figure 9a, for Epidemic routing, RC presents the best results for each TTL value. It improves the bundle delivery probability about 13, 6, 2, and 5% when compared with FIFO, 7, 3, 3, and 5% when compared with Random, and 2, 1, 1, and 1% when compared with RL. These results were expected for the same reasons stated in the previous scenario. It is important no notice that bundle delivery probability is higher in this scenario because one mobile node can receive a bundle from a source terminal node and deliver it directly to the destination node. This also causes bundles to be delivered in fewer hops.

Figure 9b shows the bundle average delay as function of bundles TTL also for Epidemic routing protocol. As may be seen, RL contributes to decrease the bundle average delay. When compared to FIFO, bundles arrive to their final destination approximately 63, 92, 193, and 279 s sooner (in average). When compared to the Random approach, bundles arrive to their final destination approximately 45, 66, 124, and 175 s sooner. When compared to the RC combination, bundles arrived to their final destination approximately 19, 21, 37, and 44 s sooner. This is happen because RL combination forwards first bundles with longer remaining TTL that have a bigger probability of reaching their final destination, and drops first bundles with smaller remaining TTL. The performance of Epidemic routing protocol with a mobility model based on bus movement presents lower delay due to the same reason described on above-presented results.

Figure 9c shows the observed bundle delivery probability when Binary Spray and Wait routing protocol is considered. As may be seen, the RC also presents the best results for this routing protocol. It presents gains of 3, 8, 2, and 2% when compared with FIFO, 2, 5, 3, and 4% when compared with Random, and 1, 2, 1, and 1% when compared with RL. This happens because of the same reasons above stated for Spray and Wait protocol with a mobility model based on bus movement. In this case, the bundle delivery probability is higher because one node can receive a bundle from source terminal node and deliver it directly to destination. This causes bundles to be delivered in fewer hops.

Figure 9d shows the bundle average delay as function of bundle TTL for the same routing protocol and network conditions. As may be seen, the RL combination contributes to decrease the bundle average delay. When compared to FIFO, bundles arrive to their final destination approximately 18, 49, 88, and 103 s sooner (in average). When compared to Random combination, bundles arrive to their final destination approximately 3, 25, 38, and 55 s sooner. When compared to RC combination, bundles arrived to their final destination approximately 19, 28, 31, and 36 s sooner. This happen because RL combination forwards first bundles with longer remaining TTL that have a bigger probability to reach the final destination, and drop first bundles with smaller remaining TTL. When compared with Spray and Wait routing with a mobility model based on bus movement, this presents lower delay due to same reason that described in above results. These results confirm previous study with results obtained by simulation and presented in [52].

6. Conclusions and future work

Over the years, several approaches have been proposed to address the problems related with achieving data communication in vehicular networks. These networks are characterized by a highly dynamic network topology, short contact durations, connectivity disruption, intermittent connectivity, and even network partition. VDTNs appeared as an innovative solution for handling these issues. VDTN architecture integrates the concepts of end-to-end, asynchronous, and variable-length bundle-oriented communication, Internet protocol over VDTN, and control and data plane separation performing out-of-band signaling.

Although theoretical analysis (such as analytical models) and simulation are helpful during the development, performance evaluation, and validation of new VDTN protocols, services, and applications, it is essential to deploy, experiment, and evaluate their performance in a testbed, in order to assess them under real-world conditions. This motivated the design and construction of the laboratory testbed for VDTN networks presented in this study. VDTN@Lab testbed is an important contribution to the literature, since, to the best of our knowledge, a VDTN testbed has not been presented before.

A performance assessment study conducted through VDTN@Lab analyzed the performance of a combination of scheduling and dropping policies enforced on two routing schemes. Two network scenarios with different vehicular mobility models were considered. The testbed

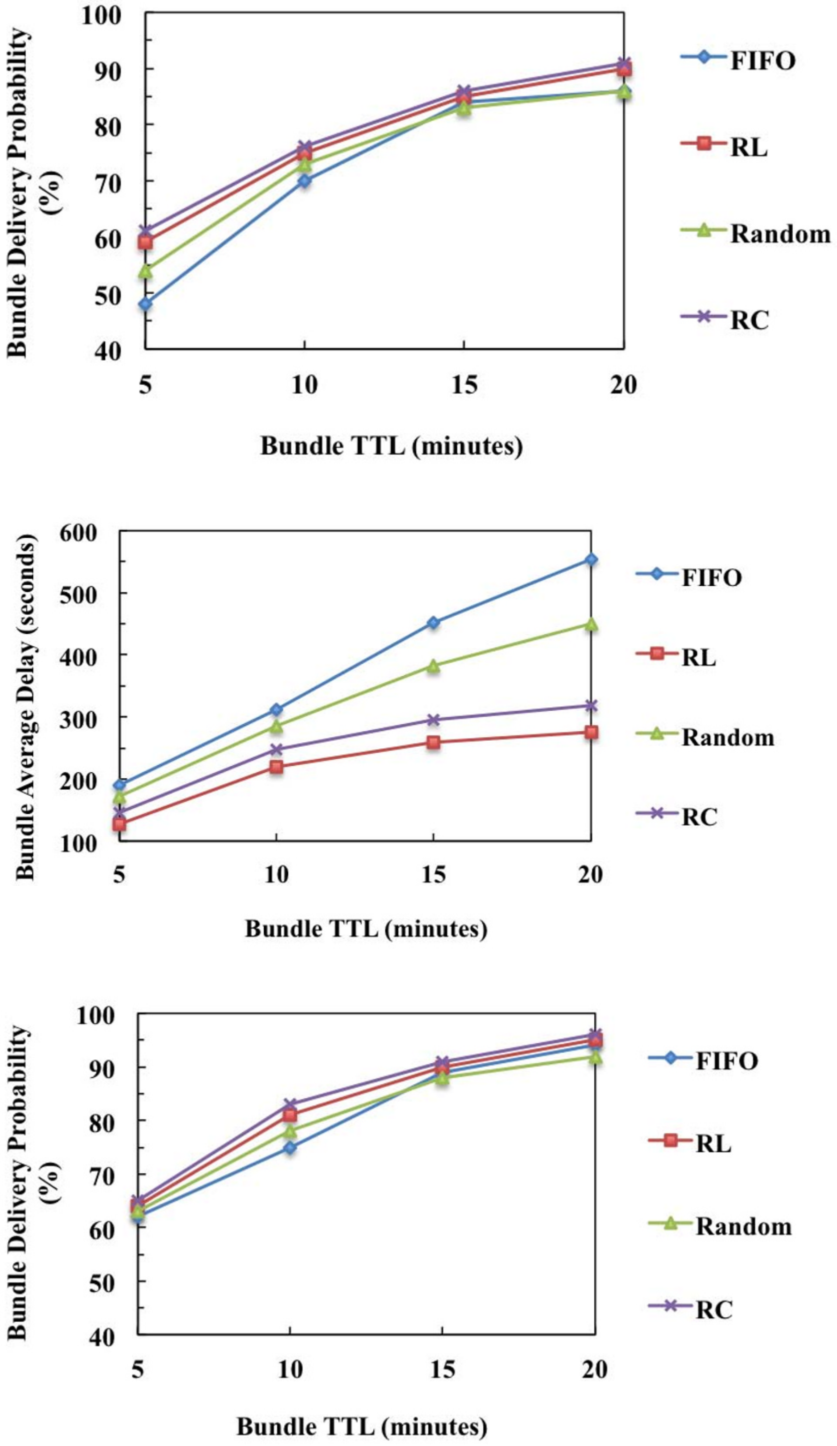


Figure 9 Bundle delivery probability for (a) Epidemic and **(c)** Spray and Wait; and bundle average delay for **(b)** Epidemic and **(d)** Spray and Wait, as function of bundle TTL using different combinations of scheduling and dropping policies, when a mobility model based on random movement along roads is considered.

results reveal, in both scenarios, a good performance obtained by a combination of a scheduling and dropping policies that gives preferential treatment to less replicated bundles. It was shown that Spray and Wait outperforms the Epidemic scheme in both scenarios. These results corroborate and validate previous conclusions from studies performed through simulation that were presented in [52].

There are many possible directions for future study over the VDTN@Lab testbed. One such direction is to add support for more routing protocols and implement fragmentation/aggregation mechanisms.

Acknowledgements

This study was partially supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, and by National Funding from the FCT - *Fundação para a Ciência e a Tecnologia* through the PEst-OE/EEI/LA0008/2011.

Author details

[1]Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
[2]Superior School of Technology, Polytechnic Institute of Castelo Branco, Portugal

Competing interests

The authors declare that they have no competing interests.

Received: 2 July 2011 Accepted: 13 January 2012
Published: 13 January 2012

References

1. R Tatchikou, S Biswas, F Dion, Cooperative vehicle collision avoidance using inter-vehicle packet forwarding, in *IEEE Global Telecommunications Conference (IEEE GLOBECOM 2005)*, St. Louis, MO, USA, pp. 2762–2766 (28 November – 2 December 2005)
2. J-S Park, U Lee, SY Oh, M Gerla, DS Lun, Emergency related video streaming in VANET using network coding, in *The Third ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2006)*, Los Angeles, CA, USA, pp. 102–103 (September 29 2006)
3. A Buchenscheit, F Schaub, F Kargl, M Weber, A VANET-based emergency vehicle warning system, in *First IEEE Vehicular Networking Conference (IEEE VNC 2009)*, Tokyo, Japan, pp. 28–30 (October 2009)
4. M Fiore, JM Barcelo-Ordinas, Cooperative download in urban vehicular networks, in *Sixth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2009)*, University of Macau, Macau SAR, P.R. China, pp. 20–29 (12–15 October 2009)
5. A Nandan, S Das, G Pau, M Gerla, MY Sanadidi, Co-operative downloading in vehicular Ad-hoc wireless networks, in *Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05)*, St. Moritz, Switzerland, pp. 32–41 (19–21 January 2005)
6. L Franck, F Gil-Castineira, Using delay tolerant networks for Car2Car communications, in *IEEE International Symposium on Industrial Electronics 2007 (ISIE 2007)*, Vigo Spain, pp. 2573–2578 (4–7 June 2007)
7. N4C and eINCLUSION, Networking for communications challenged communities: architecture, test beds and innovative alliances. [Online]. Available: http://www.n4c.eu/. Accessed July 2011
8. Wizzy Digital Courier, Wizzy digital courier - leveraging locality. [Online]. Available: http://www.wizzy.org.za/. Accessed: July 2011
9. A Pentland, R Fletcher, A Hasson, DakNet: rethinking connectivity in developing nations. IEEE Comput. **37**, 78–83 (2004)
10. A Doria, M Uden, DP Pandey, Providing connectivity to the saami nomadic community, in *2nd International Conference on Open Collaborative Design for Sustainable Innovation*, (Bangalore, India, 2002)
11. A Seth, D Kroeker, M Zaharia, S Guo, S Keshav, Low-cost communication for rural internet kiosks using mechanical backhaul, in *12th ACM International Conference on Mobile Computing and Networking (MobiCom 2006)*, Los Angeles, CA, USA, pp. 334–345 (24–29 September 2006)
12. F Farahmand, AN Patel, JP Jue, VG Soares, JJ Rodrigues, Vehicular wireless burst switching network: enhancing rural connectivity, in *The 3rd IEEE Workshop on Automotive Networking and Applications (Autonet 2008), Co-located with IEEE GLOBECOM 2008*, New Orleans, LA, USA, pp. 1–7 (4 December 2008)
13. M Asplund, S Nadjm-Tehrani, J Sigholm, Emerging information infrastructures: cooperation in disasters, in *Lecture Notes in Computer Science, Critical Information Infrastructure Security*, vol. 5508. Springer, Berlin, pp. 258–270 (2009)
14. J Burgess, B Gallagher, D Jensen, B Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in *25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, Barcelona, Catalunya, Spain, pp. 1–11 (23–29 April 2006)
15. V Bychkovsky, K Chen, M Goraczko, H Hu, B Hull, A Miu, E Shih, Y Zhang, H Balakrishnan, S Madden, The CarTel mobile sensor computing system, in *The 4th ACM Conference on Embedded Networked Sensor Systems (ACM SenSys 2006)*, Boulder, Colorado, USA, pp. 383–384 (31 October–3 November 2006)
16. J Morillo-Pozo, JM Barcelo-Ordinas, O Trullos-Cruces, J Garcia-Vidal, Applying cooperation for delay tolerant vehicular networks, in *Fourth EuroFGI Workshop on Wireless and Mobility*, Barcelona, Spain, pp. 12–13 (16–18 January 2008)
17. S Yousefi, MS Mousavi, M Fathy, Vehicular ad hoc networks (VANETs): challenges and perspectives. in *6th International Conference on ITS Telecommunications (ITST 2006)* 761–766 (21–23 June 2006)
18. H Füßler, M Torrent-Moreno, M Transier, M Festag, H Hartenstein, Thoughts on a protocol architecture for vehicular Ad-hoc networks, in *2nd International Workshop on Intelligent Transportation (WIT 2005)*, Hamburg, Germany, pp. 41–45 (15–16 March 2005)
19. Z Zhang, Routing in intermittently connected mobilead hoc networks and delay tolerant networks: overview and challenges. IEEE Commun Surv Tutor. **8**(1), 24–37 (2006)
20. M Abuelela, S Olariu, Traffic-adaptive packet relaying in VANET, in *The Fourth ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2007), in conjunction with ACM MobiCom 2007*, Montréal, QC, Canada, pp. 77–78 (10 September 2007)
21. F Li, Y Wang, Routing in vehicular ad hoc networks: a survey. IEEE Veh Technol Mag. **2**(2), 12–22 (2007)
22. V Cerf, S Burleigh, A Hooke, L Torgerson, R Durst, K Scott, K Fall, H Weiss, Delay-tolerant networking architecture. RFC 4838 (2007). [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt
23. P-C Cheng, KC Lee, M Gerla, J Härri, GeoDTN+Nav: geographic DTN routing with navigator prediction for urban vehicular environments. Mob Netw Appl. **15**(1), 61–82 (2010). doi:10.1007/s11036-009-0181-6
24. VNGJ Soares, F Farahmand, JJPC Rodrigues, A layered architecture for vehicular delay-tolerant networks, in *The Fourteenth IEEE Symposium on Computers and Communications (ISCC'09)*, Sousse, Tunisia, pp. 122–127 (5–8 July 2009)
25. JJPC Rodrigues, VNGJ Soares, F Farahmand, Stationary relay nodes deployment on vehicular opportunistic networks, in *Mobile Opportunistic Networks: Architectures, Protocols and Applications*, ed. by Denko MK (Auerbach Publications, CRC Press, USA, 2010)
26. K Scott, S Burleigh, Bundle protocol specification. RFC 5050 (2007). [Online]. Available: http://www.rfc-editor.org/rfc/rfc5050.txt
27. VNGJ Soares, JJPC Rodrigues, F Farahmand, 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, pp. 1–5 (23–27 May 2010)
28. N Banerjee, MD Corner, BN Levine, An energy-efficient architecture for DTN throwboxes, in *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, Anchorage, Alaska, USA, pp. 776–784 (6–12 May 2007)
29. R Mahajan, J Zahorjan, B Zill, Understanding WiFi-based connectivity from moving vehicles, in *7th ACM SIGCOMM Conference on Internet Measurement (IMC 2007)*, San Diego, CA, USA, pp. 321–326 (24–26 October 2007)
30. M Research, VanLan: investigating connectivity from moving vehicles. [Online]. Available: http://research.microsoft.com/en-us/projects/vanlan/. Accessed July 2011
31. V Bychkovsky, B Hull, A Miu, H Balakrishnan, S Madden, A measurement study of vehicular internet access using in situ Wi-Fi networks, in *12th Annual International Conference on Mobile Computing and Networking (ACM MobiCom 2006)*, Los Angeles, CA, USA, pp. 50–61 (24–29 September 2006)
32. H-Y Huang, P-E Luo, M Li, D Li, X Li, W Shu, M-Y Wu, Performance evaluation of SUVnet with real-time traffic data. IEEE Trans Veh Technol. **56**(6), 3381–3396 (2007)
33. J Eriksson, H Balakrishnan, S Madden, Cabernet: vehicular content delivery using WiFi, in *14th ACM International Conference on Mobile Computing and*

Networking (MobiCom 2008), San Francisco, CA, USA, pp. 199–210 (14–19 September 2008)

34. KC Lee, S-H Lee, R Cheung, U Lee, M Gerla, First experience with cartorrent in a real vehicularad hoc network testbed, in *IEEE INFOCOM Workshop on Mobile Networking for Vehicular Environments (MOVE 2007)*, Anchorage, Alaska, pp. 109–114 (11 May 2007)
35. M Jerbi, S-M Senouci, MA Haj, Extensive experimental characterization of communications in vehicular ad hoc networks within different environments, in *65th IEEE Vehicular Technology Conference (VTC 2007)*, Dublin, Ireland, pp. 2590–2594 (22–25 April 2007)
36. C Pinart, I Lequerica, I Barona, P Sanz, D García, D Sánchez-Aparisi, DRIVE: areconfigurable testbed for advanced vehicular services and communications, in *1st Workshop on Experimental Evaluation and Deployment Experiences on Vehicular Networks (WEEDEV 2008), in conjunction with TRIDENTCOM 2008*, Innsbruck, Austria, pp. 1–8 (18 March 2008)
37. UMass Diverse Outdoor Mobile Environment (DOME) project, DieselNet [Online]. Available http://prisms.cs.umass.edu/dome/umassdieselnet. Accessed July 2011
38. H Soroush, N Banerjee, MD Corner, BN Levine, B Lynn, DOME: a diverse outdoor mobile testbed, in *1st ACM International Workshop on Hot Topics of Planet-Scale Mobility Measurements (ACM HotPlanet 2009)*, Kraków, Poland, pp. 1–6 (2009)
39. CarTel [Online]. Available: http://cartel.csail.mit.edu/doku.php. Accessed July 2011
40. Drive-thru Internet [Online]. Available: http://www.drive-thru-internet.org/index.html. Accessed July 2011
41. J Ott, D Kutscher, A disconnection-tolerant transport for drive-thru internet environments, in *IEEE Conference on Computer Communications (INFOCOM 2005)*, Miami, FL, USA, pp. 1849–1862 (13–17 March 2005)
42. Fleet testbed [Online]. Available: http://cartel.csail.mit.edu/doku.php?id=fleet_testbed. Accessed June 2011
43. Object Management Group, Inc., Unified Modeling Language (UML) [Online]. Available: http://www.uml.org/. Accessed July 2011
44. The LEGO Group, LEGO mindstorms NXT [Online]. Available: http://mindstorms.lego.com/en-us/default.aspx. Accessed July 2011
45. T. O. B. S. M. Site, Bluetooth special interest group [Online]. Available: https://www.bluetooth.org/apps/content/. Accessed June 2011
46. IEEE 802.11, The Working Group Setting the Standards for Wireless LANs, IEEE 802.11 Wireless Local Area Networks [Online]. Available: http://www.ieee802.org/11/. Accessed June 2011
47. S Jain, K Fall, R Patra, Routing in a delay tolerant network. ACM SIGCOMM. Comput Commun Rev. **34**(4), 145–158 (2004). doi:10.1145/1030194.1015484
48. A Vahdat, D Becker, Epidemic routing for partially-connected ad hoc networks. Duke University, Technical Report, CS-2000-06 (April 2000)
49. T Spyropoulos, K Psounis, CS 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, pp. 252–259 (22–26 August 2005)
50. VNGJ Soares, F Farahmand, JJPC Rodrigues, Traffic differentiation support in vehicular delay-tolerant networks. Telecommun Syst. **48**(1-2), 151–162 (2011). doi:10.1007/s11235-010-9325-z
51. M Rubinstein, FB Abdesslem, MD Amorim, SR Cavalcanti, RDS Alves, LHMK Costa, OCMB Duarte, MEM Campista, Measuring the capacity of in-car to in-car vehicular networks. IEEE Commun Mag. **47**(11), 128–136 (2009)
52. VNGJ Soares, F Farahmand, JJPC Rodrigues, Performance analysis of scheduling and dropping policies in vehicular delay-tolerant networks. Int J Adv Internet Technol IARIA. **3**(1-2), 137–145 (2010)

doi:10.1186/1687-1499-2012-13
Cite this article as: Dias *et al.*: Performance assessment of IP over vehicular delay-tolerant networks through the VDTN@Lab testbed. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:13.