



UNIVERSIDADE DA BEIRA INTERIOR
Faculty of Engineering

Performance Assessment of Aggregation and Deaggregation Algorithms in Vehicular Delay-Tolerant Networks

João Nuno Gabriel Isento

Submitted to the University of Beira Interior in candidature for the Degree of
Master of Science in Informatics Engineering

Supervised by Prof. Doutor Joel José Puga Coelho Rodrigues

Department of Informatics
University of Beira Interior
Covilhã, Portugal
<http://www.di.ubi.pt>

Acknowledgements

First of all, I would like to thank to Professor Joel José Puga Coelho Rodrigues for all the support, for the constant words of encouragement and for supervising my Master's Thesis. I also would like to thank for the opportunity to belong to his research group, Next Generation Networks and Applications Group (NetGNA).

I am most grateful to the University of Beira Interior, the Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, Portugal in the framework of the Project VDTN@Lab, and by the Euro-NF Network of Excellence from the Seventh Framework Programme of EU in the framework of the Specific Joint Research Project VDTN, for many kinds of all the support that was given to me. Additionally, I'd like to thank to all members of the VDTN team, but especially to my colleagues Bruno Silva, João Alfredo, and Vasco Soares for all the daily support and help.

My highest gratitude to all my family, especially to my parents and sister, for being a constant source of support, love, and inspiration. I am forever indebted to them.

Abstract

Vehicular Delay-Tolerant Networks (VDTNs) are a new approach for vehicular communications where vehicles cooperate with each other, acting as the communication infrastructure, to provide low-cost asynchronous opportunistic communications. These communication technologies assume variable delays and bandwidth constraints characterized by a non-transmission control protocol/internet protocol architecture but interacting with it at the edge of the network.

VDTNs are based on the principle of asynchronous communications, bundle-oriented communication from the DTN architecture, employing a store-carry-and-forward routing paradigm. In this sense, VDTNs should use the tight network resources optimizing each opportunistic contact among nodes.

At the ingress edge nodes, incoming IP Packets (datagrams) are assembled into large data packets, called bundles. The bundle aggregation process plays an important role on the performance of VDTN applications. Then, this paper presents three aggregation algorithms based on time, bundle size, and a hybrid solution with combination of both. Furthermore, the following four aggregation schemes with quality of service (QoS) support are proposed: 1) single-class bundle with $N = M$, 2) composite-class bundle with $N = M$, 3) single-class bundle with $N > M$, and 4) composite-class bundle with $N > M$, where N is the number of classes of incoming packets and M is the number of priorities supported by the VDTN core network. The proposed mechanisms were evaluated through a laboratory testbed, called VDTN@Lab. The adaptive hybrid approach and the composite-class schemes present the best performance for different types of traffic load and best priorities distribution, respectively.

Keywords

Delay-Tolerant Networks, Fragmentation, Performance Assessment, Routing, Vehicular Ad Hoc Networks, Vehicular Delay-Tolerant Networks, Testbed.

Contents

Acknowledgements	iii
Abstract	v
Keywords	vii
Contents	ix
List of Figures	xi
Acronyms	xiii
1 Introduction	1
1.1 Focus	1
1.2 Objectives	3
1.3 Main Contributions	3
1.4 Dissertation Structure	3
2 Related Work	7
2.1 Introduction	7
2.2 Delay-Tolerant Networks	7
2.3 Vehicular Delay-Tolerant Networks	8
2.4 Available Aggregation approaches	11
2.5 Proposed Aggregation Solutions	13
2.5.1 Traffic Modeling	14
2.5.2 Aggregation Mechanisms	15
2.5.3 Generalized Bundle Assembly Framework for QoS Support .	18
2.5.4 De-aggregation method	20
2.6 Summary	20
3 Research Methodology	23
3.1 Introduction	23
3.2 Vehicular Testbeds	23
3.3 VDTN@Lab Testbed	25
3.4 Summary	29

4	Performance Evaluation and Demonstration	33
4.1	Introduction	33
4.2	Network Scenario	33
4.3	Performance Evaluation of Aggregation and Deaggregation Mechanisms	34
4.3.1	Aggregation Algorithms	34
4.3.2	QoS Support Schemes	35
4.3.3	Network Performance Analysis	36
4.4	Summary	38
5	Conclusion and Future work	41
5.1	Conclusion	41
5.2	Future work	42
	References	45

List of Figures

1.1	Illustration of a typical vehicular network.	2
2.1	Illustration of the DTN store-carry-and-forward paradigm.	9
2.2	Comparison of TCP/IP, DTN, and VDTN architectures	10
2.3	Control information and data bundles exchange.	11
2.4	Illustration of incoming traffic models: a) Poison; b) Self-Similar Traffic.	14
2.5	Illustration of the bundle assembly process.	16
2.6	Proposed aggregation algorithms with QoS support: (a) Single-class bundle with $N = M = 4$; (b) Composite-class bundle with $N = M = 4$; (c) Single-class bundle with $N = 4$ and $M = 2$; (d) Composite-class bundle with $N = 4$ and $M = 2$	20
3.1	UML activity diagram of a mobile node, describing the control plane and data plane interaction, coordinated by decision module	27
3.2	A mobile node created with a Lego MINDSTORMS NXT robot and a coupled netbook.	28
3.3	Illustration of VDTN nodes interaction: (a) a mobile and a terminal node establishing contact, (b) a mobile and a relay node establishing contact and (c) two mobile nodes establishing contact.	28
3.4	UML class diagram illustrating the main classes and important relationships in VDTN testbed	29
4.1	VDTN@Lab laboratory testbed scenario.	34
4.2	Performance evaluation of proposed algorithms: Average aggregation delay as function of traffic load for different traffic loads (low , medium, and high).	35
4.3	Performance evaluation of proposed algorithms: Distribution of bundle priorities after aggregation algorithms.	36
4.4	Bundle average delay as function of traffic load for timer-based algorithm.	37
4.5	Bundle average delay as function of traffic load for threshold-based algorithm.	37
4.6	Bundle average delay as function of traffic load for hybrid algorithm.	37

Acronyms

BAD	Bundle Aggregation and De-aggregation
BSC	Bundle Signaling Control
DTN	Delay-Tolerant Network
DTNRG	Delay-Tolerant Network Research Group
UBI	Universidade da Beira Interior
FIFO	First-In-First-Out
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LIFO	Last-in-First-Out
LTP	Licklider Transmission Protocol
MAC	Media Access Control
MANET	Mobile Ad hoc Network
MTU	Maximum Transmission Unit
QoS	Quality of Service
RLSMP	Region-based Location Service Management Protocol
TCP	Transmission Control Protocol
TTL	Time-To-Live
UML	Unified Modeling Language
UDP	User Datagram Protocol
VANET	Vehicular Ad hoc Network
VDTN	Vehicular Delay-Tolerant Network

Chapter 1

Introduction

1.1 Focus

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 [?, ?, ?], networks to distribute multimedia content [?, ?], and monitoring networks for data collection [?]. Vehicular networks can also be employed to provide connectivity to remote rural communities and regions [?, ?, ?, ?, ?, ?], or to assist communication between rescue teams and other emergency services in catastrophe hit areas lacking a conventional communication infrastructure [?]. Figure 1.1 illustrates a typical vehicular network scenario. 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 [?, 33]. Limited transmission ranges, physical obstacles, and interferences lead to disruption and intermittent connectivity [?]. 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) [?, ?] 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 [?, ?, ?]. These limitations were overcome by applying the store-carry-and-forward paradigm of the delay-tolerant network (DTN) architecture [8], creating the concept of “DTN-enabled VANETs” [?, ?]. 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.

More recently, vehicular delay-tolerant networks (VDTNs) were proposed as an alternative network architecture for sparse vehicular networks [7]. VDTN



Figure 1.1: Illustration of a typical vehicular network.

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 work also addresses the performance evaluation of aggregation and de-aggregation algorithms for VDTNs through a testbed, 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

Introduction

applications; and (iii) operation under emulated realistic operating conditions.

1.2 Objectives

The main objective of this dissertation is the presentation of a detailed study about the available aggregation and deaggregation approaches for vehicular networks, and their impact on the performance of VDTN networks. To accomplish this main objective, the following partial objective were identified and performed: a detailed review of the state-of-the-art in aggregation approaches for vehicular networks and their application to VDTNs; Design and creation of modules for VDTN@Lab to support the studied aggregation algorithms; Performance evaluation of the studied aggregation mechanisms for VDTNs; Finally, dissertation writing summarizes all the above-mentioned partial objectives.

1.3 Main Contributions

This section is devoted to the scientific contributions of this dissertation to the state-of-the-art on VDTNs, more precisely about aggregation algorithms.

The first contribution of this dissertation is the presentation of an overview about all the main aspects and features of VDTN networks, which was published in IEEE Intelligent Transportation Systems Magazine, 2013 [?].

The first contribution of this dissertation is the construction, validation and deployment of a VDTN testbed, called VDTN@Lab, which was published in EURASIP Journal on Wireless Communications and Networking, 2012, with an impact factor of 0.54 [?].

The second contribution elaborates an initial study about aggregation algorithms on VDTNs, which was presented at 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2011), Valencia, Spain, October 17-22, 2011 [?].

The third contribution of this dissertation is a detailed research about aggregation algorithms with QoS support, which was presented in 13th International Conference, ADHOC-NOW 2014, Benidorm, Spain [?].

1.4 Dissertation Structure

This dissertation is organized in 5 chapters. This chapter, the first, presents the context of the dissertation, focusing on the topic under study, the objec-

tives, main contributions and dissertation structure.

Chapter 2 elaborates on the related work about the topic, focusing on aggregation and deaggregation solutions for the different kinds of above-presented vehicular networks.

Chapter 3 presents the VDTN@Lab testbed used for performance evaluation. It also describes the conceptual design and used technologies.

Chapter 4 focuses on the performance evaluation study of the implemented aggregation and deaggregation mechanisms. The performance metrics and network scenarios are also described in this chapter .

Finally, Chapter 5 concludes the dissertation and points some directions for future research works.

Chapter 2

Related Work

2.1 Introduction

Vehicular networks have emerged as a new wireless network solution due to the advances of wireless technologies and the automotive industry. Vehicular networks can be applied to a wide range of real-world scenarios. For example, they can be used to provide connectivity to remote rural regions [1, 2] or being used as the communication infrastructure in case of emergency scenarios [3]. However, these networks face several problems when considering the establishment of network connectivity among vehicles and between vehicles and roadside infrastructure. Most of the connectivity issues arise from the high mobility of vehicles, which leads to constant changes in the network topology and short contact durations. Another issue that these networks should overcome is the intermittent connectivity and disruption, which is caused by physical obstacles and limited transmission ranges. Moreover, the large distances and low node densities contribute to network partition. Consequently, a contemporaneous end-to-end path from source to destination often does not exist. In order to overcome the above-presented issues, several approaches were proposed, including VANETs (Vehicular Ad-hoc Networks), DTNs (Delay Tolerant Networks), and VDTNs (Vehicular Delay-Tolerant Networks), that are considered in this work. Afterwards, the state of art about aggregation and deaggregation solutions is presented. It includes aggregation approaches used on optical burst switching networks and vehicular communications.

2.2 Delay-Tolerant Networks

Delay-/disruption-tolerant networking (DTNs) focuses on the design, implementation, evaluation, and application of architectures and protocols that intend to enable data communication among heterogeneous networks in extreme environments. Examples include interplanetary networks, underwater networks, wildlife tracking networks, sparse wireless sensor networks, people networks, military tactical networks, transient networks, disaster recovery networks, and vehicular networks.

DTNs experience any combination of the following aspects: sparse connectivity, frequent partitioning, intermittent connectivity, large or variable delays, asymmetric data rates, and low transmission reliability. More importantly, end-to-end connection cannot be assumed to be available. In order to answer to these challenges the DTN Research Group (DTNRG), proposed an architecture (RFC 4838) [4] and a communication protocol (RFC 5050) [5] for DTNs. The DTN architecture [4] introduces a store-carry-and-forward paradigm by overlaying a protocol layer, called bundle layer, above the transport layer, which provides internetworking on heterogeneous networks (regions) operating on different transmission media. The bundle protocol [5] is end-to-end, strongly asynchronous, message (bundle) oriented.

Application data units are aggregated into one or more protocol data units called "bundles" by the bundle layer. The idea is to "bundle" together all the information required for a transaction (entire blocks of application-program data and metadata). This minimizes the number of round-trip exchanges, which is useful when the round-trip time is very large. To help routing and scheduling decisions, bundles contain an originating timestamp, a useful life indicator, a class of service assignment, and a length indicator. The bundle protocol also offers an optional hop-by-hop transfer of reliable delivery responsibility, called bundle custody transfer, and an optional end-to-end acknowledgement functionality (i.e., "return receipt"). When nodes accept custody of a bundle, they commit to retain a copy of the bundle until such responsibility is transferred to another node.

The store-carry-and-forward paradigm avoids the need for constant connectivity. It is used to move bundles across a region, exploiting node mobility. This paradigm, which is illustrated in Figure 2.1, can be described as follows. A source node originates a bundle and stores it using some form of persistent storage, until an appropriate communication opportunity becomes available. The bundle will be forwarded when the source node is in contact with an intermediate node that will be more close to the destination node. Afterwards, the intermediate node stores the bundle and carries it until a suitable contact opportunity occurs. This process is repeated and the bundle will be relayed hop by hop until (eventually) reaching its destination.

2.3 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

Related Work

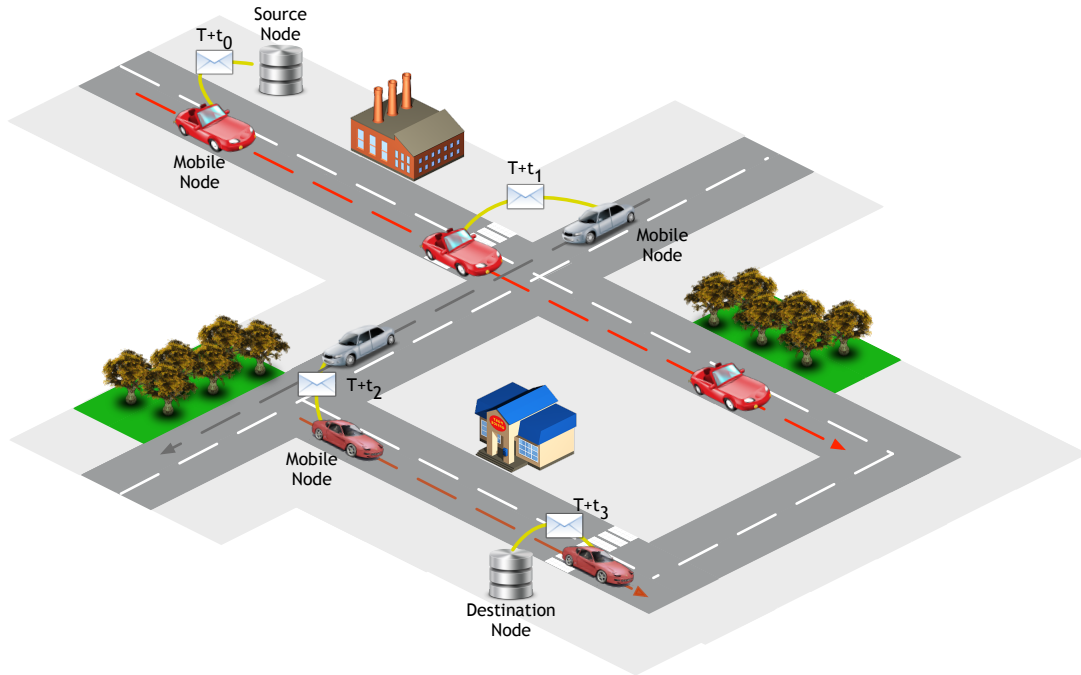


Figure 2.1: Illustration of the DTN store-carry-and-forward paradigm.

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 [6].

In order to support communication in sparse and disconnected vehicular network scenarios, VDTN presents a network architecture based on the following based design principles [7]: (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 [5], VDTN architecture places the bundle layer over the data link layer introducing an IP over VDTN approach [7]. The protocol data unit at the VDTN bundle layer is called a bundle, which 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 [8]. 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 [7], which are illustrated in Figure 2.2. 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.

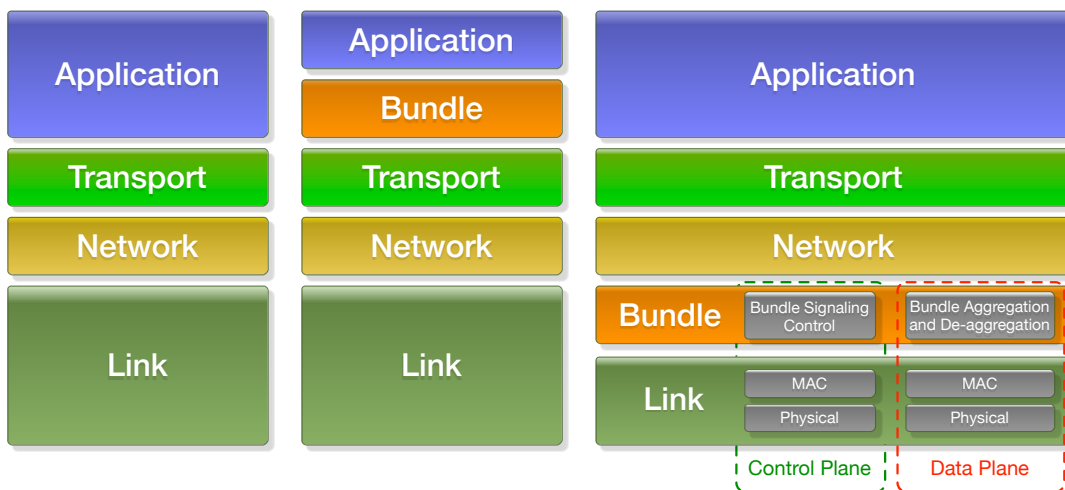


Figure 2.2: Comparison of TCP/IP, DTN, and VDTN architectures

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 [7, 9]. 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 [7, 10]. These nodes are usually power-limited since they may run on solar panels or batteries. Figure 2.3 illustrates this concept. At the time $t + t_0$, a mobile node and a relay node detect each other and start exchanging sig-

Related Work

nalizing 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.

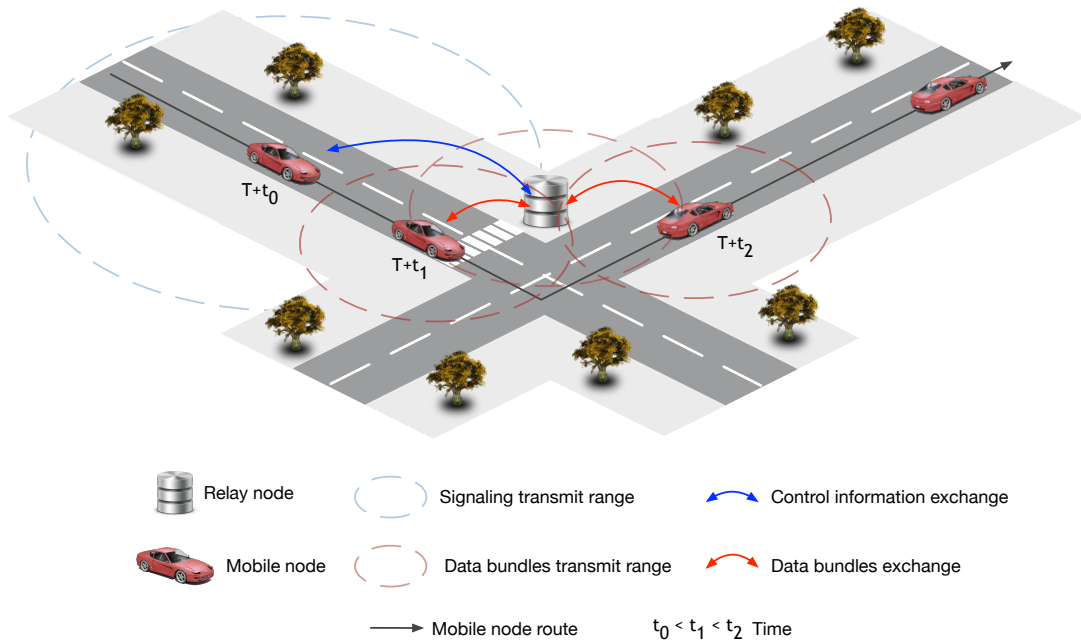


Figure 2.3: Control information and data bundles exchange.

2.4 Available Aggregation approaches

In this section, several aggregation solutions to real problems in vehicular communications are addressed. These problems show how important the aggregation algorithms are to improve the performance of protocols, services, and applications in vehicular networks.

Mobile ad-hoc networks face security risks and energy consumptions issues due to limits resources and its infrastructureless network environment. To perform a monitoring and detection in a MANET the authors in [11] develop both lossless and lossy aggregation mechanisms to reduce the energy cost and bandwidth consumption while preserving the detection accuracy. Lossless aggregation intent to reinforce the detection information, in which the de-aggregated information contains the same data as the original one. In the lossy mechanism, the decompressed data contains most of the original information accepting some loss of fidelity. Real-world experiments and simulations are conducted to assess

the effectiveness of this solution.

VoIP traffic over mobile ad-hoc networks faces several issues related to the mobility of the nodes and routing protocols introducing latency and packet loss rate. Authors in [12] present several solutions to improve the quality of VoIP calls in MANET environment. One of these solutions relates to aggregate several VoIP frames within a single packet. Due to this feature, and depending on the voice codec used, the performance of the VoIP traffic was improved. Several simulations were performed using the NS-2 simulator with the *NS2Voip++* extension and each simulation had the duration of 4000 seconds.

K-hop bandwidth aggregation [13, 14] is proposed for enhance cooperative video streaming over hybrid vehicular ad-hoc networks and 3G/3.5G cellular network. In this approach each member (vehicle) in the same group is known to each other and share the same path and the same destination. The video requests are sent over 3G/3.5G network. However, the bandwidth in this network is very limited and not support higher quality video. In order to improve the quality of video playback the requester can ask to other members, called helpers, to cooperatively download the requested video through their cellular network. Then, all video "fragments" are sent to the requester over ad hoc network. k-hop aggregation solves the problem of discovering the best helpers and how the data is sent to the requester. This approach can get better quality video that cooperative video streaming without the K-hop aggregation scheme.

In a vehicular ad hoc network, one of the main issues is the problem of the message routing, where the location of the destination node is unknown due to the high node mobility. One of the best solution for this problem is to use location service protocols. However, these protocols require large volume of signaling overhead which can result in network traffic congestion and high resource consumption. Region-based Location Service Management Protocol (RLSMP) [15] uses message aggregation enhanced by geographical clustering to minimize the volume of the signaling overhead. In this protocol, each vehicle frequently check its geographical cluster while moving, without any additional communication or delay. RLSMP uses message aggregation to bypass the drawbacks of independent updating and querying messages. The messages are aggregated so that data is grouped, resulting in more efficient bandwidth usage and minimal communication overhead. This protocol guarantees scalability with the increase of the number of nodes in the network.

In [16], authors propose an adaptive packet aggregation for header compression. Aggregation packets in a vehicular environment can cause redundant information and, therefore, high volume of traffic. This algorithm, first, perform static header compression, and then aggregate a dynamic number of pack-

Related Work

ets according to the delay constraints. Using the adaptive procedure, the system will adaptively change the number of aggregated packets according to the current end-to-end delay. Simulation results, using the NS-2 simulator, showed that the proposed aggregation algorithm outperforms the existing schemes in terms of end-to-end delay and packet delivery rate.

In [17-19], the authors intent to study the effect of aggregation of DTN bundles on space communications. This work tries to find if aggregation of multiple DTN bundles has advantages over the default approach of "one bundle per block". In space communications, bundles are passed as "service data units" to Licklider Transmission Protocol (LTP) for transmission. LTP is an end-to-end protocol for deep space communications. These service data unites are encapsulated in LTP blocks. Then, each LTP block is divide into LTP data segments, according the maximum transmission unit (MTU) size. To reduce the amount of ACK traffic, multiple DTN bundles are aggregated into a single LTP block. The aggregation of multiple bundles has significant advantages over the default approach.

The complete failure of telecommunications infrastructures is a common characteristic when a disaster occurs. A DTN-based solution is presented in [20] in order to aggregate as much information as possible from an area of interest (Aoi) within a disaster zone. Users using mobile phones generate messages with disaster-related information and aggregate them with their respective coverage areas into a new message to minimize the overall message collection delay. To prevent duplicate message from users, this system a filter is constructed to drop possible duplicated messages. Through simulation, using a real geographical map, this solution achieved a smaller delay in message delivery.

All of these abovementioned solutions have contributed to the conception and design of the proposed VDTN aggregation algorithms. All of these aggregation schemes show the influence that the packet aggregation has in vehicular communications.

2.5 Proposed Aggregation Solutions

This section will present a study of traffic modeling and its importance to evaluate the proposed aggregation algorithms for VDTNs. These algorithms will be described and analized. Lastly, some mechanism to QoS support for VDTN networks will be presented.

2.5.1 Traffic Modeling

Understanding network traffic behavior is essential for all aspects of network design and operation, e.g., component design, protocol design, and resource provisioning. The bundle assembly mechanisms search the best way to aggregate incoming data packets to bundles in order to minimize the delay time and size of created bundles. To study the performance of these algorithms is necessary to find a viable model that reproduces the real internet traffic to a laboratory environment. Then, exists two main models to traffic modeling.

Data traffic such as Internet traffic is traditionally modeled as a Poisson process with exponentially distributed time between individual packet generations. The accuracy of this model for simulating data traffic is, however, being questioned in the literature [21]. The first is a Poisson generator with exponentially distributed time between individual packet generations (see Figure 2.4 a)). This is a traditional and commonly used model, which developed from early modeling of call arrivals at telephone exchanges. The parameters to this model is shown in (2.1).

$$P(\text{inter-arrival time} = t) = \lambda e^{-\lambda t} \quad (2.1)$$

$$\text{Mean inter-arrival time} = 1/\lambda$$

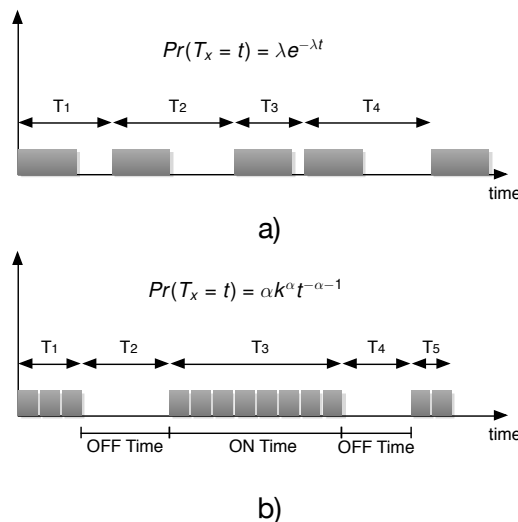


Figure 2.4: Illustration of incoming traffic models: a) Poisson; b) Self-Similar Traffic.

The second model is a self-similar traffic generator, modeled as an ON/OFF source where the time spent in each state fits a Pareto distribution (Figure 2.4 b)) with parameters $a = 1.2$ and $k = 1$. Packets are generated during the ON

Related Work

periods only, at regular intervals at a rate determined by the specified source activity level. The drawback of the Pareto distribution is that it exhibits infinite variance, so there is a non-trivial probability that the ON time may be longer than the simulation runtime. In (2.2) is shown the parameters to this model.

$$P(\text{inter-arrival time} = t) = \alpha k^\alpha t^{-\alpha-1} \quad (2.2)$$

$$\text{Mean inter-arrival time} = (\alpha/(\alpha - 1))k$$

2.5.2 Aggregation Mechanisms

The bundle assembly process can be modeled as a queuing system and is illustrated in Figure 2.5. In this model, the routing module (RM) organizes incoming packets in buffers according to their destination and QoS requirement. The terminal node assembles these packets into bundles according to a pre-select assembly algorithm. The notations q_1, q_2, \dots, q_N are the queues for packets having the same destination and QoS requirement. T_i is the assembly time of queue i . L_i is the data length of queue i (i is a positive integer and $1 \leq i \leq N$). For convenient analysis, some parameters are defined as follows.

T_{th} : Threshold of the time-based assembly algorithm.

Δt : Incremental value change of the time threshold.

L_{th} : Threshold of the bundle length algorithm.

Δl : Incremental value change of length threshold.

ξ : Encapsulate coefficient (for QoS class, etc.).

The bundle assembly process can be described as the following five steps.

Step 1 - *Assembly*: all incoming packets are sorted and stored in queues q_1, q_2, \dots, q_N ;

Step 2 - *Judgment*: when a packet arrives at queue i , judge whether the threshold is reached or not.

Step 3 - *Encapsulation*: if the threshold of queue i is reached, then encapsulate all packets into a bundle;

Step 4 - *Buffering*: store the bundle in the node buffer according to a first-in first-out principle; and

Step 5 - *Transmission*: transmit the bundle when a contact opportunity occurs.

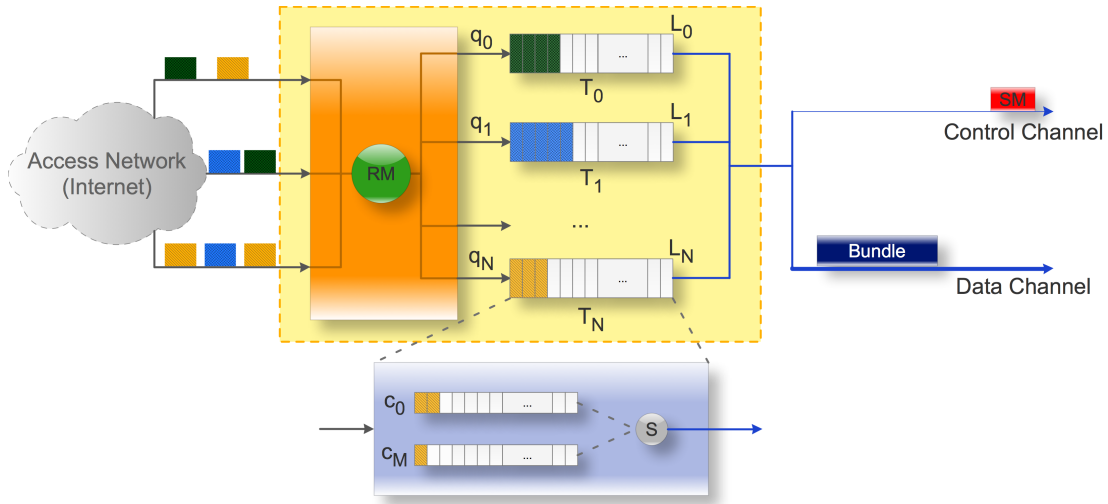


Figure 2.5: Illustration of the bundle assembly process.

Timer-based Algorithms

Usually, a bundle is created and injected into VDTN network at periodic time intervals in time-based algorithms. These algorithms use a timer to indicate an assembly cycle of each queue. There is a fixed threshold T_{th} that acts as the primary criterion to create a bundle. Thereby, they provide almost uniform gaps between successive bundles from the same queue into VDTN network. However, the bundle length varies with the offered load. To avoid injecting small bundle or even vacant bundle, the assembly time threshold algorithm can be described cursorily as follows.

1. Initialize: $T_i = 0$.
2. Waiting: when a packet with length pk_size arrives at i , insert the packet into queue i (if queue i is empty, then start its timer).
3. From $i = 0$ to N :
 - if** $T_i \geq \xi \times T_{th}$ **then**
 - Encapsulate all packets of queue i into a bundle and store the bundle in the buffer.
 - Reset $T_i = 0$ and return to step 2.
 - else**
 - return to step 2.

Time-based algorithms only consider the encapsulation coefficient and the assembly time. The assembly delay (T_{aT}) of time-based algorithms is equal to product of ξ and T_{th} . At a given encapsulation coefficient ξ , T_{aT} is only relating to T_{th} .

Related Work

Threshold-based Algorithms

In threshold-based algorithms, a bundle is created and injected into VDTN network once the assembly data length reaches or exceeds the given threshold L_{th} . Threshold-based algorithms generate bundles at non-periodic time intervals. A length threshold is used as the primary criterion to create a bundle. The threshold is placed as a limiting parameter on the minimal length or the number of packets contained. In these algorithms, once the threshold is reached, all packets in queue i are assembled into a bundle. The algorithm can simply be described as follows.

1. Initialize: $L_i = 0$.
2. Waiting: when a packet with length pk_size arrives at queue i , insert the packet into queue i and queue renews the assembling data length queue $L_i = L_i + pk_size$
3. From $i = 0$ to N :
if $L_i \geq \xi \times L_{th}$ **then**
 Encapsulate all packets of queue i into a bundle and store the bundle in the buffer.
 Reset $L_i = 0$ and return to step 2.
else
 return to step 2.

If we assume that traffic is distributed uniformly for destinations, the arrival rate for queue i is equal to V_i/N at a given real-time data arrival rate V_i from access networks. The assembly data length can be given by (2.3) in length-based algorithms, V_o is the average data rate from the access networks to an edge node. The assembly delay (T_{aL}) of length-based algorithms is given by (2.4).

$$L_i = \left(\frac{V_i}{N}\right) \times T_i = \left(\frac{V_o \times \rho_n}{N}\right) \times T_i \quad (2.3)$$

$$T_{aL} = \frac{\xi \times L_{th} \times N}{V_i} = \frac{\xi \times L_{th} \times N}{V_o \times \rho_n} \quad (2.4)$$

Adaptive Hybrid Algorithms

The above two types has some shortcomings under certain conditions. For example, the threshold-based algorithms may will experience a long assembly

waiting time when the input traffic is low. Timer-based algorithms solve this problem, however, under high traffic conditions, it will have a larger variance of bundle length. Both types of algorithms are too rigid and not adapt their parameters according to input traffic conditions. Then, some hybrid solution have been proposed. These algorithms can dynamic adjust the time, threshold, or both values, according to the real-time input traffic situation. A general adaptive hybrid algorithm can be depicted as follows.

1. Initialize: $T_i, L_i = 0$.
 2. Waiting: when a packet with length pk_size arrives at queue i , insert the packet into queue i and queue renews the assembling data length queue $L_i = L_i + pk_size$, (if queue i is empty, then start its timer).
 3. From $i = 0$ to N , calculate $B_i = \xi \times T_i \times L_i$.
- if** $L_i \geq \xi \times L_{th}$ **or** $T_i \geq \xi \times T_{th}$ **then**
 Encapsulate all packets of queue i into a bundle and store the bundle in the buffer.
 Reset $T_i = 0, L_i = 0$ and return to step 2.
- if** $T_i \geq \xi \times T_{th}$ **or** $L_i \leq \xi \times L_{th}$ **then**
 $T_{th} = T_{th} + \Delta t$ **or** $L_{th} = L_{th} - \Delta l$
- if** $T_i \leq \xi \times T_{th}$ **or** $L_i \geq \xi \times L_{th}$ **then**
 $T_{th} = T_{th} - \Delta t$ **or** $L_{th} = L_{th} + \Delta l$
- else**
 return to step 2.

2.5.3 Generalized Bundle Assembly Framework for QoS Support

In this section a generalized framework for bundle assembly, based on [22], will be presented. The primary issues are which class of packets and how many of packets of each class to put into a same bundle. To provide QoS support, the bundle assembly policies should take into account the number of packet classes as well as the number of bundle priorities supported in the VDTN core.

Let N be the number of input packet classes at the edge and let M be the number of bundle priorities supported in the VDTN core. Given packet classes and bundle priorities, the objective is to meet the QoS requirements by defining a set of bundle types, which specify how packets are aggregated, and by assigning an appropriate bundle priority to each bundle type. In this model, we define the length of the bundle by the number of packets in the bundle. Let K be the number of bundle types, where $M \leq K \leq (2^N - 1)$. A bundle type of type K is characterized by the following parameters, among others:

- R_{jk}^{MIN} : minimum number of packets of Class j in a bundle of type k ;

Related Work

- R_{jk}^{MAX} : maximum number of packets of Class j in a bundle of type k ;
- $S_k = \{j \mid R_{jk}^{MAX} > 0\}$: the set of packet classes which may be included in a bundle of type k ;
- P_k : priority of bundle of type k ;
- $C_k : C_k \subseteq S_k$:

Basically, two main approaches exist with two variants each one, as shown in Figure 2.6.

Single-Class Bundle with $N = M$

In this case (Figure 2.6 (a)), $N = M$, we can create bundle types such that each bundle contains a single class of packets, i. e., $S_k = \{k\}$ and, then, $P_k = k$. The priority of a bundle will directly correspond to a specific class of packets contained in the bundle.

Composite-Class Bundle with $N = M$

In this approach (Figure 2.6 (b)), packets are placed in the bundle in decreasing order of class, such that the higher class packets are at the head of the bundle. A bundle of type k is generated if the number of packets of Class k is equal to the threshold L_{th} or if the timeout T_{th} expires, depending on the aggregation algorithm.

Single-Class Bundle with $N > M$

In this approach (Figure 2.6 (c)), we have $K = N$ types of bundles, where each bundle consists of packets of a single class ($S_k = \{k\}$). However, several bundle types will have the same bundle priority. Each bundle is assigned one of the two bundle priorities. Bundles containing either Class 0 or Class 1 packets have Priority 0, while bundles containing either Class 2 or Class 3 packets have Priority 1.

Composite-Class Bundle with $N > M$

We now consider composite-class bundles for the case $N > M$. A bundle of type k is created when the sum of incoming packets of classes in C_k is equal to threshold L_{th} . Once the threshold or timer values are reached, a bundle of type k containing packets defined by k is generated by appending all constituent

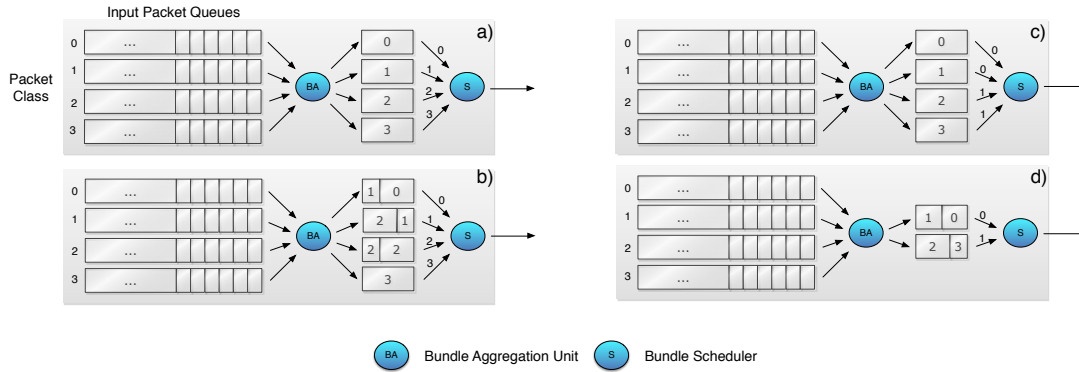


Figure 2.6: Proposed aggregation algorithms with QoS support: (a) Single-class bundle with $N = M = 4$; (b) Composite-class bundle with $N = M = 4$; (c) Single-class bundle with $N = 4$ and $M = 2$; (d) Composite-class bundle with $N = 4$ and $M = 2$.

class packets into the bundle such that the highest class packet in that bundle type is at the head of the bundle.

2.5.4 De-aggregation method

As above-mentioned, the aggregation process takes place at the source terminal nodes that generate bundles for VDTN network. Then, when these bundles arrive at the destination node, each one will be de-aggregated into their original datagrams. These disassembled datagrams are delivered to the upper layers till the Application Layer or forwarded to the external network (e.g. Internet) till their final destination user. The de-aggregation process is common for all the aggregation mechanisms.

2.6 Summary

This chapter reviewed the state of the art concerning the vehicular networks paradigm and presented them in detail taking into account its most relevant aspects. It is also presented, a reviewed about aggregation approaches.

After a brief introduction, Section 2.2 described DTN architecture and all its features. The Section 2.3 revealed the VDTN architecture, a new approach for vehicular networks. Then, in Section 2.4, some related work about the aggregation problematic were presented. Finally, Section 2.5 presents the aggregation and de-aggregation solutions to VDTN networks.

Chapter 3

Research Methodology

3.1 Introduction

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.

3.2 Vehicular Testbeds

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 [23, 24] is a testbed composed of 11 base-stations 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 [25], 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 [26]. 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 [27] 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 [28], 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 [29]. 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 [30] 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 [31, 32], CarTel [33, 34], and Drive-Thru [35, 36] 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 [31, 32] 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 [37] is composed of 27 cars, each of them running a CarTel

Research Methodology

[33, 34] 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 [35, 36] 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 [30, 33]. One of the developed vehicular mobility models available in VDTN@Lab was inspired by Diesel-Net [31, 32]. 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.

3.3 VDTN@Lab Testbed

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.

Unified modeling language (UML) [38] 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. All network nodes execute the same actions in the control plane and in the data plane. However, termi-

nal 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 be 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 3.1 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.

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

Research Methodology

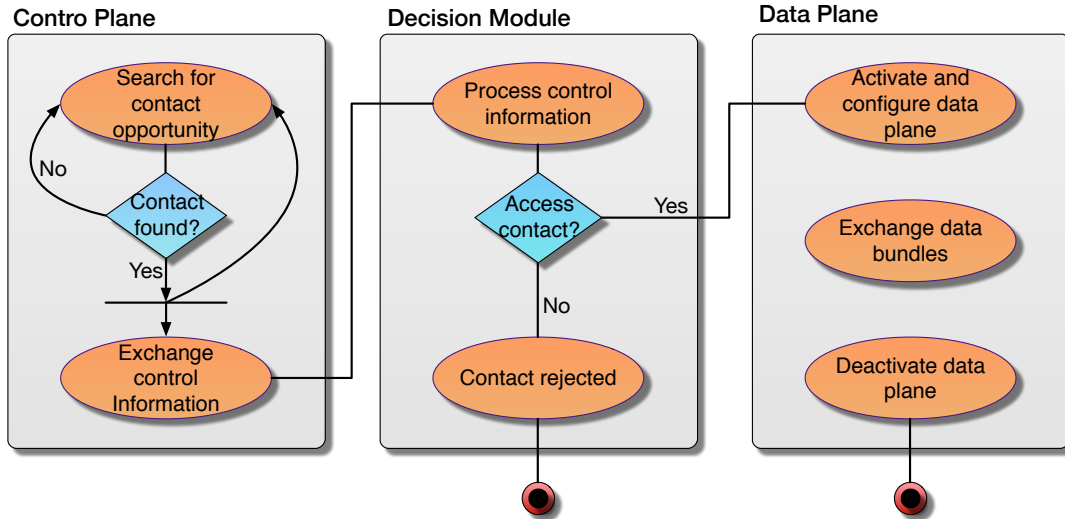


Figure 3.1: UML activity diagram of a mobile node, describing the control plane and data plane interaction, coordinated by decision module

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 [39] and a netbook computer. A mobile node is shown in Figure 3.2. 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 [40] and IEEE 802.11 b/g [41] 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 3.3 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. They also 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.



Figure 3.2: A mobile node created with a Lego MINDSTORMS NXT robot and a coupled netbook.

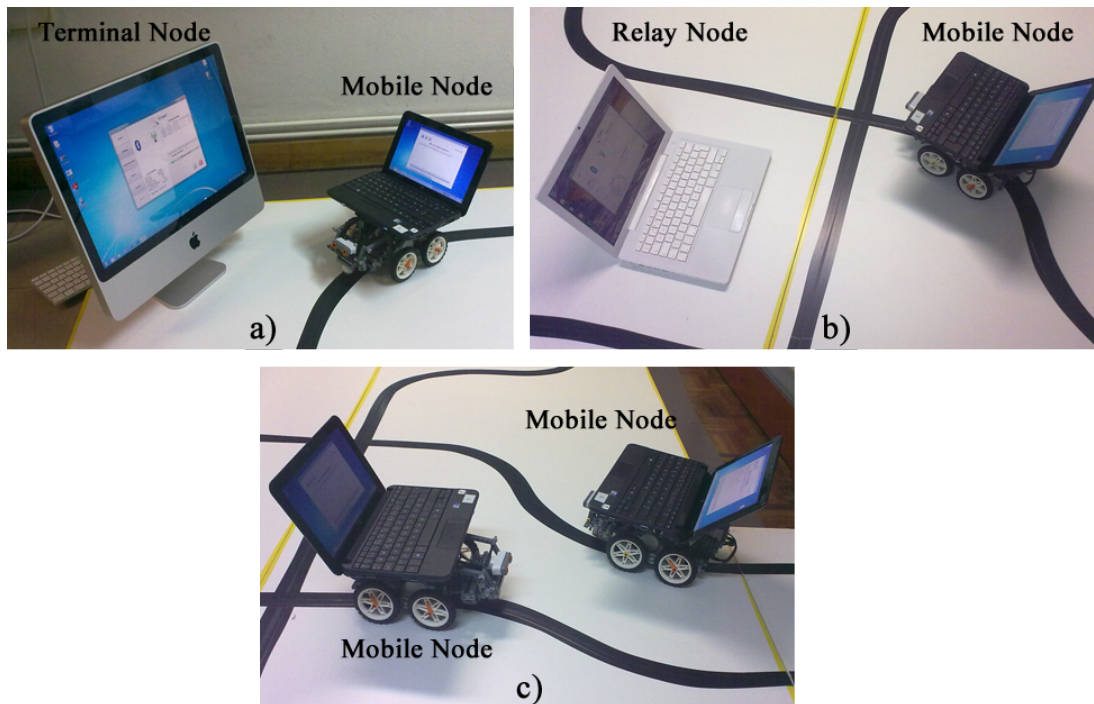


Figure 3.3: Illustration of VDTN nodes interaction: (a) a mobile and a terminal node establishing contact, (b) a mobile and a relay node establishing contact and (c) two mobile nodes establishing contact.

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.

The class diagram shown in Figure 3.4 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

Research Methodology

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 [42], and with the De/ Aggregation class that is responsible for forming data bundles by aggregating IP packets, according to a bundle assembly algorithm.

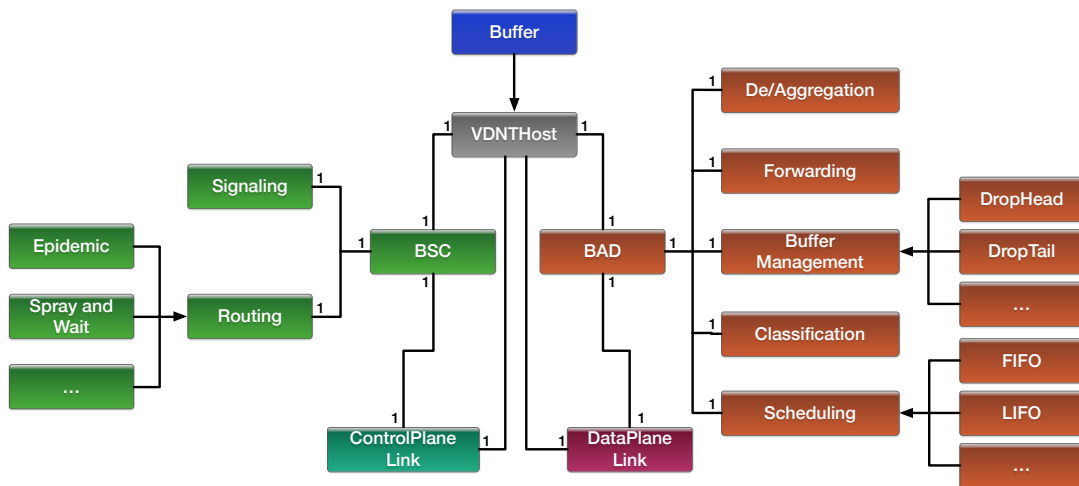


Figure 3.4: UML class diagram illustrating the main classes and important relationships in VDTN testbed

3.4 Summary

In this chapter, the design, construction of aggregation and de-aggregation mechanisms for VDTNs was discussed. These mechanisms are deployed on a laboratory testbed, called VDTN@Lab. In the literature several testbeds are used as a research method and have been developed to support the deployment and evaluation of architectures and protocols in vehicular networks. These

testbeds were presented in Section 3.2. The Section 3.3 described the laboratory testbed, called VDTN@Lab where the designed aggregation approaches were deployed.

Performance Evaluation and demonstration

Chapter 4

Performance Evaluation and Demonstration

4.1 Introduction

This chapter focuses on the performance evaluation of VDTN aggregation mechanisms described in Chapter 2. The study was conducted through experimentation, using VDTN@Lab laboratory testbed that was described in Chapter 3. In the related literature there are several studies about DTNs and VDTNs networks, namely, in [43]. This study may be considered as a basis for setting up network scenarios for performance assessment and comparison purposes.

The remainder of this chapter is organized as follows. Section 4.2 shows the networks scenarios used for the testbed experiments, and The impact of aggregation schemes on the performance of VDTN networks is presented in Section 4.3.

4.2 Network Scenario

The scenario that was set up to conduct experiments on the VDTN testbed has a dimension of 36,5m² and is shown in Figure 4.1. It consists in three fixed terminal nodes, four mobile nodes, and two relay nodes. Terminal nodes are placed at different points (edges) of the laboratory. Parallel with a study based on a testbed composed by real vehicles [44], and assuming a scale of 1:50 (1 min the laboratory testbed represents 50 min 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. Bundles have random source and destination terminal nodes and are generated assuming different aggregation mechanisms. When a bundle reaches its final destination it is marked as delivered. For testbed experiments, the bundle's 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.

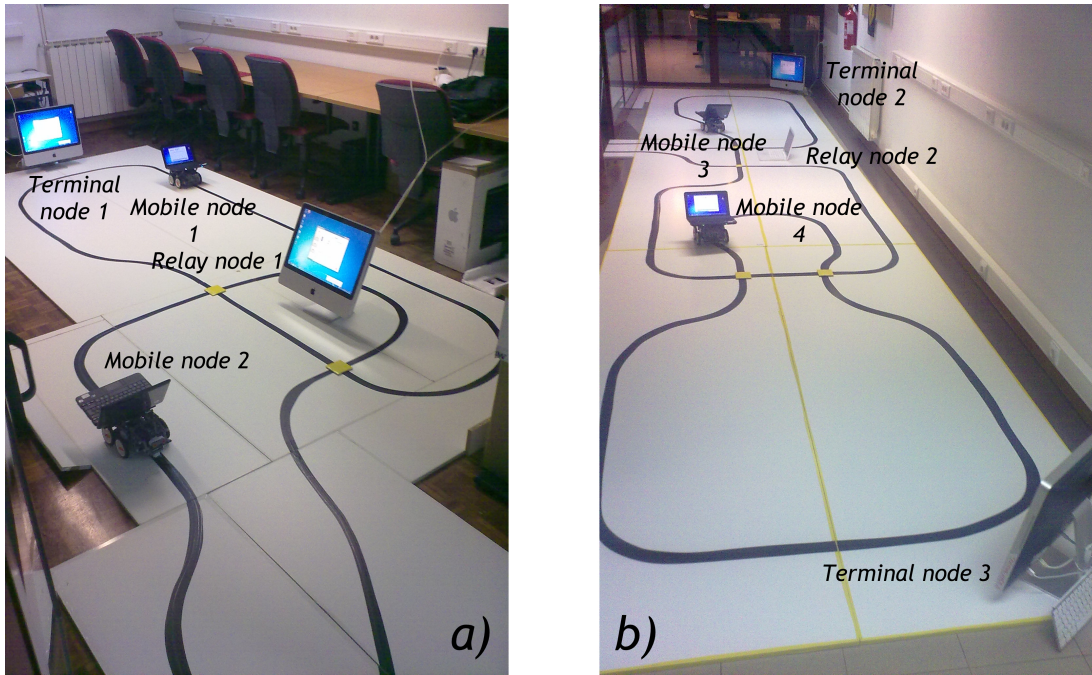


Figure 4.1: VDTN@Lab laboratory testbed scenario.

4.3 Performance Evaluation of Aggregation and Deaggregation Mechanisms

4.3.1 Aggregation Algorithms

The aggregation algorithms have been evaluated over different input traffic conditions. Input traffic (datagrams) is created in the same application of terminal nodes and this application generates different types of traffic load (considering low, medium, and high). In such a case of a low traffic load, datagrams are created in a time interval of 1 up to 3 seconds; medium traffic load in the time interval of the 0.5 up to 1.5 seconds, and the high traffic load in the time interval of the 0.25 up to 0.75 seconds. These datagrams have random destination terminal nodes and its size is uniformly distributed and average length is 1250 Bytes. Data bundles have a time-to-live (TTL) fixed on 10 minutes. The timer and threshold values changes among different testbed experiments in order to evaluate the performance of bundle assembly.

For the timer-base algorithm the timer value is fixed on 120 seconds. Threshold-based algorithm threshold value is equal to 150 datagrams. Hybrid algorithm has both the timer and threshold values.

Every result show that every algorithm performance had better results with a proper network traffic algorithm. This because it was proved in the literature as good simulators of real network traffic and real internet usage.

Figure 4.2 shows the average aggregation delay as function of traffic loads

Performance Evaluation and demonstration

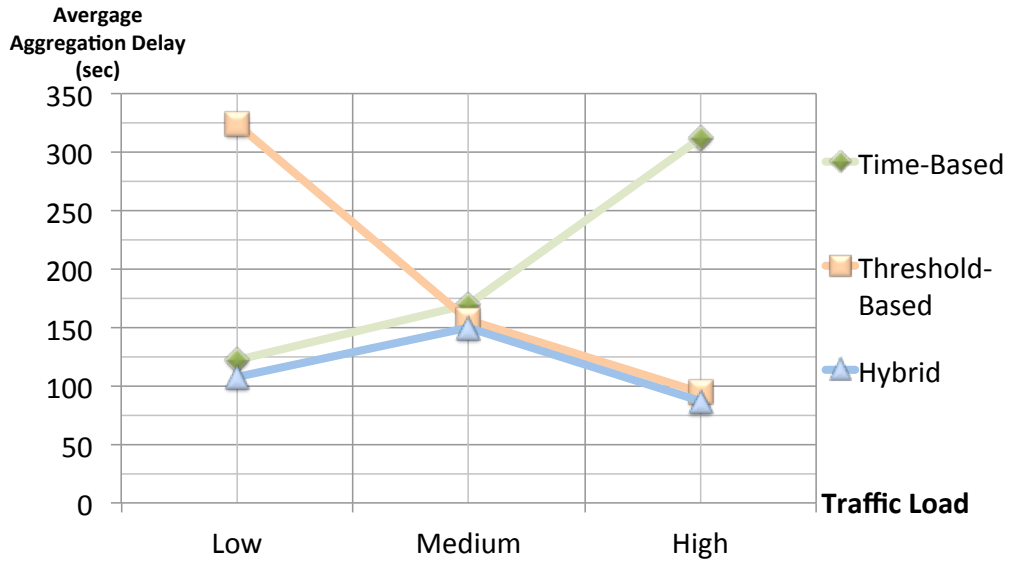


Figure 4.2: Performance evaluation of proposed algorithms: Average aggregation delay as function of traffic load for different traffic loads (low , medium, and high).

(easy, medium, and high). For this study is used all aggregation algorithms above described. As may be seen, the Hybrid Algorithm presents the best results and gains of approximately 14, 20, and 225 (for low, medium, and high traffic loads) when compared to the Timer-based Algorithm. Comparing Hybrid Algorithm with the Threshold-based Algorithm, it presents gains of approximately 216, 6, and 7 seconds. Figure 4.2 confirms the theoretical behavior of these algorithms. Under low traffic load, timer-based algorithms are the best technique for bundle generation and, under high traffic level, threshold-based algorithms have a better behavior for improvement of VDTN bundle aggregation. Thus, it is natural that Hybrid Algorithm, based on the two approaches, is the most efficient aggregation algorithm.

4.3.2 QoS Support Schemes

For the QoS schemes, we intent to prove that the QoS requirements of incoming packets not change with bundle assembly. For this, a uniform distribution of classes number was calculated. Equation (4.1) one possible distribution. Value $p(i)$ represent the percentage of packets with class i . $p(j)$ represent the percentages of previous classes that $i \geq j \leq n - 2$.

$$p(i) = \left(\frac{100 - (\sum_j^{n-1} p(j))}{i + 2} \right) \times 2 \quad (4.1)$$

Then, we create incoming packets with four classes: Class 0 (more prior-

Performance Evaluation and demonstration

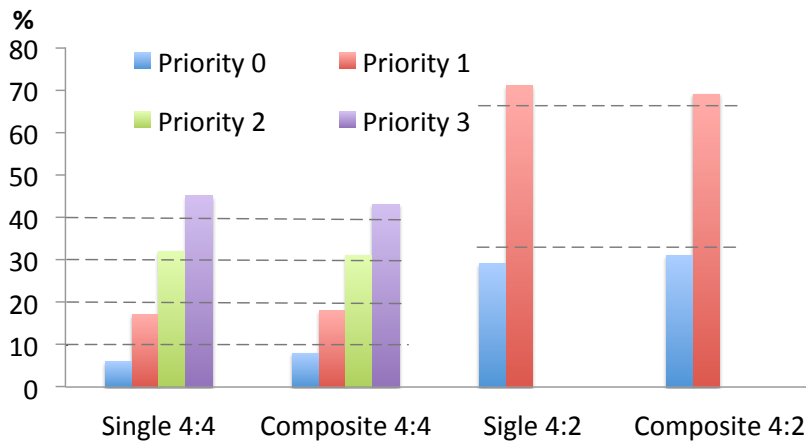


Figure 4.3: Performance evaluation of proposed algorithms: Distribution of bundle priorities after aggregation algorithms.

ity), Class 1, Class 2, and Class 3 (less priority). By (4.1), the distribution is 40%, 30%, 20%, and 10% respectively. We assumed that incoming packets arrival at the terminal node according to a Poisson process and average packet length is 1250Bytes. The four schemes above described has been tested: Single-Class Bundle with $N = M = 4$ (Single 4:4), Composite-Class Bundle with $N = M = 4$ (Composite 4:4), Single-Class Bundle with $N = 4$ and $M = 2$ (Single 4:2), and Composite-Class Bundle with $N = 4$ and $M = 2$ (Composite 4:2).

Figure 4.3 shown the distribution, in percentage, of priorities applied in VDTN network core. The gray lines represent the optimal values of these percentages given by (4.1). In general, four schemes produces an acceptable distribution of priorities, with the Composite schemes having better results. This because the aggregation of different classes in a single bundle.

4.3.3 Network Performance Analysis

The network performance of routing protocols using the proposed aggregation algorithms is analyzed in this subsection. This sub-section studies the performance evaluation of the above-presented VDTN testbed using Epidemic, Spray and Wait, and PRoPHET routing protocols. Based on previous results, three aggregations algorithms are considered to perform this analysis. Figure 4.4 shows the bundle average delay when pure timer-based aggregation algorithms are used. In average, bundle delay shown in Figure 4.5 is used threshold-based algorithms. In the last figure (Figure 4.6)), the evaluation is performed for the hybrid aggregation approach.

As shown, the network performance is very similar with different aggregation algorithms because the aggregation algorithms not influence the perfor-

Performance Evaluation and demonstration

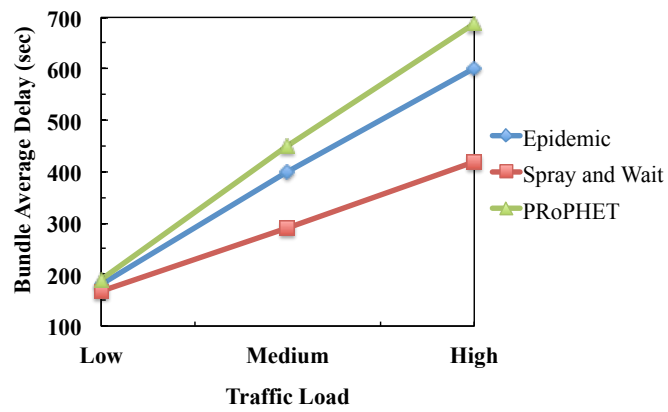


Figure 4.4: Bundle average delay as function of traffic load for timer-based algorithm.

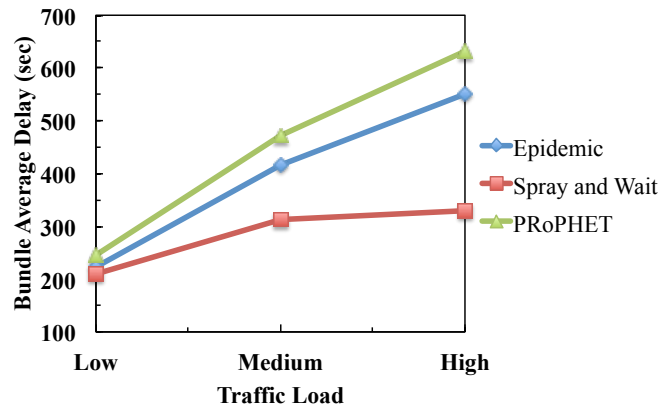


Figure 4.5: Bundle average delay as function of traffic load for threshold-based algorithm.

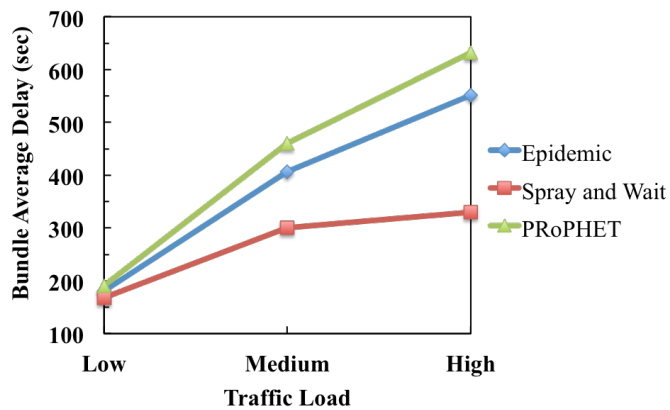


Figure 4.6: Bundle average delay as function of traffic load for hybrid algorithm.

mance of routing protocols. However, small differences can be observed on the overall network performance. Comparing the three figures, one can observe that timer-based algorithms perform better under low traffic load and the threshold-based approach perform better under high traffic load. The Hybrid Algorithm combines the advantages of both. In terms of routing protocols, Spray and Wait performs better along all the studies and P_{Ro}PHET presents the worst network performance. When traffic load increases, this different behavior is more significant.

4.4 Summary

A performance analysis of aggregation mechanisms for VDTNs was presented. The networks scenarios were described in Section 4.2, presenting its main features, such as, mobile nodes mobility patterns, datagrams creation interval, bundle TTL. Section 4.3 was focused on the performance of aggregation protocols when enforced on the VDTN@Lab testbed. This study was conducted following several approaches. First, a performance assessment of aggregation protocols is presented. With different traffic loads, the proposed aggregation approaches were evaluated. Then, different classes of datagrams were created to evaluate the performance of the QoS support framework. Finally, a generalized experiment using all nodes of VDTN testbed was created to evaluate the impact of aggregation in network performance.

Conclusion and Future work

Chapter 5

Conclusion and Future work

5.1 Conclusion

Along this dissertation the state-of-the-art of vehicular networks was presented and the performance of aggregation and deaggregation protocols was studied. This chapter presents a synthesis of the main achievements and points to several directions for future work.

After introducing, delimiting the topic of the dissertation, describing the objectives, and showing the main contributions, in Chapter 2, the state of the art of vehicular networks, their architectures were presented. The state of the art about aggregation approaches was also presented in this chapter. It began with the description of DTNs giving special attention to routing in this type of vehicular network approach. Other vehicular network approach was also considered and described, namely, VDTNs. A study of several aggregation solutions is also presented in this chapter. Finally, the aggregation algorithms to VDTNs.

Chapter 3 studied the existing testbeds for performance evaluation of vehicular networks. After, a VDTN laboratory testbed (VDTN@Lab) created for VDTNs performance evaluation was described. Afterwards, the conceptual design of the proposed aggregation algorithms to VDTN networks. To conclude this chapter the used technologies for the accomplishment of this dissertation were enumerated.

In Chapter 4, the performance assessment of routing protocols and fragmentation approaches in VDTNs were analyzed. In terms of aggregation algorithms, it was focused on the performance of Timer-based algorithms, Threshold-based algorithms and Hybrid algorithms. In what concerns to the proposed QoS support framework it was focused on Single-Class Bundle and Composite-Class Bundle.

The observed results of the testbed experiments were presented and discussed. It was shown that timer-based algorithms should be considered for low traffic conditions and threshold-based algorithms for high traffic load environments. Surely, the hybrid approach combines the advantages of both, under any traffic load. For the QoS support Framework the Composite-Class Bundle with $N > M$ presents the best results.

5.2 Future work

To conclude this dissertation, it remains to suggest future research directions that result from this research work:

- The study of compression approaches to provide more aggregation ratio in order to increase the number of datagrams in each VDTN bundle.
- Experiment these algorithms in other type of network besides the TCP/IP architecture.
- Extension of the VDTN@Lab laboratory testbed with more mobility models for mobile nodes.
- Evaluation in a larger scale with more network nodes, and a demonstration with a real testbed.

References

References

- [1] Networking for communications challenged communities: Architecture, test beds and innovative alliances. [Online]. Available: <http://www.n4c.eu/>
- [2] Wizzy digital courier - leveraging locality. [Online]. Available: <http://www.wizzy.org.za/>
- [3] M. Asplund, S. Nadjm-Tehrani, and J. Sigholm, "Emerging information infrastructures: Cooperation in disasters," in *Critical Information Infrastructure Security*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5508, pp. 258-270.
- [4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, and R. Durst, "Delay-tolerant networking architecture," RFC 4838, 2007.
- [5] K. Scott and S. Burleigh. undle protocol specification. RFC 5050 (2007). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5050.txt>
- [6] J. Rodrigues, V. Soares, and F. Farahmand, "Stationary relay nodes deployment on vehicular opportunistic networks," *Mobile Opportunistic Networks: Architectures, Protocols and Applications*, 2010.
- [7] V. Soares, F. Farahmand, and J. 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.
- [8] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant networking architecture. RFC 4838 (2007). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4838.txt>
- [9] V. Soares, J. Rodrigues, F. Farahmand, and M. Denko, "Exploiting node localization for performance improvement of vehicular delay-tolerant networks," in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1-5.
- [10] N. Banerjee, M. Corner, and B. Levine, "An energy-efficient architecture for dtn throwboxes," in *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, May 2007, pp. 776-784.

References

- [11] D. Zhang, L. Ge, R. Hardy, W. Yu, H. Zhang, and R. Reschly, "On effective data aggregation techniques in host-based intrusion detection in manet," in *IEEE Consumer Communications and Networking Conference (CCNC 2013)*, Jan 2013, pp. 85-90.
- [12] C. Mysirlidis, P. Galiotos, T. Dagiuklas, and S. Kotsopoulos, "Performance quality evaluation with voip traffic aggregation in mobile ad-hoc networks," in *IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2013)*, June 2013, pp. 1-6.
- [13] C.-M. Huang, C.-C. Yang, and H.-Y. Lin, "A bandwidth aggregation scheme for member-based cooperative networking over the hybrid vanet," in *IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS 2011)*, Dec 2011, pp. 436-443.
- [14] C.-M. Huang, T.-H. Lin, and K.-C. Tseng, "Bandwidth aggregation over vanet using the geographic member-centric routing protocol (gmr)," in *12th International Conference on ITS Telecommunications (ITST 2012)*, Nov 2012, pp. 737-742.
- [15] H. Saleet and O. Basir, "Location-based message aggregation in vehicular ad hoc networks," in *IEEE Globecom Workshops*, Nov 2007, pp. 1-7.
- [16] T.-P. Wang and Y.-C. Chen, "Adaptive packet aggregation for header compression in vehicular wireless networks," in *IEEE 13th International Conference on High Performance Computing and Communications (HPCC 2011)*, Sept 2011, pp. 935-939.
- [17] R. Wang, Z. Wei, Q. Zhang, and J. Hou, "Ltp aggregation of dtn bundles in space communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1677-1691, July 2013.
- [18] Z. Wei, R. Wang, Q. Zhang, and J. Hou, "Aggregation of dtn bundles for channel asymmetric space communications," in *IEEE International Conference on Communications (ICC 2012)*, June 2012, pp. 5205-5209.
- [19] J. Hu, R. Wang, Q. Zhang, Z. Wei, and J. Hou, "Aggregation of dtn bundles for space internetworking systems," *IEEE Systems Journal*, vol. 7, no. 4, pp. 658-668, Dec 2013.
- [20] J. Fajardo, K. Yasumoto, N. Shibata, W. Sun, and M. Ito, "Dtn-based data aggregation for timely information collection in disaster areas," in *IEEE 8th*

References

- International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2012)*, Oct 2012, pp. 333-340.
- [21] I. Signal Processing Magazine, "Self-similar network traffic and performance evaluation [Book Review]," *Signal Processing Magazine, IEEE*, vol. 18, no. 1, pp. 94-95, 2001.
- [22] V. M. Vokkarane and J. P. Jue, "Prioritized burst segmentation and composite burst-assembly techniques for qos support in optical burst-switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 7, pp. 1198-1209, Sep. 2003.
- [23] R. Mahajan, J. Zahorjan, and B. Zill, "Understanding wifi-based connectivity from moving vehicles," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*, New York, NY, USA, 2007, pp. 321-326.
- [24] M. Research. VanLan: investigating connectivity from moving vehicles. [Online]. Available: <http://research.microsoft.com/en-us/projects/vanlan/>
- [25] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden, "A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks," in *2th Annual International Conference on Mobile Computing and Networking (ACM MobiCom 2006)*, Los Angeles, CA, USA, September 24-29 2006, pp. 50-61.
- [26] H.-Y. Huang, P.-E. Luo, M. Li, D. Li, X. Li, W. Shu, and M.-Y. Wu, "Performance evaluation of suvnet with real-time traffic data," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3381-3396, November 2007.
- [27] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: Vehicular content delivery using wifi," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, New York, NY, USA, 2008, pp. 199-210.
- [28] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with cartorrent in a real vehicular ad hoc network testbed," in *IEEE INFOCOM Mobile Networking for Vehicular Environments (MOVE 2007)*, Anchorage, Alaska, May 11 2007, pp. 109-114.
- [29] M. Jerbi, S. M. Senouci, and M. Al Haj, "Extensive experimental characterization of communications in vehicular ad hoc networks within different

- environments,” in *IEEE 65th Vehicular Technology Conference (VTC2007)*, Dublin, Ireland, April 22-25 2007, pp. 2590-2594.
- [30] C. Pinart, P. Sanz, I. Lequerica, D. García, I. Barona, and D. Sánchez-Aparisi, “Drive: A reconfigurable testbed for advanced vehicular services and communications,” in *Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom '08)*, Brussels, Belgium, Belgium, 2008, pp. 1-8.
- [31] UMass Diverse Outdoor Mobile Environment (DOME) project. DieselNet. [Online]. Available: <http://prisms.cs.umass.edu/dome/umassdieselnet>
- [32] H. Soroush, N. Banerjee, A. Balasubramanian, M. D. Corner, B. N. Levine, and B. Lynn, “Dome: A diverse outdoor mobile testbed,” in *Proceedings of the 1st ACM International Workshop on Hot Topics of Planet-Scale Mobility Measurements (HotPlanet '09)*, New York, NY, USA, 2009, pp. 1-6.
- [33] 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, 31 October-3 November 2006, pp. 383-384.
- [34] H. Balakrishnan and S. Madden. Cartel. [Online]. Available: <http://cartel.csail.mit.edu/doku.php>
- [35] J. Ott and D. Kutscher. Drive-thru internet. [Online]. Available: <http://www.drive-thru-internet.org/>
- [36] D. Kutscher and J. Ott, “A disconnection-tolerant transport for drive-thru internet environments,” in *Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Miami, Florida, USA, March 13-17 2005, pp. 1849-1862.
- [37] H. Balakrishnan and S. Madden. Fleet testbed. [Online]. Available: http://cartel.csail.mit.edu/doku.php?id=fleet_testbed
- [38] Object Management Group, “Unified Modeling Language (UML),” 2007. [Online]. Available: <http://www.uml.org/>
- [39] Lego Group, “Lego Mindstorms NXT.” [Online]. Available: <http://mindstorms.lego.com/en-us/Default.aspx>

References

- [40] T. O. B. S. M. Site, "Bluetooth Special Interest Group." [Online]. Available: [https://www .bluetooth.org/apps/content/](https://www.bluetooth.org/apps/content/)
- [41] 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/>
- [42] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "Traffic Differentiation Support in Vehicular Delay-Tolerant Networks," *Telecommunication Systems: Springer*, 2010.
- [43] V. Soares, F. Farahmand, and J. Rodrigues, "Improving vehicular delay-tolerant network performance with relay nodes," in *5th EuroNGI Conference on Next Generation Internet Networks (NGI 2009)*, Aveiro, Portugal, 1-3 July 2009, pp. 1-5.
- [44] M. Rubinstein, F. B. Abdesslem, M. D. Amorim, S. Cavalcanti, R. D. S. Alves, L. Costa, O. Duarte, and M. Campista, "Measuring the capacity of in-car to in-car vehicular networks," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 127-136, 2009.